


Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра захисту інформації

## МАГІСТЕРСЬКА КВАЛІФКАЦІЙНА РОБОТА


на тему: «Інформаційна технологія збору інформації про людину з відкритих джерел»

08-20.МКР.005.00.000 ПЗ

Виконав: студент 2 курсу, групи 1БС-21м  
спеціальності 125 Кібербезпека

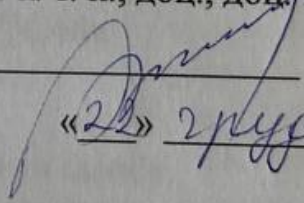
 Наумчак Д. В.

Керівник: к. т. н., доц. каф. ЗІ

 Куперштейн Л. М.

«22» грудня 2022 р.

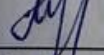
Опонент: к. т. н., доц., доц. каф. ПЗ

 Хошаба О. М.

«22» грудня 2022 р.

Допущено до захисту

Завідувач кафедри ЗІ, д.т.н., проф

 Лужецький В. А.

«22» грудня 2022 р.

Вінниця – 2022 року

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра захисту інформації  
Рівень вищої освіти II (магістерський)  
Галузь знань – 12 «Інформаційні технології»  
Спеціальність 125 – Кібербезпека  
Освітньо-професійна програма – «Безпека інформаційних і комунікаційних систем»

**ЗАТВЕРДЖУЮ**

Зав. кафедри ЗІ, д. т. н., проф.

*В. А. Лужецький*

15 вересня 2022 року



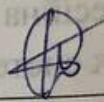



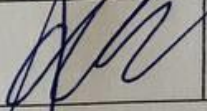

### **ЗАВДАННЯ**

#### **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Наумчаку Дмитру Валерійовичу

1. Тема роботи: «Інформаційна технологія збору інформації про людину з відкритих джерел»,  
керівник роботи: Куперштейн Леонід Михайлович, доц. каф. ЗІ,  
затверджені наказом ВНТУ від 14 вересня 2022 року № 203
2. Строк подання студентом роботи 19 грудня 2022 р.
3. Вихідні дані до роботи:
  - операційна система – Linux;
  - джерело даних – відкриті джерела і витoki даних;
  - середовища розробки – PyCharm, VSCode;
  - мови програмування – Python, Php.
  - фреймворки – Aiogram, Laravel;
  - тестування розробленого програмного засобу.
4. Зміст текстової частини: Вступ. 1. Аналіз літературних джерел за темою роботи. 2. Розробка інформаційної технології. 3. Програмна реалізація інформаційної технології. 4. Економічна частина. Висновки. Список використаних джерел. Додатки.
5. Перелік графічного матеріалу. Складові інформаційної технології (плакат, А4). Схема процесу отримання параметрів для пошуку (плакат, А4). Схема процесу пошуку інформації (плакат, А4). Схема процесу формування результату пошуку (плакат, А4). Загальна архітектура системи (плакат, А4). Загальний алгоритм роботи клієнтської частини (плакат, А4). Загальний алгоритм роботи серверної частини (плакат, А4).

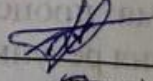
### 6. Консультанти розділів роботи

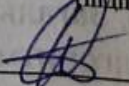
| Розділ | Прізвище, ініціали та посада консультанта        | Підпис, дата   |   |
|--------|--|--|---|
|        |  | завдання видав   | завдання прийняв  |
| 1      | Куперштейн Л. М., к.т.н., доц. кафедри ЗІ        |  |  |
| 2      | Куперштейн Л. М., к.т.н., доц. кафедри ЗІ        |  |  |
| 3      | Куперштейн Л. М., к.т.н., доц. кафедри ЗІ        |  |  |
| 4      | Лесько О. Й., к.е.н., доц. професор кафедри ЕПВМ |  |  |

7. Дата видачі завдання 1 вересня 2022 року.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Зміст етапу  | Строк виконання етапів роботи | Примітка |
|-------|--|-------------------------------|----------|
| 1     | Аналіз завдання. Вступ   | 01.09.2022 – 04.09.2022       |          |
| 2     | Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи | 05.09.2022 – 15.09.2022       |          |
| 3     | Науково-технічне обґрунтування   | 16.09.2022 – 22.09.2022       |          |
| 4     | Аналіз джерел та інструментів OSINT  | 23.09.2022 – 08.10.2022       |          |
| 5     | Аналіз та формування вимог до програмного засобу                             | 09.10.2022 – 16.10.2022       |          |
| 6     | Розробка програмного засобу для збору цифрових слідів                        | 17.10.2022 – 14.11.2022       |          |
| 7     | Тестування розробленого програмного засобу                                   | 15.11.2022 – 17.11.2022       |          |
| 8     | Розробка розділу економічного обґрунтування доцільності розробки             | 18.11.2022 – 21.11.2022       |          |
| 9     | Аналіз виконання ТЗ, висновки  | 22.11.2022 – 24.11.2022       |          |
| 10    | Оформлення пояснювальної записки   | 25.11.2022 – 29.11.2022       |          |
| 11    | Перевірка магістерської роботи на наявність плагіату                         | 30.11.2022 – 02.12.2022       |          |
| 12    | Попередній захист та доопрацювання МКР                                       | 07.12.2022 – 19.12.2022       |          |
| 13    | Представлення МКР до захисту, рецензування                                   | 20.12.2022 – 21.12.2022       |          |
| 14    | Захист МКР   | 22.12.2022 – 26.12.2022       |          |

Студент  Наумчак Д. В.

Керівник роботи  Куперштейн Л. М.

## АНОТАЦІЯ

УДК 004.056

Наумчак Д. В. Інформаційна технологія збору інформації про людину з відкритих джерел. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2022. 134 с.

На укр. мові. Бібліогр.: 60 назв; рис.: 52; табл. 17.

Магістерська кваліфікаційна робота присвячена розробці інформаційної технології збору інформації про людину з відкритих джерел. Проведено аналіз існуючих джерел пошуку інформації з відкритих джерел та існуючих інструментів. Було реалізовано додаток який здійснює пошук інформації про людину з відкритих джерел та цифрових слідів. Також реалізовано панель для адміністрування, завдяки якій зручно переглядати історію запитів та надавати доступ до користування програмним засобом новим користувачам. Виконано тестування програмного засобу.

Ілюстративна частина складається з 7 плакатів з демонстрацією результатів розробки і проведених досліджень

В економічному розділі здійснено оцінку витрат на розробку інформаційної технології.

Ключові слова: інформаційна безпека, OSINT, соціальні мережі.

## **ABSTRACT**

Naumchak D. V. Information technology for collecting information about a person. Master's thesis in the specialty 125 – cybersecurity, educational program - Security of information and communication systems. Vinnytsia: VNTU, 2022. 134 p.

In Ukrainian. Bibliographer: 60 titles; fig .: 52; table. 17.

The Master's thesis is devoted to the development of information technology for collecting information about a person from open sources. An analysis of existing sources of information search from open sources and existing tools was carried out. An application was implemented that searches for information about a person from open sources and digital traces. An administration panel has also been implemented, thanks to which it is convenient to view the history of requests and to grant access to the use of the software to new users. The software tool has been tested.

The illustrative part consists of 7 posters with a demonstration of the results of development and conducted research

In the economic section, an assessment of costs for the development of information technology was made.

Keywords: information security, OSINT, social networks.

## ЗМІСТ

|  |           |
|--|-----------|
| <b>ВСТУП .....</b>   | <b>3</b>  |
| <b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>  | <b>5</b>  |
| 1.1 Аналіз джерел для OSINT розслідувань .....   | 5         |
| 1.2 Аналіз OSINT інструментів .....  | 12        |
| 1.3 Формалізація вимог та постановка задачі .....  | 21        |
| <b>2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....</b>  | <b>23</b> |
| 2.1 Структура інформаційної технології .....   | 23        |
| 2.2 Розробка архітектури системи .....   | 27        |
| <b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ .....</b>   | <b>36</b> |
| 3.1 Обґрунтування вибору програмних засобів для реалізації інформаційної технології.....   | 36        |
| 3.2 Розробка програмного засобу .....  | 39        |
| 3.3 Тестування програмного засобу .....  | 43        |
| <b>4 ЕКОНОМІЧНА ЧАСТИНА.....</b>   | <b>57</b> |
| 4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки .....  | 57        |
| 4.2 Розрахунок узагальненого коефіцієнта якості розробки .....   | 61        |
| 4.3 Розрахунок витрат на проведення науково-дослідної роботи.....  | 63        |
| 4.4 Розрахунок економічної ефективності науково-технічної розробки при її<br>можливій комерціалізації потенційним інвестором ..... | 75        |
| <b>ВИСНОВКИ.....</b>   | <b>80</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>   | <b>82</b> |
| <b>ДОДАТКИ .....</b>   | <b>87</b> |
| Додаток А. Результат перевірки роботи на плагіат .....   | 88        |
| Додаток Б. Лістинг програми.....   | 89        |

## ВСТУП

OSINT (Open Source Intelligence) є важливим напрямом практичної діяльності та наукових розробок у сфері кібербезпеки. Трапляються випадки, коли для отримання доступу до ресурсів не вимагає хакерської атаки, оскільки через непродумані налаштування безпеки вони знаходяться у відкритому доступі. Багато державних інформаційних баз даних перебувають у відкритому доступі; інформація у соціальних мережах, яку користувачі розкривають про себе самі; скомпрометовані інформаційні ресурси, що з'являються в Dark Web. Проблема пошуку інформації про людину з відкритих джерел має велике значення. Таким чином, можна зібрати дані про хобі, інтереси, симпатії, місце проживання і т.д. Існують тисячі загальнодоступних джерел даних, вони присутні в clearnet, darknet, telegram, каналах discord і т.д.

В даний час існують різні інструменти, які проводять аналіз даних за допомогою штучного інтелекту чи машинного навчання, які створюють візуальний профіль людини. Ця інформація дуже важлива, і ці інструменти автоматично збирають цю інформацію, оскільки це забирає багато часу, якщо це робити вручну. Тому розробка інструментів автоматизації OSINT є важливою проблемою. Отже, існує потреба у системі, яка фокусується на автоматичному зборі даних з різних платформ соціальних мереж та витоків даних і забезпечує легкий доступ до всіх даних в одному місці.

Розвідка з відкритих джерел широко визнана найважливішим джерелом цінної та економічно ефективної розвідувальної інформації, одержуваної із загальнодоступних джерел.

З розвитком таких відомих платформ, як Facebook та Twitter, які записують та розкривають безліч різних наборів даних. Деякі основні перешкоди, з якими часто стикаються аналітики OSINT, є обмеження конфіденційності та платформи, які служать для захисту приватного життя людей.

**Об'єктом** дослідження є процес збору інформації про людину.

**Предметом** засоби та методи збору інформації про людину.

**Метою роботи** є розширення функціональних можливостей засобів збору інформації про людину за рахунок комбінованого використання відкритих джерел інформації.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- аналіз наукової літератури щодо існуючих методів та засобів проведення збору інформації про людину з відкритих джерел;
- удосконалення методу збору інформації про людину, шляхом розширення функціональних можливостей;
- удосконалення методу генерації відповіді за заданим запитом, шляхом структурування даних;
- розробка алгоритмів функціонування модулів програмного засобу;
- розробка програмного засобу згідно розроблених алгоритмів функціонування;
- тестування удосконалених методів збору інформації про людину з відкритих джерел.

**Наукова новизна** роботи полягає у вдосконаленні інформаційної технології збору даних про людину з відкритих джерел за рахунок комбінованого використання статичних витоків баз даних та динамічних веб-запитів, що дозволяє розширити функціональні можливості засобів збору даних.

**Практична цінність роботи** полягає у розробці програмного засобу у вигляді telegram-боту, який дозволяє здійснювати збір даних про людину за різними її ідентифікаторами, а саме за прізвищем та іменем, номером телефону, адресою електронної пошти.



# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз джерел для OSINT розслідувань

Соціальні мережі поширені всюди. Важко знайти людину, яка не мала б хоча б одного акаунту в соціальних мережах. Завдяки цьому кожна людина створює цифровий слід. Приблизно 58,4% від кількості населення світу людей мають хоча б один обліковий запис у соціальних мережах [1]. У якийсь момент хтось ставить лайк, в Twitter, ділиться чи прикріплює щось, що може бути помічене громадськістю чи тими, хто пов'язаний із цим акаунтом. Невеликий відсоток людей вмикає максимальні налаштування конфіденційності своїх облікових записів у соціальних мережах, що створює цінне сховище безкоштовної інформації. Ігноруючи це, дослідники ризикують упустити одне з найефективніших відкритих джерел для збору інформації про користувача в Інтернеті.

В інформаційну епоху пошук даних з відкритих джерел прийнято називати Open source intelligence (OSINT) [2]. Зазвичай це загальнодоступна інформація, яку будь-хто може бачити, та, яку ви не можете бачити, незважаючи на її загальнодоступність. OSINT використовується вже сотні років для опису збору розвідувальної інформації за допомогою загальнодоступних ресурсів. OSINT був запроваджений під час Другої світової війни як інструмент збору розвідданих державними службами безпеки, але зовсім недавно, з бурхливим зростанням інтернет-комунікацій та величезного обсягу цифрових даних, OSINT став необхідністю для різних видів організацій [2]. Наприклад, урядові відомства, неурядові організації та бізнес-корпорації починають значною мірою покладатися на OSINT як на додаток до приватної та секретної інформації. Джерела OSINT відрізняються від інших форм розвідки тим, що вони мають бути легально доступними громадськості без порушення авторських прав або законів про конфіденційність. Тобто будь-яка інформація може бути знайдена як онлайн,

так і офлайн, в ефірі і на папері. Ця різниця робить здатність збирати джерела OSINT застосовною не тільки для служб безпеки. OSINT включає всі загальнодоступні джерела інформації, і ця інформація може бути знайдена як онлайн, так і офлайн, в ефірі і на папері. Ви можете збирати OSINT з: Інтернету, включаючи форуми, блоги, соціальні мережі, сайти обміну відео, записи Whois зареєстрованих доменних імен, метадані та цифрові файли, ресурси Dark Web, дані геолокації, IP-адреси, пошукові системи та все, що можна знайти у мережі. Традиційні засоби масової інформації, включаючи телебачення, радіо, газети, книги, журнали, спеціалізовані журнали, наукові публікації, дисертації, матеріали конференцій, профілі компаній, річні звіти, новини компаній, профілі співробітників та резюме. Метадані на фотографіях та відео, а також інформація з карт. Інформацію з відкритих джерел можна зібрати практично звідусіль, і навіть на перший погляд неочевидні джерела можуть надати досить цінні відомості про об'єкт дослідження.

Розрізняють наступні джерела для OSINT розслідувань [3, 4]:

- новини та ЗМІ;
- сіра література;
- соціальні мережі;
- Dark Web.

Новини та ЗМІ включають публікації ЗМІ, радіо, телебачення, контент медіа-агрегаторів, книги та інші форми друкованих або традиційних ЗМІ [3, 4]. Новини та дані ЗМІ публікуються як у відкритих джерелах, так і за окрему плату, і поширюються у міжнародному, національному, регіональному та місцевому масштабах. Навіть місцеві новини можуть мати вирішальне значення, надаючи аналітикам розвідки багатий звіт про події, що відбуваються безпосередньо поблизу джерела.

Новини та ЗМІ перебувають у стані постійного руху: щодня публікуються мільйони матеріалів на різних мовах. Навігація за цими джерелами за допомогою звичайних методів пошуку та ручної обробки даних практично неможлива, особливо при виявленні зв'язків між контентом новин і

даними з інших джерел, наприклад, фінансовими звітами та документами компаній.

Оскільки контент новин часто використовується в різних мережах у різних форматах, визначення дублюючого контенту також є серйозною проблемою без допомоги спеціально розроблених інструментів OSINT. Більше того, відрізнити достовірні новини від фейкових новин чи інших ненадійних ЗМІ простіше, коли слідчі можуть правильно зрозуміти походження новин та джерел ЗМІ.

Дані, зібрані з новин та ЗМІ, забезпечують історичний контекст для широкого спектру подій, що відбуваються на різних рівнях. В результаті новини та ЗМІ мають три чіткі варіанти використання в контексті розслідувань OSINT [4]:

- Розкриття мереж: Новини та ЗМІ можуть ставитись до окремих осіб, організацій та подій, і часто можуть виявляти нові мережі або співвідносити мережі, виявлені з інших джерел, таких як повідомлення в соціальних мережах. Це може допомогти слідчим краще зрозуміти події та окремих людей. Наприклад, в новинах можуть бути зазначені особи, присутні в сюжеті або мають до нього відношення, чи то злочин, протест чи щось інше, на ім'я, зображення або відео. Ці ідентифікатори можна порівняти з додатковими даними, щоб скласти карту мереж та зв'язків.
- Складання хронології: ОСД за новинами та ЗМІ особливо корисний для відображення хронології новинних історій та подій, допомагаючи слідчим реагувати на історії, що виникають, що загрожують чиєїсь цілісності чи безпеці.
- Перевірка цілісності: Розслідування несприятливих новин також є дуже корисним при перевірці потенційних клієнтів на наявність ризиків чи питань сумлінності, незалежно від того, чи є вони приватними особами, підприємствами, організаціями чи державними відомствами.

Сіра література включає всі види загальнодоступної інформації про політику приватного і державного секторів, що не відноситься до ЗМІ[3, 5]. Сюди входять документи та звіти благодійних організацій, міжурядових інститутів та аналітичних центрів, а також статистика злочинності, дані перепису населення та інформація, що міститься в академічних базах даних, журналах та звітах.

Сіра література також включає річні звіти компаній, дані про файли та витoki інформації [6]. Як приклад можна навести виток даних, зібраних авторитетними організаціями, такими як Organised Crime and Corruption Reporting Project (OCCRP) або International Consortium of Investigative Journalists (ICIJ), які нещодавно викрили офшорну фінансову систему, яку використовують багато світових лідерів, глав держав, знаменитостей та керівниками підприємств у документі Pandora Papers [7]. Ці джерела OSINT густо наповнені добре вивченими даними, які найчастіше важко оцінити кількісно.

Основна проблема полягає в тому, що ця інформація зазвичай є платною або вимагає введення даних для доступу. Наприклад, близько 42% досліджень у галузі глобальної охорони здоров'я в даний час публікуються за стінами платної системи [8]. Це відбувається тому, що сіра література в основному існує в так званій "глибокій павутині" - частині Інтернету, яка не виявляється на стандартних сторінках результатів пошукових систем.

Сіра література часто використовується для поширення та передачі кількісних та якісних даних між підприємствами. Тому вона багата даними через свою структуру, що дозволяє слідчим отримати критично важливий контекст розслідування [6].

Корпоративні документи є ключовим об'єктом збору OSINT, надаючи інформацію про ділові операції, документи та мережеві зв'язки між різними учасниками бізнесу та пов'язаними з ними організаціями. Політичні документи та звіти можуть бути співвіднесені із фінансовими документами для виявлення невідповідностей. Така інформація може бути зіставлена з

витоком фінансових документів для збагачення та перехресного зіставлення корпоративних документів, що робить сіру літературу найбільш корисною у розслідуваннях щодо відмивання грошей та відстеження активів [6, 9].

Соціальні мережі можуть охоплювати весь спектр довгострокового контенту (наприклад, повідомлення Reddit, довгострокові пости в блогах соціальних медіа, відповіді Quora), а також короткостроковий контент (твіти, оновлення LinkedIn, підписи в Instagram), фотографії, теги та зв'язки першого та другого ступенів. Метадані, пов'язані з контентом соціальних мереж, відіграють центральну роль у OSINT, допомагаючи візуалізувати мережу та зрозуміти хронологію взаємопов'язаних подій. Дані соціальних мереж також дуже наочні та включають масу зображень та відео, часто створених у безпосередній близькості від їхнього об'єкта [10]. Для прикладу, можна отримати величезну кількість особистої інформації про будь-яку людину по всьому світу, просто переглянувши її сторінку у Facebook. Така інформація часто включає в себе зв'язки людини, її політичні погляди, релігію, етнічну приналежність, країну походження, особисті фотографії та відео, сімейний стан, домашню і робочу адреси, інформації про її громадську діяльність, які заклади відвідує, історію про місця роботи, освіту, дати важливих подій (наприклад, дата народження, дата закінчення школи, дата відносин або дата догляду/початку нової роботи) і т.д.

Більше половини населення планети сьогодні використовують соціальні мережі у тій чи іншій формі, і, як наслідок, вони мають величезну глибину та обсяг, що робить ручний пошук виключно трудомістким [11]. Більшість найкориснішої інформації із соціальних мереж не індексується і знаходиться в глибокій павутині, що робить стандартні поверхневі веб-браузери недостатніми. На щастя, спеціалізовані інструменти OSINT допомагають слідчим аналізувати ці дані та виявляти зв'язки з високою швидкістю, скорочуючи більшу частину ручної роботи.

Крім того, збереження анонімності має вирішальне значення для будь-якого слідчого з соціальних мереж. Розслідування щодо окремих осіб або

мереж повинні залишатися непоміченими і не викликати жодних сигналів, які могли б попередити об'єкт про розслідування.

Величезний обсяг даних, доступних на платформах соціальних мереж, робить їх потужним джерелом розслідувань OSINT. Він може застосовуватись у різних умовах розслідування, включаючи [12]:

- Розвідка загроз: Рівень даних від людини до людини, доступних через соціальні мережі, недоступний ніде в іншому місці, і забезпечує не поверховий спосіб моніторингу інформації про недавню діяльність, місцезнаходження чи комунікації загрозливого суб'єкта - інформації, яка може дати уявлення про попередню або заплановану ворожу діяльність.
- Візуалізація особистих зв'язків: Невід'ємною частиною соціальних мереж є зв'язки, що утворюються між акаунтами та їх відповідними повідомленнями та контентом – вони допомагають слідчим візуалізувати зв'язки всередині та між мережами.

Завдяки своїй здатності до розвідки загроз та візуалізації зв'язків, соціальні мережі можуть застосовуватись у різних контекстах розслідування OSINT [12]. До них відносяться внутрішні загрози, хабарництво та корупція, оскільки соціальні мережі дозволяють слідчим виявити несподівані зв'язки співробітника чи клієнта зі зловмисником, які інакше залишилися б таємницею [13].

Dark Web – це термін, який використовується для позначення веб-сторінок, які не індексуються і потребують спеціалізованого програмного забезпечення для отримання доступу [14]. В Dark Web користувачі та оператори залишаються невідслідковуваними, тому вона є багатим джерелом даних про злочинні мережі, їх діяльність та зв'язки. Імена користувачів, адреси та інші сигнали та ідентифікатори мають неоціненне значення для формування перехресних зв'язків з інформацією з поверхневого чи глибокого Інтернету, допомагаючи аналітикам спецслужб виявляти зв'язки між обліковими записами та профілями [14].

Dark Web – це унікальне середовище даних, яке повністю не індексується стандартними пошуковими системами. Крім того, тут зберігається така інформація як юридичні документи, наукові дослідження, академічні праці та бази даних приватних установ. Аналітики розвідки повинні бути обережними, щоб не розкрити свою особистість і не видати своє розслідування, а також триматися подалі від шкідливих програм або нелегальних носіїв інформації. Збір необхідної інформації з Dark Web при дотриманні нормативно-правових вимог є критично важливим і вимагає наявності спеціальних інструментів OSINT, які дозволяють слідчим безпечно обробляти інформацію, не наражаючи себе на ризик [15]. Крім того, слідчі повинні використовувати ці інструменти OSINT, щоб уникнути непотрібного контакту з потенційно незаконними матеріалами під час перегляду темного павутиння.

Dark Web надає аналітикам спецслужб ключову можливість для виявлення прямих зв'язків між злочинною діяльністю та пов'язаними з нею іменами користувачів, адресами та іншими ідентифікаторами. Ці дані можуть бути зіставлені та звірені з даними, знайденими в Clear Web, що дозволяє правоохоронним органам та аналітикам загроз спостерігати та виявляти злочинні мережі, збирати докази та відстежувати комунікації.

Dark Web може бути цінним джерелом розвідданих про погрози, де аналітики можуть дізнатися про способи проведення кібератак, викрадених даних, про те, які інструменти для атак продаються та купуються, а також про рівень успіху поточних кібератак. Однак пошук такої інформації та створення повної картини загроз може бути складним і потребує глибокого розуміння Dark Web та того, як проводити розслідування.

Все це робить Dark Web придатною для використання в різних видах розслідувань OSINT [16]. Сюди входять розслідування щодо продажу нелегальної продукції, наприклад, зброї, а також відстеження мереж торгівлі забороненими речовинами та дикими тваринами [17].

## 1.2 Аналіз OSINT інструментів

Інструменти OSINT можуть ідентифікувати та розділяти об'єкти в наборі даних (розбір), а також відображати ці об'єкти за категоріями, щоб отримати сенс і уникнути надмірності (нормалізація). Інструменти OSINT також можуть індексувати необроблені дані, щоб їх можна було швидко та легко шукати та фільтрувати за релевантністю [18].

Доступ до загальнодоступних даних часто безкоштовний, але справжня цінність полягає в тому, що саме можна проаналізувати та витягти з цих даних. Організаціям, які використовують OSINT для забезпечення безпеки та розвідки, необхідна здатність швидко та ефективно виявляти ключову інформацію [19].

Величезна кількість даних в Інтернеті не піддається просіюванню, завдяки складним способам дій сучасних суб'єктів онлайн загроз, вразливі місця організацій стають дедалі невловимішими. Дані з відкритих джерел при ефективному зборі, збагаченні та моніторингу можуть бути надзвичайно цінними для прогнозування, аналізу та розгляду інцидентів на кожному етапі їх виникнення.

Також слід зазначити переваги у використанні OSINT інструментів [20]:

- Однією з найбільших переваг використання OSINT це вартість: OSINT набагато дешевша порівняно з традиційними інструментами збору інформації. OSINT пропонує потенційно більшу окупність інвестицій, і ця особливість особливо актуальна для організацій з обмеженим бюджетом на розвідку.
- Інформацією можна легально та легко ділитися з ким завгодно; відкриті джерела завжди доступні та постійно оновлюються за будь-якою темою. Twitter, наприклад, є відкритим та доступним.
- Інформація, зібрана з відкритих джерел, є відмінним джерелом розвідданих для національної безпеки і може бути використана для підтримки створення довгострокових стратегій для різних бізнес-цілей.



З іншого боку, OSINT не позбавлений своїх недоліків [20, 21]:

- Однією з найбільших проблем OSINT є потенційне інформаційне навантаження; Відфільтрувати інформацію від "шуму" може бути складно. Насправді, без інструментів OSINT та пошук потрібної інформації може зайняти багато часу;
- OSINT вимагає великого обсягу аналітичної роботи з боку людей, щоб відрізнити достовірну, перевірену інформацію від фальшивих, які вводять в оману або просто неточних новин та інформації;
- Для отримання максимальної користі від OSINT потрібен детальний аналіз та розуміння вимог щодо його використання. Більше того, вибір використання OSINT не повинен ґрунтуватися на вартості порівняно з традиційною розвідкою, оскільки OSINT не виключає традиційну розвідку, зібрану із секретних джерел. Поєднання OSINT та традиційних джерел розвідданих є потужним підходом для бізнес-аналітики.

Нижче наведено списки популярних інструментів таблиці. В таблиці 1.1 вказані telegram-боти, в таблиці 1.2 наведено онлайн сервіси та пошукові системи і в таблиці 1.3 присутні утиліти.

Таблиця 1.1 – OSINT Telegram-боти

| Назва       | Характеристика  |
|-------------|---|
| Get Contact | Бот показує, як записаний номер мобільного телефону в адресній книзі інших абонентів. Інформацію отримується з мобільних пристроїв абонентів, які встановили програму Get Contact. У день бот дозволяє надіслати не більше трьох телефонних номерів [22]. |
| AVinfo      | Бот для перевірки даних про історію автомобіля за номером авто, VIN, номером ПТС, ПІБ або телефоном власника. З його допомогою неважко встановити власника транспортного засобу. Можливості у бота досить широкі, але є платним [23].                     |

Продовження таблиці 1.1

| Назва           | Характеристика   |
|-----------------|--|
| Quick_OSINT_bot | За своїми можливостями бот в схожий на SmartSearchBot – він здатний шукати по телефону, адресою електронної пошти, номером авто або VIN, номеру паспорта, СНІЛС або ПІН. Також даний бот має пошук за паролем: бот шукає в злитих базах адресу email, якому може відповідати заданий пароль. Робот дозволяє виконати кілька безкоштовних запитів на добу, після чого просить оплатити підписку [25]. |
| TeleSINT        | Безкоштовний бот, що дозволяє знайти користувача по ніку, яких групах в «Telegram» є учасником той чи інший користувач.<br>Значний недолік цього бота полягає в тому, що він показує користувача тільки у відкритих групах, а закриті групи залишаються поза його полем зору [26].   |
| GetFB           | Безкоштовний бот, за введеним номером мобільного телефону в міжнародному форматі (+7XXXAAABVCC) GetFB знаходить зареєстровану на цей номер сторінку у Facebook [27].   |
| Mailsearchbot   | Бот шукає по злитих базах паролі, що відповідають адресам електронної пошти, логінам або номеру телефону. Безкоштовно бот показує лише частину виявлених паролів, а за повною версією перенаправляє на сторонній сайт, де користувачу з пропонують оформити підписку [28].   |
| Opendatabot     | Сервіс пошуку в державних даних компаній, людей, авто та нерухомості. Даний бот шукає у всіх реєстрах за кодом ЄДРПОУ, назвою компанії, ПІБ людини та номером авто [29].   |

З вище наведених OSINT Telegram-ботів впливає, що всі вони різняться своїми можливостями та призначенням. Та можна дійти висновку що:

- більшість із них — платні, більшість з них майже ніяк не структурує дані, знайдену інформацію доводиться аналізувати та перебирати вручну;
- з даного списку є декілька безкоштовних інструментів, але вони мають обмеження на кількість запитів.

Таблиця 1.2 – Онлайн сервіси та пошукові системи для OSINT

| Назва          | Характеристика                         |
|----------------|--|
| Стоп-лист.инфо | Пошук по ніку, логіну, юзернейму [30]. |

## Продовження таблиці 1.2

|                |   |
|----------------|---|
| Webresolver.nl | Пошук у злитих даних за ім'ям, прізвищем, e-mail, логіну, юзернейму, IP адресу [31].  |
| Saverudata     | Пошук у злитих даних за номером телефону, прізвищем, e-mail, та юзернейму Pkabu [32].   |
| Intelligence X | Даний сервіс дозволяє здійснювати пошук за e-mail, номером телефону, MAC-адресою, номером кредитної картки, IBAN, ПІН і тд [33].  |
| Shodan         | Пошукова система, що дозволяє користувачам шукати різні типи серверів (веб-камери, маршрутизатори, сервери тощо), підключених до мережі Інтернет, з використанням різних фільтрів [34]. |
| Мамонт         | Мультисервісна пошукова система. Пошук в Інтернеті за оригінальним контентом. Перша історія Інтернету глобальна пошукова система по FTP архівам [35].                                   |
| Web archive    | web.archive.org – це сервіс за допомогою якого можна вивчити історію домену з моменту першого попадання в базу сервісу [36].  |
| CiteSeerX      | Електронна бібліотека та пошукова машина з наукових публікацій [37].  |
| Lookup-ID.com  | Знаходить ідентифікатор групи Facebook та ідентифікатор сторінки Facebook [38].   |
| Spoonbill      | Даний сервіс дає змогу бачити зміни в профілі людей, на які користувач підписався в Twitter або інших соціальних мережах [39].  |

Серед вище наведених сервісів та пошукових слід виділити наступні:

- webresolver.nl;
- saverudata;
- Intelligence X.

Сервіс webresolver.nl дозволяє здійснювати пошук по злитим базам даних і безкоштовно здійснювати пошук за параметрами які пов'язані з Skype. Також є додаткові можливості перевірки геолокації за IP-адресою (рис. 1.1), отримання IP-адреси сайтів котрі використовують CloudFlare (рис. 1.2), DNS резолвери (рис. 1.3) та інші.

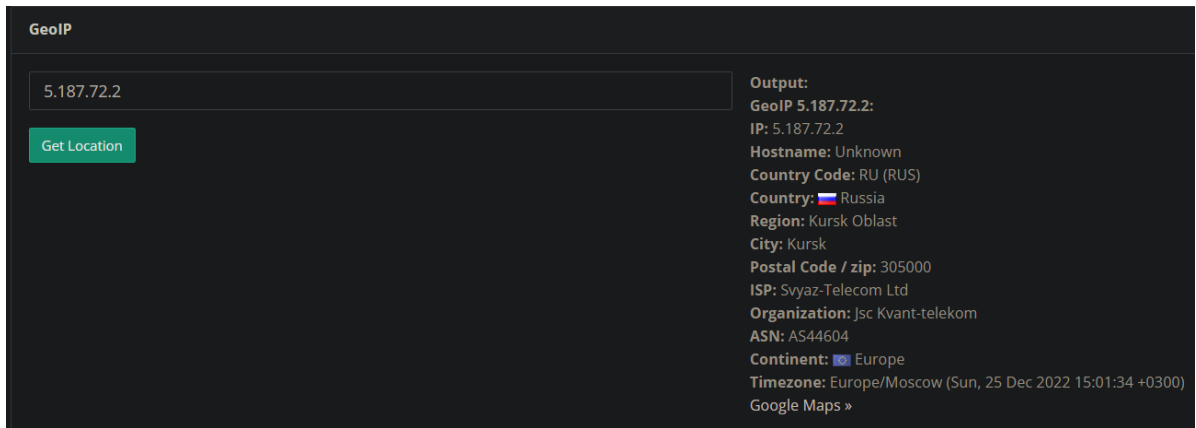


Рисунок 1.1 – Перевірка геолокації за IP-адресою

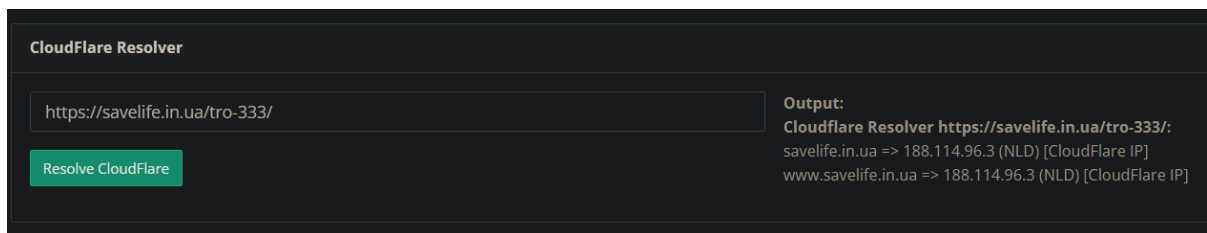


Рисунок 1.2 – Перевірка геолокації за IP-адресою

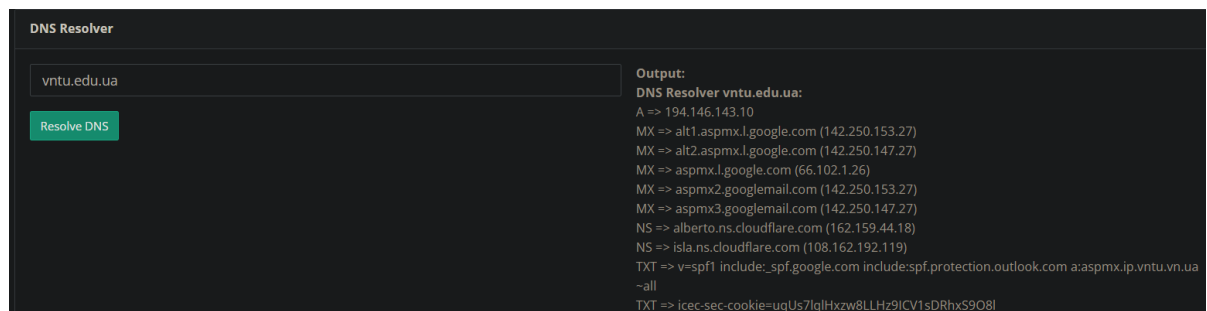


Рисунок 1.3 – Приклад результату DNS резолву за заданою адресою

Сервіс Saveridata здійснює пошук інформації з витоків даних, лише за чотирма параметрами:

- номер телефону;
- прізвище;
- нікнейм Pkabu;
- електронна пошта.

**Поиск по:**

Номеру (к примеру 79275192091):

Фамилии (к примеру Яценко):

Рiкaби никнейм (к примеру lurkeen):

Email:

Телефон: 79275192091

- 📍 **Имя:** Анжела
- 📍 **Яндекс Еда:** потрачено за 6 мес 32685 руб
- 📍 **Устройство:** Android
- 📍 **Модель устройства:** MagnitApp\_Android/1.2.6 [распознать](#)
- 📍 **Заказы:** Краснодар Колхозная улица дом 26 кв 42, подъезд 1справаВоДворе, этаж 4
- 📍 1. 2021-06-19T14:20:05.000Z - 645 руб
- 📍 2. 2021-06-24T21:06:43.000Z - 720 руб
- 📍 3. 2021-06-26T17:06:06.000Z - 649 руб
- 📍 4. 2021-06-28T12:26:57.000Z - 1001 руб
- 📍 5. 2021-07-01T15:09:54.000Z - 0 руб
- 📍 6. 2021-07-01T15:12:31.000Z - 625 руб

Рисунок 1.4 – Пример результата поиска по номеру телефона

Рiкaби никнейм (к примеру lurkeen):

Email:

Телефон: 79155708730

- 📍 **Имя:** лари
- 📍 **Яндекс Еда:** потрачено за 6 мес 1911 руб
- 📍 **Устройство:** iPhone
- 📍 **Модель устройства:** ios(5,15.0) [распознать](#)
- 📍 **Заказы:** Воронеж Большая Манежная улица дом 13В кв 77, подъезд 1, этаж 13, домофон 77
- 📍 1. 2021-07-24T00:23:50.000Z - 0 руб
- 📍 2. 2021-07-24T00:24:44.000Z - 0 руб
- 📍 3. 2021-07-24T00:25:19.000Z - 1134 руб
- 📍 **Заказы:** Белгород Преображенская улица дом 78Б кв 83, подъезд 2, этаж 15, домофон 83
- 📍 1. 2021-09-25T22:14:55.000Z - 777 руб
- 📍 **Имя:** илья
- 📍 **Delivery Club:** потрачено 955 руб
- 📍 **Заказы:** Белгород Преображенская улица дом 78Б кв 83, подъезд 2, этаж 15, домофон 83 (Белгород, Преображенская улица, 78Б)
- 📍 1. 2021-06-09T23:03:00+00:00 - 955 руб (Папа Джонс)
- 1шт Салат "Греческий", 1шт Сок "Я" Яблочный, 0.97 л; 1шт Aqua Minerale Газированная, 0.5 л; 1шт Барбекью, 25 мл; 1шт Веганская с овощами , 23 см;
- 📍 IP: 213.87.145.67
- 📍 **Имя в СДЭК:** Васильев Илья Алексеевич
- 📍 **Email:** anomaly-pk@mail.ru
- 📍 **Никнейм на Пикабу:** lurkeen
- 📍 **Email:** anomaly-pk@mail.ru

Рисунок 1.5 – Пример результата поиска по никнейму Рiкaби

Серед недоліків даного сервісу слід зазначити те, що вибірка даних досить мала, із за кількості злитих баз даних які використовуються цим сервісом. Нижче наведено список використаних витоків даних які використовуються для пошуку на даному сервісі:

- «СДЭК»;
- Wildberries;
- Avito;
- «Билайн»;

– «Яндекс еда».

Сервіс Intelligence X має багато параметрів, за якими здійснює пошук інформації. Даний сервіс дає змогу переглянути за заданим параметром пошуку, джерела виток даних та додаткові дані. Але головний недолік полягає в тому, що повна інформація про виток доступна лише користувачам котрі оформили підписку на даний сервіс. На рисунку 1.6 наведено приклад пошуку інформації за електронною поштою в Intelligence X.

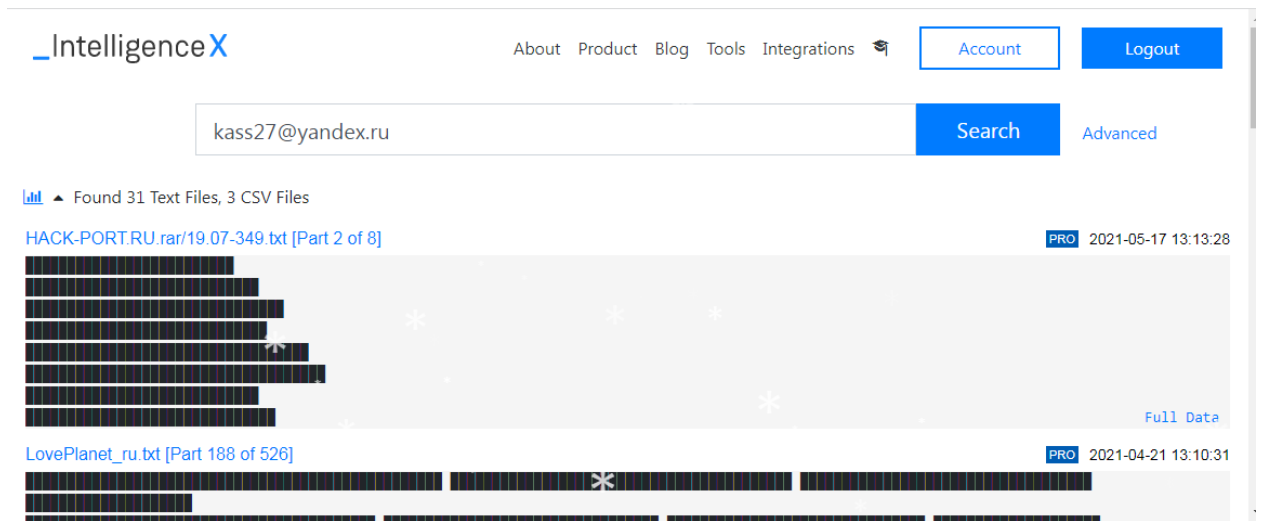


Рисунок 1.6 – Приклад результату пошуку за електронною поштою

З наведених онлайн сервісів та пошукових систем, можна дійти висновку що:

- багато з представлених сервісів працюють лише за більшість із них — платні, більшість з них майже ніяк не структурує дані, знайдену інформацію доводиться аналізувати та перебирати вручну;
- певна частина з даного списку сервісів здійснюють пошук інформації по старим витокам даних, які є вже неактуальні, чи взагалі мають досить малу вибірку даних або мають обмежений функціонал.

Таблиця 1.3 – Список утиліт

| Назва  | Характеристика   |
|--------|--|
| Gideon | Інструмент для пошуку за номером телефону, автомобілю, завантажень torrent за IP адресою, пошук за ніком (логіном) [40]. |

## Продовження таблиці 1.3

| Назва        | Характеристика   |
|--------------|--|
| sn0int       | sn0int напівавтоматично обробляє загальнодоступну інформацію та відображає результати в єдиному форматі для подальших розслідувань.<br>Дана утиліта здійснює пошук:<br>- Акаунтів (Instagram, Github, Keybase, Twitch, Steam, npm, Soundcloud, Tiktok, Tellonym, Tinder, Fediverse, namechk, Chaturbate, Venmo, Patreon, Deviantart, Protondb, letterboxd, chefkoch, Periscope);<br>- доменів;<br>- ip-адрес;<br>- e-mail;<br>- номеру телефону;<br>- зображення, фото [41]. |
| Maigret      | Maigret збирає досьє на людину лише за іменем користувача, перевіряючи облікові записи на величезній кількості сайтів і збираючи всю доступну інформацію з веб-сторінок [42].  |
| Sherlock     | Потужний консольний інструмент, здійснює пошук за іменем користувача у багатьох соціальних мережах [43].   |
| GHunt        | GHunt — це модульний інструмент, який містить багато методів для дослідження облікових записів або об'єктів Google. Включає в себе модулі електронної пошти, документів, youtube і gaia [44].  |
| Osintgram    | Потужний інструмент для аналізу облікового запису Instagram будь-якого користувача за його ніком [45].   |
| Maltego      | Інструмент для збору інформації та об'єднання її для графічного кореляційного аналізу [46].  |
| Datasploit   | Інструмент для виконання різних методів OSINT для імен користувачів, адрес електронної пошти та доменів [47].  |
| Orbit        | Встановлює зв'язки між криптогаманцями за допомогою рекурсивного сканування історії транзакцій [48].   |
| Osintstalker | Скрипт Python для OSINT Facebook і геолокації [49].  |
| FOCA         | Інструмент для пошуку метаданих і прихованої інформації в документах [50].   |

Серед вище наведених утиліт слід виділити наступні:

- Maigret;
- sn0int;

– Sherlock.

Maigret – це автоматизований інструмент, який здійснює пошук за нікнеймом в понад 3000 сайтах. Результати зберігаються у вигляді файлів звітів (.pdf та .html). Приклад процесу пошуку наведено нижче.

```
$ python3 maigret --ids --print-found --skip-errors alexaimphotographycars
[*] Checking username alexaimphotographycars on:
[+] 500px: https://500px.com/p/alexaimphotographycars
  |uid: dXJpOmsvZGU6VXN1cjoyNjQwMzQxNQ==
  |legacy_id: 26403415
  |username: alexaimphotographycars
  |name: Alex Aimé
  |website: www.flickr.com/photos/alexaimphotography/
  |facebook_link: www.instagram.com/street.reality.photography/
  |instagram_username: alexaimphotography
  |twitter_username: Alexaimphotogr
[*] Checking username alexaimphotography on:
[+] DeviantART: https://alexaimphotography.deviantart.com
  |country: France
  |registered_for_seconds: 55041733
  |gender: male
  |username: Alexaimphotography
  |twitter_username: alexaimphotogr
  |website: www.instagram.com/alexaimphotography/
  |links:
  |   | https://www.instagram.com/alexaimphotography/
[+] EyeEm: https://www.eyeem.com/u/alexaimphotography
  |eyeem_id: 21974802
  |eyeem_username: alexaimphotography
  |fullname: Alex
  |followers: 10
  |friends: 2
  |liked_photos: 37
  |photos: 10
  |facebook_uid: 1534915183474093
[+] Facebook: https://www.facebook.com/alexaimphotography
[+] Gramho: https://gramho.com/explore-hashtag/alexaimphotography
[+] Instagram: https://www.instagram.com/alexaimphotography
  |username: alexaimphotography
  |full_name: Alexaimphotography
  |id: 6828488620
  |biography: IT MF DE
  |Amateur photographer
  |Follow me @street.reality.photography
  |Sony A7ii
  |external_url: https://www.flickr.com/photos/alexaimphotography2020/
```

Рисунок 1.7 – Процес пошуку за нікнеймом

sn0int – утиліта, котра здійснює пошук в соціальних мережах. Також присутній пошук за номером телефону, електронною поштою і тд.

Sherlock – інструмент, який як і Maigret здійснює пошук за нікнеймом. Звіти формуються у зручному форматі. Приклад процесу пошуку наведено нижче.



```
[+] Spotify: https://open.spotify.com/user/jcrodriguez
[+] Strava: https://www.strava.com/athletes/jcrodriguez
[+] Telegram: https://t.me/jcrodriguez
[+] TikTok: https://tiktok.com/@jcrodriguez
[+] TrackmaniaLadder: http://en.tm-ladder.com/jcrodriguez_rech.php
[+] TradingView: https://www.tradingview.com/u/jcrodriguez/
[+] Trakt: https://www.trakt.tv/users/jcrodriguez
[+] Trello: https://trello.com/jcrodriguez
[+] Twitch: https://www.twitch.tv/jcrodriguez
[+] Unsplash: https://unsplash.com/@jcrodriguez
[+] VK: https://vk.com/jcrodriguez
[+] VSCO: https://vSCO.co/jcrodriguez
[+] Venmo: https://venmo.com/jcrodriguez
[+] VirusTotal: https://www.virustotal.com/ui/users/jcrodriguez/trusted_users
[+] Wattpad: https://www.wattpad.com/user/jcrodriguez
[+] Windy: https://community.windy.com/user/jcrodriguez
[+] YouNow: https://www.younow.com/jcrodriguez/
[+] fixya: https://www.fixya.com/users/jcrodriguez
[+] geocaching: https://www.geocaching.com/p/default.aspx?u=jcrodriguez
[+] hackster: https://www.hackster.io/jcrodriguez
[+] interpals: https://www.interpals.net/jcrodriguez
[+] kofi: https://ko-fi.com/jcrodriguez
[+] last.fm: https://last.fm/user/jcrodriguez
[+] mercadolibre: https://www.mercadolibre.com.br/perfil/jcrodriguez
[+] nairaland.com: https://www.nairaland.com/jcrodriguez
[+] npm: https://www.npmjs.com/~jcrodriguez
[+] osu!: https://osu.ppy.sh/users/jcrodriguez
```

Рисунок 1.8 – Процес пошуку за нікнеймом

Як видно з вище наведених таблиць існує безліч різних OSINT інструментів, які можна використовувати, але всі вони різняться категоріями, функціоналом і можливостями.

### 1.3 Формалізація вимог та постановка задачі

Виконавши аналіз даних, можна зробити висновок, що OSINT відіграє важливу у багатьох галузях. Основна мета використання інструментів – виявлення будь-якої потенційно підозрілої чи незаконної діяльності. Вони також використовуються компаніями для захисту себе та своїх клієнтів від витоків даних та порушення конфіденційності. Але основними недоліками проаналізованих вище інструментів є те, що вони мають обмежений функціонал та майже ніяк не структурують знайдені дані.

Головною метою даної роботи є розширення функціональних можливостей засобів збору інформації про людину за рахунок комбінованого використання відкритих джерел інформації.

Для досягнення поставленої мети необхідно:

- удосконалення методу збору інформації про людину, шляхом розширення функціональних можливостей;

- удосконалення методу генерації відповіді за заданим запитом, шляхом структурування даних;
- розробити алгоритми функціонування інформаційної технології;
- розробка програмного засобу згідно розроблених алгоритмів функціонування;
- тестування удосконалених методів збору інформації про людину з відкритих джерел;
- зробити висновки про виконану роботу.

Програмний засіб має відповідати наступним вимогам, відповідно до вихідних даних розробки:

- забезпечувати логування запитів користувачів;
- мати змогу формувати звіти на основі запитів користувачі.

Для реалізації клієнтської частини було запропоновано використання фреймворку Aiogram, а для серверної частини – фреймворку Laravel.

## 2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 2.1 Структура інформаційної технології

Інформаційна технологія – це комплекс методів і засобів, завдяки яким реалізується певна задача. Інформаційна технологія складається з процесів, що являють собою незалежні частини. Такий підхід дозволяє виявляти проблеми на ранніх стадіях розробки та швидко їх вирішувати.

Інформаційна технологія збору інформації про людину з відкритих джерел складається із наступних процесів:

- отримання параметрів – зчитування вхідних параметрів та визначення типу параметрів для пошуку;
- пошук інформації – пошук з відкритих джерел та витоків даних, аналіз знайденої інформації та її структуризація;
- формування результату пошуку за заданими даними.

В кожен процес подаються певні дані на вхід, які обробляються та надсилаються далі. Загальна схема інформаційної технології збору інформації про людину зображена на рисунку 2.1.

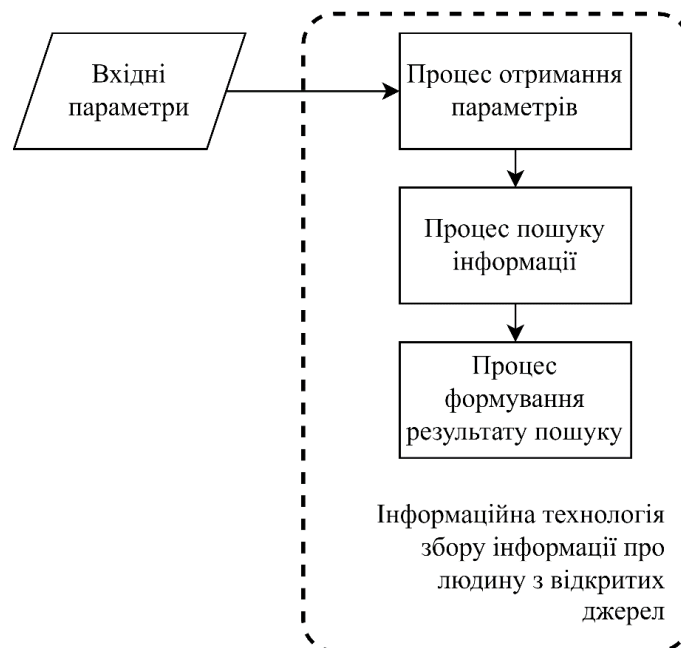


Рисунок 2.1 – Структура інформаційної технології збору інформації про людину з відкритих джерел

Процес отримання параметрів включає в себе два етапи. Перший етап полягає в тому, що на вхід надходять параметри для здійснення пошуку які вводить користувач, це може бути номер телефону, електронна пошта чи повний ПІБ з датою народження. Далі отримані параметри надходять до другого етапу, де визначається їх тип і передаються до наступного процесу (рис. 2.2).

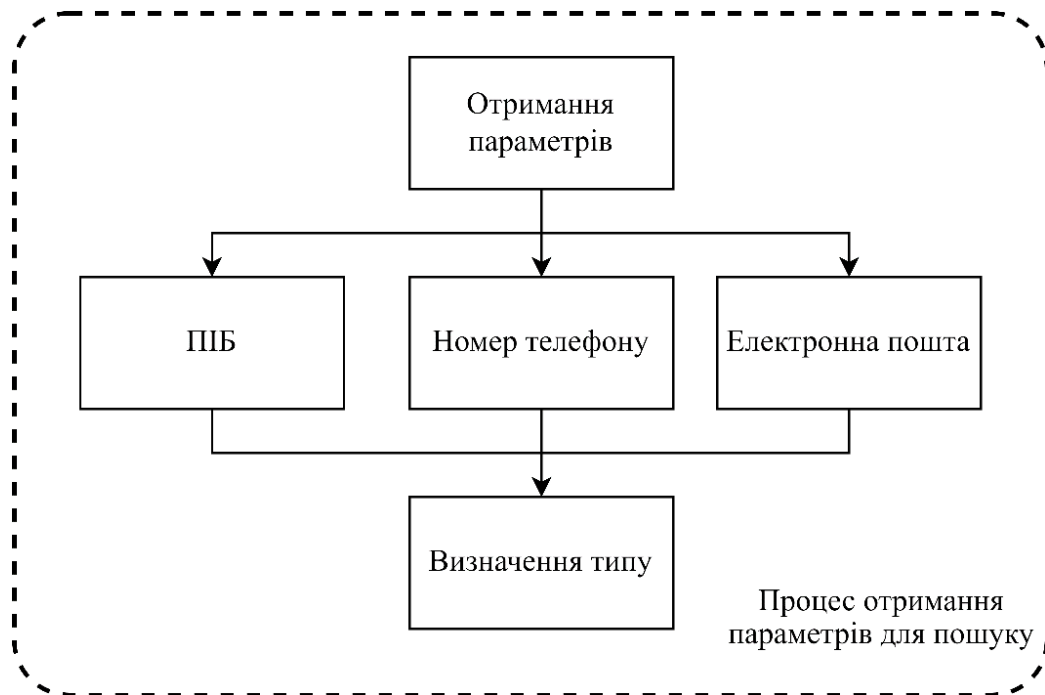


Рисунок 2.2 – Схема процесу отримання даних для пошуку

Модель процесу має наступний вигляд:

$$D = \{F|P|E\},$$

де  $D$  – отримані дані,  $F$  – ПІБ,  $P$  – номер телефону,  $E$  – електронна пошта.

Оскільки визначення типу введених даних відбувається за допомогою регулярних виразів, то пошук здійснюється лише за одним параметром. Після визначення типу вхідного параметра для пошуку інформації, відбувається формування запиту та процес пошуку на стороні сервера. Перед початком пошуку інформації здійснюється перевірка, чи даний користувач який здійснює запит є авторизованим. Для перевірки користувача чи є він авторизованим чи ні, було створено спеціальну таблицю з користувачами яким

надано доступ до системи. Так як розроблювана система це telegram-бот, користувач коли взаємодіє з клієнтською частиною, бот отримує від користувача його telegram-id. Далі цей id разом з вхідним параметром надсилається серверній частині. Якщо ж, даний id є в таблиці довірених користувачів, то виконується процес пошуку інформації, в іншому випадку користувачу буде відправлене повідомлення про помилку в авторизації.

Процес пошуку інформації складається з трьох етапів. Спочатку відбувається пошук інформації з відкритих джерел та витоків даних. Далі зі знайденої інформації проводиться її аналіз та структуризація. Потім ця інформація передається до процесу формування результату пошуку (рис. 2.3).

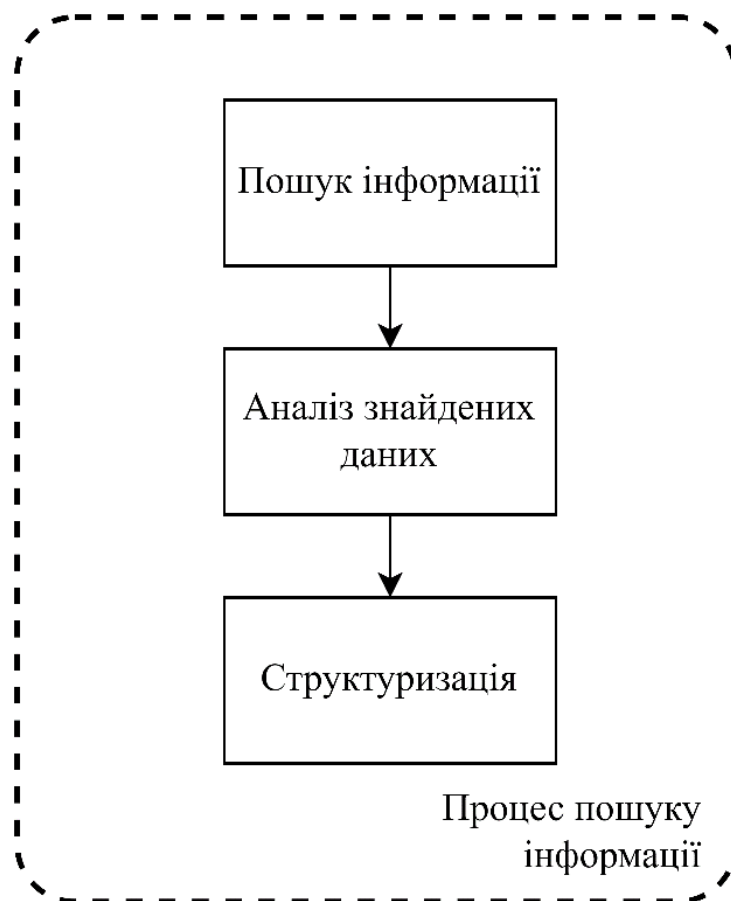


Рисунок 2.3 – Схема процесу пошуку інформації

Модель процесу має наступний вигляд:

$$ST_D = A_D(S_D),$$

де  $ST_D$  – структуровані дані,  $A_D$  – функція аналізу даних,  $S_D$  – знайдені дані.

Після виконання процесу пошуку інформації, виконується останній процес – це виведення результату пошуку. Якщо, за заданим параметром пошуку нічого не знайдемо, то формується відповідне повідомлення про не знайдену інформацію. Також слід зазначити, що незалежно від результату пошуку, кожен запит записується в базу даних, для подальшого аналізу адміністраторами додатку.

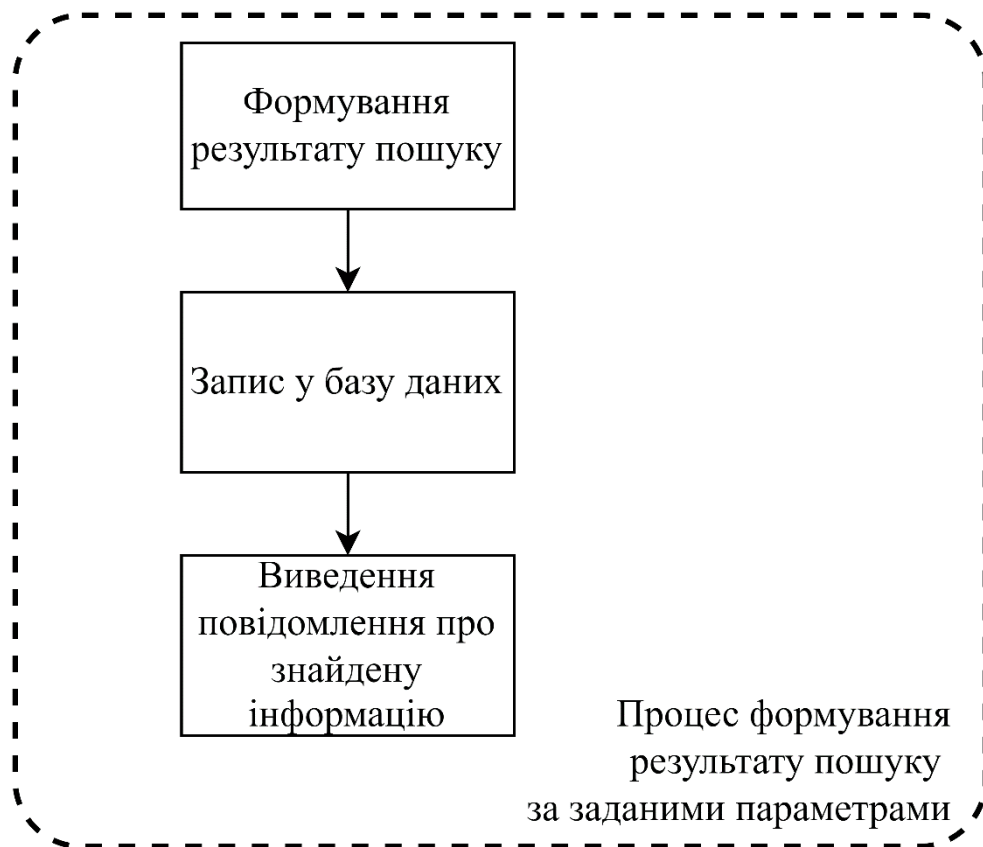


Рисунок 2.4 – Схема процесу формування результату пошуку за заданими параметрами

Модель процесу має наступний вигляд:

$$R_D = \langle ST_D \cup DB_D \rangle,$$

де  $R_D$  – результат,  $ST_D$  – структуровані дані,  $DB_D$  – запис у базу даних.

В загальному інформаційну технологію збору інформації про людину можна представити наступним чином:

$$IT = \{D, ST_D, R_D\}$$

Отже, результатом має бути інформаційна технологія, яка шукає інформацію про людину з відкритих джерел, аналізує і структурує її та формує результат пошуку за заданими даними.

## 2.2 Розробка архітектури системи

Архітектура системи, що реалізує інформаційну технологію для збору інформації про людину складається з кількох модулів, які з'єднуються між собою та використовують дані один від одного.

Архітектура системи зображена на рисунку 2.5. Архітектура інформаційної технології збору інформації про людину складається з наступних елементів:

- клієнтська частина;
- серверна частина;
- онлайн ресурси;
- база даних.

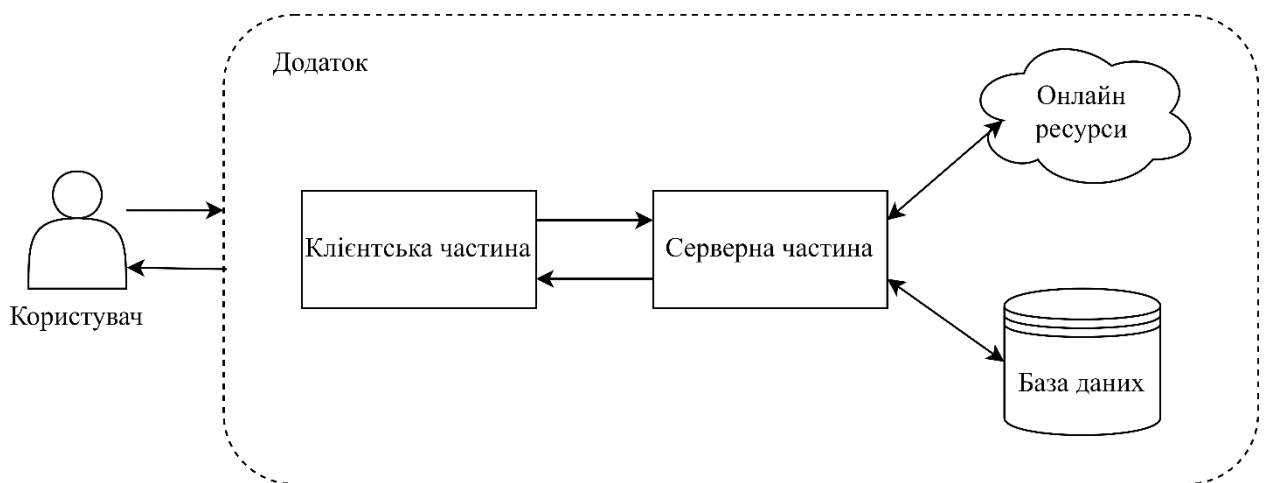


Рисунок 2.5 – Архітектура системи

З рисунку видно, що користувач безпосередньо взаємодіє з клієнтською частиною, яка в свою чергу обробляє вхідні параметри, формує запити та

надсилає далі до сервера. Для користування додатком, користувач має бути авторизованим, тобто його telegram-id має бути в таблиці користувачів. Сервер перед виконанням запиту, перевіряє в таблиці користувачів чи є даний користувач авторизованим, якщо користувача немає в таблиці, то формується повідомлення про помилку авторизації, в іншому випадку починається пошук інформації. Для пошуку інформації, сервер здійснює запити до локальної бази даних з витоками та до онлайн ресурсів. Далі зі знайдених даних відбувається їх структуризація. Після структуризації знайдених даних, серверна частина формує JSON відповідь, яка надсилається назад клієнту та формує звіт у зручному форматі. Всі запити, здійснені авторизованими чи не авторизованими користувачами, записуються в таблицю історії запитів, для подальшого їх аналізу адміністраторами системи. Розглянемо детально кожен складову системи окремо.

Принцип роботи клієнтської частини полягає в тому що спочатку отримує вхідні параметри, далі за допомогою регулярних виразів визначається тип введених даних (рис. 2.7).

Було використано наступні параметри за якими можна здійснювати пошук інформації:

- ПІБ та дата народження;
- номер телефону;
- електронна пошта.

Слід зазначити, пошук інформації здійснюється лише за одним параметром. Загальний алгоритм роботи клієнтської частини наведено на рисунку 2.6.





Рисунок 2.6 – Алгоритм роботи клієнтської частини

Коли користувач надсилає запити для пошуку інформації, клієнтська частина, тобто telegram-бот отримує об'єкт типу `Message` з якого отримується telegram-id користувача: `message.from_user.id`, та сам вміст повідомлення: `message.text`, які далі надсилаються серверу, де відбувається перевірка чи є даний id бази даних користувачів. Вже потім, з отриманих параметрів відбувається запит до сервера. Далі зі знайдених даних які надіслав у відповідь сервер, відбувається вивід повідомлення користувачеві.

З рисунку 2.7 видно, що після введення параметра для пошуку здійснюється визначення його типу. Як було згадано вище, визначення типу вхідного параметра здійснюється за допомогою регулярних виразів, якщо користувач вводить параметр, тип якого неможливо визначити то виводиться інструкція користування, в іншому випадку формується і надсилається запит до сервера.

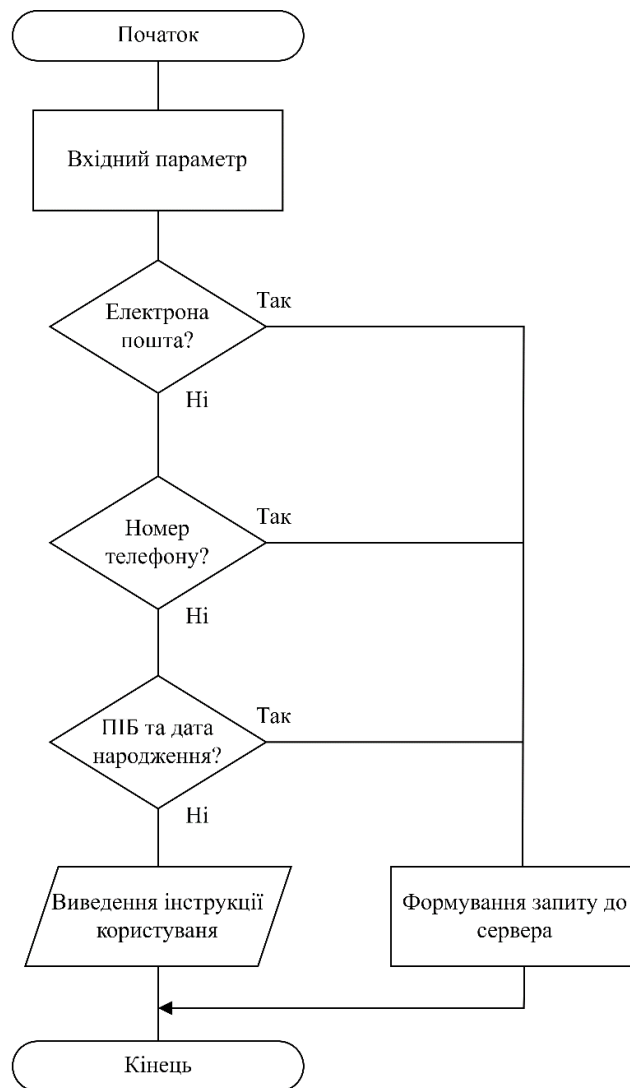


Рисунок 2.7 – Алгоритм виявлення типу вхідних параметрів

Далі отримується JSON відповідь від сервера за заданим запитом і формується результат пошуку. В залежності від введеного вхідного параметра виводиться відповідне повідомлення. Якщо користувач здійснює пошук за номером телефону чи електронною поштою, тоді виводиться звичайне повідомлення. Пошук за ПІБ та датою народження результатом формує HTML звіт, який в собі містить знайдену інформацію.

Алгоритм роботи серверної частини наведений на рис. 2.8. Серверна частина в свою чергу, отримує вхідні параметри для пошуку які передає клієнтська частина та перевіряє telegram-id користувача, чи є він у таблиці авторизованих користувачів. Сама таблиця містить в собі telegram-id користувачів, яким надано доступ до користуванням даним додатком. Якщо ж

користувача немає в даній таблиці, то формується повідомлення про помилку в авторизації. В іншому випадку відбувається процес пошуку інформації з сервісів і витоків даних та відправлення знайденої інформації до клієнтської частини рис. 2.9. Також слід зазначити, що незалежно чи є користувач у таблиці авторизованих користувачів чи ні, всі запити записуються в таблицю історії запитів. Таблиця з історією запитів містить в собі наступні поля:

- query\_id;
- query\_type;
- query;
- query\_from;
- status\_code;
- response;
- created\_at;
- updated\_at.

Поле query\_id містить ідентифікатор запиту який автоматично генерується і за яким можна переглянути сам запит. У query\_type записується сам тип запиту за яким здійснюється пошук. Поле query зберігає параметр пошуку який вводить користувач і за яким здійснюється пошук інформації про людину. Query\_from містить telegram-id користувача, який здійснює запит. У полі status\_code зберігає статус код HTTP запиту, який формується після процесу пошуку інформації. В поле response записується JSON відповідь, яку відправляє сервер після пошуку інформації за заданим типом пошуку. Поля created\_at та updated\_at містять в собі дату відповідно коли було створено та оновлено запит.

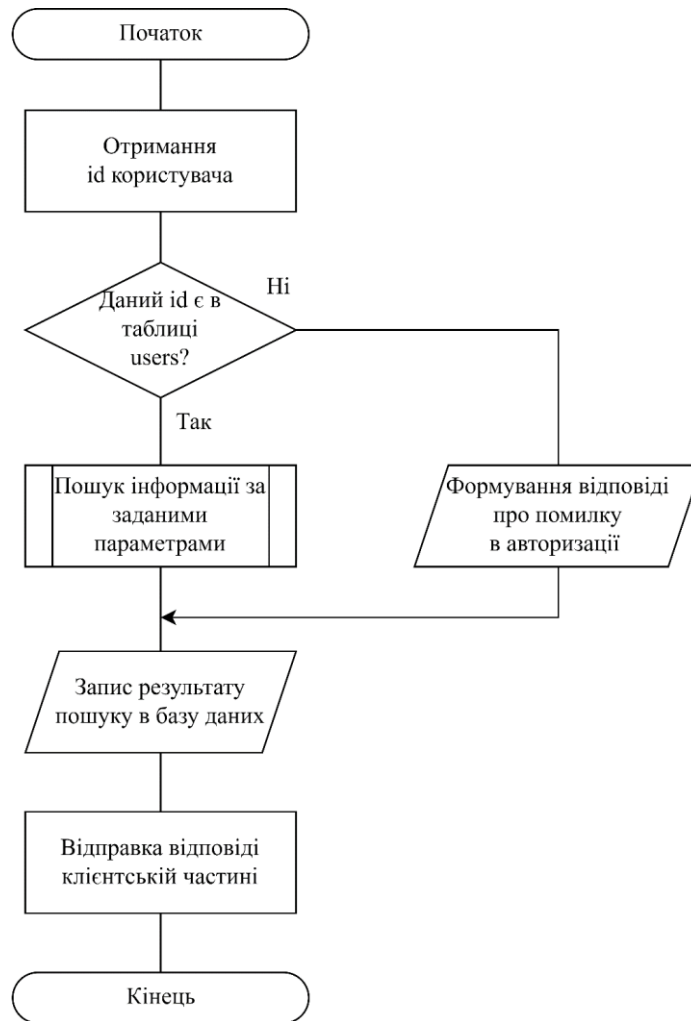


Рисунок 2.8 – Алгоритм роботи серверної частини

Процес пошуку інформації складається з трьох підпроцесів, а саме пошук інформації з інших сервісів, витоків даних та структуризації знайденої інформації. Загальний алгоритм роботи процесу наведено на рисунку 2.9.

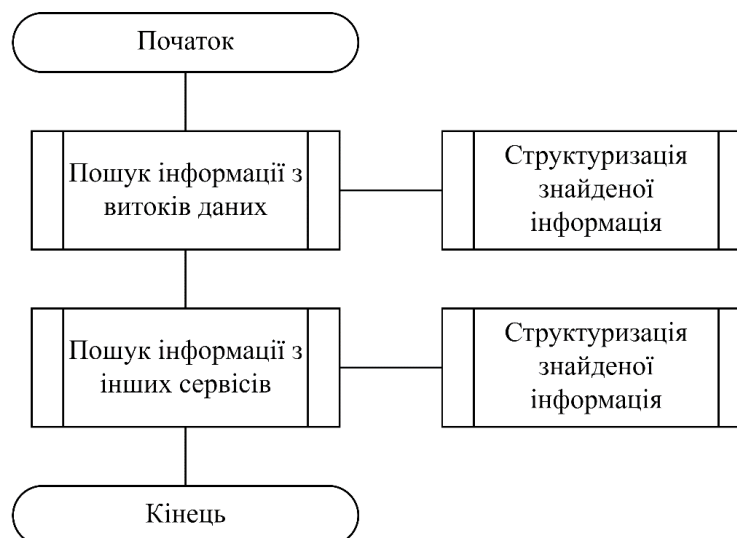


Рисунок 2.9 – Алгоритм роботи процесу пошуку інформації

Підпроцес пошуку інформації з витоків даних включає в себе наступні сервіси:

- сервіс пошуку інформації за номером телефону;
- сервіс пошуку інформації за електронною поштою;
- сервіс пошуку інформації за ПІБ та датою народження.

В таблиці 2.1 наведено витoki даних які використовуються для пошуку інформації.

Таблиця 2.1 – Список використаних витоків даних

| Назва таблиці    | Вміст таблиці  |
|------------------|--|
| alfa_bank        | ПІБ, номер телефону, email, адреса проживання  |
| avito_full       | Ім'я, номер телефону, місто проживання   |
| beeline_full     | ПІБ, номер телефону, адреса проживання   |
| cdek_full        | ПІБ, номер телефону, email   |
| delivery2_full   | Ім'я, номер телефону, email, адреса проживання                                       |
| delivery_full    | Ім'я, номер телефону, адреса проживання  |
| dobro_ru         | ПІБ, номер телефону, дата народження, email, регіон                                  |
| email_phones     | Номер телефону, email  |
| exluxair         | ПІБ, номер телефону, email, адреса проживання  |
| eyeofgod_users   | Номер телефону, telegram-id, нікнейм   |
| facebook_full    | Ім'я, номер телефону, facebook-id  |
| fl_users         | Ім'я, номер телефону, email  |
| fotostrana       | Ім'я, номер телефону, email, дата народження   |
| geek_brains      | Ім'я, номер телефону, email  |
| gemotest         | ПІБ, дата народження, номер телефону, email, адреса проживання, паспорт, СНІОР       |
| gibdd2_full      | ПІБ, дата народження, номер телефону, адреса проживання, паспорт, номерний знак, VIN |
| gosuslugi        | ПІБ, дата народження, номер телефону, email, адреса проживання, паспорт, СНІОР, ПІН  |
| hhru_users       | ПІБ, дата народження, номер телефону, email  |
| igzakon_users    | Ім'я, email  |
| kari             | Ім'я, номер телефону   |
| kari_summer2022  | ПІБ, дата народження, номер телефону, email, регіон                                  |
| lenta_shop_users | ПІБ, номер телефону  |

Продовження таблиці 2.1

|                  |  |
|------------------|--|
| linkedin_full    | Ім'я, номер телефону, email                          |
| mailru_full      | Ім'я, номер телефону, email                          |
| miltor_full      | Ім'я, номер телефону, email                          |
| moscow10k        | ПІБ, дата народження, email, адреса проживання       |
| mydocuments36    | ПІБ, email   |
| nadpo_ru_users   | ПІБ, email, номер телефону, регіон                   |
| okrug_full       | ПІБ, дата народження, номер телефону                 |
| payad_me         | ПІБ, номер телефону, email                           |
| pikabu_full      | Юзернейм, номер телефону, email                      |
| pirogidomoy      | Ім'я, номер телефону, адреса проживання              |
| r_sushi          | Ім'я, номер телефону                                 |
| rfcont_full      | ПІБ, номер телефону, email                           |
| roll46           | ПІБ, номер телефону, адреса проживання               |
| ruskoe_slovo     | ПІБ, номер телефону, email, адреса проживання        |
| sushi_full       | Ім'я, номер телефону, адреса проживання              |
| tele2            | Ім'я, номер телефону, email                          |
| telegram         | Telegram-id, номер телефону, нікнейм                 |
| tochkaedy        | ПІБ, номер телефону, адреса проживання               |
| turniketov_net   | ПІБ, дата народження, номер телефону, email, паспорт |
| tutubus_orders   | Ім'я, номер телефону, email                          |
| vivadengi_ru     | ПІБ, дата народження, номер телефону, email, регіон  |
| wildberries_full | ПІБ, номер телефону, email, адреса проживання        |
| yandex_full      | Ім'я, номер телефону, адреса проживання              |
| yandex_pract     | Ім'я, номер телефону, email                          |

Загальна кількість записів з таблиць витоками даних складає близько 294 мільйонів. В загальному заліку становить 63 гігабайт. Дані які наведено вище були зібрані з загальнодоступних спільнот різних ресурсів.

Підпроцес пошуку інформації з інших сервісів містить в собі API соціальних мереж, сервісів та пошукових систем.

Нижче наведено API які використовуються для пошуку інформації:

- vk\_api;
- ok\_api;
- facebook\_api;

- intelx.io\_api;
- webresolver.nl\_api.

Підпроцес структуризації знайденої інформації в свою чергу формує об'єкт зі структурованими даними який потім відправляється у вигляді JSON відповіді клієнтській частині, в залежності від типу запиту, яка потім відображається користувачу у зручному форматі. В таблиці 2.2 наведено приклади JSON відповідей за типом пошуку.

Таблиця 2.2 – Представлення JSON відповіді в залежності від запиту

| Тип запиту                       | JSON відповідь  |
|----------------------------------|---|
| Пошук за номером телефону        | {<br>"names": [...],<br>"bday": [...],<br>"addresses": [...],<br>"emails": [...],<br>"social": {...},<br>"transport": [...],<br>}               |
| Пошук за електронною поштою      | {<br>"names": [...],<br>"bday": [...],<br>"phone_numbers": [...],<br>"addresses": [...],<br>"social": {...},<br>"leaked_passwords": [...],<br>} |
| Пошук за ПІБ та датою народження | {<br>"emails": [...],<br>"phone_numbers": [...],<br>"addresses": [...],<br>"documents": {...},<br>"social": {...},<br>"transport": [...],<br>}  |

Якщо жодної інформації не знайдено, то сервер відправить відповідь про невдалий пошук клієнтській частині.

Отже, у даному розділі виконано розробку структури інформаційної технології. Проаналізовано кожен процес, який призначений для певних задач. В кожен процес подаються певні дані на вхід, які обробляються та надсилаються далі. Розроблено архітектуру системи та розроблено узагальнений алгоритм роботи засобу. Розглянуто функції клієнтської та серверної частин окремо та представлено алгоритми їх роботи.

## **3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ**

### **3.1 Обґрунтування вибору програмних засобів для реалізації інформаційної технології**

На даному етапі необхідно виконати обґрунтування програмних засобів, які будуть використані для реалізації інформаційної технології і поставленої мети у магістерській кваліфікаційній роботі мети.

Найкращими мовами для створення серверної частини додатків є PHP, Python та JavaScript.

PHP – це мова програмування з відкритим вихідним кодом, створена ще 1994 року [51]. Незважаючи на те, що це одна з найстаріших мов, вона часто покращується та оновлюється, і зараз це одна з найгнучкіших і найпростіших у вивченні мов. Вона широко використовується фреймворками та системами CMS, такими як Laravel, Symfony та WordPress, і може бути інтегрована з багатьма іншими мовами, такими як JavaScript, HTML, XML та базами даних: MySQL, Oracle, SQLite та іншими.

У 2022 році мову Python використовують 48% розробників [52]. Ця мова досить проста у вивченні. Одним з найкращих фреймворків для створення backend-частини це – Django, який є безпечним, стабільним, швидким і масштабованим [53]. Тим не менш, є ще багато фреймворків на вибір, що роблять розробку на Python доступною. Python також сумісний з механізмами штучного інтелекту і машинного навчання та надає безліч інструментів, розроблених спеціально для цих технологій.

JavaScript широко використовується для розробки frontend-частини, але в останні роки його почали застосовувати і для розробки backend-частини. JavaScript пропонує безліч фреймворків, таких як Jext.JS або Node.JS, які роблять процес розробки швидким і забезпечують масштабованість додатку та її інтеграцію з такими базами даних, як MySQL та MongoDB.



Для розробки серверної частини було обрано мову програмування PHP та фреймворк Laravel. Додатки написані на Laravel забезпечують високу продуктивність порівняно з додатками, створеними іншими фреймворками. Це можливо завдяки системі кешування. Драйвер файлового кешування зберігає багато елементів у файловій системі [54]. Це дозволяє швидко розробляти програми. Laravel забезпечує просту та ефективну систему аутентифікації. Вона забезпечує контроль доступу до ресурсів. Це дозволяє обмежити доступ для неавторизованих користувачів до тих чи інших ресурсів. Laravel забезпечує безпеку додатків. Кодова база фреймворку захищена від загроз, наприклад, SQL-ін'єкцій або міжсайтових підробок запитів (CSRF). Це захищає користувачів від втрати важливих даних.

Laravel дозволяє швидко створювати додатки, тому що в процесі роботи розробники можуть обходитися без складного коду. Даний фреймворк побудований на архітектурі MVC [54], а це забезпечує доступ до всієї необхідної для створення додатку інфраструктури та економить час.

Для системи керування базами даних було обрано MySQL. MySQL – це база даних з відкритим вихідним кодом, яка дозволяє управляти базами даних шляхом їх підключення до програмного забезпечення. Це стабільне, надійне та потужне рішення з розширеними можливостями [55]. MySQL працює на різних платформах, таких як Linux, Solaris, Windows та інші. Дана система керування базами даних є хорошим вибором для тих проектів, які орієнтовані на кілька платформ, особливо для веб-застосунків. Фактично MySQL є частиною відомого серверного стека LAMP (Linux Apache MySQL PHP), який використовується в усьому світі для розробки веб-додатків. MySQL підтримує безліч платформ з різними мовами, такими як C, C++, PHP, PERL, JAVA, Python та інші.

Для розробки клієнтської частини, було використано мову програмування Python. Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою [38]. Вбудовані структури даних високого рівня у поєднанні з динамічною типізацією та

зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, використання як мови сценаріїв або зв'язування існуючих компонентів. Дана мова програмування має простий і легкий синтаксис, що впливає на зручність читання коду, та значно знижує витрати часу на обслуговування програми.

Мова Python має багато фреймворків для розробки Telegram-ботів, в якості фреймворка було обрано Aiogram. Aiogram – це досить простий та повністю асинхронний фреймворк для Telegram Bot API, написаний на Python 3.7 з використанням `asyncio` та `aiohttp` [56]. Для реалізації інформаційної технології і досягнення поставленої у магістерській кваліфікаційній роботі мети, потрібно визначити, які саме інструментальні засоби будуть використані під час розробки програмного засобу, що реалізує інформаційну технологію.

Для розробки серверної частини було використано Visual Studio Code. Visual Studio Code – це інструмент із відкритим кодом, завдяки якому можна створювати сучасні веб-додатки. Visual Studio Code є безкоштовним і доступний на платформах – Linux, Mac OSX та Windows [57].

Слід зазначити ряд переваг:

- Visual Studio Code не потребує багато місця;
- підтримує настільні програми та має відмінний інструмент для розробки веб-додатків;
- допомагає заощадити багато часу, оскільки присутнє автодоповнення коду;
- підтримує синтаксис багатьох мов програмування;
- має безліч плагінів, що дозволяє розширити функціональність.

Клієнтська частина розроблялась в середовищі PyCharm. PyCharm – це інтегроване середовище розробки, яке містить в собі повний набір засобів для розробки на мові Python [58]. PyCharm сумісний з платформами Linux, macOS і Windows. PyCharm підтримує версії Python 2 та Python 3 і вище. Переваги даної IDE:

- функціональний редактор коду з під світлою синтаксису та форматуванням коду;

- зручна навігація в коді;
- наявність Bash терміналів;
- швидкий перегляд документації будь-якого елемента безпосередньо в самому редакторі, автодоповнення і багато іншого.

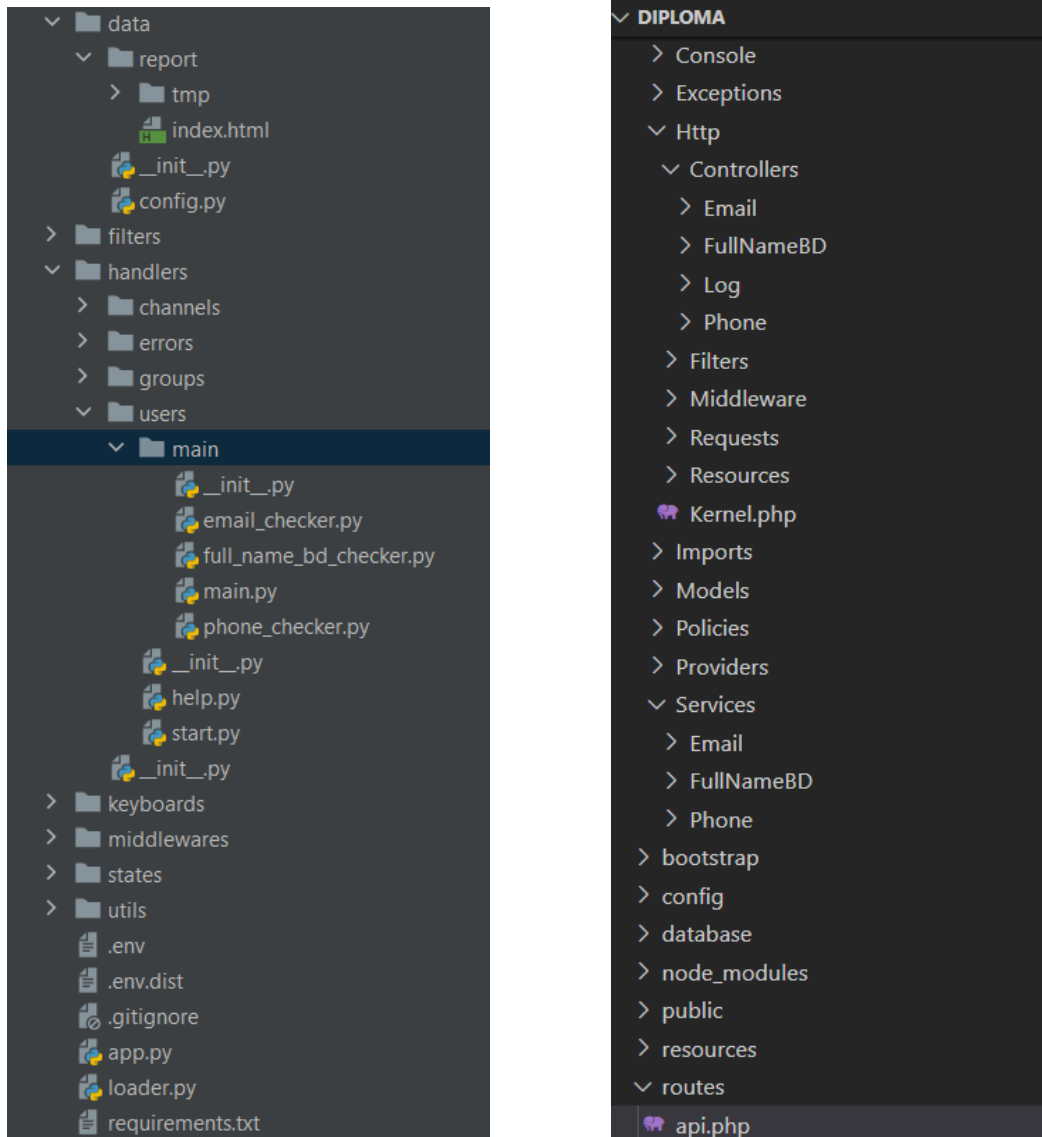
Визначивши основний перелік інструментів розробки, можна перейти до розробки структури програмного засобу.

## 3.2 Розробка програмного засобу

Інформаційна технологія складається з клієнтської та серверної частини, які призначені для різних задач, структури проекту наведено на рисунку 3.1. В папці users клієнтської частини містяться файли для обробки команд та даних введених користувачем. Файл start.py обробляє команду /start та виводить привітання користувачу, котрий вперше користується додатком. У файлі help.py обробляється команда /help та виводиться інструкція користування додатком. Файл main.py отримує вхідні дані для пошуку та визначає тип даних за допомогою регулярних виразів. Частина коду наведено нижче:

```
phone = re.compile(r"^[+]?([0-9]{3})?[-\s.]?[0-9]{3}[-\s.]?[0-9]{4,6}$")
full_name_with_bd = re.compile(r"([^\s]+) ([^\s]+) ([^\s]+\d{2}.\d{2}.\d{4}")
email = re.compile(r"([-!#-'*+/-9=?A-Z^-~]+(\.[-!#-'*+/-9=?A-Z^-~]+)*|\\"([!#-^~ \t]|(\\"[\\t -~]))+\\")@([-!#-'*+/-9=?A-Z^-~]+(\.[-!#-'*+/-9=?A-Z^-~]+)*|\\"[\\t -Z^-~]*\\")")
```

Після визначення типу, дані передаються у файл email\_checker.py, full\_name\_bd\_checker.py чи phone\_checker.py для здійснення запиту пошуку інформації до серверної частини. В директорії report міститься файл index.html, який генерує звіт коли користувач здійснює пошук за ПІБ та датою народження.



a)

б)

Рисунок 3.1 – Структура проекту(а – клієнтська частина, б – серверна частина)

В файлі `api.php` серверної частини містяться роути, за якими клієнтська частина здійснює пошук інформації. В папці `Middleware` містяться файли які забезпечують зручний механізм для перевірки та фільтрації HTTP-запитів, що надходять до серверної частини. Нижче наведено частину коду `middleware` який перевіряє, чи `telegram-id` користувача котрий виконує запит є авторизований:

```
$tg_id = $request->request->parameters['tg_id'];
$user = BotUsers::where('tg_id', $tg_id);
if (!$user->first()) {
    return response()->json(['error' => 'Not authorized.'], 403);
}
```

```

    }
    return $next($request);

```

Директорія **Controllers** включає в себе контролери, які обробляють запити з клієнтської частини рис. 3.2. Для зручності в читанні коду, значну частину коду контролерів було винесено в папку **Services**, в якій безпосередньо відбувається пошук інформації з витоків даних та сервісів рис. 3.3.

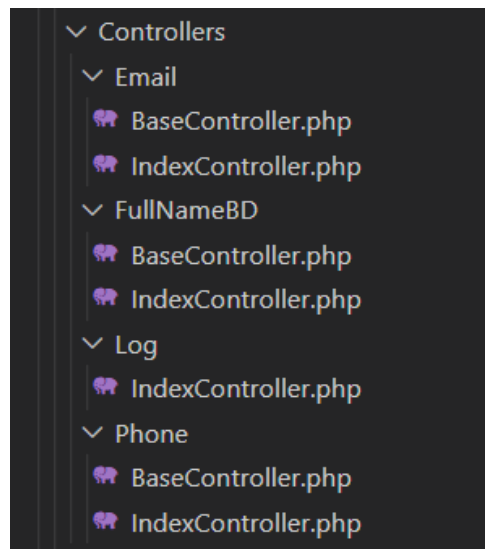


Рисунок 3.2 – Список контролерів

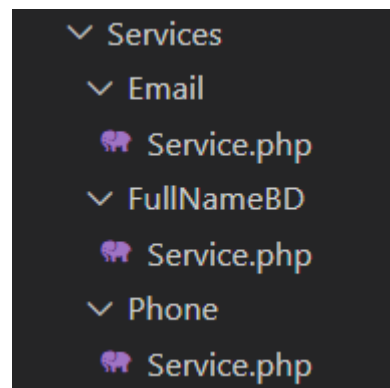


Рисунок 3.3 – Список сервісів

Нижче наведено частину коду одного із сервісів:

```

foreach ($list_of_used_models as $model) {
    $info = $model::where('email', $email)->get()-
>toArray();

    $this->structData($info);
}

foreach ($list_of_used_services as $service) {
    $client = Http::timeout(60);
    $url = $service . $email;

```

```

        $info = $client->get($url);
        $this->structData($info);
    }

```

Далі формується JSON відповідь з знайдених даних і відправляється клієнтській частині. Якщо не знайдено жодної інформації, то формується відповідна відповідь. Нижче наведено код формування відповіді:

```

if ($is_data_null) {
    return [
        'not_found' => null
    ];
} else
    return [
        'names' => $this->names,
        'addresses' => $this->addresses,
        'birthday' => $this->birthday,
        'phone_number' => $this->phone_numbers,
        'associated_ip' => $this->associated_ip,
        'social' => $this->social,
        'leaked_passwords' => $this->leaked_passwords
    ];

```

Для запуску клієнтської частини необхідно запустити файл `app.py`, в якому використовуються всі вище описані методи. Серверна частина запускається командою: `php artisan serve`.

Також для зручності перегляду історії запитів і щоб мати можливість додавати нових користувачів додатку, було створено панель для адміністрування, структуру наведено на рисунку 3.4. Файл `web.php` містить в собі роути за якими здійснюються запити до веб-сторінок. Директорія `Controllers` містить контролери які обробляють запити. Папка `resources` включає в себе різні файли для відображення та функціонування клієнтської частини панелі адміністрування.

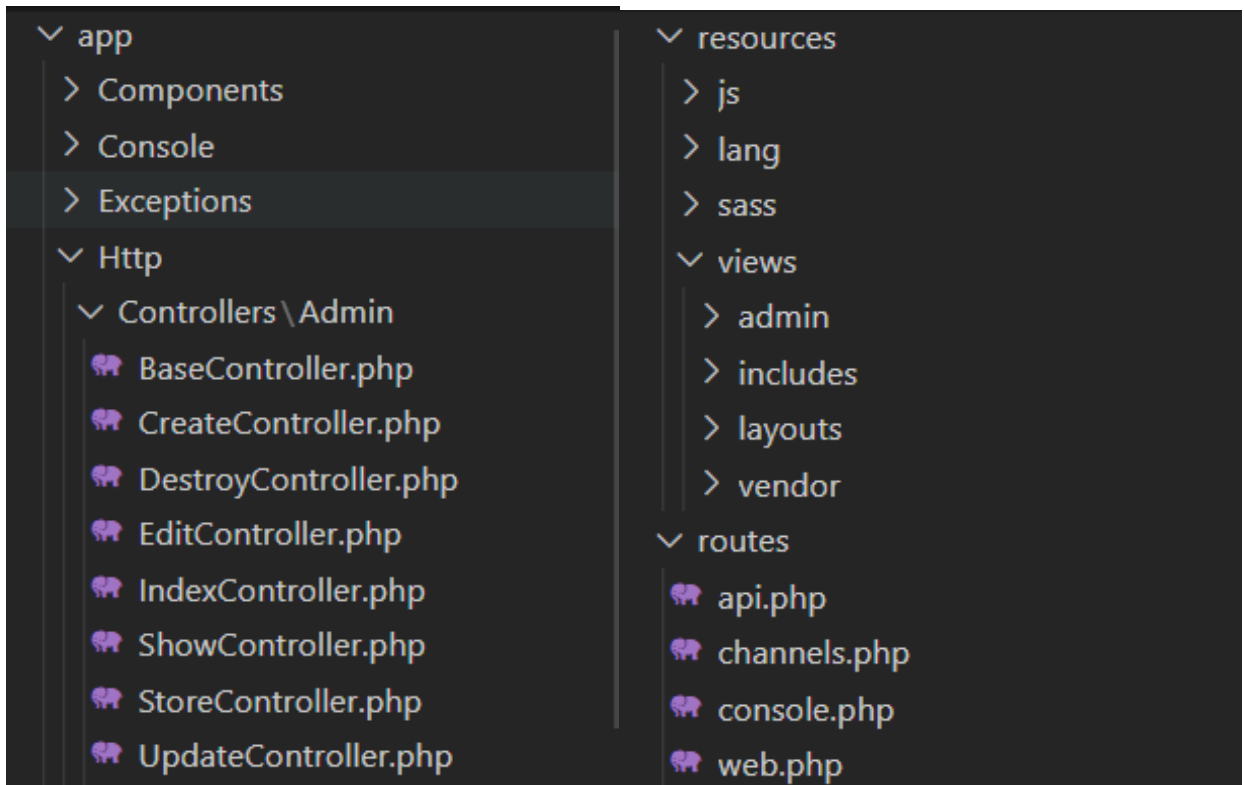


Рисунок 3.4 – Структура панелі для адміністрування

Код клієнтської, серверної частини та панелі для адміністрування наведено у додатку Б. Для перевірки реалізованих алгоритмів додатку необхідно провести тестування.

### 3.3 Тестування програмного засобу

Запускаємо клієнтську і серверну частину та переходимо до облікового запису, який не є авторизованим в додатку. Виконаємо тестування пошуку інформації за номером телефону і бачимо повідомлення про помилку рис. 3.4.



Рисунок 3.4 – Результат запити за номером телефону від неавторизованого користувача

|   | query_id           | query_type | query       | query_from | st... | response      | created_at          |
|---|--------------------|------------|-------------|------------|-------|---------------|---------------------|
| 1 | 21 b8cef4e7-622... | phone      | 79268454405 | 5798789003 |       | 403 Forbidden | 2022-12-21 19:51:36 |

Рисунок 3.5 – Запис даних запиту в таблиці query\_history

Як видно з рисунку 3.5. даний запит записано в базу даних. Далі спробуємо виконати пошук за електронною поштою та ПІБ з датою народження.

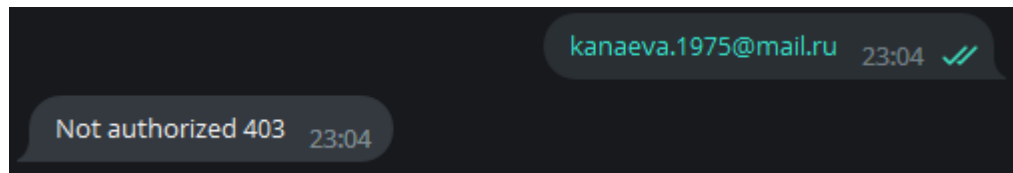


Рисунок 3.6 – Результат запиту за електронною поштою від неавторизованого користувача

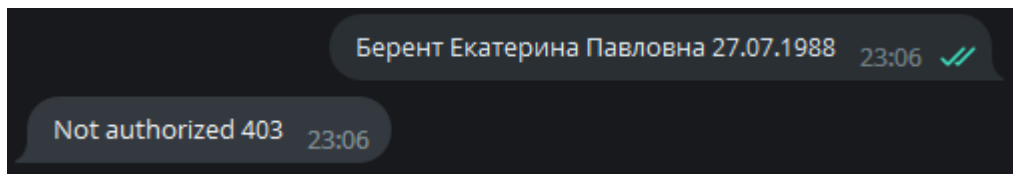


Рисунок 3.7 – Результат запиту за ПІБ та датою народження від неавторизованого користувача

|   | query_id           | query_type     | query           | query_from | st... | response      | created_at          |
|---|--------------------|----------------|-----------------|------------|-------|---------------|---------------------|
| 1 | 23 675dcf82-a08... | full_name_date | Берент Екате... | 5798789003 |       | 403 Forbidden | 2022-12-21 20:06:37 |
| 2 | 22 3cd5636d-aff... | email          | kanaeva.1975... | 5798789003 |       | 403 Forbidden | 2022-12-21 20:04:06 |

Рисунок 3.8 – Записи даних запитів в таблиці query\_history

Як видно рисунків 3.6 та 3.7 додаток видає повідомлення про помилку і записує дані запити в таблицю query\_history незалежно чи є користувач авторизований, чи ні (рис. 3.8). Далі виконуємо запити за номером телефону з облікового запису який є авторизованим в додатку і бачимо знайдену інформацію (рис. 3.9, рис. 3.10, рис. 3.11).



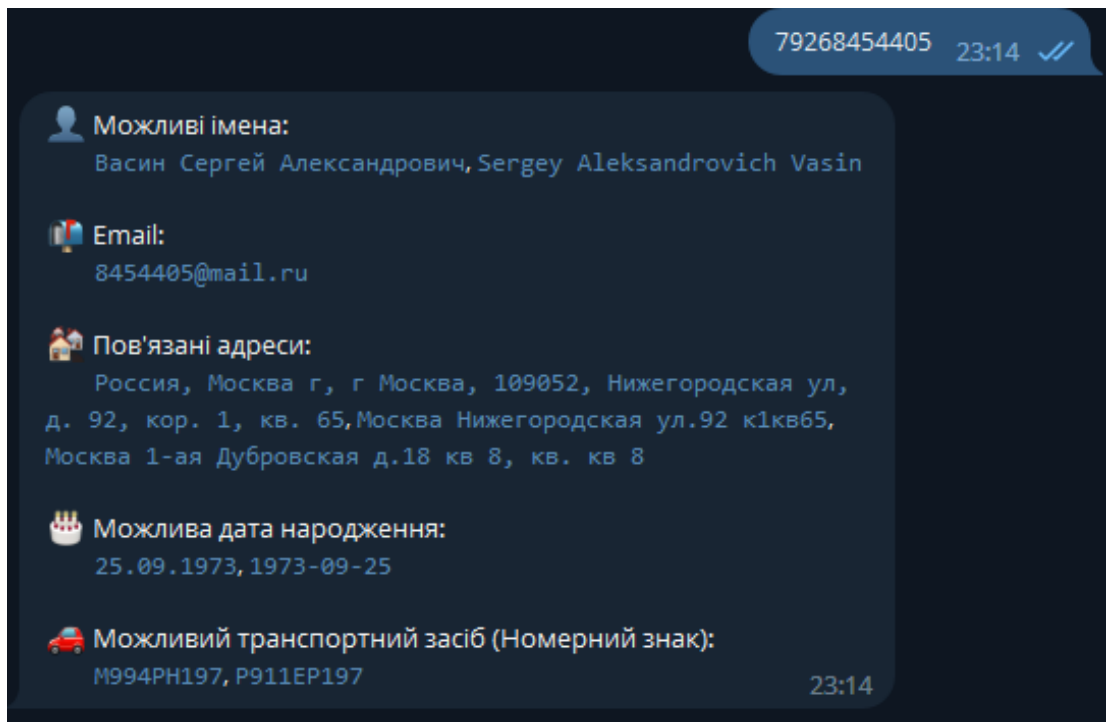


Рисунок 3.9 – Результат запиту за номером телефону від авторизованого користувача

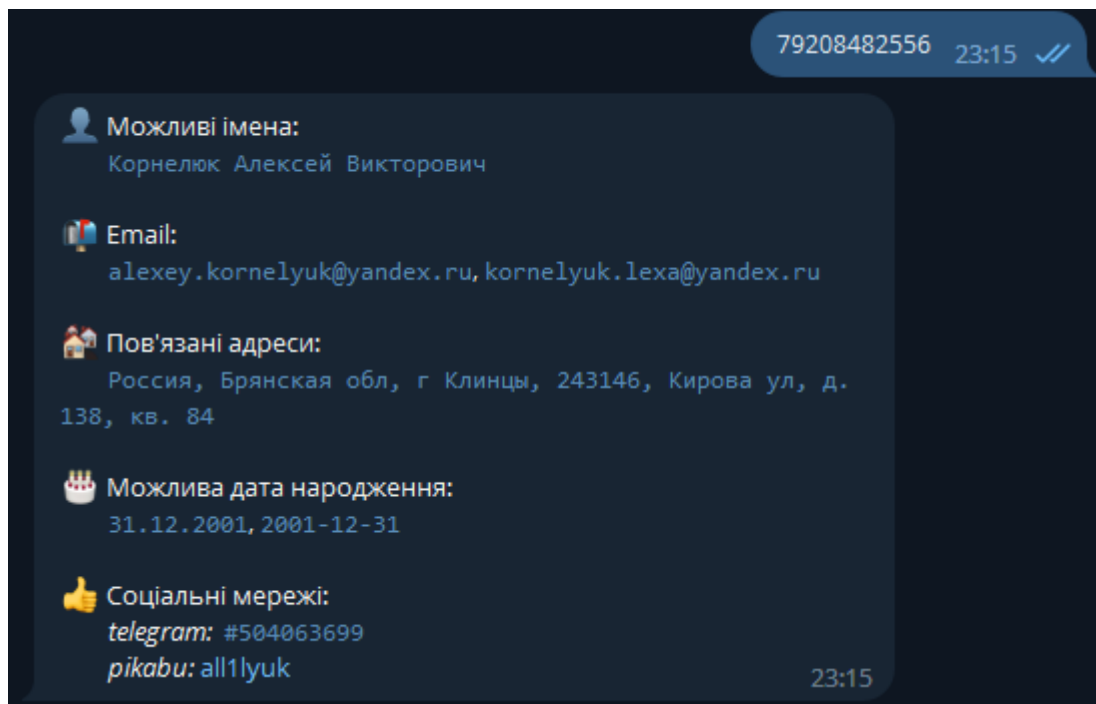


Рисунок 3.10 – Результат запиту за номером телефону від авторизованого користувача

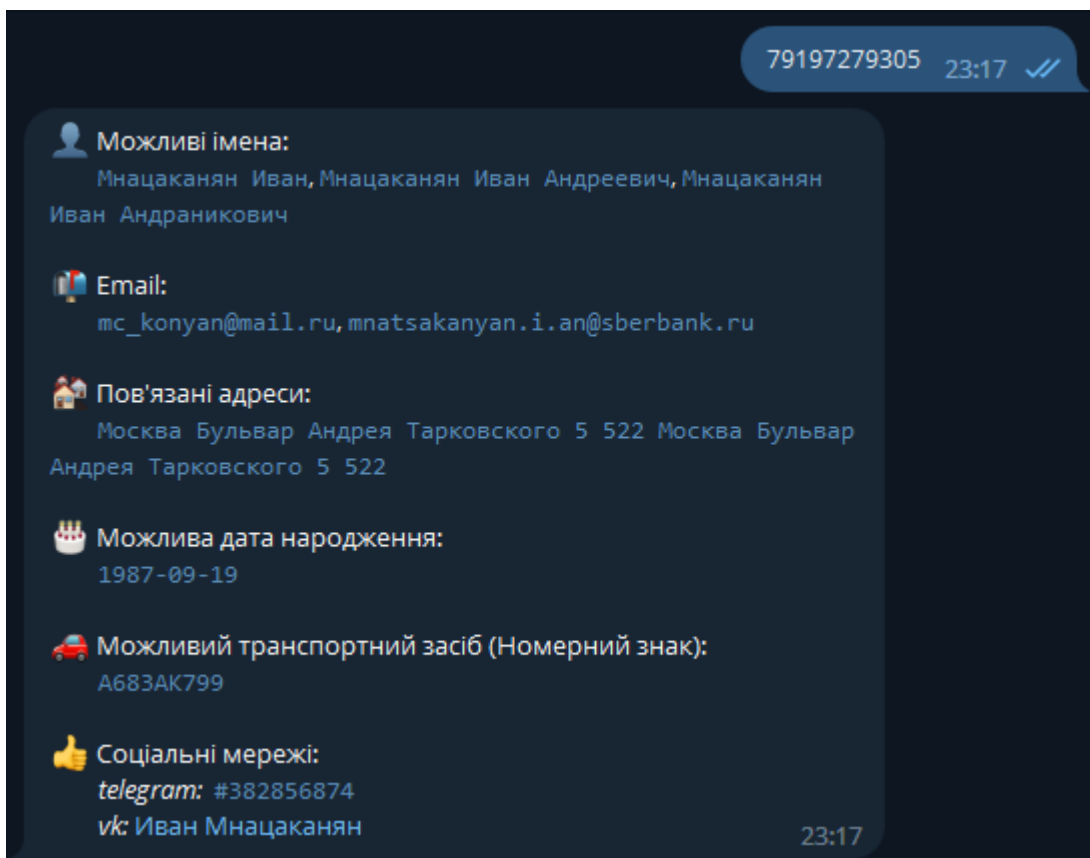


Рисунок 3.11 – Результат запиту за номером телефону від авторизованого користувача

Якщо система не знайде жодної інформації, то виведе відповідне повідомлення рис. 3.12. Також з рисунку 3.13 видно, що всі вище показані запити успішно записані в таблицю `query_history`.

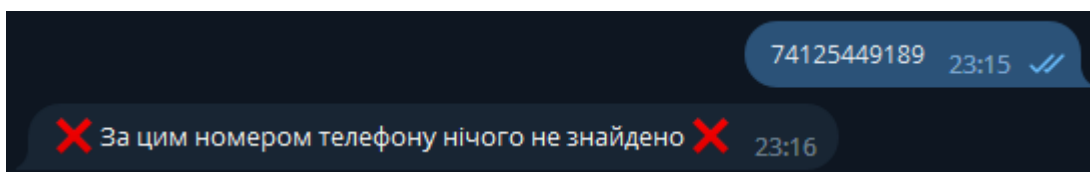


Рисунок 3.12 – Результат запиту за номером телефону від авторизованого користувача

|   | query_id | query_type      | query | query_from  | st...      | response               | created_at          |
|---|----------|-----------------|-------|-------------|------------|------------------------|---------------------|
| 1 | 27       | 40de4a1f-20f... | phone | 79197279305 | 5968137855 | 200 {'names': ['М...   | 2022-12-21 20:17:57 |
| 2 | 26       | 71f5a811-38b... | phone | 74125449189 | 5968137855 | 200 {'not_found': :... | 2022-12-21 20:15:58 |
| 3 | 25       | 138566c7-50a... | phone | 79208482556 | 5968137855 | 200 {'names': ['К...   | 2022-12-21 20:15:17 |
| 4 | 24       | 9eb67b78-460... | phone | 79268454405 | 5968137855 | 200 {'names': ['В...   | 2022-12-21 20:14:15 |

Рисунок 3.13 – Запис даних запитів в таблиці `query_history`

Далі виконаємо пошук за електронною поштою (рис. 3.14, рис. 3.15) та бачимо знайдену інформацію. Якщо ж, нічого не знайдено то система виводить відповідне повідомлення рис. 3.16.

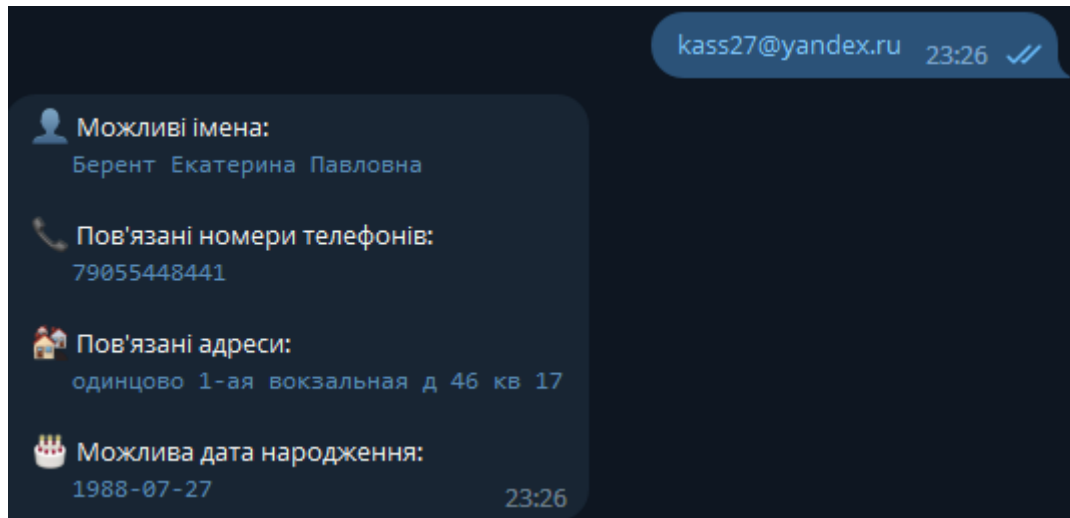


Рисунок 3.14 – Результат запиту за електронною поштою від авторизованого користувача

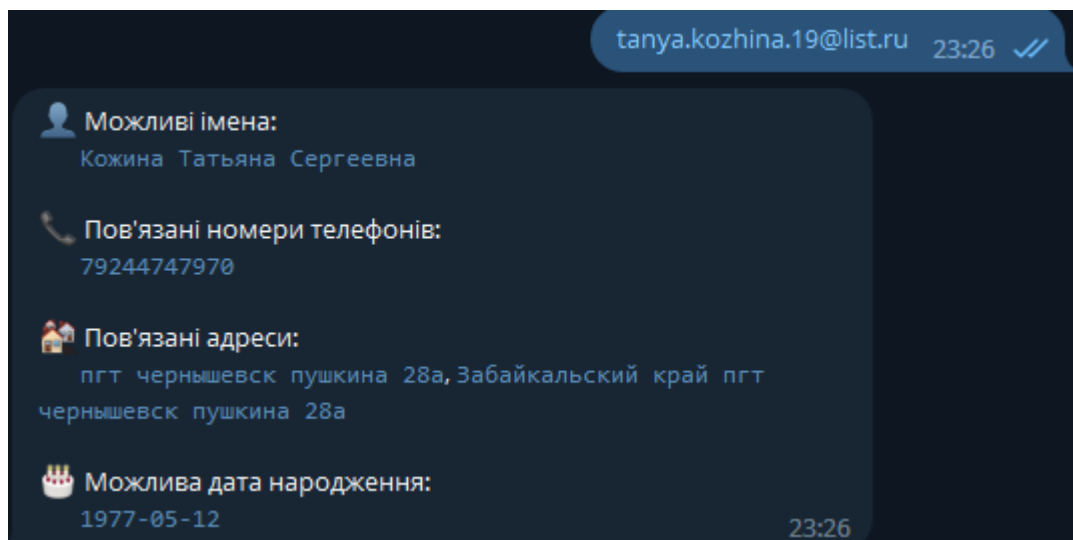


Рисунок 3.15 – Результат запиту за електронною поштою від авторизованого користувача

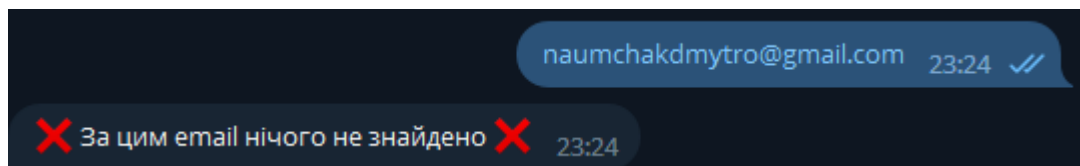


Рисунок 3.16 – Результат запиту за електронною поштою від авторизованого користувача

|    | query_id | query_type      | query | query_from      | st...      | response              | created_at          |
|----|----------|-----------------|-------|-----------------|------------|-----------------------|---------------------|
| 14 | 33       | 495ecfd9-dd5... | email | tanya.kozhin... | 5968137855 | 200 {'names': ['K...  | 2022-12-21 20:26:40 |
| 15 | 32       | 8a235c2e-3e7... | email | kass27@yande... | 5968137855 | 200 {'names': ['Б...  | 2022-12-21 20:26:16 |
| 16 | 31       | 3ddf2ec9-32c... | email | naumchakdmyt... | 5968137855 | 200 {'not_found': ... | 2022-12-21 20:24:02 |

Рисунок 3.17 – Записи даних запитів в таблиці query\_history

З рисунку 3.17 видно, що всі вище показані запити за електронною поштою успішно збережено в таблицю query\_history.

Виконаємо тестування пошуку за ПІБ та датою народження (рис. 3.18, рис. 3.19), результатом формує HTML звіт, який в собі містить знайдену інформацію (рис. 3.21, рис. 3.22). Якщо за даним типом пошуку нічого не знайдено, то виведеться відповідне повідомлення рис. 3.20.

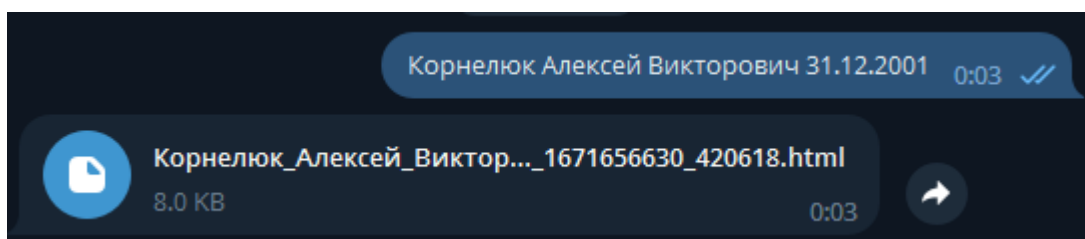


Рисунок 3.18 – Результат запити за ПІБ та датою народження від авторизованого користувача

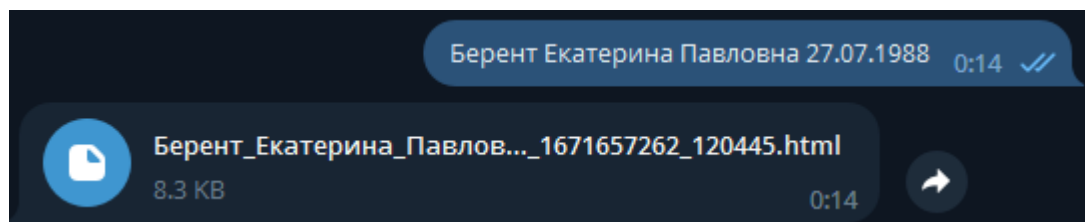


Рисунок 3.19 – Результат запити за ПІБ та датою народження від авторизованого користувача

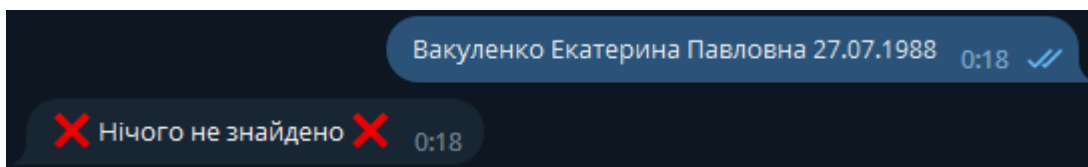


Рисунок 3.20 – Результат запити за ПІБ та датою народження від авторизованого користувача

**КОРНЕЛЮК  
АЛЕКСЕЙ  
ВИКТОРОВИЧ**  
31.12.2001

---

**Пов'язані email адреси** alexey.kornelyuk@yandex.ru  
kornelyuk.lexa@yandex.ru

---

**Пов'язані адреси** Россия, Брянская обл, г Клинцы, 243146, Кирова ул, д. 138, кв. 84

---

**Документи** Паспорт гражданина РФ:  
-1515 238750, выдан 27.01.2016 Межрайонным отделом УФМС России по Брянской обл. в г. Клинцы 320017  
СНИЛС:  
-13670764171  
ИНН:  
-324109181890

---

**Пов'язані номери телефонів** 79208482556

---

**Пов'язані транспортні засоби** ✘ Не найдено ✘

---

**Соціальні мережі** telegram: #504063699  
pikabu: [allilyuk](#)  
vk: [Алексей Корнелюк](#)

Рисунок 3.21 – Результат пошуку за ПІБ та датою народження

З рисунку 3.21 видно, що запит за ПІБ та датою народження сформував HTML звіт. Для зручності знайдену інформацію структуровано по категоріям. Якщо певної інформації за даною категорією не знайдено, то в звіті буде відповідне повідомлення.

Отже, виконаємо перевірку знайдених даних. Так як, в звіті було знайдено telegram аккаунт, то спробуємо додати даний номер телефону до списку контактів і на рисунку 3.22 видно, що дійсно такий обліковий запис існує та їх telegram-id однакові. Далі перейдемо до сторінки vk, як видно зі сторінки профілю, день народження такий самий, як і в сформованому звіті.

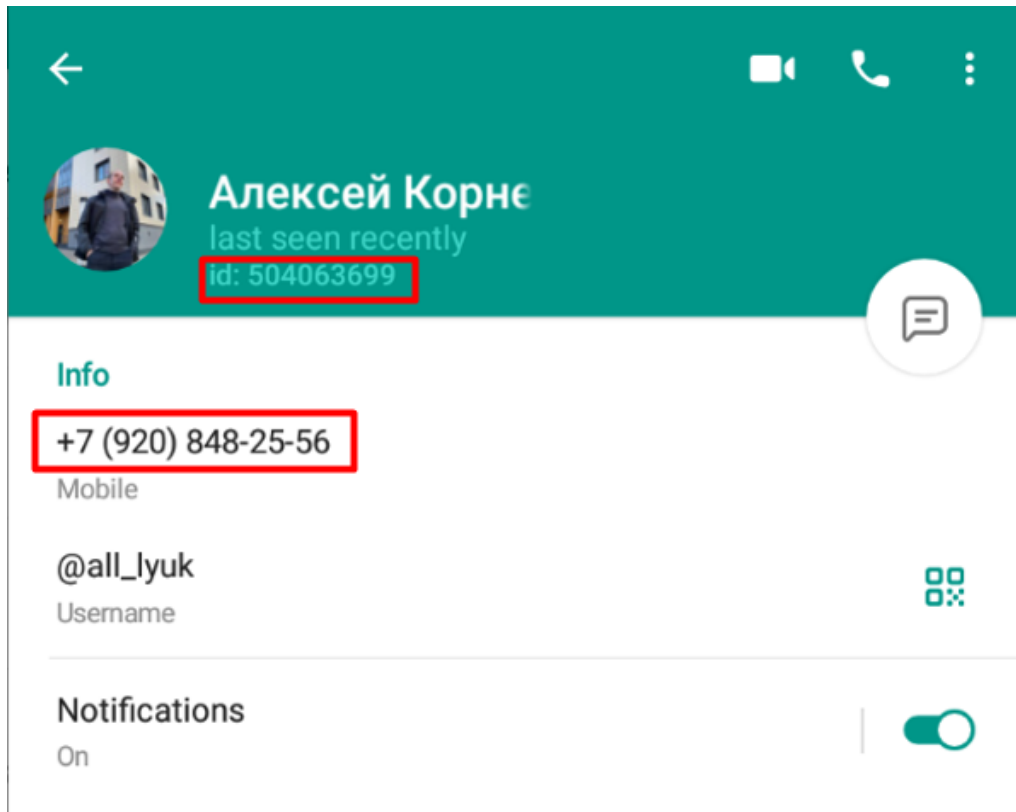


Рисунок 3.22 – Результаты проверки найденных данных в telegram

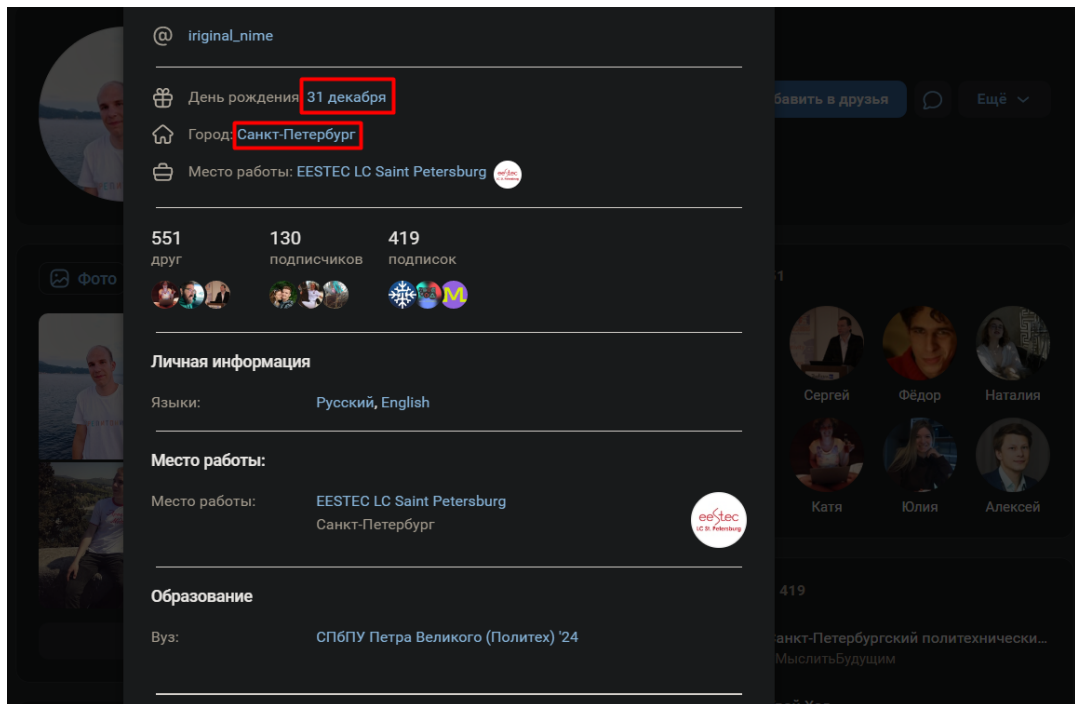


Рисунок 3.23 – Результаты проверки найденных данных в социальной сети vk

Далі виконавши пошук друзів за прізвищем, можна побачити список знайдених можливих родичів рис. 3.24. Перейшовши на перший профіль можна впевнитись, що це батько досліджуваного суб'єкта. З даної сторінки

можна припустити, що досліджуваний суб'єкт скоріш за все народився в місті, яке вказане на сторінці батька, і на даний момент мешкає іншому місті (рис. 3.23, рис. 3.25). Також проаналізувавши профіль батька, можна підтвердити, що знайдені друзі на рисунку 3.24 – це близькі родичі суб'єкта.

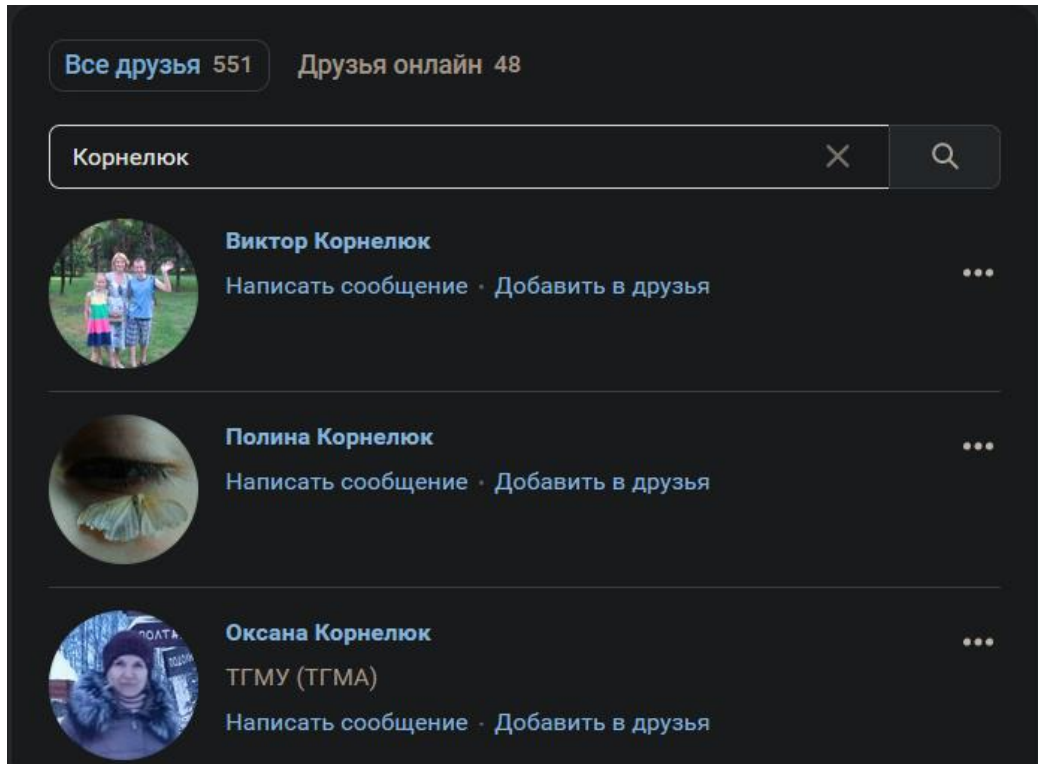


Рисунок 3.24 – Результати перевірки знайдених даних в соціальній мережі vk

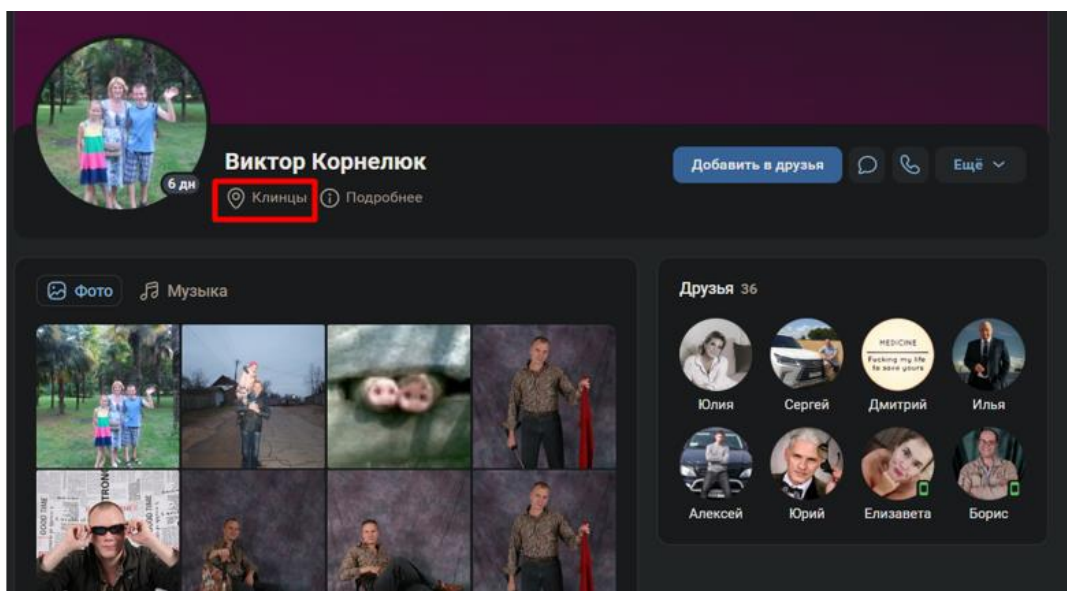


Рисунок 3.25 – Результати перевірки знайдених даних в соціальній мережі vk

З вище наведеного тестування, можна зробити висновок, що розроблена система успішно здійснює пошук інформації з витоків даних та онлайн

ресурсів та знаходить достовірну інформацію. Нижче наведено приклад ще одного звіту пошуку інформації за ПІБ та датою народження.

# БЕРЕНТ ЕКАТЕРИНА ПАВЛОВНА

27.07.1988

---

**Пов'язані email адреси**  
 kass27@yandex.ru  
 LLAKOMKAS27@YANDEX.RU

---

**Пов'язані адреси**  
 одиночно 1-ая вокзальная д 46 кв 17

---

**Документи**  
 Паспорт гражданина РФ:  
 -4614738861 тп №01 межрайонного оуфмс россии по мо в городском поселении одиночно 13/01/2015

СНИЛС:  
 ✗ Не найдено ✗  
 ИНН:  
 ✗ Не найдено ✗

---

**Пов'язані номери телефонів**  
 79055448441

---

**Пов'язані транспортні засоби**  
 Номерний знак: С490ТК197  
 ПІН: WDB12322010104834  
 Модель Авто: МЕРСЕДЕС-БЕНЦ Е200  
 Колір:  
 Рік: 1982

---

**Соціальні мережі**  
 ✗ Не найдено ✗

Рисунок 3.26 – Результат пошуку за ПІБ та датою народження

На рисунку 3.27 видно, що всі запити пошуку за ПІБ та датою народження записані в таблицю query\_history.

|   | query_id           | query_type     | query            | query_from | st... | response          | created_at          |
|---|--------------------|----------------|------------------|------------|-------|-------------------|---------------------|
| 1 | 52 5339c432-bc9... | full_name_date | Вакуленко Ек...  | 5968137855 | 200   | {'not_found': ... | 2022-12-21 21:18:47 |
| 2 | 51 bec15a1b-0d4... | full_name_date | Берент Екате...  | 5968137855 | 200   | {'email': ['k...  | 2022-12-21 21:14:22 |
| 3 | 50 92d92e00-35a... | full_name_date | Корнелиюк Але... | 5968137855 | 200   | {'email': ['a...  | 2022-12-21 21:03:51 |

Рисунок 3.27 – Запис даних запиту в таблиці query\_history



Для зручності було створено панель адміністрування, на рисунку 3.28 зображено історію всіх запитів. Натиснувши на будь-який запит можна переглянути детальну інформацію рис. 3.29.

| #  | Тип запиту     | Запит                                  | Користувач | Статус | Дата                |
|----|----------------|--|------------|--------|---------------------|
| 52 | full_name_date | Вакулєнко Катерина Павловна 27.07.1988 | 5968137855 | 200    | 2022-12-21 21:18:47 |
| 51 | full_name_date | Берент Катерина Павловна 27.07.1988    | 5968137855 | 200    | 2022-12-21 21:14:22 |
| 50 | full_name_date | Корнєлок Алексєй Викторович 31.12.2001 | 5968137855 | 200    | 2022-12-21 21:03:51 |
| 33 | email          | tanya.kozhina.19@list.ru               | 5968137855 | 200    | 2022-12-21 20:26:40 |
| 32 | email          | kass27@yandex.ru                       | 5968137855 | 200    | 2022-12-21 20:26:16 |
| 31 | email          | naumchakdmytro@gmail.com               | 5968137855 | 200    | 2022-12-21 20:24:02 |
| 30 | email          | tanya.kozhina.19@list.ru               | 5968137855 | 200    | 2022-12-21 20:22:47 |
| 29 | email          | kass27@yandex.ru                       | 5968137855 | 200    | 2022-12-21 20:21:50 |
| 28 | email          | kanaeva.1975@mail.ru                   | 5968137855 | 200    | 2022-12-21 20:21:09 |
| 27 | phone          | 79197279305                            | 5968137855 | 200    | 2022-12-21 20:17:57 |

Рисунок 3.28 – Вигляд сторінки історії запитів

Результат запиту bec15a1b-0d43-44f9-8020-2e7eba3f48a6

Telegram id користувача: 5968137855

Тип пошуку: full\_name\_date

Запит: Берент Катерина Павловна 27.07.1988

Статус: 200

Результат: {"email": ["kass27@yandex.ru", "LLAKOMKAS27@YANDEX.RU", "LLAKOMKAS27@YANDEX.RU"], "phone": ["79055449441"], "addr": ["одиночво 1-ав вокзальная д 46 кв 17"], "doc": {"passport": ["4614738861 тп №1 межрайонного оуфмс россия по мо в городском поселении одиночво 13/01/2015"], "snils": [], "inn": [], "social": [], "transport": [{"car\_plate": "С490ТК197", "vin": "WDB12322010104834", "car\_model": "МЕРСЕДЕС-БЕНЦ Е200", "car\_color": "", "car\_year": 1982}, {"car\_plate": "С490ТК197", "vin": "WDB12322010104834", "car\_model": "МЕРСЕДЕС-БЕНЦ Е200", "car\_color": "", "car\_year": 1982}]}}

Рисунок 3.29 – Вигляд вікна історії запитів

У вкладці «Список користувачів» реалізовано функцію надання доступу до користуванням додатком рис. 3.30. Щоб додати нового користувача потрібно ввести його телеграм id. Також реалізовано можливість редагувати та видаляти користувачів.

tg\_id

введіть id користувача від telegram

Додати

| # | tg_id      | Кількість запитів | Додано              | Дії                 |
|---|------------|-------------------|---------------------|---------------------|
| 1 | 5968137855 | 123               | 2022-12-15 19:30:48 | Редагувати Видалити |

Рисунок 3.30 – Вигляд сторінки списку користувачів

Порівнюємо результати пошуків розроблюваної системи з аналогом. Для порівняння було виконано пошук інформації за ПІБ, за електронною поштою та номером телефону. Результати пошуку за ПІБ наведено на рис. 3.21 та рис. 3.31.

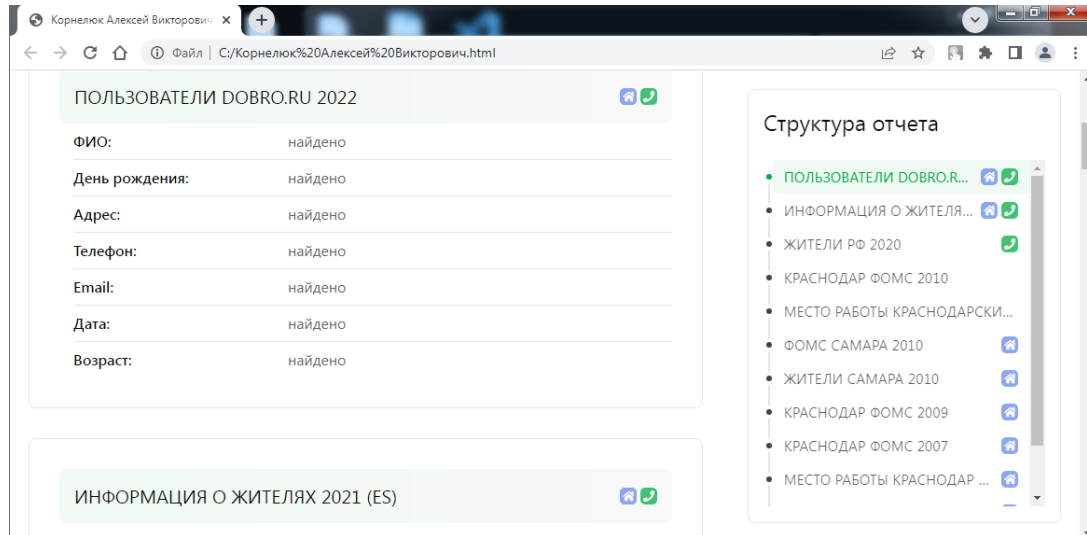


Рисунок 3.31 – Результат пошуку за ПІБ в аналога

Як видно з результатів пошуку за ПІБ, аналог майже ніяк не структурує знайдені дані. Лише вказує джерело витоку даних і знайдену інформацію звідти, що значно сповільнює процес аналізу та пошуку важливих даних користувачам. В розроблюваній системі все структуровано по категоріям, що значно зручніше і пришвидшує аналіз знайдених даних користувачам. Результати пошуку за електронною поштою наведено на рис. 3.32 та рис. 3.33.

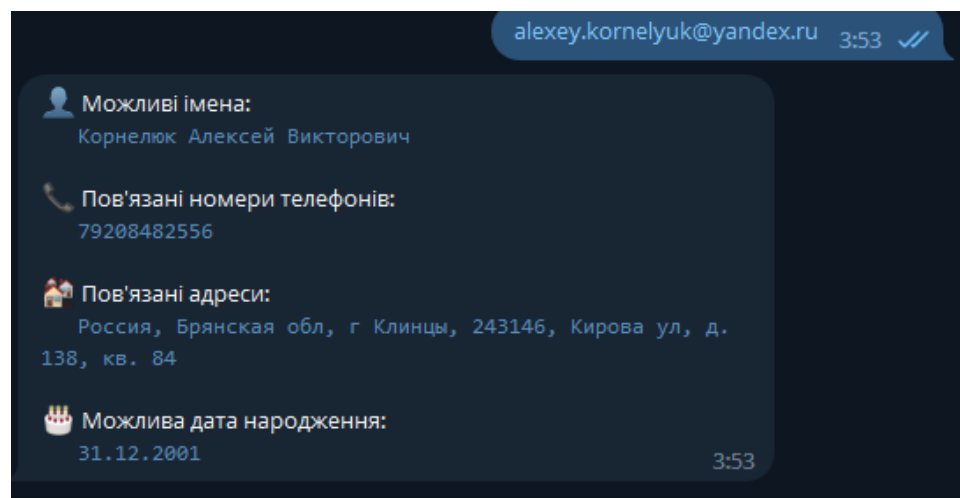


Рисунок 3.32 – Результат пошуку за електронною поштою в розроблюваній системі

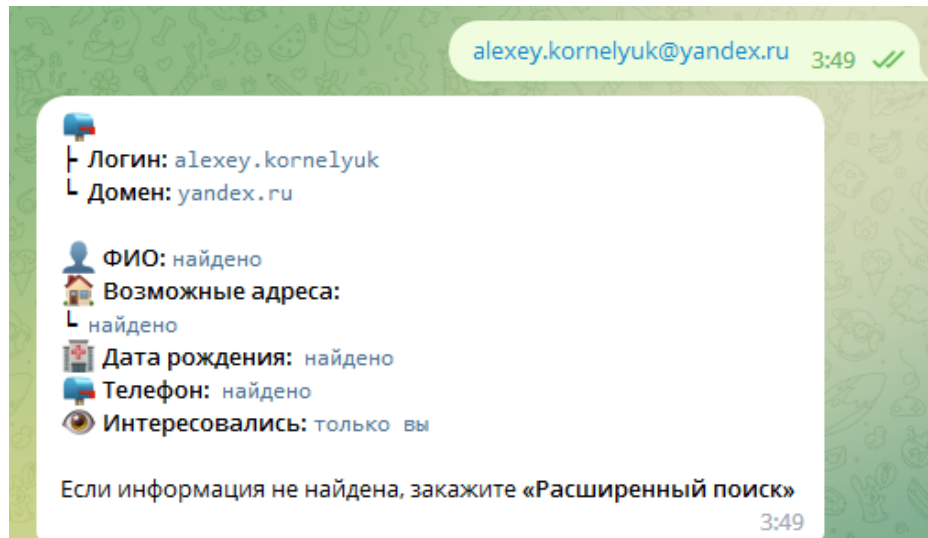


Рисунок 3.33 – Результат пошуку за електронною поштою в аналога

Як видно з рисунків обидві системи знаходять один і той самий вид інформації. Результати пошуку за номером телефону наведено на рис. 3.10 та рис. 3.34.

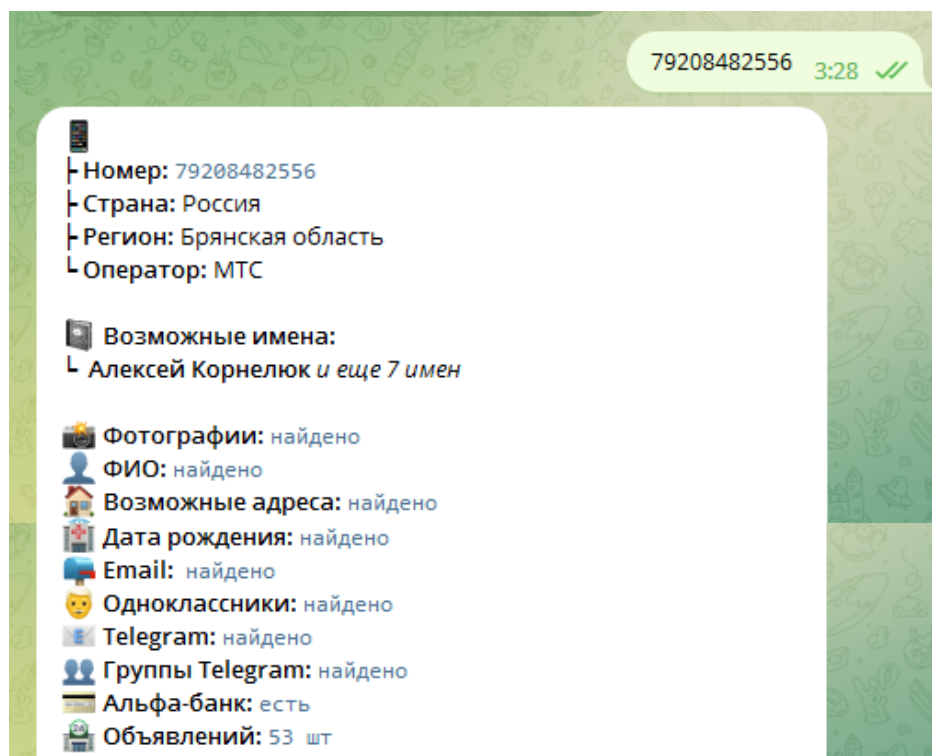


Рисунок 3.34 – Результат пошуку за номером телефону в аналога

З результатів пошуку інформації за номером телефону видно, що аналог знаходить більше інформації ніж розроблювальна система. І вказує на те що,

для поліпшення процесу пошуку інформації в розроблювальній системі, в майбутньому слід додати більше сервісів та витоків даних.

В даному розділі було обґрунтовано програмні засоби які використовувались при розробці інформаційної технології. Описано структури клієнтської та серверної частини. Для зручності було створено панель адміністрування, в якій можна переглядати історію запитів та додавати нових користувачів. Виконано тестування додатку та порівняння з аналогами. Як видно з вище показаних результатів, всі описані функції додатку працюють вірно, як це передбачено алгоритмами.

## 4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Інформаційна технологія збору інформації про людину з відкритих джерел» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

### **4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія збору інформації про людину з відкритих

джерел» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [59].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

| Бали (за 5-ти бальною шкалою)           |  |   |   |   |  |
|---|--|---|---|---|--|
|   | 0  | 1   | 2   | 3   | 4  |
| <b>Технічна здійсненність концепції</b> |  |   |   |   |  |
| 1                                       | Достовірність концепції не підтверджена                                | Концепція підтверджена експертними висновками                         | Концепція підтверджена розрахунками                             | Концепція перевірена практиці   | Перевірено на працездатність продукту в реальних умовах                |
| <b>Ринкові переваги (недоліки)</b>      |  |   |   |   |  |
| 2                                       | Багато аналогів на малому ринку  | Мало аналогів на малому ринку   | Кілька аналогів на великому ринку                               | Один аналог на великому ринку   | Продукт не має аналогів на великому ринку                              |
| 3                                       | Ціна продукту значно вища за ціни аналогів                             | Ціна продукту дещо вища за ціни аналогів                              | Ціна продукту приблизно дорівнює цінам аналогів                 | Ціна продукту дещо нижче за ціни аналогів                             | Ціна продукту значно нижче за ціни аналогів                            |
| 4                                       | Технічні та споживчі властивості продукту значно гірші, ніж в аналогів | Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів | Технічні та споживчі властивості продукту на рівні аналогів     | Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів | Технічні та споживчі властивості продукту значно кращі, ніж в аналогів |
| 5                                       | Експлуатаційні витрати значно вищі, ніж в аналогів                     | Експлуатаційні витрати дещо вищі, ніж в аналогів                      | Експлуатаційні витрати на рівні експлуатаційних витрат аналогів | Експлуатаційні витрати трохи нижчі, ніж в аналогів                    | Експлуатаційні витрати значно нижчі, ніж в аналогів                    |
| <b>Ринкові перспективи</b>              |  |   |   |   |  |
| 6                                       | Ринок малий і не має позитивної динаміки                               | Ринок малий, але має позитивну динаміку                               | Середній ринок з позитивною динамікою                           | Великий стабільний ринок  | Великий ринок з позитивною динамікою                                   |
| 7                                       | Активна конкуренція великих компаній на ринку                          | Активна конкуренція   | Помірна конкуренція   | Незначна конкуренція  | Конкурентів немає  |

Продовження таблиці 4.1

| Практична здійсненність |   |  |   |   |   |
|-------------------------|---|--|---|---|---|
| 8                       | Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї  | Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців  | Необхідне незначне навчання фахівців та збільшення їх штату   | Необхідне незначне навчання фахівців  | Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї              |
| 9                       | Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні   | Потрібні незначні фінансові ресурси. Джерела фінансування відсутні   | Потрібні значні фінансові ресурси. Джерела фінансування є   | Потрібні незначні фінансові ресурси. Джерела фінансування є                               | Не потребує додаткового фінансування  |
| 10                      | Необхідна розробка нових матеріалів   | Потрібні матеріали, що використовуються у військово-промисловому комплексі   | Потрібні дорогі матеріали   | Потрібні дешеві та досяжні матеріали  | Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві    |
| 11                      | Термін реалізації ідеї більший за 10 років  | Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років  | Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років                       | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років |
| 12                      | Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту | Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу | Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу | Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту  | Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту       |

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

| Критерії  | Експерт (ПІБ, посада) |    |    |
|---|-----------------------|----|----|
|   | 1                     | 2  | 3  |
|   | Бали:                 |    |    |
| 1. Технічна здійсненність концепції                         | 4                     | 5  | 4  |
| 2. Ринкові переваги (наявність аналогів)                    | 2                     | 2  | 3  |
| 3. Ринкові переваги (ціна продукту)                         | 2                     | 3  | 3  |
| 4. Ринкові переваги (технічні властивості)                  | 1                     | 2  | 2  |
| 5. Ринкові переваги (експлуатаційні витрати)                | 2                     | 2  | 2  |
| 6. Ринкові перспективи (розмір ринку)                       | 2                     | 2  | 2  |
| 7. Ринкові перспективи (конкуренція)                        | 3                     | 2  | 3  |
| 8. Практична здійсненність (наявність фахівців)             | 5                     | 5  | 5  |
| 9. Практична здійсненність (наявність фінансів)             | 3                     | 4  | 3  |
| 10. Практична здійсненність (необхідність нових матеріалів) | 4                     | 5  | 5  |
| 11. Практична здійсненність (термін реалізації)             | 4                     | 4  | 4  |
| 12. Практична здійсненність (розробка документів)           | 4                     | 3  | 4  |
| Сума балів  | 36                    | 39 | 40 |
| Середньоарифметична сума балів $СБ_c$                       | 38,3                  |    |    |

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [59].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

| Середньоарифметична сума балів $СБ_c$ , розрахована на основі висновків експертів | Науково-технічний рівень та комерційний потенціал розробки |
|---|--|
| 41...48   | Високий  |
| 31...40   | Вище середнього  |
| 21...30   | Середній   |
| 11...20   | Нижче середнього   |
| 0...10  | Низький  |

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія збору інформації про людину з відкритих



джерел» становить 38,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

#### 4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення розраховуємо за формулою [60]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де  $k$  – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

$\alpha_i$  – коефіцієнт, який враховує питому вагу  $i$ -го технічного показника в загальній якості розробки. Коефіцієнт  $\alpha_i$  визначається експертним шляхом і

при цьому має виконуватись умова  $\sum_{i=1}^k \alpha_i = 1$ ;

$\beta_i$  – відносне значення  $i$ -го технічного показника якості нової розробки.

Відносні значення  $\beta_i$  для різних випадків розраховуємо за такими формулами:

для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де  $I_{ni}$  та  $I_{na}$  – чисельні значення конкретного  $i$ -го технічного показника якості відповідно для нової розробки та аналога;

для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

| Показники (параметри)              | Одиниця вимірювання | Аналог | Проектований пристрій | Відношення параметрів нової розробки до аналога | Питома вага показника |
|------------------------------------|---------------------|--------|-----------------------|---|-----------------------|
| Рівень унікальності                | бал                 | 6      | 8                     | 1,33  | 0,15                  |
| Швидкість пошуку                   | с                   | 0,1    | 0,06                  | 1,67  | 0,25                  |
| Захищеність системи                | %                   | 65     | 80                    | 1,23  | 0,1                   |
| Точність ідентифікації об'єкта     | %                   | 78     | 94                    | 1,21  | 0,3                   |
| Доступність інтерфейсу користувача | бал                 | 6,2    | 8,5                   | 1,37  | 0,2                   |

Узагальнений коефіцієнт якості ( $B_n$ ) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,33 \cdot 0,15 + 1,67 \cdot 0,25 + 1,23 \cdot 0,1 + 1,21 \cdot 0,3 + 1,37 \cdot 0,2 = 1,38.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,38 рази.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія збору інформації про людину з відкритих джерел», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

#### 4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

#### Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [59]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=21$  дні.

$$Z_o = 17020,00 \cdot 38 / 21 = 30798,10 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

| Найменування посади                        | Місячний посадовий оклад, грн | Оплата за робочий день, грн | Число днів роботи | Витрати на заробітну плату, грн |
|--|-------------------------------|-----------------------------|-------------------|---------------------------------|
| Керівник проекту                           | 17020,00                      | 810,48                      | 38                | 30798,10                        |
| Інженер-розробник програмного забезпечення | 16750,00                      | 797,62                      | 36                | 28714,29                        |
| Технік                                     | 7320,00                       | 348,57                      | 18                | 6274,29                         |
| Всього                                     |                               |                             |                   | 65786,67                        |

#### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія збору інформації про людину з відкритих джерел» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [59];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днїв в мїсяцї, приблизно  $T_p = 21$  дн;

$t_{зм}$  – тривалїсть змїни, год.

$$C_l = 6700,00 \cdot 1,10 \cdot 1,65 / (21 \cdot 8) = 72,38 \text{ грн.}$$

$$Z_{pl} = 72,38 \cdot 6,00 = 434,30 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробїтну плату робїтників

| Найменування робїт   | Тривалїсть роботи, год | Розряд роботи | Тарифний коефїцієнт | Погодинна тарифна ставка, грн | Величина оплати на робїтника грн |
|--|------------------------|---------------|---------------------|-------------------------------|----------------------------------|
| Пїдготовка робочого мїсця інженера-розробника програмного забезпечення | 6,00                   | 2             | 1,10                | 72,38                         | 434,30                           |
| Інсталяцїя програмного забезпечення середовища розробки                | 5,10                   | 3             | 1,35                | 88,83                         | 453,06                           |
| Формування кодїв програмних блоків                                     | 9,20                   | 5             | 1,70                | 111,87                        | 1029,17                          |
| Формування бази даних дослїдження                                      | 14,50                  | 2             | 1,10                | 72,38                         | 1049,57                          |
| Контроль проходження програмних експериментїв                          | 7,80                   | 4             | 1,50                | 98,71                         | 769,90                           |
| Всього   |                        |               |                     |                               | 3736,00                          |

Додаткова заробїтна плата дослїдникїв та робїтникїв

Додаткову заробїтну плату розраховуємо як 10 ... 12% вїд суми основної заробїтної плати дослїдникїв та робїтникїв за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.7)$$

де  $H_{\text{дод}}$  – норма нарахування додаткової заробїтної плати. Приймемо 10%.

$$Z_{\text{дод}} = (65786,67 + 3736,00) \cdot 10 / 100\% = 6952,27 \text{ грн.}$$

### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{zn}}{100\%} \quad (4.8)$$

де  $H_{zn}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (65786,67 + 3736,00 + 6952,27) \cdot 22 / 100\% = 16824,48 \text{ грн.}$$

### 4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія збору інформації про людину з відкритих джерел».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (4.9)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$C_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{\text{в}j}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 4,0 \cdot 220,00 \cdot 1,11 - 0 \cdot 0 = 976,80 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

| Найменування матеріалу, марка, тип, сорт                            | Ціна за 1 кг, грн | Норма витрат, кг | Величина відходів, кг | Ціна відходів, грн/кг | Вартість витраченого матеріалу, грн |
|---|-------------------|------------------|-----------------------|-----------------------|-------------------------------------|
| Папір офісний UPM-Kymmene A4 80 г/м NEW Future Laser білий          | 220,00            | 4,0              | 0                     | 0                     | 976,80                              |
| Папір Crystal A5 80 г/м PRO білий                                   | 65,00             | 3,0              | 0                     | 0                     | 216,45                              |
| Набір настільний Transformers BumbleBee Movie TF19-205 KITE         | 125,00            | 4,0              | 0                     | 0                     | 555,00                              |
| Набір канцелярський Milan Silver (08737)                            | 449,00            | 2,0              | 0                     | 0                     | 996,78                              |
| Тонер IPM HP LJ P1005/1006/150 5/CANON LBP-3010/3020 (TSH87B) black | 520,00            | 2,0              | 0                     | 0                     | 1154,40                             |
| Диск оптичний CD-RW   | 21,00             | 3,0              | 0                     | 0                     | 69,93                               |
| USB флеш накопичувач Transcend 64Gb JetFlash 700 (TS64GJF700)       | 290,00            | 1,0              | 0                     | 0                     | 321,90                              |
| Всього  |                   |                  |                       |                       | 4291,26                             |

#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_6$ ), які використовують при проведенні НДР на тему «Інформаційна технологія збору інформації про людину з відкритих джерел», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (4.10)$$

де  $H_j$  – кількість комплектуючих  $j$ -го виду, шт.;

$C_j$  – покупна ціна комплектуючих  $j$ -го виду, грн;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ ).

$$K_6 = 1 \cdot 5120,00 \cdot 1,11 = 5683,20 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

| Найменування комплектуючих   | Кількість, шт. | Ціна за штуку, грн | Сума, грн |
|--|----------------|--------------------|-----------|
| Графічний адаптер: - тип відеокарти Дискретна - відеокарта GeForce GT710 - об'єм відеопам'яті 2 ГБ | 1              | 5120,00            | 5683,20   |
| Концентратор Speedlink SNAPPY SLIM USB Hub, 4-Port, USB 2.0, Passive, Black                        | 1              | 250,00             | 277,50    |
| Кабель для передачі даних SATA III 1.0m Cablexpert (CC-SATAM-DATA-XL)                              | 1              | 50,00              | 55,50     |
| Всього   |                |                    | 6016,20   |

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:



$$B_{спец} = \sum_{i=1}^k C_i \cdot C_{пр.i} \cdot K_i, \quad (4.11)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$  – кількість одиниць устаткування відповідного найменування, які

придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  – кількість найменувань устаткування.

$$B_{спец} = 3329,00 \cdot 1 \cdot 1,11 = 3695,19 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

| Найменування устаткування  | Кількість, шт | Ціна за<br>одиницю, грн | Вартість, грн |
|--|---------------|-------------------------|---------------|
| Зовнішній жорсткий диск 2.5" 2TB Game Drive for Xbox Seagate (STKX2000400) | 1             | 3329,00                 | 3695,19       |
| Комутатор мережевий TP-Link TL-SG1005D                                     | 1             | 780,00                  | 865,80        |
| Маршрутизатор Mikrotik hAP ac (RBD52G-5HacD2HnD-TC)                        | 1             | 3000,00                 | 3330,00       |
| Всього   |               |                         | 7890,99       |

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прог} = \sum_{i=1}^k C_{инрг} \cdot C_{прог.i} \cdot K_i, \quad (4.12)$$

де  $C_{инрг}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг.}i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1,10 \dots 1,12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 150,00 \cdot 1 \cdot 1,11 = 166,50 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

| Найменування програмного засобу           | Кількість, шт | Ціна за одиницю, грн | Вартість, грн |
|---|---------------|----------------------|---------------|
| Середовища розробки: - Visual Studio Code | 1             | 150,00               | 166,50        |
| - PyCharm Community                       | 1             | 110,00               | 122,10        |
| Доступ до мережі Інтернет "DATA GROUP"    | 1             | 800,00               | 888,00        |
| Всього                                    |               |                      | 1176,60       |

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{е}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.13)$$

де  $Ц_{\text{б}}$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{е}}$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (41450,00 \cdot 2) / (2 \cdot 12) = 3454,17 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

| Найменування обладнання   | Балансова вартість, грн | Строк корисного використання, років | Термін використання обладнання, місяців | Амортизаційні відрахування, грн |
|---|-------------------------|-------------------------------------|---|---------------------------------|
| Персональний комп'ютер проведення розробки Vinga Creator A1087 (I7M32INT.A1087) | 41450,00                | 2                                   | 2                                       | 3454,17                         |
| Робоче місце інженера-розробника ПЗ   | 8700,00                 | 5                                   | 2                                       | 290,00                          |
| Пристрої передачі даних комутатор мережевий TP-Link                             | 2780,00                 | 2                                   | 2                                       | 231,67                          |
| Пристрій виводу інформації  | 6740,00                 | 5                                   | 2                                       | 224,67                          |
| Оргтехніка  | 7250,00                 | 4                                   | 2                                       | 302,08                          |
| Приміщення лабораторії  | 675200,00               | 25                                  | 2                                       | 4501,33                         |
| ОС Windows 10   | 8370,00                 | 2                                   | 2                                       | 697,50                          |
| Прикладний пакет Microsoft Office 2016  | 7825,00                 | 2                                   | 2                                       | 652,08                          |
| Всього  |                         |                                     |   | 10353,50                        |

## 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{vni}}{\eta_i}, \quad (4.14)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 6,20$  грн;

$K_{eni}$  – коефіцієнт, що враховує використання потужності,  $K_{eni} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$B_e = 0,32 \cdot 300,0 \cdot 6,20 \cdot 0,95 / 0,97 = 595,20$  грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

| Найменування обладнання   | Встановлена потужність, кВт | Тривалість роботи, год | Сума, грн |
|---|-----------------------------|------------------------|-----------|
| Персональний комп'ютер проведення розробки Vinga Creator A1087 (I7M32INT.A1087) | 0,32                        | 300,0                  | 595,20    |
| Робоче місце інженера-розробника ПЗ   | 0,15                        | 300,0                  | 279,00    |
| Пристрої передачі даних комутатор мережевий TP-Link                             | 0,01                        | 600,0                  | 37,20     |
| Пристрій виводу інформації  | 0,50                        | 30,0                   | 93,00     |
| Оргтехніка  | 0,62                        | 6,0                    | 23,06     |
| Комутатор мережевий TP-Link TL-SG1005D  | 0,01                        | 600,0                  | 37,20     |
| Маршрутизатор Mikrotik hAP ac (RBD52G-5HacD2HnD-TC)                             | 0,02                        | 600,0                  | 74,40     |
| Всього  |                             |                        | 1139,06   |

#### 4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія збору інформації про людину з відкритих джерел» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження

на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» відсутні.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.15)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 30\%$ .

$$B_{cn} = (65786,67 + 3736,00) \cdot 30 / 100\% = 20856,80 \text{ грн.}$$

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (4.16)$$

де  $H_{is}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{is} = 50\%$ .

$$I_s = (65786,67 + 3736,00) \cdot 50 / 100\% = 34761,33 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів;

витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.17)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 100\%$ .

$$B_{нзв} = (65786,67 + 3736,00) \cdot 100 / 100\% = 69522,66 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія збору інформації про людину з відкритих джерел» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{доо} + Z_n + M + K_в + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_в + B_{нзв}. \quad (4.18)$$

$$B_{заг} = 65786,67 + 3736,00 + 6952,27 + 16824,48 + 4291,26 + 6016,20 + 7890,99 + 1176,60 + 10353,50 + 1139,06 + 0,00 + 20856,80 + 34761,33 + 69522,66 = 249307,83 \text{ грн.}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.19)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo  $\eta=0,9$ .

$$ZB = 249307,83 / 0,9 = 277008,70 \text{ грн.}$$

#### 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія збору інформації про людину з відкритих джерел» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

$\Delta N$  – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

| Показник                              | 1-й рік | 2-й рік | 3-й рік | 4-й рік |
|---------------------------------------|---------|---------|---------|---------|
| Збільшення кількості споживачів, осіб | 600     | 950     | 1000    | 500     |

$N$  – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 10000 осіб;

$C_o$  – вартість послуги у році до впровадження інформаційної системи, прийmemo 7500,00 грн;

$\pm \Delta C_o$  – зміна вартості послуги від впровадження результатів, прийmemo 327,50 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [59]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.20)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2022 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo  $\rho = 35\%$ ;

$\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2022 році  $\vartheta = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (327,50 \cdot 10000,00 + 7827,50 \cdot 600) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 1898891,02 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (327,50 \cdot 10000,00 + 7827,50 \cdot 1550) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 3670250,35$$

грн.

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (327,50 \cdot 10000,00 + 7827,50 \cdot 2550) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 5534839,13$$

грн.

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (327,50 \cdot 10000,00 + 7827,50 \cdot 3050) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 6467133,51$$

грн.

Приведена вартість збільшення всіх чистих прибутків  $ПП$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.21)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;



$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau=0,26$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 1898891,02/(1+0,26)^1 + 3670250,35/(1+0,26)^2 + 5534839,13/(1+0,26)^3 + \\ &+ 6467133,51/(1+0,26)^4 = 1507056,36 + 2311823,10 + 2766899,39 + 2565840,44 = 9151 \\ &619,29 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.22)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв}=2,1$ ;

$3B$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 277008,70 грн.

$$PV = k_{инв} \cdot 3B = 2,1 \cdot 277008,70 = 581718,26 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (4.23)$$

де  $ПП$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9151619,29 грн;

$PV$  – теперішня вартість початкових інвестицій, 581718,26 грн.

$E_{abc} = ПП - PV = 9151619,29 - 581718,26 = 8569901,03$  грн.

Внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = T_{жс} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.24)$$

де  $E_{abc}$  – абсолютний економічний ефект вкладених інвестицій, 8569901,03 грн;

$PV$  – теперішня вартість початкових інвестицій, 581718,26 грн;

$T_{жс}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_g = T_{жс} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 8569901,03/581718,26)^{1/4} = 0,99.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{min}$ :

$$\tau_{min} = d + f, \quad (4.25)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = 0,1$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,27.

$\tau_{min} = 0,1 + 0,27 = 0,37 < 0,99$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_g$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну

роботу за темою «Інформаційна технологія збору інформації про людину з відкритих джерел» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.26)$$

де  $E_g$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,99 = 1,01 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія збору інформації про людину з відкритих джерел» становить 38,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,38 рази.

Також термін окупності становить 1,01 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія збору інформації про людину з відкритих джерел».

## ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було проаналізовано завдання, визначені завдання та поставлені вимоги, дотримавшись яких, можна досягти поставленої мети.

Було проаналізовано основні джерела для OSINT розслідувань. З цього випливає що отримати максимальну віддачу від OSINT, потрібен детальний аналіз та розуміння вимог щодо його використання. Крім того, OSINT не виключає звичайні розвідувальні дані, зібрані із секретних джерел, тому вибір на користь використання OSINT не повинен ґрунтуватися на вартості та традиційних розвідувальних даних. Поєднання OSINT із традиційними джерелами розвідданих – потужний підхід до бізнес-аналітики.

Також були проаналізовані популярні OSINT інструменти, які можуть бути використані для пошуку інформації про людину. Усі вони мають різні категорії, функції та особливості.

Створено модель інформаційної технології, яка включає в себе три процеси: процес отримання параметрів, процес пошуку інформації та процес формування результату пошуку за заданими параметрами. Для кожного з процесів було описано його математичну модель та загальну математичну модель інформаційної технології.

На основі моделі інформаційної технології було розроблено архітектуру додатку, яка складається з трьох частин: клієнтської частини, серверної частини та бази даних. Після створення архітектури та на основі поставлених вимог, були розроблені алгоритми функціонування системи.

Після аналізу інструментів розробки, для клієнтської частини було вирішено використовувати фреймворк Aiogram та для серверної частини обрано фреймворк Laravel.

Дотримуючись вимог, виставлених на етапі постановки задачі, та за допомогою розробленої архітектури та алгоритмів було реалізовано telegram-bot, який виконує всі поставлені задачі. Також було створено панель

адміністрування, завдяки якій зручно переглядати історію запитів та додавати нових користувачів.

Виконано тестування додатку пошуком за різними параметрами та порівняння з аналогами. Під час тестування помилок в роботі виявлено не було.

Було проведено комерційних аудит розробки, розраховано узагальнений коефіцієнт якості розробки, проаналізовані витрати, яких потребує розробка інформаційної технології. Після аналізу всіх витрати, було пораховано, що окупність розробленої системи становить  $\sim 1,01$  року, що менше трьох років і це дає змогу зробити висновок, що розробка інформаційної технології є доцільною.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jennifer Golbeck. Introduction to Social Media Investigation: A Hands-on Approach: Syngress, 2015. 13 с.
2. A Brief History of Open Source Intelligence: веб-сайт. URL: <https://www.bellingcat.com/resources/articles/2016/07/14/a-brief-history-of-open-source-intelligence/> (дата звернення: 01.09.2022)
3. OSINT 2021 guide: tools and techniques for threat intelligence: веб-сайт. URL: <https://www.authentic8.com/blog/OSINT-2021-guide-tools-and-techniques> (дата звернення: 01.09.2022)
4. The value of open source intelligence in the 21st century: веб-сайт. URL: <https://www.expert.ai/blog/value-open-source-intelligence-21st-century/> (дата звернення: 02.09.2022)
5. The “Grey” Intersection of Open Source information and Intelligence. URL: [http://www.greynet.org/images/GL10,\\_page\\_82.pdf](http://www.greynet.org/images/GL10,_page_82.pdf) (дата звернення: 02.09.2022)
6. Open-source intelligence: веб-сайт. URL: <https://data.europa.eu/es/node/10386> (дата звернення: 05.09.2022)
7. The largest investigation in journalism history exposes a shadow financial system that benefits the world’s most rich and powerful.: веб-сайт. URL: <https://www.icij.org/investigations/pandora-papers/> (дата звернення: 05.09.2022)
8. Open to the public: paywalls and the public rationale for open access medical research publishing.: веб-сайт. URL: [https://www.researchgate.net/publication/339580287\\_Open\\_to\\_the\\_public\\_paywalls\\_and\\_the\\_public\\_rationale\\_for\\_open\\_access\\_medical\\_research\\_publishing](https://www.researchgate.net/publication/339580287_Open_to_the_public_paywalls_and_the_public_rationale_for_open_access_medical_research_publishing) (дата звернення: 06.09.2022)
9. Open source intelligence reader. URL: <https://cyberwar.nl/d/NATO%20OSINT%20Reader%20FINAL%20Oct2002.pdf> (дата звернення: 06.09.2022)
10. A Guide To Social Media Intelligence Gathering (SOCMINT): веб-сайт. URL: <https://www.secjuice.com/social-media-intelligence-socmint/> (дата звернення: 07.09.2022)

11. Open Source Intelligence: What Social Media Can Tell Us: веб-сайт. URL: <https://www.cameraforensics.com/blog/2020/02/14/open-source-intelligence-what-social-media-can-tell-us/> (дата звернення: 07.09.2022)
12. Everything About Social Media Intelligence (SOCMINT) and Investigations: веб-сайт. URL: <https://www.maltego.com/blog/everything-about-social-media-intelligence-socmint-and-investigations/> (дата звернення: 08.09.2022)
13. Promoting Open Source Intelligence Techniques to Investigate Corruption in Southeast Asia: веб-сайт. URL: <https://www.unodc.org/roseap/en/what-we-do/anti-corruption/topics/2021/01-open-source-intelligence-techniques.html> (дата звернення: 08.09.2022)
14. Top Open Source Intelligence (OSINT) Tools for Dark Web: веб-сайт. URL: <https://brandefense.io/blog/dark-web/top-open-source-intelligence-osint-tools-for-dark-web/> (дата звернення: 09.09.2022)
15. How to Collect OSINT on the Dark Web: веб-сайт. URL: <https://www.liferaftinc.com/blog/how-to-collect-osint-on-the-dark-web> (дата звернення: 09.09.2022)
16. Dark Web Searching: веб-сайт. URL: <https://www.osintcombine.com/post/dark-web-searching> (дата звернення: 12.09.2022)
17. Automating OSINT, OPSEC, and Dark Web Research. URL: [https://www.threatngsecurity.com/s/OSINT\\_OPSEC\\_DarkWeb\\_ver2.pdf](https://www.threatngsecurity.com/s/OSINT_OPSEC_DarkWeb_ver2.pdf) (дата звернення: 12.09.2022)
18. Top 10 OSINT (Open Source Intelligence) Software Tools 2022: веб-сайт. URL: <https://seon.io/resources/the-best-tools-for-osint/> (дата звернення: 13.09.2022)
19. Open Source Intelligence (OSINT): What Is It?: веб-сайт. URL: <https://www.skopenow.com/news/open-source-intelligence-osint-what-is-it> (дата звернення: 13.09.2022)
20. All You Need to Know About Open Source Intelligence (OSINT): веб-сайт. URL: <https://www.makeuseof.com/all-you-need-to-know-about-open-source-intelligence-osint/> (дата звернення: 14.09.2022)

21. Open Source Intelligence (OSINT): issues and trends: веб-сайт. URL: [https://www.researchgate.net/publication/340301223\\_Open\\_Source\\_Intelligence\\_OSINT\\_issues\\_and\\_trends](https://www.researchgate.net/publication/340301223_Open_Source_Intelligence_OSINT_issues_and_trends) (дата звернення: 14.09.2022)
22. Get Contact. URL: [https://t.me/get\\_kontakt\\_bot](https://t.me/get_kontakt_bot) (дата звернення: 15.09.2022)
23. AvinfoBot. URL: <https://t.me/AVinfoBot> (дата звернення: 15.09.2022)
24. SmartSearchBot. URL: [https://t.me/smart\\_searchbot](https://t.me/smart_searchbot) (дата звернення: 15.09.2022)
25. Quick OSINT. URL: [https://t.me/Quick\\_OSINT\\_bot](https://t.me/Quick_OSINT_bot) (дата звернення: 15.09.2022)
26. TeleSINT. URL: [https://t.me/telesint\\_bot](https://t.me/telesint_bot) (дата звернення: 15.09.2022)
27. Getfb. URL: [https://t.me/getfb\\_bot](https://t.me/getfb_bot) (дата звернення: 15.09.2022)
28. Mailsearch\_bot. URL: <https://t.me/mailsearchbot> (дата звернення: 15.09.2022)
29. OpenDataUA. URL: <https://t.me/OpenDataUABot> (дата звернення: 15.09.2022)
30. Stop-list.info: веб-сайт. URL: <http://login.stop-list.info/> (дата звернення: 16.09.2022)
31. Leaked database lookup: веб-сайт. URL: [https://webresolver.nl/tools/leaked\\_database](https://webresolver.nl/tools/leaked_database) (дата звернення: 16.09.2022)
32. Saverudata.online: веб-сайт. URL: <https://saverudata.online/> (дата звернення: 16.09.2022)
33. Intelligence X: веб-сайт. URL: <https://intelx.io/> (дата звернення: 16.09.2022)
34. Shodan.io: веб-сайт. URL: <https://www.shodan.io/> (дата звернення: 16.09.2022)
35. Мамонт: веб-сайт. URL: <https://www.mmnt.ru/> (дата звернення: 16.09.2022)
36. Web.archive: веб-сайт. URL: <https://web.archive.org/> (дата звернення: 16.09.2022)



37. CiteSeerX: веб-сайт. URL: <https://citeseer.ist.psu.edu/> (дата звернення: 16.09.2022)
38. Lookup-ID.com: веб-сайт. URL: <https://lookup-id.com/> (дата звернення: 16.09.2022)
39. Spoonbill lets you see profile changes from the people you follow on Twitter or other social networks.: веб-сайт. URL: <https://spoonbill.io/> (дата звернення: 16.09.2022)
40. Gideon: веб-сайт. URL: <https://github.com/YouVBeenHacked/gideon> (дата звернення: 19.09.2022)
41. sn0int: веб-сайт. URL: <https://github.com/kpcyrd/sn0int> (дата звернення: 19.09.2022)
42. Maigret: веб-сайт. URL: <https://github.com/soxoj/maigret> (дата звернення: 19.09.2022)
43. Sherlock: веб-сайт. URL: <https://github.com/sherlock-project/sherlock> (дата звернення: 19.09.2022)
44. GHunt: веб-сайт. URL: <https://github.com/mxrch/GHunt> (дата звернення: 19.09.2022)
45. Osintgram: веб-сайт. URL: <https://github.com/Datalux/Osintgram> (дата звернення: 19.09.2022)
46. Maltego: веб-сайт. URL: <https://www.maltego.com/> (дата звернення: 19.09.2022)
47. DataSploit: веб-сайт. URL: <https://github.com/DataSploit/datasploit> (дата звернення: 19.09.2022)
48. Orbit Blockchain Transactions Investigation Tool: веб-сайт. URL: <https://github.com/s0md3v/Orbit> (дата звернення: 19.09.2022)
49. Osintstalker: веб-сайт. URL: <https://github.com/milo2012/osintstalker> (дата звернення: 19.09.2022)
50. FOCA (Fingerprinting Organizations with Collected Archives): веб-сайт. URL: <https://github.com/ElevenPaths/FOCA> (дата звернення: 19.09.2022)

51. History of PHP: веб-сайт. URL: <https://www.php.net/manual/en/history.php.php> (дата звернення: 03.10.2022)
52. Size of programming language communities worldwide as of 2022: веб-сайт. URL: <https://www.statista.com/statistics/1241923/worldwide-software-developer-programming-language-communities/#:~:text=According%20to%20the%20survey%2C%20the,programmers%2C%20with%2015.7%20million%20developers.> (дата звернення: 03.10.2022)
53. Why Django?: веб-сайт. URL: <https://www.djangoproject.com/start/overview/> (дата звернення: 03.10.2022)
54. Why Laravel?: веб-сайт. URL: <https://laravel.com/docs/9.x> (дата звернення: 03.10.2022)
55. MySQL Documentation: веб-сайт. URL: <https://dev.mysql.com/doc/> (дата звернення: 03.10.2022)
56. Welcome to aiogram's documentation!: веб-сайт. URL: <https://docs.aiogram.dev/en/latest/> (дата звернення: 03.10.2022)
57. Getting Started: веб-сайт. URL: <https://code.visualstudio.com/docs> (дата звернення: 03.10.2022)
58. The Python IDE for Professional Developers: веб-сайт. URL: <https://www.jetbrains.com/pycharm/> (дата звернення: 03.10.2022)
59. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
60. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

## **ДОДАТКИ**

## Додаток А. Результат перевірки роботи на плагіат

**ПРОТОКОЛ ПЕРЕВІРКИ  
МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Інформаційна технологія збору інформації про людину з відкритих джерел

Автор роботи: Наумчак Дмитро Валерійович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ  
(кафедра, факультет)

**Показники звіту подібності Unicheck**

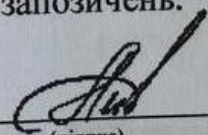
Оригінальність – 97.4%.

Схожість – 2.6%.

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

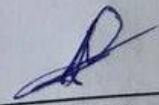
Особа, відповідальна за перевірку

  
(підпис)

Каплун В. А.  
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи

  
(підпис)

Наумчак Д. В.  
(прізвище, ініціали)

Керівник роботи

  
(підпис)

Гуцаристей Д. В.  
(прізвище, ініціали)

## Додаток Б. Лістинг програми

### Клієнтська частина

#### Файл app.py

```

from aiogram import executor

from loader import dp
import middlewares, filters, handlers
from utils.notify_admins import on_startup_notify
from utils.set_bot_commands import set_default_commands

async def on_startup(dispatcher):
    # Встановлюємо дефолтні команди
    await set_default_commands(dispatcher)

    # Повідомляє про запуск
    await on_startup_notify(dispatcher)

if __name__ == '__main__':
    executor.start_polling(dp, on_startup=on_startup)

```

#### Файл loader.py

```

from aiogram import Bot, Dispatcher, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage

from data import config

bot = Bot(token=config.BOT_TOKEN, parse_mode=types.ParseMode.HTML)

storage = MemoryStorage()

dp = Dispatcher(bot, storage=storage)

```

#### Файл utils\notify\_admins.py

```
import logging
```

```

from aiogram import Dispatcher

from data.config import ADMINS

async def on_startup_notify(dp: Dispatcher):
    for admin in ADMINS:
        try:
            await dp.bot.send_message(admin, "Бот Працює")
        except Exception as err:
            logging.exception(err)

```

#### Файл utils\set\_bot\_commands.py

```

from aiogram import types

async def set_default_commands(dp):
    await dp.bot.set_my_commands(
        [
            types.BotCommand("start", "Запустити бота"),
            types.BotCommand("help", "Вивести довідку"),
        ]
    )

```

#### Файл handlers\users\start.py

```

from aiogram import types

from aiogram.dispatcher.filters.builtin import CommandStart

```

```

from loader import dp

@dp.message_handler(CommandStart())
async def bot_start(message:
types.Message):
    await message.answer(f"Вітаю
\U0001F44B,
{message.from_user.full_name}!")

```

### Файл handlers\users\help.py

```

from aiogram import types

from
aiogram.dispatcher.filters.builtin
import CommandHelp

from loader import dp

@dp.message_handler(CommandHelp())
async def bot_help(message:
types.Message):
    text = ("Список команд: ",
           "/start - старт",
           "/help - хелп")

    await
message.answer("\n".join(text))

```

### Файл handlers\users\main\main.py

```

import re

from aiogram import types

from aiogram.dispatcher import
FSMContext

from loader import dp, bot

@dp.message_handler(state=None)

```

```

async def bot_echo(message:
types.Message):

    tmp_msg = await
message.answer(f"\U0001F50D
           Шукаємо, зачекайте, будь
ласка. \U0001F50D")

    phone =
re.compile(r"^[+]?[(]?[0-9]{3}[)]?[-
\s.]?[0-9]{3}[-\s.]?[0-9]{4,6}$")

    full_name_with_bd =
re.compile(r"([^\ ]+) ([^\ ]+) ([^\ ]+)
\d{2}.\d{2}.\d{4}")

    name = re.compile(r"^([^\ ]+)|([^\
 ]+) ([^\ ]+)|([^\ ]+) ([^\ ]+) ([^\
 ]+)")

    email = re.compile(r"([-!#-'*+/-
9=?A-Z^~]+)(\. [-!#-'*+/-9=?A-Z^~
~]+)*|\\"([!#-[\^~ \t]|(\\"[\t -
~]))+\\"")@([-!#-'*+/-9=?A-Z^~]+)(\.[
!#-'*+/-9=?A-Z^~]+)*|\\"([\t -Z^~
~]*])")

    if
email.fullmatch(message.text):

        from .email_checker import
main

        try:

            email_info =
main(message.text,
message.from_user.id)

            await
message.answer(email_info)

        except requests.HTTPError as
e:

            await
message.answer(f'Not authorized
{e.response.status_code}')

            finally:

                await tmp_msg.delete()

    elif
phone.fullmatch(message.text):

        from .phone_checker import
main

        try:

```



```

def main(email, tg_id):
    try:
        r =
requests.get(f"{config.API}/info_by_
email?email={email}&tg_id={tg_id}")

        if r.status_code == 200:
            data = r.json()
            result =
report_msg(data)

requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=email&que
ry={email}&response={data}&status_co
de={r.status_code}")

            return result

            raise r.raise_for_status()
    except requests.HTTPError as e:

requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=email&que
ry={email}&response={e.response.reas
on}&status_code={e.response.status_c
ode}")

            raise e

```

## Файл handlers\users\main\phone\_checker .py

```

import re
from data import config
import requests

def report_msg(data):
    cross = "\U0000274C"
    res = ''

    if 'not_found' in data:
        res += cross + ' За цим
номером телефону нічого не знайдено
' +cross

    return res

```

```

else:
    if data['names']:
        res += f"\U0001F464
        Можливі імена:
        \n\t\t\t\t\t\t\t\t\t {'',
        '.join(['<code>' + name + '</code>'
        for name in data['names']])}\n\n"

        if data['emails']:
            res += f"\U0001F4EC
            Email: \n\t\t\t\t\t\t\t\t\t {'',
            '.join(['<code>' + email + '</code>'
            for email in data['emails']])}\n\n"

            if data['addresses']:
                res += f"\U0001F3D8
                Пов'язані адреси:
                \n\t\t\t\t\t\t\t\t\t {'', '.join('<code>'
                + address + '</code>' for address in
                data['addresses'])}\n\n"

                if data['birth']:
                    res += f"\U0001F382
                    Можлива дата народження:
                    \n\t\t\t\t\t\t\t\t\t {'', '.join('<code>'
                    + birth + '</code>' for birth in
                    data['birth'])}\n\n"

                    if data['transport']:
                        res += f"\U0001F697
                        Можливий транспортний засіб
                        (Номерний знак): \n\t\t\t\t\t\t\t\t\t {'',
                        '.join('<code>' + plate +
                        '</code>' for plate in
                        data['transport'])}\n\n"

                        if data['social']:
                            res += f"\U0001F44D
                            Соціальні мережі: \n"

                            if 'telegram' in
                            data['social']:

                                res +=
                                f"\t\t\t\t\t\t\t\t\t <i>telegram:</i>
                                {'', '.join(link['nickname']+
                                '<code>'+link['uid']+</code>' for
                                link in
                                data['social']['telegram'])}\n"

                                if 'facebook' in
                                data['social']:
                                    slash = '\\'

                                    res +=
                                    f"\t\t\t\t\t\t\t\t\t <i>facebook:</i>

```



```

{'', '.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['facebook'])}\n"

        if 'linkedin' in
data['social']:

            slash = '\\'

            res +=
f"\t\t\t\t\t\t\t\t <i>linkedin:</i>
{'', '.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['linkedin'])}\n"

        if 'mailru' in
data['social']:

            slash = '\\'

            res +=
f"\t\t\t\t\t\t\t\t <i>mailru:</i> {'',
'.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['mailru'])}\n"

        if 'pikabu' in
data['social']:

            slash = '\\'

            res +=
f"\t\t\t\t\t\t\t\t <i>pikabu:</i> {'',
'.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['pikabu'])}\n"

        if 'vk' in
data['social']:

            slash = '\\'

            res +=
f"\t\t\t\t\t\t\t\t <i>vk:</i> {'',
'.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['vk'])}\n"

        if 'instagram' in
data['social']:

            slash = '\\'

            res +=
f"\t\t\t\t\t\t\t\t <i>instagram:</i>
{'', '.join('<a
href='+slash+link['url']+slash+'>' +
link['full_name'] + '</a>' for link
in data['social']['instagram'])}\n"

    return res

def phone_type(phone, tg_id):

    rus_phone =
re.compile(r"(\+7|7|8)?[\s\-\
]?(? [489] [0-9] {2}\)\)?[\s\-\]?[0-
9] {3}[\s\-\]?[0-9] {2}[\s\-\]?[0-
9] {2}$")

    if rus_phone.fullmatch(phone):

        try:

            r =
requests.get(f"{config.API}/info_by_
phone?phone_number={phone}&tg_id={tg
_id}")

            if r.status_code == 200:

                status_code =
r.status_code

                data = r.json()

                result =
report_msg(data)

                requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=phone&que
ry={phone}&response={data}&status_co
de={status_code}")

                return result

            raise
r.raise_for_status()

        except requests.HTTPError as
e:

            requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=phone&que
ry={phone}&response={e.response.reas
on}&status_code={e.response.status_c
ode}")

            raise e

        else:

            return None

def main(phone, tg_id):

```

```

    p_type = phone_type(phone,
tg_id)

    exclamation_mark = '\U00002757'

    if p_type is None:

        return exclamation_mark+"
Вкажіть номер телефону у
міжнародному форматі
"+exclamation_mark

    return p_type

```

## Файл handlers\users\main\full\_name\_bd\_ checker.py

```

from datetime import datetime

from pprint import pprint

from jinja2 import Template

import requests

import os

from data import config

def report_msg(full_name, date,
data):

    data_dict = {

        'title': f"{full_name}
{date}",

        'full_name': full_name,

        'birth_date': date,

        'email': None,

        'addrs': None,

        'phone': None,

        'doc': None,

        'social': None,

    }

    if 'not_found' in data:

        return None

    else:

        if data['email']:

            email = ""

```

```

            email += ''.join('<li>'
+ email + '</li>' for email in
data['email']) if len(

                data['email']) > 0
            else '&#10060; Не знайдено &#10060;'

            data_dict['email'] =
email

        else:

            data_dict['email'] =
'&#10060; Не знайдено &#10060;'

            if data['addr']:

                addr = ""

                addr += ''.join('<li>' +
addr + '</li>' for addr in
data['addr']) if len(

                    data['addr']) > 0
            else '&#10060; Не знайдено &#10060;'

            data_dict['addrs'] =
addr

        else:

            data_dict['addrs'] =
'&#10060; Не знайдено &#10060;'

            if data['phone']:

                phone = ""

                phone += ''.join('<li>'
+ phone + '</li>' for phone in
data['phone']) if len(

                    data['phone']) > 0
            else '&#10060; Не знайдено &#10060;'

            data_dict['phone'] =
phone

        else:

            data_dict['phone'] =
'&#10060; Не знайдено &#10060;'

            if data['doc']:

                pprint(data['doc'])

                doc = ""

                doc += f"<p>Паспорт
громадянина РФ:</p>{''.join(['<li>
&ndash;' + passport + '</li>' for
passport in
data['doc']['passport'])} if
len(data['doc']['passport']) > 0

```

```

else '&#10060; Не найдено
&#10060;}'

        doc +=
f"<br><p>СНИЛС:</p>{''.join(['<li>
&ndash;' + snils + '</li>' for snils
in data['doc']['snils']]) if
len(data['doc']['snils']) > 0 else
'&#10060; Не найдено &#10060;}'"

        doc +=
f"<br><p>ИНН:</p>{''.join(['<li>
&ndash;' + inn + '</li>' for inn in
data['doc']['inn']]) if
len(data['doc']['inn']) > 0 else
'&#10060; Не найдено &#10060;}'"

        data_dict['doc'] = doc

        if data['social']:

            target_blank =
"target=\"_blank\""

            social = ""

            if 'telegram' in
data['social']:

                social +=
f"<i>telegram:</i> {'',
'.join(link['nickname']+
<code>'+link['uid']+</code>' for
link in data['social']['telegram'])}
<br>"

            if 'facebook' in
data['social']:

                slash = '\\'

                social +=
f"<i>facebook:</i> {'', '.join('<a
href='+slash+link['url']+slash+target_
blank+'>' + link['full_name'] +
'</a>' for link in
data['social']['facebook'])}<br>"

            if 'linkedin' in
data['social']:

                slash = '\\'

                social +=
f"<i>linkedin:</i> {'', '.join('<a
href='+slash+link['url']+slash+target_
blank+'>' + link['full_name'] +
'</a>' for link in
data['social']['linkedin'])}<br>"

            if 'mailru' in
data['social']:

                slash = '\\'

                social +=
f"<i>mailru:</i> {'', '.join('<a
href='+slash+link['url']+slash+target_
blank+'>' + link['full_name'] +
'</a>' for link in
data['social']['mailru'])}<br>"

            if 'pikabu' in
data['social']:

                slash = '\\'

                social +=
f"<i>pikabu:</i> {'', '.join('<a
href='+slash+link['url']+slash+target_
blank+'>' + link['full_name'] +
'</a>' for link in
data['social']['pikabu'])}<br>"

            if 'vk' in
data['social']:

                slash = '\\'

                social +=
f"<i>vk:</i> {'', '.join('<a
href='+slash+link['url']+slash+target_
blank+'>' + link['full_name'] +
'</a>' for link in
data['social']['vk'])}<br>"

            if 'instagram' in
data['social']:

                slash = '\\'

                social +=
f"<i>instagram:</i> {'', '.join('<a
href='+slash+link['url']+slash+
target_blank+'>' + link['full_name']
+ '</a>' for link in
data['social']['instagram'])}<br>"

            data_dict['social'] =
social

        else:

            data_dict['social'] =
'&#10060; Не найдено &#10060;'

            if data['transport']:

                transport = ""

                for transp in
data['transport']:

                    transport += '<li>
Номерный знак: ' +
transp['car_plate'] + '</li> <br>' +
'<li> VIN: ' + transp['vin'] +
'</li> <br>' + '<li> Модель Авто: '
+ transp['car_model'] + '</li> <br>'

```

```
+ '<li> Колір: ' +
transp['car_color'] + '</li> <br>' +
'<li> Пік: ' +
str(transp['car_year']) + '</li>
<br><br>'
```

```
data_dict['transport'] =
transport
```

```
else:
```

```
data_dict['transport'] =
'&#10060; Не знайдено &#10060;'
```

```
file_path =
os.path.abspath("data/report/index.h
tml")
```

```
with open(file_path,
encoding='utf-8') as file_:
```

```
template =
Template(file_.read())
```

```
html = template.render(
data_dict
)
```

```
dt = datetime.today()
```

```
report_file_name = full_name
+ " " + date + " " +
str(dt.timestamp())
```

```
with
open(f"data/report/tmp/{report_file
name}.html", 'w', encoding='utf-8')
as f:
```

```
f.write(html)
```

```
file_path =
os.path.abspath(f"data/report/tmp/{r
eport_file_name}.html")
```

```
return file_path
```

```
def main(full_name, date, tg_id):
```

```
query_text = f"{full_name}
{date}"
```

```
try:
```

```
r = requests.get(
```

```
f"{config.API}/info_by_full_name_bir
th_date?full_name={full_name}&birth_
date={date}&tg_id={tg_id}")
```

```
if r.status_code == 200:
```

```
data = r.json()
```

```
result =
report_msg(full_name, date, data)
```

```
requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=full_name
_date&query={query_text}&response={d
ata}&status_code={r.status_code}")
```

```
return result
```

```
raise r.raise_for_status()
```

```
except requests.HTTPError as e:
```

```
requests.post(f"{config.API}/log?que
ry_from={tg_id}&query_type=full_name
_date&query={query_text}&response={e
.response.reason}&status_code={e.res
ponse.status_code}")
```

```
raise e
```

## Файл data\report\index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xht
ml1-strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>{{title}}</title>
```

```
<meta http-equiv="content-type"
content="text/html; charset=utf-8"/>
```

```
<meta name="keywords"
content=""/>
```

```
<meta name="description"
content=""/>
```

```
<link rel="stylesheet"
type="text/css"
```

```
href="http://yui.yahooapis.com/2.7.0
/build/reset-fonts-grids/reset-
fonts-grids.css" media="all"/>
```

```
<style>
    .msg { padding: 10px;
background: #222; position:
relative; }

    .msg h1 { color: #fff; }

    .msg a { margin-left: 20px;
background: #408814; color: white;
padding: 4px 8px; text-decoration:
none; }

    .msg a:hover { background:
#266400; }

    body { font-family: Georgia;
color: #444; }

    #inner { padding: 10px 80px;
margin: 80px auto; background:
#f5f5f5; border: solid #666; border-
width: 8px 0 2px 0; }

    .yui-gf { margin-bottom:
2em; padding-bottom: 2em; border-
bottom: 1px solid #ccc; }

    #hd { margin: 2.5em 0 3em 0;
padding-bottom: 1.5em; border-
bottom: 1px solid #ccc }

    #hd h2 { text-transform:
uppercase; letter-spacing: 2px; }

    #bd, #ft { margin-bottom:
2em; }

    #ft { padding: 1em 0 5em 0;
font-size: 92%; border-top: 1px
solid #ccc; text-align: center; }

    #ft p { margin-bottom: 0;
text-align: center; }

    #hd h1 { font-size: 48px;
text-transform: uppercase; letter-
spacing: 3px; }

    h2 { font-size: 152% }

    h3, h4 { font-size: 122%; }
```

```
h1, h2, h3, h4 { color:
#333; }

    p { font-size: 100%; line-
height: 18px; padding-right: 3em; }

    a { color: #990003 }

    a:hover { text-decoration:
none; }

    strong { font-weight: bold;
}

    /*li { line-height: 24px;
border-bottom: 1px solid #ccc; */

    p.enlarge { font-size: 144%;
padding-right: 6.5em; line-height:
24px; }

    p.enlarge span { color: #000
}

    .contact-info { margin-top:
7px; }

    .first h2 { font-style:
italic; }

    .last { border-bottom: 0 }

    #pdf { display: block;
float: left; background: #666;
color: white; padding: 6px 50px 6px
12px; margin-bottom: 6px; text-
decoration: none; }

    #pdf:hover { background:
#222; }

    .job { position: relative;
margin-bottom: 1em; padding-bottom:
1em; border-bottom: 1px solid #ccc;
}

    .job h4 { position:
absolute; top: 0.35em; right: 0 }

    .job p { margin: 0.75em 0
3em 0; }

    .last { border: none; }

    .skills-list { }

    .skills-list ul { margin: 0;
}
```

```

        .skills-list li { margin:
3px 0; padding: 3px 0; }

        .skills-list li span { font-
size: 152%; display: block; margin-
bottom: -2px; padding: 0 }

        .talent { width: 32%; float:
left }

        .talent h2 { margin-bottom:
6px; }

        #srt-ttab { margin-bottom:
100px; text-align: center; }

        #srt-ttab img.last { margin-
top: 20px }

        .yui-gf .yui-u{width:80.2%;}

        .yui-gf
div.first{width:18.3%;}

        .title-item{display: flex;
align-items: center; margin: auto;
justify-content: space-between;}

</style>
</head>
<body>

<div id="doc2" class="yui-t7">
    <div id="inner">

        <div id="hd">
            <div class="yui-gc">
                <div class="yui-u
first">

<h1>{{full_name}}</h1>

<h2>{{birth_date}}</h2>
                </div>
            </div><!--// .yui-gc -->
        </div><!--// hd -->

        <div id="bd">

```

```

<div id="yui-main">
    <div class="yui-b">
        <div class="yui-
gf">
            <div
class="yui-u first title-item">
                <span>#128236;&nbsp;</span><h
2>Пов'язані email адреси</h2>
            </div>
            <div
class="yui-u">
                <p
class="enlarge">
                    {{email}}
                </p>
            </div>
        </div><!--//
.yui-gf -->
            <div
class="yui-gf">
                <div
class="yui-u first title-item">
                    <span>#127969;&nbsp;</span><h
2>Пов'язані адреси</h2>
                </div>
                <div
class="yui-u">
                    <p
class="enlarge">
                        {{addr}}
                    </p>
                </div>
            </div><!--//
.yui-gf -->
            <div class="yui-
gf">
                <div
class="yui-u first title-item">

```

```

        <span>#128196; &nbsp;</span><h2>Документи</h2>
    </div>
    <div
class="yui-u">
        <p
class="enlarge">
            {{doc}}
        </p>
    </div>
</div><!--//
.yui-gf -->
        <div
class="yui-gf">
            <div
class="yui-u first title-item">
                <span>#128077; &nbsp;</span><h2>Соці
                альні мережі</h2>
            </div>
            <div
class="yui-u">
                <p
class="enlarge">
                    {{social}}
                </p>
            </div>
        </div><!--//
.yui-gf -->
        <div
class="yui-gf">
            <div
class="yui-u">
                <p
class="enlarge">
                    {{phone}}
                </p>
            </div>
        </div><!--//
.yui-gf -->
        <div
class="yui-gf">
            <div
class="yui-u first title-item">
                <span>#128663; &nbsp;</span><h2>Пов'
                язані транспортні засоби</h2>
            </div>
            <div
class="yui-u">
                <div><!--// doc -->
            </div>
        </div>
    </body>
</html>

```

## Серверна частина

**Файл api.php**

```
<?php

use
Illuminate\Support\Facades\Route;

Route::group(['namespace' =>
'Phone', 'middleware' => 'botUsers'],
function () {

    Route::get('/info_by_phone',
'IndexController');

});

Route::group(['namespace' =>
'Email', 'middleware' =>
'botUsers'], function () {

    Route::get('/info_by_email',
'IndexController');

});

Route::group(['namespace' =>
'FullNameBD', 'middleware' =>
'botUsers'], function () {

Route::get('/info_by_full_name_birth
_date', 'IndexController');

});

Route::group(['namespace' => 'Log'],
function () {

    Route::post('/log',
'IndexController');

});
```

**Файл****Middleware\BotUsersMiddleware.php**

```
<?php

namespace App\Http\Middleware;

use App\Models\BotUsers;
```

```
use Closure;
use Illuminate\Http\Request;

class BotUsersMiddleware
{

    /**
     * Handle an incoming request.
     *
     * @param
     \Illuminate\Http\Request $request
     * @param
     \Closure(\Illuminate\Http\Request):
     (\Illuminate\Http\Response|\Illumina
     te\Http\RedirectResponse) $next
     * @return
     \Illuminate\Http\JsonResponse
     */
    public function handle(Request
$request, Closure $next)
    {

        $tg_id = $request->request-
>parameters['tg_id'];

        $user =
BotUsers::where('tg_id', $tg_id);

        if(!$user->first()){

            return response()-
>json(['error' => 'Not
authorized.'], 403);

        }

        return $next($request);

    }
}
```

**Файл Requests\FilterRequest.php**

```
<?php

namespace App\Http\Requests\Post;

use
Illuminate\Foundation\Http\FormReque
st;
```



```

class FilterRequest extends
FormRequest
{
    /**
     * Determine if the user is
authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that
apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'phone_number'=>
'digits_between: 1,12',
            'email'=> 'email',
            'full_name'=> 'string',
            'birth_date'=>'',
        ];
    }
}

```

**Файл  
Controllers\Email\BaseController.p  
hp**

<?php

```

namespace
App\Http\Controllers\Email;

use App\Http\Controllers\Controller;
use App\Services\Email\Service;

class BaseController extends
Controller
{
    public $service;

    public function
__construct(Service $service)
    {
        $this->service = $service;
    }
}

```

**Файл  
Controllers\Email\IndexController.  
php**

<?php

```

namespace
App\Http\Controllers\Email;

use
App\Http\Requests\Post\FilterRequest
;

class IndexController extends
BaseController
{
    public function __invoke
(FilterRequest $request){
        $data = $request-
>validated();
    }
}

```

```

        $res = $this->service-
>getInfoByEmail($data['email']);
        return response()-
>json($res, 200);
    }
}

```

## Файл Services\Email\Service.php

```

<?php

namespace App\Services\Email;

use Illuminate\Support\Facades\Http;

class Service
{
    protected array $phone_numbers =
[];
    protected array $names = [];
    protected array $addresses = [];
    protected array $birth = [];
    protected array $social = [];
    protected array $associated_ip =
[];

    protected function
IsNullOrEmptyString($str)
    {
        return ($str === null ||
trim($str) === '');
    }

    protected function
structData($data)
    {
        $black_list_values = [': ',
'не заполнен', 'не указано'];
        foreach ($data as $item) {

```

```

            if
(array_key_exists('phone_number',
$item) && !$this-
>IsNullOrEmptyString($item['phone_nu
mber'])) {

                if
(!in_array($item['phone_number'],
$this->phone_numbers)) {

                    $this-
>phone_numbers[] =
$item['phone_number'];
                }
            }

            if
(array_key_exists('full_name',
$item) && !$this-
>IsNullOrEmptyString($item['full_nam
e'])) {

                if
(!in_array($item['full_name'],
$this->names)) {

                    $this->names[] =
$item['full_name'];
                }
            }

            if
((array_key_exists('name', $item) &&
!$this-
>IsNullOrEmptyString($item['name']))
&& (array_key_exists('fname', $item)
&& !$this-
>IsNullOrEmptyString($item['fname']
))) {

                $this->names[] =
$item['name'] . ' ' .
$item['fname'];
            }

            if
((array_key_exists('first_name',
$item) && !$this-
>IsNullOrEmptyString($item['first_na
me'])) &&
(array_key_exists('last_name',
$item) && !$this-

```

```

>IsNullOrEmptyString($item['last_name']))) {

        if
(array_key_exists('middle_name',
$item) && !$this-
>IsNullOrEmptyString($item['middle_name'])) {

                if
(!in_array($item['last_name'] . ' '
. $item['first_name'] . ' ' .
$item['middle_name'], $this->names))
{
                        $this-
>names[] = $item['last_name'] . ' '
. $item['first_name'] . ' ' .
$item['middle_name'];
                }

        } elseif
(array_key_exists('patronymic_name',
$item) && !$this-
>IsNullOrEmptyString($item['patronymic_name'])) {

                if
(!in_array($item['last_name'] . ' '
. $item['first_name'] . ' ' .
$item['patronymic_name'], $this-
>names)) {
                        $this-
>names[] = $item['last_name'] . ' '
. $item['first_name'] . ' ' .
$item['patronymic_name'];
                }

        } elseif ($this-
>IsNullOrEmptyString($item['patronymic_name']) && $this-
>IsNullOrEmptyString($item['middle_name'])) {

                if
(!in_array($item['last_name'] . ' '
. $item['first_name'], $this-
>names)) {
                        $this-
>names[] = $item['first_name'] . ' '
. $item['last_name'];
                }

        }

    }

    if
((array_key_exists('user_name',
$item) && !$this-
>IsNullOrEmptyString($item['user_name'])) &&
(array_key_exists('user_lastname',
$item) && !$this-
>IsNullOrEmptyString($item['user_lastname']))) {

        $this->names[] =
$item['user_name'] . ' ' .
$item['user_lastname'];

    }

    if
(array_key_exists('tags', $item)) {

        if
(!in_array($item['tags'], $this-
>names)) {

                $new_tags =
explode(' ', $item['tags']);

                foreach
($new_tags as $i) {

                        $this-
>names[] = $i;
                }

        }

    }

    if
(array_key_exists('name', $item) &&
!$this-
>IsNullOrEmptyString($item['name']))
{

        if
(!in_array($item['name'], $this-
>names)) {

                $this->names[] =
$item['name'];

        }

    }

}

```

```

        if
((array_key_exists('address_city',
$item) &&
!empty($item['address_city'])) &&
(array_key_exists('address_street',
$item) &&
!empty($item['address_street']))) {
            if (0 <
!count(array_intersect(array_map('st
rtolower', explode(' ',
$item['address_city'])),
$black_list_values)) || 0 <
!count(array_intersect(array_map('st
rtolower', explode(' ',
$item['address_street'])),
$black_list_values))) {
                if
(!in_array($item['address_city'] . '
' . $item['address_street'], $this-
>addresses)) {
                    $this-
>addresses[] = $item['address_city']
. ' ' . $item['address_street'];
                }
            }
        }
    }
    if
(array_key_exists('beeline_address_c
ity', $item) &&
!empty($item['beeline_address_city']
)) {
        if
(!in_array($item['beeline_address_ci
ty'] . ' ' .
$item['beeline_address_street'] . '
' . $item['beeline_address_house'] .
' ' . $item['beeline_address_appt']
. ' (lat=' .
$item['beeline_latitude'] . ', lng='
. $item['beeline_longitude'] . ')',
$this->addresses)) {
            $this-
>addresses[] =
$item['beeline_address_city'] . ' '
. $item['beeline_address_street'] .
' ' . $item['beeline_address_house']
. ' ' .
$item['beeline_address_appt'] . '
(lat=' . $item['beeline_latitude'] .
', lng=' .
$item['beeline_longitude'] . ')';
        }
    }
}

        }
    }
    if
(array_key_exists('delivery2_address
_full', $item) &&
!empty($item['delivery2_address_full
'])) {
        if
(!in_array($item['delivery2_address_
full'] . ' ' .
$item['delivery2_address_flat_number
'] . ' (lat=' .
$item['delivery2_latitude'] . ',
lng=' . $item['delivery2_longitude']
. ')', $this->addresses)) {
            $this-
>addresses[] =
$item['delivery2_address_full'] . '
' .
$item['delivery2_address_flat_number
'] . ' (lat=' .
$item['delivery2_latitude'] . ',
lng=' . $item['delivery2_longitude']
. ')';
        }
    }
    if
(array_key_exists('delivery_address'
, $item) &&
!empty($item['delivery_address'])) {
        if
(!in_array($item['delivery_address']
. ' (lat=' . $item['delivery_lat'] .
', lng=' . $item['delivery_long'] .
')', $this->addresses)) {
            $this-
>addresses[] =
$item['delivery_address'] . ' (lat='
. $item['delivery_lat'] . ', lng=' .
$item['delivery_long'] . ')';
        }
    }
    if
(array_key_exists('wildberries_adre
ss', $item) &&
!empty($item['wildberries_address']
)) {
        if
(!in_array($item['wildberries_adre
ss'] . ' (lat=' .
$item['wildberries_lat'] . ', lng='

```



```

];
}
if
(array_key_exists('linkedin_link',
$item)) {
    $this->social[] = [
        'linkedin' =>
'https://linkedin.com' .
$item['linkedin_link']
    ];
}
if
(array_key_exists('mailru_profile',
$item)) {
    $this->social[] = [
        'mailru' =>
$item['mailru_profile']
    ];
}
if
(array_key_exists('pikabu_username',
$item)) {
    $this->social[] = [
        'pikabu' =>
'https://pikabu.ru/@' .
$item['pikabu_username']
    ];
}
if
(array_key_exists('user_vkontakte_ur
l', $item)) {
    $this->social[] = [
        'vk' => ['url'
=> [$item['user_vkontakte_url']]
    ];
}
if
(array_key_exists('user_vk_id',
$item)) {
    $this->social[] = [
        'vk' => ['url'
=> ['https://vk.com/id' .
$item['user_vk_id']]
];
};
}
if
(array_key_exists('uid', $item) &&
array_key_exists('nickname', $item))
{
    $this->social[] = [
        'tg' => [
            'uid' =>
$item['uid'],
            'nickname'
=> $item['nickname'],
            'name' =>
$item['name'],
            'fname' =>
$item['fname'],
        ]
    ];
}
}

public function
getInfoByEmail($email)
{
    $list_of_used_models = [
        \App\Models\AlfaBank::class,
        \App\Models\Cdek::class,
        \App\Models\DobroRu::class,
        \App\Models>EmailsPhones::class,
        \App\Models\Exluxair::class,
        \App\Models\FlUsers::class,
        \App\Models\Fotostana::class,

```

```

\App\Models\GeekBrains::class,
\App\Models\Gemotest::class,
\App\Models\Gosuslugi::class,
    \App\Models\HH::class,
\App\Models\Igzakon::class,
\App\Models\KariSummer2022::class,
\App\Models\Linkedin::class,
\App\Models\MailRu::class,
\App\Models\Miltor::class,
\App\Models\Moscow10k::class,
\App\Models\Mydocuments36::class,
\App\Models\NadpoRu::class,
\App\Models\Oneland::class,
\App\Models\PayadMe::class,
\App\Models\Pikabu::class,
\App\Models\RfCont::class,
\App\Models\TurniketovNet::class,
\App\Models\TutuBus::class,
\App\Models\Vivadengi::class,
\App\Models\Wildberries::class,
\App\Models\YandexPract::class,
\App\Models\RusskoeSlovo::class,
\App\Models\Tele2::class,
\App\Models\DeliveryPartTwo::class,
    ];
    foreach
($list_of_used_models as $model) {
        $info =
$model::where('email', $email)-
>get()->toArray();
        $this-
>structData($info);
    }
        $this->phone_numbers =
array_unique(array_filter($this-
>phone_numbers));
        $this->names =
array_filter(array_unique($this-
>names));
        $this->addresses =
array_unique(array_filter($this-
>addresses));
        $this->social =
array_filter($this->social);
        $this->birth =
array_unique(array_filter($this-
>birth));
        $this->associated_ip =
array_unique(array_filter($this-
>associated_ip));
        if (!$this->names && !$this-
>addresses && !$this->birth &&
!$this->phone_numbers && !$this-
>associated_ip && !$this->social) {
            return [
                'not_found' => null
            ];
        } else
            return [
                'names' => $this-
>names,
                'addresses' =>
$this->addresses,

```

```

        'birth' => $this->birth,
    ];
    'phone_number' => $this->phone_numbers,
    'associated_ip' => $this->associated_ip,
    'social' => $this->social,
    'leaked_passwords' => []
];
}
}

```

### Файл **Controllers\FullNameBD\BaseController.php**

```

<?php

namespace
App\Http\Controllers\FullNameBD;

use App\Http\Controllers\Controller;
use App\Services\FullNameBD\Service;

class BaseController extends Controller
{
    public $service;

    public function __construct(Service $service)
    {
        $this->service = $service;
    }
}

```

### Файл **Controllers\FullNameBD\IndexController.php**

```

<?php

namespace
App\Http\Controllers\FullNameBD;

use
App\Http\Requests\Post\FilterRequest;

class IndexController extends BaseController
{
    public function __invoke(
        FilterRequest $request)
    {
        $data = $request->validated();

        $res = $this->service->getInfoByFullNameDB($data['full_name'], $data['birth_date']);

        return response()->json($res, 200);
    }
}

```

### Файл **Services\FullNameBD\Service.php**

```

<?php

namespace App\Services\FullNameBD;

class Service
{
    protected array $emails = [];
    protected array $phone_numbers = [];
    protected array $addresses = [];
    protected array $documents = [

```



```

'passport' => [],
'snils' => [],
'inn' => []
];
protected array $transport = [];
protected array $social = [];
protected array $add_info = [];

protected function
IsNullOrEmptyString($str)
{
    return ($str === null ||
trim($str) === '');
}

protected function
structData($data)
{
    $black_list_values = [': ',
'не заполнен', 'не указано'];

    foreach ($data as $item) {
        if
(array_key_exists('email', $item) &&
!$this->IsNullOrEmptyString($item['email'])
) {
            if
(!in_array($item['email'], $this->emails)) {
                $this->emails[]
= $item['email'];
            }
        }

        if
(array_key_exists('phone_number',
$item) && !$this->IsNullOrEmptyString($item['phone_number'])) {
            if
(!in_array($item['phone_number'],
$this->phone_numbers)) {
                $this->phone_numbers[] =
$item['phone_number'];
            }
        }

        if
((array_key_exists('address_city',
$item) &&
!empty($item['address_city'])) &&
(array_key_exists('address_street',
$item) &&
!empty($item['address_street']))) {
            if (0 <
!count(array_intersect(array_map('strtolower', explode(' ',
$item['address_city'])),
$black_list_values)) || 0 <
!count(array_intersect(array_map('strtolower', explode(' ',
$item['address_street'])),
$black_list_values))) {
                if
(!in_array($item['address_city'] . '
' . $item['address_street'], $this->addresses)) {
                    $this->addresses[] = $item['address_city']
. ' ' . $item['address_street'];
                }
            }

            if
(array_key_exists('beeline_address_c
ity', $item) &&
!empty($item['beeline_address_c
ity'])) {
                if
(!in_array($item['beeline_address_c
ity'] . ' ' .
$item['beeline_address_street'] . '
' . $item['beeline_address_house'] .
' ' . $item['beeline_address_appt']
. ' (lat=' .
$item['beeline_latitude'] . ', lng='
. $item['beeline_longitude'] . ')',
$this->addresses)) {

```



```

        if
        ((array_key_exists('region', $item)
        && !empty($item['region'])) &&
        (array_key_exists('city', $item) &&
        !empty($item['city']))) {

            if
            (!in_array($item['region'] . ' ' .
            $item['city'], $this->addresses)) {

                $this-
                >addresses[] = $item['region'] . ' '
                . $item['city'];
            }
        }

        if
        (array_key_exists('region', $item)
        && !empty($item['region'])) {

            if
            (!in_array($item['region'], $this-
            >addresses)) {

                $this-
                >addresses[] = $item['region'];
            }
        }

        if
        (array_key_exists('city', $item) &&
        !empty($item['city'])) {

            if
            (!in_array($item['city'], $this-
            >addresses)) {

                $this-
                >addresses[] = $item['city'];
            }
        }

        if
        (array_key_exists('car_plate_number'
        , $item) && !$this-
        >IsNullOrEmptyString($item['car_plat
        e_number'])) {

            $this->transport[] =
            [

                'car_plate'

=> !$this-
>IsNullOrEmptyString($item['car_plat
e_number']) ?
$item['car_plate_number'] : '',

                'vin' =>
                !$this-
                >IsNullOrEmptyString($item['car_vin'
                ]) ? $item['car_vin'] : '',

                'car_model'
                => !$this-
                >IsNullOrEmptyString($item['car_mode
                l']) ? $item['car_model'] : '',

                'car_color'
                => !$this-
                >IsNullOrEmptyString($item['car_colo
                r']) ? $item['car_color'] : '',

                'car_year'
                => !$this-
                >IsNullOrEmptyString($item['car_year
                ']) ? $item['car_year'] : '',

            ];
        }

        if
        ((array_key_exists('passport_number'
        , $item) &&
        !empty($item['passport_number'])) &&
        (array_key_exists('passport_issued_b
        y', $item) &&
        !empty($item['passport_issued_by'])))
        {

            if
            (!in_array(implode(" ",
            array_filter([$item['passport_number
            '], $item['passport_issued_by']]),
            $this->documents['passport']))) {

                $this-
                >documents['passport'][] =
                $item['passport_number'] . ' ' .
                $item['passport_issued_by'];
            }
        }

        if
        (array_key_exists('passport', $item)
        && !empty($item['passport'])) {

            if
            (!in_array($item['passport'], $this-
            >documents['passport'])) {

                $this-
                >documents['passport'][] =
                $item['passport'];
            }
        }

```

```

        }
        if
(array_key_exists('snils', $item) &&
!empty($item['snils'])) {
            if
(!in_array($item['snils'], $this-
>documents['snils'])) {
                $this-
>documents['snils'][] =
$item['snils'];
            }
        }
        if
(array_key_exists('inn', $item) &&
!empty($item['inn'])) {
            if
(!in_array($item['inn'], $this-
>documents['inn'])) {
                $this-
>documents['inn'][] = $item['inn'];
            }
        }
        if
(array_key_exists('work', $item)) {
            $this-
>add_info['work'][] = $item['work'];
        }
        if
(array_key_exists('fb_profile_id',
$item)) {
            $this-
>social['facebook'][] = [
                'full_name' =>
$item['full_name'],
                'url' =>
'https://www.facebook.com/profile.ph
p?id=' . $item['fb_profile_id']
            ];
        }
    };
}
else {
    $this-
>social['mailru'][] = [
        'full_name' => $item['full_name'],
        'url' => $item['mailru_profile']
    ];
}
}
else {
    $this-
>social['mailru'][] = [

```

```

        'full_name'
=> $item['full_name'],

        'url' =>
$item['mailru_profile']

    ];

}

}

if
(array_key_exists('pikabu_username',
$item)) {

    $this-
>social['pikabu'][] = [

        'full_name' =>
$item['pikabu_username'],

        'url' =>
'https://pikabu.ru/@' .
$item['pikabu_username']

    ];

}

if
(array_key_exists('vk_id', $item)) {

    $this-
>social['vk'][] = [

        'full_name' =>
$item['first_name'] . ' ' .
$item['last_name'],

        'url' =>
$item['url']

    ];

}

if
(array_key_exists('vk_id', $item) &&
(array_key_exists('instagram',
$item) && !$this-
>IsNullOrEmptyString($item['instagra
m']))) {

        $this-
>social['instagram'][] = [

            'full_name' =>
!$this-
>IsNullOrEmptyString($item['instagra
m']) ? '@' . $item['instagram'] :
null,

            'url' =>
'https://instagram.com/' .
$item['instagram']

        ];

    }

    if
(array_key_exists('user_vkontakte_ur
l', $item) && !$this-
>IsNullOrEmptyString($item['user_vko
ntakte_url'])) {

        $this-
>social['vk'][] = [

            'full_name' =>
$item['name'],

            'url' =>
$item['user_vkontakte_url']

        ];

    }

    if
(array_key_exists('user_vk_id',
$item) && $item['user_vk_id'] != 0)
{

        $this-
>social['vk'][] = [

            'full_name' =>
$item['user_name'] . ' ' .
$item['user_lastname'],

            'url' =>
'https://vk.com/id' .
$item['user_vk_id']

        ];

    }
}

```

```

        }
        $this->social[] = [
            'associated_ip'
            => $item['customerIP']
        ];
    }
    if
    (array_key_exists('uid', $item) &&
    array_key_exists('nickname', $item))
    {
        $this->social['telegram'][] = [
            'uid' => '#' .
            $item['uid'],
            'nickname' =>
            !$this->
            >IsNullOrEmptyString($item['nickname
            ']) ? '@' . $item['nickname'] : '',
            'full_name' =>
            !$this->
            >IsNullOrEmptyString(implode(" ",
            array_filter([$item['name'],
            $item['fname']])))) ? implode(" ",
            array_filter([$item['name'],
            $item['fname']])) : null
        ];
    }
    if
    (array_key_exists('delivery2_ip',
    $item)) {
        $this->social[] = [
            'associated_ip'
            => $item['delivery2_ip']
        ];
    }
    if
    (array_key_exists('user_ip', $item))
    {
        $this->social[] = [
            'associated_ip'
            => $item['user_ip']
        ];
    }
    if
    (array_key_exists('customerIP',
    $item)) {
        $this->social[] = [
            'associated_ip'
            => $item['lastip']
        ];
    }
    if
    (array_key_exists('ip', $item)) {
        $this->social[] = [
            'associated_ip'
            => $item['ip']
        ];
    }
}

protected function
searchInfoByPhone($data)
{
    $list_of_used_models = [
        \App\Models\Beeline::class,
        \App\Models\Facebook::class,
        \App\Models\DeliveryPartTwo::class,
        \App\Models\DeliveryPartOne::class,
        \App\Models\HH::class,
        \App\Models\Linkedin::class,
        \App\Models\MailRu::class,
    ];
}

```

```

\App\Models\Miltor::class,
\App\Models\Okrug::class,
\App\Models\Sushi::class,
\App\Models\Pikabu::class,
\App\Models\Wildberries::class,
    \App\Models\Vtb::class,
\App\Models\RfCont::class,
\App\Models\EmailsPhones::class,
\App\Models\FlUsers::class,
\App\Models\AlfaBank::class,
\App\Models\Exluxair::class,
\App\Models\GeekBrains::class,
\App\Models\EyeOfGod::class,
\App\Models\Vivadengi::class,
\App\Models\TurniketovNet::class,
\App\Models\TutuBus::class,
    \App\Models\Cdek::class,
\App\Models\Telegram::class,
    \App\Models\Vk::class,
\App\Models\Yandex::class,
\App\Models\LentaShop::class,
\App\Models\NadpoRu::class,
\App\Models\Pirogidomoy::class,
    \App\Models\YandexPract::class,
    \App\Models\Fotostana::class,
    \App\Models\RSushi::class,
    \App\Models\Avito::class,
    \App\Models\DobroRu::class,
    \App\Models\Gemotest::class,
        \App\Models\Kari::class,
    \App\Models\KariSummer2022::class,
    \App\Models\PayadMe::class,
    \App\Models\Roll46::class,
    \App\Models\Tele2::class,
    \App\Models\Tochkaedy::class,
    \App\Models\Gibdd2::class,
        ];
        foreach ($data as $item) {
            foreach
                ($list_of_used_models as $model) {
                    $info =
                        $model::where('phone_number',
                            $item)->get()->toArray();
                    $this-
                        >structData($info);
                }
            }
        }
        protected function
        searchInfoByEmail($data)
        {
            $list_of_used_models = [

```

```

\App\Models\AlfaBank::class,
    \App\Models\Cdek::class,

\App\Models\DeliveryPartTwo::class,

\App\Models\EmailsPhones::class,

\App\Models\Exluxair::class,

\App\Models\FlUsers::class,
\App\Models\Fotostana::class,
\App\Models\GeekBrains::class,
\App\Models\Linkedin::class,
\App\Models\MailRu::class,
\App\Models\Miltor::class,
\App\Models\NadpoRu::class,
\App\Models\PayadMe::class,
\App\Models\Pikabu::class,
\App\Models\RfCont::class,
\App\Models\TutuBus::class,
\App\Models\Wildberries::class,
\App\Models\YandexPract::class,
\App\Models\Tele2::class,

    ];

    foreach ($data as $item) {

        foreach
($list_of_used_models as $model) {

            $info =
$model::where('email', $item)-
>get()->toArray();

            $this-
>structData($info);

        }

    }

    protected function
searchInfoByFullName($full_name,
$birth_date)
    {

        $list_of_used_models = [

\App\Models\Gosuslugi::class,
    \App\Models\HH::class,

\App\Models\Moscow10k::class,

\App\Models\TurniketovNet::class,

\App\Models\Vivadengi::class,

        ];

        foreach
($list_of_used_models as $model) {

            $info = $model::where([

                'full_name' =>
                $full_name,

                'birth_date' =>
                $birth_date,

            ])->get()->toArray();

            $this-
            >structData($info);

        }

    }

    protected function
searchInfoByLastName($last_name,

```



```

$first_name, $middle_name,
$birth_date)

    {
        $list_of_used_models = [
\
App\Models\KariSummer2022::class,
\
App\Models\DobroRu::class,
\
App\Models\Gemotest::class,
];

        foreach
($list_of_used_models as $model) {
            $info = $model::where([
                'last_name' =>
$last_name,
                'first_name' =>
$first_name,
                'middle_name' =>
$middle_name,
                'birth_date' =>
$birth_date,
            ]->get()->toArray());

            $this->structData($info);
        }

        public function
getInfoByFullNameDB($full_name,
$birth_date)
    {
        $full_name_expload =
explode(' ', $full_name);

        $last_name =
$full_name_expload[0];

        $first_name =
$full_name_expload[1];

        $middle_name =
$full_name_expload[2];

        $date =
strtotime($birth_date);

        $possible_date_1 =
date('d/m/Y', $date);

        $possible_date_2 = date('Y-m-
d', $date);

        $this->searchInfoByFullName($full_name,
$birth_date);

        $this->searchInfoByLastName($last_name,
$first_name, $middle_name,
$possible_date_2);

        $this->emails =
array_unique(array_filter($this->
emails));

        $this->phone_numbers =
array_unique(array_filter($this->
phone_numbers));

        $this->addresses =
array_unique(array_filter($this->
addresses));

        $this->add_info =
array_unique(array_filter($this->
add_info));

        $this->searchInfoByPhone($this->
phone_numbers);

        $this->searchInfoByEmail($this->
emails);

        $this->emails =
array_unique(array_filter($this->
emails));

        $this->phone_numbers =
array_unique(array_filter($this->
phone_numbers));

        $this->addresses =
array_unique(array_filter($this->
addresses));

```

```

        $this->add_info =
array_unique(array_filter($this-
>add_info));

        if (empty($this->emails) &&
empty($this->phone_numbers) &&
empty($this->addresses) &&
empty($this-
>documents['passport'])&&
empty($this->documents['inn'])&&
empty($this->documents['snils']) &&
empty($this->social) && empty($this-
>transport)) {

            return [

                'not_found' => null

            ];

        }

        else {

            return [

                'email' => $this-
>emails,

                'phone' => $this-
>phone_numbers,

                'addr' => $this-
>addresses,

                'doc' => $this-
>documents,

                'social' => $this-
>social,

                'transport' =>
$this->transport,

            ];

        }

    }

}

```

**Файл  
Controllers\Phone\BaseController.p  
hp**

```
<?php
```

```

namespace
App\Http\Controllers\Phone;

use App\Http\Controllers\Controller;
use App\Services\Phone\Service;

class BaseController extends
Controller

{

    public $service;

    public function
__construct(Service $service)

    {

        $this->service = $service;

    }

}

```

**Файл  
Controllers\Phone\IndexController.  
php**

```
<?php
```

```

namespace
App\Http\Controllers\Phone;

use
App\Http\Requests\Post\FILTERRequest
;

class IndexController extends
BaseController

{

    public function __invoke
(FILTERRequest $request){

        $data = $request-
>validated();

        $res = $this->service-
>getInfoByPhone($data['phone_number'
]);

```

```

        return response()-
>json($res, 200);

    }
}

```

## Файл Services\Phone\Service.php

```
<?php
```

```
namespace App\Services\Phone;
```

```
class Service
```

```
{
    protected array $emails = [];
    protected array $names = [];
    protected array $addresses = [];
    protected array $birth = [];
    protected array $social = [];
    protected array $transport = [];
    protected array $associated_ip =
[];
```

```

    protected function
IsNullOrEmptyString($str)
    {
        return ($str === null ||
trim($str) === '');
    }

```

```

    protected function
structData($data)
    {
        $black_list_values = [': ',
'не заполнен', 'не указано'];

        foreach ($data as $item) {

            if
(array_key_exists('full_name',
$item) && !$this-

```

```

>IsNullOrEmptyString($item['full_nam
e'])) {

```

```

                if
(!in_array($item['full_name'],
$this->names)) {

                    $this->names[] =
$item['full_name'];
                }
            }

```

```

                if
((array_key_exists('name', $item) &&
!$this-
>IsNullOrEmptyString($item['name']))
&& (array_key_exists('fname', $item)
&& !$this-
>IsNullOrEmptyString($item['fname']
)) {

```

```

                    if
(!in_array($item['name'] . ' ' .
$item['fname'], $this->names)) {

                        $this->names[] =
$item['name'] . ' ' .
$item['fname'];
                    }
                }

```

```

                if
((array_key_exists('first_name',
$item) && !$this-
>IsNullOrEmptyString($item['first_na
me'])) &&
(array_key_exists('last_name',
$item) && !$this-
>IsNullOrEmptyString($item['last_nam
e']))) {

```

```

                    if
(array_key_exists('middle_name',
$item) && !$this-
>IsNullOrEmptyString($item['middle_n
ame'])) {

```

```

                        if
(!in_array($item['last_name'] . ' '
. $item['first_name'] . ' ' .
$item['middle_name'], $this->names))
{

                            $this-
>names[] = $item['last_name'] . ' '

```

```

        . $item['first_name'] . ' ' .
        $item['middle_name'];
    }

    } else {
        if
        (!in_array($item['last_name'] . ' '
        . $item['first_name'], $this-
        >names)) {
            $this-
            >names[] = $item['first_name'] . ' '
            . $item['last_name'];
        }
    }

    if
    ((array_key_exists('user_name',
    $item) && !$this-
    >IsNullOrEmptyString($item['user_nam
    e'])) &&
    (array_key_exists('user_lastname',
    $item) && !$this-
    >IsNullOrEmptyString($item['user_las
    tname']))) {
        if
        (!in_array($item['user_name'] . ' '
        . $item['user_lastname'], $this-
        >names)) {
            $this->names[] =
            $item['user_name'] . ' ' .
            $item['user_lastname'];
        }
    }

    if
    (array_key_exists('tags', $item)) {
        $new_tags =
        explode(', ', $item['tags']);
        foreach
        ($new_tags as $i) {
            if
            (!in_array($i, $this->names)) {
                $this-
                >names[] = $i;
            }
        }
    }

    if
    (array_key_exists('name', $item) &&
    !$this-
    >IsNullOrEmptyString($item['name'])) {
        if
        (!in_array($item['name'], $this-
        >names)) {
            $this->names[] =
            $item['name'];
        }
    }

    if
    ((array_key_exists('address_city',
    $item) &&
    !empty($item['address_city'])) &&
    (array_key_exists('address_street',
    $item) &&
    !empty($item['address_street']))) {
        if (0 <
        !count(array_intersect(array_map('st
        rtolower', explode(' ',
        $item['address_city'])),
        $black_list_values)) || 0 <
        !count(array_intersect(array_map('st
        rtolower', explode(' ',
        $item['address_street'])),
        $black_list_values))) {
            if
            (!in_array(implode(" ",
            array_filter([$item['address_city'],
            $item['address_street']])), $this-
            >addresses)) {
                $this-
                >addresses[] = implode(" ",
                array_filter([$item['address_city'],
                $item['address_street']]));
            }
        }
    }

```

```

    }
    if
(array_key_exists('beeline_address_c
ity', $item) &&
!empty($item['beeline_address_city']
)) {
        if
(!in_array(implode(" ",
array_filter([$item['beeline_address
_city'],
$item['beeline_address_street'],
$item['beeline_address_house'],
$item['beeline_address_appt']])) . '
(lat=' . $item['beeline_latitude'] .
', lng=' .
$item['beeline_longitude'] . ')',
$this->addresses))
        {
            $this-
>addresses[] = implode(" ",
array_filter([$item['beeline_address
_city'],
$item['beeline_address_street'],
$item['beeline_address_house'],
$item['beeline_address_appt']])) . '
(lat=' . $item['beeline_latitude'] .
', lng=' .
$item['beeline_longitude'] . ')';
        }
    }
    if
(array_key_exists('delivery2_address
_full', $item) &&
!empty($item['delivery2_address_full
'])) {
        if
(!in_array(implode(" ",
array_filter([$item['delivery2_adre
ss_full'],
$item['delivery2_address_flat_number
']])) . ' (lat=' .
$item['delivery2_latitude'] . ',
lng=' . $item['delivery2_longitude']
. ')', $this->addresses)) {
            $this-
>addresses[] = implode(" ",
array_filter([$item['delivery2_adre
ss_full'],
$item['delivery2_address_flat_number
']])) . ' (lat=' .
$item['delivery2_latitude'] . ',

```

```

lng=' . $item['delivery2_longitude']
. ')';
        }
    }
    if
(array_key_exists('delivery_address'
, $item) &&
!empty($item['delivery_address'])) {
        if
(!in_array($item['delivery_address']
. ' (lat=' . $item['delivery_lat'] .
', lng=' . $item['delivery_long'] .
) ', $this->addresses)) {
            $this-
>addresses[] =
$item['delivery_address'] . ' (lat='
. $item['delivery_lat'] . ', lng=' .
$item['delivery_long'] . ')';
        }
    }
    if
(array_key_exists('wildberries_adre
ss', $item) &&
!empty($item['wildberries_address']
)) {
        if
(!in_array($item['wildberries_adres
s'] . ' (lat=' .
$item['wildberries_lat'] . ', lng='
. $item['wildberries_lon'] . ')',
$this->addresses)) {
            $this-
>addresses[] =
$item['wildberries_address'] . '
(lat=' . $item['wildberries_lat'] .
', lng=' . $item['wildberries_lon']
. ')';
        }
    }
    if
(array_key_exists('address', $item)
&& !empty($item['address'])) {
        if
(!in_array($item['address'], $this-
>addresses)) {
            $this-
>addresses[] = $item['address'];
        }
    }

```

```

        }
        if
        ((array_key_exists('actual_region',
        $item) &&
        !empty($item['actual_region'])) &&
        (array_key_exists('actual_address',
        $item) &&
        !empty($item['actual_address']))) {
            if
            (!in_array($item['actual_region'] .
            ' ' . $item['actual_address'],
            $this->addresses)) {
                $this-
                >addresses[] =
                $item['actual_region'] . ' ' .
                $item['actual_address'];
            }
        }
        if
        ((array_key_exists('region', $item)
        && !empty($item['region'])) &&
        (array_key_exists('city', $item) &&
        !empty($item['city']))) {
            if
            (!in_array($item['region'] . ' ' .
            $item['city'], $this->addresses)) {
                $this-
                >addresses[] = $item['region'] . ' ' .
                $item['city'];
            }
        }
        if
        (array_key_exists('region', $item)
        && !empty($item['region'])) {
            if
            (!in_array($item['region'], $this-
            >addresses)) {
                $this-
                >addresses[] = $item['region'];
            }
        }
        if
        (array_key_exists('birth_date',
        $item) &&
        !is_null($item['birth_date'])) {
            if
            (!in_array($item['birth_date'],
            $this->birth)) {
                $this->birth[] =
                $item['birth_date'];
            }
        }
        if
        (array_key_exists('email', $item) &&
        !$this-
        >IsNullOrEmptyString($item['email'])) {
            if
            (!in_array($item['email'], $this-
            >emails)) {
                $this->emails[]
                = $item['email'];
            }
        }
        if
        (array_key_exists('ip', $item) &&
        !$this-
        >IsNullOrEmptyString($item['ip'])) {
            if
            (!in_array($item['ip'], $this-
            >associated_ip)) {
                $this-
                >associated_ip[] = $item['ip'];
            }
        }
        if
        (array_key_exists('fb_profile_id',
        $item)) {
            $this-
            >social['facebook'] = [
                [
                    'full_name'
                    => $item['full_name'],
                    'url' =>
                    'https://www.facebook.com/profile.ph
                    p?id=' . $item['fb_profile_id']
                ]
            ]
        }
    }
}

```

```

        ];
    }
    if
    (array_key_exists('linkedin_link',
    $item)) {
        $this->social['linkedin'] = [
            [
                'full_name'
=> $item['name'],
                'url' =>
                'https://linkedin.com' .
                $item['linkedin_link']
            ]
        ];
    }
    if
    (array_key_exists('mailru_profile',
    $item)) {
        $this->social['mailru'] = [
            [
                'full_name'
=> $item['full_name'],
                'url' =>
                $item['mailru_profile']
            ]
        ];
    }
    if
    (array_key_exists('pikabu_username',
    $item)) {
        $this->social['pikabu'] = [
            [
                'full_name'
=> $item['pikabu_username'],
                'url' =>
                'https://pikabu.ru/@' .
                $item['pikabu_username']
            ]
        ];
    }
    }
    if
    (array_key_exists('vk_id', $item)) {
        $this->social['vk'] = [
            [
                'full_name'
=> $item['first_name'] . ' ' .
                $item['last_name'],
                'url' =>
                $item['url']
            ]
        ];
    }
    if
    (array_key_exists('vk_id', $item) &&
    (array_key_exists('instagram',
    $item) && !$this->
    >IsNullOrEmptyString($item['instagra
    m']))) {
        $this->social['instagram'] = [
            [
                'full_name'
=> !$this->
                >IsNullOrEmptyString($item['instagra
                m']) ? '@' . $item['instagram'] :
                null,
                'url' =>
                'https://instagram.com/' .
                $item['instagram']
            ]
        ];
    }
    if
    (array_key_exists('user_vkontakte_ur
    l', $item) && !$this->
    >IsNullOrEmptyString($item['user_vko
    ntakte_url'])) {
        $this->social['vk']
    = [
            [
                'full_name'
=> $item['name'],

```

```

        'url' =>
$item['user_vkontakte_url']
    ]
    ];
    }
    if
(array_key_exists('user_vk_id',
$item) && $item['user_vk_id'] != 0)
{
        $this->social['vk']
= [
        [
                'full_name'
=> $item['user_name'] . ' ' .
$item['user_lastname'],
        'url' =>
'https://vk.com/id' .
$item['user_vk_id']
        ]
    ];
    }
    if
(array_key_exists('uid', $item) &&
array_key_exists('nickname', $item))
{
        $this-
>social['telegram'] = [
        [
                'uid' =>
'#'.$item['uid'],
                'nickname'
=> !$this-
>IsNullOrEmptyString($item['nickname
']) ? '@' . $item['nickname'] : '',
                'full_name'
=> !$this-
>IsNullOrEmptyString(implode(" ",
array_filter([$item['name'],
$item['fname']])) ? implode(" ",
array_filter([$item['name'],
$item['fname']])) : null
        ]
    ];
    }
    if
(array_key_exists('car_plate_number'
, $item) && !$this-
>IsNullOrEmptyString($item['car_plat
e_number'])) {
        if
(!in_array($item['car_plate_number']
, $this->transport)) {
                $this-
>transport[] =
$item['car_plate_number'];
        }
    }
}
}

public function
getInfoByPhone($filter,
$phone_number)
{
        $list_of_used_models = [
\App\Models\Beeline::class,
\App\Models\Facebook::class,
\App\Models\DeliveryPartTwo::class,
\App\Models\DeliveryPartOne::class,
\App\Models\HH::class,
\App\Models\Linkedin::class,
\App\Models\MailRu::class,
\App\Models\Miltor::class,
\App\Models\Okrug::class,
\App\Models\Sushi::class,
\App\Models\Pikabu::class,
\App\Models\Wildberries::class,

```



```

        \App\Models\Vtb::class,
    \App\Models\RfCont::class,
    \App\Models>EmailsPhones::class,
    \App\Models\FIUsers::class,
    \App\Models\AlfaBank::class,
    \App\Models\Gosuslugi::class,
    \App\Models\Exluxair::class,
    \App\Models\GeekBrains::class,
    \App\Models\EyeOfGod::class,
    \App\Models\Vivadengi::class,
    \App\Models\TurniketovNet::class,
    \App\Models\TutuBus::class,
        \App\Models\Cdek::class,
    \App\Models\Telegram::class,
        \App\Models\Vk::class,
    \App\Models\Yandex::class,
    \App\Models\LentaShop::class,
    \App\Models\NadpoRu::class,
    \App\Models\Pirogidomoy::class,
    \App\Models\YandexPract::class,
    \App\Models\Phones::class,
    \App\Models\Fotostana::class,
    \App\Models\RSushi::class,
    \App\Models\Avito::class,
    \App\Models\DobroRu::class,
    \App\Models\Gemotest::class,
        \App\Models\Kari::class,
    \App\Models\KariSummer2022::class,
    \App\Models\PayadMe::class,
    \App\Models\Roll46::class,
    \App\Models\Tele2::class,
    \App\Models\Tochkaedy::class,
    \App\Models\Gibdd2::class,
    \App\Models\RusskoeSlovo::class,
    ];
    foreach
($list_of_used_models as $model) {
        $info =
$model::where('phone_number',
$phone_number)->get()->toArray();
        $this->
>structData($info);
    }
    $this->emails =
array_unique(array_filter($this->
emails));
    $this->names =
array_filter(array_unique($this->
names));
    $this->addresses =
array_unique(array_filter($this->
addresses));
    $this->social =
array_filter($this->social);
    $this->birth =
array_unique(array_filter($this->
birth));

```

```

        $this->associated_ip =
array_unique(array_filter($this-
>associated_ip));

        if (!$this->names && !$this-
>addresses && !$this->birth &&
!$this->emails && !$this-
>associated_ip && !$this->social) {

            return [

                'not_found' => null

            ];

        } else return [

            'names' => count($this-
>names) > 0 ? $this->names : null,

            'addresses' =>
count($this->addresses) > 0 ? $this-
>addresses : null,

            'birth' => count($this-
>birth) > 0 ? $this->birth : null,

            'emails' => count($this-
>emails) > 0 ? $this->emails : null,

            'associated_ip' =>
count($this->associated_ip) > 0 ?
$this->associated_ip : null,

            'social' => count($this-
>social) > 0 ? $this->social : null,

            'transport' =>
count($this->transport) > 0 ? $this-
>transport : null,

        ];

    }

}

```

## Файл

### Requests\Log\StoreRequest.php

```

<?php

namespace App\Http\Requests\Log;

use
Illuminate\Foundation\Http\FormReque
st;

```

```

class StoreRequest extends
FormRequest

{

    /**

        * Determine if the user is
authorized to make this request.

        *

        * @return bool

        */

    public function authorize()

    {

        return true;

    }

    /**

        * Get the validation rules that
apply to the request.

        *

        * @return array

        */

    public function rules()

    {

        return [

            'query_from'=>'string',

            'query_type'=>'string',

            'query'=>'string',

            'response'=>'string',

            'status_code'=>'string',

        ];

    }

}

```

## Файл

### Controllers\Log/IndexController.ph p

```

<?php

namespace App\Http\Controllers\Log;

```

```

use App\Http\Controllers\Controller;

use App\Http\Requests\Log\StoreRequest;

use App\Models\Log;

use Illuminate\Support\Str;

class IndexController extends Controller
{
    public function __invoke(Log $log, StoreRequest $request)
    {
        $data = $request->validated();

        $data['query_id'] = Str::uuid()->toString();

        $log->create($data);
    }
}

```

## Панель адміністрування

### Файл web.php

```

<?php

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Route;

/*
|-----
|-----
|-----
| Web Routes
|-----
|-----
|-----
|
| Here is where you can register web
routes for your application. These

```

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

```

|
*/

```

```

Route::group(['namespace'=> 'Admin', 'prefix'=> 'admin'], function () {
    Route::get('/history', 'IndexController')->name('admin.index');

    Route::get('users', 'UserController')->name('admin.user');

    Route::get('/user/add', 'CreateController')->name('admin.create');

    Route::post('/user/add', 'StoreController')->name('admin.store');

    Route::get('/history/{id}', 'ShowController')->name('admin.show');

    Route::get('/user/{id}/edit', 'EditController')->name('admin.edit');

    Route::patch('/user/{id}', 'UpdateController')->name('admin.update');

    Route::delete('/user/{id}', 'DestroyController')->name('admin.destroy');
});

```

### Файл Controllers/IndexController.php

```

<?php

namespace App\Http\Controllers\Admin;

```

```
use App\Http\Controllers\Controller;
use App\Models\QueryHistory;
```

```
class IndexController extends
Controller
```

```
{
    public function __invoke()
    {
```

```
<?php
```

```
namespace
App\Http\Controllers\Admin\Post;
```

```
use App\Http\Controllers\Controller;
use App\Services\Post\Service;
```

```
class BaseController extends
Controller
```

```
<?php
```

```
namespace
App\Http\Controllers\Admin\Post;
```

```
use
App\Http\Requests\Post\StoreRequest;
```

```
class StoreController extends
BaseController
```

```
{
    public function
__invoke(StoreRequest $request)
```

```
<?php
```

```
namespace
App\Http\Controllers\Admin\Post;
```

```
$queries =
QueryHistory::latest()-
>paginate(10);
```

```
return view('admin.index',
compact('queries'));
```

```
}
}
```

## Файл

### Controllers\StoreController.php

```
{
    public $service;

    public function
__construct(Service $service)
    {
        $this->service = $service;
    }
}
```

## Файл

### Controllers\StoreController.php

```
{
    $data = $request-
>validated();

    $this->service-
>store($data);

    return redirect()-
>route('admin.user');
}
```

## Файл

### Controllers\CreateController.php

```
use App\Http\Controllers\Controller;
```

```

class CreateController extends
Controller
{
    public function __invoke()
    {

<?php
namespace
App\Http\Controllers\Admin\Post;

use App\Models\BotUsers;

class DestroyController extends
BaseController

```

### Файл

### Controllers\ShowController.php

```

<?php

namespace
App\Http\Controllers\Admin\Post;

use App\Models\QueryHistory;

class ShowController extends
BaseController
{
    public function __invoke(
$query_id)
    {
        $query
=QueryHistory::where('query_id',$que
ry_id)->get()->first();

        return view('admin.show',
compact('query'));
    }
}

```

### Файл layouts\admin\balde.php

```

<!DOCTYPE html>
<html lang="en">

```

```

return view('admin.create');
    }
}

```

### Файл

### Controllers\DestroyController.php

```

{
    public function
__invoke(BotUsers $user)
    {
        $user->delete();
        return redirect()-
>route('admin.user');
    }
}
<head>

```

```

    <meta charset="utf-8">

    <meta name="viewport"
content="width=device-width,
initial-scale=1">

    <title>AdminLTE 3 |
Dashboard</title>

    <!-- Google Font: Source Sans
Pro -->

    <link rel="stylesheet"
href="https://fonts.googleapis.com/c
ss?family=Source+Sans+Pro:300,400,40
0i,700&display=fallback">

    <!-- Font Awesome -->

    <link rel="stylesheet"
href="{{asset('plugins/fontawesome-
free/css/all.min.css')}}">

    <!-- Ionicons -->

    <link rel="stylesheet"
href="https://code.ionicframework.co
m/ionicons/2.0.1/css/ionicons.min.cs
s">

    <!-- Tempusdominus Bootstrap 4 -
->

    <link rel="stylesheet"
href="{{asset('plugins/tempusdominus-
bootstrap-4/css/tempusdominus-
bootstrap-4.min.css')}}">

    <!-- iCheck -->

```

```

    <link rel="stylesheet"
href="{{asset('plugins/icheck-
bootstrap/icheck-
bootstrap.min.css')}}">

    <!-- JQVMap -->

    <link rel="stylesheet"
href="{{asset('plugins/jqvmap/jqvmap
.min.css')}}">

    <!-- Theme style -->

    <link rel="stylesheet"
href="{{asset('dist/css/adminlte.min
.css')}}">

    <!-- overlayScrollbars -->

    <link rel="stylesheet"
href="{{asset('plugins/overlayScroll
bars/css/OverlayScrollbars.min.css')
}}">

    <!-- Daterange picker -->

    <link rel="stylesheet"
href="{{asset('plugins/daterangepick
er/daterangepicker.css')}}">

    <!-- summernote -->

    <link rel="stylesheet"
href="{{asset('plugins/summernote/su
mmernote-bs4.min.css')}}">
</head>

<body class="hold-transition
sidebar-mini layout-fixed">

<div class="wrapper">

    <!-- Preloader -->

    <div class="preloader flex-
column justify-content-center align-
items-center">

        </div>

    <!-- Main Sidebar Container -->

    <aside class="main-sidebar
sidebar-dark-primary elevation-4">

        <!-- Brand Logo -->

        <a href="index3.html"
class="brand-link">

            <span class="brand-text
font-weight-light">AdminLTE 3</span>

        </a>

        <!-- Sidebar -->

        <div class="sidebar">

            <!-- Sidebar user panel
(optional) -->

            <div class="user-panel
mt-3 pb-3 mb-3 d-flex">

                <div class="image">

                </div>

                <div class="info">

                    <a href="#"
class="d-block">Admin</a>

                </div>

            </div>

            <!-- Sidebar Menu -->
@include('includes.admin.sidebar')

            <!-- /.sidebar-menu -->

        </div>

        <!-- /.sidebar -->

    </aside>

```

```

    <!-- Content Wrapper. Contains
page content -->
    <div class="content-wrapper">
        <!-- Content Header (Page
header) -->

        <!-- Main content -->
        <section class="content">
            <div class="container-
fluid">
                @yield('content')
            </div><!-- /.container-
fluid -->
        </section>
        <!-- /.content -->
    </div>
    <!-- /.content-wrapper -->
    <footer class="main-footer">
        <strong>Copyright &copy;
2014-2021 <a
href="https://adminlte.io">AdminLTE.
io</a>.</strong>
        All rights reserved.
        <div class="float-right d-
none d-sm-inline-block">
            <b>Version</b> 3.2.0
        </div>
    </footer>

    <!-- Control Sidebar -->
    <aside class="control-sidebar
control-sidebar-dark">
        <!-- Control sidebar content
goes here -->
    </aside>
    <!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->

    <!-- jQuery -->
    <script
src="{{asset('plugins/jquery/jquery.
min.js')}}"></script>
    <!-- jQuery UI 1.11.4 -->
    <script
src="{{asset('plugins/jquery-
ui/jquery-ui.min.js')}}"></script>
    <!-- Resolve conflict in jQuery UI
tooltip with Bootstrap tooltip -->
    <script>
        $.widget.bridge('uibutton',
$.ui.button)
    </script>
    <!-- Bootstrap 4 -->
    <script
src="{{asset('plugins/bootstrap/js/b
ootstrap.bundle.min.js')}}"></script
>
    <!-- ChartJS -->
    <script
src="{{asset('plugins/chart.js/Chart
.min.js')}}"></script>
    <!-- Sparkline -->
    <script
src="{{asset('plugins/sparklines/spa
rkline.js')}}"></script>
    <!-- JQVMap -->
    <script
src="{{asset('plugins/jqvmap/jquery.
vmap.min.js')}}"></script>
    <script
src="{{asset('plugins/jqvmap/maps/jq
uery.vmap.usa.js')}}"></script>
    <!-- jQuery Knob Chart -->
    <script
src="{{asset('plugins/jquery-
knob/jquery.knob.min.js')}}"></scrip
t>
    <!-- daterangepicker -->
    <script
src="{{asset('plugins/moment/moment.
min.js')}}"></script>

```

```

<script
src="{{asset('plugins/daterangepicker/daterangepicker.js')}}"></script>
<!-- Tempusdominus Bootstrap 4 -->
<script
src="{{asset('plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js')}}"></script>
<!-- Summernote -->
<script
src="{{asset('plugins/summernote/summernote-bs4.min.js')}}"></script>
<!-- overlayScrollbars -->
<script
src="{{asset('plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js')}}"></script>
<!-- AdminLTE App -->
<script
src="{{asset('dist/js/adminlte.js')}}"></script>
<!-- AdminLTE for demo purposes -->
<script
src="dist/js/demo.js"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script
src="{{asset('dist/js/pages/dashboard.js')}}"></script>
</body>
</html>

```

## Файл `includes\sidebar.balde.php`

```

<nav class="mt-2">
    <ul class="nav nav-pills nav-sidebar flex-column" data-

```

## Файл `views\index.balde.php`

```

@extends('layouts.admin')

@section('content')
    <style>
        .clickable-row {
            cursor: pointer;
        }
    </style>

```

```

widget="treeview" role="menu" data-accordion="false">
    <!-- Add icons to the links using the .nav-icon class
        with font-awesome or any other icon font library -->
    <li class="nav-header">ПАНЕЛЬ АДМІНІСТРУВАННЯ</li>
    <li class="nav-item">
        <a
            href="{{route('admin.post.index')}}"
            class="nav-link">
                <i class="nav-icon fas fa-align-justify"></i>
            <p>
                Історія запитів
            </p>
        </a>
    </li>
    <li class="nav-item">
        <a
            href="{{route('admin.post.user')}}"
            class="nav-link">
                <i class="fas fa-user-friends nav-icon"></i>
            <p>
                Список користувачів
            </p>
        </a>
    </li>
</ul>
</nav>
</style>
<div>
    <div>
        <table class="table table-hover">
            <thead>
                <tr>

```



```

        <th
scope="col">#</th>
        <th
scope="col">Тип запиту</th>
        <th
scope="col">Запит</th>
        <th
scope="col">Користувач</th>
        <th
scope="col">Статус</th>
        <th
scope="col">Дата</th>
    </tr>
</thead>
<tbody>
    <div><a
href="{{route('admin.post.create')}}
" class="btn btn-primary mb-
3">create</a></div>--}}
    @foreach($queries as
$query)
        <tr
class='clickable-row' data-
href="{{route('admin.post.show',
$query->query_id)}}">
            <th
scope="row">{{$query->id}}</th>
            <td>{{$query->query_type}}</td>
            <td>{{$query->query}}</td>
            <td>{{$query->query_from}}</td>
            <td>{{$query->status_code}}</td>
            <td>{{$query->created_at}}</td>
        </tr>
    @endforeach
</tbody>
</table>
<div>{{$queries-
>withQueryString()->links()}}</div>
</div>
<!-- jQuery -->
<script
src="{{asset('plugins/jquery/jquery.
min.js')}}"></script>
<script>

jQuery(document).ready(function($) {
    $(".clickable-
row").click(function() {
        window.location =
$(this).data("href");
    });
});
</script>
@endsection

```

## Файл views\user.balde.php

```

@extends('layouts.admin')

@section('content')
    <div>
        <div>
            <form
action="{{route('admin.post.store')}}
" method="post">
                @csrf
                <div class="mb-3">
                    <label
for="title" class="form-label mt-
3">tg_id</label>
                    <input
value="{{old('title')}}"
                    type="text"
name="tg_id" class="form-control"
id="tg_id" placeholder="введіть id
користувача від telegram">
                @error('title')
                <p class="text-
danger">{{$message}}</p>
                @enderror
            </div>
            <button
type="submit" class="btn btn-
primary">Додати</button>
        </form>
        <br>
        <table class="table
table-hover">
            <thead>
                <tr>
                    <th
scope="col">#</th>
                    <th
scope="col">tg_id</th>
                    <th
scope="col">Кількість запитів</th>
                    <th
scope="col">Додано</th>
                    <th
scope="col">Дії</th>
                </tr>
            </thead>
            <tbody>
                @foreach($users as
$user)
                    <tr
class='clickable-row' >
                        <th
scope="row">{{$user->id}}</th>
                        <td
>{{$user->tg_id}}</td>
                        <td
>123</td>
                        <td
>{{$user->created_at}}</td>

```

```

                <td ><button
type="button" class="btn btn-primary
mr-3">Редарувати</button> <button
type="button" class="btn btn-
danger">Видалити</button></td>
            </tr>
        @endforeach
    </tbody>
</table>
<div>{{ $users-
>withQueryString() ->links() }}</div>
</div>
<!-- jQuery -->
@endsection

```

### Файл `views\show.balde.php`

```

@extends('layouts.admin')
@section('content')
<div class="content-header">

```

```

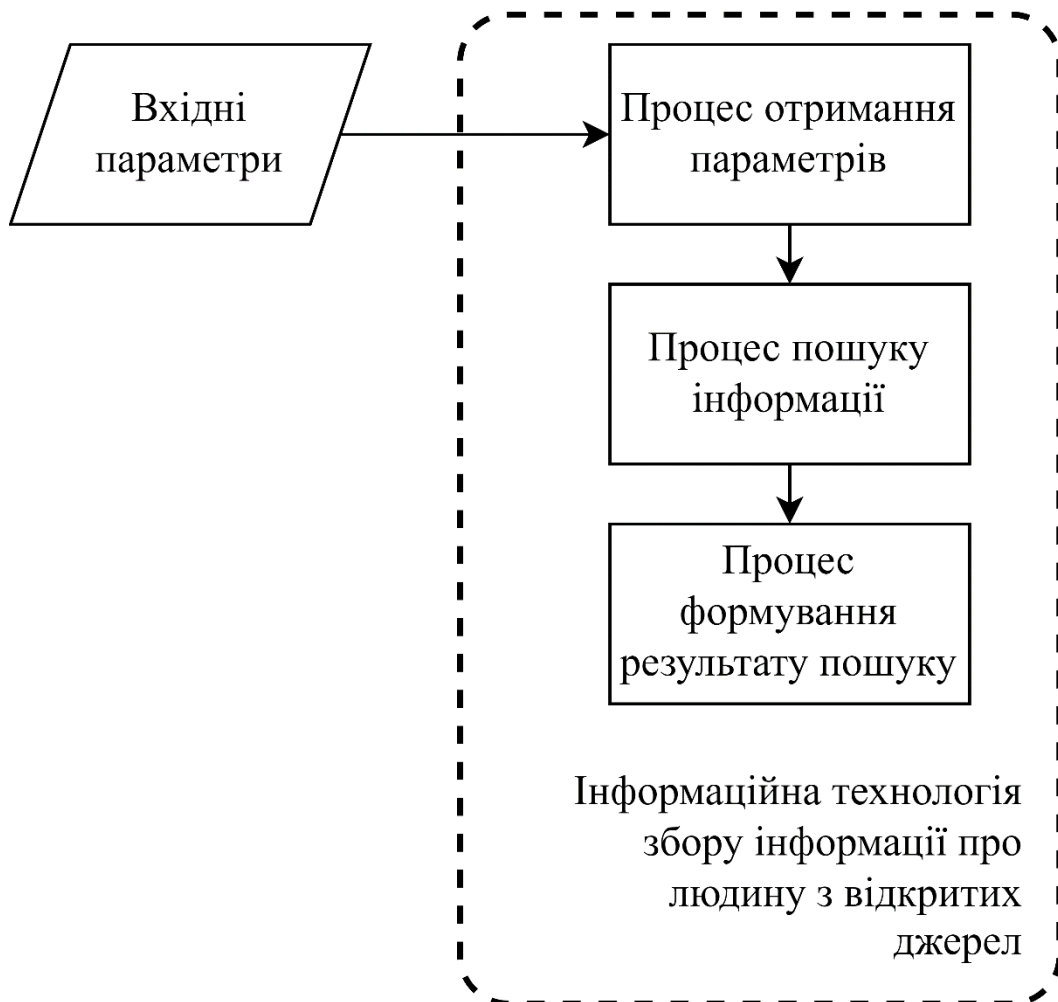
</div>
<!-- /.content-header -->
<div class="col">
    <h1 class=" mb-
5">Результат запиту {{ $query-
>query_id}}</h1>
    <p>#127380; Телеграм id
користувача: {{ $query-
>query_from}}</p>
    <p>#128270; Тип пошуку:
{{ $query->query_type}}</p>
    <p>#10067; Запит: {{ $query-
>query}}</p>
    <p>#127760; Статус:
{{ $query->status_code}}</p>
    <p>#128196; Результат:
{{ $query->response}}</p>
</div>
@endsection

```

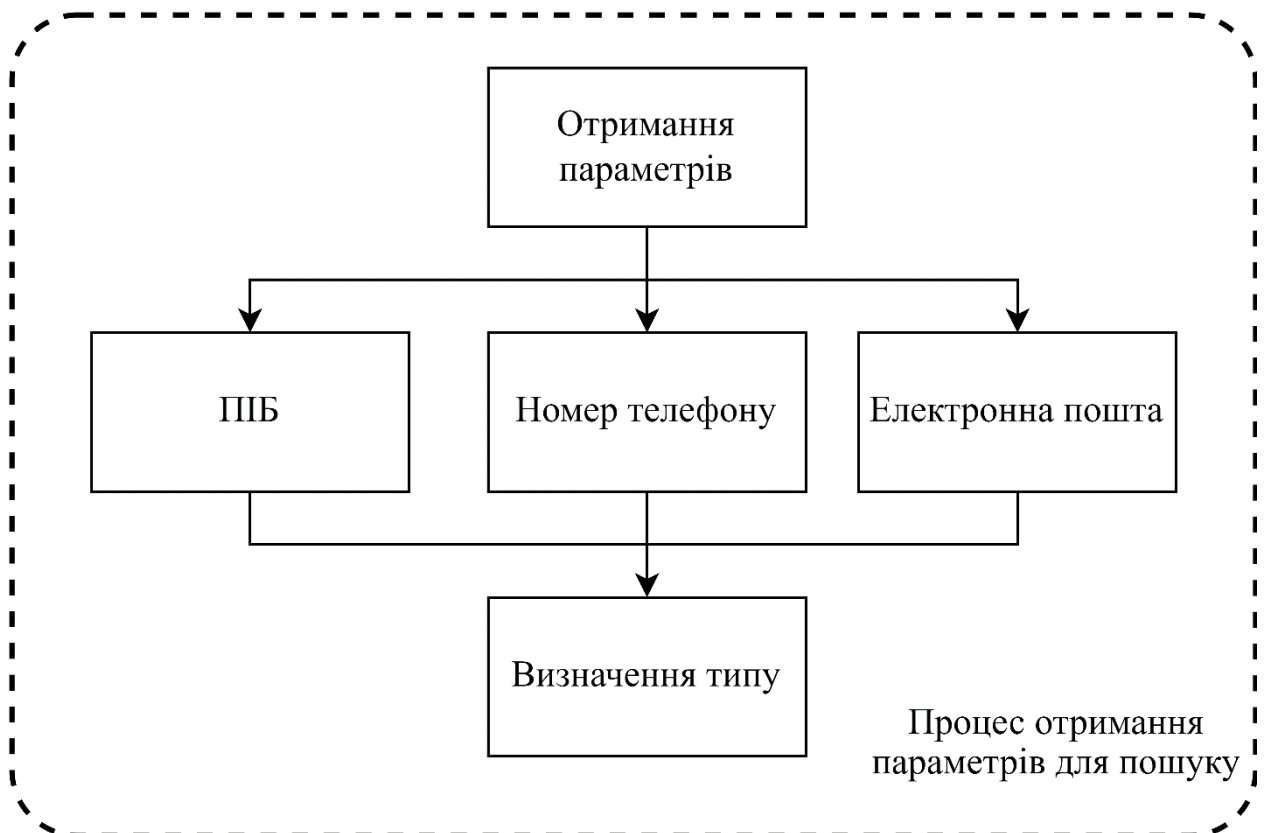
**ІЛЮСТРАТИВНА ЧАСТИНА**

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗБОРУ ІНФОРМАЦІЇ ПРО ЛЮДИНУ З  
ВІДКРИТИХ ДЖЕРЕЛ

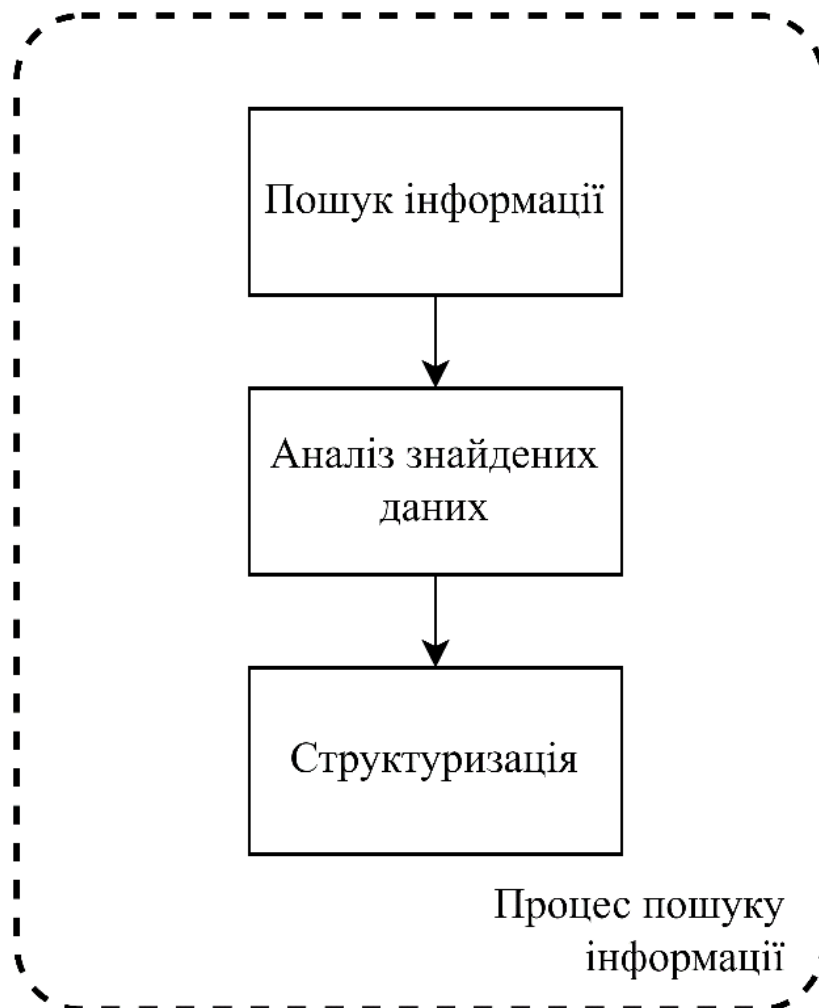
## СТРУКТУРА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ



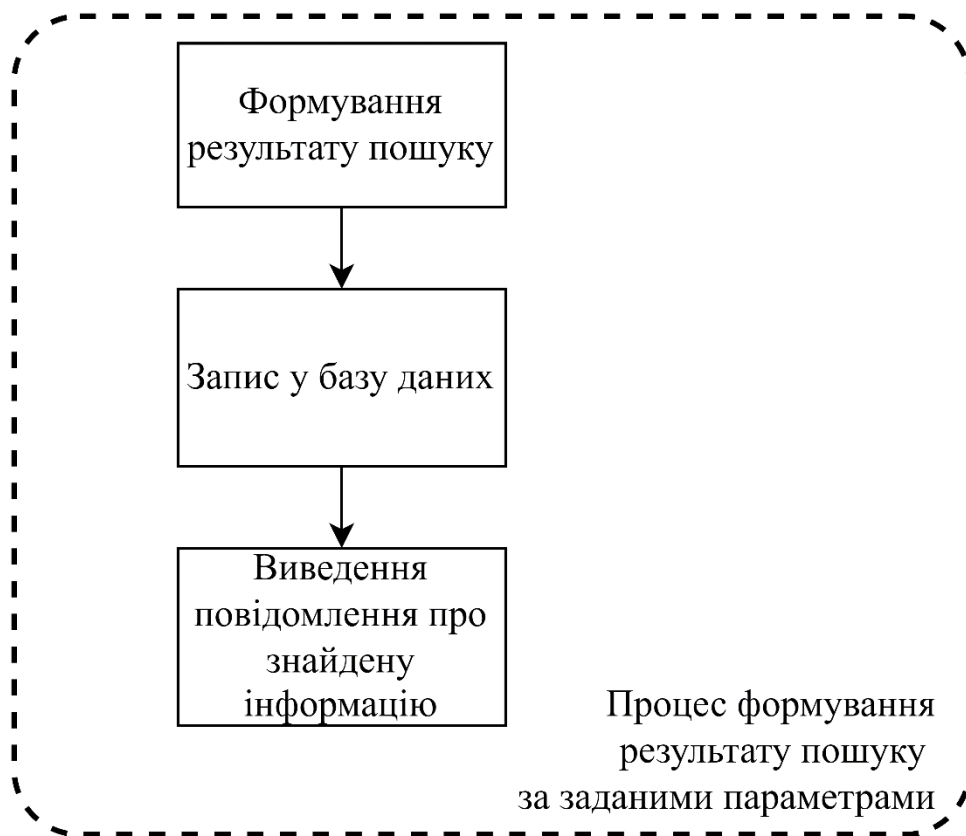
## СХЕМА ПРОЦЕСУ ОТРИМАННЯ ПАРАМЕТРІВ ДЛЯ ПОШУКУ



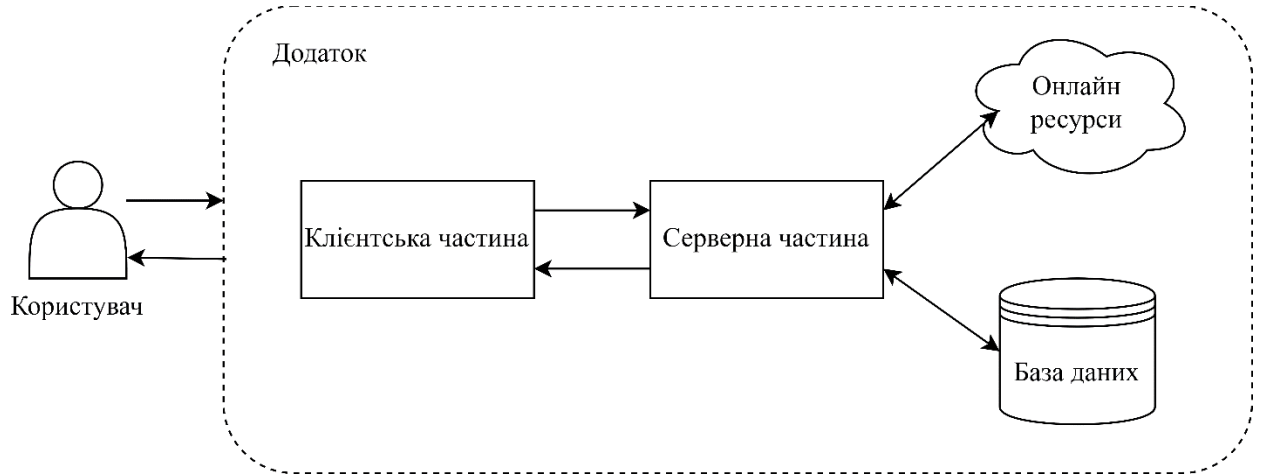
## СХЕМА ПРОЦЕСУ ПОШУКУ ІНФОРМАЦІЇ



## СХЕМА ПРОЦЕСУ ФОРМУВАННЯ РЕЗУЛЬТАТУ ПОШУКУ



# ЗАГАЛЬНА АРХІТЕКТУРА СИСТЕМИ





## ЗАГАЛЬНИЙ АЛГОРИТМ РОБОТИ КЛІЄНТСЬКОЇ ЧАСТИНИ



## ЗАГАЛЬНИЙ АЛГОРИТМ РОБОТИ СЕРВЕРНОЇ ЧАСТИНИ

