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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу»

Виконав: студент 2-го курсу, групи 1БС-21м
спеціальності 125 – Кібербезпека
_____Касьянчук М. Ф.

Керівник: к. т. н., ст. викл. каф. ЗІ
_____ Лукічов В.В.

Опонент: к. т. н., доц., доц. каф . ПЗ
_____Майданюк В. П.

«19» грудня 2022 р.

Допущено до захисту

Завідувач кафедри ЗІ
д. т. н., проф.
_____ Лужецький В. А.
«19» грудня 2022 р.

Вінниця ВНТУ – 2022 рік

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
_____ В. А. Лужецький
«1» вересня 2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Касьянчуку Максиму Федоровичу

1. Тема роботи: «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу»,
керівник роботи: Лукічов Віталій Володимирович, к. т. н., ст. викл. каф. ЗІ,
затверджені наказом ректора ВНТУ від 14 вересня 2022 року №203.
2. Строк подання студентом роботи 19 грудня 2022 р.
3. Вихідні дані до роботи:
   - дані, зібрані з інтернет-форм, фактори ризиків ІБ;
   - спосіб реалізації – веб-ресурс, модулі автентифікації, збирання та аналізування даних, модуль оцінювання ризиків інформаційної безпеки.
4. Зміст текстової частини: Вступ. 1. Аналіз інформаційних джерел. 2. Розробка структури та моделей захищеної системи збирання та аналізування даних. 3. Розробка програмного додатку. 4. Експериментальне дослідження. 5. Економічна частина. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: Порівняльні характеристики програмних аналогів (плакат, А4). Етапи управління ризиками за стандартом ISO 27005 (плакат, А4). Кольорова карта критичності ризику (плакат, А4). Загальний алгоритм роботи підсистеми аналізу (плакат, А4). Форма з внесеними результатами оцінених ризиків(плакат, А4). Повідомлення про результат оцінювання загального стану ІБ системи (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанти	Підпис, дата	
		завдання видав	виконання прийняв
1	Лукічов В.В., к.т.н., ст. викл. каф. ЗІ		
2	Лукічов В.В., к.т.н., ст. викл. каф. ЗІ		
3	Лукічов В.В., к.т.н., ст. викл. каф. ЗІ		
4	Лукічов В.В., к.т.н., ст. викл. каф. ЗІ		
5	Лесько О.Й., к.е.н., проф. каф. ЕПВМ		

7. Дата видачі завдання 1 вересня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2022 – 14.09.2022	
2	Аналіз інформаційних джерел за напрямком магістерської кваліфікаційної роботи	15.09.2022 – 25.09.2022	
3	Розробка рішень	25.09.2022 – 05.10.2022	
4	Практична реалізація, моделювання, експериментування, результати	06.10.2022 – 26.10.2022	
5	Розробка розділу економічного обґрунтування доцільності розробки	26.10.2022 – 18.11.2022	
6	Аналіз виконання ТЗ, висновки	19.11.2022 – 25.11.2022	
7	Оформлення пояснювальної записки	26.11.2022 – 30.11.2022	
8	Попередній захист та доопрацювання МКР	07.12.2022 – 09.12.2022	
9	Представлення МКР до захисту	19.12.2022	
10	Захист МКР	22.12.2022	

Студент _____ М. Ф. Касьянчук

Керівник роботи _____ В. В. Лукічов

АНОТАЦІЯ

УДК 004.056

Касьянчук М. Ф. Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу. Комплексна магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2022. 97 с.

На укр. мові. Бібліогр.: 22 назв; рис.: 34; табл. 22.

Комплексна магістерська кваліфікаційна робота присвячена розробці захищеної системи збирання та аналізування даних для спеціальних задач. Для успішної розробки програмного засобу проведено дослідження наявних аналогів програмних реалізацій систем збирання та аналізування. Під час роботи обґрунтовано вибір власних методів, розроблено ряд схем і алгоритмів, здійснено програмну реалізацію. Засіб перевірено на коректність роботи.

Ілюстративна частина складається з 6 плакатів з демонстрацією схеми алгоритму роботи системи та прикладами її використання.

В економічному розділі оцінено витрати на розробку.

Ключові слова: захищена система, збирання та аналізування даних, Інтернет-форми, інформаційна безпека.

ABSTRACT

Kasyanchuk M. F. Secure data collection and analysis system for special tasks. Part 1. Analysis subsystem. Comprehensive master's qualification work on specialty 125 – Cybersecurity, educational program – Security of information and communication systems. Vinnytsia: VNTU, 2022. 97 p.

In Ukrainian language. Bibliographer: 22 titles; fig.: 34; tabl.: 22.

Comprehensive master's qualification work, dedicated to the development of a protective system of data collection and analysis for special tasks. For the successful development of a software test, a study of available analogues of software implementations, a collection and analysis system, was carried out. During the work, the choice of own methods was substantiated, a number of schemes and algorithms were developed, and indeed software implementation. The tool has been checked for correctness.

The graphic part consists of 6 posters with a demonstration of the scheme of the algorithm of the system and examples of its use.

Development costs are estimated in the economic section.

Keywords: secure system, data collection and analysis, Internet forms, information security.

ЗМІСТ

ВСТУП 7
1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ 9
1.1 Науково-технічне обґрунтування розробки захищеної системи збирання та аналізування даних для спеціальних задач 9
1.2 Аналіз засобів аналогів захищених систем збору та аналізу даних 15
1.3 Постановка завдання 21
2 РОЗРОБКА СТРУКТУРИ ТА МОДЕЛЕЙ ЗАХИЩЕНОЇ СИСТЕМИ ЗБИРАННЯ ТА АНАЛІЗУВАННЯ ДАНИХ 23
2.1 Основні положення стандарту управління ризиками в кібербезпеці 23
2.2 Розробка методу оцінювання ризиків в кібербезпеці 27
2.3 Структура та бізнес процеси підсистеми аналізу 33
3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ 36
3.1 Обґрунтування вибору засобів реалізації програмного засобу 36
3.2 Розробка модуля автентифікації користувача 44
3.3 Розробка модуля збирання даних 47
3.4 Розробка модуля аналізування 49
3.5 Розробка модуля оцінювання ризиків 51
4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ 53
4.1 Тестування роботи програмного засобу 53
4.2 Оцінювання стану інформаційної безпеки системи 63
5 ЕКОНОМІЧНА ЧАСТИНА 66
5.1 Оцінювання наукового ефекту 66
5.2 Розрахунок витрат на здійснення науково-дослідної роботи 70
5.3 Оцінювання важливості та наукової значимості науково-дослідної роботи 83
ВИСНОВКИ 85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 87
ДОДАТКИ 89
Додаток А. Протокол перевірки на наявність плагіату 90
Додаток Б. Код програмного додатку 91

ВСТУП

В сучасному світі системи збирання та аналізування даних використовуються повсякчас. Кожна сучасна компанія незалежно від своїх розмірів, ресурсів та фінансів має справу з такими системами, використовує існуючі або розробляє власну. Такі системи використовуються для збору інформації, що необхідна для компанії та нерідко такі дані є конфіденційними. Вони можуть містити інформацію про клієнтів, замовників, товар та послуги які не повинні бути загальнодоступними та відкритими для загалу. Отже постає питання захищеності та оцінювання загроз інформаційної безпеки таких систем.

Найчастіше такі системи реалізують функціонал для захисту інформації шляхом автентифікації користувача та інших методів захисту. Та для кожної такої системи, в якій є обіг конфіденційних даних необхідно залучати експертів кіберзахисту та проводити аналіз стану безпеки. Для запобігання витоків конфіденційної інформації та персональних даних клієнтів такі системи мають бути оснащені великою кількістю різних засобів для підвищення рівня конфіденційності та інформаційної безпеки. На сьогоднішній день існує багато реалізацій подібних систем, вони використовують ті чи інші засоби для підвищення рівня інформаційної безпеки, але як виявляється на практиці майже всі вони є недосконалими та мають власні недоліки. Отже актуальною є розробка захищеної системи збирання та аналізування даних для спеціальних задач, яка буде імплементувати в собі захист конфіденційних даних своїх користувачів, збирати інформацію та проводити її аналізування.

Об’єктом дослідження є процес захисту систем збирання та аналізування даних.

Предмет дослідження є модулі підсистеми аналізу захищеної системи збирання та аналізування даних для спеціальних задач.

Метою комплексної магістерської кваліфікаційної роботи є покращення безпеки систем збирання та аналізування даних.

Для досягнення мети потрібно виконати наступні завдання:

- виконати аналіз засобів аналогів захищених систем збору та аналізу даних;
- проаналізувати відомі методи реалізації захищених систем;
- розробити моделі системи;
- розробити програмний засіб підсистеми аналізу;
- виконати експериментальне дослідження підсистеми.

Наукова новизна дослідження: розроблено підсистему аналізу захищеної системи збирання та аналізування даних для спеціальних задач з урахуванням міжнародного стандарту ISO/IEC 27005.

Практичне значення розроблюваної системи полягає в захищеному програмному продукті, який надасть можливість збирати та аналізувати дані для спеціальних задач.

Публікації результатів магістерської кваліфікаційної роботи. Результати магістерської роботи доповідалися на таких конференції "Контроль і управління в складних системах (КУСС-2022)" ВНТУ місто Вінниця, 15-17 листопада 2022 р. [1].

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1.1 Науково-технічне обґрунтування розробки захищеної системи збирання та аналізування даних для спеціальних задач

На сьогоднішній день існує безліч систем, які займаються збором та аналізом даних. Усі ці системи різні, працюють у різний спосіб та мають різне призначення. Деякі з них знаходяться у вільному доступі, а деякі є закритими системами з обмеженим колом доступу користувачів. Закриті системи зазвичай захищені від кіберзагроз різних типів тому, що ці системи можуть містити конфіденційну або навіть секретну інформацію компанії чи установи для якої вони розроблялись.

Система збору та аналізу даних (Data collection and analysis system DCAS) – це система реалізована через програмний додаток або веб-ресурс, яка полегшує процес збору даних [2]. Вона дозволяє систематично збирати та обробляти структуровані дані які відповідають заданим параметрам, надає можливість аналізувати зібрані дані та зберегти їх у сховище у ролі якого найчастіше використовують бази даних. Найчастіше DCAS влаштовані як перелік форм з полями вводу відповідних даних від користувача. DCAS мусить перевіряти інформацію введену в форму користувачем, після чого зберігає в постійне сховище. Потім може виконувати певний аналіз даних, формувати статистику, робити підсумки, прогнозувати майбутні показники, тощо.

Як правило, найвживаніші типи для класифікації DCAS мають наступний вигляд:

1) Анкети або опитування.
2) Іспити або вікторини.
3) Реєстри даних.
4) Системи вимірювання продуктивності.
5) Системи управління справами.

6) Системи подання та звітності, інтернет-форми.
7) Системи парсингу.

Варто зазначити, що сучасні системи збору та аналізу даних дуже часто мають властивості кількох вище наведених типів одразу, так як кожна установа, компанія чи державний орган розробляють такі системи під власні потреби. Задовольнити які можна об’єднавши наприклад реєстр даних, інтернет-форми та анкети [3].

Для кращого розуміння методів збору інформації доцільно розглянути детальніше такі засоби як інтернет-форма та системи парсингу. Інтернет-форма – це спеціальна обмежена область на веб-сторінці. В цю область користувач може власноруч ввести дані або вибрати дії із запропонованих. По своїй суті інтернет-форма є аналогом паперової анкети, але покликана спростити збір даних, так як надає можливість автоматизованої обробки. Перевагою засобу є те, що для конкретної цілі збору даних від користувача можна визначити кількість необхідних полів, забезпечити їх коректність введення за допомогою валідації (перевірки коректності введених даних). Недоліком такого засобу є людський фактор, але зменшити цю вразливість допомагає валідація відповідних полів. Системи парсингу – програмне забезпечення, що використовуються для автономного пошуку інформації то заданих критеріях шляхом перебору коду веб-сторінок сайтів та їх текстового наповнення. Перевагою є автономність та гнучкість налаштувань пошуку інформації. Натомість недоліком може стати збір зайвої або некоректної інформації так, як при налаштуваннях парсера користувач може ввести не чіткі налаштування або такі налаштування, що не повністю відповідають цілям збору даних [4].

Одним із важливих аспектів сучасної держави є цифровізація її органів влади. Така цифровізація досягається шляхом розробки програмного забезпечення, що бере на себе частину завдань пов’язану з збиранням зберіганням та обробкою даних для відповідного органу чи установи держави. Також таке програмне забезпечення може втілювати в собі можливість надання

відповідних державних послуг, минаючи величезні бюрократичні ланцюги паперів, працівників та установ, що не аби як уповільнюють процес надання та ефективність таких послуг, іноді навіть до нульового рівня. Також особливої уваги вимагає захист інформації у таких програмних додатках. Адже кожна державна установа має доступ до деяких конфіденційних даних які неповинні потрапити у відкритий доступ.

Захищена система збору та аналізу даних, підсистема аналізу буде реалізована на базі веб-застосунку призначеного для реєстру випадків домашнього насилля, на прохання представників з управління національної поліції вінницької області. Дана система дозволить спростити та пришвидшити надання послуг різних соціальних служб постраждалим особам. Тому що на даному етапі всі анкети і довідки заповнюються в паперовому вигляді, проходять великі ланцюжки бюрократичних операцій, що не аби як уповільнює процес надання допомоги постраждалим. Всі ці недоліки можуть спричинити навіть найгірші наслідки для особи, яка не отримала допомогу вчасно.

Системи збору та аналізу даних реалізовані через веб-застосунки та розміщені в мережі Інтернет часто можуть бути атаковані різними кіберзлочинцями (хакерами) або зловмисним програмним забезпеченням. Для того щоб розуміти які загрози є небезпечними саме для веб-сайтів розглянемо основні типи небезпечного програмного забезпечення та види атак на такі ресурси:

- Denial of Service – відмова в обслуговуванні. Мережева атака, яка перенавантажує запитами сайт таким чином, став недоступний для звичайних користувачів. Метою атаки є вичерпання ресурсів сайту, досягнення стану, коли він не в змозі відповідати на запити клієнтів і змушений «відмовляти в обслуговуванні»;
- DDoS – це випадки множинної Denial of Service атаки, коли сайт атакують запитами не з одного пристрою, а відразу з великої кількості різних пристроїв та IP-адрес. В атаки даного типу більша кількість

пристроїв, а отже більше навантаження на сервер і вищою э ймовірність зробити сайт недоступним швидше. Складність атаки вимірюється в часових межах, чим довша атака, тим вона небезпечніша. Деякі іноземні компанії виділяють основні 5 типів DDoS-атак: HTTP, TCP, ICMP, UDP, інші [5];

– Trojan (троянскі програмні засоби, також відомі як трояни) – різновид шкідливого програмного забезпечення, що завдають шкоди системі шляхом маскування під будь-які корисні додатки. Небезпека даних програм в тому, що вони можуть використовувати як прикриття знайомі користувачеві додатки з якими він міг працювати раніше, ще до появи на персональному комп'ютері «троянського коня». Таке програмне забезпечення може у фоновому режимі зливати персональні дані та конфіденційну інформацію користувача в мережу інтернет або конкретному зловмиснику;
– Worm (мережеві черв'яки) – програмні засоби такого типу здатні до самостійного поширення власних копій серед вузлів локальної мережі, а також за допомогою глобальної мережі здатні до поширення на інші комп'ютери без будь-якої участі користувачів мережі. Так як основним засобом передачі мережевих черв'яків є файли, основним способом їх поширення є мережеві служби, які відповідають за файловий обмін. Таким чином черв'яки можуть розповсюджувати власні копії по мережі у вигляді вкладень електронних листів, або шляхом розміщення гіперпосилань на заражений файл. Однак є й інакші види черв'яків, що для своєї експансії використовують слабкі місця «дірки» у захисті програмного забезпечення;
– Spyware (шпигунські програми) – такий тип шкідливого програмного забезпечення, як правило таємно (віддалено) встановлюється на пристрій жертви яка нічого не підозрює. Щоб таємно відстежувати усі дії користувача, введення паролів та логінів, банківських рахунків та

іншої конфіденційної інформації. Зібрана інформація пересилається зловмиснику, який може використати її на свій розсуд, найчастіше в злочинних цілях;

– спам – тип атаки який використовує як вразливість надмірне навантаження на можливості електронної пошти своєї жертви. З урахуванням важливої ролі яку відіграє електронна пошта в житті сучасних корпорацій, установ та компаній можна зробити висновок, що загроза даного типу є суттєвою небезпекою для ведення бізнесу та надання послуг у сучасному світі [6].

Захист від даних видів атак досягається різними комплексними засобами. Деякі атаки можна зупинити на клієнтській частині, а інші можливо виявити і зупинити лише на серверній частині веб-системи. Та найкращим варіантом було б попередити виникнення таких загроз шляхом їх завчасного передбачення та посилення захисту інформації на тих напрямках безпеки де це найбільш необхідно. Для виконання таких завдань необхідно звернутись до стандартів забезпечення інформаційної безпеки ISO/IEC. Від так чинним законодавством України визначено перелік державних стандартів, які покликані забезпечити надійний рівень захисту інформації у програмному забезпеченні, що розробляється для тих чи інших державних установ [7]. Серед таких стандартів є наступні:

1) ДСТУ ISO/IEC 27000:2019 (ISO/IEC 27000:2018, IDT) Системи керування інформаційною безпекою.
2) ДСТУ ISO/IEC 27005:2019 (ISO/IEC 27005:2018, IDT) Управління ризиками інформаційної безпеки.
3) ДСТУ ISO/IEC TS 27008:2019 (ISO/IEC TS 27008:2019, IDT) Настанова щодо оцінювання захисту інформаційної безпеки.
4) ДСТУ ISO/IEC 27018:2019 (ISO/IEC 27018:2019, IDT) Кодекс усталеної практики для захисту персональної ідентифікаційної інформації (PII) у загальнодоступних хмарах, що діють як процесори PII.

5) ДСТУ ISO/IEC TS 27034-5-1:2019 (ISO/IEC TS 27034-5-1:2018, IDT)Структура даних керування протоколами та захистом застосунків. Схеми XML.

Система збирання та аналізування даних передбачає використання веб-технологій, як бази для збору інформації. Щоб система була захищеною вона має відповідати переліку стандартів ISO/IEC визначених чинним законодавством України. Оскільки система збирання та аналізування даних може бути атакованою як різним шкідливим програмним забезпеченням, так і кіберзлочинцями – потрібно передбачати ризики інформаційної безпеки. Для таких цілей у списку стандартів ISO/IEC є стандарт ДСТУ ISO/IEC 27005:2019 (ISO/IEC 27005:2018, IDT). Цей стандарт відповідає за управління ризиками інформаційної безпеки. Та якщо розглядати цей стандарт детально, ми можемо побачити, що він розрахований на велику кількість різних структур [8]. Безумовно така варіативність є його перевагою адже надає можливість адаптувати рекомендації даного стандарту до власної структури для тої чи іншої компанії. Та недоліком стандарту ISO/IEC 27005:2019 є відсутність нормативного аспекту. А точніше коли справа доходить до визначення сфери управління ризиками, організація повинна робити все незалежно від того, чи це сфера застосування системи управління інформаційними ризиками, чи навіть критерії ризику. Тому такий підхід є доцільним лише для організацій які готові інвестувати значні внутрішні ресурси в розробку власної методології. Для того, щоб компанії які неготові витрачати значні ресурси на розробку власних методів управління ризиками могли ефективно використовувати даний стандарт захисту, необхідною є також розробка простого та гнучкого методу управління ризиками, який зможе забезпечити задовільний чи навіть високий рівень безпеки для організацій з обмеженими ресурсами. Метод, який можна буде використовувати і для оцінювання ризиків систем збирання та аналізування даних.

Зважаючи на необхідність розробки захищеної системи збору та аналізу даних на базі веб-ресурсу реєстру випадків домашнього насилля та виявлений недолік стандарту ДСТУ ISO/IEC 27005:2019 (ISO/IEC 27005:2018, IDT), а саме відсутність предметного методу управління ризиками для проектів з обмеженим фінансуванням, що не здатні витрачати великі ресурси на розробку власної методології, можемо зробити висновок, що з науково-технічної точки зору розробка такої системи є доцільною.

1.2 Аналіз засобів аналогів захищених систем збору та аналізу даних

На сьогоднішній день у сучасному світі широко розповсюджені системи збору та аналізу даних тому, що кожна організація чи установа працює з великою кількістю різних даних клієнтів чи товарів. Великий обіг інформації потребує структуризації та обробки, саме в таких цілях і застосовують дані системи. Аналогів таких систем в мережі Інтернет достатньо, хоч і їхнє призначення може бути спрямоване в абсолютно різні галузі та професії, основним їх завданням все ж залишається збір та аналіз даних. Тож давайте розглянемо досить відомі аналоги з такими назвами: IBS мобільний склад, SendPulse CRM, Loger Pro, REGMIK система збору даних.

IBS мобільний склад – це програмний додаток терміналів для універсального збору даних на операційній системі Android (рис. 1.1). Спеціальна система для ТЗД (смартфон чи планшет). Цей додаток працює на мобільних пристроях з ОС андроїд від 4.1 версії та вище. Надає двосторонню комунікацію та обмін даними через Wi-Fi з обліковою базою підприємства. Також має можливість роботи з Bluetooth-сканером штрих-кодів, під’єднаним до смартфону. Працює як в онлайн так і офлайн режимах. Такий додаток для збору даних тісно інтегровано з програмами обліку IBS. Також можлива робота з віддаленими складами в режимі офлайн [9].

IBS мобільний склад має ряд суттєвих переваг:

– працює в онлайн режимі з документами в обліковій базі по Wi-Fi;

– працює в офлайн режимі з дампами облікових баз з подальшим їх вивантаженням до облікової бази;
– в автоматизованому режимі створює та заповнює документи: продаж, замовлення, списання та переоцінка.

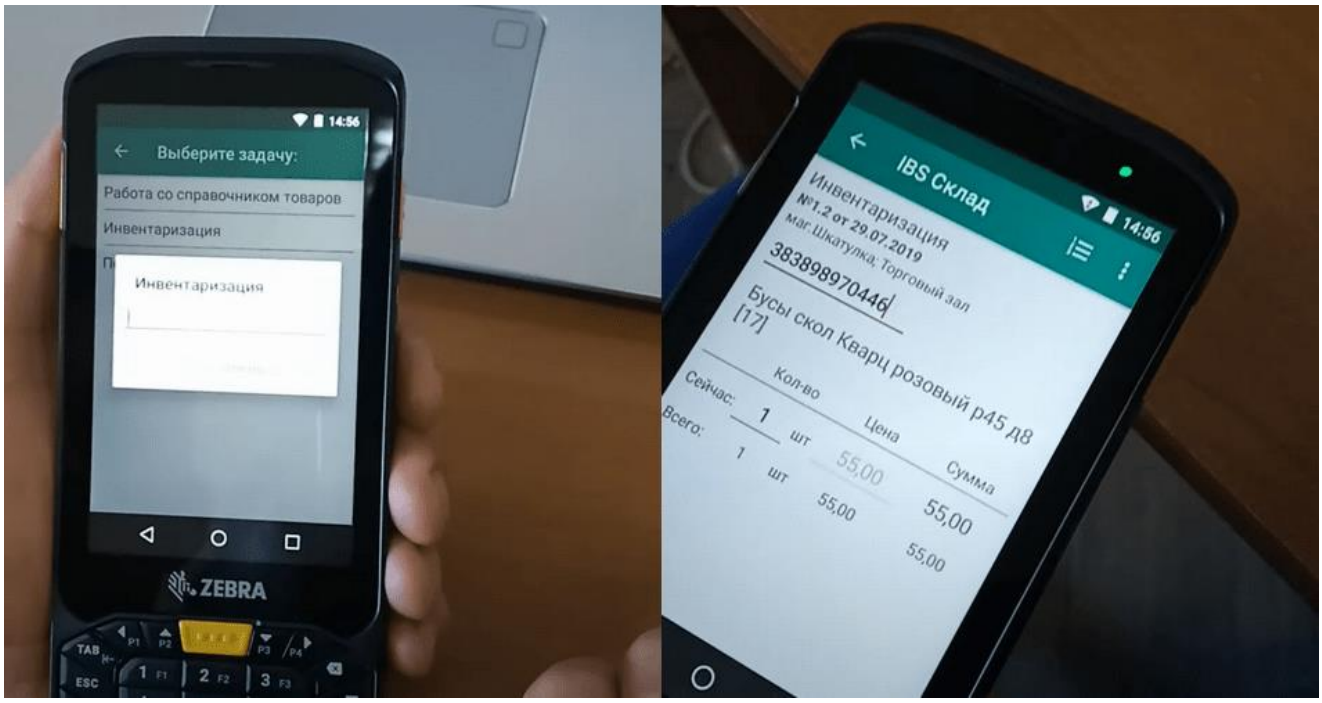


Рисунок 1.1 – Зображення робочої області програми IBS мобільний склад

Недоліками системи «IBS мобільний склад» можна назвати відсутність можливості одночасні роботи над одним документом одразу декількох користувачів. Також недоліком є відсутність авторизації користувача.

SendPulse CRM – автоматизований онлайн-ресурс для клієнтської бази. Ця система спрямована на маркетинг та продаж, з її допомогою користувач може приймати замовлення , збирати контакти в базу та запускати маркетингові компанії. Також є можливість командної роботи над проектом, кілька користувачів можуть одночасно працювати над виконанням одного замовлення, вносити правки в базу клієнтів. Підтримує інтеграцію різних каналів спілкування з клієнтами. Надає можливість створювати воронку продажів для власного бізнесу, для прискорення закриття вигідних угод [10].

Це дозволить користувачеві працювати з клієнтом за сценарієм: консультація, усунення заперечень, заохочення та супроводження угоди від заявки до доставки. Можна налаштовувати розсилку для оптимізації роботи команди та автоматизації спілкування зі своїми клієнтами (рис. 1.2).

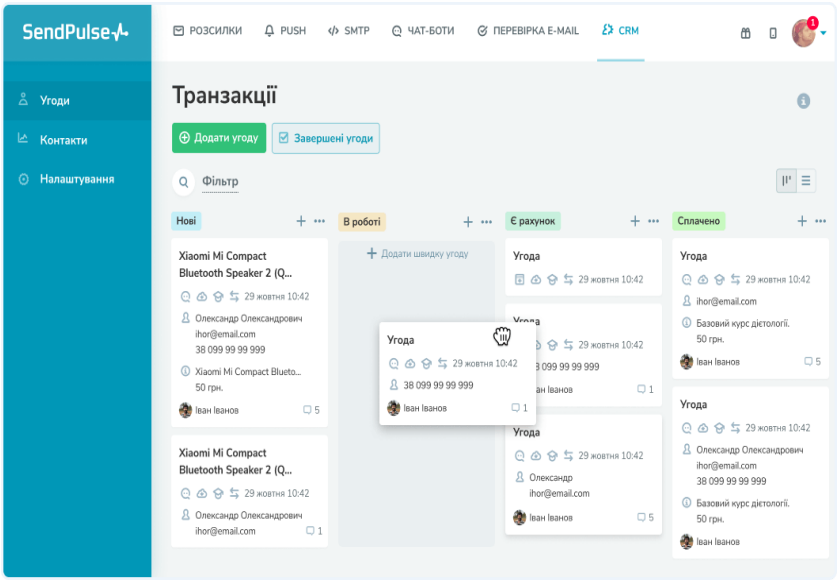


Рисунок 1.2 – Зображення робочої області програми SendPulse CRM

Переваги даної системи:

– можливість інтеграції різних каналів спілкування з клієнтами;
– безкоштовність та можливість командної роботи;
– налаштування авторозсилки повідомлень.

Недоліками – суттєво збільшена вразливість до спам атак через канали комунікацій, відсутність можливості роботи в офлайн режимі.

Loger Pro –програмний продукт призначений для збору та аналізу даних на персональному комп’ютері (рис. 1.3). Це програмне середовище на Mac та Windows комп’ютери для збирання та аналізування даних, що є потужним та функціональним інструментом для професійного підходу до навчання. До основних характеристик цього програмного забезпечення належать такі функції: автоматична ідентифікація лічильників, збір інформації з великої

кількості датчиків, створення графіків прогнозу для збору даних. Враховуючи сучасні тенденції розвитку великих даних та потребу в експертах у сфері data science – простого збору даних недостатньо! Різні типи навичок аналізу й оцінки інформації та її розподілу важливі для групування, побудови тенденцій і пошуку дотичних до переломних точок. Разом з розробкою потужного інструменту для соціології та міжнародних торгових відносин, Logger Pro є інноваційним освітнім інструментом, який виходить далеко за межі дисциплін фізики та математики. Для того щоб навчитися використовувати даний програмний засіб можна з легкістю використовувати записані на екрані інструкції та відео уроки, Також наявні відео для початкових аналізів. Покращена система ліцензування яка дозволяє купувати одну ліцензію на весь освітній заклад, не потрібно рахувати кількість учнів чи комп'ютерів, освітяни можуть продовжувати навчання вдома, а оновлення продукту – безкоштовні [11].

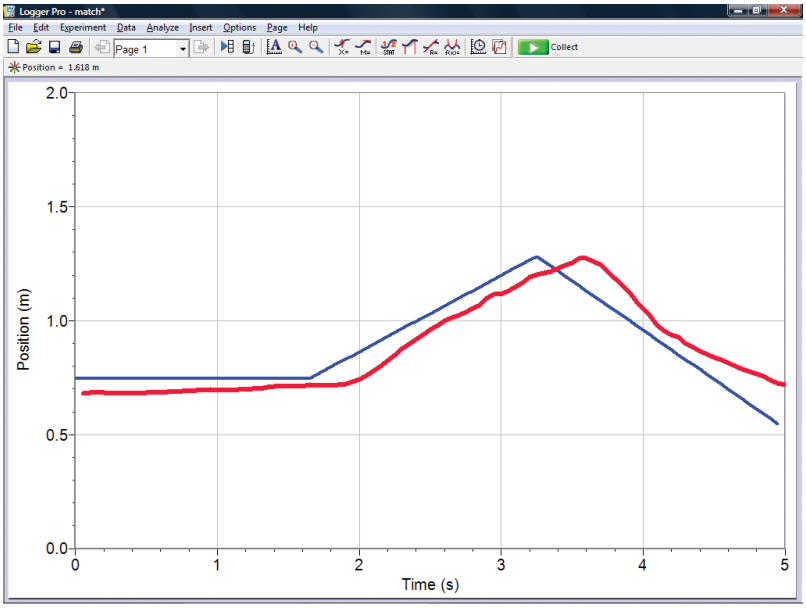


Рисунок 1.3 – Зображення робочої області програми Loger Pro

До переваг Loger Pro можна віднеси:

– синхронні візуалізація експерименту та запис відео;

– створення графіків до збору даних;
– аналіз статистики.

Недоліками стають відсутність можливості імпорту даних, вбудованого друку схем та статистичних графіків.

REGMIC система збору даних – це програмне забезпечення, що створене для збирання, візуалізації та зберігання даних в реальному режимі часу (рис. 1.4). Дані здатні передаватися по мережі за протоколом ModBus RTU, Ethernet, FTP або можуть безпосередньо завантажуватися з приладу-логгер чи файлу. Зберігання даних відбувається в базі даних Firebird 3.0. Дані візуалізуються у формі графіка/таблиці та можуть бути відображені на мнемонічній діаграмі потоку процесу. Підтримка архівації, експорту та імпорту даних, формування різноманітних аварійних оповіщень, звітів та дистанційне керування обладнанням. Призначенням системи є збирання даних з різних вимірювальних приладів і обладнання, обробка та візуалізація сигналів у реальному часі, також наявний аналіз даних, надсилання повідомлень про несправність приладів і аварійні ситуації [12].

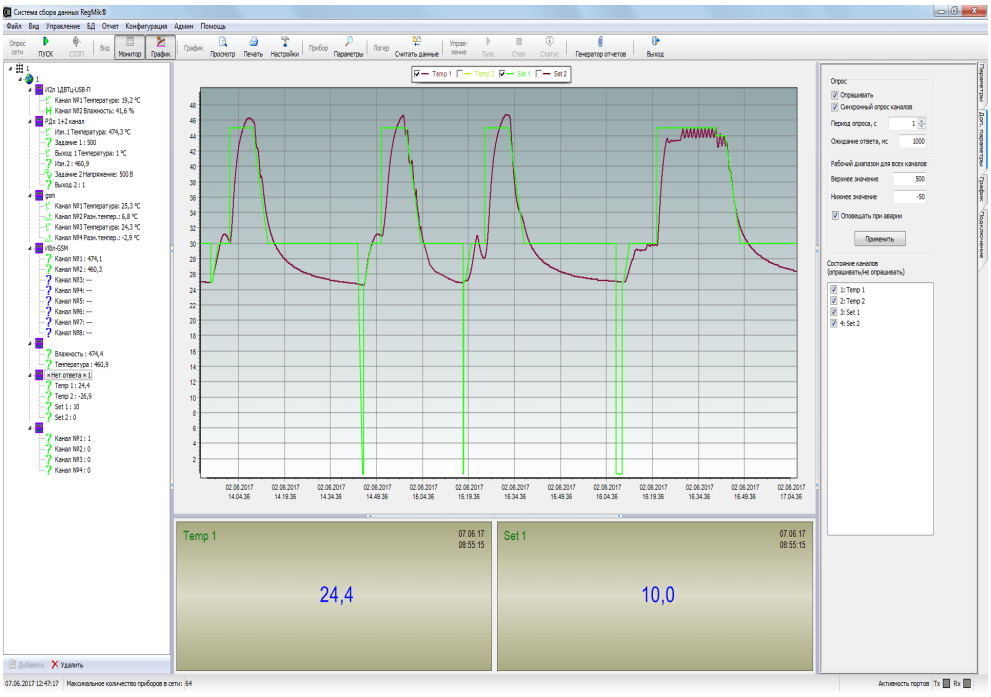


Рисунок 1.3 – Зображення робочої області програми REGMIC

До переваг даного програмного забезпечення можна віднести можливість дистанційно керувати та візуалізувати технологічні процеси, можливість надсилання оповіщень SMS або Email, наявність роботи з резервними копіями БД. Недоліком є відсутність захисту від несанкціонованого доступу.

Провівши аналіз наведених вище програмних аналогів, визначено їх переваги й недоліки, визначено доречні та недоречні підходи до реалізації захищеної системи збору та аналізу даних, які будуть враховані при розробці власної системи на базі веб-сайту реєстру випадків домашнього насилля (табл. 1.1).

Таблиця 1.1 – Порівняльні характеристики програмних продуктів

Критерій	IBS мобільний склад	SendPulse CRM	Loger Pro	REGMIK система збору даних	Захищена система збирання та аналізування даних
Командна робота над проектом	-	+	-	-	+
Захист від несанкціонованого доступу	+	+	+	-	+
Оцінка ризиків кібербезпеки	-	-	+	+	+
Побудова статистичних графіків	-	+	+	+	+
Email або SMS сповіщення	-	+	-	+	+
Робота в офлайн режимі	+	-	+	+	+
Хешування даних при передачі на сервер	-	-	-	-	+

З результатів порівняльного аналізу аналогів у таблиці можемо побачити, що розробка власної системи є доцільною. В результаті виконання кваліфікаційної роботи отримаємо веб-ресурс, що виправить всі перелічені

вище недоліки аналогів. Він дасть змогу зручно та швидко зібрати необхідні дані, після чого передасть їх до бази даних та виконає їх аналіз.

1.3 Постановка завдання

В сучасному світі захищені системи збору та аналізу даних не є рідкістю та все ж кожна з реалізацій хоч і призначена для своїх специфічних завдань, має ряд недоліків пов’язаних з сферою захисту інформації чи стійкості до кібератак. Для захисту від DDoS-атак можливо відсікання трафіку з типовими заголовками запитів за відведений час між ними. Провівши аналіз стан сучасних захищених систем збору та аналізу даних та їх порівнянь з існуючими аналогами було визначено перелік завдань що потрібно виконати для розробки власної захищеної системи збору та аналізу даних на базі веб-ресурсу реєстру випадків домашнього насилля:

- розробити метод оцінювання ризиків у сфері кіберзахисту для захищеної системи збору та аналізу даних з урахуванням стандарту ДСТУ ISO/IEC 27005:2019;
- розробити модуль захисту від несанкціонованого доступу до даних – такий модуль можливо реалізувати на принципах авторизації та автентифікації;
- розробити модуль збирання даних – передбачається реалізація за допомогою інтернет-форм з подальшою обробкою та надсиланням до місця постійного збереження даних (бази даних) ;
- розробити модуль аналізування введених користувачами даних – створення статистики, виведення графіків;
- розробити модуль оцінювання ризиків ІБ;
- провести експериментальне дослідження системи.

Отже, проаналізовано сучасний стан розробок захищених систем для збору та аналізу даних, визначено ряд основних можливих вразливостей таких систем, розглянуто перелік державних стандартів для інформаційної безпеки. Визначено та розглянуто аналоги захищених систем, описано їхні переваги та недоліки, проведено порівняльний аналіз та визначено актуальність розробки власної системи для спеціальних задач. Також сформовано перелік завдань, урахуванням усіх недоліків існуючих аналогів, які необхідно виконати задля створення власної захищеної системи збору та аналізу даних.

2 РОЗРОБКА СТРУКТУРИ ТА МОДЕЛЕЙ ЗАХИЩЕНОЇ СИСТЕМИ ЗБИРАННЯ ТА АНАЛІЗУВАННЯ ДАНИХ

2.1 Основні положення стандарту управління ризиками в кібербезпеці

Для управління ризиками в сфері інформаційної безпеки було розроблено міжнародний стандарт ISO/IEC 27005. Він був створений та опублікований Міжнародною організацією стандартизації ISO та міжнародною електотехнічною комісією IEC. Цей стандарт підтримує інформаційну безпеку керуючись підходом до управління ризиками. На відміну від методів таких як NIST структура кібербезпеки, цей стандарт підлягає сертифікації [13].

ISO 27005 базується на вказівках, наданих у ISO/IEC 27001 та ISO/IEC 27002. Спочатку він був опублікований 2008 року у червні під назвою ISO/IEC 27005:2008, згодом був перевиданий у 2011 році і знову вже у 2018. ISO 27005 розділи з шостого по дванадцятий розробляють підхід до управління ризиками для інформаційних систем. Розділ сьомий більш конкретно стосується аналізу ризиків, який є основою будь-якої хорошої стратегії кібербезпеки. Глава восьма присвячена оцінці ризиків. Розділи з дев'ятого по дванадцятий детально описують, як реалізувати та контролювати свою стратегію управління ризиками.

Міжнародна організація стандартизації радить стандарт ISO 27005 не лише для компаній, а й для державних органів, таких як урядові установи різних рівнів, а також для некомерційних організацій (НКО). На ділі цей стандарт інформаційної безпеки використовується для забезпечення конфіденційності, доступності та цілісності даних основних інформаційних активів організації. Він призначений для всіх структур, схильних до кіберризиків і постійно зростаючого обсягу даних усередині служби.

Стандарт ISO 27005 заснований на підході до управління ризиками, він призначений для підтримки надійної реалізації інформаційної безпеки.

Навчання співробітників, зазвичай потрібне для розвитку навичок впровадження ефективного процесу управління ризиками інформаційної безпеки. Персонал, який пройшов навчання за ISO 27005 може виявляти, аналізувати, вимірювати та усувати ризики.

Стандарт також має допомогти компаніям налаштувати свої СУІБ (система управління інформаційною безпекою). СУІБ включає створення процесів і політик кібербезпеки, постійне поліпшення управління ризиками та облік людських і технічних факторів у процесі. З цією метою стандарт ISO 27005 слідує логіці, що нагадує методологію безперервного вдосконалення PDCA (плануй, роби, перевіряй, дій) [14].

– плануй: ідентифікація та оцінка кіберризиків з подальшим стратегічним розглядом відповідних заходів щодо пом'якшення наслідків;
– роби: виконайте розглянуті заходи;
– перевіряй: запустити перевірку продуктивності;
– дій: відстежуйте та покращуйте свою стратегію управління ризиками.

У цьому міжнародному стандарті міститься понад двадцять сторінок з описом різних підходів до управління ризиками інформаційної безпеки. Проте загалом документ підтримує загальну концепцію методології на чотирьох основних етапах:

1) Контекстуалізація управління ризиками. Контекстуалізація аналізу ризиків визначає початкову та кінцеву точки управління ризиками. Це також час для встановлення деяких критеріїв:

– критерії оцінки допомагають визначити активи, схильні до ризику кіберризику, і порогові значення, за якими необхідно усувати ризики;
– критерії впливу відповідають мінімальному рівню наслідків, за межами якого слід розглядати ризик;
– критерії прийнятності – це порогові значення нижче яких ризик можна допускати.

2) Оцінка ризиків.

На цьому кроці спочатку визначаємо фактори ризику: організацію, інформаційні системи, групи даних та служби. Далі потрібно визначити вразливі місця та загрози, що пов’язані з цими елементами.

За стандартом ISO 27005 далі потрібно узгоджувати такі загрози і їх виникнення з потребами безпеки організації. Весь цей процес допомагає розставити пріоритети відповідно до критеріїв, які ви визначили на першому етапі.

Стандарт ISO 27005 допомагає виявляти вразливість кібербезпеки, але в той же час не надає шкали оцінки ризиків. Команда, яка є відповідальною за застосування стандартів, має створити власну рейтингову систему. Така система може бути заснована на якісних чи кількісних методах оцінювання, причому останні ґрунтуються на вимірних витратах. Виявляється найчастіше аналіз є якісним так, як прписи стандарту ISO – відсутні. На рисунку 2.1 зображено етапи управління ризиками за стандартом ISO 27005

Рисунок 2.1 – Етапи управління ризиками за стандартом ISO 27005

3) Стратегія реагування на ризики

На цьому етапі структура повинна встановити цілі кібербезпеки з урахуванням результатів, що були отримані на другому етапі. Як тільки ці цілі встановлені, можна скласти власні специфікації, що допоможе розробити заходи для боротьби з ризиком.

У ISO 27005 концептуалізація таких заходів означає зважування ризиків та витрат на їхнє виправлення. Після чого є чотири можливі стратегії:

– заперечувати або уникати: заява про те, що ваша організація надто серйозно ставиться до кіберзагроз і її слід уникати за будь-яку ціну. Потім ви можете ухвалити рішення про припинення діяльності, яка може бути причиною цього;
– зередача: об'єкт поділяє ризик із третьою стороною (страховою або субпідрядником з кібербезпеки), яка може принаймні фінансово захистити від ризику;
– пом'якшення: розробка заходів для зниження впливу або ймовірності ризику, щоб зробити його більш прийнятним;
– утримання: ризик вважається прийнятним та недостатньо небезпечним для організації, щоб не брати його до уваги. Кожен варіант має залишковий ризик, який слід систематично оцінювати.

4) Прийняття ризику.

Стратегії реагування на ризики та залишкові ризики мають пройти етап «прийняття». Насправді це означає, що весь план реагування має отримати схвалення вищого керівництва. На цьому етапі керівники різних відділів здатні поставити під сумнів витрати, які здаються їм занадто високими, або йти на певні ризики. Такі винятки мають бути обґрунтовані.

Хоча теоретично методологія ISO 27005 закінчується на цьому, слід зазначити, що робота, виконана організацією для її впровадження, може використовуватися як частина її процедур моніторингу та перевірки. Це

включає історію виявлених ризиків, виявлених сценаріїв, виконаного аналізу ризиків і налаштованих стратегій виправлення. Звісно, цю методологію слід повторювати у міру розвитку загроз та вразливостей. Ця робота також підтримує спілкування із зацікавленими сторонами.

Стандарт керування кіберизиками має декілька переваг. Одним із найбільш примітних є його адаптованість до різних типів організацій та структур Однак йому не вистачає директивного виміру критеріїв аналізу ризиків.

До переваг методології ISO 27005 можна віднести такі характеристики:

- цей метод можна використати самостійно;
- команда розвиває навички, необхідні для структурованого управління кіберризиками;
- він виявляє організаційні слабкості та різні загрози;
- організація починає отримувати вигоду від використання стійкої СУІБ;
- цей метод адаптується до всіх структур, включаючи організації, що адаптуються до умов, які постійно змінюються;
- підвищення довіри зацікавлених сторін.

До недоліків ISO 27005 можна віднести відсутність нормативних аспектів. Коли доводиться визначати галузь управління ризиками, компанія має робити все незалежно від того, чи це критерії ризику чи застосування СУІБ. Тому такий підхід підходить лише для організацій, що готові вкладати значні ресурси в створення власної методології [15].

2.2 Розробка методу оцінювання ризиків в кібербезпеці

Так як стандарт ISO 27005 в першу чергу призначений для експертів у сфері кіберзахисту та в деякій мірі є ненормативним з точки зору рішень про те, як саме реагувати та вимірювати той чи інший ризик. Цей стандарт базується на суб’єктивному підході окремих експертів залучених в той чи інший момент

до розгляду серйозності конкретного кіберризику. Даний стандарт можна назвати якісним. Він заснований на «досвіді експертів з ІТ», для класифікації ризиків за шкалами, що є суб'єктивними. Ці шкали класифікують ризик згідно кольорової карти, що змінюється від червоного до зеленого (рис. 2.2).

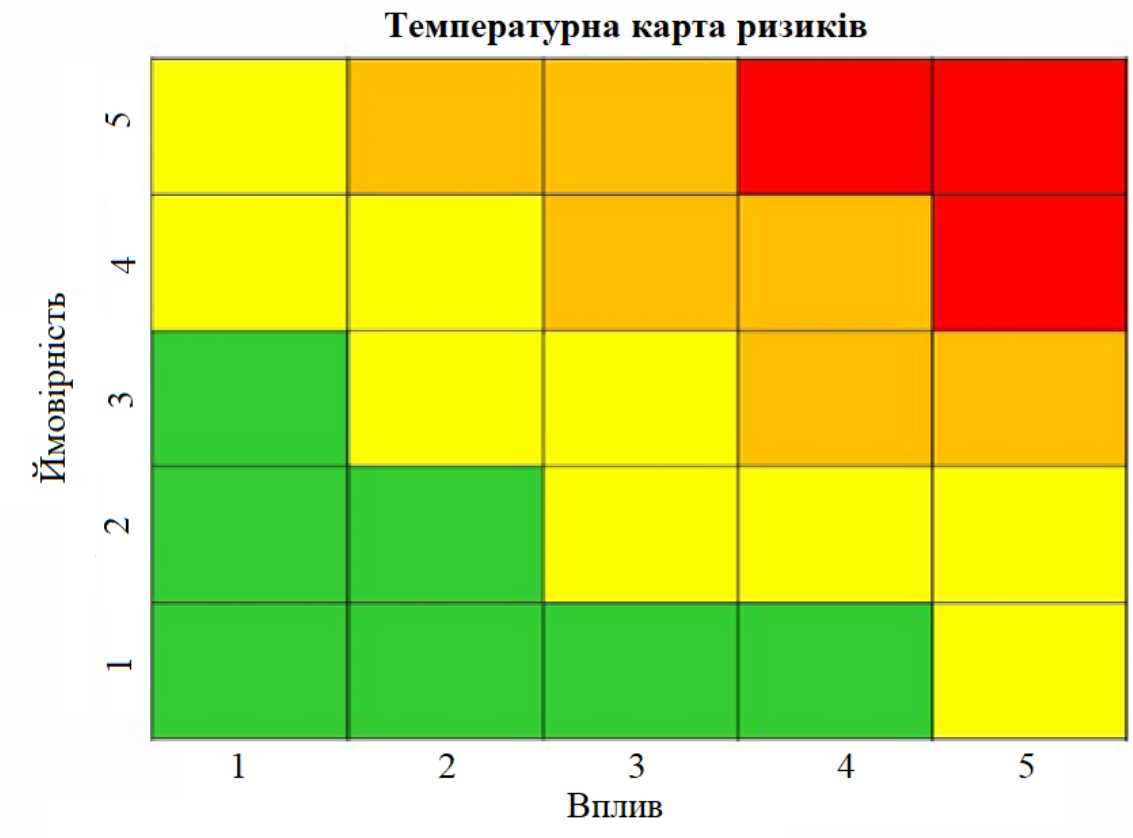


Рисунок 2.2 – Кольорова карта критичності ризику

Авжеж, такий метод забезпечує необхідні звички кібергігієни та хорошу практику. Та оскільки він грунтується на суб'єктивному аналізі ризиків, він не надає загальної структури для усіх бізнес-функцій.

Для того щоб залучений до оцінювання ризиків експерт з кібербезпеки міг швидко і точно розрахувати стан ІБ системи з урахуванням методології ISO 27005 та надання зрозумілих показників оцінювання ризиків для усіх бізнес-функцій необхідно розробити метод оцінювання ризиків у якому використати елементи суміжних методологій, де використовуються елементи кількісного

виміру для математичної оцінки ризику яка в свою чергу буде відповідати одному з критеріїв ступеня ризику.

Крок 1. На першому кроці вводимо набори термінів для опису основних наборів стану інформаційної безпеки та їх підмножин. Загальний набір стану ІБ підмножина «А» поділяється на п'ять підмножин станів:

A_1 – вкрай поганий стан ІБ;

A_2 – поганий стан ІБ;

A_3 – середній стан ІБ;

A_4 – відносно безпечний стан ІБ;

A_5 – максимальний безпечний стан ІБ.

Відповідно набору А повної множини ризиків ІБ згідно якого множина загрози В також поділяється на 5 підмножин:

B_1 – граничний ризик загрози;

B_2 – високий ризик загрози;

B_3 – середній ризик загрози;

B_4 – низький ризик загрози;

B_5– незначний ризик загрози.

Нехай В прийматиме значення від одиниці до нуля. Для будь-якого окремого показника оцінки інформаційної безпеки позначатиметься F_i де, повний набір його значень позначатиметься як M_i та поділяється на такі підмножини:

M_{i1} – дуже низький рівень M_i;

M_{i2} – низький рівень M_i;

M_i3 – середній рівень M_i;

M_{i4} – високий рівень M_i;

M_{i5} – дуже високого рівня M_i.

Також є умова відповідності множин A, B та M такого вигляду: коли усі показники аналізу мають, згідно класифікації, рівень підмножини M_{ij}, то стан інформаційної безпеки кваліфікується як A_j, і ступінь ризику загрози кваліфікується як B_j. Виконання такої умови має вплив на вірну кількісну класифікацію всіх рівнів показників.

Крок 2. Побудувати набір показників $F = \{F_i\}$ в кількості $N = 4$, що на думку експерта-аналітика впливає на оцінювання загрози кіберризику (табл. 2.1).

Таблиця 2.1 – Набір показників F

Назва показника	Значення показника
F_1	1.2
F_2	0.7
F_3	0.025
F_4	0.004

Крок 3. Визначимо відповідність для кожного показника рівень його значення для аналізу P_i. Для оцінки такого рівня, необхідно розташувати значення в спадному порядку величин, аби дотримуватися правила:

$$P_i = 1/N \tag{2.1}$$

Необхідно враховувати що, якщо систему показників розмістити в порядку спадання їх значень – значення і-того індексу має визначатися за правилом Фішберна.

$$P_i = \frac{1}{N} = \frac{1}{4} = 0.25 \tag{2.2}$$

Правило Фішберна [16] показує те, що нам нічого не відомо про рівень значення показників (2.1). В той час як оцінка (2.2) яка показує максимум ентропії існуючої невизначеності інформації, що стосується нашого об'єкту дослідження.

Крок 4.Створення класифікації значень b фактора ризику B як критерію поділу даного набору на підмножини (табл. 2.2).

Таблиця 2.2 – Значення показника b.

Інтервал B	Набір назв відповідних підмножинам
$0.8 < b \leq 1$	B_1 –граничний ризик загрози
$0.6 < b \leq 0.8$	B_2 –високий ризик загрози
$0.4 < b \leq 0.6$	B_3 – середнього ризику загрози
$0.2 < b \leq 0.4$	B_4 – низький ризик загрози
$0 < b \leq 0.2$	B_5– незначна загроза ризику

Крок 5. Створити класифікацію значення f для показників F як критерію поділу повного набору їх значень у підмножині типу M (табл. 2.3).

Таблиця 2.3 – Розподіл відносно підмножини значень

Назва показника	Критерії поділу підмножини				
	M_{i1}	M_{i2}	M_{i3}	M_{i4}	M_{i5}
F_1	$f_1 \leq 0.02$	$0.02 < f_1 \leq 0.16$	$0.16 < f_1 \leq 0.84$	$0.84 < f_1 \leq 1$	$1 < f_1$
F_2	$f_2 \leq 0.02$	$0.02 < f_2 \leq 0.16$	$0.16 < f_2 \leq 0.84$	$0.84 < f_2 \leq 1$	$1 < f_2$
F_3	$f_3 \leq 0.02$	$0.02 < f_3 \leq 0.16$	$0.16 < f_3 \leq 0.84$	$0.84 < f_3 \leq 1$	$1 < f_3$
F_4	$f_4 \leq 0.02$	$0.02 < f_4 \leq 0.16$	$0.16 < f_4 \leq 0.84$	$0.84 < f_4 \leq 1$	$1 < f_4$

Крок 6. Потрібно зробити оцінку поточного рівня показників та обмежити результати (табл. 2.4).

Таблиця 2.4 – Оцінка рівнів показників F.

Назва показника	Діапазон показника F
Дуже високий	$F_1 > 1$
Високий	$0.1 < F_2 \leq 1$
Середній	$0.01 < F_3 \leq 0.1$
Низький	$0.001 < F_4 \leq 0.01$
Дуже низький	< 0.001

Крок 7. На даному кроці ми класифікуємо поточні значення f у відповідності до критеріїв що наведені в таблиці 3. У результаті класифікації маємо таблицю 5: де $\lambda_{ij} = 1$ при $m_{i(j-1)} < F_i < m_{ij}$ і $\lambda_{ij} = 0$, коли значення не потрапляє у обраний діапазон класифікації (табл. 2.5).

Таблиця 2.5 – Результат класифікації

Назва показника	Значення	Результат класифікації відносно підмножин				
		M_{i1}	M_{i2}	M_{i3}	M_{i4}	M_{i5}
F_1	0.25	0	0	0	0	1
F_2	0.25	0	0	1	0	0
F_3	0.25	0	1	0	0	0
F_4	0.25	1	0	0	0	0

Крок 8. Потрібно виконати арифметичні обчислення для оцінювання ступеня ризику B за допомогою наступної функції:

$$B = \sum_{i=1}^{N} P_i \lambda_{ij} \ \ \sum_{j=1}^{5} b_i, \tag{2.3}$$

де

$$b_i = 0.8 - 0.2 * (j - 1). \tag{2.4}$$

Обчислюємо наступним чином:

$$B = 0.2 * 0.25 + 0.4 * 0.25 + 0.6 * 0.25 + 0.8 * 0.25 = 0.5.$$

Після обчислення В бачимо, що його значення відповідає підмножині середнього ризику загрози інформаційної безпеки. А отже отриманий результат ступеня ризику ІБ відповідає результатам дослідження.

2.3 Структура та бізнес процеси підсистеми аналізу

Оскільки веб-система має бути захищеною від несанкціонованого доступу згідно нормативного документу систем технічного захисту інформації – є необхідним процес автентифікації користувача. На рисунку 2.3 зображено модель взаємодії користувача та модуля автентифікації з наданням доступу до системи.

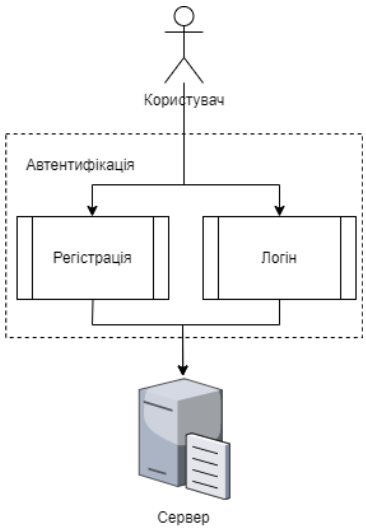


Рисунок 2.3 – Модель взаємодії користувача та модуля автентифікації

Відтак з вищенаведеного рисунку можна побачити, що отримати доступ до сервера оминувши модуль автентифікації – неможливо.

Для модуля збирання даних спочатку необхідно щоб користувач ввів свої дані, потім формується запит із зібраними даними на сервер, після чого сервер зберігає внесені користувачем дані до сховища (бази даних). Цей процес взаємодії користувача та модуля збирання даних зображено на рисунку 2.4.

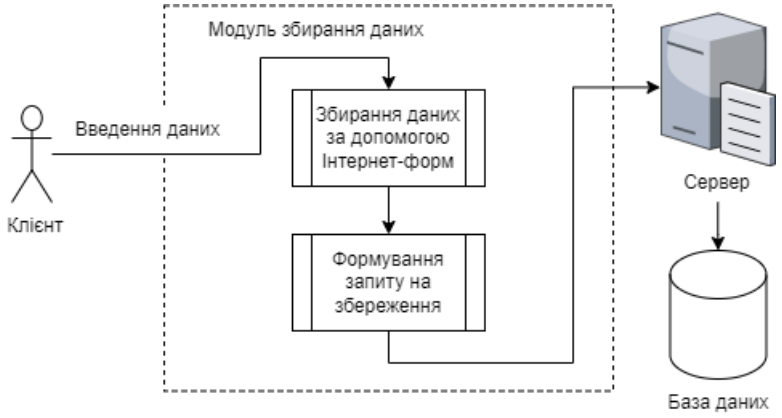


Рисунок 2.4 – Модель взаємодії користувача та модуля збирання даних

Також необхідно розробити модель взаємодії модуля аналізування даних та сервера. Де сервер буде надсилати дані зібрані за допомогою модуля збирання (рис. 2.5).

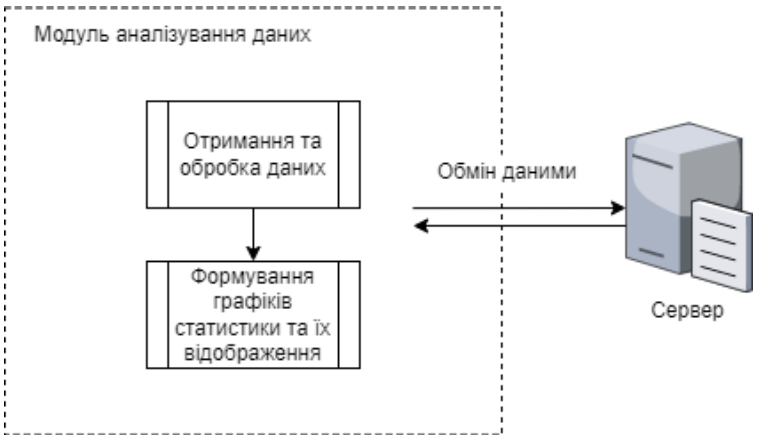


Рисунок 2.5 – Модель взаємодії модуля аналізування даних та сервера

Як можна побачити з вище наведеного рисунку, модуль отримує дані з сервера проводить їх обробку та виводить графіки, а потм надсилає оброблену статистику назад на сервер.

Загалом структура підсистеми аналізу має розгалужений алгоритм роботи зображений на рисунку 2.6, підсистема поділена на модулі, які взаємодіють між собою:

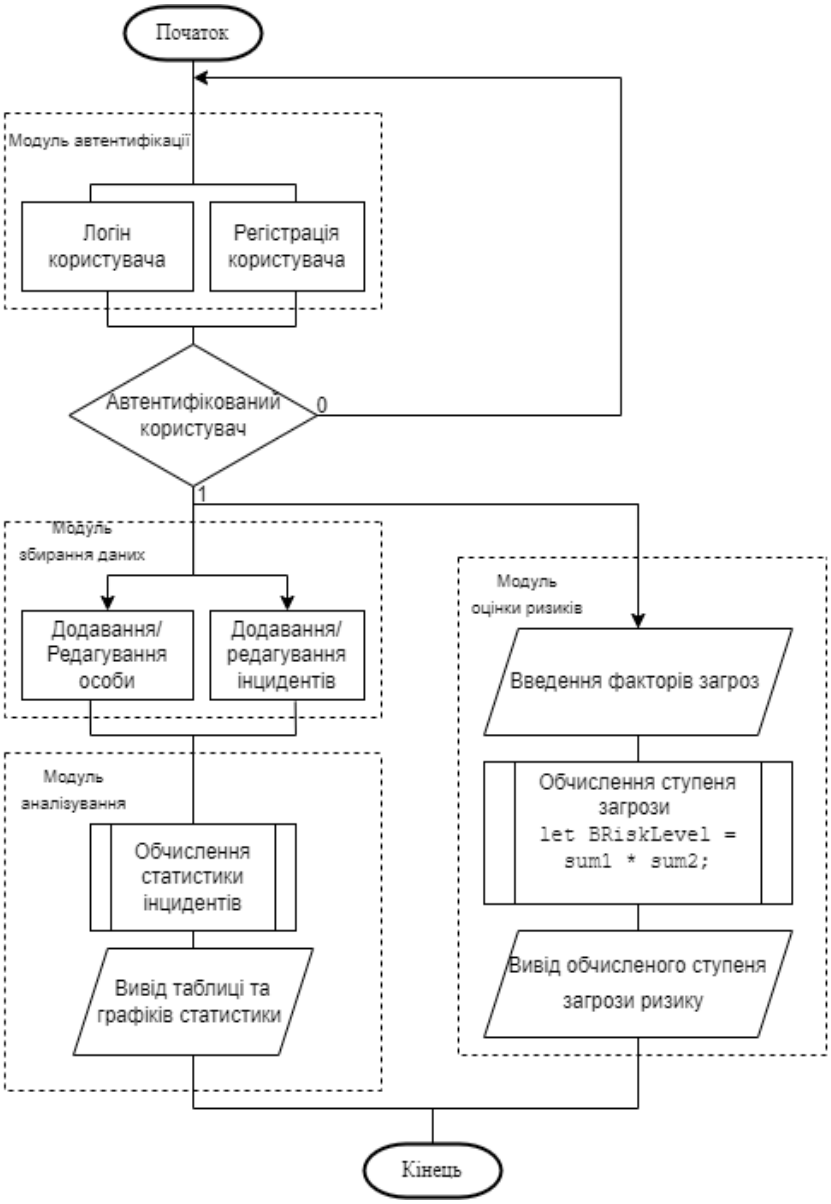


Рисунок 2.6 – Загальний алгоритм роботи підсистеми аналізу

Тож можна зробити висновок, що у даному розділі було розроблено моделі поведінки користувача та роботи підсистеми аналізу, також розроблено метод оцінювання ризиків ІБ. Розроблено загальну блоксхему алгоритму роботи клієнтської частини системи збирання та аналізування даних для спеціальних задач.

3 РОЗРОБКА ПРОГРАМНОГО ДОДАТКУ

3.1 Обґрунтування вибору засобів реалізації програмного засобу

JavaScript – це однопоточна об'єктно-орієнтована прототипна мова програмування [17]. Вона реалізована за стандартом ECMAScript. Є найпопулярнішою у сфері веб-розробки для створення сценаріїв логіки сайту. Забезпечує взаємодію з користувачем на стороні клієнта, надсилає асинхронні запити на сервер та керує діями браузера. JS також забезпечує інтерактивність та анімацію для елементів веб-сторінки. Мова побудована за принципами наслідування прототипів, має динамічну та слабку типізацію і підтримує як імперативну, і декларативну парадигми програмування [17].

Python – це мова високого рівня зі строгою типізацією даних. Завдяки структурам даних та динамічному зв'язуванню ви можете швидко розробляти програми та комбінувати існуючі компоненти. Мова підтримує такі парадигми програмування, як ООП, функціональна та процедурна. Мова підтримує модульні пакети, що дозволяють повторно використовувати вже написаний код. Стандартні бібліотеки та інтерпретатори доступні у вихідному та скомпільованому вигляді [18].

PHP – це мова програмування, яка базується на створенні веб-сторінок з використанням сценаріїв на стороні сервера. Ця мова створює HTML-розмітку на стороні сервера та передає її у браузер. Такий підхід гарантує конфіденційність та безпеку даних, але блокує можливість створення повністю інтерактивних сторінок сайту. Тандемне з'єднання двох мов програмування часто використовується для вирішення задач інтерактивності. Іншими словами, PHP створює сценарії JavaScript, які забезпечують інтерактивність сторінки, але не ставлять під загрозу безпеку даних [19].

Згідно зі статистикою опитування на сайті Stackoverflow, мова програмування JavaScript є найпопулярнішою мовою програмування серед розробників різного рівня. JavaScript у поєднанні з мовою гіпертекстової

розмітки HTML5 та мовою листа у стилі CSS3 робить його потужним інструментом для створення веб-сайтів з інтерфейсом різної складності. Пам'ятайте, що ця мова програмування використовується лише для цілей, згаданих вище. Варто зазначити, що він також використовується для створення мобільних та десктопних програм. Оскільки JavaScript написаний відповідно до стандартів організації ECMA, у спільноті розробників його іноді називають ECMAScript. Як і будь-яка інша сучасна мова програмування, JavaScript активно підтримується. Глобальне оновлення ES6 (ECMAScript 6) – це друга за величиною версія JavaScript за останнє десятиліття, яка містить безліч корисних рішень, раніше недоступних. З того часу з'явилося багато нових ключових слів, таких як «let» та «const», що дозволяють визначити тип даних (змінна чи константа). Гнучкість мови програмування також є важливою. Оскільки JavaScript має безліч розширень таких, як JQuery та фреймворків – React.js, Node.js, Vue.js та інші.

Судячи з рейтингу мов, показаному на рисунку 3.1, JavaScript є найпопулярнішою комерційною мовою розробки станом на 2022.

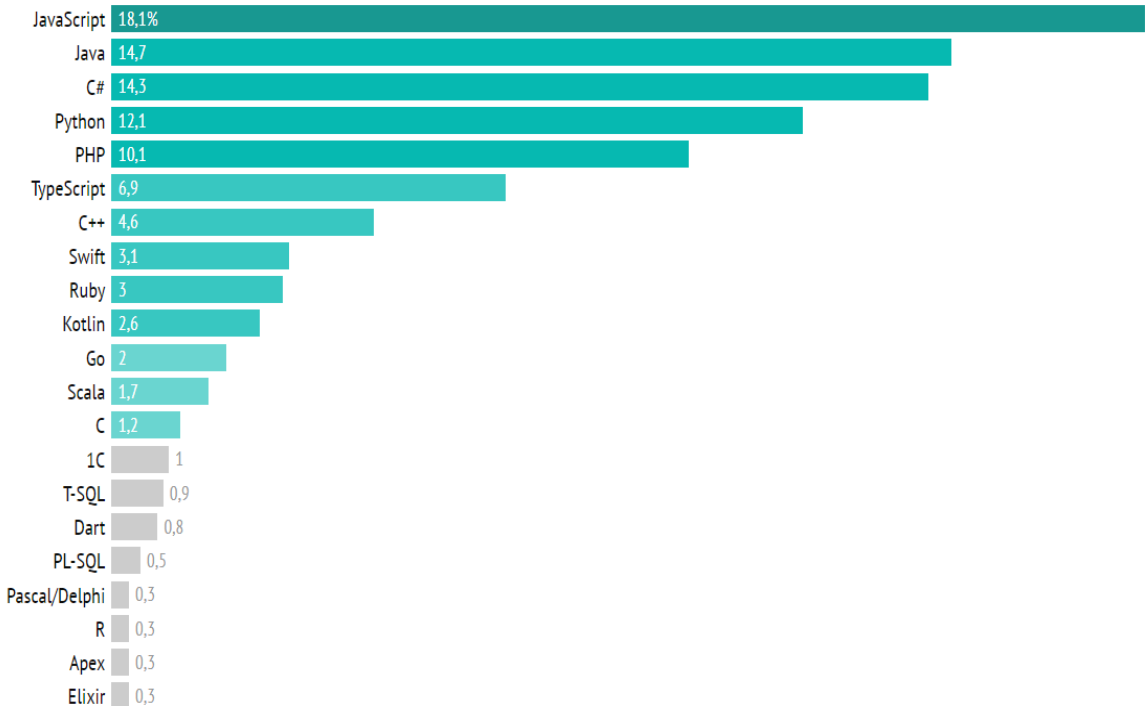


Рисунок 3.1 – Рейтинг мов програмування 2022р.

В таблиці 3.1 проведено порівняльний аналіз розглянутих мов програмування згідно визначених критеріїв.

Таблиця 3.1 – Порівняння мов програмування

Характеристики	Мова програмування		
	JavaScript	Python	PHP
Об'єктно-орієнтована	+	+	-
Динамічна типізація та асинхронність	+	+	+
Слабка типізація	+	-	-
Імплементація засобів кібербезпеки	+	+	+
Забезпечення інтерактивності сторінки	+	+	-
Робота на серверній частині	+	+	+
Функції роботи з інтерфейсом клієнта	+	+	-
Вбудовані функції математичної статистики	+	-	-

Отже, розглянувши наведений вище порівняльний аналіз, можна зробити висновок, що мова JavaScript підходить для розробки підсистеми аналізу захищеної системи збирання та аналізування даних для спеціальних задач. Він

відповідає основним вимогам для розробки програмних продуктів, оскільки має суттєві переваги перед Python та PHP. Мова реалізує необхідні парадигми ООП-програмування, надає значні ресурси та інструменти для проектування інтерфейсів на стороні клієнта та забезпечує їхню інтерактивність.

Задля визначення найкращого середовища розробки веб-систем і програм для розробки модулів системи збирання та аналізування даних для спеціальних задач розглянемо такі середовища розробки: Atom, Visual Studio Code, Sublime Text 3, WebStorm.

Atom – це редактор, створений командою GitHub Inc. Він кроссплатформенний, безкоштовний і має вбудовану інтеграцію з GitHub. Його можна використовувати як самостійний повноцінний інструмент розробки, так і як допоміжний модуль для побудови веб-продуктів [20]. Приклад його роботи показано на рисунку 3.2.

Рисунок 3.2 – Знімок екрану робочої області середовища розробки Atom

Переваги включають:

1) Сучасний інтерфейс.
2) Інтеграція з GitHub.
3) Безкоштовний.

Недоліки:

1) Невдале підсвічування шрифту.
2) Помилки відображаються лише у новому вікні.

Visual Studio Code – один із найпопулярніших на сьогоднішній день текстових редакторів, розроблений Microsoft, є безкоштовним та кросплатформним. Редактор включає інструменти для роботи з GitHub, рефакторинги і деструктори, автодоповнення текстової структури і контекстні підказки. Приклад роботи з цим редактором показано на рисунку 3.3.

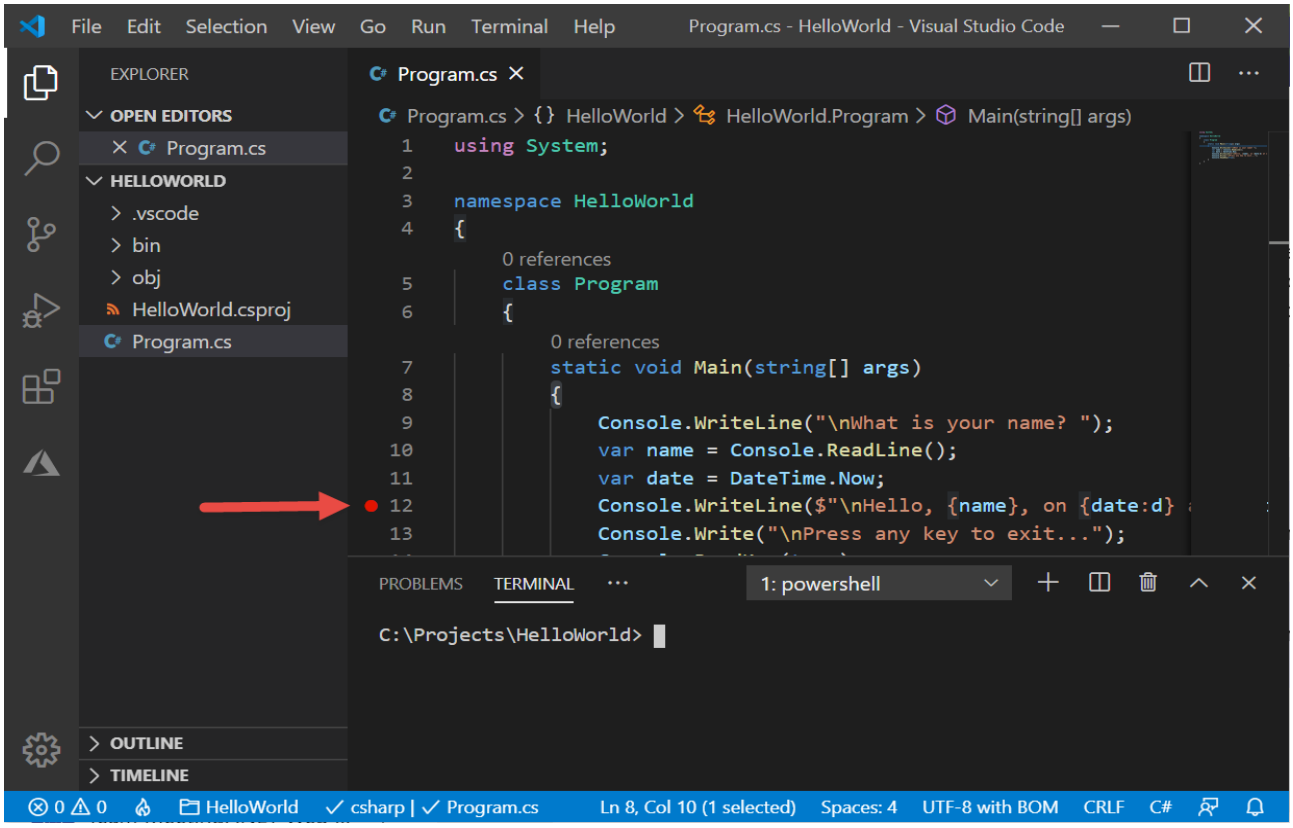


Рисунок 3.3 – Знімок екрану робочої області середовища розробки Visual Studio Code

VS Code до переваг віднесемо розширювану бібліотеку готових рішень та можливість роботи з багатьма видами веб-розробки. Недоліком може стати необхідність відносно великого об'єму займаних ресурсів комп'ютера.

Sublime Text 3 – умовно безкоштовний легкий кроссплатформенний текстовий редактор. Це один із найстаріших редакторів, створених до епохи редакторів для веб-технологій, таких як Atom та VS Code. Він досі підтримується до цього дня і був розроблений Sublime HQ. Редактор підтримує множину розширень, написаних на Python [21]. Приклад роботи у вікні програми показано на рисунку 3.4.

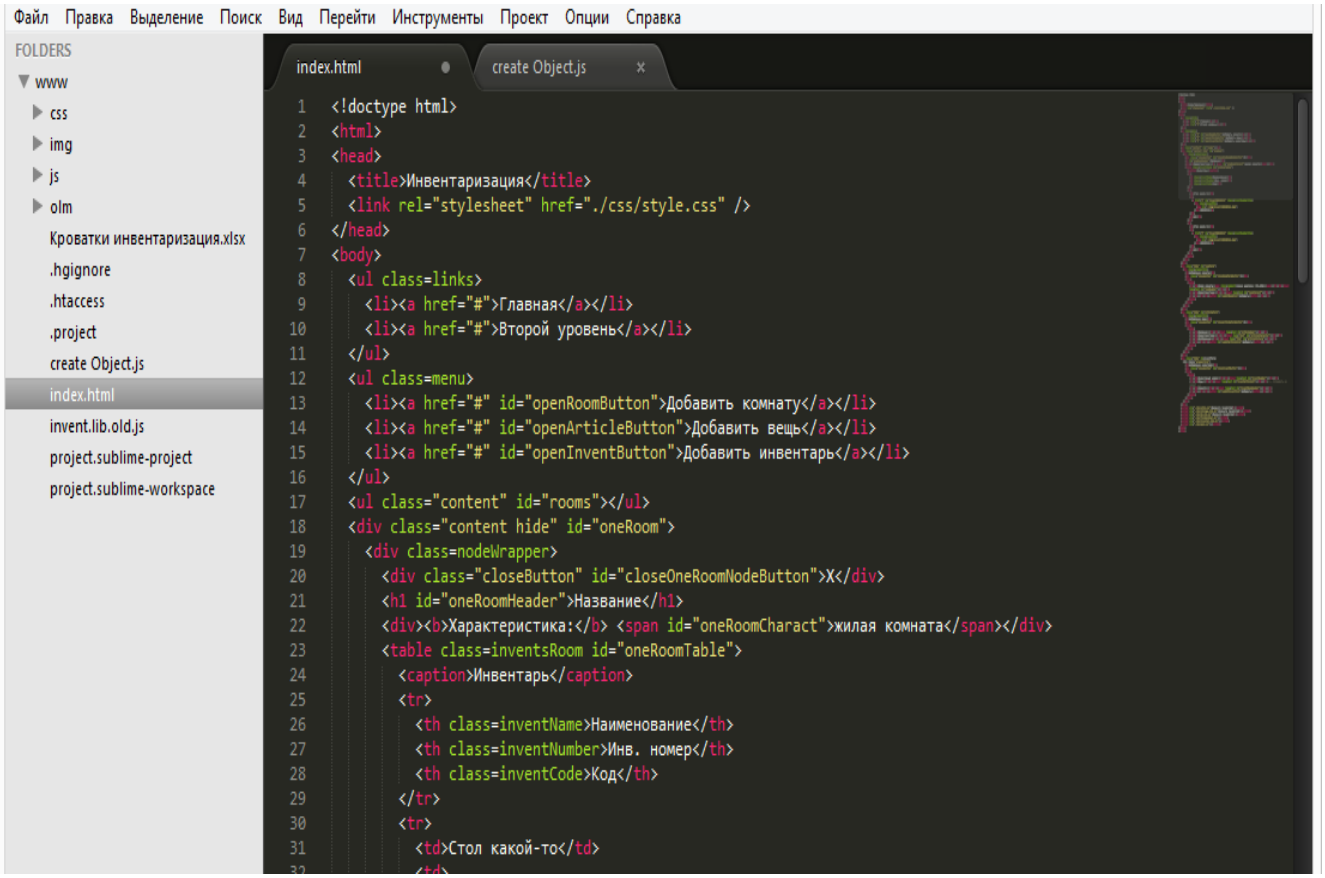


Рисунок 3.4 – Знімок екрану робочої області середовища Sublime Text 3

Переваги Sublime Text 3:

1) Займає мало ресурсів комп’ютера.
2) Кросплатформова сумісність.

3) Швидкість роботи.

Недоліки:

1) Застаріння використовуваних технологій.
2) Умовно безкоштовно (регулярні нагадування про ліцензійні вимоги).

WebStorm – редактор, створений JetBrains. Повноцінний інструмент для розробки веб-сайтів, підтримує безліч плагінів та розширень, не потребує багато часу для вивчення, також підтримує контроль версій, має налагоджувач коду та контекстні підказки про помилки. Приклад роботи з цим редактором показано на рисунку 3.5.

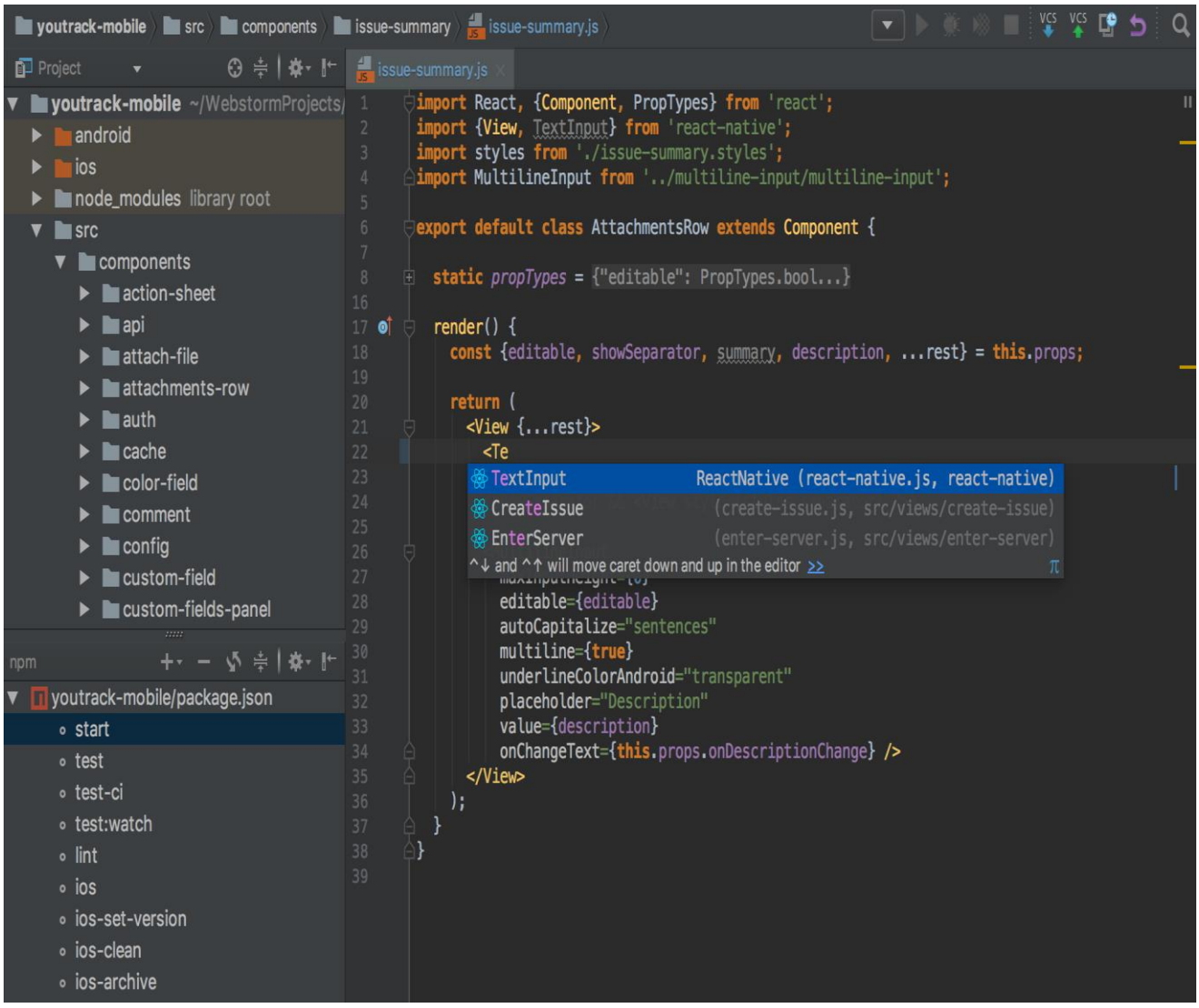


Рисунок 3.5 – Знімок екрану робочої області середовища розробки WebStorm

Переваги цього редактора:

1) Підтримка контролю версії.
2) Підсвічування коду.
3) Зручний формат коммітів (збереження прогресу).

Недоліки:

1) Вага (займає багато ресурсів ПК).
2) Не працює автозаповнення під час індексації файлів.

На основі поведеного аналізу створено структуровану таблицю 3.2 порівняльного аналізу середовищ розробки.

Таблиця 3.2 – Порівняння середовищ програмування

Функції	Середовище розробки (редактор)			
	Atom	Sublime Text 3	WebStorm	Visual Studio Code
Безкоштовність	+	+/-	-	+
Зручна підсвітка коду	-	-	+	+
Легковаговість	+	+	-	+/-+
Кросплатформеність	+	+	+	+
Підтримка контролю версій, інтегрований GitHub	+	-	+	+
Простота освоєння	+	+	-	+
Підтримка розширень	+	-	+	+

Керуючись проведеним аналізом середовищ, було обрано Visual Studio Code, як найбільш доцільний засіб розробки підсистеми аналізу захищеної системи збирання та аналізування даних для спеціальних задач. Це пов'язано з

тим, що він відповідає всім критеріям, необхідним для розробки модулів підсистеми. Середовище відрізняється потужною базою розширень, плагінів та бібліотек для веб-розробки. Ще одним важливим аспектом є те, що редактор є безкоштовним та підтримує систему контролю версій GitHub. Отже, враховуючи все сказане вище, можна однозначно сказати, що Visual Studio Code є кращим середовищем розробки серед конкурентів.

3.2 Розробка модуля автентифікації користувача

Система збирання та аналізування даних для спеціальних задач має бути реалізована на базі веб-ресурсу під назвою «Рєстр випадків домашнього насилля» на прохання представників головного управління національної поліції Вінницької області. Оскільки в сучасному світі веб-ресурси державних структур є цілями для атак збоку різних зловмисників дана система має бути захищеною від таких загроз.

Враховуючи, що збирання даних передбачається за допомогою інтернет-форм можна передбачити, що зросте загроза DDoS-атак. Для запобігання загрозам такого типу передбачено модуль авторизації який реалізовано як на бекенді та і фронтенді сайту. Для автентифікації користувача передбачено дві форми, одна відповідає за регістрацію нового, друга – за вхід у систему вже існуючого.

Для початку роботи нам потрібно винести посилання на бекенд в окремий файл, щоб надалі можна було його імпортувати в різні файли де будемо проводити запити для роботи з базою даних:

```
const API_URL = 'https://c04f-195-114-144-225.eu.ngrok.io/';
export default API_URL;
```

Також відразу використаємо експортування за замовчуванням, щоб мати доступ з інших модулів до даної змінної.

Усі дані кодуються за допомогою бібліотеки crc32, яка дає змогу хешувати дані для їх безпечного надсилання через мережу Інтернет. Щоб використати її необхідно імпортувати її модуль та використати метод str(). Для пояснення принципу дії використаємо просту стрічку «Hello, world»:

```
import crc32 from 'crc-32';
const data = 'Hi there!';
const hash = crc32.str(data);
```

У відповідь отримаємо значення -337197338

Модуль логіну користувача починається з імпортування URL змінної та створення слухача подій кнопки «Вхід»:

```
import API_URL from '../api'
function login(){
$('#login-btn').on('click', function(e){})
}
```

Далі збираємо дані з полів форми та якщо вони не пусті створюємо запит на бекенд для входу в систему:

```
let loginRequest = $.ajax({
type: "POST",
url: `${API_URL}users/login`,
crossDomain: true,
data: JSON.stringify(data),
})
```

Надсилаємо зібрані дані конвертуючи їх в JSON формат та отримуємо два сценарії запиту успіх (success), та помилка (error):

```
success: function(data){
sessionStorage.setItem("userId", data.userId);
sessionStorage.setItem("token", data.token);
window.location.replace('/index.html');},
```

При успішному виконанні запиту як можна побачити в коді наведеному вище ми записуємо отримані токен користувача та його ідентифікатор в пам’ять сесії браузера, та переводимо користувача на головну сторінку сайту. Токен –

це стрічка ідентифікатор у якій закодована хешфункціями інформація про користувача. Ідентифікатор користувача це звичайне ціле число але воно є унікальним. При невдалому запиті виводимо помилку в консоль:

```
error: function (data) {console.log(data);}
```

Для використання модуля за межами даного файлу також проекспортуємо дану функцію «login» за допомогою команди:

```
export default login;
```

Принцип регістрації користувача такий самий як і входу до системи, але має дещо більше полів, так як нам необхідно зібрати особисту інформацію користувача. Отже за наступним принципом проходимо по полях форми та збираємо значення:

```
let firstname   = $('#register-firstname').val();
```

Далі записуємо всі отримані змінні в об’єкт даних:

```
let data =  {
"firstName": firstname,
"middleName": surname,…}
```

Потім вже відомим нам чином створюємо подібний до логіну запит на регістрацію з типом «POST» так як ми творюємо нового користувача:

```
let registerRequest = $.ajax({
type: "POST",
url: `${API_URL}users/register`,
crossDomain: true,
data: JSON.stringify(data),})
```

Не забуваємо вказувати в url нашу змінну API_URL, поле crossDomain слугує для того щоб було дозволено проводити запити між різними доменами, а також варто зазначити, що для того аби запити проходили успішно до заголовків додаємо двінаступні стрічки:

```
'Access-Control-Allow-Origin': '*',
'Content-Type': 'application/json'
```

Перший рядок повідомляє, що дозволено всі кросдоменні запити, друга – те що ми надсилаємо дані саме в JSON форматі. Обробка успішної відповіді на запит така ж як і в попередньому запиті на логін але з відмінністю що нам потрібно зачистити поля наступним чином:

```
$('#register-firstname').val('');
```

Модуль автентифікації реалізований за допомогою видачі унікальних токену та ідинтифікатора за допомогою яких користувач має можливість взаємодіяти інформацією бази даних, та виконувати низку запитів, використовуючи модулі збирання та аналізування даних підсистеми аналізу.

3.3 Розробка модуля збирання даних

Модуль збирання даних має розгалужену структуру файлів з функціями обробки різних інтернет-форм системи. До нього входять функції створення та редагування інформації про дійових осіб та інцидентів домашнього насилля. Від так розглянемо спочатку обробку форми додавання особи:

```
function addNewPerson(){
$('.add-new-person-btn').on('click', function(e){
let userToken = sessionStorage.getItem('token');
```

Для початку створюємо прослухолвувач події натискання кнопки відправки форми та записуємо токен залогіненого користувача в змінну userToken. Ця маніпуляція потрібна для того, щоб при створенні запиту недопустити незалогіненого користувача до роботи з бекендом. Далі потрібно зібрати дані з полів форми про створювану особу і приводити поля до потрібних форматів яких вимагає бекенд. Відтак усі дати які відсилаємо маємо привести до стандарту ISO 8601, в мові JS за це відповідає функція toISOString, отже змінна дати народження створюваної особи буде виглядати наступним чином:

```
let birthDate = new Date($('#newperson-birthday').val());
birthDate.toISOString()
```

Наступним кроком буде перевірка наявності токена користувача та створення запиту для додавання нової особи:

```
if(userToken){
let createPersonRequest = $.ajax({
type: "POST",
url: `${API_URL}persons`,
crossDomain: true,
data: JSON.stringify(data),})
}
```

Усі запити, що не стосуються логіну і регістрації користувача, вимагають наявності токену, який вказується в заголовках і має наступний вигляд:

```
headers: {
'Access-Control-Allow-Origin': '*',
'Content-Type': 'application/json',
'Authorization': `Bearer ${userToken}`},
```

В полі Authorization вказуємо шаблонний рядок з початком «Bearer» та додаємо токен залогіненого користувача. Таким чином, це дає змогу проводити запити на бекенд і він дає дозвіл на взаємодію з даними.

Створення запису про інцидент проводиться за допомогою відповідних інтернет-форм. Ці форми мають два вити які є взаємопов'язані – форма самого інциденту та форма оцінювання ризику. Друга прикладається до першої. Форма оцінювання ризику спрямована на оцінку загрози життю постраждалої особи з боку особи кривдника. Оцінка вираховується автоматично при заповненні відповідних полів працівником поліції, для того щоб її зберегти обов'язковою є умова заповнення першої форми інциденту. При заповненні форми інциденту в ній реалізовано функціонал для додавання великої кількості дійових осіб. Алгоритм створення інцидентів такий самий як і додавання нової особи, збір значень полів, формування запиту, отримання успішної відповіді про створення з бекенду та очистка форми від попередніх значень.

Перегляд усіх інцидентів відбувається на відповідних сторінках веб-ресурсу для цього спочатку завантажується список інцидентів і далі конвертується в розмітку за допомогою циклу з шаблоном розмітки за умови, що користувач залогінений.

```
data.map((item) => {
const tplStr = `<tr data-record-id="${id}"><td>${id}</td>
…</tr>`;
$('.my-info-table').append(tplStr);
```

У відповіді з бекенду маємо масив data який ми ітеруємо за допомогою функції map() та підставляємо отриманий шаблон в таблицю my-info-table.

Отже, модуль збирання даних має розгалужену архітектуру, яка складається з функцій створення, перегляду та редагування інцидентів домашнього насилля та дійових осіб цих подій.

3.4 Розробка модуля аналізування

Цей модуль використовує математичні функції для обрахування статистики пов'язаної з інцидентами, що були внесені до бази даних. Він відповідає за показання таких даних: скільки отримано звернень за місяць, квартал чи рік, скільки було прийнято рішень, середній час розгляду звернення. Також враховує чи своєчасно було зроблено кожен з вище описаних пунктів. Для того, щоб почати обробку статистики, спочатку модуль забирає інформацію про всі інциденти:

```
let allIcidentsRequest = $.ajax({
type: "GET",
url: `${API_URL}incidents`,
success: function(data){
data.map((item) => {…})}});
```

Далі приходимо по масиву та забираємо дату створення інциденту та перевіряємо в яких часових рамках вона знаходиться, для того аби розподілити інциденти по місяцях та кварталах:

```
if(creationDate >= monthStart && creationDate <= monthEnd){
monthInc ++;
…
}
```

Додаємо кількість інцидентів за три місяці та отримуємо кількість звернень за квартал, та аналогічно отримуємо кількість за рік. Далі перевіряємо чи своєчасно було надано допомогу. Для цього ми маємо ліміт днів який не повинен перевищуватися різницею кількості днів від дати регістрації інциденту до дати її виконання:

```
let currentIncExecDays = executionDate – creationDate;
if(currentIncExecDays > executionLimit){
goodJobInc = false;
return goodJobInc;}
```

У цій частині функції повертається негативне значення goodJobInc якщо було перевищено ліміт часу на своєчасне виконання надання послуг конкретного інциденту.

Наступним кроком потрібно розглянути скільки було прийнято рішень. Цей критерій відповідає за те, що та чи інша послуга була надана під час розгляду інциденту. Отже якщо інциденту присвоєно дату виконання executionDate то це автоматично свідчить про те що рішення було прийняте. Тому щоб підрахувати кількість прийнятих рішень необхідно підрахувати кількість дат виконання та розподілити їх за часовими проміжками місяць, квартал, рік описаним вище способом:

```
if(executionDate >= monthStart && executionDate <= monthEnd){
executedMonthInc ++;
…}
```

Також необхідно підрахувати скільки часу в середньому йде на розгляд інциденту. Це час який відповідає різниці дати регістрації інциденту в базі та дати коли статус «необроблено» цього інциденту змінюється на статус «в обробці». І відбувається ця зміна коли працівник соціальної служби переглядає запис про конкретний інцидент. Вираховується цей параметр наступним чином:

```
let allConsiderTime = data.reduce((accumulator, item) =>{
let considerTime = item.statusChangeDate – item.creationDate;
return accumulator + considerTime;
}, initValue);
```

Тепер коли ми маємо значення яке містить дні за час розгляду усіх інцидентів просто знайдемо середнє арифметичне цього значення та отримаємо середній час розгляду:

```
let averageConsider = allConsiderTime / data.length;
```

Таким чином знаходимо третій показник для таблиці статистики. Всі дані вносяться в таблицю за допомогою шаблону розмітки подібним чином, як це відбувається в вищеописаному модулі збирання даних.

3.5 Розробка модуля оцінювання ризиків

Модуль оцінки ризиків було розроблено для того щоб оцінити стан інформаційної безпеки системи збирання та аналізування даних для спеціальних задач. Цей модуль розміщено на окремій сторінці, на якій спеціаліст з кібербезпеки може самостійно ввести вхідні данні для запуску алгоритму оцінки ризиків.

Для почату експерт має ввести фактори та їх значимість, наприклад: F1: Загроза DDoS-атаки – значення 1.2, і наступні загрози зі значеннями наведеними в таблиці 1 другого розділу. Вводимо ці значення для тесту щоб побачити чи буде результат обчилення таким як і в теоритичній частині.

Ці значення записуються в масив:

```
let factorsArr = [];
$('#risk-factor').each(function (){
    factorsArr.push(Number($(this).val()));
});
```

Масив цих значень відповідає множині факторів F описаних у другому розділі. Далі необхідно визначити їх кількість N, зробити це легко адже ми маэмо довжину масиву:

```
let numberOfValues = factorsArr.length;
```

Далі необхідно визначити P_i за формулою (2) наведеною в другому розділі:

```
let maxEntVal = 1/ numberOfValues;
```

Необхідно визначити класифікацію факторів F відносно діапазонів підмножини M_i, якщо показник фактора потрапляє у вибраний діапазон то йому присвоюється значення 1, якщо ні – 0:

```
const MFactorArr = [0.2, 0.16, 0.84, 1];
let lambdaFactVal = [];
for(let i = 0; i < numberOfValues; i++){
   if(factorsArr[i] < MFactorArr[1])  {lambdaFactVal.push(1);} else
lambdaFactVal.push(0);…}
```

Тепер маючи набір значень λ_{ij} (масив lambdaFactVal), можемо перейти до наступних обчислень. Так як підмножина значень B розподілена в діапазонах

які є в межах від 0 до 1 кожен з яких відповідає своєму показнику степені захищеності системи можемо обчислити b_i за формулою (4):

```
let bFactVal = [];
for(num in numberOfValues){
   let curBFactVal = 0.9 - 0.2 * (num - 1);
   bFactVal.push(curBFactVal);
}
```

Далі обчислимо оцінку ступеня ризику b за формулою (3), що наведена у другому розділі:

```
let sum1;
for(let i = 0; i < numberOfValues; i++){
   sum1 = maxEntVal * lambdaFactVal[i];
};
let sum2 = bFactVal.reduce( (curBFactVal, acc) => { acc+= curBFactVal});
let BRiskLevel = sum1 * sum2;
console.log(BRiskLevel);
```

При виводенні в консоль обчислення BRiskLevel бачимо, що результат збігається з обчисленим у другому розділі (рис. 3.6).

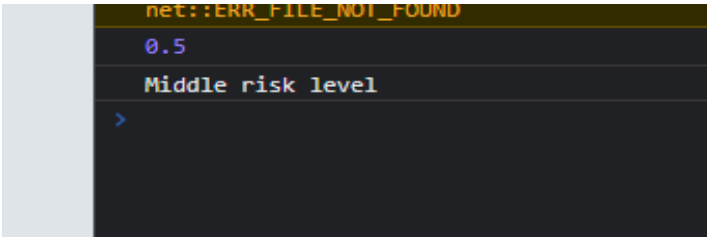


Рисунок 3.6 – Результат обчислення оцінки ступеня ризику b

Отже, в даному розділі було розроблено чотири модулі підсистеми аналізу системи збирання та аналізування даних для спеціальних задач, що розроблялася на базі веб-ресурсу під назвою «Реєстр випадків домашнього насилля». Проведено порівняльний аналіз мов розробки та визначено JavaScript, як найдоцільнішу. Розроблено модуль автентифікації користувача системи, модуль збирання даних за допомогою інтернет-форм, модуль аналізування внесених даних для відображення статистичної інформації про інциденти, та модуль для оцінювання факторів впливу кіберзагроз на стан інформаційної безпеки системи.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Тестування роботи програмного засобу

Для того, щоб протестувати підсистему аналізу захищеної системи збирання та аналізування даних для спеціальних задач, яка є реалізованою за допомогою веб-технологій необхідно розглянути існуючі види тестування. Тестування веб-ресурсу має підхід, який називається QA (забезпечення якості). За такого підходу система перевіряє відповідність умовам експлуатації, заданим у технічному завданні. QC (контроль якості) метою, якого є виявлення вразливостей у програмному коді системи. На практиці, щоб сайт був безпечним та надійним, вільним від уразливостей і помилок, які можуть призвести до катастрофічних збоїв або розкриття конфіденційної інформації. Важливо, щоб сайт регулярно тестувався та виявлялися будь-які проблеми та було реалізовано їх усунення.

Веб-ресурс, який необхідно протестувати зображено на рисунку 4.1.

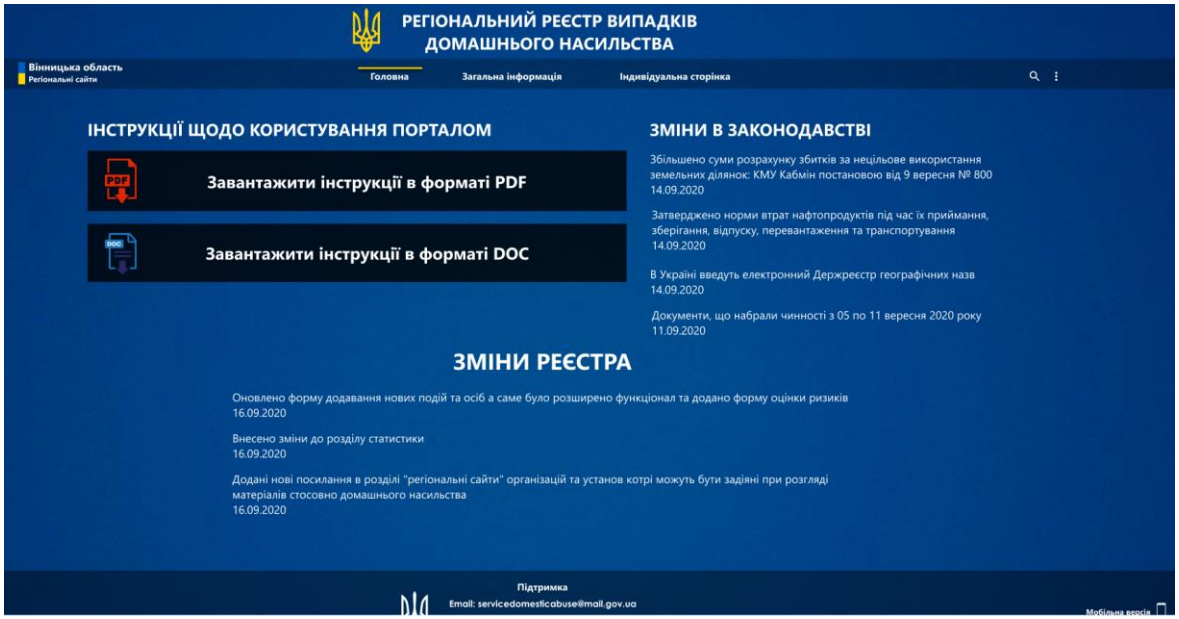


Рисунок 4.1 – Вигляд головної сторінки сайту «Реєстр випадків домашнього насилля»

На головній сторінці відображено посилання з інструкціями та секціями новин та змін в законодавстві. Варто зазначити що, для переходу на інші сторінки при умові що користувач на ввійшов у систему буде виведено відповідне повідомлення (рис. 4.2).

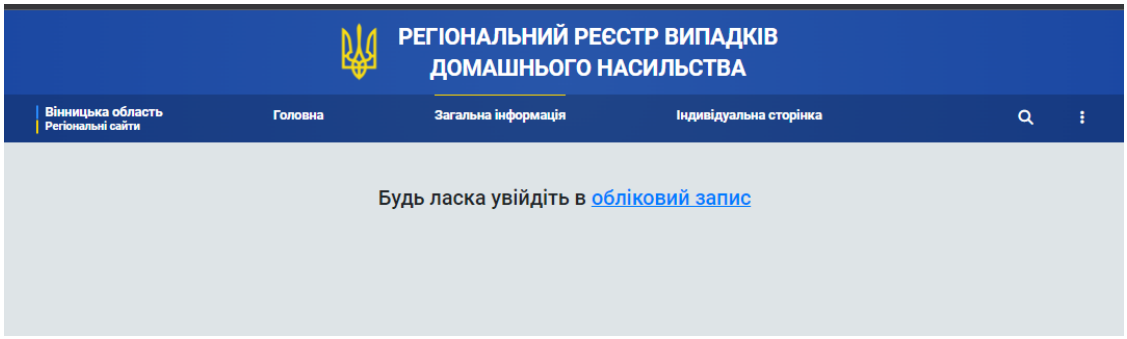


Рисунок 4.2 – Повідомлення про відсутність автентифікації користувача

Для того щоб ввійти а обліковий запис натискаємо на відповідне посилання та потрапляємо на сторінку з формою для входу в систему, з якої також можна перейти на сторінку з формою регістрації за умови якщо в нас немає власного акаутну (рис. 4.3).

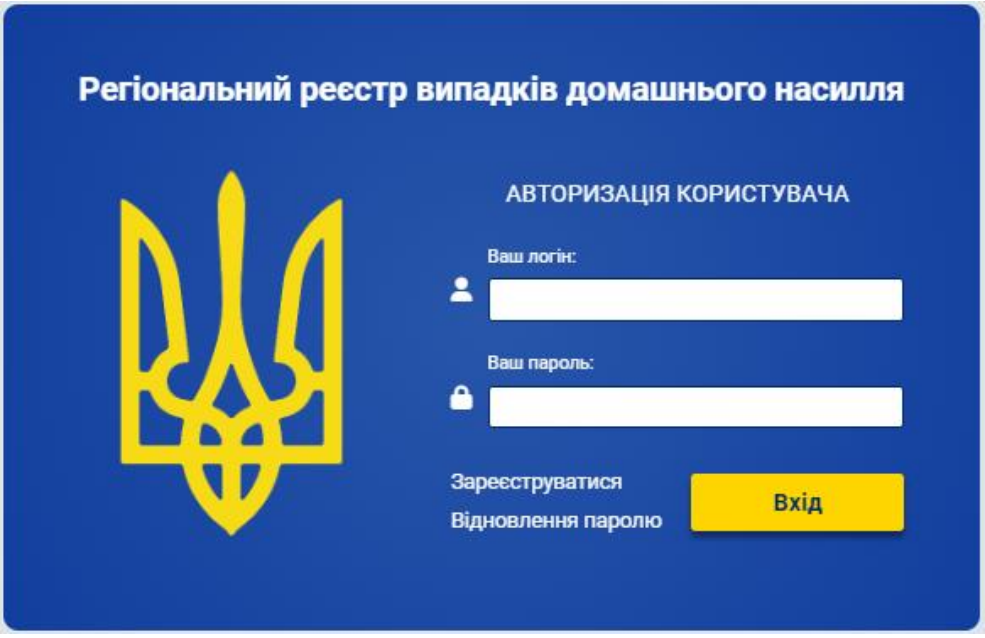


Рисунок 4.3 – Форма для входу в систему

Для регістрації нового користувача натискаємо на посилання «Зареєструватися» і переходимо на відповідну форму реєстрації (рис. 4.4).

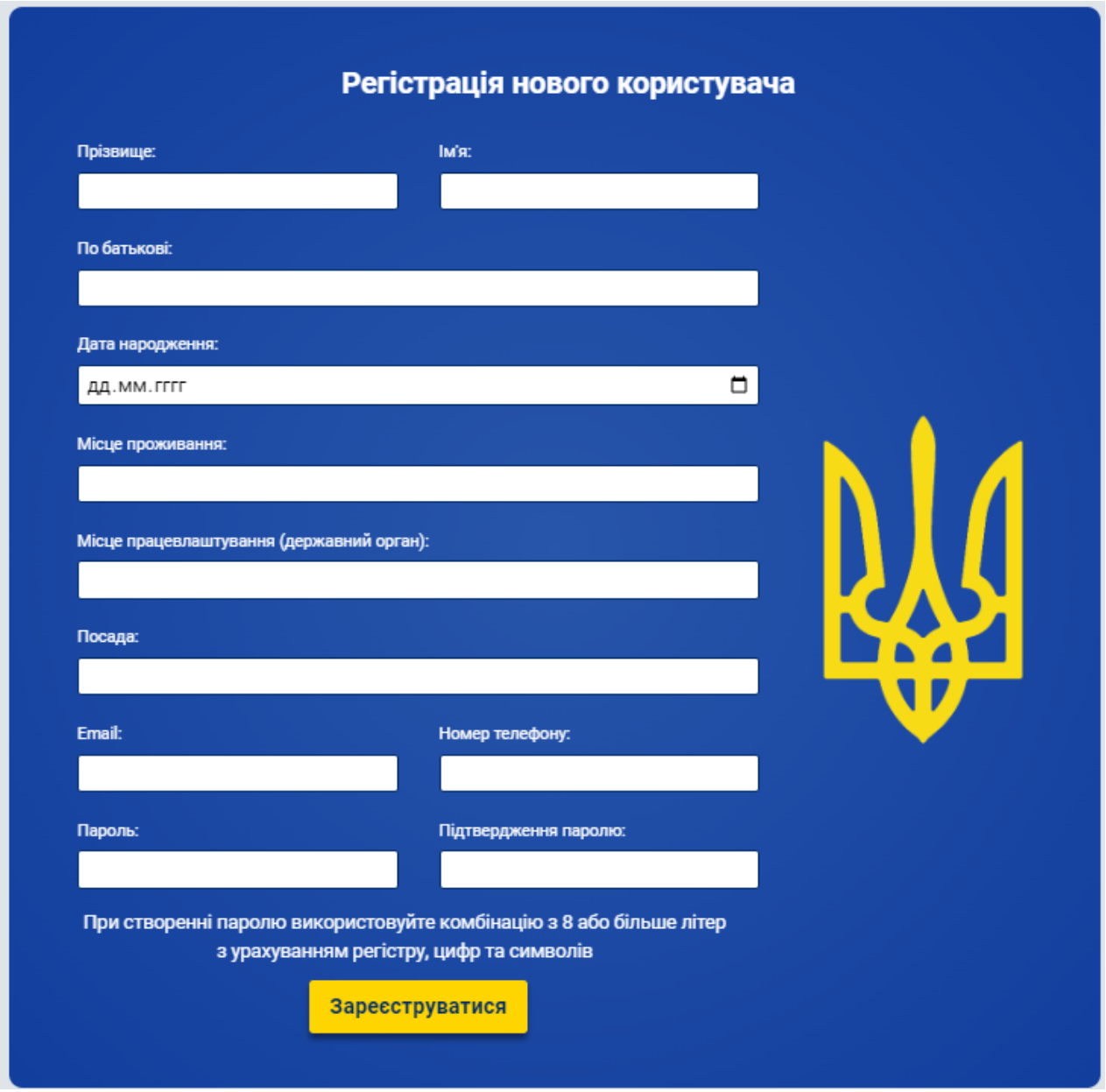


Рисунок 4.4 – Форма реєстрації нового користувача

При реєстрації та вхід в систему відбувається автоматично і в сховище сесії зберігаються дані з токеном та ідентифікатором користувача, що зображено на рисунку 4.5.

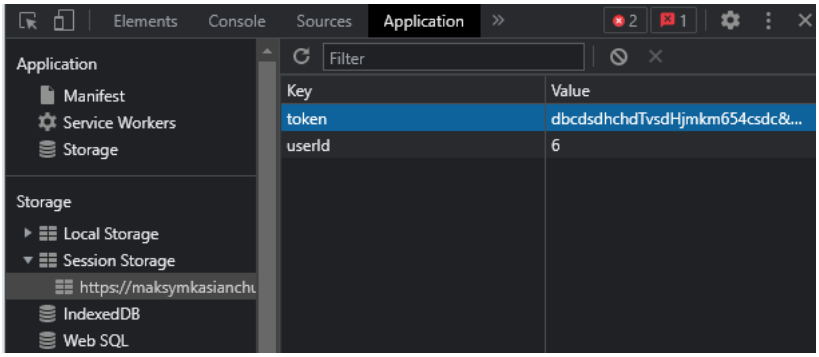


Рисунок 4.5 – Токен та ідентифікатор користувача збережені в пам’яті сесії браузера

Надалі при запитах на бекенд системи ці дані будуть часто використовуватись та свідчити про те зо користувач ввійшов у обліуовий запис системи.

На сторінці «загальна інформація» знаходяться всі записи про внесені до бази даних інциденти, а на сторінці «індивідуальна сторінка» відображено записи, які були внесені до бази саме активним цим користувачем (рис. 4.6).

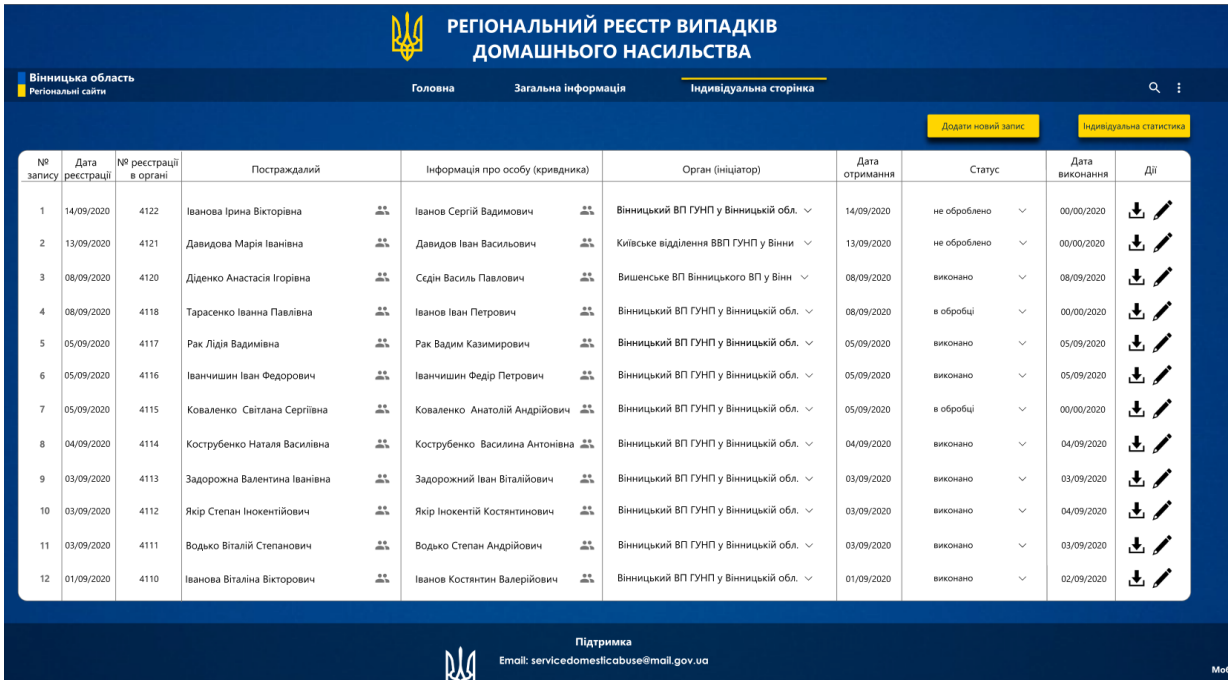


Рисунок 4.6 – Індивідуальна сторінка з записами активного користувача

При виборі імені особи в таблиці можна переглянути повну інформацію про дану особу в формі під назвою «Інформаційна картка особи», що зображена на рисунку 4.7:

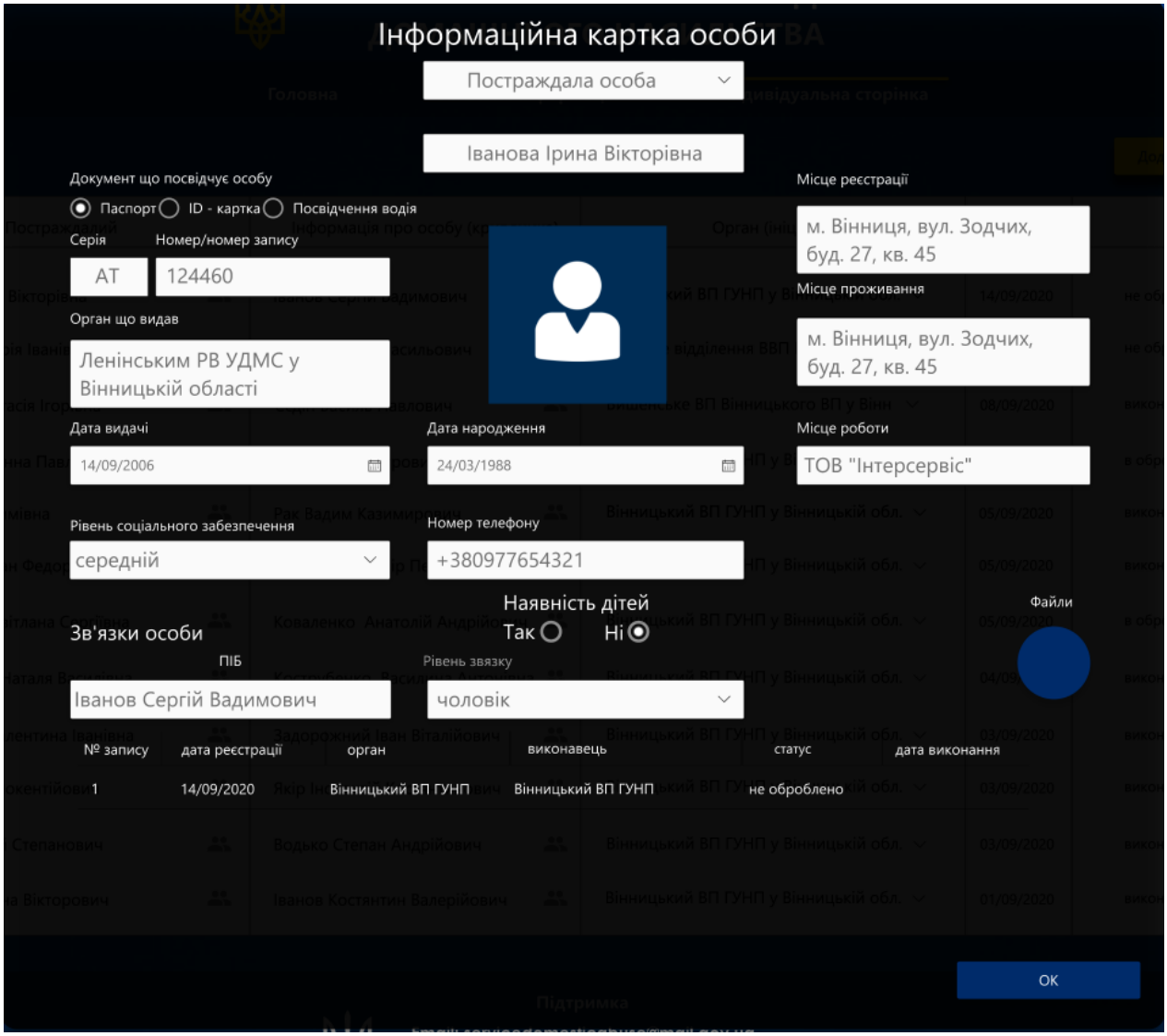


Рисунок 4.7 – Інформаційна картка особи з персональними даними

Аналогічна інформаційна картка створена і для детального опису інциденту, при натисканні кнопки із зображенням олівця на таблиці записів можна переглянути детальну інформацію про ту чи іншу подію внесену до бази даних (рис. 4.8).

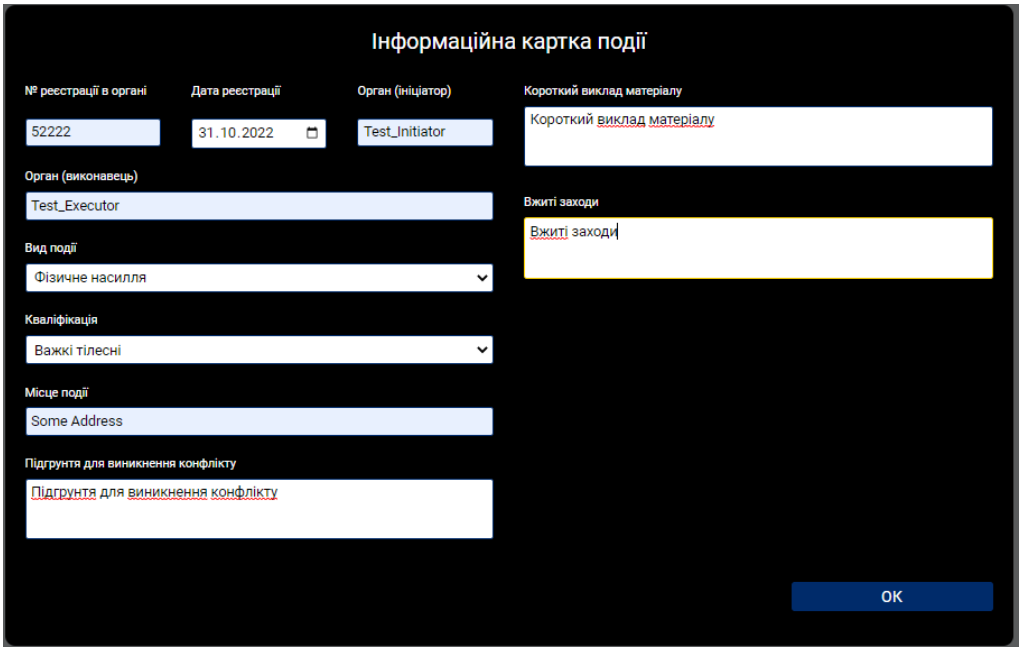


Рисунок 4.8 – Інформаційна картка події

Для додавання нових осіб до бази даних необхідно перейти на відповідну сторінку «add-person» та заповнити форму з персональною інформацією особи (рис. 4.9).

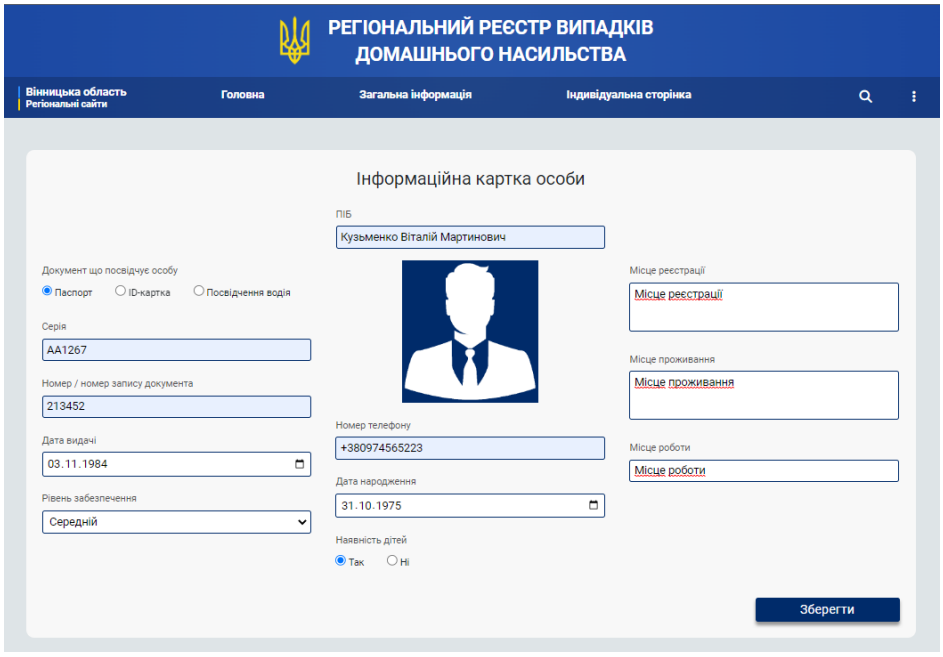


Рисунок 4.9 – Форма для створення нової особи

Для додавання нового запису про інцидент реалізовано форму зображену на рисунку 4.10. У формі реалізовано «живий пошук» осіб, що занесені до бази даних. В одному записі можна додавати будь-яку кількість постраждалих та кривдників.

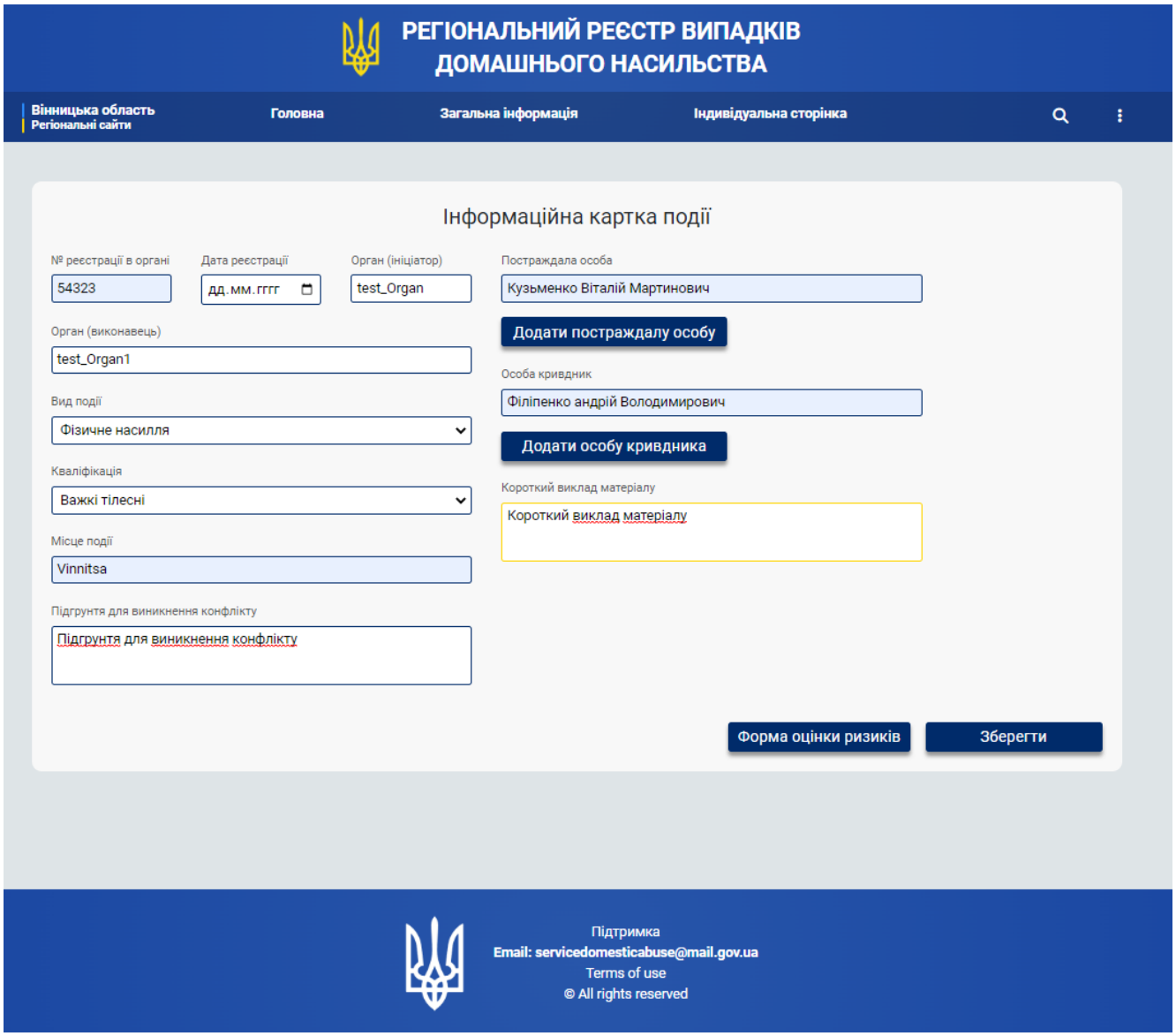


Рисунок 4.10 – Форма для створення нового запису про інцидент

З вище наведеної форми можна відкрити форму оцінки ризиків (рис. 4.11), натиснувши відповідну кнопку. Важливо зазначити, що зберегти форму оцінки ризиків неможливо, якщо не заповнити форму «Інформаційна картка події».

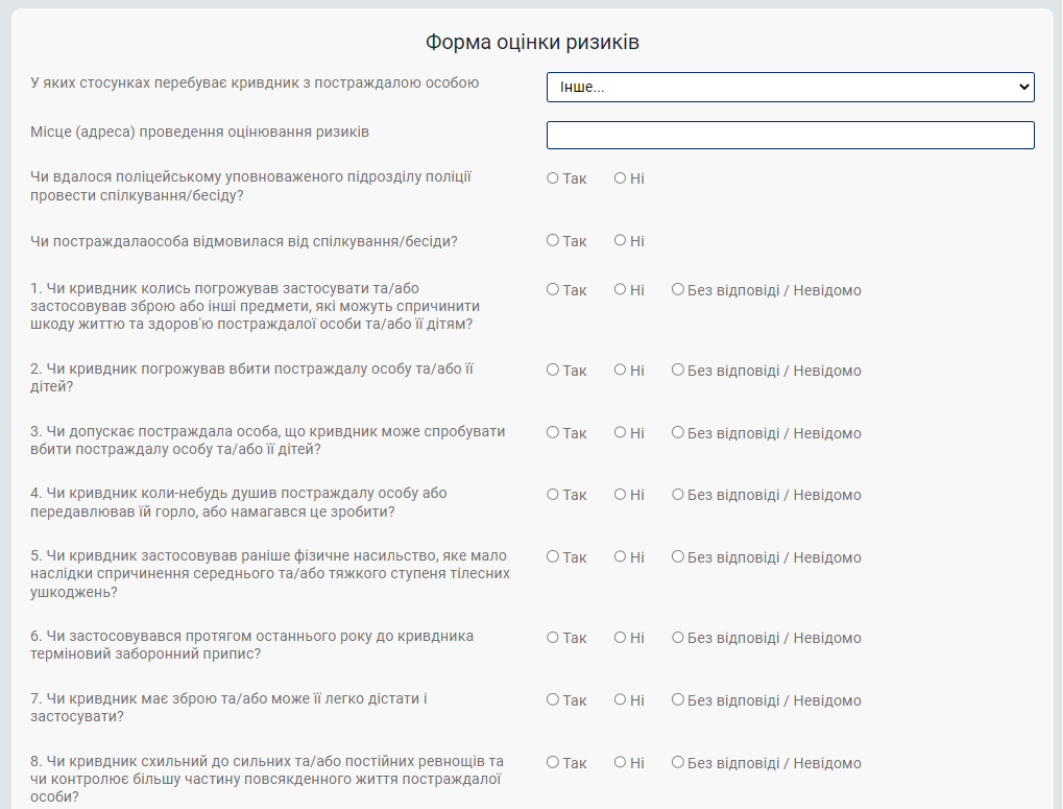


Рисунок 4.11 – Форма оцінки ризиків інциденту

Після додавання особи та запису про подію до бази даних переходимо на індивідуальну сторінку та бачимо новостворений запис (рис. 4.12).

Рисунок 4.12 – Результат додавання нового запису у таблицю інцидентів на індивідуальній сторінці

Модуль аналізування виводить статистичні дані на окрему сторінку, перейти на яку можна за допомогою кнопки «Індивідуальна статистика» на сторінці «Індивідуальна сторінка». Сторінка статистики відображає таблицю (рис. 4.13) та графік з даними про своєчасність надання та виконання послуг.

ІНДИВІДУАЛЬНА СТАТИСТИКА

НАЦІОНАЛЬНА ПОЛІЦІЯ

вибірка з до

	1	2	3	I кв.	4	5	6	II кв.	7	8	9	III кв.	10	11	12	IV кв.	Загалом за рік
отримано звернень	30	30	35	95	30	37	24	91	42	24	30	96	20	43	32	95	377
з них несвоєчасно	2	0	1	3	0	3	0	3	1	0	0	1	1	2	1	4	11
принято рішень	25	30	33	88	29	30	23	82	39	22	28	89	17	40	30	87	346
з них несвоєчасно	3	0	1	4	1	4	1	6	2	2	2	6	2	1	1	4	20
середній час розгляду (діб)	10	15	14	13	20	17	14	17	16	23	15	18	13	16	22	17	16,25
несвоєчасно розглянуті	5	0	2	7	1	7	1	9	3	2	2	7	3	3	2	8	31
результат виконання																	
складено протоколів	20	23	20	63	27	27	21	75	30	19	26	75	17	32	30	79	292
винесено ТЗП	18	20	17	55	27	27	21	75	30	19	26	75	16	32	30	78	283
Зареєстровано КП	5	7	13	25	2	3	2	7	6	2	3	11	0	5	0	5	48
ст. 126¹	3	2	10	15	2	2	0	4	4	2	1	7	0	3	0	3	29
ст. 390¹	2	5	3	10	0	1	2	3	2	0	2	4	0	2	0	2	19

Рисунок 4.13 – Таблиця індивідуальної статистики

Для того щоб модулю аналізування було що обраховувати в базу було завантажено тестовий перелік записів про інциденти. Дані виведені в таблицю на попередньому рисунку також конвертуються в графік на рисунку 4.14:

Рисунок 4.14 – Графік індивідуальної статистики

Для того щоб скористатися модулем оцінки ризиків кібербезпеки перейдемо на відповідну сторінку та введемо дані в запропоновані поля (рис. 4.15).

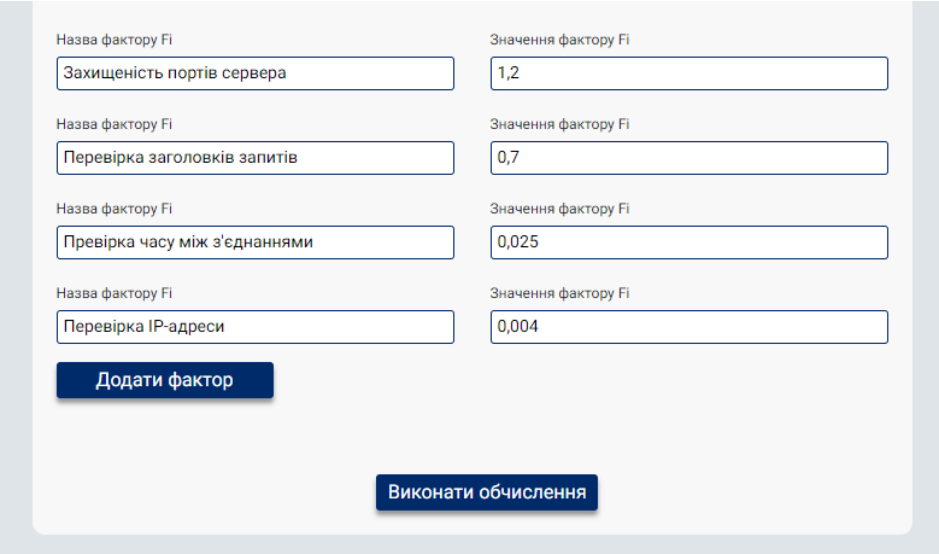


Рисунок 4.15 – Форма оцінювання ризиків кібербезпеки

Після чого натиснемо кнопку «Виконати обчислення» і модуль поверне повідомлення про ступінь обчислюваного кіберризику (рис. 4.16).

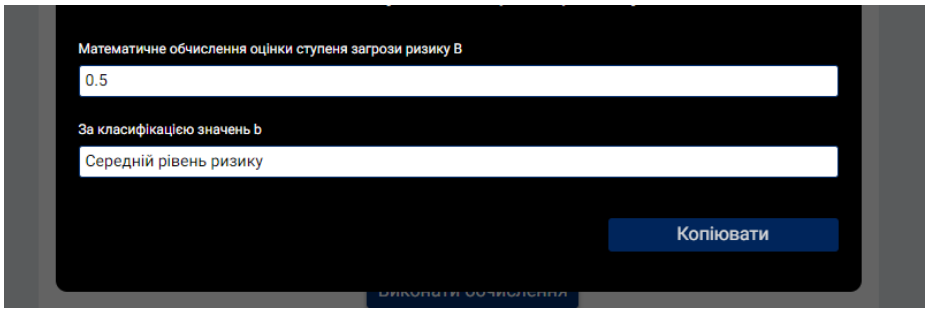


Рисунок 4.16 – Повідомлення про результат обчислення оцінювання ступеня ризику

Отже, з проведеного тестування роботи програмного засобу можна зробити висновок, що програмний засіб працює повністю коректно та виконує всі зазначені функції. Веб-ресурс має захист від несанкціонованого доступу у вигляді модуля авторизації. Також можна побачити, що модулі збирання та аналізування даних підсистеми аналізу працюють та виконують свої функції правильно. Модуль збирання даних проводить збір інформації за допомогою відповідних інтернет-форм, а модуль аналізування виводить статистичні дані на окрему сторінку, де користувач зможе контролювати якість надання та виконання послуг.

4.2 Оцінювання стану інформаційної безпеки системи

Проведемо тестування оцінювання ризиків ІБ розробленої системи за допомогою модуля оцінювання ризиків. Для цього визначимо фактори що впливають на оцінювання окремих ризиків та внесемо їх у відповідні таблиці. Далі проведемо оцінку кожного кіберризику і у підсумку оцінимо загальний стан інформаційної безпеки. Так наприклад створимо перелік факторів які впливають на загрозу ризику DDoS-атаки (табл. 4.1).

Таблиця 4.1 – Набір факторів що впливають на загрозу DDoS-атаки

Назва фактору	Значення фактору
Захищеність портів сервера	1.2
Перевірка заголовків запитів	0.7
Перевірка часу між з’єднаннями	0.025
Перевірка IP-адреси	0.004

Після внесення цих факторів до модуля через форму оцінювання ми отримали обчислений результат 0.5 – середній рівень ризику. Таким же чином зазначимо фактори які впливають на загрозу несанкціонованого доступу (табл. 4.2). Та внесемо ці фактори у форму оцінювання розробленого модуля.

Таблиця 4.2 – Набір факторів загрози несанкціонованого доступу

Назва показника	Значення показника
Хешування даних при відправці	0.9
Зберігання даних в сесії користувача	0.6
Відновлення паролю	0.4
Двофакторна автентифікація	0.2

Після обрахунку отримали результат 0.3 – низький ризик загрози. Далі необхідно повторити таку маніпуляцію і для оцінювання ризику загрози можливої спам-атаки, (табл. 4.3). Такий сценарій можливий якщо на інтернет-формі не буде застосовано інструменту під назвою CAPTCHA (автоматизований тест Тюринга для розрізнення користувачів машин та людей).

Таблиця 4.3 – Набір факторів загрози спам-атаки

Назва показника	Значення показника
Наявніть захисту Інтернет-форм	0.8
Обмеження часу між надсиланнями форм	0.5
Використання CAPTCHA	0.2

У результаті отримано відповідь 0.4 – що також відповідає низькому ризику загрози. Тепер можемо внести три отримані результати в Інтернет-форму модуля оцінювання ризиків (рис. 4.17).

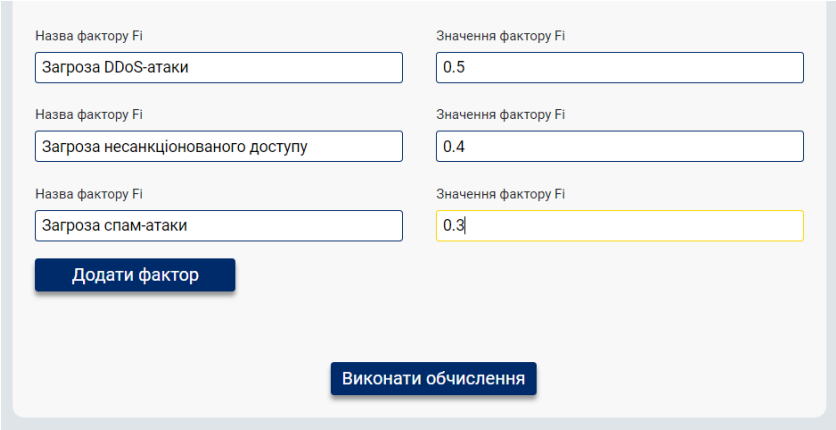


Рисунок 4.17 – Форма з внесеними загрозами оцінених ризиків

Далі запускаємо оцінювання загального стану ІБ системи. І в результаті отримуємо 0.45 – що відповідає середньому рівню загроз ІБ (рис 4.18).

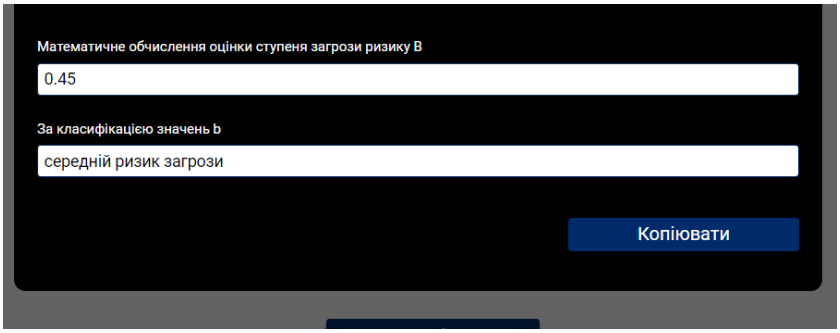


Рисунок 4.18 – Результат оцінювання загального стану ІБ системи

Таким чином, можемо побачити, що рівень загального стану ризиків інформаційній безпеці розроблюваної системи дорівнює 0.45 і це значення відповідає шкалі середнього рівня загроз. Тож можемо зробити висновок, що систему можна вдосконалити понизивши показники факторів оцінюваних ризиків, що в свою чергу призведе до покращення стану ІБ системи. Обраховані показники наочно зображають степінь ризиків, їх можна надати відділу розробки та відділу фінансів який зможе прогнозувати витрати на запобігання ризиків, тощо.

5 ЕКОНОМІЧНА ЧАСТИНА

Виконання науково-дослідної роботи завжди передбачає отримання певних результатів і вимагає відповідних витрат. Результати виконаної роботи завжди дають нам нові знання, які в подальшому можуть бути використані для удосконалення та/або розробки (побудови) нових, більш продуктивних зразків техніки, процесів та програмного забезпечення.

Дослідження на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» може бути віднесено до фундаментальних і пошукових наукових досліджень і спрямоване на вирішення наукових проблем, пов’язаних з практичним застосуванням. Основою таких досліджень є науковий ефект, який виражається в отриманні наукових результатів, які збільшують обсяг знань про природу, техніку та суспільство, які розвивають теоретичну базу в тому чи іншому науковому напрямку, що дозволяє виявити нові закономірності, які можуть використовуватися на практиці.

Для цього випадку виконаємо такі етапи робіт:

1) здійснимо проведення наукового аудиту досліджень, тобто встановлення їх наукового рівня та значимості;

2) проведемо планування витрат на проведення наукових досліджень;

3) здійснимо розрахунок рівня важливості наукового дослідження та перспективності, визначимо ефективність наукових досліджень.

5.1 Оцінювання наукового ефекту

Основними ознаками наукового ефекту науково-дослідної роботи є новизна роботи, рівень її теоретичного опрацювання, перспективність, рівень розповсюдження результатів, можливість реалізації. Науковий ефект НДР на тему «Захищена система збирання та аналізування даних для спеціальних задач.

Частина 1. Підсистема аналізу» можна охарактеризувати двома показниками: ступенем наукової новизни та рівнем теоретичного опрацювання.

Значення показників ступеня новизни і рівня теоретичного опрацювання науково-дослідної роботи в балах наведені в табл. 5.1 та 5.2.

Таблиця 5.1 – Показники ступеня новизни науково-дослідної роботи виставлені експертами

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ, посада)		
		1	2	3
Принципово нова	Робота якісно нова за постановкою задачі і ґрунтується на застосуванні оригінальних методів дослідження. Результати дослідження відкривають новий напрям в даній галузі науки і техніки. Отримані принципово нові факти, закономірності; розроблена нова теорія. Створено принципово новий пристрій, спосіб, метод	0	0	0
Нова	Отримана нова інформація, яка суттєво зменшує невизначеність наявних значень (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту). Проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів	54	60	59
Відносно нова	Робота має елементи новизни в постановці задачі і методах дослідження. Результати дослідження систематизують і узагальнюють наявну інформацію, визначають шляхи подальших досліджень; вперше знайдено зв’язок (або знайдено новий зв’язок) між явищами. В принципі відомі положення розповсюджені на велику кількість об’єктів, в результаті чого знайдено ефективне рішення. Розроблені більш прості способи для досягнення відомих результатів. Проведена часткова раціональна модифікація (з ознаками новизни)	0	0	0
Традиційна	Робота виконана за традиційною методикою. Результати дослідження мають інформаційний характер. Підтверджені або поставлені під сумнів відомі факти та твердження, які потребують перевірки. Знайдено новий варіант рішення, який не дає суттєвих переваг в порівнянні з існуючим	0	0	0
Не нова	Отримано результат, який раніше зафіксований в інформаційному полі, та не був відомий авторам	0	0	0
Середнє значення балів експертів		57,7		

Згідно отриманого середнього значення балів експертів ступінь новизни характеризується як нова, тобто отримана нова інформація, яка суттєво зменшує невизначеність наявних знань (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту) та проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів.

Таблиця 5.2 – Показники рівня теоретичного опрацювання науково-дослідної роботи виставлені експертами

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерт (ПІБ, посада)		
	1	2	3
Відкриття закону, розробка теорії	0	0	0
Глибоке опрацювання проблеми: багатоаспектний аналіз зв’язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу	67	68	65
Розробка способу (алгоритму, програми), пристрою, отримання нової речовини	0	0	0
Елементарний аналіз зв’язків між фактами та наявною гіпотезою, класифікація, практичні рекомендації для окремого випадку тощо	0	0	0
Опис окремих елементарних фактів, викладення досвіду, результатів спостережень, вимірювань тощо	0	0	0
Середнє значення балів експертів	66,7		

Згідно отриманого середнього значення балів експертів рівень теоретичного опрацювання науково-дослідної роботи характеризується як глибоке опрацювання проблеми: багатоаспектний аналіз зв’язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу.

Показник, який характеризує рівень наукового ефекту, визначаємо за формулою [22]:

$$E_{нау} = 0,6 \cdot k_{нов} + 0,4 \cdot k_{теор}, \quad (5.1)$$

де $k_{нов}, k_{теор}$ - показники ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи, $k_{нов} = 57,7, k_{теор} = 66,7$ балів;

0,6 та 0,4 – питома вага (значимість) показників ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи.

$$E_{нау} = 0,6 \cdot k_{нов} + 0,4 \cdot k_{теор} = 0,6 \cdot 57,7 + 0,4 \cdot 66,67 = 61,27 \text{ балів.}$$

Визначення характеристики показника $E_{нау}$ проводиться на основі висновків експертів виходячи з граничних значень, які наведені в табл. 5.3.

Таблиця 5.3 – Граничні значення показника наукового ефекту

Досягнутий рівень показника	Кількість балів
Високий	70…100
Середній	50…69
Достатній	15…49
Низький (помилкові дослідження)	1…14

Відповідно до визначеного рівня наукового ефекту проведеної науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу», даний рівень становить 61,27 балів і відповідає статусу - середній рівень. Тобто у даному випадку можна вести мову про потенційну фактичну ефективність науково-дослідної роботи.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.2.1 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 ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [22]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \qquad (5.2)$$

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.

$$З_о = 17400{,}00 \cdot 21 / 21 = 17400{,}00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.4.

Таблиця 5.4 – 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	17400,00	828,57	21	17400,00
Науковий співробітник	17150,00	816,67	12	9800,00
Інженер-програміст 1-ї категорії	17050,00	811,90	21	17050,00
Лаборант	6750,00	321,43	10	3214,29
Всього				47464,29

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (5.3)$$

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:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (5.4)$$

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

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [22];

$К_с$ – 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.

$Т_р$ – середнє число робочих днів в місяці, приблизно $Т_р$ = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$$С_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}65 / (21 \cdot 8) = 72{,}38 \text{ грн.}$$

$$З_{р1} = 72{,}38 \cdot 10{,}50 = 760{,}03 \text{ грн.}$$

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення допоміжного обладнання	10,50	2	1,10	72,38	760,03
Інсталяція програмного забезпечення	6,25	3	1,35	88,83	555,22

Продовження таблиці 5.5 – Величина витрат на основну заробітну плату робітників.

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення цифрових обчислювальних систем	5,15	5	1,70	111,87	576,11
Відлагодження програмних модулів аналізу даних	5,75	4	1,50	98,71	567,56
Підготовка дослідження	9,12	4	1,50	98,71	900,19
Формування бази даних результатів дослідження	16,00	2	1,10	72,38	1158,14
Всього					4517,25

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (5.5)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод} = (47464,29 + 4517,25) \cdot 11 / 100\% = 5717,97$ *грн.*

5.2.2 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:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad , \tag{5.6}$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (47464,29 + 4517,25 + 5717,97) · 22 / 100% = 12693,89 грн.

5.2.3 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 «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу».

Витрати на матеріали на даному етапі проведення досліджень в основному пов’язані з використанням моделей елементів та моделювання роботи і досліджень за допомогою комп’ютерної техніки та створення експериментальних математичних моделей або програмного забезпечення, тому дані витрати формуються на основі витратних матеріалів характерних для офісних робіт.

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

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{в\,j} \, , \tag{5.7}$$

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

n – кількість видів матеріалів;

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

K_j – коефіцієнт транспортних витрат, (K_j = *1,1 … 1,15*);

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

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 2,0 \cdot 225,00 \cdot 1,1 - 0 \cdot 0 = 495,00$ *грн.*

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – 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 канцелярський офісний (А4)	225,00	2,0	0	0	495,00
Папір для заміток (А5)	116,00	4,0	0	0	510,40
Начиння канцелярське	195,00	3,0	0	0	643,50
Органайзер офісний	183,00	3,0	0	0	603,90
Картридж для принтера	950,00	1,0	0	0	1045,00
Всього					3297,80

5.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($K_в$)*, які використовують при проведенні НДР на тему «Захищена система збирання та аналізування даних для спеціальних

задач. Частина 1. Підсистема аналізу», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \quad (4.8)$$

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

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

$К_j$ – коефіцієнт транспортних витрат, ($К_j = 1,1 \ldots 1,15$).

$К_в = 1 \cdot 3079,00 \cdot 1,1 = 3386,90$ *грн.*

Проведені розрахунки зведемо до таблиці 5.7.

Таблиця 5.7 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішній жорсткий диск 2.5" 2TB Seagate (STGD2000200)	1	3079,00	3386,90
Концентратор Defender SEPTIMA SLIM (83505)	1	400,00	440,00
Кабель для передачі даних USB to COM 1.0m Patron (CAB-PN-USB-COM)	1	354,00	389,40
Всього			4216,30

5.2.5 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:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (5.9)$$

де $Ц_i$ – 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…1,12);*

k – кількість найменувань устаткування.

$B_{спец} = 39460{,}00 \cdot 1 \cdot 1{,}1 = 43406{,}00$ *грн.*

Отримані результати зведемо до таблиці 5.8.

Таблиця 5.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Ноутбук ASUS RU451-UJ, оперативна пам'ять 4gb RAM, процесор intel core i5 2.5Ггц, Пам'ять на жорсткому диску 40mb	1	39460,00	43406,00
Всього			43406,00

5.2.6 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:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (5.10)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – 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…1,12);*

k – кількість найменувань програмних засобів.

$B_{прг}$ = *7910,00 · 1 · 1,1 = 8701,00 грн.*

Отримані результати зведемо до таблиці 5.9.

Таблиця 5.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне програмне забезпечення розробки системи аналізу	1	7910,00	8701,00
Всього			8701,00

5.2.7 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:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12} \ , \qquad (5.11)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

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

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (35830,00 \cdot 1) / (2 \cdot 12) = 1492,92$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 5.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-обчислювальний комплекс розробки системи аналізу даних	35830,00	2	1	1492,92
Місце оператора спеціалізоване	8200,00	5	1	136,67
Офісна оргтехніка	9600,00	4	1	200,00
Дослідницька лабораторія	500000,00	20	1	2083,33
Всього				3912,92

5.2.8 Паливо та енергія для науково-виробничих цілей

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

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (5.12)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

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

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

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

$B_e = 0,05 \cdot 200,0 \cdot 6,20 \cdot 0,95 / 0,97 = 62,00$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 5.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук ASUS RU451-UJ, оперативна пам'ять 4gb RAM, процесор intel core i5 2.5Ггц, Пам'ять на жорсткому диску 40mb	0,05	200,0	62,00
Програмно-обчислювальний комплекс розробки системи аналізу даних	0,42	200,0	520,80
Місце оператора спеціалізоване	0,10	200,0	124,00
Офісна оргтехніка	0,60	5,0	18,60
Всього			725,40

5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» 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» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (5.13)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$В_{св}$ = (47464,29 + 4517,25) · 20 / 100% = 10396,31 грн.

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

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

5.2.11 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:

$$І_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.14)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 70%.

$І_в$ = (47464,29 + 4517,25) · 70 / 100% = 36387,08 грн.

5.2.12 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:

$$B_{нзв} = (З_о + З_р) \cdot \frac{H_{нзв}}{100\%}\ , \qquad (5.15)$$

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

$B_{нзв} = (47464,29 + 4517,25) \cdot 100 / 100\% = 51981,54$ *грн.*

Витрати на проведення науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв}. \qquad (5.16)$$

$B_{заг}$= *47464,29 +4517,25 +5717,97 +12693,89118 +3297,80 +4216,30 +43406,00 + 8701,00 + 3912,92 +725,40 +10396,31 +0,00 +36387,08 +51981,54 = 233417,73 грн.*

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

$$ЗВ = \frac{B_{заг}}{\eta}, \qquad (5.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,95.

$ЗВ = 233417,73 / 0,95 = 245702,88$ *грн.*

5.3 Оцінювання важливості та наукової значимості науково-дослідної роботи

Оцінювання та доведення ефективності виконання науково-дослідної роботи фундаментального чи пошукового характеру є достатньо складним процесом і часто базується на експертних оцінках, тому має вірогідний характер.

Для обґрунтування доцільності виконання науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» використовується спеціальний комплексний показник, що враховує важливість, результативність роботи, можливість впровадження її результатів у виробництво, величину витрат на роботу.

Комплексний показник K_P рівня науково-дослідної роботи може бути розрахований за формулою:

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t}, \tag{5.18}$$

де I – коефіцієнт важливості роботи. Приймемо $I = 4$;

n – коефіцієнт використання результатів роботи; $n = 0$, коли результати роботи не будуть використовуватись; $n = 1$, коли результати роботи будуть використовуватись частково; $n = 2$, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; $n = 3$, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок. Приймемо $n = 2$;

T_C – коефіцієнт складності роботи. Приймемо $T_C = 3$;

R – коефіцієнт результативності роботи; якщо результати роботи плануються вище відомих, то $R = 4$; якщо результати роботи відповідають

відомому рівню, то $R=3$; якщо нижче відомих результатів, то $R=1$. Приймемо $R=4$;

B – вартість науково-дослідної роботи, тис. грн. Приймемо B =245702,88 грн;

t – час проведення дослідження. Приймемо t = 0,08 років, (1 міс.).

Визначення показників I, n, T_C, R, B, t здійснюється експертним шляхом або на основі нормативів [22].

$$K_P = \frac{I^n \cdot T_C \cdot R}{B \cdot t} = 4^2 \cdot 3 \cdot 4 \,/\, 245{,}7 \cdot 0{,}08 = 9{,}38.$$

Якщо $K_P > 1$, то науково-дослідну роботу на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» можна вважати ефективною з високим науковим, технічним і економічним рівнем.

Витрати на проведення науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» складають 245702,88 грн. Відповідно до проведеного аналізу та розрахунків рівень науково-економічного ефекту проведеної науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» є середній, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_P > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

ВИСНОВКИ

Проаналізовано сучасний стан розробки захищених систем для збору та аналізу даних, визначено ряд основних можливих вразливостей таких систем, розглянуто перелік державних стандартів для інформаційної безпеки. Визначено та розглянуто аналоги захищених систем, описано їхні переваги та недоліки, проведено порівняльний аналіз на основі якого сформовано актуальність розробки системи.

Було розроблено моделі поведінки користувача та роботи підсистеми аналізу, розроблено метод оцінювання ризиків ІБ, описано загальну блоксхему алгоритму роботи клієнтської частини системи збирання та аналізування даних для спеціальних задач.

Розроблено чотири модулі підсистеми аналізу системи збирання та аналізування даних для спеціальних задач, що розроблялася на базі веб-ресурсу під назвою «Реєстр випадків домашнього насилля». Проведено порівняльний аналіз мов розробки та визначено JavaScript, як найдоцільнішу. Розроблено модуль автентифікації користувача системи, модуль збирання даних за допомогою інтернет-форм, модуль аналізування внесених даних для відображення статистичної інформації про інциденти, та модуль для оцінювання факторів впливу кіберзагроз на стан інформаційної безпеки системи. Побудовано блок-схему алгоритму роботи підсистеми аналізу.

Проведено тестування роботи програмного засобу та визначено, що програмний засіб працює повністю коректно та виконує всі зазначені функції. Також проведено оцінювання стану ІБ системи з використанням розробленого методу та модуля оцінювання ризиків. В результаті оцінювання з’ясовано що рівень ІБ системи дорівнює 0.45 – середній рівень загроз. Тож можемо зробити висновок, що систему можна вдосконалити понизивши показники факторів оцінюваних ризиків, що в свою чергу призведе до покращення стану ІБ системи. Обраховані показники наочно зображають степінь ризиків, їх можна надати різним відділам компанії для вчасного реагування на оцінені загрози та

виділення ресурсів як фінансових так і кадрових для усунення таких загроз інформаційної безпеки.

Проведено економічне обґрунтування і доведено доцільність розробки захищеної системи збирання та аналізування даних для спеціальних задач. Під час дослідження виявлено – що витрати на проведення науково-дослідної роботи на тему «Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу» складають 245702,88 грн. Рівень науково-економічного ефекту – середній. Рівень доцільності виконання $K_P > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Захищена система аналізування даних для спеціальних задач. М. Ф. Касьянчук, В. В. Лукічов, І. О. Волокітенко, матеріали конференції "Контроль і управління в складних системах (КУСС-2022)" ВНТУ, Вінниця, 15-17 листопада 2022 р. URL: https://conferences.vntu.edu.ua/index.php/mccs/mccs2022/paper/view/16480 (дата звернення: 16.10.2022)
2. Data collection and analysis system. URL: https://wikiukuk.top/wiki/Data_collection_system (дата звернення: 05.09.2022)
3. Найкращі інструменти збору даних. URL: https://uk.myservername.com/10-best-data-collection-tools-with-data-gathering-strategies (дата звернення: 10.09.2022)
4. Avada Media. Парсери. URL: https://avada-media.ua/ua/services/parser/ (дата звернення: 15.09.2022)
5. TrendMicro. DDoS. URL: https://www.trendmicro.com/ vinfo/us/security/definition/denial-of-service-dos (дата звернення: 21.09.2022)
6. Захист локальної мережі. Типи атак. URL: https://sites.google.com/site/zahistlokalnoiemerezi/tipi-atak (дата звернення: 24.09.2022)
7. Наказ про прийняття та скасування державних стандартів. URL: https://zakon.rada.gov.ua/rada/show/v0312774-19#Text (дата звернення: 19.10.2022)
8. C-risk. Everithing you need to know about ISO 27005. URL: https://www.c-risk.com/en/blog/iso-27005/ (дата звернення: 23.10.2022)
9. IBS мобільний склад. URL: http://www.ibsystems.com.ua/ua (дата звернення: 25.10.2022)
10. SendPulse CRM. URL: https://sendpulse.ua/features/crm
11. Loger Pro. URL: https://cvkk.com.ua/lp/ (дата звернення: 27.10.2022)
12. REGMIC система збору даних. URL: https://regmik.ua/uk/product/sistema-sbora-dannyh-ssd-versiya-4-2/ (дата звернення: 30.10.2022)

13. Everything you need to know about ISO 27005. URL: https://www.c-risk.com/en/blog/iso-27005/ (дата звернення: 02.11.2022)
14. The FAIR™ Methodology for Cyber Risks. URL: https://www.c-risk.com/en/blog/fair-analysis/ (дата звернення: 05.11.2022)
15. Оновлений стандарт ISO/IEC 27005. URL: http://csm.kiev.ua/index.php?option=com_content&view=article&id=4232%3A-isoiec-27005-&catid=127%3A2018-01-16-07-36-07 (дата звернення: 07.11.2022)
16. Система вагових коефіцієнтів Фішберна. URL: https://cutt.ly/F0zLFy8 (дата звернення: 11.11.2022)
17. Елізабет Робсон, Ерік Фрімен. Head First JavaScript. Нью-Йорк, O'REILY, 2017. 34 с.
18. Ерік Метіз. Вивчаємо Python. Нью-Йорк, O'REILY, 2016. 22 с.
19. PHP Вірний Шлях. URL: http://iflista.github.io/php-the-right-way/ (дата звернення: 15.11.2022)
20. Atom. URL: https://atom.io/ (дата звернення: 17.11.2022)
21. Sublime Text 3. URL: https://www.sublimetext.com/3 (дата звернення: 17.11.2022)
22. В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с.

ДОДАТКИ

90

Додаток А

Протокол перевірки на наявність плагіату

ПРОТОКОЛ ПЕРЕВІРКИ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Захищена система збирання та аналізування даних для спеціальних задач. Частина 1. Підсистема аналізу.

Автор роботи: Касьянчук Максим Федорович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність – 97,5%. Схожість – 2,5%.

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 _____ Каплун В. А.
(підпис) (прізвище, ініціали)

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

Автор роботи _____ Касьянчук М. Ф.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Лук'янов Р. В.
(підпис) (прізвище, ініціали)

Додаток Б

Код програмного додатку

Модуль автентифікації

```
import { data } from "jquery";
import API_URL from '../api';
import notifications from "../notifications";

function login(){
    $('#login-btn').on('click', function(e){
        e.preventDefault();
        let login = $('#login-user-login').val();
        let password = $('#login-user-password').val();

        if(login && password){
            let data = {
                "userName": `${login}`,
                "password": `${password}`
            }

            let loginRequest = $.ajax({
                type: "POST",
                url: `${API_URL}users/login`,
                crossDomain: true,
                headers: {
                    'Access-Control-Allow-Origin': '*',
                    'Content-Type': 'application/json'
                },
                data: JSON.stringify(data),
                success: function(data){
                    // console.log(data);
                    sessionStorage.setItem("userId", data.userId);
                    sessionStorage.setItem("token", data.token);
                    $('#login-user-login').val('');
                    $('#login-user-password').val('');
                    window.location.replace('/index.html');
                },
                error: function (data) {
                    notifications.errorNotif(data.responseJSON);
                }

            });
        }
    });
}
export default login;

function register(){
    $("#register-btn").on("click", function(e){
        e.preventDefault();

        if(!$('#register-birthday').val()){
            notifications.errorNotif('Некоректно введена дата народження!');
            return;
        }
        let lastname    = $('#register-lastname').val();
        let firstname   = $('#register-firstname').val();
        let surname     = $('#register-surname').val();
        let birthday    = new Date($('#register-birthday').val());
        let address     = $('#register-address').val();
        let workplace   = $('#register-workplace').val();
        let position    = $('#register-rank').val();
        let mail        = $('#register-mail').val();
        let phone       = $('#register-phone').val();
        let password    = $('#register-password').val();
        let confirm     = $('#register-password-confirm').val();
        if(!lastname || !firstname || !surname || !birthday || !address || !workplace || !position
|| !mail || !phone || !password || !confirm){
            notifications.errorNotif('Не заповнені всі поля форми!');
            return;
```

```
        }
        if(password !== confirm){
            notifications.errorNotif('Пароль та підтвердження паролю не збігаються!');
            return;
        }
        if(!isEmail(mail)){
            notifications.errorNotif('Email введено не коректно!');
            return;
        }
        if(!isPhone(phone)){
            notifications.errorNotif('Телефонний номер введено не коректно!');
            return;
        }

        // console.log(address);
        let data =  {
            "firstName": firstname,
            "middleName": surname, //?
            "lastName": lastname,
            "birthDate": birthday.toISOString(), //?
            "address" : address,
            "authorityName": workplace,
            "positionName": position, //?
            "email": mail,
            "phoneNumber": phone,
            "password": password,
            "confirmPassword": confirm,
        }

        let registerRequest = $.ajax({
            type: "POST",
            url: `${API_URL}users/register`,
            crossDomain: true,
            headers: {
                'Access-Control-Allow-Origin': '*',
                'Content-Type': 'application/json'
            },
            data: JSON.stringify(data),
            success: function(data){
                // console.log(data);
                sessionStorage.setItem("userId", data.user.id);
                sessionStorage.setItem("token", data.token);

                $('#register-lastname').val('');
                $('#register-firstname').val('');
                $('#register-surname').val('');
                $('#register-birthday').val('');
                $('#register-address').val('');
                $('#register-workplace').val('');
                $('#register-rank').val('');
                $('#register-mail').val('');
                $('#register-phone').val('');
                $('#register-password').val('');
                $('#register-password-confirm').val('');

                window.location.replace('/index.html');
            },
            error: function (data) {
                notifications.errorNotif();
                console.log(data);
            }

        });

    });
}

function isEmail(email) {
  var regex = /^([a-zA-Z0-9_.+-])+\@(([a-zA-Z0-9-])+\.)+([a-zA-Z0-9]{2,4})+$/;
  return regex.test(email);
}
function isPhone(phone) {
    var regex = /\+38[0-9]{10}/;
    return regex.test(phone);
}
export default register;
```

Модуль збирання даних

```
  $('.add-new-person-btn').on('click', function(e){
        e.preventDefault();
            if(!$('#newperson-pasport-date').val()){
                notifications.errorNotif('Некоректно введена дата видачі!');
                return;
            }
            if(!$('#newperson-birthday').val()){
                notifications.errorNotif('Некоректно введена дата народження!');
                return;
            }
            let userToken = sessionStorage.getItem('token');
            let fullName =  $('#newperson-name').val();
            let documentTypeId = Number($('input[name="newperson-pasport-type"]:checked').val());
            let documentSeries = $('#newperson-pasport-ser').val();
            let documentNumber =   Number($('#newperson-pasport-number').val());
            let documentIssueDate =  new Date($('#newperson-pasport-date').val());
            let issuingAuthority = 0; //???
            let birthDate = new Date($('#newperson-birthday').val());
            let phoneNumber =  $('#newperson-phone').val();
            let hasChildren = false;
            let registrationAddress = $('#newperson-reg-address').val();
            let livingAddress =  $('#newperson-home-address').val();
            let workingPlace = $('#newperson-work').val();
            let socialSecurityId = 1;
            if(!fullName || !documentTypeId || !documentSeries || !documentNumber ||
!documentIssueDate || !birthDate || !phoneNumber || !registrationAddress || !livingAddress ||
!workingPlace){
                notifications.errorNotif('Не заповнені всі поля форми!');
                return;
            }
            if(!isPhone(phoneNumber)){
                notifications.errorNotif('Телефонний номер введено не коректно!');
                return;
            }
            if($('#newperson-children-yes').is(':checked')) { hasChildren = true; }
            else { hasChildren = false; }
            switch($('#newperson-money').val()){
                case 'Низький':
                    socialSecurityId = 1;
                    break;
                case 'Середній':
                    socialSecurityId = 2;
                    break;
                case 'Високий':
                    socialSecurityId = 3;
                    break;
            }
            console.log(socialSecurityId);

            const data = {
                "fullName": fullName,
                "documentTypeId": documentTypeId,
                "documentSeries": documentSeries,
                "documentNumber": documentNumber,
                "issuingAuthority": issuingAuthority,
                "documentIssueDate" : documentIssueDate.toISOString(),
                "birthDate": birthDate.toISOString(),
                "phoneNumber": phoneNumber,
                "hasChildren": hasChildren,
                "registrationAddress": registrationAddress,
                "livingAddress": livingAddress,
                "workingPlace": workingPlace,
                "socialSecurityId": socialSecurityId
            };
            console.log(data);
            if(userToken){
                let newpersonId;
                // let relId = $('.newperson-relationship-wrapper').attr('data-pers-rel-id');
                let createPersonRequest = $.ajax({
                    type: "POST",
                    url: `${API_URL}persons`,
                    crossDomain: true,
                    headers: {
```

```
                        'Access-Control-Allow-Origin': '*',
                        'Content-Type': 'application/json',
                        'Authorization': `Bearer ${userToken}`,
                    },
                    data: JSON.stringify(data),
                    success: function(data){
                        // console.log(data);
                        newpersonId = data;

                        $('#newperson-name').val('');
                        $('#newperson-pasport-ser').val('');
                        $('#newperson-pasport-number').val('');
                        $('#newperson-pasport-date').val('');
                        $('#newperson-birthday').val('');
                        $('#newperson-phone').val('');
                        $('#newperson-reg-address').val('');
                        $('#newperson-home-address').val('');
                        $('#newperson-work').val('');
                        notifications.succsessNotif('Запис про особу успішно створено');
                    },
                    error: function (data) {
                        notifications.errorNotif(data.responseJSON);
                        //console.log(data);
                    }
                });
    $('#add-new-record-btn').on('click', function(e){
        e.preventDefault();
        if(!$("#newrecord-register-date").val()){
            notifications.errorNotif('Некоректно введена дата реєстрації!');
            return;
        }
        let registryNumber = Number($('#newrecord-registryNumber').val());
        let registrationDate = new Date($("#newrecord-register-date").val());
        let initiatorAuthority = $("#newrecord-init-org").val();
        let executorAuthority = $("#newrecord-execute-org").val();
        let incidentTypeId = Number($('#newrecord-event-type').val());
        let qualificationId = Number($("#newrecord-qualification").val());
        let address = $("#newrecord-event-place").val();
        let reason = $("#newrecord-event-reason").val();
        let description = $("#newrecord-describe").val();
        let takenActions = $("#newrecord-measures").val();
        let incidentPersons = {};

        incidentPersons = {
            "1": agressor arr,
            "2": victim_arr
        }

        if(!registryNumber || !registrationDate || !initiatorAuthority || !executorAuthority ||
!incidentTypeId || !qualificationId || !address || !reason || !description || !incidentPersons){
            notifications.errorNotif('Не заповнені всі поля форми!');
            return;
        }

        const data = {
            "registryNumber": registryNumber,
            "registrationDate": registrationDate.toISOString(),
            "initiatorAuthority": initiatorAuthority,
            "executorAuthority": executorAuthority,
            "incidentTypeId": incidentTypeId,
            "qualificationId": qualificationId,
            "address": address,
            "reason": reason,
            "description": description,
            "takenActions": takenActions,
            "incidentPersons": incidentPersons
        };
        // console.log(data);
        if(userToken){
            let createIncidentRequest = $.ajax({
                type: "POST",
                url: `${API_URL}incidents`,
                crossDomain: true,
                headers: {
                    'Access-Control-Allow-Origin': '*',
                    'Content-Type': 'application/json',
                    'Authorization': `Bearer ${userToken}`,
                },
```

```
                data: JSON.stringify(data),
                success: function(data){
                    // console.log(data);
                    $('#newrecord-registryNumber').val('');
                    $("#newrecord-register-date").val('');
                    $("#newrecord-init-org").val('');
                    $("#newrecord-execute-org").val('');
                    $('#newrecord-event-type').val('');
                    $("#newrecord-qualification").val('');
                    $("#newrecord-event-place").val('');
                    $("#newrecord-event-reason").val('');
                    $("#newrecord-describe").val('');
                    $("#newrecord-measures").val('');
                    notifications.succsessNotif("Запис про подію успішно створено");
                },
                error: function (data) {
                    notifications.emptyNotif(data.responseJSON)
                }
            });
        }

    });

}
```

Модуль аналізування даних

```
    if(userToken){
        let getAllPersonsRequest = $.ajax({
            type: "GET",
            url: `${API URL}persons`,
            crossDomain: true,
            headers: {
                'Access-Control-Allow-Origin': '*',
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${userToken}`,
            },
            success: function(data){
                data.map(item => {
                    personsArr.push({
                        'fullName': item.fullName,
                        'id': item.id,
                        'normalizedName': item.fullName.toLowerCase()
                    });
                });
            },
            error: function (data) {
                // console.log(data);
            }
        });
        getAllPersonsRequest.done(function(){
            $('#search-person-name').on('input', function(){
                $('.search-res').html('');
                if($(this).val().length > 1){
                    personsArr.map(elem=>{
                        let name = elem.fullName;
                        let normalizedName = elem.normalizedName;
                        if (normalizedName.includes($(this).val().toLowerCase())){
                            $('.search-res').append(`<li class="search-res-item" data-person-
id="${elem.id}"><i class="fas fa-user"></i> ${name}</li>`);
                            $('.search-res-item').each(function(){
                                if($(this).attr('data-listener') === 'true'){
                                    return ;
                                } else{
                                    $(this).on('click', function(){
                                        $('#search-person-name').val($(this).text());
                                        $('#search-person-name').attr('data-person-id',
$(this).attr('data-person-id'));
                                        $('.search-res').html('');
                                    });
                                    $(this).attr('data-listener', 'true');
                                }
                            });
                        }
                    });
```

```
                }
            });
            // console.log(personsArr);
        });

    }
if(creationDate >= monthStart && creationDate <= monthEnd){
monthInc ++;
if(personId && userToken){
            let getPersonInfoRequest = $.ajax({
                type: "GET",
                url: `${API URL}persons/${personId}`,
                crossDomain: true,
                headers: {
                    'Access-Control-Allow-Origin': '*',
                    'Content-Type': 'application/json',
                    'Authorization': `Bearer ${userToken}`,
                },
            if(executionDate >= monthStart && executionDate <= monthEnd){
          executedMonthInc ++;
}
let allConsiderTime = data.reduce((accumulator, item) =>{
let considerTime = item.statusChangeDate - item.creationDate;
return accumulator + considerTime;
}, initValue);
let averageConsider = allConsiderTime / data.length;
```

Модуль оцінювання ризиків

```
function riskMan() {
    $('.add-risk-management-btn').on('click', function(e){
        e.preventDefault();
        const strTpl = `
        <div class="row factor">
            <div class="col-12 col-md-6">
                <div class="form-group">
                    <label class="form-label">Назва фактору Fi</label>
                    <input type="text" name="" class="fac-name" value="">
                </div>
            </div>
            <div class="col-12 col-md-6">
                <div class="form-group">
                    <label class="form-label">Значення фактору Fi</label>
                    <input type="text" name="" class="fac-val" value="">
                </div>
            </div>
        </div>
        `;
        $('.risk-man-table').append(strTpl);
    });

    $('.risk-management-btn').on('click', function(e){
        e.preventDefault();
        $('.fac-name').each(function(){
            let val = $(this).val();
            if(!val) {
                notifications.emptyNotif("Не заповено поле назва фактору!");
            }
        });

        let facValues = [];
        $('.fac-val').each(function(){
            let val = Number($(this).val());
            if(!val) {
                notifications.emptyNotif("Не заповено поле значення фактору!");
            }
            facValues.push(val);
        })

        let numberOfValues = facValues.length;

        let lastFactor = facValues[facValues.length-1];
```

```
        let bFactVal = [];
        for(let i = 1; i<= numberOfValues; i++){
            let curBFactVal = 0.8 - 0.2 * (i - 1);
            bFactVal.push(Number(curBFactVal.toFixed(2)));
        }

        let pi = 1/numberOfValues;

        let BRiskLevel = 0;
        for(let i = 0; i < bFactVal.length; i++){
            BRiskLevel += bFactVal[i] * pi;
        }


        let finishBRisk = 0;
        if(lastFactor<=0.2){
            finishBRisk = Number((BRiskLevel - lastFactor).toFixed(2));
        } else{
            finishBRisk = Number((BRiskLevel + lastFactor).toFixed(2));
        }

        let finishBRiskText;

        if(finishBRisk > 0 && finishBRisk <= 0.2){
            finishBRiskText = 'незначний ризик загрози';
        }
        if(finishBRisk > 0.2 && finishBRisk <= 0.4){
            finishBRiskText = 'низький ризик загрози';
        }
        if(finishBRisk > 0.4 && finishBRisk <= 0.6){
            finishBRiskText = 'середній ризик загрози';
        }
        if(finishBRisk > 0.6 && finishBRisk <= 0.8){
            finishBRiskText = 'високий ризик загрози';
        }
        if(finishBRisk > 0.8 && finishBRisk <= 1){
            finishBRiskText = 'граничний ризик загрози';
        }

        $('.finish-fact-val').val(finishBRisk);
        $('.finish-fact-text').val(finishBRiskText);
        notifications.succsessNotif('Оцінювання успішно проведено');

        $('.risk-management-modal').addClass('show-modal');
    });
    $(".risk-management-modal-btn").on('click', function(e){
        e.preventDefault();
        $('.finish-fact-val').val('');
        $('.finish-fact-text').val('');
        $('.risk-management-modal').removeClass('show-modal');
    });
}
export default riskMan;
```

ІЛЮСТРАТИВНА ЧАСТИНА

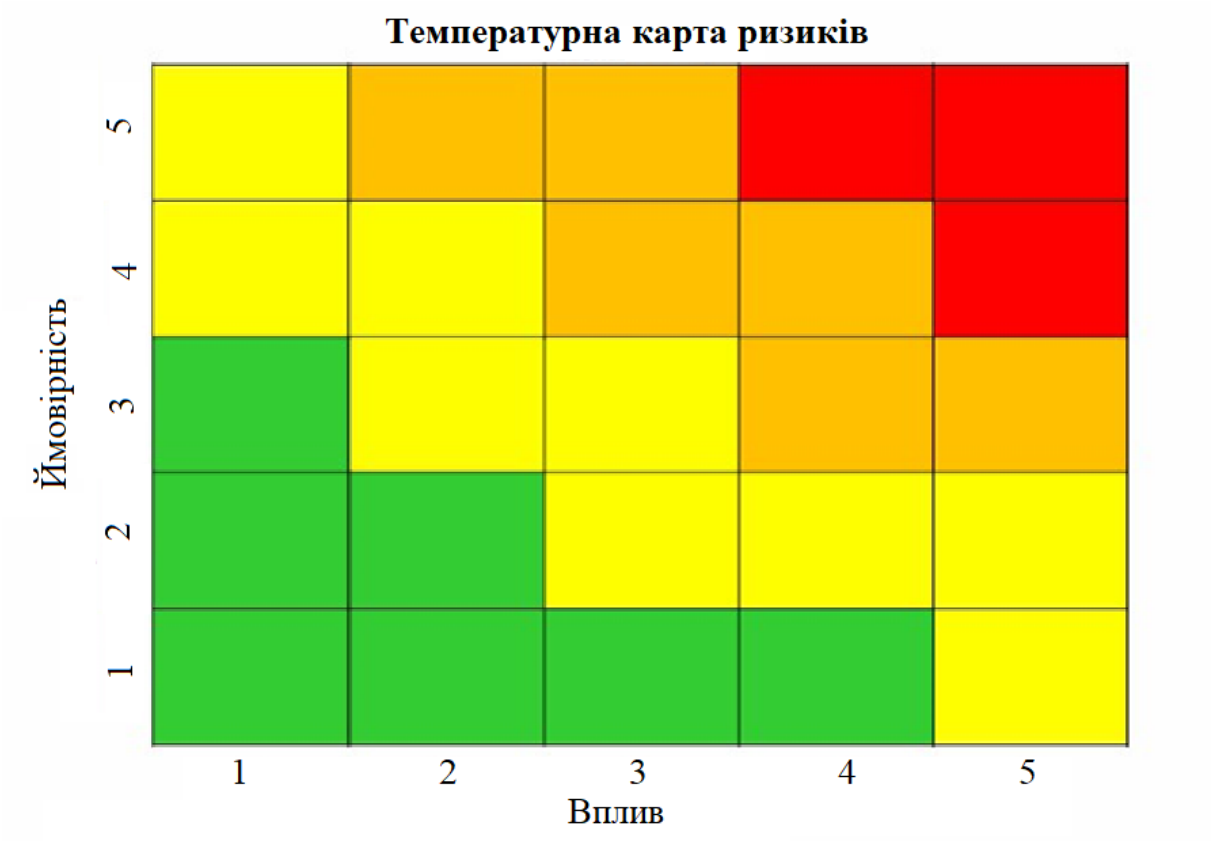
ЗАХИЩЕНА СИСТЕМА ЗБИРАННЯ ТА АНАЛІЗУВАННЯ ДАНИХ ДЛЯ СПЕЦІАЛЬНИХ ЗАДАЧ. ЧАСТИНА 1. ПІДСИСТЕМА АНАЛІЗУ

Порівняльні характеристики програмних аналогів

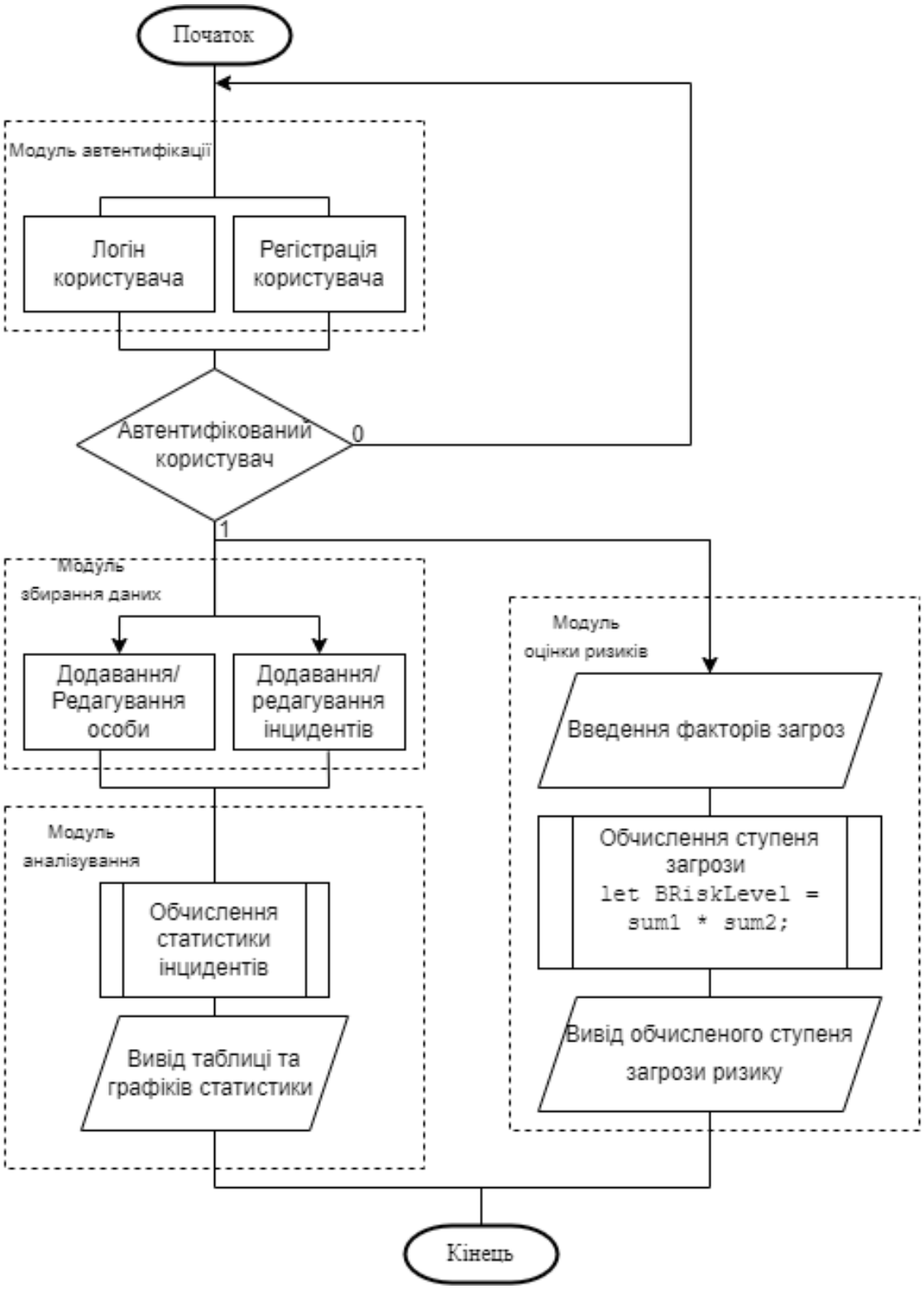
Критерій	IBS мобільний склад	SendPulse CRM	Loger Pro	REGMIK система збору даних	Захищена система збирання та аналізування даних
Командна робота над проектом	-	+	-	-	+
Захист від несанкціонованого доступу	+	+	+	-	+
Оцінка ризиків кібербезпеки	-	-	+	+	+
Побудова статистичних графіків	-	+	+	+	+
Email або SMS сповіщення	-	+	-	+	+
Робота в офлайн режимі	+	-	+	+	+
Хешування даних при передачі на сервер	-	-	-	-	+

Етапи управління ризиками за стандартом ISO 27005

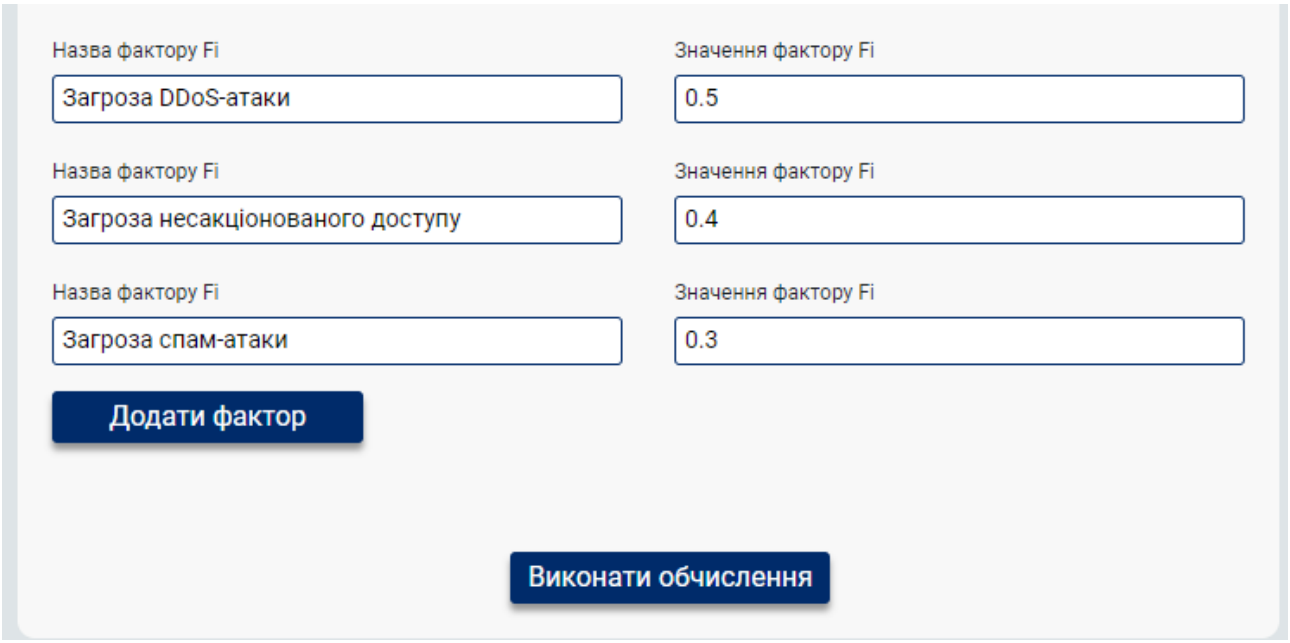
Кольорова карта критичності ризику



Загальний алгоритм роботи підсистеми аналізу



Результати тестування програмного засобу



Повідомлення про результат оцінювання загального стану ІБ системи