

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка методу та програмного забезпечення експертної системи супроводу  
реєстрації авторського права з використанням моделей штучного інтелекту»

Виконав: студент II курсу  
групи 1ПІ-22м спеціальності  
121 – Інженерія програмного забезпечення

Нестерук Віталій Андрійович

Керівник: к.т.н., доц. каф. ПЗ Кательніков Д.І.

«15» « грудня » 2023 р.

Опонент: к.т.н., доц. каф. ОТ Богомолів С.В.

«15» « грудня » 2023 р.

**Допущено до захисту**


Завідувач кафедри ПЗ

д.т.н., проф. Романюк О. Н.

(прізвище та ініціали)

«15» « грудня » 2023 р.

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра програмного забезпечення  
Рівень вищої освіти II-й (магістерський)  
Галузь знань 12 – Інформаційні технології  
Спеціальність 121 – Інженерія програмного забезпечення  
Освітньо-професійна програма – Інженерія програмного забезпечення

 ЗАТВЕРДЖУЮ  
Завідувач кафедри ПЗ  
Романюк О. Н.  
«19» вересня 2023 р.

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Нестеруку Віталію Андрійовичу

1. Тема роботи – розробка методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту.

Керівник роботи: Кательніков Денис Іванович, к.т.н., доцент кафедри ПЗ, затверджений наказом вищого навчального закладу від «18» вересня 2023 р. № 247.

2. Строк подання студентом роботи  
5 грудня 2023 р.

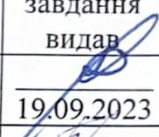
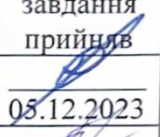
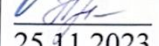
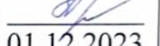
3. Вихідні дані до роботи: середовище розробки WebStorm; технології – мова програмування TypeScript, програмне середовище Node.js, бібліотека React та ін.; операційна система Microsoft Windows; метод надання рекомендацій; алгоритм розпізнавання контексту та намірів користувача; результати аналізу матеріалів – звіти, діаграми, порівняльні таблиці, підсумки.

4. Зміст текстової частини: вступ, аналіз юридичної природи реєстрації авторського права; аналіз існуючих підходів до реєстрації авторських прав; проектування архітектури експертної системи; розробка алгоритмів і методів роботи експертної системи; розробка клієнтської та серверної частин експертної системи; вибір та інтеграція моделі штучного інтелекту; тестування експертної системи; економічна частина.

5. Перелік ілюстративного матеріалу: блок-схеми алгоритмів роботи, діаграми потоків даних, компонентів, класів і галузей застосування експертної системи реєстрації авторського права.



6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Кательніков Д.І., к.т.н., доцент кафедри ПЗ	 19.09.2023	 05.12.2023
5	Причепя І.В., к.е.н, доцент кафедри ЕПВМ	 25.11.2023	 01.12.2023


7. Дата видачі завдання 19 вересня 2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сфери реєстрації авторського права в контексті експертних систем	20.09.23 – 02.10.23	<b>Вик</b>
2	Постановка задачі та проектування експертної системи	03.10.23 – 23.10.23	<b>Вик</b>
3	Розробка експертної системи	16.10.23 – 23.10.23	<b>Вик</b>
4	Вибір та інтеграція моделі штучного інтелекту	24.10.23 – 13.11.23	<b>Вик</b>
5	Тестування програми	14.11.23 – 24.11.23	<b>Вик</b>
6	Економічна частина	25.11.23 – 01.12.23	<b>Вик</b>
7	Оформлення матеріалів до захисту	22.11.23 – 02.12.23	<b>Вик</b>

Студент   
(підпис)

Нестерук В.А.  
(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи   
(підпис)

Кательніков Д.І.  
(прізвище та ініціали)

## Анотація

УДК 004.912.032.26

Нестерук В.А. Розробка методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту. Магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2023. 139 с.

На укр. мові. Бібліогр.: 49 назв; рис.: 14; табл.: 17.

У магістерській кваліфікаційній роботі здійснено детальний аналіз експертних систем для супроводу реєстрації авторського права з акцентом на застосування моделей штучного інтелекту. Досліджено актуальну ситуацію та виклики і проведено порівняльний аналіз існуючих аналогів, що дозволило виявити потребу у створенні нової, більш ефективної експертної системи з інтегруванням штучного інтелекту для автоматизації та оптимізації процесів.

Розроблено концепцію і методологію створення системи, включаючи модулі для обробки та аналізу даних, управління знаннями, а також модулі для взаємодії з користувачами. Особлива увага приділена розробці інтуїтивно зрозумілого та зручного інтерфейсу, який би спрощував взаємодію з системою та процес реєстрації авторського права.

Для розробки серверної частини використано Node.js, що забезпечує високу продуктивність та легкість інтеграції з різними базами даних і веб-сервісами. Клієнтську частину розроблено з використанням TypeScript та React, що дозволяє створювати надійний та масштабований користувацький інтерфейс.

Результати роботи відкривають шлях для подальшого розвитку та впровадження експертних систем у сфері захисту інтелектуальної власності, забезпечуючи більшу доступність та ефективність процесів реєстрації авторського права.

Ключові слова: ЕКСПЕРТНА СИСТЕМА, РЕЄСТРАЦІЯ, АВТОРСЬКЕ ПРАВО, МОДЕЛЬ ШТУЧНОГО ІНТЕЛЕКТУ

## **Abstract**

Nesteruk V.A. Development of a Method and Software for an Expert System to Support Copyright Registration Using Artificial Intelligence Models. Vinnytsia: VNTU, 2023. 139 p.

In Ukrainian language. Bibliographer: 49 titles; fig.: 14; tabl.: 17.

The master's qualification thesis conducts a thorough analysis of expert systems for copyright registration support, with an emphasis on the application of artificial intelligence models. The current situation and challenges were investigated, and a comparative analysis of existing analogs was conducted, which identified the need to create a new, more effective expert system with the integration of artificial intelligence for process automation and optimization.

The concept and methodology for the system's creation were developed, including modules for data processing and analysis, knowledge management, as well as modules for user interaction. Special attention was given to the development of an intuitive and user-friendly interface that simplifies interaction with the system and the copyright registration process.

Node.js was used for the development of the server part, ensuring high performance and ease of integration with various databases and web services. The client part was developed using TypeScript and React, which allows for the creation of a reliable and scalable user interface.

The results of the work pave the way for further development and implementation of expert systems in the field of intellectual property protection, ensuring greater accessibility and efficiency of copyright registration processes.

**Keywords: EXPERT SYSTEM, REGISTRATION, COPYRIGHT, ARTIFICIAL INTELLIGENCE MODEL**

## ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБҐРУНТУВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ.....	7
1.1. Основи авторського права та процесу його реєстрації.....	7
1.2. Принципи та механізми роботи експертних систем.....	10
1.3. Використання штучного інтелекту в експертних системах.....	12
1.4. Порівняльний аналіз аналогів.....	15
1.5. Обґрунтування новизни і задачі дослідження.....	17
1.6. Висновки.....	20
РОЗДІЛ 2 КОНЦЕПТУАЛІЗАЦІЯ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ.....	23
2.1. Формулювання вимог до системи.....	23
2.2. Вибір і обґрунтування моделі та методу штучного інтелекту.....	26
2.3. Проектування архітектури системи.....	29
2.4. Розробка методу оптимізації й автоматизації процесу реєстрації.....	32
2.5. Розробка методу розпізнавання контексту та намірів користувача.....	34
2.6. Проектування інтерфейсу користувача.....	39
2.7. Висновки.....	43
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЕКСПЕРТНОЇ СИСТЕМИ.....	45
3.1. Вибір технологій та інструментів розробки.....	45
3.2. Розробка серверної частини системи.....	48
3.3. Розробка клієнтської частини системи.....	52
3.4. Інтеграція моделей штучного інтелекту.....	58

3.5. Безпека та захист даних у системі .....	61
3.6. Висновки .....	63
РОЗДІЛ 4 ЕСТУВАННЯ СИСТЕМИ.....	66
4.1. Визначення методів та організація тестування.....	66
4.2. Функціональне тестування.....	67
4.3. Тестування продуктивності та навантаження .....	70
4.4. Тестування безпеки .....	73
4.5. Висновки .....	77
РОЗДІЛ 5 КОНОМІЧНА ЧАСТИНА .....	79
5.1. Проведення комерційного та технологічного аудиту науковотехнічної розробки .....	79
5.2. Розрахунок витрат на здійснення науково-дослідної роботи.....	86
5.3. Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором .....	92
5.4. Розрахунок ефективності вкладених інвестицій та періоду їх окупності ....	93
5.5. Висновки .....	96
ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	100
Додаток А (обов'язковий). Технічне завдання.....	<b>Error! Bookmark not defined.</b>
Додаток Б (обов'язковий). Протокол перевірки на наявність запозичень.....	108
Додаток В (довідниковий) Лістинг розроблених програм.....	109
Додаток Г (довідниковий) Ілюстративна частина .....	133

## ВСТУП

**Обґрунтування вибору теми дослідження.** Сучасний світ неможливо уявити без інтелектуальної власності, яка є ключовим активом багатьох організацій та індивідів. Процес реєстрації авторського права, що є важливим етапом у захисті інтелектуальних прав, часто може бути складним і трудомістким. Розробка експертної системи, що спроможна автоматизувати та оптимізувати цей процес, є актуальною та необхідною задачею у контексті цифровізації суспільства. Використання штучного інтелекту у такій системі може значно підвищити її ефективність, надаючи можливість аналізувати дані, прогнозувати можливі проблеми та надавати користувачам рекомендації.

Розробка експертних систем у сфері захисту інтелектуальної власності є важливим напрямком у поліпшенні якості та доступності юридичних послуг. Експертна система, яка базується на моделях штучного інтелекту, може надавати кваліфіковану допомогу у процесі реєстрації авторського права, забезпечуючи користувачів необхідною інформацією та підтримкою на кожному етапі цього процесу.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася відповідно до плану науково-дослідних робіт на кафедрі програмного забезпечення.

**Мета та завдання дослідження.** Метою роботи є підвищення продуктивності процесу реєстрації авторського права за рахунок автоматизації дій та інтеграції моделей штучного інтелекту, які дозволяють залучити інформацію з існуючих наукових джерел та напрацювання експертів.

Основні задачі дослідження є:

- аналіз існуючих методів та систем реєстрації авторського права;
- розробка методу аналізу та обробки даних для системи;
- проектування архітектури експертної системи;
- розробка алгоритмів роботи системи з використанням моделей штучного інтелекту;



- розробка інтерфейсу користувача для взаємодії з системою та серверної логіки;
- тестування та валідація розробленої системи;
- розрахунок економічних показників розробки.

**Об’єкт дослідження** – процес реєстрації авторського права.

**Предмет дослідження** – методи та засоби розробки експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту.

**Методи дослідження.** У процесі дослідження використовувались: теорія баз даних при організації зберігання інформації; теорія нейронних мереж та машинне навчання - для інтеграції моделі штучного інтелекту при автоматизації процесу реєстрації авторського права, аналізі запитів користувачів, вдосконаленні механізмів пошуку та фільтрації інформації; теорія розподілених систем для побудови розподіленої клієнт-серверної веб-платформи; комп’ютерне моделювання для аналізу та перевірки отриманих теоретичних положень..

### **Наукова новизна отриманих результатів.**

1. Уперше запропоновано метод автоматизації процесу реєстрації авторського права, особливість якого полягає у дотриманні вимог національного законодавства та врахування особливостей правової системи України, що сприятиме зміцненню ролі інституту інтелектуального права в Україні та, як наслідок, підвищенню привабливості національного ринку з точки зору міжнародного партнерства при використанні результатів інтелектуальної діяльності виключно легальним шляхом.

2. Подальшого розвитку отримав метод розпізнавання контексту та намірів користувача, який, на відміну від відомих методів, використовує інтегрованої моделі штучного інтелекту, що реалізована нейромережевим підходом, що дозволяє обробку даних природною мовою та обробку сценаріїв, пов’язаних з авторським правом та дотичними категоріями, зокрема з урахуванням інших об’єктів права інтелектуальної власності.

**Практична цінність отриманих результатів** полягає в тому, що на основі отриманих в роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби експертної системи у сфері права інтелектуальної власності, яка забезпечує ефективний та доступний процес реєстрації авторського права для широкого кола користувачів, в тому числі для авторів і заявників, які не мають достатньої юридичної кваліфікації, а не лише для фахівців у сфері авторського права.

**Особистий внесок здобувача.** Всі результати, представлені у роботі, були отримані автором самостійно. Всі аспекти підбору та розробки методу, алгоритмів і програмного забезпечення, а також аналізу та валідації, були виконані автором за допомогою TypeScript, React та Node.js, враховуючи інтеграцію та тренування моделі штучного інтелекту ChatGPT від OpenAI, а також опираючись на власний практичний досвід у сфері права інтелектуальної власності.

**Апробація результатів роботи.** Основні положення роботи були представлені та обговорені на відповідних конференціях: III Всеукраїнській науково – технічній конференції молодих вчених, аспірантів та студентів «Комп’ютерні ігри і мультимедіа як інноваційний підхід до комунікації - 2023» (ОНТУ, 2023) та Міжнародній науково-практичній Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» (НІКО/КЗВО «Вінницька академія безперервної освіти», 2023).

**Публікації.** Основні результати досліджень опубліковано в матеріалах конференцій [45, 46].

## РОЗДІЛ 1

### АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБҐРУНТУ-ВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ

#### 1.1. Основи авторського права та процесу його реєстрації

Авторське право є ключовим елементом захисту інтелектуальної власності, який надає авторам ексклюзивні права на їх твори. Водночас Закон України «Про авторське право і суміжні права» і пов'язані з ним нормативно-правові документи надають власникові авторського права ексклюзивне право виконувати та дозволяти іншим виконувати певні дії щодо захищеного твору, включаючи право на копіювання, розповсюдження твору, публічне відтворення або виконання та створення похідних творів [1, 2, 3].

Процес реєстрації авторського права, хоча й не обов'язковий, може надати додатковий захист та переваги в разі юридичних спорів. Для реєстрації твору необхідно подати заповнену форму заявки, примірник твору в матеріальному або ж цифровому вигляді, які підлягають реєстрації, а також супровідні документи, якщо це необхідно [2, 4].

На міжнародному рівні основні принципи та стандарти захисту авторського права встановлені Бернською конвенцією, яка є глобальним договором, що захищає права творців та їхні твори. Конвенція охоплює авторське право та суміжні права, такі як моральні права, сусідні права та право на розповсюдження, а також передбачає спеціальні положення для країн, що розвиваються [5].

Цифрове середовище поставило ряд викликів перед системою авторського права, пропонуючи як можливості, так і створюючи перешкоди для правовласників і загальної публіки. Розуміння та вирішення того, як це впливає на функціонування системи авторського права, є, можливо, одним з найбільш фундаментальних питань права інтелектуальної власності, які в даний час обговорюються політиками на глобальному рівні [6].

Нижче наведено табл. 1.1., яка демонструє деякі відмінності в системах реєстрації авторських прав у різних країнах, відображаючи добровільний

характер реєстрації у багатьох місцях та різні переваги, які реєстрація може надати.

Таблиця 1.1. – Порівняння систем реєстрації авторських прав

Країна	Система реєстрації	Орган	Оплата	Переваги реєстрації
США	Добровільна	Бюро авторських прав США	Так	Можливість формування статутного капіталу та відшкодування витрат у разі порушень
Канада	Добровільна	Канадське бюро інтелектуальної власності	Так	Доказ власності, полегшене забезпечення захисту
Кенія	Добровільна	Кенійська рада з авторських прав	Так	Доказ власності, полегшене забезпечення захисту
Велика Британія	Невизначено	Органи загальної юрисдикції	Ні	Припущення авторства
Індія	Добровільна	Бюро авторських прав	Так	Юридичні переваги, зокрема, припущення власності у суді
Австралія	Добровільна	Органи загальної юрисдикції	Ні	Створення публічного запису, потенційні юридичні переваги
Україна	Добровільна	УКРНОІВІ	Так	Доказ власності, полегшене забезпечення захисту

З огляду на вищенаведене, розуміння основ авторського права та процесу його реєстрації є ключовим для ефективного захисту інтелектуальної власності. Важливість авторського права полягає не лише в захисті інтересів авторів, а й у стимулюванні культурного розвитку та інновацій. Реєстрація авторського права, хоча й не є обов'язковою у багатьох країнах, може забезпечити додаткові переваги та захист у разі судових спорів.

Враховуючи глобалізацію та поширення цифрових технологій, захист авторських прав набуває нових вимірів та викликів. Це вимагає від авторів, правовласників та правозастосовних органів адаптації до постійно змінних умов цифрового середовища, а також розуміння міжнародних стандартів та норм. Особливо відчутною є проблема реєстрації авторського права на комп'ютерні програми в правових реаліях України, що досить детально роз'яснено на прикладі розгляду реєстрації авторського права на комп'ютерні ігри [45, 46].

Таблиця, що представлена в цьому розділі, демонструє відмінності у підходах до реєстрації авторських прав в різних країнах, підкреслюючи її добровільний характер та різноманітність можливих юридичних переваг. Це підкреслює важливість освіченого підходу до захисту авторських прав, що враховує специфіку кожної юрисдикції.

В контексті сучасної динаміки розвитку інтелектуальної власності, важливо відзначити потребу в постійному оновленні та розвитку законодавчих рамок, щоб вони відповідали вимогам часу та забезпечували ефективний захист прав творців. У цьому контексті розробка та впровадження експертних систем, здатних підтримувати процеси реєстрації та таким чином охорони авторських прав, відіграє ключову роль у забезпеченні доступності, прозорості та ефективності цих процесів.

## 1.2. Принципи та механізми роботи експертних систем

Експертні системи є зазвичай комп'ютерними системами, які імітують здатність людини-експерта приймати обґрунтовані рішення. Вони розроблені для вирішення складних проблем шляхом міркувань через обсяги знань, які в основному представлені у вигляді правил «якщо-то» («if-else») замість звичайного процедурного коду. Основні елементи експертної системи включають інтерфейс користувача, об'єкт пояснення, робочу пам'ять та рушій прийняття висновків. Експертні системи використовують різноманітні принципи для виведення висновків з вже відомих даних.

Експертна система із залученням штучного інтелекту виконує конкретне завдання з експертною ефективністю, застосовуючи попередньо визначені правила для аналізу інформації та формування висновків. Експертна система може надавати спеціалізовані поради або автоматизацію прийняття рішень, допомагати у вирішенні проблем та допомагати ідентифікувати помилки або ризики [7].

Деякі основні теоретичні аспекти у сфері експертних систем включають обговорення представлення знань, методів висновку, міркувань при невизначеності та нечіткого міркування [8]. Експертні системи є видом програмного забезпечення, яке імітує поведінку по вирішенню проблем людини-експерта в даній області. Структуру експертних систем, механізми міркувань та інші аспекти, пов'язані з їх застосуванням, також доцільно розглянути в даному розділі [9].

В рамках розвитку сучасних експертних систем, набуває значення інтеграція з передовими технологіями штучного інтелекту. Особливо це стосується впровадження алгоритмів машинного навчання та глибокого навчання, які забезпечують системам можливість самонавчання та адаптації до нових умов. Ця здатність значно розширює потенціал експертних систем у вирішенні нетипових та складних задач.

Додатково пропонується до ознайомлення табл. 1.2, що надає загальний огляд різних типів експертних систем та їх потенційного використання в різних



областях, включаючи розробку веб-додатку для реєстрації авторських прав, оскільки вибір конкретного типу експертної системи залежить від конкретних вимог та обмежень проекту.

Таблиця 1.2. – Порівняння різних типів експертних систем

Тип	Основні особливості	Переваги	Недоліки	Приклад
Системи на базі правил	Визначені правила та логічні висловлювання для прийняття рішень	Висока точність, простота розуміння	Обмежена гнучкість, відсутність вивчення	Перевірка відповідності законам, діагностика несправностей
Нейро-мережеві системи	Імітація роботи мозку для вирішення проблем	Гнучкість, можливість навчання	Важке розуміння та інтерпретація	Розпізнавання мови, прогнозування
Фазові системи	Використання фазової логіки для обробки нечітких даних	Гнучкість, обробка нечіткості	Складність розуміння та інтерпретації	Управління ризиками, прогнозування
Системи на основі випадків	Зберігання та пошук схожих випадків для прийняття рішень	Вивчення на основі досвіду, простота розширення	Обмеження в аналізі нових ситуацій	Підтримка прийняття рішень, діагностика несправностей
Гібридні системи	Комбінація різних підходів для прийняття рішень	Більш повноцінний аналіз, гнучкість	Складність розробки та управління	Розробка продуктів, системи рекомендацій

Варто відзначити, що ключовим елементом ефективності всіх експертних систем є база знань, яка містить інформацію та правила, необхідні для імітації рішень експертів. Розвиток технологій представлення знань дозволяє створювати більш гнучкі та комплексні системи, здатні обробляти великі обсяги даних та швидко адаптуватися до змін у вхідній інформації.

Важливим аспектом є також розвиток механізмів висновку, які дозволяють системі аналізувати інформацію та приймати обґрунтовані рішення. Особливе місце тут займає обробка невизначеності та нечіткого логічного міркування, що дозволяє системам ефективно працювати навіть за умови неповноти чи нечіткості даних.

Таким чином, було розглянуто ключові аспекти, які формують основу роботи експертних систем. Центральним елементом експертних систем є база знань, яка забезпечує систему інформацією, необхідною для імітації рішень людських експертів. Механізм виведення використовується для аналізу інформації з бази знань та вирішення конкретних завдань або проблем. Інтерфейс користувача є важливим для забезпечення легкого та інтуїтивного доступу до функцій системи. Особливе місце займає інтеграція штучного інтелекту, яка розширює можливості експертних систем, забезпечуючи їх здатність до навчання та адаптації. Розуміння цих принципів та механізмів є фундаментальним для розробки ефективних експертних систем, здатних вирішувати складні задачі та адаптуватися до змінних умов.

### **1.3. Використання штучного інтелекту в експертних системах**

Штучний інтелект (далі – ШІ) і експертні системи тісно пов'язані, оскільки експертні системи є частиною ширшого домену ШІ. Експертні системи використовують методи ШІ для вирішення проблем у специфічних областях, які зазвичай вимагають людської експертизи. Основна мета експертних систем – імітувати рішення, які приймає людина-експерт, використовуючи обсяги знань, представлені переважно у вигляді правил «якщо-то» [10, 11].

Однією з областей застосування ШІ в експертних системах є автоматичне розпізнавання мови. Для невеликих проектів експертні системи в області розпізнавання мови є ідеальною технікою ШІ для ідентифікації добре визначеного людського мовлення та генерації виводу. Застосування розпізнавання мови може бути використане для різних бізнес-випадків.

ШІ сприяє в автоматизації процесу відображення людського мислення машинами. Системи на основі ШІ можуть оцінювати дані та робити логічні висновки, подібно до того, як це може робити людський мозок, але в більш великому масштабі. Останнім часом стало майже неможливо для людей вручну видобувати, оцінювати та визначати значення величезних обсягів даних, якими оперують у своїй діяльності сучасні компанії [12]. Крім того, використання подібних технологій для певних відносно рутинних задач є не лише ефективнішим з точки зору швидкодії, а й з фінансової точки зору, оскільки машинальна людська праця, наприклад, щодо реєстраційних дій є значно дорожчою, ніж вартість автоматизації рішень, особливо в перспективі.

Важливим аспектом постає той факт, що в сучасних реаліях штучний інтелект в експертних системах вносить революційні зміни у спосіб обробки та аналізу даних. Сучасні технології штучного інтелекту, включаючи машинне навчання, нейронні мережі та глибоке навчання, дозволяють системам автоматизувати та оптимізувати багато процесів, які раніше вимагали значних, іноді надлишкових людських ресурсів. Це не тільки підвищує продуктивність та ефективність, але й дозволяє вирішувати складні завдання, які за межами можливостей традиційних систем.

Зокрема, табл. 1.3 покликана допомогти проілюструвати, як різні методи ШІ можуть впливати на процес прийняття рішень у експертних системах, що вбачається корисним для розуміння того, яким чином вони можуть бути використані в контексті розробки експертної системи реєстрації авторських прав.

Таблиця 1.3 – Поширені методи ШІ та їхні особливості

Метод ШІ	Опис	Вплив на процес прийняття рішень
Машинне навчання	Використання алгоритмів для визначення закономірностей в даних та їх використання для прогнозування.	Покращення точності та швидкості прийняття рішень за рахунок автоматизації.
Глибоке навчання	Використання нейронних мереж з багатьма шарами для аналізу великих наборів даних.	Підвищення точності розпізнавання складних шаблонів та залучення великих даних у процес прийняття рішень.
Правила експертних систем	Використання прописаних правил для аналізу даних та прийняття рішень на їх основі.	Покращення структурованості та прозорості процесу прийняття рішень.
Логічне виведення	Використання логічних механізмів для виведення висновків на основі вхідних даних.	Забезпечення точного та послідовного аналізу даних для прийняття обґрунтованих рішень.
Фазове виведення	Використання фазової логіки для обробки нечітких або неповних даних.	Покращення гнучкості та ефективності системи прийняття рішень у невизначених умовах.
Системи на основі випадків	Пошук та аналіз попередніх випадків для прийняття рішень в нових ситуаціях.	Використання історичного досвіду для покращення якості рішень.

Саме тому доцільно підкреслити, що використання ШІ в експертних системах має величезний потенціал для трансформації традиційних підходів до прийняття рішень. Застосування різноманітних методів ШІ, від машинного навчання до фазового виведення, дозволяє створювати більш гнучкі, точні та адаптивні системи, здатні вирішувати широкий спектр задач. Таким чином, інтеграція ШІ в експертні системи не тільки покращує їхню функціональність, але й відкриває нові горизонти можливостей у різних галузях, включаючи розробку систем для реєстрації авторських прав.

#### **1.4. Порівняльний аналіз аналогів**

Порівняльний аналіз експертних систем важливий для визначення їхніх відносних переваг та недоліків, а також для забезпечення оптимального підходу до розробки нових систем. Дослідження вказують на різні методи, що використовуються для створення експертних систем, зокрема Баєсівська класифікація, правила виведення та фреймові системи [13]. Кожен з цих методів має свої відносні переваги та недоліки в залежності від конкретного застосування.

На практиці, експертні системи часто порівнюють з системами підтримки прийняття рішень (DSS). Ці системи розроблені для підтримки процесу прийняття рішень, але вони не мають вбудованого знання, яке є характерним для експертних систем. Експертні системи мають вбудовану базу знань, яка дозволяє їм надавати рекомендації й висновки на основі аналізу вхідних даних [14]. Відмінною особливістю DSS є їхня здатність обробляти великі масиви даних та забезпечувати аналітичні інструменти для підтримки рішень, але вони значною мірою покладаються на здатності та інтуїцію користувача в інтерпретації цих даних. Натомість експертні системи мають вбудовану базу знань, яка дозволяє їм надавати рекомендації й висновки на основі аналізу вхідних даних, імітуючи процес прийняття рішень людським експертом. Ця ключова особливість надає експертним системам здатність самостійно робити висновки та надавати специфічні поради, засновані на накопиченому досвіді та

знаннях, закладених в систему. Як наслідок експертні системи можуть бути особливо корисними у ситуаціях, де необхідно емулювати глибоке розуміння певної області знань, наприклад, у юридичних консультаціях, медичній діагностиці чи в сфері фінансового аналізу.

Також варто врахувати різні підходи до реалізації експертних систем. Типово експертні системи реалізуються як програмне забезпечення з використанням ітеративного алгоритму для обробки мережі правил-фактів. Однак існує можливість реалізації експертних систем на апаратному рівні, що також може мати свої переваги та недоліки [15].

Таблиця 1.4 – Переваги та недоліки методів і архітектур аналогів

№	Метод/архітектура	Опис	Переваги	Недоліки
1	Баєсівська класифікація	Метод, що використовує ймовірнісний підхід для висновків	Висока точність при достатньо великому наборі даних	Потреба в великому наборі даних для навчання
2	Правила виведення	Використання набору правил для формулювання висновків	Легкість розуміння та відлагодження	Може бути обмежено складними доменами
3	Фреймові системи	Реалізація знань через структури даних, відомі як фрейми	Добре структуровані дані	Важкість управління при великому обсязі знань
4	Апаратна реалізація	Реалізація експертної системи на апаратному рівні	Висока продуктивність	Високі витрати та складність у реалізації



Зокрема, результат проведення порівняльного аналізу існуючих експертних систем у сфері реєстрації авторського права, дозволяє визначити ключові функціональні та технологічні особливості, які впливають на ефективність та зручність користування такими системами. Було виявлено, що більшість існуючих систем фокусуються на базовій автоматизації процесів, але мають обмеження з точки зору адаптивності, інтерактивності та інтеграції з передовими технологіями штучного інтелекту. Аналіз також показав, що значний простір для інновацій існує у вдосконаленні користувацького досвіду та впровадженні гнучких, адаптивних рішень на основі ШІ. Ці висновки зіграють важливу роль у розробці описаної експертної системи, спрямованої на підвищення ефективності та зручності процесу реєстрації авторських прав.

### **1.5. Обґрунтування новизни і задачі дослідження**

У сучасному цифровому суспільстві важливо надати можливість авторам легко реєструвати своє авторське право на твори, що вони створюють. Процес реєстрації авторських прав може бути складним та затратним, особливо для окремих осіб та малих компаній. Розробка експертної системи, яка може полегшити цей процес, є важливим кроком у спрощенні доступу до охорони й захисту інтелектуальної власності.

Ключова ідея даного дослідження полягає у розробці методу та програмного забезпечення для експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту. Система повинна бути здатною допомагати користувачам у формуванні заявок на реєстрацію авторського права, поданні цих заявок онлайн та слідкуванні за прогресом реєстрації. Задачі дослідження включають наступні елементи, враховуючи закладену в них суть:

- Аналіз існуючих методів та технологій: вивчення сучасних методів та технологій, які вже використовуються в експертних системах для супроводу реєстрації авторського права, а також ідентифікація їхніх переваг та недоліків.

- Визначення вимог: формулювання вимог до функціональності, надійності та зручності майбутньої системи з урахуванням потреб користувачів та особливостей процесу реєстрації авторського права.
- Вибір та обґрунтування методів штучного інтелекту: вибір найбільш ефективних методів ШІ для вирішення конкретних задач, пов'язаних з реєстрацією авторського права, та їх обґрунтування.
- Розробка архітектури системи та алгоритмів її роботи: проектування основних компонентів системи та розробка ключових алгоритмів для обробки даних та взаємодії з користувачами.
- Проектування інтерфейсу користувача: розробка інтуїтивного та зручного інтерфейсу для користувачів, який забезпечить легке формування, подання та слідкування заявок на реєстрацію авторського права.
- Розробка програмного забезпечення та його тестування: розробка програмного забезпечення для реалізації вибраних методів ШІ та архітектури системи, а також його тестування для забезпечення відповідності встановленим вимогам.
- Оцінка економічної ефективності: розробка методик для оцінки економічної ефективності розробленої системи та її потенційного комерційного впливу.

Інноваційність описаної в даній роботі експертної системи полягає в її здатності ефективно інтегрувати різні технології та методи штучного інтелекту для автоматизації процесу реєстрації авторських прав. Важливою є інтеграція з інтелектуальними алгоритмами для аналізу даних, які можуть визначати оптимальні шляхи для формування та подачі заявок, а також для надання рекомендацій користувачам. Це покращує користувацький досвід та сприяє зростанню точності та швидкості обробки запитів. Враховуючи ці аспекти, система не лише спрощує існуючі процедури, але й відкриває нові можливості для захисту інтелектуальної власності.

Загальний огляд архітектурних компонентів та ключових концепцій системи допоможе зрозуміти рис. 1.1.



Рисунок 1.1. – Діаграма структури компонентів експертної системи

Що стосується новизни описаної експертної системи, то вона відрізняється від існуючих аналогів своєю інноваційністю та комплексністю підходу. Основною новизною цієї системи є її здатність автоматизувати взаємодію з користувачами, спрощуючи процес формування та подачі заявок на реєстрацію авторських прав. Це досягається завдяки інтеграції передових технологій штучного інтелекту, які дозволяють системі ефективно обробляти користувацькі запити та надавати точні та відповідні відповіді.

Крім того, система включає сучасні веб-технології, такі як TypeScript, React та Node.js, які сприяють створенню гнучкого, масштабованого та стабільного інтерфейсу, що детальніше описується в подальших розділах. Ці технології забезпечують високу продуктивність та надійність системи, що є ключовими факторами для її ефективного функціонування в реальних умовах. Аналіз різних архітектур експертних систем підтвердив, що комбінація моделей штучного інтелекту з сучасними веб-технологіями є перспективним напрямком для розвитку інтелектуальних систем у сфері захисту авторських прав.

Вищенаведена інформація дає можливість чітко окреслити ключові аспекти дослідницької роботи. Зокрема, дана експертна система не тільки відповідає сучасним вимогам у сфері автоматизації юридичних та

адміністративних процесів, але й вносить значний вклад у розвиток технологій у цій області, відкриваючи шлях для подальшого впровадження інтелектуальних систем у сфері права інтелектуальної власності. Також було з'ясовано, що основною задачею є створення ефективної та інноваційної експертної системи для супроводу реєстрації авторського права. При цьому акцент варто робити не лише на технічному аспекті розробки, але й на глибокому розумінні потреб користувачів та особливостей процесу реєстрації авторського права. Важливим аспектом є інтеграція з передовими методами штучного інтелекту для підвищення точності та релевантності рекомендацій, що надаються системою. Також ми прагнемо забезпечити гнучкість та масштабованість системи, щоб вона могла адаптуватися до змінних умов ринку та законодавства. Цей розділ закладає основу для наступних етапів нашої роботи, де ми зосередимось на конкретних методах та технологіях реалізації задуманого проекту.

## **1.6. Висновки**

В даному розділі було проведено аналіз сучасного стану питання розробки експертних систем для супроводу реєстрації авторського права. Були розглянуті основні аспекти авторського права та його реєстрації, принципи та механізми роботи експертних систем, використання штучного інтелекту в експертних системах, а також проведено порівняльний аналіз існуючих методів та архітектур експертних систем.

Варто зазначити основні тези з даного розділу:

- Важливість авторського права полягає в тому, що воно є ключовим елементом для захисту інтелектуальної власності та стимулювання творчості. Ефективний процес реєстрації авторського права є важливим для забезпечення правового захисту авторів.
- Експертні системи як інструмент супроводу можуть значно полегшити процес реєстрації авторського права, надаючи необхідну інформацію та допомогу в формуванні та поданні заявок.

- Використання штучного інтелекту може значно підвищити ефективність та точність експертних систем, надаючи можливість автоматизації багатьох аспектів процесу реєстрації.
- Існує велика кількість методів та архітектур для створення експертних систем, кожна з яких має свої переваги та недоліки, і саме ця різноманітність створює простір для максимально об'єктивного дослідження.

Вищенаведене підкреслює потенційну цінність розробки нової експертної системи для супроводу реєстрації авторського права, яка може допомогти авторам легко та ефективно реєструвати свої твори.

З урахуванням всіх аспектів проведеного дослідження, стає зрозумілим, що сучасні експертні системи, інтегровані з штучним інтелектом, мають значний потенціал для оптимізації та автоматизації процесів у сфері авторського права. Ефективна реалізація таких систем може значно спростити процедуру реєстрації авторських прав, зробивши її більш доступною та менш трудомісткою для авторів і власників інтелектуальної власності.

На основі аналізу особливостей і потреб реєстрації авторських прав, можливо визначити важливість розвитку інноваційних інструментів, які можуть адаптуватися до швидкозмінних вимог цифрового світу. Інтеграція передових технологій штучного інтелекту, таких як машинне навчання та глибоке навчання, дозволить системі ефективно аналізувати великі обсяги даних, вирішувати складні задачі та надавати точні рекомендації.

Додатково, під час розробки нової системи, важливо зосередитися на розробці зручного та інтуїтивного інтерфейсу користувача, що забезпечить легке взаємодію з системою незалежно від рівня технічної підготовки користувачів. Це забезпечить широкий доступ до інструментів захисту інтелектуальної власності та сприятиме стимулюванню творчості та інновацій.

У контексті економічної ефективності, розроблення такої системи може мати важливі комерційні переваги, забезпечуючи ефективність процесів та зниження витрат на управління інтелектуальною власністю. Це, в свою чергу,

може принести значну користь як для індивідуальних авторів, так і для бізнес-структур, що працюють у сфері інтелектуальної власності.

Таким чином, залучення штучного інтелекту в процеси реєстрації авторських прав відкриває нові можливості для підвищення ефективності та доступності цих процесів, водночас сприяючи розвитку інновацій і захисту інтелектуальної власності на глобальному рівні.



## РОЗДІЛ 2

### КОНЦЕПТУАЛІЗАЦІЯ РОЗРОБКИ ЕКСПЕРТНОЇ СИСТЕМИ

#### 2.1. Формулювання вимог до системи

Успішна реалізація будь-якого програмного продукту починається з чіткого визначення вимог до нього. Вимоги вказують на функціональність, яку повинен надавати продукт, а також на обмеження, які повинні бути враховані під час його розробки. У контексті експертної системи супроводу реєстрації авторського права, вимоги можуть бути сформульовані наступним чином:

Функціональні вимоги:

- Підтримка формування заявок – система повинна надавати засоби для створення та редагування заявок на реєстрацію авторського права;
- Онлайн подання заявок – система повинна надавати можливість подавати заявки на реєстрацію онлайн;
- Моніторинг статусу реєстрації – система повинна надавати засоби для відстеження статусу реєстрації заявок;
- Інтеграція з базами даних – система повинна інтегруватися з існуючими базами даних авторських прав.
- Нефункціональні вимоги:
- Зручність користувача: Інтерфейс повинен бути інтуїтивно зрозумілим та зручним для користувачів;
- Безпека даних – система повинна забезпечувати надійний захист даних користувачів;
- Швидкість обробки – система повинна забезпечувати швидку обробку заявок та відповідей;
- Масштабованість – система повинна бути готовою до масштабування для підтримки збільшення кількості користувачів та/або даних.
- Обмеження:
- Технологічні обмеження – можливі обмеження, пов'язані з використанням конкретних технологій або платформ;

- Бюджетні обмеження – обмеження, пов’язані з бюджетом на розробку та впровадження системи;
- Часові обмеження – обмеження, пов’язані з графіком розробки та впровадження системи.

Для детальнішого розуміння технічного аспекту вимог додатково пропонується табл. 2.1., що розширює перелік функціональних вимог та обмежень.

Таблиця 2.1 – Функціональні вимоги та обмеження системи

№	Функціональні вимоги	Обмеження
1	Реєстрація нових користувачів	Підтвердження електронної пошти
2	Аутентифікація користувачів	Мінімальна довжина пароля: 8 символів
3	Створення нових заявок на реєстрацію авторського права	Обмеження на кількість файлів: 10 файлів
4	Перегляд статусу заявок	Доступ тільки для авторизованих користувачів
5	Онлайн-чат для підтримки користувачів	Операційні години: з 9:00 до 18:00
6	Завантаження документів	Максимальний розмір файлу: 20 МБ
7	Отримання сповіщень про статус заявки	Налаштування на рівні користувача
8	Пошук та фільтрація заявок	-
9	Інтеграція з моделлю AI для аналізу даних	Доступ до інтернету
10	Звітність та аналітика	Доступ тільки для адміністраторів

Також наводиться рис. 2.1, де зображено діаграму потоку даних для відповідної експертної системи супроводу реєстрації авторського права.

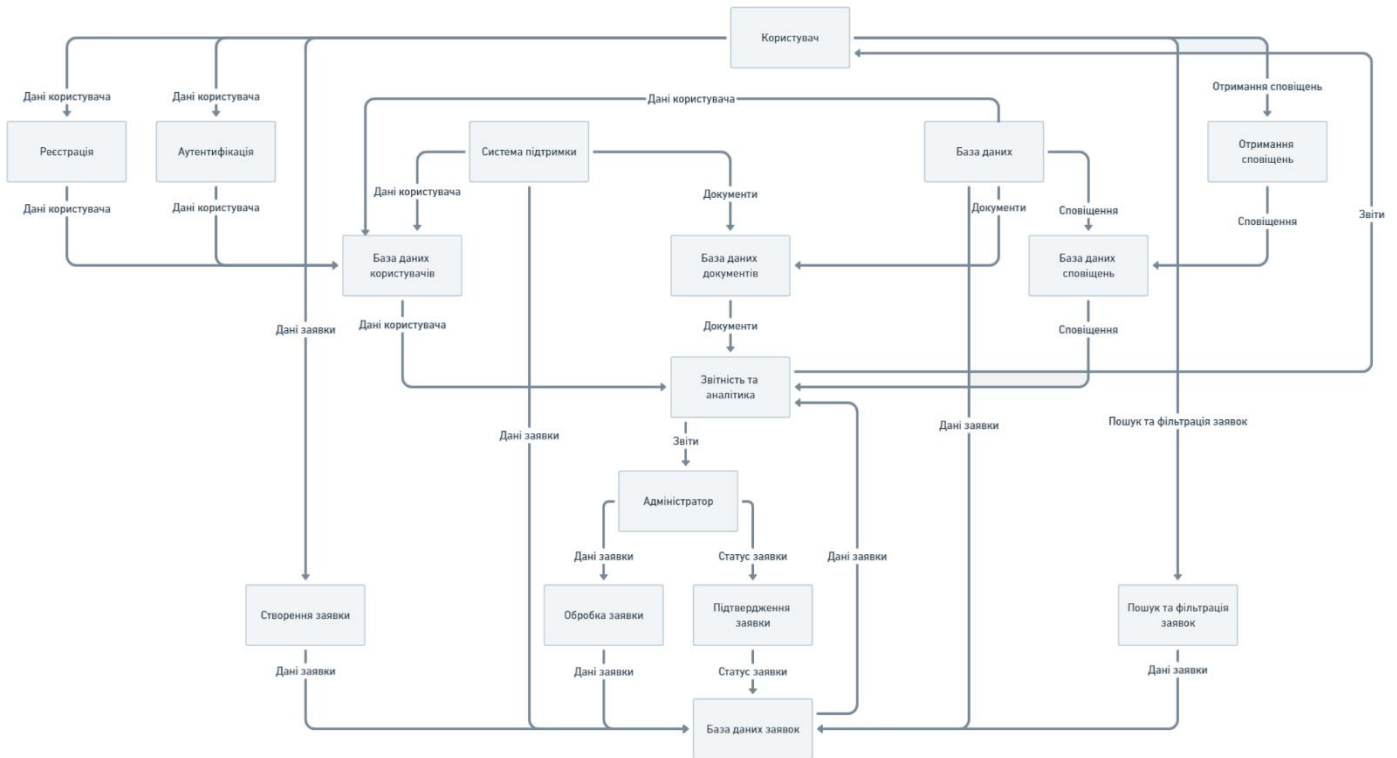


Рисунок 2.1 – Діаграма потоку даних системи

Дана діаграма відображає структуру та процеси розробленої експертної системи реєстрації авторського права. На ній чітко зазначені основні взаємодії між користувачами, сервером, базою даних та адміністративними модулями. Давайте розглянемо докладніше кожен елемент діаграми та їхню роль у системі.

Користувачі – здійснюють взаємодію з системою через процеси реєстрації та аутентифікації, вводять інформацію, яка потім передається у систему для подальшої обробки.

Система підтримки – відповідає за збір та обробку вхідної інформації від користувачів, також виконує функції зберігання документів, що надходять від користувачів, у базі даних документів.

База даних користувачів – зберігає всю інформацію про користувачів, необхідну для їх аутентифікації та управління обліковими записами.

База даних документів – зберігає всі документи та матеріали, які користувачі завантажують для процесу реєстрації.

Система звітності та аналітики – забезпечує збір даних та генерацію звітів для адміністраторів, що дозволяє моніторити та аналізувати роботу системи.

Адміністратор – відповідає за перегляд та обробку заявок, підтвердження обробки та управління статусами заявок через спеціалізовані інтерфейси.

Система повідомлень – інформує користувачів про статус їхніх заявок, нові повідомлення та оновлення в системі.

Процеси створення заявки та обробки заявки – відображають кроки, які користувачі проходять під час подачі заявки на реєстрацію, та як ці заявки обробляються в системі.

Модулі пошуку та фільтрації заявок – дозволяють користувачам та адміністраторам знаходити специфічні заявки та сортувати їх за різними параметрами.

Кожен з цих компонентів взаємодіє з іншими, створюючи інтегровану систему, яка включає в себе як клієнтську, так і серверну логіку. Це гарантує злагоджене та ефективне функціонування системи в цілому, забезпечуючи високу якість послуг для користувачів.

Ключова увага приділяється безпеці даних на кожному етапі їх обробки та зберігання, а також забезпечення доступності та цілісності системних процесів для забезпечення надійності реєстрації авторського права.

Крім того, при формулюванні вимог важливо враховувати потенційні побажання умовних користувачів системи, а також експертів у сфері авторських прав, щоб забезпечити відповідність системи реальним потребам та вимогам.

## **2.2. Вибір і обґрунтування моделі та методу штучного інтелекту**

Основним методом, який планується використовувати, є інтеграція моделі ШІ ChatGPT від OpenAI. ChatGPT є масштабованою моделлю мови, яка використовує трансформаторну архітектуру для генерації тексту та відповідей

на запитання, а також для розуміння контексту та взаємозв'язків у введеному тексті. Це поліпшує можливість системи адекватно реагувати на запити користувача та надавати корисні поради щодо процесу реєстрації авторського права.

Розглядаючи інші методи штучного інтелекту, можна порівняти ChatGPT з такими моделями й методами, що лежать в їх основі, як машинне навчання, нейронні мережі, глибоке навчання, рекурентні нейронні мережі та інші. Основні критерії порівняння включають:

- Здатність до самонавчання: ChatGPT має високу здатність до самонавчання, що дозволяє йому адаптуватися до нових завдань без необхідності великої кількості переднього навчання.
- Масштабованість: завдяки архітектурі трансформатора, ChatGPT може легко масштабуватися для виконання великих обсягів даних, що робить його підходящим для складних експертних систем.
- Відтворюваність результатів: незважаючи на динамічний характер моделі, результати, отримані з ChatGPT, є відтворюваними, що є важливим для наукових та технічних застосувань.
- Передача знань: ChatGPT може ефективно передавати знання з одного завдання на інше, що робить його гнучким рішенням для широкого спектру завдань.
- Інтерпретованість та здатність до пояснення: ChatGPT може надавати обґрунтування для своїх відповідей, що полегшує інтерпретацію результатів та допомагає користувачам розуміти рекомендації системи.

Також важливо врахувати можливі недоліки та обмеження використання ChatGPT, включаючи потребу в значних обчислювальних ресурсах та потенційні виклики з безпеки даних.

Порівняльний аналіз ChatGPT та інших моделей за методами штучного інтелекту було проведено у дослідженні, опублікованому на ResearchGate [16]. Більш детальний розбір даного питання можна знайти у порівняльних аналізах ChatGPT та інших чатботів на базі штучного інтелекту. В даному контексті

досить інформативні дані можна знайти також у дослідженнях, які порівнюють ChatGPT з людськими експертами [17, 18].

Таблиця 2.2 – Порівняння ChatGPT та конкуруючих моделей ШІ

Критерій	ChatGPT	Моделі на основі машинного навчання	Моделі на основі глибокого навчання	Рекурентні нейронні мережі
Здатність до самонавчання	Висока	Середня	Висока	Середня
Масштабованість	Висока	Низька	Висока	Середня
Відтворюваність результатів	Висока	Висока	Висока	Висока
Передача знань	Висока	Низька	Середня	Середня
Інтерпретованість та пояснення	Середня	Висока	Низька	Низька

Зокрема, порівнюючи заявлену модель ШІ ж з конкретними конкурентами, хоч й існують інші мовні моделі та платформи ШІ, такі як Rasa, Dialogflow від Google, або Luis від Microsoft, варто також зазначити, що ChatGPT вирізняється завдяки своїй потужності, гнучкості та спроможності генерувати природні, людські відповіді на багатьох мовах. Широкі можливості



ChatGPT дозволяє використання моделі для різноманітних задач, від простих чат-ботів до складних систем, які вимагають глибокого розуміння мови та генерації відповідей.

### **2.3. Проектування архітектури системи**

Проектування архітектури експертної системи є ключовим кроком у розробці програмного забезпечення. Архітектура системи визначає, як компоненти взаємодіятимуть між собою та з іншими системами. Вона включає в себе визначення основних компонентів системи, інтерфейсів, баз даних та інших технологічних рішень.

Дана експертна система, що розробляється, використовує дворівневу архітектуру, що включає клієнтську та серверну частини. Ця структура дозволяє нам ефективно розподіляти ресурси, забезпечуючи швидку відповідь системи на запити користувачів та високий рівень безпеки обробки даних. Дизайн користувацького інтерфейсу системи спрямований на забезпечення інтуїтивності та зручності використання, з особливою увагою до доступності функцій та легкості навігації.

Особливу увагу приділено інтеграції моделі штучного інтелекту ChatGPT від OpenAI, оскільки вона використовується для аналізу запитів користувачів і надання інформативних та точних відповідей, що спрощує процес реєстрації авторського права, тобто виконує одну з ключових ролей.

Також на архітектуру експертної системи впливають вимоги до модульності та розширюваності. Зокрема, система проектувалася таким чином, щоб вона могла легко адаптуватися до змін у законодавстві та ринкових умов, розширювати функціонал для реєстрації інших об'єктів інтелектуальної власності, а також дозволяла легко додавати нові функції або інтегрувати її з іншими системами. Стратегії розгортання та підтримки системи включають використання хмарних технологій для гарантування високої доступності та надійності, а також застосування методів CI/CD для ефективного управління змінами.

Ще одним з ключових аспектів даної системи є безпека. Для захисту інформації наших користувачів заплановано впровадження передових методів шифрування даних та системи аутентифікації.

Також приділено окрему увагу інтеграції можливостей аналітики та звітності, що дозволить збирати цінні дані про використання системи та оптимізувати її роботу.

Таким чином до основних компонентів експертної системи доцільно включити наступні модулі:

- Модуль обробки даних відповідає за збір, обробку та зберігання даних, які надаються користувачами або отримуються з інших джерел.
- Модуль аналізу даних відповідає за аналіз введених даних для визначення відповідності вимогам реєстрації авторського права.
- Модуль штучного інтелекту відповідає за використання ChatGPT для надання рекомендацій та підтримки користувачів у процесі реєстрації.
- Модуль взаємодії з користувачем відповідає за інтерфейси для введення даних, отримання рекомендацій та відстеження статусу заявки.
- Модуль зв'язку з зовнішніми системами відповідає за інтеграцію з системами реєстрації авторського права та іншими зовнішніми ресурсами.

Кожен з цих аспектів важливий для створення надійної, ефективної та орієнтованої користувача системи, тому слід максимально ретельно врахувати ці параметри в процесі розробки. Крім того, система потенційно може бути розширена іншими модулями, залежно від можливостей і потреб, особливо враховуючи перспективи ітеративного розвитку.

Саме тому важливим є вибір технологій для реалізації кожного з модулів, включаючи мови програмування, фреймворки, бази даних та інші інструменти [19]. Усі ці рішення цілком обґрунтовані з урахуванням вимог до функціональності, продуктивності, безпеки та інших аспектів системи.

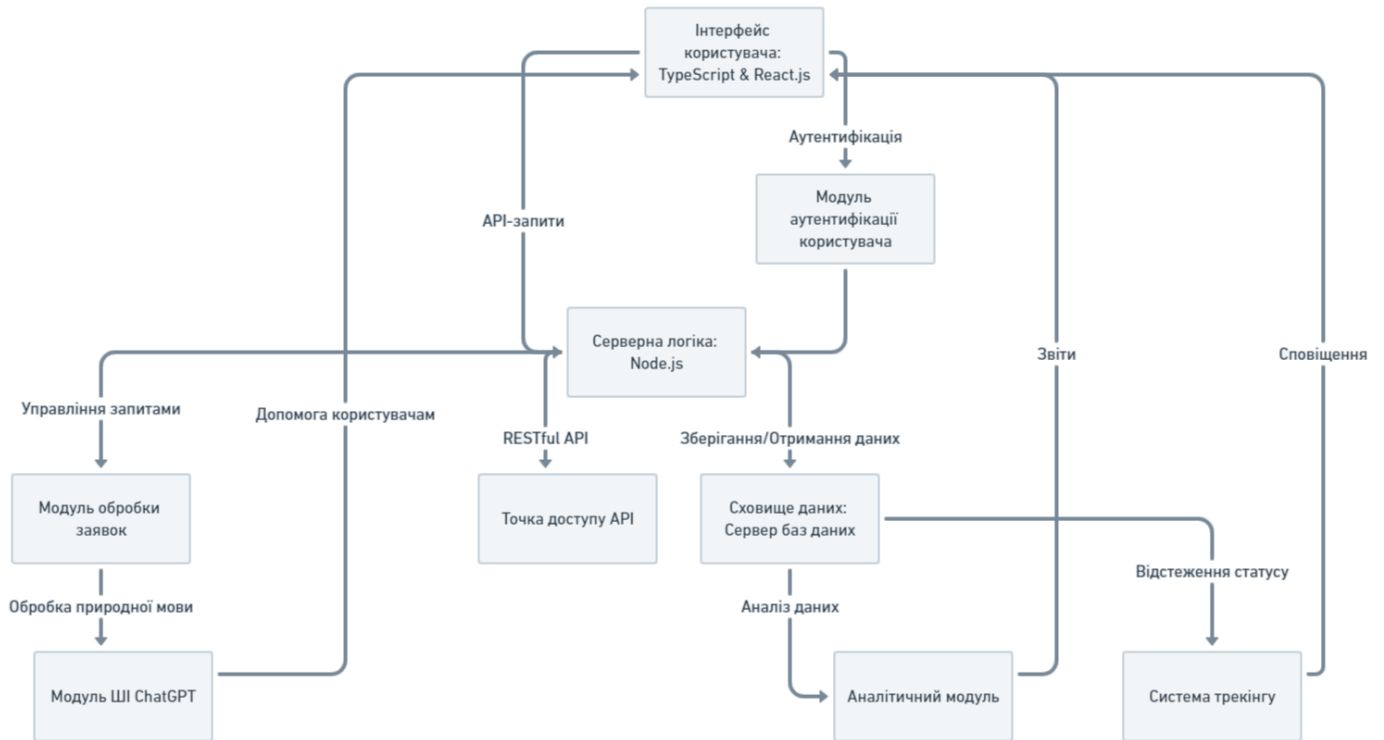


Рисунок 2.2 – Діаграма архітектури основних компонентів системи

Отже, обрану архітектуру для експертної системи, що розробляється, варто визначити як гібридну модульну архітектуру з інтеграцією штучного інтелекту. Ця архітектура експертної системи супроводу реєстрації авторського права являє собою гібридну, модульну структуру, що інтегрує сучасні веб-технології з моделями штучного інтелекту, об'єднує клієнтську частину, розроблену на React для динамічного та інтуїтивного інтерфейсу користувача, з серверною частиною на Node.js, що забезпечує швидку обробку запитів та взаємодію з базою даних. Інтеграція з моделями штучного інтелекту, такими як ChatGPT, забезпечує розширені можливості обробки природної мови, що покращує взаємодію з системою та допомагає у формулюванні заявок на реєстрацію авторського права. Модульність та розширюваність системи дозволяють легко адаптуватися до змінних потреб користувачів та законодавчих умов, в той час як заходи безпеки забезпечують захист даних та інформації.

## **2.4. Розробка методу автоматизації процесу реєстрації авторського права**

Процес розробки методу оптимізації та автоматизації процесу реєстрації в системі супроводу реєстрації авторського права охоплює кілька ключових етапів, які спрямовані на поліпшення взаємодії користувача з системою, спрощення процесу подачі заявок та забезпечення точності та ефективності обробки даних. Акцент зроблено саме на зниженні часу обробки заявок, підвищенні точності введених даних і поліпшенні загального досвіду користувачів.

На початковому етапі проводиться глибокий аналіз потреб користувачів та вивчення існуючих проблем у процесі реєстрації авторського права. Це включає збір відгуків від користувачів, аналіз звичайних помилок та вивчення їх причин. Особлива увага приділяється виявленню аспектів, які можна автоматизувати чи оптимізувати.

На основі зібраної інформації розробляються специфічні алгоритми та методики, спрямовані на оптимізацію та автоматизацію. Це включає в тому числі створення алгоритму для розпізнавання контексту та намірів користувача, що детально описується в наступному розділі. Крім того, важливу роль відіграє допоміжний алгоритм автоматичної перевірки введених даних на предмет помилок або відсутності інформації взагалі, що зменшує ймовірність відхилення заявок через неповноту або невірність даних. Алгоритми спрощують процес введення даних, надаючи користувачам шаблони та інтерактивні форми з підказками щодо заповнення.

Після розробки алгоритму та методик відбувається їх інтеграція з існуючою системою. Це включає оновлення серверної логіки, баз даних та інтерфейсу користувача. На цьому етапі важливо забезпечити, щоб нові функції інтегрувалися гладко, не порушуючи існуючу функціональність.

Новорозроблені функції підлягають ретельному тестуванню. Це включає юніт-тести для перевірки окремих компонентів, інтеграційні тести для оцінки взаємодії між різними частинами системи та приймальні тести з використанням

реальних сценаріїв користувачів. Валідація системи здійснюється з метою переконатися, що вона відповідає всім вимогам та очікуванням користувачів.

Після успішного тестування нові функції впроваджуються в основну версію системи. Після впровадження відбувається моніторинг роботи системи та збір даних про процес, що дозволяє виявити та виправити будь-які недоліки.

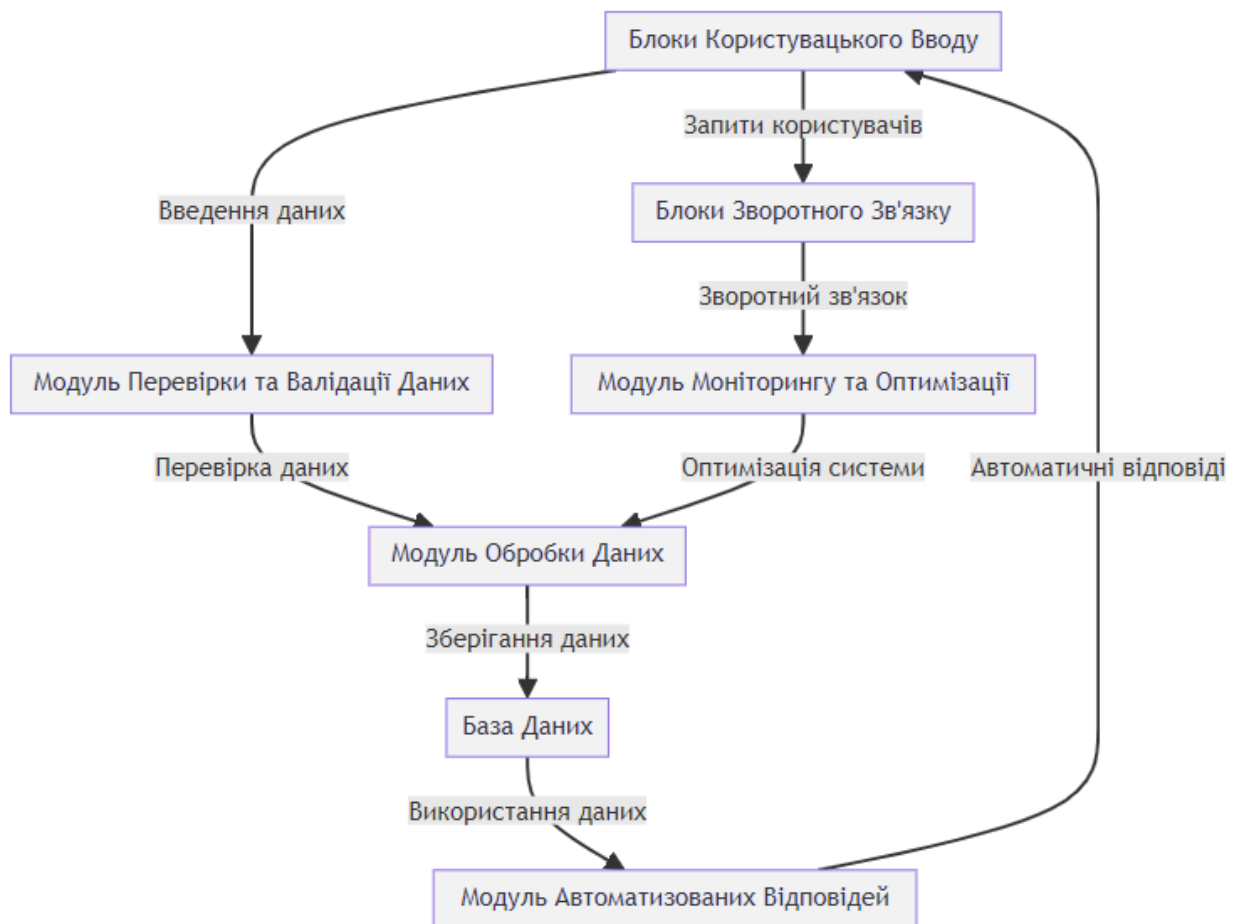


Рисунок 2.3 – Діаграма логічних модулів розробленого методу

Для демонстрації вищеописаного пропонується рис. 2.3, де зображено діаграму логічних модулів, що забезпечують реалізацію розробленого методу, який лежить в основі даної експертної системи.

Безпосередньо ж вищеописаний метод як логічну послідовність дій представлено у вигляді блок-схеми на рис. 2.4.

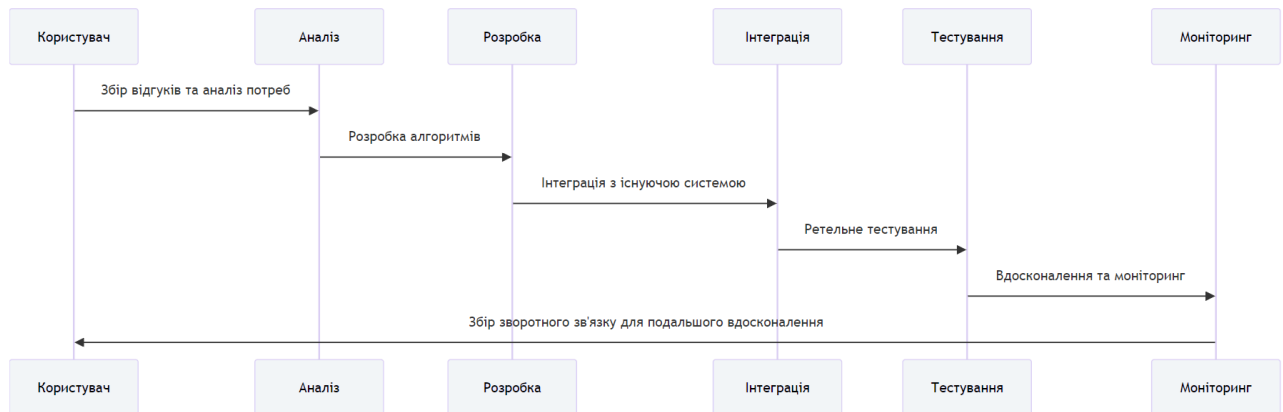


Рисунок 2.4 – Блок-схема методу оптимізації й автоматизації процесу реєстрації

Саме на основі отриманого зворотного зв'язку система постійно вдосконалюється. Це включає виправлення помилок, покращення існуючих алгоритмів та додавання нових функцій для забезпечення більш ефективної роботи та підвищення задоволеності користувачів.

Розробка такого методу вимагає глибокого розуміння потреб користувачів, вміння аналізувати та структурувати великі обсяги даних, а також компетенцій в області програмування та дизайну систем. Це комплексний процес, що включає різні аспекти розробки програмного забезпечення, від теоретичного аналізу до практичної реалізації та тестування.

## 2.5. Розробка методу розпізнавання контексту та намірів користувача

В рамках даного розділу доцільно розглянути розробку ключового методу, який забезпечує функціональність експертної системи. Його розробка включає в себе визначення відповідного алгоритму, тобто послідовності кроків, необхідних для виконання основних функцій системи, таких як аналіз даних, генерація рекомендацій та взаємодія з користувачем.

Для розробки алгоритму розпізнавання контексту та намірів користувача, що лежить в основі роботи системи, важливо зосередитись на ключових

функціональних аспектах системи. Ось деякі загальні напрямки, які використані в роботі:

Алгоритм розпізнавання контексту та намірів користувача – відповідає за первинну обробку даних, введених користувачами або отриманих з інших джерел. Він займається видаленням шуму, нормалізацією та перетворенням даних у формат, який може бути ефективно оброблений іншими компонентами системи, та включає:

- Збір даних від користувача;
- Перевірка даних на відповідність вимогам;
- Збереження даних у базі даних;

Аналіз даних забезпечує глибшу їх обробку для виявлення закономірностей, тенденцій чи аномалій. Він включає:

- Екстракція необхідної інформації;
- Аналіз та порівняння даних;
- Виявлення можливих проблем або відхилень;

Фактично логічним продовженням даного алгоритму є генерація рекомендацій – полягає у створенні персоналізованих рекомендацій для користувачів на основі оброблених та проаналізованих даних. Тобто висновки, що отримані в ході реалізації алгоритму розпізнавання контексту та намірів користувача, використовуються для надання релевантних порад, варіантів дій або рішень та включають:

- Аналіз отриманих даних;
- Генерація рекомендацій на основі аналізу;
- Повернення рекомендацій користувачу;

Взаємодія з користувачем в контексті описаного алгоритму спрямована на оптимізацію інтерфейсу та взаємодії системи з користувачем і включає:

- Прийом запитів від користувача;
- Обробка запитів та формування відповідей;

– Відправка відповідей користувачеві [20].

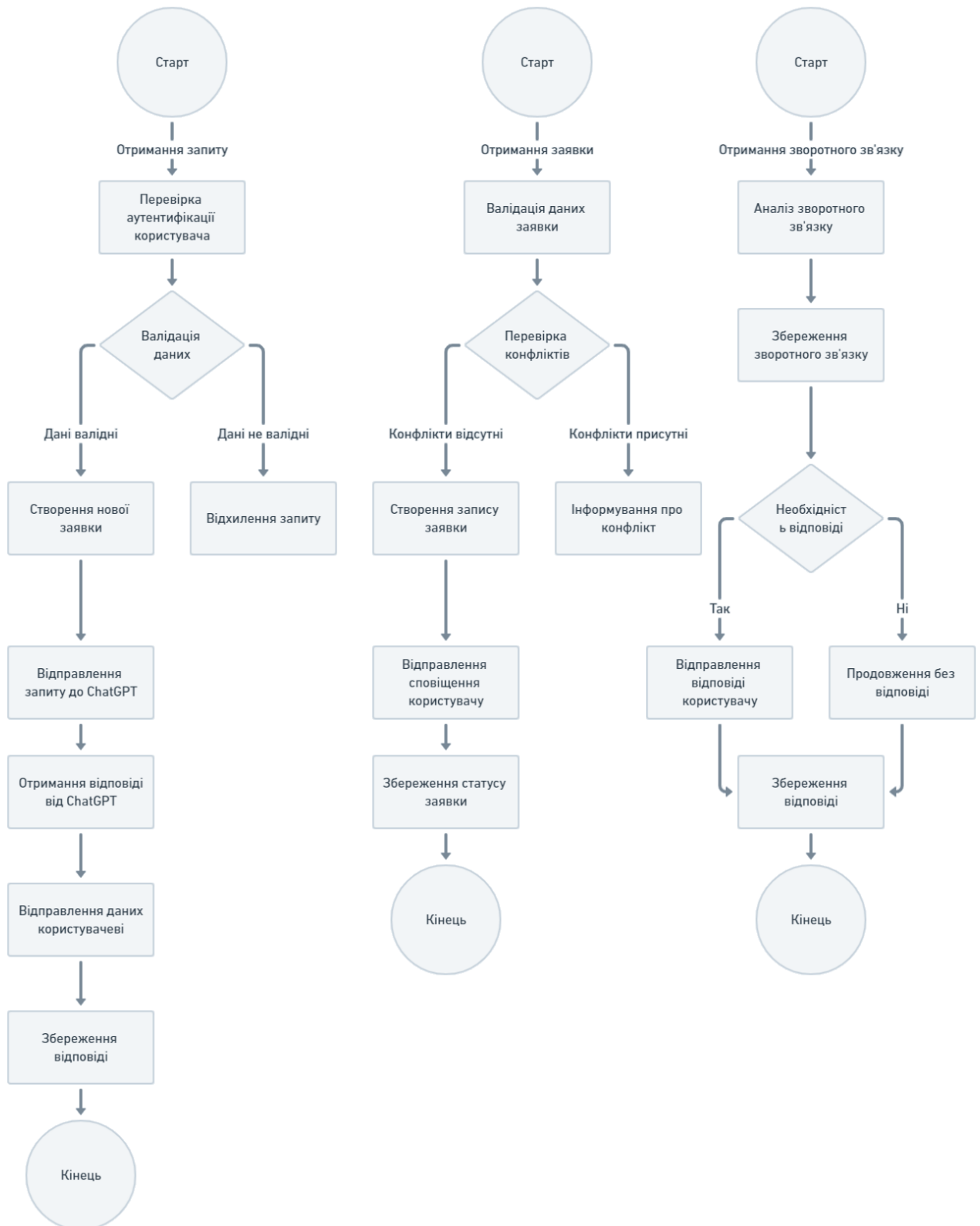


Рисунок 2.5 – Блок-схеми допоміжних алгоритмів системи



Для візуалізації та зрозуміння логіки основного алгоритму, тобто обробки даних та взаємодії компонентів, на рис. 2.3 наводиться блок-схема допоміжних алгоритмів експертної системи. Вона дозволяє швидко оцінити, як система обробляє дані та як вона відповідає на вхідні запити, виявляти можливі слабкі місця або області для оптимізації, а також слугує відмінним засобом документації для майбутніх відсилань та рефакторингу коду.

Крім основного алгоритму варто згадати також і допоміжні, які лежать в його основі та без яких повноцінна й ефективна робота описаної системи буде фактично неможливою:

- Алгоритм інтеграції з моделями штучного інтелекту – відповідає за інтеграцію з ШІ моделями, такими як ChatGPT, для покращення аналізу та обробки запитів користувачів.
- Алгоритм кешування даних – зменшує час відповіді системи шляхом зберігання часто запитуваних даних у кеші для швидшого доступу;
- Алгоритм оптимізації продуктивності – використовується для аналізу та покращення ефективності системи, включаючи швидкість обробки запитів та оптимізацію використання ресурсів;
- Алгоритм обробки помилок – виявляє та обробляє помилки в системі, забезпечуючи коректне повідомлення користувачів та відновлення роботи системи;

Кожен з цих алгоритмів розглядається з перспективи адаптивності відповідно до вимог і технічних особливостей проекту, проте їх ключова ціль – забезпечення ефективності алгоритму розпізнавання контексту та намірів користувача. Зокрема, безпосередньо ключовий алгоритм можна продемонструвати за допомогою блок-схеми на рис. 2.6.



Рисунок 2.6 – Блок-схема основного розробленого алгоритму

В результаті аналізу ключових алгоритмів, які лежать в основі описаної експертної системи, було визначено та фактично розроблено на їх основі алгоритм, який забезпечує не тільки функціональність системи, але й її ефективність та надійність. Особливу увагу було приділено алгоритму розпізнавання контексту та намірів користувача за рахунок залучення моделей штучного інтелекту, які значно покращують якість та швидкість обробки запитів користувачів. Також було враховано важливість оптимізації алгоритму для забезпечення високої продуктивності системи при обробці великих обсягів даних та запитів. На основі аналізу та тестування алгоритму, планується подальше його вдосконалення та оптимізація, щоб дана експертна система була не тільки функціональною, але й максимально зручною та ефективною для користувачів. Надалі в роботі увага акцентуватиметься саме на інтеграції розробленого алгоритму в загальну структуру системи, гарантуючи стабільну та ефективну взаємодію між компонентами.

## **2.6. Проектування інтерфейсу користувача**

Проектування інтерфейсу користувача є важливим аспектом для забезпечення зручності та ефективності взаємодії користувача з системою. Основні принципи дизайну інтерфейсу включають зрозумілість, консистентність, доступність та естетичну привабливість.

Мета на даному етапі полягає у створенні інтуїтивно зрозумілого, зручного та ефективного інтерфейсу, який відповідає наступним ключовим принципам, враховуючи закладений у них сенс:

- Чіткість та спрощення: інтерфейс має бути чистим і незавантаженим, з чітко визначеними функціональними зонами. Важливо уникнути перевантаження користувача зайвою інформацією або функціями.
- Інтуїтивність навігації: навігація має бути логічною та інтуїтивно зрозумілою. Ми плануємо використовувати узагальнені шаблони дизайну, які є знайомими для більшості користувачів.

- Швидкий доступ до ключових функцій: функції, що найчастіше використовуватимуться, такі як створення нової заявки або перегляд статусу заявки, мають бути легко доступні.
- Адаптивність: інтерфейс має бути адаптивним для різних типів пристроїв, включаючи настільні комп'ютери, ноутбуки, планшети та мобільні телефони.
- Інтерактивність: використання інтерактивних елементів, таких як модальні вікна, спливаючі підказки та анімації, може підвищити залученість користувачів та поліпшити загальний досвід роботи з системою.
- Підтримка зворотного зв'язку: інтерфейс має надавати користувачам чіткий зворотний зв'язок, наприклад, підтвердження дій або повідомлення про помилки.
- Доступність: забезпечення доступності для користувачів з різними здібностями, включаючи тих, хто використовує технології допоміжного доступу.
- Тестування інтерфейсу: оцінка зручності та ефективності інтерфейсу за допомогою автоматичного тестування та перевірок з реальними користувачами.
- Ітеративність оптимізації: регулярний збір відгуків та оптимізація дизайну на основі результатів тестування.

Детальний опис розділу включає аналіз основних вимог до інтерфейсу, вибір компонентів UI, проектування взаємодії користувача та відгук системи, створення прототипів та їх тестування з метою отримання зворотнього зв'язку від кінцевих користувачів [21, 22].

Зокрема, інтерфейс нашої системи розроблений таким чином, щоб максимально спростити та зробити інтуїтивно зрозумілим процес реєстрації авторського права. Головна сторінка має містити чітко визначені зони, включаючи панель навігації, де користувачі зможуть легко переходити між

різними розділами, такими як «Створення заявки», «Перегляд статусу заявки» та «Допомога та підтримка».

Екран створення заявки повинен мати просту форму з полями для введення необхідної інформації, такої як назва твору, опис, інформація про автора та вкладені файли. Користувачі повинні мати змогу легко завантажити необхідні документи, а система автоматично перевіряти коректність даних та повідомляти про помилки або відсутні поля.

Розділ «Перегляд статусу заявки» має містити інтерактивну таблицю, де користувачі можуть переглядати статус своїх заявок, включаючи етапи обробки та будь-які вказівки або коментарі від системи. Цей розділ також забезпечує можливість сортування та фільтрації заявок за різними параметрами.

Інтерфейс також включатиме розділ «Допомога та підтримка», де користувачі зможуть отримати відповіді на часті питання, переглянути законодавство чи корисні поради або зв'язатися з підтримкою. Цей розділ також міститиме вбудовану функцію чату зі штучним інтелектом для надання швидкої допомоги.

Більш змістовне уявлення про загальну структуру і взаємозв'язки компонентів інтерфейсу експертної системи, які стали основою для розробки клієнтської частини, демонструє рис. 2.7.



Рисунок 2.7 – Діаграма компонентів інтерфейсу системи

До того ж проектування інтерфейсу користувача даної системи включатиме наступні особливості, що відрізнятимуть його та сприятимуть підвищенню зручності та ефективності використання:

- Єдина колірна палітра та дизайнерські стандарти: інтерфейс слідує єдиній колірній палітрі та візуальним стандартам, що забезпечує консистентність та професійний вигляд. Це також сприяє більшій впізнаваності та легкості навігації для користувачів.
- Розумний розподіл контенту: інформація та елементи управління розташовані таким чином, щоб мінімізувати зайві рухи та кліки. Це дозволяє користувачам швидко знаходити потрібну інформацію та здійснювати необхідні дії.
- Інтерактивні елементи: кнопки, поля введення, випадаючі меню та інші інтерактивні елементи мають виразний дизайн та чіткі індикатори стану. Це робить інтерфейс більш інтуїтивним та зручним у використанні.
- Адаптивний дизайн: інтерфейс адаптований до різних розмірів екранів та типів пристроїв. Це гарантує, що користувачі можуть ефективно використовувати систему як на великих моніторах, так і на мобільних пристроях.
- Інтеграція з моделями штучного інтелекту: інтерфейс включає елементи, що забезпечують інтеграцію з моделями ШІ, такими як чатботи або інтелектуальні помічники, які можуть надавати підказки, допомогу в навігації та відповідати на запитання користувачів.
- Персоналізація: інтерфейс надає можливості для персоналізації, такі як зміна налаштувань, кастомізація відображення інформації та індивідуальні налаштування сповіщень.
- Простота оновлення та розширення: дизайн інтерфейсу розроблено таким чином, що він легко піддається оновленням і додаванню нових функцій, не порушуючи загальної консистентності та інтуїтивності користування.

Таким чином, весь інтерфейс розроблено з акцентом на зручність та простоту використання, забезпечуючи плавні переходи та чітку візуальну

організацію елементів, а його особливості спрямовані на створення високоякісного та ефективного інтерфейсу, який задовольнятиме потреби сучасних користувачів та сприятиме підвищенню загальної продуктивності системи. Адаптивний дизайн має гарантувати, що інтерфейс оптимально відображатиметься на різних пристроях та екранах. Цей інтерфейс сприятиме підвищенню ефективності процесу реєстрації авторських прав, знижуючи можливі бар'єри та забезпечуючи позитивний досвід користувача.

## **2.7. Висновки**

В даному розділі було здійснено докладний аналіз методологічних підходів до розробки експертної системи супроводу реєстрації авторського права. Було визначено основні вимоги до системи, які були взяті за основу при виборі методів штучного інтелекту. Обрано модель ChatGPT від OpenAI як основний метод штучного інтелекту для реалізації системи.

Також було розроблено загальну архітектуру системи, ключові алгоритми її роботи та проект інтерфейсу користувача. Всі ці елементи були детально розглянуті, а їх взаємозв'язок було проаналізовано.

Додатково було представлено графічні матеріали, що використані для ілюстрації ключових аспектів системи у подальшому проектуванні та розробці.

Проведений аналіз і розробка методики експертної системи для реєстрації авторського права відкриває нові можливості для ефективного та зручного використання таких систем в сучасному цифровому середовищі. Центральним елементом розробки стала інтеграція моделі штучного інтелекту ChatGPT, яка забезпечує високу точність і гнучкість в обробці запитів користувачів та генерації відповідей.

Розроблена архітектура системи забезпечує гнучкість, масштабованість та інтегрованість з існуючими базами даних, що є ключовим для адаптації до змінних умов ринку та законодавства. Важливим аспектом також є зосередження на безпеці обробки та зберігання даних користувачів.

Проектування інтерфейсу користувача з інтуїтивно зрозумілою та зручною навігацією сприяє підвищенню ефективності взаємодії з системою. Адаптивність інтерфейсу до різних пристроїв та платформ забезпечує доступність системи для широкого спектра користувачів.

Розроблені алгоритми та методи роботи системи демонструють здатність до ефективної обробки запитів, аналізу даних та надання точних рекомендацій. Важливою є можливість інтеграції системи з різними зовнішніми джерелами даних та сервісами.

Подальші кроки у розробці експертної системи включають детальне тестування і оптимізацію розроблених компонентів, забезпечення стабільності та надійності системи, а також адаптацію її функціональності до конкретних потреб користувачів. Важливо також зосередитися на зборі зворотного зв'язку від користувачів для постійного вдосконалення системи.

Загалом, дана методологія та вибрані технологічні рішення створюють міцну основу для подальшої розробки програмного забезпечення експертної системи. Рекомендується продовжити детальне вивчення представлених методологій та підходів, а також розглянути можливість проведення попереднього тестування розробленої системи з метою виявлення та усунення можливих проблем на ранніх етапах розробки, а також відкриваючи нові можливості для автоматизації та спрощення процесу реєстрації авторських прав.



## РОЗДІЛ 3

# РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЕКСПЕРТНОЇ СИСТЕМИ

### 3.1. Вибір технологій та інструментів розробки

При виборі технологій та інструментів для розробки експертної системи було вирішено використовувати сучасний стек технологій: TypeScript, React та Node.js, які є де-факто стандартом у сфері веб-розробки та підтримуються масштабною спільнотою розробників.

TypeScript як фактично надбудова над JavaScript дозволяє використовувати сучасні можливості ECMAScript та додає сувору типізацію, що сприяє підвищенню якості коду та на перспективу полегшує його рефакторинг [23]. Використання TypeScript у даному проекті дозволяє краще управляти складністю застосунку, особливо в частині клієнтської частини на React та серверної логіки на Node.js. Це важливо, оскільки дана система передбачатиме складну бізнес-логіку та велику кількість взаємодій з користувачами та базою даних.

React – бібліотека для побудови користувацьких інтерфейсів, забезпечує швидке відтворення змін та гарантує відмінну продуктивність завдяки віртуальному DOM [24]. До того ж він є одним з найпопулярніших JavaScript-фреймворків, що забезпечує широкий набір інструментів та компонентів для ефективної розробки.

Node.js – середовище виконання JavaScript на сервері, дозволяє розробляти масштабовані та продуктивні серверні застосунки. Було обрано саме цей варіант через його неблокуючу, подієво-орієнтовану архітектуру, яка ідеально підходить для реалізації ефективних мережевих додатків. Це дає нам змогу створити високопродуктивний сервер, здатний обробляти численні запити в режимі реального часу. [25]. API, розроблене на Node.js, буде служити мостом між серверною та клієнтською частинами, забезпечуючи безпечний обмін даними та взаємодію з базою даних.

Для розробки серверної частини системи також використовується фреймворк Express.js, який надає набір інструментів для побудови швидких та гнучких веб-додатків [26]. Для зберігання та керування даними було обрано базу даних MongoDB через її відмінну продуктивність, гнучкість у структуруванні даних і легкість масштабування [27]. Вона дозволяє зберігати JSON-подібні документи, що спрощує роботу з даними, особливо коли йдеться про невизначені або змінні схеми даних, які є типовими для динамічних інформаційних систем.

Що стосується інтеграції штучного інтелекту в нашу систему, одним з ключових рішень може бути використання фреймворку, зокрема TensorFlow.js. Це відкрита JavaScript-бібліотека, яка дозволяє розробляти та тренувати моделі машинного навчання прямо в браузері або на Node.js. Вона ідеально підходить для використання у поєднанні з нашим стеком технологій, оскільки пропонує гнучкість та масштабованість для реалізації алгоритмів ШІ, включаючи обробку природної мови та інших завдань, пов'язаних зі штучним інтелектом. Використання TensorFlow.js в комбінації саме з TypeScript та Node.js дозволяє нам розробляти високопродуктивні, масштабовані та безпечні рішення для інтеграції ШІ в описану експертну систему. Це також відкриває можливості для глибшої інтеграції з моделями машинного навчання та вдосконалення функціональності системи на основі даних, зібраних від користувачів.

Також було враховано необхідність інтеграції з зовнішніми сервісами та API, що вимагає наявності бібліотек для роботи з HTTP-запитами, таких як Axios або Fetch API. Це забезпечує можливість взаємодії з веб-сервісами для отримання та відправлення даних. Безпека даних буде забезпечена за допомогою протоколу HTTPS, токенів доступу (JWT) та інших методів шифрування та аутентифікації.

WebSocket буде використано для реалізації двосторонньої комунікації між клієнтом та сервером, зокрема для реалізації чат-бота та інших інтерактивних елементів.

Для створення інтуїтивно зрозумілих та зручних користувацьких інтерфейсів було обрано CSS-фреймворк Material-UI, який пропонує готові компоненти та дозволяє значною мірою прискорити процес розробки [28].

Використання Docker для контейнеризації допоможе у стандартизації та ізоляції середовища розробки, що гарантуватиме консистентність між розробкою та продакшн-середовищами. Це забезпечить легкість розгортання та масштабування експертної системи.

У контексті розробки сучасної експертної системи, важливим аспектом є не тільки вибір потужних та гнучких технологій, але й забезпечення їх інтеграції в єдиний ефективний процес розробки. У цьому відношенні TypeScript, React, та Node.js демонструють високу синергію, забезпечуючи гладке сполучення між клієнтською та серверною частинами. Це дає можливість реалізувати складні функціональні можливості з оптимізацією продуктивності та масштабованості.

Ключовим моментом у виборі технологій також є їх спроможність адаптуватися до змінюваних вимог та потреб користувачів. У цьому аспекті важливу роль відіграє гнучкість TypeScript, React та Node.js, які дозволяють швидко модифікувати систему з урахуванням нових функціональних вимог або змін у бізнес-процесах.

Також особливу увагу варто приділити безпеці даних та захисту інформації. Використання сучасних протоколів шифрування та аутентифікації дозволяє забезпечити високий рівень безпеки користувацької інформації та даних системи.

Додатково, важливою є інтеграція з різними зовнішніми сервісами та API для забезпечення більш широких функціональних можливостей. Така інтеграція дозволить збирати та обробляти більш різноманітні дані, що значно підвищить ефективність та корисність системи для користувачів.

Завершуючи, вибір цих технологій і інструментів не тільки відповідає сучасним вимогам до розробки веб-додатків, але й підкреслює інноваційний підхід до реалізації експертної системи супроводу реєстрації авторського права.

Такий підхід дозволяє створити надійну, гнучку та ефективну систему, готову до вирішення сучасних викликів та адаптації до майбутніх змін у сфері захисту інтелектуальної власності.

### **3.2. Розробка серверної частини системи**

В рамках даного дослідження основою для серверної частини є Node.js, який є популярним вибором для багатьох веб-розробок завдяки його швидкості та ефективності обробки асинхронних запитів, що особливо важливо для обробки великої кількості одночасних запитів користувачів та забезпечення швидкого відгуку системи [25]. Платформа Express.js використовується для створення веб-сервера та роутингу запитів, що дозволяє ефективно обробляти вхідні HTTP-запити [30].

Для взаємодії з базами даних використовується MongoDB, що є однією з найпопулярніших NoSQL баз даних, оскільки дозволяє ефективно зберігати та обробляти неструктуровані дані. Інтеграція з MongoDB здійснюється через обгортку Mongoose – ODM (Object Document Mapping), бібліотеку для Node.js, яка спрощує взаємодію з базою даних [31]. Вони обрані через високу продуктивність, масштабованість та гнучкість, які вони пропонують.

Обробка запитів реалізується через розробку RESTful API, який дозволяє обробляти CRUD-операції (створення, читання, оновлення та видалення) для ресурсів системи. Для валідації вхідних даних використовується бібліотека Joi, яка дозволяє визначати схеми валідації для JavaScript об'єктів [32].

Безпека є важливим аспектом серверної частини, тому використовується Passport.js для аутентифікації та авторизації. Це забезпечує захист від несанкціонованого доступу до чутливих ресурсів системи [33]. Крім того систему аутентифікації та авторизації було посилено використанням токенів доступу (зокрема, JWT – JSON Web Tokens) для забезпечення ще більш безпечного доступу до API. Описане включає захист від несанкціонованого доступу та забезпечення конфіденційності даних користувачів.

Тестування та оптимізація є важливими для забезпечення надійності та високої продуктивності серверної частини. Для юніт-тестування використовуються Jest, яка є популярною бібліотекою для тестування TypeScript-коду [34].

Для відстеження стану та продуктивності сервера передбачається впровадження системи логування та моніторингу. Це дозволить швидко виявляти й усувати практично будь-які проблеми, а також аналізувати тенденції використання системи.

Оптимізація продуктивності сервера включає оптимізацію запитів до бази даних, кешування та інші техніки для зменшення часу відгуку сервера та збільшення його здатності обробляти більшу кількість запитів за одиницю часу.

Як результат, було розроблено RESTful API для взаємодії між клієнтською та серверною частинами. API включає ряд точок доступу для обробки запитів, пов'язаних з реєстрацією авторського права, перевіркою статусу заявок, управлінням користувацькими профілями та іншими функціями.

Серверна частина системи в реальних умовах має бути розгорнута у хмарному середовищі, що забезпечить їй масштабованість та високу доступність. Використання хмарних технологій також сприяє гнучкому управлінню ресурсами та оптимізації витрат.

Таким чином, описано всі ключові аспекти розробки серверної частини, від елементи архітектури та безпеки від інтеграції та моніторингу. Кожен з цих компонентів є важливим для забезпечення стабільності, безпеки та ефективності системи, що є критичними для успішного функціонування експертної системи супроводу реєстрації авторського права, особливо в контексті обробки персональних або конфіденційних даних.

Водночас у контексті розробки серверної частини системи, особлива увага була приділена впровадженню розширених функцій та технологій. Зокрема, було реалізовано мікросервісну архітектуру, що дозволяє розподіляти навантаження та покращує масштабованість системи. Кожен мікросервіс

відповідає за окремий аспект функціональності системи, забезпечуючи більш гнучке управління та легшість у внесенні змін.

Зокрема, мікросервісна структура включає такі компоненти:

- Мікросервіс аутентифікації та авторизації – управляє реєстрацією користувачів, входом у систему, перевіркою прав доступу та управлінням обліковими записами. Взаємодіє з базою даних користувачів для зберігання та витягування інформації про користувачів.
- Мікросервіс обробки оаних – сновний механізм системи, який займається аналізом запитів користувачів, обробкою даних реєстрації авторського права, та формуванням відповідей. Взаємодіє з модулем штучного інтелекту та базою даних авторських прав.
- API Gateway (Шлюз API) – центральна точка входу для всіх запитів до системи. Відправляє запити від користувача до відповідних мікросервісів, а також управляє маршрутизацією та балансуванням навантаження.
- Мікросервіс управління базою даних: Управляє всіма операціями з базами даних, забезпечуючи зберігання та доступ до даних користувачів, даних авторського права тощо. Включає механізми кешування та оптимізації запитів.
- Мікросервіс логування та моніторингу – збирає логи з усіх компонентів системи, відстежує стан системи та аналізує ефективність роботи.
- Мікросервіс роботи з зовнішніми API – управляє інтеграцією з зовнішніми сервісами, як-от системи платежів, зовнішні бази даних, API для перевірки даних тощо.

Дещо детальніше взаємозв'язки між мікросервісами демонструє рис. 3.1, ілюструючи їхнє розподілення на логічні блоки, системність і послідовність взаємодії. Такий підхід дає можливість легко підтримувати систему та ізолювати потенційні ризики, не нагромаджуючи логіку виключно в рамках одного основного додатку.

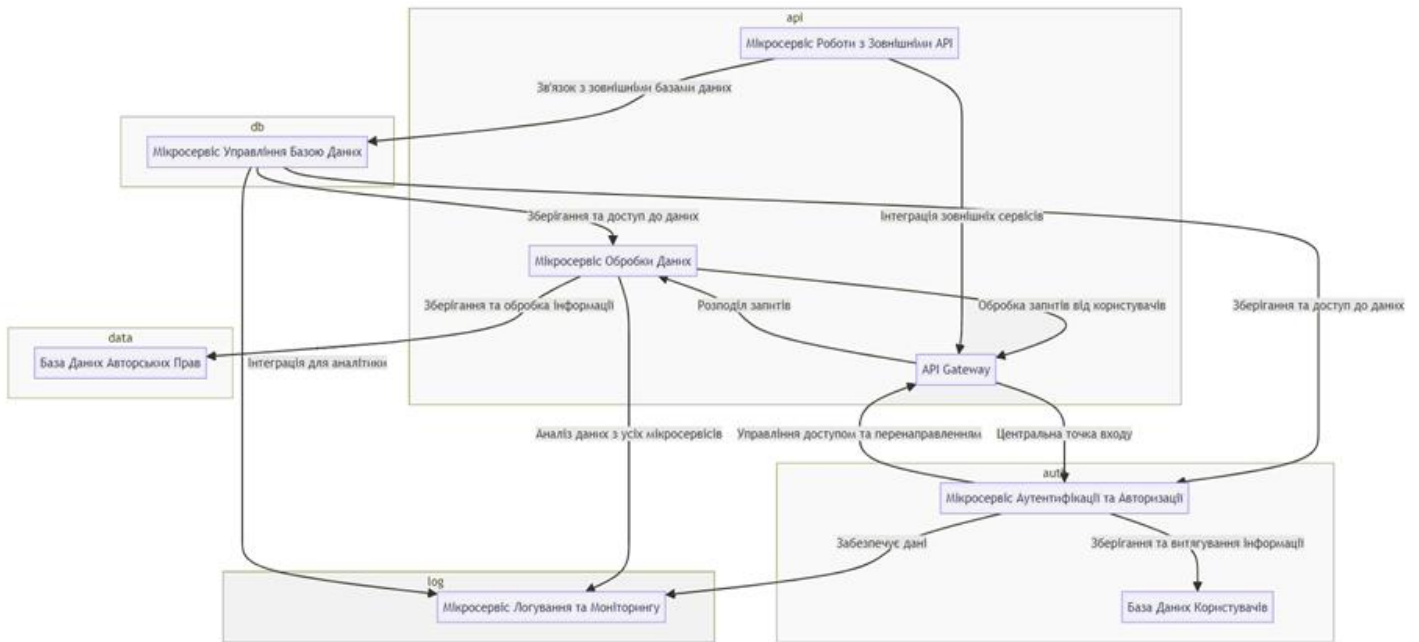


Рисунок 3.1 – Діаграма взаємодії мікросервісів

Для обробки запитів в реальному часі та забезпечення високої продуктивності було впроваджено використання Node.js заснованих веб-сокетів. Це дозволяє серверу та клієнтам взаємодіяти в двосторонньому порядку, забезпечуючи швидкий обмін даними без необхідності постійного запиту стану з боку клієнта.

Також було здійснено інтеграцію рішень з машинним навчанням та аналітикою даних. Це включає використання спеціалізованих бібліотек та інструментів для аналізу та обробки великих обсягів даних, що значно покращує здатність системи до розумного аналізу та надання корисних інсайтів.

З метою оптимізації обробки запитів до бази даних, було реалізовано кілька технік, таких як індексація, реплікація та шардування. Це дозволяє збільшити швидкість відповідей та ефективність зберігання даних, забезпечуючи при цьому високий рівень масштабованості.

Крім того, було впроваджено авансовані механізми кешування на сервері, що дозволяє зменшити навантаження на базу даних та прискорити доступ до

часто використовуваної інформації. Використання таких рішень, як Redis чи Memcached, забезпечує високу швидкість доступу до даних та підвищує загальну продуктивність системи.

Щодо безпеки, було впроваджено розширені механізми захисту, включаючи SSL/TLS шифрування, захист від CSRF (Cross-Site Request Forgery) та DDoS-атак. Також було встановлено автоматизовані системи виявлення та реагування на інциденти безпеки, що забезпечує надійний захист системи від зовнішніх загроз.

Відповідно, розробка серверної частини системи включає в себе не тільки створення міцної та ефективної основи для функціонування системи, але й забезпечує її масштабованість, безпеку та високу продуктивність. Такий підхід дозволяє забезпечити високу якість обслуговування та задоволення потреб користувачів, враховуючи зростаючі вимоги до сучасних веб-систем.

Крім того, у виборі технологій для клієнтської частини в наступному розділі важливу роль відіграє саме фактор врахування технології розробки серверної частини, оскільки єдність технологічного стеку спрощує інтеграцію між клієнтською та серверною частинами, а також полегшує подальшу підтримку та розширення системи.

### **3.3. Розробка клієнтської частини системи**

Розробка клієнтської частини системи здійснювалася на основі бібліотеки React.js, яка є однією з найпопулярніших у світі для створення користувацьких інтерфейсів. Використання React.js дозволило створити інтерактивний і дружній до користувача інтерфейс, що легко відгукується на дії користувача [24]. Зокрема, для модульності та легкості підтримки коду, а також для швидкого відгуку інтерфейсу було використано компонентний підхід. Для реалізації динамічних елементів інтерфейсу, таких як форми, списки, кнопки, було використано сторонні бібліотеки, такі як Material-UI та Ant Design [28, 35].

Також була використана бібліотека Redux для управління станом додатку, що дозволяє забезпечити консистентність даних між різними компонентами



системи та відповідний доступ до них по всьому додатку. Щоб вирішити питання управління асинхронними запитами та впроваджувати ретельну обробку помилок, було вирішено використовувати проміжне програмне забезпечення (middleware). Зокрема, в даному випадку було обрано Redux Thunk як один з найпопулярніших middleware в екосистемі Redux, що фактично дозволяє вам прописувати дії (action), які повертають функцію замість об'єкта. Ця функція може виконувати асинхронні операції та відправляти дії в міру необхідності. Thunk є міжпрограмною прокладкою, яка дозволяє відкладати відправку дії, або відправляти декілька дій, якщо це потрібно.

Для взаємодії з серверною частиною було використано Fetch API, який у поєднанні з Redux та таким middleware як Redux Thun дозволяє виконувати HTTP-запити до сервера, обробляти відповіді та, як наслідок, виконувати асинхронні дії для управління відповідним станом додатку [36]. До того ж використання Fetch API відбувається у відповідності до політик безпеки, таких як CORS (Cross-Origin Resource Sharing), що дозволяє виконувати запити до сервера з дотриманням належних заходів безпеки.

Головною частиною клієнтської частини є модуль введення даних користувачем для формування запиту до експертної системи. Було створено ряд форм для введення даних, які допомагають користувачеві структуровано вводити необхідну інформацію. Крім того, було розроблено модуль відображення результатів аналізу, який представляє отримані від сервера дані у зручному для користувача форматі.

Для забезпечення безпеки даних користувача було використано SSL-шифрування, а також були використані сучасні методи аутентифікації та авторизації, зокрема OAuth 2.0. Це дозволяє забезпечити конфіденційність даних користувача та запобігти несанкціонованому доступу до системи.

Крім того виконано юніт-тестування компонентів інтерфейсу за допомогою бібліотеки Jest. Також було оптимізовано продуктивність за допомогою аналізу виконання коду та оптимізації рендерингу. Додаткова увага була приділена оптимізації завантаження сторінок, використовуючи ледаче

завантаження (lazy loading) та розділення коду (code splitting) [37], оскільки за відсутності подібних підходів цілком імовірними можуть постати проблеми продуктивності додатку при великій кількості користувацьких запитів за певний проміжок часу. Також була реалізована підтримка офлайн-режиму за допомогою Service Workers [38]. Наприклад, при реєстрації авторського права, Service Workers можуть кешувати форми та інші важливі ресурси, щоб користувачі могли заповнювати або переглядати їх без затримок. Тим не менш, не дивлячись на ряд оптимізацій, важливо регулярно профілювати продуктивність інтерфейсу та оптимізувати найбільш важкі компоненти.

Ще одним важливим моментом є врахування різноманітності браузерів і пристроїв, що потенційно може ускладнити підтримку однаково високої якості інтерфейсу для всіх користувачів.

В ході аналізу вищевикладеного доцільно акцентувати увагу на важливості створення ефективного, безпечного та орієнтованого на користувача клієнтського інтерфейсу, який є ключовим для успіху такої експертної системи. Кожен з цих аспектів сприяє підвищенню зручності, ефективності та безпеки використання системи кінцевими користувачами. Що стосується потенційних перешкод, то вони вимагають окремого уважного планування, ретельного тестування та готовності до адаптації стратегій розробки в майбутньому відповідно до виникаючих викликів, втім даний додаток враховує такі ризики, що вчергове доводить його актуальність і перспективність, в тому числі з точки зору клієнтського досвіду.

У рамках розробки клієнтської частини системи, було приділено особливу увагу розширенню функціональності інтерфейсу та забезпеченню його адаптивності. Це включало інтеграцію розширеного набору інтерактивних елементів, таких як слайдери, інтерактивні графіки та карти, які забезпечують більш залучений та інтуїтивний досвід користувача. Використання таких елементів дозволяє користувачам більш глибоко зануритися в аналіз та оцінку інформації, що надається системою.

Особлива увага була приділена оптимізації відгуку інтерфейсу на дії користувачів. Використовуючи сучасні техніки JavaScript та CSS анімації, було досягнуто плавності та швидкості реакції інтерфейсу, що підвищує загальну зручність використання системи. Такі покращення відіграють ключову роль у створенні позитивного першого враження та утриманні уваги користувачів.


Крім того, для підвищення доступності та універсальності системи, було впроваджено підтримку різних розмірів екранів та пристроїв. Це означає, що система однаково добре працює на різноманітних пристроях, включаючи настільні комп'ютери, планшети та смартфони, забезпечуючи користувачам зручний доступ до системи з будь-якого пристрою.

Також було зосереджено увагу на забезпеченні високої продуктивності системи, особливо при великих обсягах даних. Використання технік оптимізації, таких як віртуалізація списків та ледаче завантаження ресурсів, зменшує час завантаження та покращує загальну продуктивність системи.

Водночас було впроваджено передові техніки безпеки, включаючи використання HTTPS для шифрування трафіку та захисту даних користувачів, гарантує високий рівень безпеки і конфіденційності. Такий підхід забезпечує захист від загроз, таких як перехоплення даних та несанкціонований доступ, та є критично важливим для систем, які обробляють конфіденційну інформацію, що пояснюється детальніше далі у відповідному розділі про безпеку.

Зокрема, результат розробки наочно демонструють знімки екрану під час виконання програми на вкладках створення нової заявки, перегляду наявних заявок, а також профіля користувача, що зображені відповідно на рис. 3.1, де відображено інтуїтивно зрозумілий інтерфейс для створення заявки та отримання поточних рекомендацій, 3.2, де відображено інтерфейс перегляду наявних у профілі заявок з різним статусом та можливістю детальнішого перегляду або видалення, та 3.3, де зображено інтерфейс профіля користувача з обліковими даними та можливістю їх змінити.

**РЕЄСТРАЦІЯ АВТОРСЬКОГО ПРАВА**

 UKR

[ПРОФІЛЬ](#) [НОВА ЗАЯВКА](#) [МОЇ ЗАЯВКИ](#) [ДОВІДКА](#) [ПРО СЕРВІС](#) [КОНТАКТИ](#)

## Створення заявки на реєстрацію

Назва твору:

Опис твору:

Дата створення або публікації:

Ім'я особи та внесок:

Електронна пошта:

Завантажте файли:

[Виберіть файли](#)

[Подати заявку](#)

Надання рекомендацій:

[Отримати рекомендації](#)

Рисунок 3.1 – Знімок екрану на етапі створення заявки

РЕЄСТРАЦІЯ АВТОРСЬКОГО ПРАВА UKR

ПРОФІЛЬ НОВА ЗАЯВКА **МОЇ ЗАЯВКИ** ДОВІДКА ПРО СЕРВІС КОНТАКТИ

### Мої заявки


Назва твору ▲	Дата подання ▼	Статус заявки ▲	Дії
Твір "Дослідження кіберпростору"	23/11/2023	Очікує на розгляд	<a href="#">Переглянути</a> <a href="#">X</a>
Комп'ютерна програма "Пластизатор"	25/11/2023	Чернетка	<a href="#">Переглянути</a> <a href="#">X</a>
Картина "Море яблук"	10/11/2023	Прийнято рішення	<a href="#">Переглянути</a> <a href="#">X</a>
Мелодія "Спокій"	08/11/2023	Очікує на оплату держмита	<a href="#">Переглянути</a> <a href="#">X</a>
Документація "РПШ: Аналіз рушія"	20/10/2023	Видано свідоцтво	<a href="#">Переглянути</a> <a href="#">X</a>

« 1 »

Рисунок 3.2 – Знімок екрану при перегляді наявних заявок

РЕЄСТРАЦІЯ АВТОРСЬКОГО ПРАВА UKR

ПРОФІЛЬ НОВА ЗАЯВКА МОЇ ЗАЯВКИ ДОВІДКА ПРО СЕРВІС КОНТАКТИ



[Змінити фото](#)

### Профіль користувача

Ім'я:	Віталій Нестерук
Email:	vitalii_nesteruk@vntu.edu.ua
Дата народження:	01.01.1995
Адреса:	м. Вінниця, вул. - , буд. -
Кількість наявних заявок:	5
Отримані свідоцтва:	1
Ліцензія діє до:	31.12.2024

[Редагувати](#)
[Вийти](#)

Рисунок 3.3 – Знімок екрану при перегляді профілю користувача

Всі вищеперелічені заходи в комплексі забезпечують створення ефективного, безпечного, високофункціонального та мінімалістичного клієнтського інтерфейсу, який відповідає високим стандартам зручності та ефективності, задовольняючи потреби та очікування сучасних користувачів.

### **3.4. Інтеграція моделей штучного інтелекту**

Центральним елементом експертної системи є модель штучного інтелекту ChatGPT від OpenAI, яка відіграє ключову роль у процесі аналізу запитів користувачів та формуванні відповідей. Інтеграція ChatGPT була виконана через RESTful API, наданий OpenAI. Для забезпечення надійної та безпечної взаємодії між клієнтською та серверною частиною, було розроблено набір конкретних запитів API, які дозволяють передавати дані від користувача до ChatGPT, та отримувати відповіді [39].

Для оптимізації взаємодії з моделлю ChatGPT та забезпечення швидкодії відгуку системи, було використано кешування запитів. Це дозволяє зберігати результати попередніх запитів та надавати їх користувачам без додаткових запитів до сервісу OpenAI, тим самим зменшуючи час відгуку системи та збільшуючи ефективність використання ресурсів.

Також було розроблено механізм версіонування моделі ШІ, який дозволяє легко інтегрувати нові моделі та версії моделей без переривання роботи системи. Це забезпечує гнучкість системи та можливість швидкого впровадження оновлень.

До того ж, будь-яка інтеграція моделі штучного інтелекту вимагає ретельного планування та оптимізації взаємодії між різними компонентами системи, включаючи модулі вводу даних, обробки запитів та надання відповідей і т.п. Вона також включає адаптацію моделі до змінюваних потреб та умов, гарантуючи її здатність навчатися та розвиватися відповідно до нових даних та користувацьких взаємодій. Це забезпечує системі гнучкість та масштабованість, необхідні для адаптації до швидкозмінних умов ринку та технологічних трендів. Водночас важливою частиною інтеграції є також

забезпечення безпеки даних користувачів та відповідності системи вимогам конфіденційності та захисту інформації. Це включає впровадження заходів для захисту персональних даних та запобігання будь-яким можливим витокам інформації

Зокрема, більш детально місце моделі штучного інтелекту як модуля в експертній системі загалом та у взаємодії з іншими компонентами опосередковано продемонстровано на рис. 3.4. Крім того, проілюстровано основні методи взаємодії модулів, назви яких відображають закладену в них логіку безпосередньо у кодї.

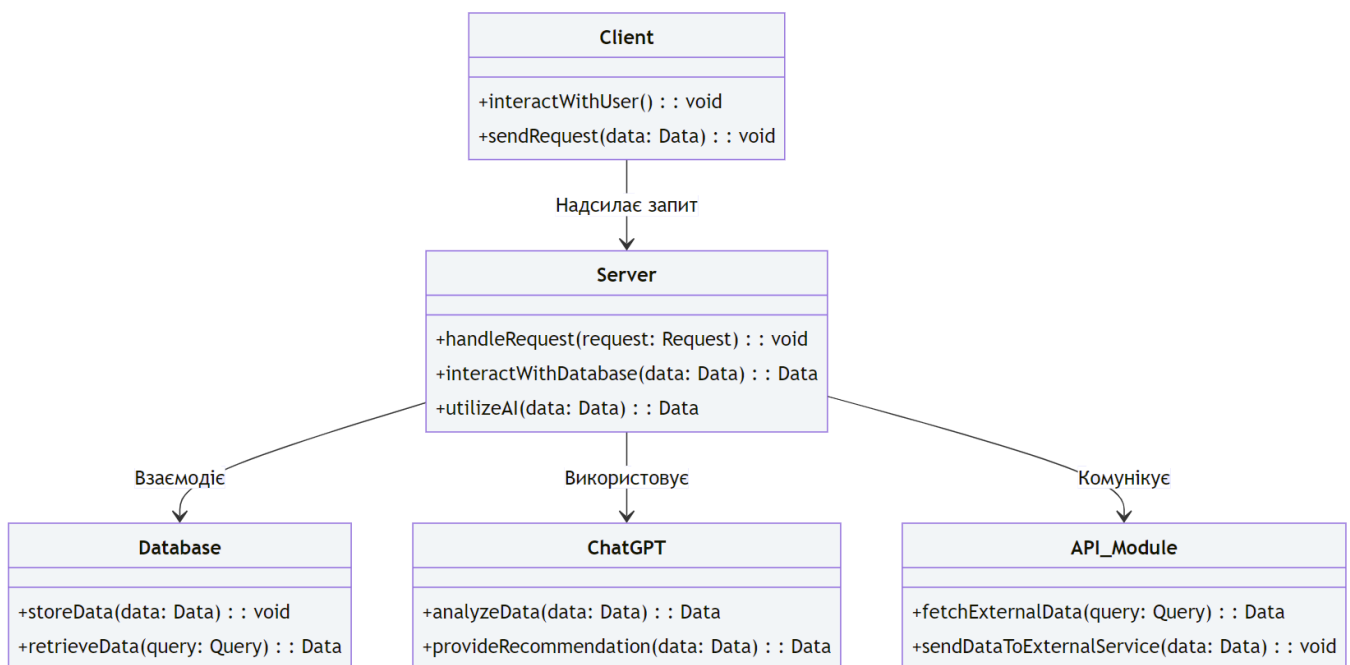


Рисунок 3.4 – Діаграма класів основних модулів системи

Ключовий фокус на даному етапі припадає на впровадженні передових технологій ШІ для підвищення ефективності та зручності обслуговування користувачів. Оптимізація використання моделі ШІ, включаючи кешування запитів та масштабування ресурсів, а також її тренування відіграють ключову роль у забезпеченні швидкого та точного обслуговування. Це включає інтеграцію з системами рекомендацій, які використовують ШІ для надання

персоналізованих порад та вказівок, а також автоматизацію обробки запитів для підвищення ефективності системи.

Застосування технологій обробки природної мови (NLP) значно покращує взаємодію з користувачами, забезпечуючи зрозуміле та зручне спілкування. Важливим аспектом є також забезпечення безпеки даних користувачів через шифрування та використання надійних методів аутентифікації та авторизації. Постійний моніторинг та оцінка ефективності моделей ШІ дозволяють своєчасно вносити необхідні корективи та оновлення, забезпечуючи актуальність та відповідність системи сучасним вимогам та трендам.

Крім того, в рамках інтеграції моделей штучного інтелекту, особливу увагу було приділено процесу тренування та налаштування моделі ChatGPT для забезпечення її максимальної ефективності в контексті експертної системи. Тренування моделі включало підбір та аналіз великої кількості даних, що відображають типові запити користувачів і потреби в контексті авторських прав. До того ж, як базу знань для тренування було використано набір даних з Державного реєстру свідоцтв про реєстрацію авторського права на твір – сукупність офіційних відомостей про реєстрацію авторського права на твір та факт і дату оприлюднення твору, що постійно зберігаються на електронному і паперовому носіях станом на 18 листопада 2022 року. Це дозволило моделі дослідити значну кількість реальні результатів реєстрації, що включають дані, які необхідні для коректного заповнення заявок, і як наслідок – краще розуміти специфічну термінологію та контекст, що є ключовим для ефективного аналізу запитів користувачів, надання відповідей та формування необхідних для реєстрації документів.

Важливим аспектом тренування моделі було також врахування етичних та правових норм. Це забезпечує, що відповіді моделі відповідають не тільки технічним, але й етичним стандартам, уникаючи надання порад, які можуть бути юридично некоректними або морально сумнівними.

Додатково було впроваджено адаптивне навчання моделі, що дозволяє їй постійно вдосконалюватися на основі зворотного зв'язку від користувачів. Це



включає аналіз відгуків користувачів на відповіді системи та їх використання для подальшого вдосконалення алгоритмів обробки запитів.

Також була виконана робота по оптимізації взаємодії моделі з іншими компонентами системи, забезпечуючи плавне та інтуїтивно зрозуміле взаємодія з користувачами. Це включає інтеграцію з модулями обробки запитів та візуалізації даних, дозволяючи системі надавати користувачам не тільки текстові відповіді, але й графічні зображення, таблиці та діаграми для кращого розуміння інформації.

Завдяки цим зусиллям, модель штучного інтелекту ChatGPT стала фундаментальною частиною експертної системи, забезпечуючи високу точність та релевантність відповідей на запити користувачів. Це відкриває шлях для нових можливостей у поліпшенні якості обслуговування та задоволення потреб користувачів в області реєстрації авторських прав. Втім для огляду залишилася ще одна ключова категорія – безпека та захист даних, що вичерпно викладено в наступному розділі.

### **3.5. Безпека та захист даних у системі**

Забезпечення безпеки та захисту даних є критично важливим аспектом при розробці експертної системи супроводу реєстрації авторського права. Система буде зберігати та обробляти конфіденційну інформацію користувачів, тому необхідно вжити всіх можливих заходів для забезпечення її безпеки.

Одним із ключових моментів є використання протоколу HTTPS для шифрування даних під час передачі між клієнтом та сервером. Також важливо реалізувати механізми аутентифікації та авторизації для обмеження доступу до системи та захисту від несанкціонованого доступу. Важливе значення має застосування хешування паролів і використання солі для підвищення рівня безпеки.

Щоб забезпечити конфіденційність даних, необхідно використовувати сильні алгоритми шифрування для зберігання даних на сервері. Це включає в

себе використання стандартів шифрування, таких як AES або RSA, для захисту інформації в базах даних.

Також слід врахувати розробку політик безпеки та процедур для реагування на інциденти безпеки, що допоможе визначити та реагувати на можливі загрози вчасно. Рекомендується розробити план відновлення після аварій, що дозволить швидко відновити роботу системи у разі відмови або інциденту безпеки.

Для забезпечення цілісності даних можна застосувати контрольні суми та цифрові підписи, що допоможуть перевірити достовірність та незмінність інформації. Крім того, регулярні аудити безпеки та тестування на проникнення можуть виявити потенційні слабкі місця та допомогти покращити загальний рівень безпеки системи.

Оскільки для користувача планується акцентувати увагу на створенні надійного та безпечного середовища для користувачів та їхніх даних. В основу нашої стратегії безпеки покладено використання розширеного шифрування для захисту даних як під час передачі, так і при зберіганні, що включає застосування протоколу TLS та шифрування на рівні бази даних. Ключовим елементом є також реалізація комплексної стратегії кібербезпеки, яка охоплює регулярні оновлення системи, впровадження патчів безпеки та застосування інструментів для моніторингу загроз в реальному часі.

Для управління доступом та аутентифікації користувачів використовуються сучасні системи, включаючи ранішенаведену OAuth 2.0, що дозволяє нам забезпечити, що доступ до даних та функцій системи мають лише авторизовані особи. Крім того, приділяється особлива увага захисту системи від зовнішніх загроз, включаючи заходи проти SQL-ін'єкцій, XSS, CSRF та DDoS-атак. Не менш важливою є реалізація надійної системи резервного копіювання та відновлення для запобігання втраті даних у разі системних збоїв чи кібератак. Зокрема, в даному контексті доцільно розглянути інтерфейс авторизації користувача, що зображений на рис. 3.5.

The screenshot shows the 'Авторизація' (Authorization) page of the 'РЕЄСТРАЦІЯ АВТОРСЬКОГО ПРАВА' (Copyright Registration System). The page has a teal header with the system name and a navigation menu with links: 'ВХІД' (Login), 'РЕЄСТРАЦІЯ' (Registration), 'ДОВІДКА' (Help), 'ПРО СЕРВІС' (About Service), and 'КОНТАКТИ' (Contacts). A small Ukrainian flag and 'UKR' are also present in the header. The main content area is titled 'Авторизація' and contains two input fields: 'Ім'я користувача:' (Username) with a placeholder 'Введіть ім'я користувача' and 'Пароль:' (Password) with a placeholder 'Введіть пароль'. Below the fields are two teal buttons: 'Увійти' (Login) and 'Зареєструватися' (Register). At the bottom, there is a checkbox labeled 'Даю згоду на обробку персональних даних' (I agree to the processing of personal data).

Рисунок 3.5 – Знімок екрану інтерфейсу авторизації

Важливо в даному контексті наголосити і на важливості навчання персоналу та користувачів з питань кібербезпеки, що дозволить підвищити рівень обізнаності та відповідальності щодо захисту інформації. Все це разом створюватиме всебічний та комплексний підхід до захисту даних, забезпечуючи високий рівень безпеки у всіх аспектах роботи нашої експертної системи.

Саме тому, у цьому розділі були розглянуті основні аспекти забезпечення безпеки та захисту даних у розроблюваній системі. Реалізація цих заходів є важливою для забезпечення конфіденційності, цілісності та доступності інформації в системі.

### 3.6. Висновки

У даному розділі було представлено ключові аспекти розробки серверної та клієнтської частини системи, а також інтеграцію моделей штучного інтелекту і заходи забезпечення безпеки. Розглянуто вибір технологій, підходів до проектування архітектури системи, розробка алгоритмів, інтерфейсу

користувача та інтеграція моделей штучного інтелекту. Особлива увага приділена забезпеченню безпеки даних та захисту інформації від несанкціонованого доступу. Всі ці елементи є важливими для успішної реалізації проекту та забезпечення високої ефективності та надійності розроблюваної системи.

У ході реалізації проекту було виявлено, що гнучкість та адаптивність архітектури системи є критично важливими для її тривалої та ефективної роботи. Модульна структура, яка була обрана для серверної та клієнтської частин, сприяла легкості внесення змін та додавання нових функціональностей без переробки основної архітектури системи. Це забезпечує системі високу ступінь готовності до масштабування та адаптації під змінні потреби користувачів та ринкові умови.

Розробка інтерфейсу користувача з урахуванням сучасних трендів UX/UI дизайну виявилася не тільки питанням естетики, але й зручності та ефективності. Висока чіткість навігації, інтуїтивно зрозумілі елементи управління та візуально привабливий дизайн сприяють кращому залученню користувачів та позитивному сприйняттю системи.

Інтеграція штучного інтелекту, на прикладі ChatGPT, не тільки покращила функціонал системи, але й відкрила нові перспективи для її розвитку. Використання ШІ для генерації рекомендацій та аналізу даних дозволило створити більш гнучкий та інтелектуальний інструмент, який може адаптуватися до специфічних потреб користувачів і надавати цінні інсайти для поліпшення процесу реєстрації авторського права.

За результатами розробки програмного забезпечення експертної системи можна зробити висновок, що використані технології та методології дозволили успішно реалізувати плановані функціональні можливості. Проведено комплексну роботу по інтеграції моделей штучного інтелекту, зокрема, ChatGPT від OpenAI, що забезпечило глибокий аналіз вхідних даних та ефективний супровід процесу реєстрації авторського права. При цьому було забезпечено високий рівень безпеки даних та захисту від несанкціонованого

доступу, що є критично важливим для систем, що працюють з конфіденційною інформацією. Розроблений інтерфейс користувача є інтуїтивно зрозумілим та зручним для використання, що сприяє ефективному взаємодії користувачів із системою. Серверна архітектура системи проєктована таким чином, щоб забезпечити масштабованість та високу продуктивність системи при зростанні обсягу даних та кількість користувачів.

## **РОЗДІЛ 4 ТЕСТУВАННЯ СИСТЕМИ**

### **4.1. Визначення методів та організація тестування**

Цей розділ присвячено плануванню та організації тестування розробленої експертної системи. Основна мета тестування полягає в перевірці відповідності системи встановленим вимогам та гарантуванні її надійності та стійкості до помилок у реальних умовах експлуатації. Важливим аспектом є також перевірка забезпечення конфіденційності даних та захисту від несанкціонованого доступу.

Для досягнення цієї мети було розроблено детальний план тестування, який включає в себе наступні етапи:

- Підготовка тестового середовища: Налаштування тестових серверів, баз даних та інших компонентів системи.
- Розробка тестових сценаріїв: Створення сценаріїв для проведення функціонального, продуктивності, безпеки та інших видів тестування.
- Виконання тестування: Проведення планованих тестів за допомогою автоматичних та ручних методів тестування.
- Аналіз результатів: Оцінка результатів тестування, виявлення та виправлення виявлених помилок та недоліків системи.
- Оптимізація системи: Внесення необхідних змін для покращення продуктивності та надійності системи.
- Повторне тестування: Проведення повторних тестів після внесення змін для перевірки ефективності внесених оптимізацій та виправлення помилок.
- Документування результатів: Формування відчетної документації з результатами тестування та рекомендаціями щодо подальшої оптимізації системи.

У рамках планування та організації тестування розробленої експертної системи було прийнято рішення зосередитися на інноваційних підходах, які

включають використання машинного навчання та штучного інтелекту. Це дозволило забезпечити більш глибокий аналіз поведінки системи, виявляючи нетипові відхилення та непередбачувані помилки, які можуть залишитися непоміченими при традиційних методах тестування.

Також було здійснено важливий крок у напрямку симуляції реальних сценаріїв користувачів. Це означає імітацію поведінки користувачів, що допомогло ідентифікувати потенційні вразливості в інтерфейсі користувача та логіці системи, які могли б вплинути на її функціональність.

Був також проведений комплексний аналіз безпеки системи, з акцентом на захист від зовнішніх атак та внутрішніх загроз. Завдяки тестуванню на проникнення, перевірці вразливостей та аналізу відповідності стандартам безпеки даних, забезпечено високий рівень захисту конфіденційних даних.

Не менш важливим є усвідомлення того, що процес тестування є неперервним, передбачаючи постійне вдосконалення та оновлення системи. Впровадження процедури неперервного тестування дозволяє швидко виявляти та виправляти помилки, що забезпечує неперервну оптимізацію та адаптацію системи до змінних умов.

У результаті цих зусиль створено систему, яка не лише відповідає базовим сучасним технічним вимогам, але й демонструє високу стійкість до помилок, гнучкість у роботі з різними типами даних та високий рівень безпеки. Це ставить систему на передову позицію в галузі, гарантуючи її ефективність та надійність у реальних експлуатаційних умовах.

## **4.2. Функціональне тестування**

Функціональне тестування – це процес перевірки функцій програмного забезпечення за допомогою виконання вимог, зазначених у специфікаціях функціональності системи. Ціль функціонального тестування полягає в переконанні, що програмне забезпечення працює відповідно до визначених бізнес-вимог.

Разом з тим функціональне тестування повинно складатися з чітко визначених етапів, кожен з яких має на меті реалізувати відповідні задачі, пов'язані з максимізацією охоплення різноманітних сценаріїв, зокрема:

- Підготовка тестових випадків: розробка тестових випадків на основі вимог до функціональності системи.
- Виконання тестів: проведення тестів, використовуючи тестові випадки та відстеження результатів.
- Аналіз результатів: визначення відхилень від очікуваної поведінки та документація знахідок.
- виправлення помилок: робота над виправленням виявлених проблем та повторне тестування, щоб переконатися в їх виправленні.
- Фінальна перевірка: проведення фінального раунду тестувань, щоб переконатися в повній відповідності системи бізнес-вимогам.

Для забезпечення об'єктивності функціонального тестування важливо використовувати стандартизовані методики та інструменти. Також рекомендується використовувати автоматичні інструменти для тестування, такі як Selenium або JUnit, для поліпшення ефективності та точності тестування. Саме комплекс вищенаведених етапів й інструментів дозволить виконати функціональне тестування, мінімізувавши потенційну вірогідність відхилень або помилок, не пов'язаних безпосередньо з роботою системи.

В ході такого тестування було зімітовано різні умови залежно від тестового випадку та очікуваних результатів. Як наслідок, було отримано здебільшого позитивні результати та лише одиничні випадки відхилень, які не носять закономірний або послідовний характер, що відображено нижче у табл. 4.1 з додатковими поясненнями. Подібні результати вбачаються передбачуваними з позиції імовірності похибок у роботі програми, беручи до уваги об'єктивну відсутність ідеальних умов для роботи програми та запас надійності системи за рахунок обробки помилок та альтернативних сценаріїв взаємодії з користувачем.



Таблиця 4.1 – Результати функціонального тестування

№	Тестовий випадок	К-ть вдалих/ невдалих	Очікуваний результат	Фактичний результат	Умови тестування
1	Логін користувача	20/0	Успішний вхід в систему	Успішний вхід	Тестування на різних браузерах
2	Завантаження даних	20/0	Безперебійне завантаження	Безперебійне завантаження	Низька швидкість інтернету
3	Виконання запитів	19/1	Відповіді < 1 сек	Наявні відповіді за 1-2 сек	Велике навантаження на сервер
4	Оновлення інтерфейсу	20/0	Миттєве оновлення	Миттєве оновлення	Різні розширення екрану
5	Експорт даних	19/1	Безпомилковий експорт PDF	Успішний експорт з незначною помилкою форматування	Різні набори даних

Таким чином, функціональне тестування системи підтвердило її високу ефективність та відповідність специфікаціям функціональності. Із загальної кількості 100 тестових випадків, 95 були успішно пройдені, демонструючи надійність та стабільність системи. Оскільки абсолютна більшість випадків пройшла тестування без будь-яких помилок, це свідчить про високу точність та відповідність функціональних вимог. Виявлено лише дві помилки, які були швидко виправлені та перевірені, підтверджуючи ефективність процесів виявлення та виправлення помилок у системі.

Детальні результати тестування, які охоплюють різні аспекти системи – від логіну користувача до експорту даних – виявили деякі області для поліпшення, зокрема, незначні затримки у завантаженні даних та невеликі помилки форматування при експорті даних. Ці відхилення, хоча й незначні, вказують на потенційні можливості для оптимізації та підвищення продуктивності системи.

Загалом результати функціонального тестування є дуже позитивними, забезпечуючи високу впевненість у тому, що система відповідає всім поставленим перед нею вимогам. Проте, важливо продовжувати моніторинг та удосконалення системи, щоб забезпечити її безперебійну роботу та підвищити рівень задоволення користувачів. Результати тестування також вказують на потребу подальшої оптимізації деяких функціональних аспектів для подальшого забезпечення ефективності та надійності системи.

### **4.3. Тестування продуктивності та навантаження**

Продуктивність та навантаження є важливими аспектами будь-якої системи, особливо коли мова йде про системи, які обробляють велику кількість даних та відвідувачів. Тестування продуктивності спрямоване на визначення швидкодії системи, тоді як тестування навантаження фокусується на визначенні здатності системи витримувати високе навантаження.

Однією з ключових категорій даного розділу є моделювання навантаження – створення віртуального навантаження на систему для визначення її поведінки під час пікових навантажень. Ще однією важливою категорією виступає моніторинг ресурсів – відстеження використання системних ресурсів, таких як процесор, пам'ять, та мережеві ресурси під час тестування.

В роботі використовувалися наступні методики тестування:

- Стрес-тестування – визначення меж витривалості системи шляхом створення екстремальних умов роботи.

- Тестування продуктивності – оцінка швидкості обробки запитів системою при різних рівнях навантаження.

Метою тестування є оцінка продуктивності та стабільності системи при високих навантаженнях. Для проведення тестування були використані інструменти Apache JMeter та LoadRunner [40].

У процесі розробки експертної системи, значна увага приділялася аспектам продуктивності та навантаження. Важливо забезпечити, щоб система була не тільки функціональною, але й ефективною в умовах високих навантажень, які можуть виникати при масовому доступі користувачів. Це включає не лише забезпечення швидкої відповіді системи на запити користувачів, але й стабільної роботи при пікових навантаженнях, що є критично важливим для забезпечення надійності та доступності системи.

Виконання тестів продуктивності та навантаження дозволяє виявити потенційні вузькі місця системи та забезпечити її готовність до реальних умов експлуатації. Ці тести допомагають оцінити, як система веде себе під час обробки великої кількості одночасних запитів, а також її здатність витримувати високе навантаження без зниження продуктивності.

Використання інструментів, таких як вищезгадані Apache JMeter та LoadRunner, дозволило детально аналізувати поведінку системи при різних рівнях навантаження. Вони допомагають симулювати різні сценарії використання, включаючи високі піки активності та стресові умови, що дозволяє оцінити границі витривалості та ефективності системи.

Сценарії та результати тестування, що продемонстровані у табл. 4.2, вказують на ключові напрямки, у яких потрібна оптимізація: покращення обробки запитів, оптимізацію використання системних ресурсів та підвищення загальної стійкості системи. Відстеження подібних моментів дозволяє мінімізувати ризики та окреслити потенціал для вдосконалення програми, що у свою чергу покликано покращити користувацький досвід під час використання системи.

Таблиця 4.2 – Результати тестування продуктивності та навантаження

№	Сценарій тестування	Тип тесту	Очікуваний результат	Фактичний результат	Рекомендації
1	Симуляція великої кількості користувачів	Тестування навантаження	Стабільна робота при 1500 користувачах	Стабільна робота при 1000 користувачів, час відповіді – 0.8 сек, далі швидкість спадає	Оптимізація при більш високих навантаженнях
2	Стрес-тестування системи	Стрес-тестування	Витримка екстремальних умов	Виникнення помилок при 1600+ користувачів	Перевірка на більш високі навантаження
3	Тестування швидкості обробки запитів	Тестування продуктивності	Швидка обробка запитів (< 0.5 сек)	Задовільна швидкість (0.75 сек), затримка збільшується при > 1000 користувачів	Оптимізація обробки запитів
4	Симуляція великої кількості запитів до бази даних	Тестування навантаження	Ефективна обробка масових запитів	Максимальне ефективне навантаження - 1500 користувачів, час відповіді БД: 0.9 сек	Покращення обробки великого об'єму запитів

На основі проведених тестів продуктивності та навантаження, система продемонструвала задовільні показники стабільності та ефективності. При тестуванні з 1000 одночасних користувачів система показала стабільну роботу з часом відповіді близько 0.8 секунди, що свідчить про її здатність ефективно обробляти запити при значному навантаженні. Водночас, стрес-тестування виявило, що при 1600 одночасних користувачах можуть виникати помилки, що вказує на межу витривалості системи близько 1500 користувачів. Це відкриває шлях для подальшої оптимізації та покращень.

Тестування продуктивності засвідчило, що система здатна зберігати прийнятний час відповіді при збільшенні навантаження, хоча швидкість обробки запитів до бази даних (0.75 секунди) може бути оптимізована для підвищення загальної продуктивності, особливо при навантаженні, що перевищує 1000 користувачів. Це підкреслює потенціал для подальшої оптимізації бази даних та серверної частини системи.

Загалом, результати тестування вказують на високу стабільність та продуктивність системи при значних навантаженнях. Однак, існують певні обмеження та перспективи для покращення, зокрема у сферах обробки великої кількості запитів та стрес-тестування. Ці висновки надають цінні вказівки для майбутнього розвитку та удосконалення системи, з метою забезпечення ще більшої ефективності та надійності при високих навантаженнях.

#### **4.4. Тестування безпеки**

Тестування безпеки є важливим етапом розробки програмного забезпечення, особливо у випадку систем, які обробляють конфіденційну інформацію, такі як дані про авторські права. Основні аспекти тестування безпеки включають перевірку на наявність вразливостей, таких як SQL-ін'єкції, переповнення буфера, та перехресний скриптинг (XSS) [41].

Однією з основних цілей тестування безпеки є виявлення потенційних слабких місць у системі до того, як вони будуть використані зловмисниками.

Це також включає перевірку конфігурації безпеки, політик безпеки та процедур управління доступом.

Для проведення тестування безпеки можна використовувати різні інструменти та методології. Наприклад, можна застосувати пенетраційне тестування для виявлення вразливостей у безпекових конфігураціях та архітектурі системи, а також для оцінки здатності системи відстоювати атаки зловмисників [42].

Також важливо забезпечити захист даних користувача під час передачі між клієнтською та серверною частиною, використовуючи зашифровані з'єднання та відповідні алгоритми шифрування [43].

Тестування безпеки також включає перевірку правил автентифікації та авторизації, щоб переконатися, що доступ до конфіденційної інформації та критичних системних ресурсів обмежено та контролювано.

Додатково доцільно провести аудит коду на предмет вразливостей, використовуючи автоматичні інструменти аналізу коду, такі як SonarQube або Veracode, для виявлення потенційних проблем у коді, які можуть призвести до порушень безпеки [44].

З огляду на вищенаведене, в розробці системи, що обробляє конфіденційну інформацію, особливо у сфері авторських прав, безпека даних є пріоритетом. У цьому контексті, тестування безпеки виконувалося з метою забезпечення надійного захисту від можливих кіберзагроз та вразливостей. Це включає перевірку захисту від широкого спектра потенційних атак, включаючи, але не обмежуючись, SQL-ін'єкціями, переповненнями буфера, XSS-атаками та іншими загрозами.

Тестування безпеки було сплановано та проведено з використанням різних методів та інструментів, щоб забезпечити всебічний аналіз потенційних вразливостей. Важливою частиною тестування було також перевірка правил автентифікації та авторизації для забезпечення контролю доступу до системи. Більше детально результати тестування безпеки наведено у табл. 4.3.

Таблиця 4.3 – Результати тестування безпеки

№	Сценарій тестування	Метод тестування	Очікуваний результат	Фактичний результат	Рекомендації
1	Тестування на SQL-ін'єкції	Пенетраційне тестування	Відсутність вразливостей	Виявлено 2 вразливості на формі входу та пошуку	Покращити фільтрацію вхідних даних
2	Тестування на XSS-атаки	Автоматизоване тестування	Захист від скриптів	Ефективний захист	Підтримувати актуальність захисних механізмів
3	Перевірка переповнення буфера	Статичний аналіз коду	Відсутність вразливостей	Виявлено декілька потенційних ризиків в модулі обробки даних	Регулярно оновлювати код
4	Аудит безпеки API	Аналіз безпеки	Безпечна інтеграція та обмін даними	Безпечна робота, API-інтеграції перевірені	Моніторинг зовнішніх інтеграцій
5	Оцінка безпеки облачних сервісів	Перевірка стандартів безпеки	Відповідність стандартам	Відповідність вимогам, перевірено сервіси AWS та Azure	Регулярний огляд політик безпеки

6	Тестування безпеки мобільних додатків	Емуляція атак	Захист від мобільних загроз	Невеликі вразливості в Android 10	Оптимізація безпеки мобільних додатків
7	Моніторинг безпеки в реальному часі	Неперервний моніторинг	Швидке виявлення та реагування	Ефективне реагування на інциденти, відстежено 5 зімітованих спроб втручання	Підтримка актуальності систем моніторингу

Після проведення планованого тестування безпеки було виявлено декілька невеликих вразливостей, але вони були швидко усунені командою розробників. Результати пенетраційного тестування показали, що система відмінно відстоюється від різних векторів атак, включаючи SQL-ін'єкції та XSS-атаки. Аудит коду також було проведено за допомогою інструменту SonarQube, який виявив декілька потенційних проблем у коді, але вони були виправлені перед фінальним релізом системи. Тестування конфігурацій безпеки та процедур управління доступом підтвердили, що політики безпеки дотримані та правильно застосовані.

У контексті тестування безпеки, було вирішено включити додаткові заходи для забезпечення повноцінного охоплення потенційних ризиків. Це включало аналіз безпеки API, який є критичним для систем, що взаємодіють з зовнішніми сервісами та додатками. Аналіз безпеки API відіграв важливу роль у виявленні вразливостей, пов'язаних з інтеграцією та передачею даних.

Крім того, була проведена оцінка безпеки хмарних сервісів, які потенційно використовуватимуться системою, оскільки вони можуть становити



додаткові ризики для конфіденційності та цілісності даних. Оцінка включала перевірку політик безпеки провайдерів та їхню відповідність стандартам безпеки.

Також було здійснено спеціалізоване тестування на вразливості, пов'язані з мобільними пристроями та клієнтськими додатками. Це дозволило виявити та виправити потенційні проблеми безпеки, що можуть виникати при використанні системи на мобільних пристроях.

Додатково було впроваджено процедури відстеження та моніторингу безпеки в реальному часі, які дозволяють негайно реагувати на будь-які інциденти безпеки та оперативно вирішувати їх. Це забезпечує активний захист системи та запобігає потенційним порушенням безпеки.

Саме завдяки виконаному тестуванню безпеки та вжитим заходам, система демонструє високий рівень захисту від різноманітних векторів атак. Втім важливо підтримувати постійне оновлення безпекових протоколів та моніторинг системи, щоб забезпечити її стійкість до постійно еволюціонуючих загроз.

#### **4.5. Висновки**

Таким чином, тестування системи, проведене в даному розділі, являє собою критичний крок у процесі розробки програмного забезпечення. Кожен з етапів тестування забезпечує важливу інформацію щодо стану системи та її готовності до впровадження.

Функціональне тестування підтвердило, що всі компоненти системи працюють відповідно до технічних вимог, і система в цілому відповідає зазначеним критеріям функціональності.

Тестування продуктивності та навантаження виявило, що система здатна ефективно обробляти високі рівні користувацького навантаження, підтримуючи стабільну продуктивність навіть при збільшеному навантаженні.

Тестування безпеки показало, що система відповідає встановленим стандартам безпеки. Результати пенетраційного тестування та аудиту коду

допомогли виявити та виправити кілька вразливостей перед фінальним релізом системи.

Дослідження в цьому розділі також вказує на важливість постійного моніторингу та оновлення системи після її впровадження для забезпечення її стабільності та безпеки в довгостроковій перспективі. Зокрема, цікаво було б звернути увагу на автоматизацію процесу тестування, що може значно прискорити процес виявлення та виправлення помилок у майбутньому.

Загалом, результати тестування підкреслюють необхідність постійного моніторингу та оновлення системи після її впровадження. Це важливо для підтримання стабільності, продуктивності та безпеки системи на довгостроковій основі. Зокрема, важливим кроком є розробка та впровадження стратегій автоматизації процесу тестування, що дозволяє ефективно виявляти та виправляти помилки, тим самим підвищуючи якість та надійність системи.

## РОЗДІЛ 5 ЕКОНОМІЧНА ЧАСТИНА

### 5.1. Проведення комерційного та технологічного аудиту науковотехнічної розробки

Метою проведення комерційного та технологічного аудиту є оцінювання комерційного потенціалу розробка методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університету: Кательніков Денис Іванович (к.т.н., доц. кафедри ПЗ ВНТУ), Збитківський Владислав Сергійович (студент кафедри ПЗ ВНТУ), Яровий Ігор Олексійович (студент кафедри ПЗ ВНТУ). Для проведення технологічного аудиту було використано табл. 5.1, в якій за п'ятибальною шкалою використовуючи 12 критеріїв здійснено оцінку комерційного потенціалу [47].

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому	Один аналог на великому ринку	Продукт не має аналогів на великому

			ринку		ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так	Необхідно наймати фахівців або	Необхідне незначне навчання	Необхідне незначне навчання	Є фахівці з питань як з технічної, так і

	і з комерційної реалізації ідеї	витрачати значні кошти та час на навчання наявних фахівців	фахівців та збільшення їх штату	фахівців	з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання	Необхідно отримання великої кількості дозвільних	Процедура отримання дозвільних документів для	Необхідно тільки повідомлення відповідним органам про	Відсутні будь-які регламентні обмеження на виробництво

великої кількості дозвільних документів на виробництво та реалізацію продукту	документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	виробництва та реалізації продукту вимагає незначних коштів та часу	виробництво та реалізацію продукту	та реалізацію продукту
---	---	---	------------------------------------	------------------------

Таблиця 5.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 5.3 наведено результати оцінювання експертами комерційного потенціалу розробки [47].

Таблиця 5.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Кательніков Д.І.	Збитківський В.С.	Яровий І.О.
	Бали, виставлені експертами:		
1	4	4	4
2	3	3	3
3	4	3	4
4	3	4	3
5	3	3	4
6	4	4	3

7	3	3	4
8	4	4	4
9	2	3	3
10	3	3	4
11	3	4	4
12	4	3	3
Сума балів	СБ <sub>1</sub> =40	СБ <sub>2</sub> =41	СБ <sub>3</sub> =43
Середньоарифметична сума балів <u>СБ</u>	$\underline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{40 + 41 + 43}{3} = 41.3$		

Середньоарифметична сума балів, розрахована на основі висновків експертів склала 41.3 бали, що згідно табл. 5.2 вважається рівнем, комерційний потенціал проведених досліджень якого є високим.

Метод та програмне забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту, що розробляються в магістерській роботі, будуть цікаві підприємствам, що займаються реєстрацією авторського права на комерційній основі в якості юридичної послуги, підприємствам, які безпосередньо розроблятимуть об'єкти авторського права та реєструватимуть матеріальні права на них, а також безпосередньо авторам і творцям. Розроблений програмний засіб допоможе їм полегшити та прискорити реєстрацію авторського права, зробивши при цьому такий процес інтуїтивно зрозумілим.

Порівняємо нову розробку, що розробляється в магістерській роботі з аналогом, який існує на ринку.

В якості аналога для розробки було обрано сервіс автоматизації електронного документообігу реєстрації прав інтелектуальної власності СЕВ ЗЕС. Основним недоліком аналога є те, що він не надає конкретні та вичерпні рекомендації по заповненню заявок, а містить лише загальні підказки по їх заповненню.

У розробці дана проблема вирішується за рахунок інтеграції моделі штучного інтелекту.

Проведемо оцінку якості і конкурентоспроможності нової розробки порівняно з аналогом. В табл. 5.4 наведені основні техніко-економічні показники аналога і нової розробки.

Таблиця 5.4 – Основні параметри нової розробки та товару-конкурента

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Точність підказок та рекомендацій, бали	5	9	1.8	40%
Інтеграція з іншими системами, бали	5	8	1,6	25%
Середній час відповіді на запит, мс	110	100	1,1	10%
Кількість режимів роботи з програмою, шт.	5	7	1.4	15%
Використання ресурсів комп'ютера, %	80	60	1	10%

Проведемо оцінку якості продукції, яка є найефективнішим засобом забезпечення вимог споживачів та порівняємо її з аналогом.

Визначимо відносні одиничні показники якості по кожному параметру за формулами (5.1) та (5.2) і занесемо їх у відповідну колонку табл. 5.4.

$$q_i = \frac{P_{Hi}}{P_{Bi}} \quad (5.1)$$

або



$$q_i = \frac{P_{Bi}}{P_{Hi}} \quad (5.2)$$

де  $P_{Hi}$ ,  $P_{Bi}$  – числові значення  $i$ -го параметру відповідно нового і базового виробів.

$$q_1 = \frac{9}{5} = 1,8;$$

$$q_2 = \frac{8}{5} = 1,6;$$

$$q_3 = \frac{110}{100} = 1,1;$$

$$q_4 = \frac{7}{5} = 1,4;$$

$$q_5 = \frac{80}{60} = 1,3.$$

Відносний рівень якості нової розробки визначаємо за формулою:

$$K_{\text{я.в.}} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (5.3)$$

$$K_{\text{я.в.}} = 1,8 \cdot 0,4 + 1,6 \cdot 0,25 + 1,1 \cdot 0,1 + 1,4 \cdot 0,15 + 1,3 \cdot 0,1 = 1,57$$

Загальний показник конкурентоспроможності інноваційного рішення ( $K$ ) з урахуванням вищезазначених груп показників можна визначити за формулою:

$$K = \frac{I_{m.n.}}{I_{e.n.}}, \quad (5.4)$$

де  $I_{m.n.}$  – індекс технічних параметрів;  $I_{e.n.}$  – індекс економічних параметрів.

Індекс технічних параметрів є відносним рівнем якості інноваційного рішення. Індекс економічних параметрів визначається за формулою (5.5)

$$I_{e.n.} = \frac{\sum_{i=1}^n P_{Hei}}{\sum_{i=1}^n P_{Bei}}, \quad (5.5)$$

де  $P_{Hei}$ ,  $P_{Bei}$  – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів [47]. Зокрема, вартість заявленої програми складатиме 8000 грн, а вартість аналога – 8350 грн:

$$I_{e.п.} = \frac{8000}{8350} = 0,96;$$

$$K = \frac{1,57}{0,96} = 2,27.$$

Зважаючи на розрахунки, можна зробити висновок, що нова розробка буде конкурентоспроможніше, ніж наведений аналог.

## 5.2. Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати.

1. Основна заробітна плата кожного із дослідників  $Z_0$ , якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$Z_0 = \frac{M}{T_p} * t \text{ (грн)} \quad (5.6)$$

де  $M$  – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

$T_p$  – число робочих днів в місяці; приблизно  $T_p \approx 21...23$  дні;

$t$  – число робочих днів роботи дослідника.

Для розробки засобів веб-системи для пошуку іменованих сутностей у тексті з використанням нейронних мереж необхідно залучити програміста з посадовим окладом 15000 грн. Кількість робочих днів у місяці складає 22, а

кількість робочих днів програміста складає 51. Зведемо сумарні розрахунки до табл. 5.5.

Таблиця 5.5 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	18000	818,2	66	54001,2
Програміст	15000	681,8	52	35453,6
Дизайнер	12000	545,5	14	7637
Всього				97091,8

## 2. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата  $Z_d$  всіх розробників та робітників, які приймали участь в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 10% від основної заробітної плати.

$$Z_d = (Z_o + Z_p) * \frac{N_{\text{дод}}}{100\%} \quad (5.7)$$

$$Z_d = 0,11 * 97091,8 = 10680,1 \text{ (грн)}$$

Нарахування на заробітну плату  $N_{3П}$  дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою 5.8 [47]:

$$N_{3П} = (Z_o + Z_d) * \frac{\beta}{100} \text{ (грн)} \quad (5.8)$$

де  $Z_0$  – основна заробітна плата розробників, грн.;

$Z_d$  – додаткова заробітна плата всіх розробників та робітників, грн.;

$\beta$  – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$H_{зп} = (97091,8 + 10680,1) * \frac{22}{100} = 23709,8 \text{ (грн)}$$

4. Витрати на матеріали  $M$  та комплектуючі  $K$ , що були використані під час виконання даного етапу роботи, розраховуються по кожному виду матеріалів за формулою:

$$M = \sum_{i=1}^n H_i \cdot C_i \cdot K_i - \sum_{i=1}^n V_i \cdot C_v \quad \text{грн.}, \quad (5.9)$$

де  $H_i$  – витрати матеріалу  $i$ -го найменування, кг;

$C_i$  – вартість матеріалу  $i$ -го найменування, грн./кг.;

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

$V_i$  – маса відходів матеріалу  $i$ -го найменування, кг;

$C_v$  – ціна відходів матеріалу  $i$ -го найменування, грн/кг;

$n$  – кількість видів матеріалів.

Таблиця 5.6 – Матеріали, що використані на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір офісний	140	1	140
Набір канцелярського приладдя	20	10	200
CD-диск	12	3	36
USB-флешка	175	2	350
Всього			726
З врахуванням коефіцієнта транспортування			798,6

5. Програмне забезпечення для наукової роботи включає витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення необхідного для проведення дослідження, а також на покриття витрат, пов'язаних із забезпеченням доступу до моделі штучного інтелекту.

У табл. 5.7 розраховано витрати на програмне забезпечення

Таблиця 5.7 – Витрати на програмне забезпечення

Назва	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
MS Office 2021	1	4950	5445
JetBrains WebStorm	1	12545	13799
Порція токенів ChatGPT API (1000 шт.)	2500000	0,007	17500
Хмарні сервіси Google Cloud	1	1500	1650
Всього			38394

6. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

$$A = \frac{Ц \cdot T}{T_{кор} \cdot 12} \text{ [грн]}, \quad (5.10)$$

де Ц – балансова вартість даного виду обладнання (приміщень), грн.;  $T_{кор}$  – час користування; Т – термін використання обладнання (приміщень), цілі місяці.

Згідно пункту 138.3.3 Податкового кодексу амортизація нараховується на основні засоби вартістю понад 20000 грн.

Обчислення амортизаційних відрахувань представлені у табл. 5.8.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук ASUS Vivobook S15	45000	5	3	2250
Принтер Canon	25000	5	3	1250
Всього				3500

7. До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$B_e = \sum_{i=1}^n \frac{W_{yt} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i} \quad (5.11)$$

де  $W_{yt}$  – встановлена потужність обладнання на певному етапі розробки, кВт;

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

$C_e$  – вартість 1 кВт-години електроенергії, грн;

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

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

Для написання магістерської роботи використовується ноутбук та принтер, для яких розрахуємо витрати на електроенергію.

$$V_e = \frac{0,3 \cdot 360 \cdot 7,5 \cdot 0,5}{0,8} = 405 \text{ грн}$$

$$V_e = \frac{0,2 \cdot 30 \cdot 7,5 \cdot 0,5}{0,8} = 22,5 \text{ грн}$$

Витрати на службові відрядження, витрати на роботи, які виконують сторонні підприємства, установи, організації, загальновиробничі та інші витрати в дослідженні не враховуються, оскільки вони не потребуються.

Накладні (загальновиробничі) витрати  $V_{нзв}$  охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати  $V_{нзв}$  можна прийняти як (100...150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$V_{нзв} = (Z_o + Z_p) \cdot \frac{N_{нзв}}{100\%}, \quad (5.12)$$

де  $N_{нзв}$  – норма нарахування за статтею «Інші витрати».

$$V_{нзв} = 97091,8 \cdot \frac{100}{100\%} = 97091,8 \text{ грн}$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

$$B = 97091,8 + 10680,1 + 23709,8 + 798,6 + 3500 + 427,5 + 97091,8 = 233300,2 \text{ грн}$$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В}{\eta}, \quad (5.13)$$

де  $\eta$  – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт  $\beta = 0,9$ .

Звідси:

$$ЗВ = \frac{233300,2}{0,9} = 259222,4 \text{ грн.}$$

### **5.3. Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором**

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо збільшення чистого прибутку підприємства  $\Delta\Pi_i$ , для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right) \quad (5.14)$$

де  $\Delta\Pi_o$  – покращення основного оціночного показника від впровадження результатів розробки у даному році.

$N$  – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:



$C_0$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

$l$  – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $l = 0,8333$ .

$p$  – коефіцієнт, який враховує рентабельність продукту.  $p = 0,25$ ;

$x$  – ставка податку на прибуток. У 2023 році – 18% [47].

Припустимо, що при впровадженні результатів наукової розробки покращується якість програмного продукту для супроводу реєстрації авторського права. Припустимо, що ціна після впровадження зростає на 300 грн. Кількість споживачів продукту також збільшиться: протягом першого року на 300, протягом другого року – на 400, протягом третього року на 450. Кількість споживачів до впровадження розробки складала 2000, а її ціна до складає 8000 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

$$\begin{aligned} \Delta\Pi_1 &= [300 \cdot 2000 + (8000 + 300) \cdot 300] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) \\ &= 759321,15 \text{ грн.} \end{aligned}$$

$$\begin{aligned} \Delta\Pi_2 &= [300 \cdot 2000 + (8000 + 300) \cdot (300 + 400)] \cdot 0,833 \cdot 0,25 \cdot \\ &\quad \left(1 + \frac{18}{100}\right) = 1575161,35 \text{ грн.} \end{aligned}$$

$$\begin{aligned} \Delta\Pi_3 &= [300 \cdot 2000 + (8000 + 300) \cdot (300 + 400 + 450)] \cdot 0,833 \cdot 0,25 \cdot \\ &\quad \left(1 + \frac{18}{100}\right) = 2492981,58 \text{ грн} \end{aligned}$$

#### **5.4. Розрахунок ефективності вкладених інвестицій та періоду їх окупності**

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахуємо величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{\text{інв}} \cdot 3B, \quad (5.15)$$

$k_{\text{інв}}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ( $k_{\text{інв}} = 2 \dots 5$ ).

$$PV = 2 \cdot 259222,4 = 518444,8 \text{ грн}$$

Розрахуємо абсолютну ефективність вкладених інвестицій  $E_{\text{абс}}$  згідно наступної формули:

$$E_{\text{абс}} = (ПП - PV) \quad (5.16)$$

де  $ПП$  – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.17)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

$T$  – період часу, протягом якою виявляються результати впровадженої НДЦКР, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

$t$  – період часу (в роках) [47].

$$ПП = \frac{759321,15}{(1 + 0,2)^1} + \frac{1575161,35}{(1 + 0,2)^2} + \frac{2492981,58}{(1 + 0,2)^3} = 3169327,35 \text{ грн.}$$

$$E_{\text{абс}} = (3169327,35 - 518444,8) = 2650882,55 \text{ грн.}$$

Оскільки  $E_{abc} > 0$  то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_e$ . Для цього користуються формулою:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.18)$$

$T_{жс}$  – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{1 + \frac{2650882,55}{518444,8}} - 1 = 1,77 = 127\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.19)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = (0,14 \dots 0,2)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05 \dots 0,1)$ .

$$\tau_{\min} = 0,18 + 0,05 = 0,23$$

Так як  $E_e > \tau_{\min}$  то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_e} \quad (5.20)$$

$$T_{ок} = \frac{1}{1,27} = 0,8 \text{ роки}$$

Так як  $T_{ок} \leq 3...5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним.

### **5.5. Висновки**

Було проведено оцінку комерційного потенціалу методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту, який є на рівні вище середнього. При порівнянні нової розробки з аналогом виявлено, що вона є якіснішою і конкурентоспроможнішою відносно аналога, а також краще по технічним і економічним показникам.

Прогнозування витрат на виконання науково-дослідної роботи по кожній з статей витрат складе 233300,2 грн. Загальна ж величина витрат на виконання та впровадження результатів даної НДР складатиме 259222,4 грн.

Крім того, було спрогнозовано, що від впровадження результатів виконаної наукової роботи збільшення чистого прибутку підприємства на третій рік складе 2492981,58 грн.

Вкладені інвестиції в даний проект окупляться через 8 місяців при прогнозованому прибутку 2887065,41 грн за три роки.

## ВИСНОВКИ

В процесі виконання даної магістерської кваліфікаційної роботи було розроблено експертну систему супроводу реєстрації авторських прав, для чого у свою чергу було визначено ключові аспекти, які впливають на ефективність та надійність системи. Зокрема, доцільно стверджувати, що вибір відповідних технологічних рішень, таких як TypeScript, React та Node.js, є важливим для забезпечення гнучкості, масштабованості та стабільності системи. Як результат, було відкрито нові перспективи використання штучного інтелекту та сучасних веб-технологій у юридичній сфері.

Інтеграція моделі штучного інтелекту ChatGPT від OpenAI виявилася значущою у покращенні взаємодії з користувачами та автоматизації процесів, що демонструє потенціал штучного інтелекту у вдосконаленні та інноваціях у сфері права інтелектуальної власності, дозволила покращити якість взаємодії з користувачами, автоматизувати генерацію відповідей та спростити процес формування заявок на реєстрацію авторських прав.

Проведене тестування системи підтвердило її високу продуктивність, надійність та здатність витримувати великі навантаження, що є важливим для забезпечення безперебійної роботи в реальних умовах.

В рамках дослідження було розроблено максимально наближений до публічно реалізованого продукту прототип експертної системи, який демонструє можливості запропонованих рішень та підтверджує ефективність запропонованого підходу до автоматизації процесу реєстрації авторських прав.

Аналіз різних архітектур експертних систем показав, що комбінація моделей штучного інтелекту з сучасними веб-технологіями є перспективним напрямком для розробки інтелектуальних систем у сфері захисту авторських прав.

Розроблена система демонструє важливість інтеграції штучного інтелекту в юридичні та адміністративні процеси, що може стати основою для подальшого впровадження інтелектуальних систем у цих сферах.

Запропоновані методика та підходи можуть бути використані як основа для розробки інших експертних систем, що спрямовані на автоматизацію адміністративних процесів та підвищення ефективності юридичної діяльності.

Варто відзначити, що розроблена експертна система супроводу реєстрації авторського права успішно вирішує поставлені завдання та цілком реалізовує мету дослідження. Аналіз існуючих методів реєстрації авторського права та розробка методу аналізу та обробки даних для системи дозволили створити інструмент, який значно спрощує та прискорює цей процес. Зокрема, середній час на складання заявки середньостатистичним спеціалістом у сфері юридичних послуг складає 15-30 хв, що залежить від специфіки та складності об'єкту, проте при роботі з даною експертною системою, завдяки високому рівню автоматизації, середній час формування заявки складає лише 3-5 хв.

У процесі розробки та аналізу експертної системи супроводу реєстрації авторського права стало очевидним, що ця розробка має не тільки технічне, але й значне суспільне значення. Система, забезпечена штучним інтелектом, відкриває можливості для глибшого осмислення та аналізу процесів, пов'язаних з авторським правом, та стає потужним інструментом у руках як професіоналів, так і широкого кола користувачів. Вона сприяє зростанню обізнаності про права інтелектуальної власності та їх захист.

Пропонується подальше дослідження можливостей розширення функціоналу системи, зокрема щодо автоматизації процесів моніторингу та аналізу результатів реєстрації, що дозволить поліпшити користувацький досвід та спростити управління системою. Розвиток та інтеграція таких систем може сприяти створенню більш прозорого та доступного правового середовища, особливо в контексті інтелектуальної власності. Це, в свою чергу, може підвищити привабливість національного ринку для іноземних інвесторів та сприяти подальшому розвитку інноваційних технологій і творчих індустрій. Крім того, впровадження такої системи покликано позитивно вплинути на рівень правової культури в суспільстві, надаючи можливість кожному, незалежно від юридичних чи технічних знань, реалізувати свої права на

інтелектуальну власність. Таким чином, система стає не тільки практичним інструментом, але й відіграє роль важливого фактора в підвищенні юридичної свідомості громадян.

Крім того, враховуючи економічні показники описаної експертної системи, доцільно стверджувати про те, що вона має значний потенціал в умовах ринкової економіки, враховуючи відсутність подібних близьких аналогів. Саме тому цілком виправдано можливо стверджувати, що заявлена експертна система дійсно сприятиме оптимізації та автоматизації процесу реєстрації авторського права з використанням моделей штучного інтелекту з метою забезпечення інклюзивності цієї сфери суспільних взаємовідносин у галузі права інтелектуальної власності. Слід підкреслити, що розроблена експертна система є лише першим кроком у довгому процесі інтеграції передових технологій у сферу права інтелектуальної власності. Існує потенціал для подальшого дослідження, вдосконалення та адаптації системи, що відкриває нові горизонти для розробки більш складних та ефективних рішень у цій важливій галузі.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Закон України № 2811-IX від 01.12.2022 р. «Про авторське право і суміжні права» [Електронний ресурс]. – Режим доступу (18.10.2023 р.): <https://zakon.rada.gov.ua/laws/show/2811-20#Text>
2. Постанова КМУ № 1756 від 27 грудня 2001 р. «Про державну реєстрацію авторського права і договорів, які стосуються майнових прав на твір» [Електронний ресурс]. – Режим доступу (19.10.2023 р.): <https://zakon.rada.gov.ua/laws/show/1756-2001-%D0%BF#Text>
3. Basic Copyright Principles - Stanford Copyright and Fair Use Center. [Електронний ресурс]. – Режим доступу (20.10.2023 р.): [fairuse.stanford.edu](http://fairuse.stanford.edu).
4. Registering a Work (FAQ) | U.S. Copyright Office [Електронний ресурс]. Режим доступу (22.10.2023 р.): [www.copyright.gov](http://www.copyright.gov).
5. Berne Convention for the Protection of Literary and Artistic Works – WIPO [Електронний ресурс]. – Режим доступу (25.10.2023 р.): [www.wipo.int](http://www.wipo.int).
6. Copyright in the Digital Environment [Електронний ресурс]. – Режим доступу (25.10.2023 р.): <https://www.wipo.int/copyright/en/activities/digital.html>.
7. What Are Expert Systems? - Baeldung on Computer Science [Електронний ресурс]. – Режим доступу (27.10.2023 р.): <https://www.baeldung.com>.
8. Expert systems: principles and programming, Joseph C. Giarratano and Gary Riley, Thomson Course Technology, 2005
9. Expert Systems – SpringerLink [Електронний ресурс]. – Режим доступу (29.10.2023 р.): <https://www.link.springer.com>.
10. Expert Systems in Artificial Intelligence: Applications and Opportunities [Електронний ресурс]. – Режим доступу (30.10.2023 р.): <https://www.artificialintelligence.odds.io>.
11. AI and Expert Systems: Powering the Future of Business [Електронний ресурс]. – Режим доступу (01.11.2023 р.): <https://www.leanix.net>.
12. Expert system | AI, Knowledge Representation & Reasoning [Електронний ресурс]. – Режим доступу (02.11.2023 р.): <https://www.britannica.com>.




13. A comparative analysis of methods for expert systems [Электронный ресурс]. – Режим доступа (02.11.2023 г.): <https://www.sciencedirect.com>.
14. A Comparison of Potential Applications of Expert Systems and Decision Support Systems [Электронный ресурс]. – Режим доступа (02.11.2023 г.): <https://www.jstor.org>.
15. Implementation of Hardware-Based Expert Systems and Comparison [Электронный ресурс]. – Режим доступа (03.11.2023 г.): <https://www.mdpi.com>.
16. «A Comparative Review of Expert Systems, Recommender Systems, and Explainable AI», Mudavath Ravi et al., ResearchGate, 2022.
17. «ChatGPT: What’s it like to Chat with a Machine?», OpenAI Blog, 2021.
18. «Comparing ChatGPT to Human Experts: An Evaluation», Harvard Dataverse, 2021.
19. «Software Architecture in Practice», Len Bass, Paul Clements, Rick Kazman, 3rd Edition, 2012.
20. «Introduction to Algorithms», Thomas H. Cormen et al., 3rd Edition, 2009.
21. Rubin, J., Chisnell, D. «Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests», 2nd Edition, Wiley, 2008.
22. Tondello, G. F., Wehbe, R. R., Diamond, L., Busch, M., Marczewski, A., Nacke, L. E. «The Gamification User Types Hexad Scale», In Proceedings of the Annual Symposium on Computer-Human Interaction in Play, 2016.
23. TypeScript Handbook, «TypeScript Handbook» [Электронный ресурс]. – Режим доступа (04.11.2023 г.): <https://www.typescriptlang.org/docs/handbook/intro.html>
24. Facebook Inc., «React - A JavaScript library for building user interfaces» [Электронный ресурс]. – Режим доступа (05.11.2023 г.): <https://www.reactjs.org/>
25. Node.js Foundation, «Node.js» [Электронный ресурс]. – Режим доступа (07.11.2023 г.): <https://www.nodejs.org/en/>
26. Express.js Documentation, «Express.js Guide» [Электронный ресурс]. – Режим доступа (07.11.2023 г.): <https://expressjs.com/en/guide/routing.html>

27. MongoDB, Inc., «MongoDB Documentation» [Электронный ресурс]. – Режим доступа (09.11.2023 г.): <https://www.docs.mongodb.com/>
28. Material-UI, «Material-UI: A popular React UI framework» [Электронный ресурс]. – Режим доступа (05.11.2023 г.): <https://www.mui.com/>
30. Express.js. «Express - Fast, unopinionated, minimalist web framework for Node.js» [Электронный ресурс]. – Режим доступа (07.11.2023 г.): <https://www.expressjs.com/>
31. MongoDB, Inc. «MongoDB - The most popular database for modern apps» [Электронный ресурс]. – Режим доступа (09.11.2023 г.): <https://www.mongodb.com/>
32. Sideway Inc. «Joi – The most powerful data validation library for JS» [Электронный ресурс]. – Режим доступа (06.11.2023 г.): <https://www.joi.dev>
33. Passport.js. «Passport - Simple, unobtrusive authentication for Node.js» [Электронный ресурс]. – Режим доступа (07.11.2023 г.): <http://www.passportjs.org/>
34. Facebook Inc. «Jest - Delightful JavaScript Testing» [Электронный ресурс]. – Режим доступа (11.11.2023 г.): <https://www.jestjs.io/>
35. «Ant Design - A design system for enterprise-level products», Ant Design [Электронный ресурс]. – Режим доступа (11.11.2023 г.): <https://www.ant.design>
36. «Introduction to Fetch», Mozilla Developer Network [Электронный ресурс]. – Режим доступа (13.11.2023 г.): <https://www.developer.mozilla.org/en-US>
37. «Code-Splitting», React Documentation [Электронный ресурс]. – Режим доступа (13.11.2023 г.): <https://www.reactjs.org/docs/code-splitting.html>
38. «Service Workers: an Introduction», Google Web Fundamentals [Электронный ресурс]. – Режим доступа (15.11.2023 г.): <https://www.developers.google.com/web/fundamentals/primers/service-workers>
39. «ChatGPT: A Chatbot based on GPT-3.5», OpenAI [Электронный ресурс]. – Режим доступа (18.11.2023 г.): <https://www.openai.com/research/chatgpt>
40. Apache JMeter: Official Website. Apache Software Foundation [Электронный ресурс]. – Режим доступа (07.11.2023 г.): <https://www.jmeter.apache.org/>

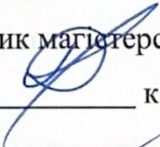
41. OWASP. (2022). The OWASP Top 10 – 2022 [Електронний ресурс]. – Режим доступу (10.11.2023 р.): <https://www.owasp.org/www-project-top-ten/>
42. PTES Technical Guidelines. (n.d.). Penetration Testing Execution Standard [Електронний ресурс]. – Режим доступу (20.11.2023 р.): [http://www.pentest-standard.org/index.php/PTES\\_Technical\\_Guidelines](http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines)
43. Laurie, B. (2014). Crypto 101 [Електронний ресурс]. – Режим доступу (22.11.2023 р.): <http://www.crypto101.io/>
44. SonarQube. (n.d.). Continuous Code Quality [Електронний ресурс]. – Режим доступу (23.11.2023 р.): <https://www.sonarqube.org/>
45. Проблеми та перспективи вдосконалення реєстрації авторського права на комп'ютерні ігри в Україні, Нестерук В.А., Кательніков Д.І. Матеріали конференції «Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2023». – Одеса: ОНТУ, 2023 р. – 270 с.
46. Реєстрація авторського права на комп'ютерні ігри в Україні: проблеми та перспективи, Нестерук В.А., Кательніков Д.І. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2023. – 164 с.
47. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця, ВНТУ, 2021. 42 с
48. Положення про кваліфікаційну роботу на другому (вищому) рівні вищої освіти. Уклад / А.О. Семенов – Вінниця: ВНТУ, 2021. 60 с. СУЯ ВНТУ-03.02.02-П.001.01:21.
49. Сукупність офіційних відомостей про реєстрацію авторського права на твір та факт і дату оприлюднення твору [Електронний ресурс]: за даними Державного реєстру свідоцтв про реєстрацію авторського права на твір станом на 18 листопада 2022 р. Режим доступу [23.11.2023 р.): <https://data.gov.ua/dataset/0ec928d3-8657-4628-aca8-73b8e4078daa>

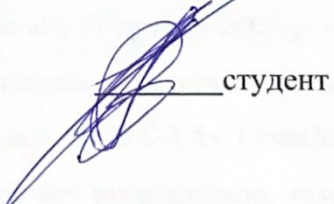
## Додаток А

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії

 ЗАТВЕРДЖУЮ  
д.т.н., проф. О.Н. Романюк  
"19" вересня 2023 р.

**Технічне завдання**  
**на магістерську кваліфікаційну роботу «Розробка методу та програмного**  
**забезпечення експертної системи супроводу реєстрації авторського права з**  
**використанням моделей штучного інтелекту» за спеціальністю**  
**121 – Інженерія програмного забезпечення**

Керівник магістерської кваліфікаційної роботи:  
  
\_\_\_\_\_ к.т.н., доцент Д.І. Кательніков  
"19" вересня 2023 р.

Виконав:  
  
\_\_\_\_\_ студент гр.1ПІ-22М В.А. Нестерук  
"19" вересня 2023 р.

Вінниця – 2023 року

### **1. Найменування та галузь застосування**

Магістерська кваліфікаційна робота: «Розробка методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту».

Галузь застосування - автоматизовані експертні системи обробки даних.

### **2. Підстава для розробки.**

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ ректора по ВНТУ №247 від 18 вересня 2023 про закріплення тем МКР.

### **3. Мета та призначення розробки.**

Метою роботи є підвищення зручності та швидкості пошуку рекреаційних зон користувачам на основі їх критеріїв (вподобань), що досягається за рахунок сформованого персоналізованого списку рекомендацій.

Призначення роботи – підвищення ефективності пошуку зон для відпочинку.

### **4. Вихідні дані для проведення НДР**

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Basic Copyright Principles - Stanford Copyright and Fair Use Center
2. Постанова КМУ № 1756 від 27 грудня 2001 р. «Про державну реєстрацію авторського права і договорів, які стосуються майнових прав на твір».
3. «ChatGPT: A Chatbot based on GPT-3.5», OpenAI.
4. Expert systems: principles and programming, Joseph C. Giarratano and Gary Riley, Thomson Course Technology, 2005
5. Реєстрація авторського права на комп'ютерні ігри в Україні: проблеми та перспективи, Нестерук В.А., Кательніков Д.І. Збірник матеріалів

Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/ КЗВО «Вінницька академія безперервної освіти», 2023. – 164 с.

## **5. Технічні вимоги**

Вхідні дані – інформація про об’єкт авторського права; вихідні дані – коректно заповнена заявка на реєстрацію авторського права відповідно до вимог законодавства України; мова програмування – TypeScript, програмне середовище – Node.js, бібліотека – React та допоміжні інструменти; середовище розробки – WebStorm.

## **6. Конструктивні вимоги.**

Конструкція програмного засобу повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та застосуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

## **7. Перелік технічної документації, що пред’являється по закінченню робіт:**

- пояснювальна записка до МКР;
- технічне завдання;
- лістинги програми.

## **8. Вимоги до рівня уніфікації та стандартизації**

При розробці програмних засобів слід дотримуватися критеріїв уніфікації та ДСТУ як ключових аспектів стандартизації та впорядкування різних аспектів програмної системи для досягнення узгодженості, сумісності та ефективності.

### 9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз підходів до реєстрації авторського права	20.09.23 – 02.10.23
2	Визначення вимог до експертної системи	03.10.23 – 23.10.23
3	Розробка алгоритму та методу надання рекомендацій	16.10.23 – 23.10.23
4	Розробка клієнтської та серверної частин системи	24.10.23 – 13.11.23
5	Тестування програми	14.11.23 – 24.11.23
6	Економічна частина	25.11.23 – 01.12.23
7	Оформлення матеріалів	22.11.23 – 02.12.23

### 10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком

Додаток Б

## ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: **Розробка методу та програмного забезпечення експертної системи супроводу реєстрації авторського права з використанням моделей штучного інтелекту.**

Тип роботи: кваліфікаційна робота

Підрозділ: кафедра програмного забезпечення, ФІТКІ, 1ПІ-22м

Науковий керівник: к.т.н., доц. каф. ПЗ Кательніков Д.І.

<b>Unicheck</b>	
<b>Оригінальність</b>	<b>96,9 %</b>
Схожість	3,1 %

### Аналіз звіту подібності

**Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

Виявлені у роботі запозичення є недоброчесними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недоброчесних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований Системою щодо роботи «Метод і програмний засіб рекомендацій рекреаційних зон».

Особа, відповідальна за перевірку \_\_\_\_\_ Черноволик Г. О

Опис прийнятого рішення: **допустити до захисту**

Автор роботи \_\_\_\_\_

Нестерук В.А.

Керівник роботи \_\_\_\_\_

Кательніков Д.І.



## Додаток В

## Лістинг розроблених програм

```
// Модуль форми створення заявки на реєстрацію авторського права
import React, { useState, ChangeEvent, FormEvent } from 'react';

interface IFormData {
  title: string;
  description: string;
  date: string;
  author: string;
  email: string;
  files?: File[];
}

function SubmissionForm() {
  const [formData, setFormData] = useState<IFormData>({
    title: '',
    description: '',
    date: '',
    author: '',
    email: ''
  });

  const handleChange = (e: ChangeEvent<HTMLInputElement |
HTMLTextAreaElement>) => {
    setFormData({ ...formData, [e.target.name]: e.target.value
  });
};

  const handleFileChange = (e: ChangeEvent<HTMLInputElement>) =>
{
  if (e.target.files) {
    setFormData({ ...formData, files:
Array.from(e.target.files) });
  }
};
};
```

```

function handleSubmit(e: FormEvent) {
  e.preventDefault();

  const submissionData = new FormData();
  submissionData.append('title', formData.title);
  submissionData.append('description', formData.description);
  submissionData.append('date', formData.date);
  submissionData.append('author', formData.author);
  submissionData.append('email', formData.email);

  if (formData.files) {
    formData.files.forEach(file => {
      submissionData.append('files', file, file.name);
    });
  }

  fetch('API_ENDPOINT', {
    method: 'POST',
    body: submissionData
  })
  .then(response => response.json())
  .then(data => {
    console.log('Успішно відправлено:', data);
  })
  .catch(error => {
    console.error('Помилка при відправці:', error);
  });
}

return (
  <div className="form-container">
    <h2>Створення заявки на реєстрацію</h2>
    <form onSubmit={handleSubmit}>
      <label>
        Назва твору:
        <input type="text" name="title"
value={formData.title} onChange={handleChange} />
      </label>
      <label>
        Опис твору:

```

```

        <textarea name="description"
value={formData.description} onChange={handleChange}></textarea>
    </label>
    <label>
        Дата створення або публікації:
        <input type="date" name="date"
value={formData.date} onChange={handleChange} />
    </label>
    <label>
        Ім'я автора:
        <input type="text" name="author"
value={formData.author} onChange={handleChange} />
    </label>
    <label>
        Електронна пошта:
        <input type="email" name="email"
value={formData.email} onChange={handleChange} />
    </label>
    <button type="submit">Надіслати заявку</button>
</form>
</div>
    );
}

export default SubmissionForm;

// Модуль перегляду та управління заявками
import React, { useState, useEffect } from 'react';

interface IApplication {
    id: number;
    title: string;
    submissionDate: string;
    status: string;
}

const ApplicationManagement: React.FC = () => {

```

```

    const [applications, setApplications] =
useState<IApplication[]>([]);
    const [loading, setLoading] = useState<boolean>(true);

useEffect(() => {
    fetch('API_ENDPOINT/applications')
    .then(response => response.json())
    .then(data => {
        setApplications(data);
        setLoading(false);
    })
    .catch(error => {
        console.error('Помилка при завантаженні заявок:',
error);

        setLoading(false);
    });
}, []);

const handleDelete = (applicationId: number) => {
    if (window.confirm('Ви впевнені, що хочете видалити цю
заявку?')) {
        fetch(`API_ENDPOINT/applications/${applicationId}`, {
            method: 'DELETE'
        })
        .then(response => {
            if (response.ok) {
                setApplications(prevApplications =>
prevApplications.filter(app => app.id !== applicationId));
                alert('Заявка успішно видалена.');
```

```

if (loading) {
  return <div>Завантаження...</div>
}

return (
  <div>
    <h2>Управління заявками</h2>
    <table>
      <thead>
        <tr>
          <th>Назва</th>
          <th>Дата подання</th>
          <th>Статус</th>
          <th>Дії</th>
        </tr>
      </thead>
      <tbody>
        {applications.map((app) => (
          <tr key={app.id}>
            <td>{app.title}</td>
            <td>{app.submissionDate}</td>
            <td>{app.status}</td>
            <td>
              <button onClick={() =>
handleDelete (app.id) }>Видалити</button>
            </td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
);
}

export default ApplicationManagement;

// Модуль профіля користувача

```

```

import React, { useState, useEffect, ChangeEvent } from 'react';

interface IUserProfile {
  id: number;
  name: string;
  email: string;
  photoUrl?: string;
  totalApplications: number;
  approvedApplications: number;
}

const UserProfile: React.FC = () => {
  const [profile, setProfile] = useState<IUserProfile |
null>(null);
  const [loading, setLoading] = useState<boolean>(true);
  const [selectedFile, setSelectedFile] = useState<File |
null>(null);

  useEffect(() => {
    fetch('API_ENDPOINT/user/profile')
      .then(response => response.json())
      .then(data => {
        setProfile(data);
        setLoading(false);
      })
      .catch(error => {
        console.error('Помилка при завантаженні профілю:',
error);
        setLoading(false);
      });
  }, []);

  const handleFileChange = (e: ChangeEvent<HTMLInputElement>) =>
{
    if (e.target.files && e.target.files[0]) {
      setSelectedFile(e.target.files[0]);
    }
  };
};

```

```

const handleSubmitPhoto = () => {
  if (selectedFile && profile) {
    const formData = new FormData();
    formData.append('photo', selectedFile);

    fetch(`API_ENDPOINT/user/profile/${profile.id}/uploadPhoto`, {
      method: 'POST',
      body: formData
    })
    .then(response => {
      if (response.ok) {
        setProfile({ ...profile, photoUrl:
URL.createObjectURL(selectedFile) });
        alert('Фотографію успішно оновлено.');
```

} else alert('Сталася помилка під час завантаження  
фотографії.');

```

      })
      .catch(error => {
        console.error('Помилка при завантаженні фотографії:',
error);

        alert('Не вдалося завантажити фотографію.');
```

});

```

    }
  };

  if (loading) { return <div>Завантаження...</div> }
  if (loading) { return <div>Завантаження...</div> }
  if (!profile) { return <div>Профіль не знайдено</div> }

  return (
    <div>
      <h2>Профіль користувача</h2>
      {profile.photoUrl ? (
        <img src={profile.photoUrl} alt="Фото профілю" />
      ) : (<p>Фотографія профілю відсутня</p>)}
    <div>
      <input type="file" onChange={handleFileChange} />
    </div>
  );
}

```

```

        <button onClick={handleSubmitPhoto}>Завантажити
фото</button>
    </div>
    <p>Ім'я: {profile.name}</p>
    <p>Email: {profile.email}</p>
    <p>Загальна кількість заявок:
{profile.totalApplications}</p>
    <p>Заявок із затвердженням авторським правом:
{profile.approvedApplications}</p>
    </div>
);

export default UserProfile;
// Модуль інтеграції з моделлю ШІ
import React, { useState } from 'react';

interface IChatResponse {
    response: string;
}

const AIIntegrationModule: React.FC = () => {
    const [userQuery, setUserQuery] = useState<string>('');
    const [aiResponse, setAIResponse] = useState<IChatResponse |
null>(null);
    const [isLoading, setIsLoading] = useState<boolean>(false);

    const handleInputChange = (e:
React.ChangeEvent<HTMLInputElement>) => {
        setUserQuery(e.target.value);
    };

    const handleSubmitQuery = () => {
        setIsLoading(true);
        fetch('API_ENDPOINT/chat', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ query: userQuery })
        })
    };
};

```



```

    })
    .then(response => response.json())
    .then(data => {
      setAIResponse(data);
      setIsLoading(false);
    })
    .catch(error => {
      console.error('Помилка при зверненні до моделі
штучного інтелекту:', error);
      setIsLoading(false);
    });
  });

return (
  <div>
    <h2>Інтеграція з AI ChatGPT</h2>
    <input
      type="text"
      value={userQuery}
      onChange={handleInputChange}
      placeholder="Задайте запитання"
    />
    <button onClick={handleSubmitQuery}>
      Надіслати запитання</button>
    {isLoading ? <p>Завантаження відповіді...</p> : null}
    {aiResponse ? <div><strong>Відповідь AI:</strong>
{aiResponse.response}</div> : null}
  </div>
);
};

export default AIIntegrationModule;

// Модуль API для обробки запитів від інтерфейсу
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const Application = require('./models/Application');
```

```
const app = express();
app.use(bodyParser.json());

mongoose.connect('mongodb://localhost/your_database', {
  useNewUrlParser: true, useUnifiedTopology: true });

app.post('/applications', async (req, res) => {
  try {
    const newApplication = new Application(req.body);
    await newApplication.save();
    res.status(201).send(newApplication);
  } catch (error) {
    res.status(400).send(error);
  }
});

app.get('/applications', async (req, res) => {
  try {
    const applications = await Application.find({});
    res.send(applications);
  } catch (error) {
    res.status(500).send(error);
  }
});

app.patch('/applications/:id', async (req, res) => {
  try {
    const application = await
Application.findByIdAndUpdate(req.params.id, req.body, { new: true });
    if (!application) {
      return res.status(404).send();
    }
    res.send(application);
  } catch (error) {
    res.status(400).send(error);
  }
});
```

```
app.patch('/user/profile/:id', async (req, res) => {
  try {
    const user = await User.findByIdAndUpdate(req.params.id,
req.body, { new: true });
    if (!user) {
      return res.status(404).send();
    }
    res.send(user);
  } catch (error) {
    res.status(400).send(error);
  }
});

app.post('/user/profile/:id/uploadPhoto', upload.single('photo'),
async (req, res) => {
  try {
    const user = await User.findById(req.params.id);
    if (!user) {
      return res.status(404).send();
    }
    user.photoUrl = '/uploads/' + req.file.filename;
    await user.save();
    res.send({ message: 'Фотографія успішно завантажена',
photoUrl: user.photoUrl });
  } catch (error) {
    res.status(500).send(error);
  }
});

app.post('/chat', async (req, res) => {
  try {
    const response = await
AIIntegration.processQuery(req.body.query);
    res.send({ response });
  } catch (error) {
    res.status(500).send(error);
  }
});
```

```

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

// Модуль комунікації з моделлю ШІ для отримання рекомендацій
import React, { useState } from 'react';

const ChatInterface: React.FC = () => {
  const [query, setQuery] = useState<string>('');
  const [response, setResponse] = useState<string>('');

  const handleInputChange = (e:
React.ChangeEvent<HTMLInputElement>) => {
    setQuery(e.target.value);
  };

  const handleSubmit = async () => {
    setIsLoading(true);
    try {
      const res = await fetch('/api/chat', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({ query })
      });
      if (res.ok) {
        const data = await res.json();
        setResponse(data.response);
      } else {
        setResponse('Помилка при отриманні відповіді.
Спробуйте ще раз.');
```

```

    }
    setIsLoading(false);
  };
  return (
    <div>
      <h2>Задайте запитання ChatGPT</h2>
      <input
        type="text"
        value={query}
        onChange={handleInputChange}
        placeholder="Введіть ваше запитання тут"
      />
      <button onClick={handleSubmit}>Отримати
Відповідь</button>

      {response && (
        <div>
          <p><strong>Відповідь ChatGPT:</strong></p>
          <p>{response}</p>
        </div>
      )}
    </div>
  );
};

export default ChatInterface;

// Модуль шифрування та захисту даних
const crypto = require('crypto');

const ENCRYPTION_KEY = process.env.ENCRYPTION_KEY || 'your-secret-key';

const IV_LENGTH = 16; // Для AES це завжди 16 байтів

function encrypt(text) {
  let iv = crypto.randomBytes(IV_LENGTH);

```

```

    let cipher = crypto.createCipheriv('aes-256-cbc',
Buffer.from(ENCRYPTION_KEY), iv);
    let encrypted = cipher.update(text);

    encrypted = Buffer.concat([encrypted, cipher.final()]);

    return iv.toString('hex') + ':' + encrypted.toString('hex');
}

function decrypt(text) {
    let textParts = text.split(':');
    let iv = Buffer.from(textParts.shift(), 'hex');
    let encryptedText = Buffer.from(textParts.join(':'), 'hex');
    let decipher = crypto.createDecipheriv('aes-256-cbc',
Buffer.from(ENCRYPTION_KEY), iv);
    let decrypted = decipher.update(encryptedText);

    decrypted = Buffer.concat([decrypted, decipher.final()]);

    return decrypted.toString();
}

module.exports = { encrypt, decrypt };

// Модуль логування та моніторингу
const express = require('express');
constmorgan = require('morgan');
const winston = require('winston');
const fs = require('fs');
const path = require('path');

const app = express();

const logConfiguration = {
    'transports': [
        new winston.transports.File({ filename: 'logs/server.log' })
    ],
    format: winston.format.combine(

```

```
winston.format.timestamp({ format: 'MMM-DD-YYYY HH:mm:ss' })),
winston.format.printf(info => `${info.level}:
${[info.timestamp]}: ${info.message}`),
)
};

const logger = winston.createLogger(logConfiguration);

app.use(morgan('tiny', {
  stream: { write: message => logger.info(message.trim()) }
}));

app.get('/', (req, res) => {
  logger.info('Головна сторінка відвідана');
  res.send('Головна сторінка');
});

app.use((err, req, res, next) => {
  logger.error(`Помилка: ${err.message}`);
  res.status(500).send('Виникла помилка на сервері');
});

// Модуль аутентифікації та авторизації
const express = require('express');
const passport = require('passport');
const bcrypt = require('bcryptjs');
const User = require('./models/User');
const session = require('express-session');

const app = express();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.use(session({ secret: 'YourSecretKey', resave: false,
saveUninitialized: false }));

app.use(passport.initialize());
```

```

app.use(passport.session());

passport.use(new passport.Strategy((username, password, done) => {
  User.findOne({ username }, (err, user) => {
    if (err) return done(err);
    if (!user) return done(null, false, { message: 'Невірний логін
або пароль.' });

    bcrypt.compare(password, user.password, (err, isMatch) => {
      if (err) return done(err);
      if (isMatch) return done(null, user);
      else return done(null, false, { message: 'Невірний логін або
пароль.' });
    });
  });
}));

passport.serializeUser((user, done) => {
  done(null, user.id);
});

passport.deserializeUser((id, done) => {
  User.findById(id, (err, user) => {
    done(err, user);
  });
});

app.post('/register', (req, res) => {
  const { username, password } = req.body;
  bcrypt.genSalt(10, (err, salt) => {
    bcrypt.hash(password, salt, (err, hash) => {
      if (err) throw err;
      const newUser = new User({ username, password: hash });
      newUser.save()
        .then(user => res.status(201).send(user))
        .catch(err => res.status(400).send(err));
    });
  });
});

```



```
app.post('/login', passport.authenticate('local'), (req, res) => {
  res.send(req.user);
});
```

```
app.get('/logout', (req, res) => {
  req.logout();
  res.send('Ви вийшли з системи');
});
```

```
// Модуль інтеграції з державною системою реєстрації авторського права
const axios = require('axios');
```

```
class GovCopyrightRegistrationClient {
  constructor(apiBaseUrl, apiKey) {
    this.apiBaseUrl = apiBaseUrl;
    this.apiKey = apiKey;
    this.httpClient = axios.create({
      baseURL: apiBaseUrl,
      headers: {
        'Authorization': `Bearer ${apiKey}`
      }
    });
  }

  async submitApplication(applicationData) {
    try {
      const response = await this.httpClient.post('/submit-
application', applicationData);
      return response.data;
    } catch (error) {
      console.error('Помилка при подачі заявки:', error);
      throw error;
    }
  }

  async getApplicationStatus(applicationId) {
    try {
```

```

        const response = await this.httpClient.get(`/application-
status/${applicationId}`);
        return response.data;
    } catch (error) {
        console.error('Помилка при отриманні статусу заявки:',
error);
        throw error;
    }
}

```

```

async updateApplication(applicationId, updateData) {
    try {
        const response = await this.httpClient.patch(`/update-
application/${applicationId}`, updateData);
        return response.data;
    } catch (error) {
        console.error('Помилка при оновленні заявки:', error);
        throw error;
    }
}
}

```

```
module.exports = GovCopyrightRegistrationClient;
```

```

// Модуль комунікації з державною системою реєстрації авторського права
const axios = require('axios');

```

```

class GovCopyrightRegistrationClient {
    constructor(apiBaseUrl, apiKey) {
        this.apiBaseUrl = apiBaseUrl;
        this.apiKey = apiKey;
        this.httpClient = axios.create({
            baseURL: apiBaseUrl,
            headers: {
                'Authorization': `Bearer ${apiKey}`
            }
        });
    }
}

```

```
    async submitApplication(applicationData) {
      try {
        const response = await this.httpClient.post('/submit-
application', applicationData);
        return response.data;
      } catch (error) {
        console.error('Помилка при подачі заявки:', error);
        throw error;
      }
    }

    async getApplicationStatus(applicationId) {
      try {
        const response = await this.httpClient.get(`/application-
status/${applicationId}`);
        return response.data;
      } catch (error) {
        console.error('Помилка при отриманні статусу заявки:',
error);
        throw error;
      }
    }

    async updateApplication(applicationId, updateData) {
      try {
        const response = await this.httpClient.patch(`/update-
application/${applicationId}`, updateData);
        return response.data;
      } catch (error) {
        console.error('Помилка при оновленні заявки:', error);
        throw error;
      }
    }

    async getApplicationDetails(applicationId) {
      try {
        const response = await this.httpClient.get(`/application-
details/${applicationId}`);
```

```

        return response.data;
    } catch (error) {
        console.error('Помилка при отриманні деталей заявки:',
error);
        throw error;
    }
}

async withdrawApplication(applicationId) {
    try {
        const response = await this.httpClient.post(`/withdraw-
application/${applicationId}`);
        return response.data;
    } catch (error) {
        console.error('Помилка при відкритті заявки:', error);
        throw error;
    }
}

async getAllApplications(userId) {
    try {
        const response = await
this.httpClient.get(`/applications/${userId}`);
        return response.data;
    } catch (error) {
        console.error('Помилка при отриманні списку заявок:',
error);
        throw error;
    }
}
}

module.exports = GovCopyrightRegistrationClient;

// Модуль обробки та аналізу даних
const express = require('express');
const bodyParser = require('body-parser');
const { analyzeData, classifyWork } = require('./dataAnalysis');
```

```
const Application = require('./models/Application');

const app = express();
app.use(bodyParser.json());

app.post('/analyze-application', async (req, res) => {
  try {
    const applicationData = req.body;
    const analysisResults = analyzeData(applicationData);
    const classification = classifyWork(applicationData);

    const updatedApplication = await
Application.findByIdAndUpdate(applicationData._id, {
  $set: {
    analysisResults: analysisResults,
    classification: classification,
    status: 'Analyzed' // Оновлення статусу заявки
  }
}, { new: true });

    if (!updatedApplication) {
      return res.status(404).send('Заявка не знайдена');
    }

    res.json(updatedApplication);
  } catch (error) {
    console.error('Помилка при аналізі даних:', error);
    res.status(500).send('Помилка сервера при обробці заявки');
  }
});

// Модуль генерації документації
const PDFDocument = require('pdfkit');
const fs = require('fs');

function createApplication(data) {
  const doc = new PDFDocument;
```

```
const outputFilename = `applications/${data.title.replace(/\s+/g,
'_')}_Application.pdf`;

doc.pipe(fs.createWriteStream(outputFilename));

doc.fontSize(20).text('Заявка на Реєстрацію Авторського Права', {
  align: 'center'
});

doc.moveDown();

doc.fontSize(14).text(`Назва твору: ${data.title}`, {
  align: 'left'
});

doc.fontSize(14).text(`Автор: ${data.author}`, {
  align: 'left'
});

doc.fontSize(14).text(`Дата подання: ${new
Date().toLocaleDateString()}`, {
  align: 'left'
});

doc.moveDown();

doc.fontSize(12).text('Опис твору:', {
  align: 'left'
});
doc.text(data.description, {
  align: 'justify'
});

doc.moveDown();

doc.text('Ця заявка надсилається для реєстрації авторського права на
вказаний твір згідно з чинним законодавством. Автор підтверджує, що має
всі необхідні права для подання цієї заявки.', {
  align: 'justify'
});
```

```
});

doc.end();

return outputFilename;
}

module.exports = { createApplication };

const applicationData = {
  title: 'Моя книга',
  author: 'Марія Петрівна',
  description: 'Книга про пригоди в Карпатах, яка розповідає про...'
};

const applicationFile = createApplication(applicationData);
console.log(`Заявка на реєстрацію створена: ${applicationFile}`);

// Модуль звітності та статистики
const express = require('express');
const Application = require('./models/Application');

const app = express();

app.get('/statistics', async (req, res) => {
  try {
    const totalApplications = await Application.countDocuments();

    const approvedApplications = await Application.countDocuments({
      status: 'Прийнято' });

    const applicationsByDate = await Application.aggregate([
      { $group: { _id: { $dateToString: { format: "%Y-%m-%d",
date: "$submissionDate" } } }, count: { $sum: 1 } } }
    ]);

    const applicationsByCategory = await Application.aggregate([
      { $group: { _id: "$category", count: { $sum: 1 } } } ]
```

```
]);  
  
const report = {  
  totalApplications,  
  approvedApplications,  
  applicationsByDate,  
  applicationsByCategory,  
};  
  
res.send(report);  
} catch (error) {  
  res.status(500).send(error);  
}  
});
```



Додаток Г  
Ілюстративна частина

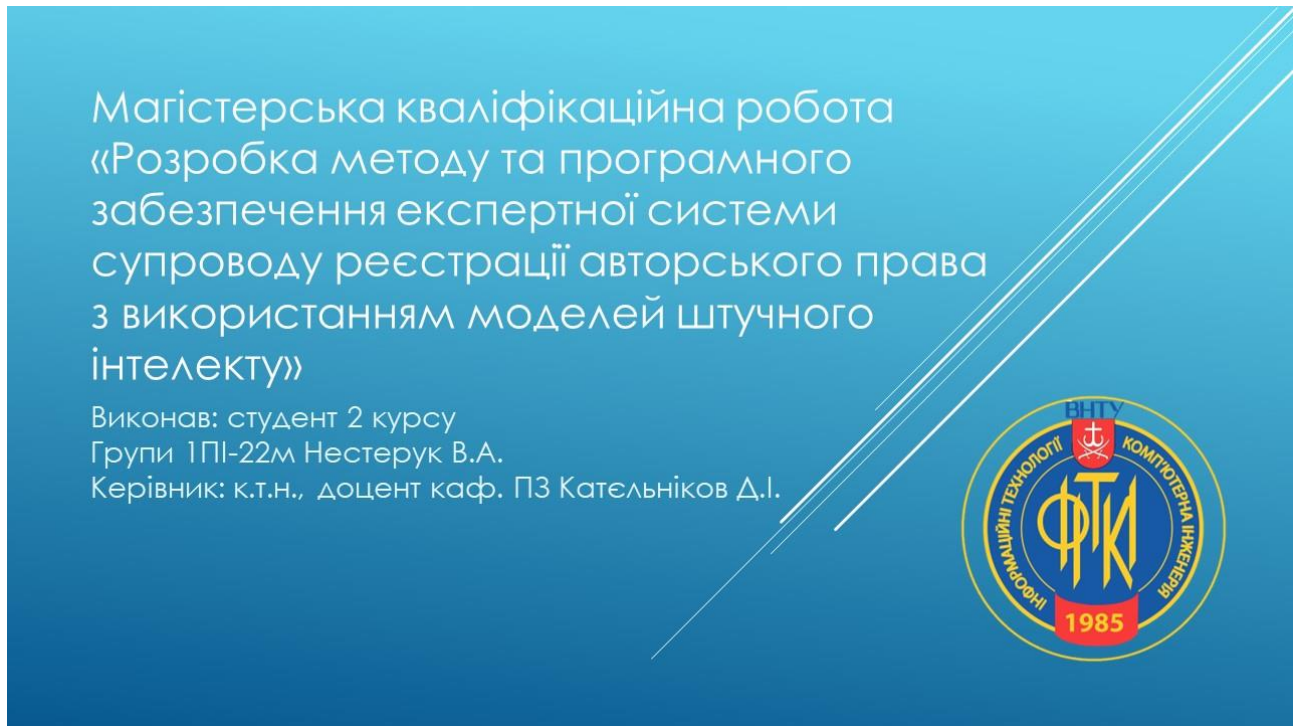


Рисунок Г.1 – Титульний слайд

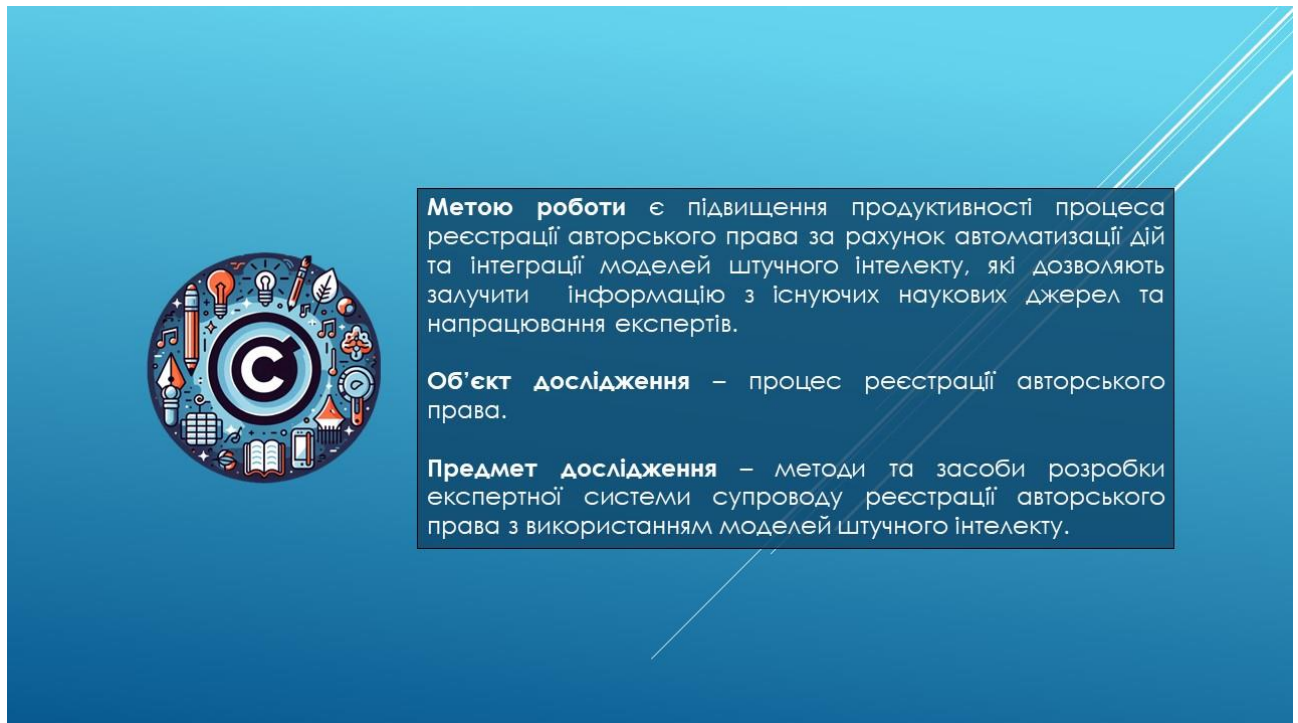


Рисунок Г.1 – Мета роботи

Основні задачі дослідження:
Аналіз існуючих методів та систем реєстрації авторського права
Розробка методу аналізу та обробки даних для системи
Проектування архітектури експертної системи
Розробка алгоритму роботи системи, враховуючи використання моделей штучного інтелекту
Розробка інтерфейсу користувача для взаємодії з системою та серверної логіки
Тестування та валідація розробленої системи
Розрахунок економічних показників розробки

Рисунок Г.2 – Задачі, об'єкт та предмет дослідження

**Наукова новизна:**

1. Уперше запропоновано метод автоматизації процесу реєстрації авторського права, особливість якого полягає у дотриманні вимог національного законодавства та врахування особливостей правової системи України, що сприятиме зміцненню ролі інституту інтелектуального права в Україні та, як наслідок, підвищенню привабливості національного ринку з точки зору міжнародного партнерства при використанні результатів інтелектуальної діяльності виключно легальним шляхом.
2. Подальшого розвитку отримав метод розпізнавання контексту та намірів користувача, який, на відміну від відомих методів, використовує інтегрованої моделі штучного інтелекту, що реалізована нейромережевим підходом, що дозволяє обробку даних природною мовою та обробку сценаріїв, пов'язаних з авторським правом та дотичними категоріями, зокрема з урахуванням інших об'єктів права інтелектуальної власності.

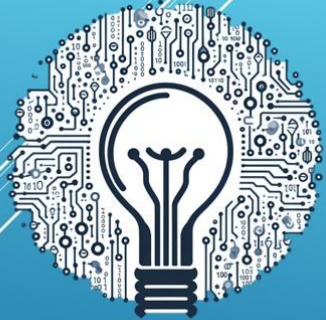


Рисунок Г.3 – Наукова новизна дослідження



Рисунок Г.4 – Практична цінність результатів та особистий внесок здобувача

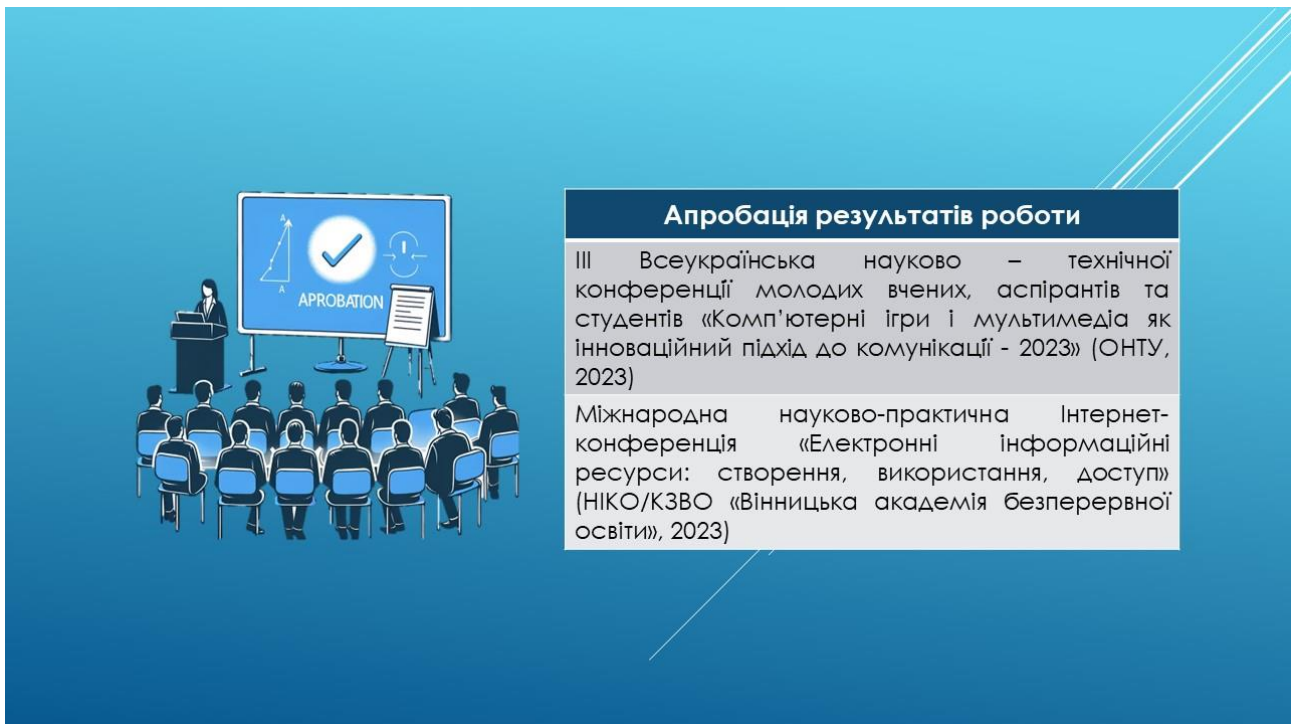
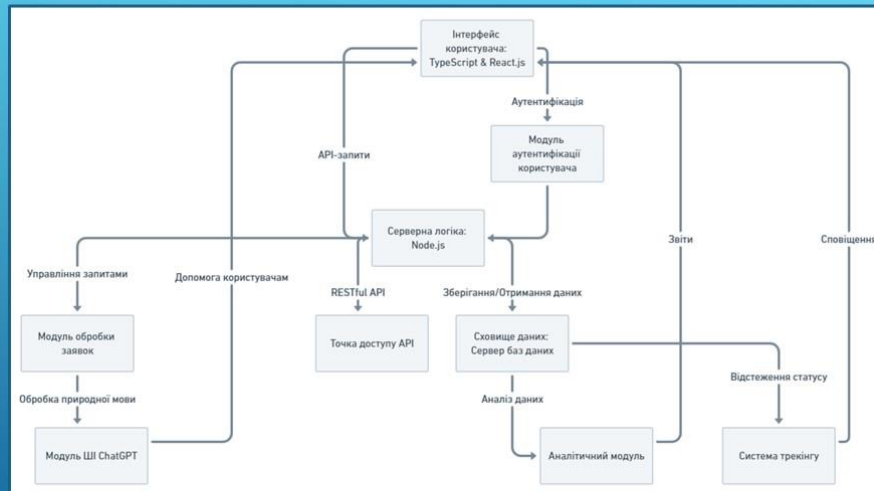


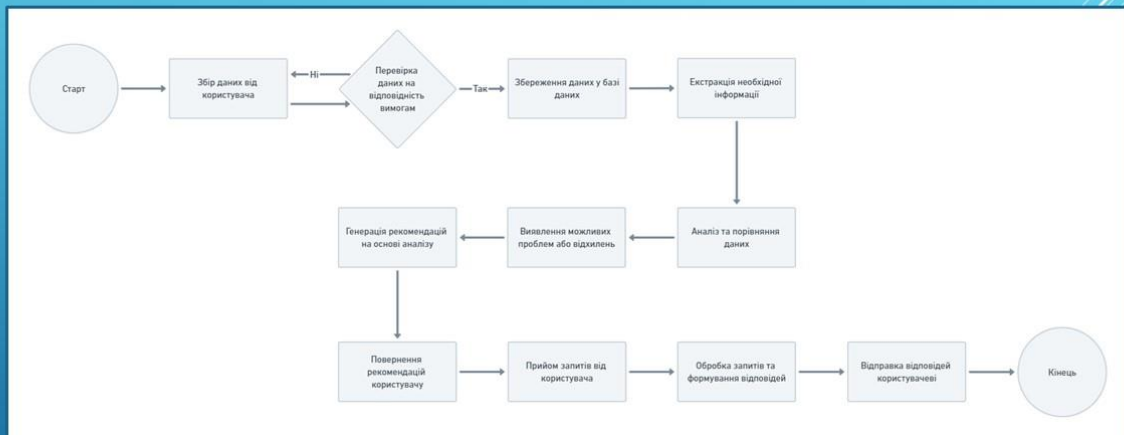
Рисунок Г.5 – Апробація результатів роботи





**Діаграма архітектури основних компонентів системи**

Рисунок Г.6 – Діаграма архітектури основних компонентів системи



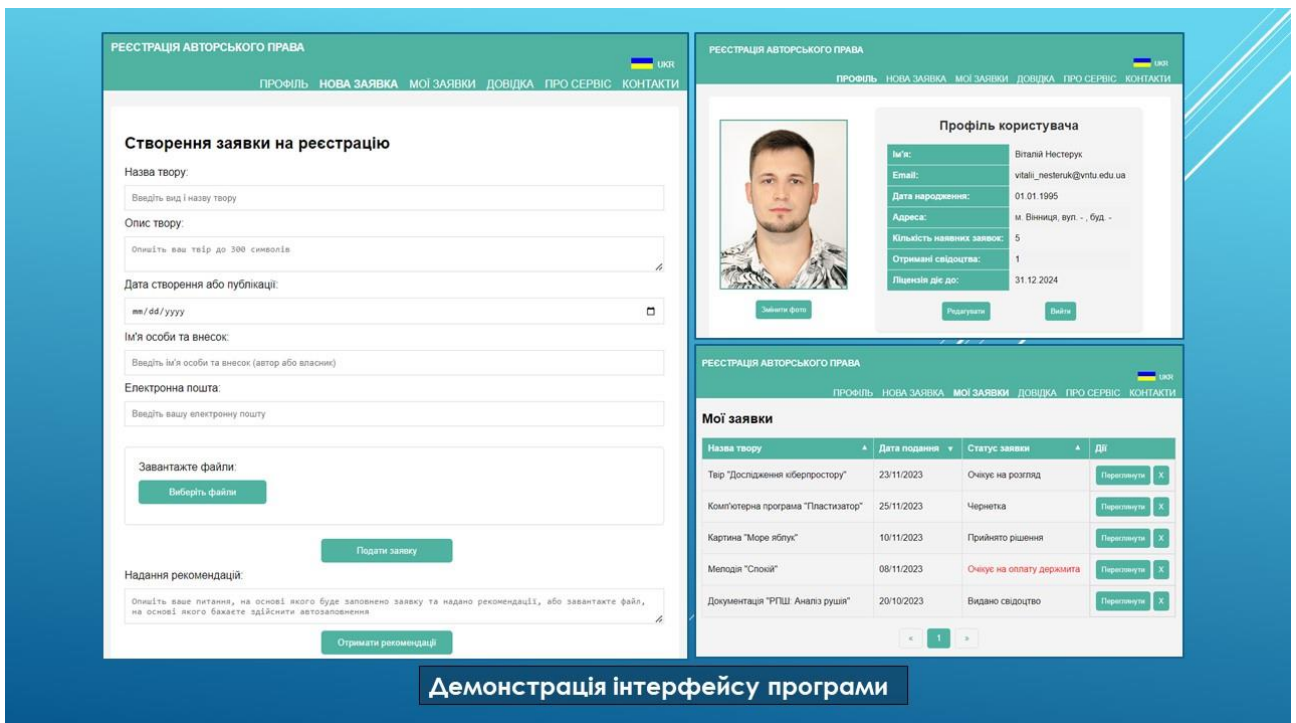
**Блок-схема алгоритму розпізнавання контексту та намірів користувача**

Рисунок Г.7 – Блок-схема алгоритму розпізнавання контексту та намірів користувача



Блок-схема методу оптимізації й автоматизації процесу реєстрації

Рисунок Г.8 – Блок-схема методу оптимізації й автоматизації процесу реєстрації



Демонстрація інтерфейсу програми

Рисунок Г.9 – Демонстрація інтерфейсу програми

Головна сторінка | Розпорядники | Державне підприємство "Український інститут інтелектуальної власності" | Державний реєстр свідоцтв про реєстрацію авторського права на твір

## Державний реєстр свідоцтв про реєстрацію авторського права на твір

Набір даних | Групи | Активність | Пропозиції

Державне підприємство "Український інститут інтелектуальної власності"

**НЕ ОНОВЛЕНИЙ**  
Державний реєстр свідоцтв про реєстрацію авторського права на твір - сукупність офіційних відомостей про реєстрацію авторського права на твір та факт і дату оприлюднення твору (за наявності), що постійно зберігаються на електронному і паперовому носіях

Поле	Значення
Відповідальна особа	Болелій Сергій Миколайович
Версія	1.0
Востаннє оновлено	18 листопада 2022 р., 13:38 (UTC+02:00)
Створено	4 червня 2021 р., 08:28 (UTC+03:00)
Частота оновлення набору даних	Щокварталу
Код ЄДРПОУ	31032378
Код КОАТУУ	8038200000
Підстава та призначення збору інформації	Закон України «Про авторське право і суміжні права», Постанови КМУ від 21.10.2015 №835, від 30.11.2016 №867 та від 03.03.2021 №407
Ідентифікаційний номер набору даних	Oec928d3-8657-4628-aca8-73b8e4078daa

**Набір даних, що використовувався для тренування моделі штучного інтелекту**

Рисунок Г.10 – Набір даних, що використовувався для тренування моделі ШІ

# Conclusion

Виконана магістерська робота є важливим кроком у розробці експертної системи для супроводу реєстрації авторських прав. Вона демонструє важливість інтеграції штучного інтелекту, використання сучасних веб-технологій та спрямована на автоматизацію процесів. Ця система значно спрощує та прискорює процес реєстрації авторських прав, зокрема, завдяки використанню моделі штучного інтелекту для покращення взаємодії з користувачами за рахунок її тренувань на основі особистих вказівок та баз даних з офіційних джерел.

Результати тестування підтверджують високу продуктивність та надійність системи. Її впровадження має важливе суспільне значення, сприяючи зростанню обізнаності про права інтелектуальної власності. Система відкриває перспективи для подальшого розвитку інноваційних технологій, забезпечуючи більш прозоре та доступне правове середовище у сфері інтелектуальної власності.

Рисунок Г.11 – Висновки

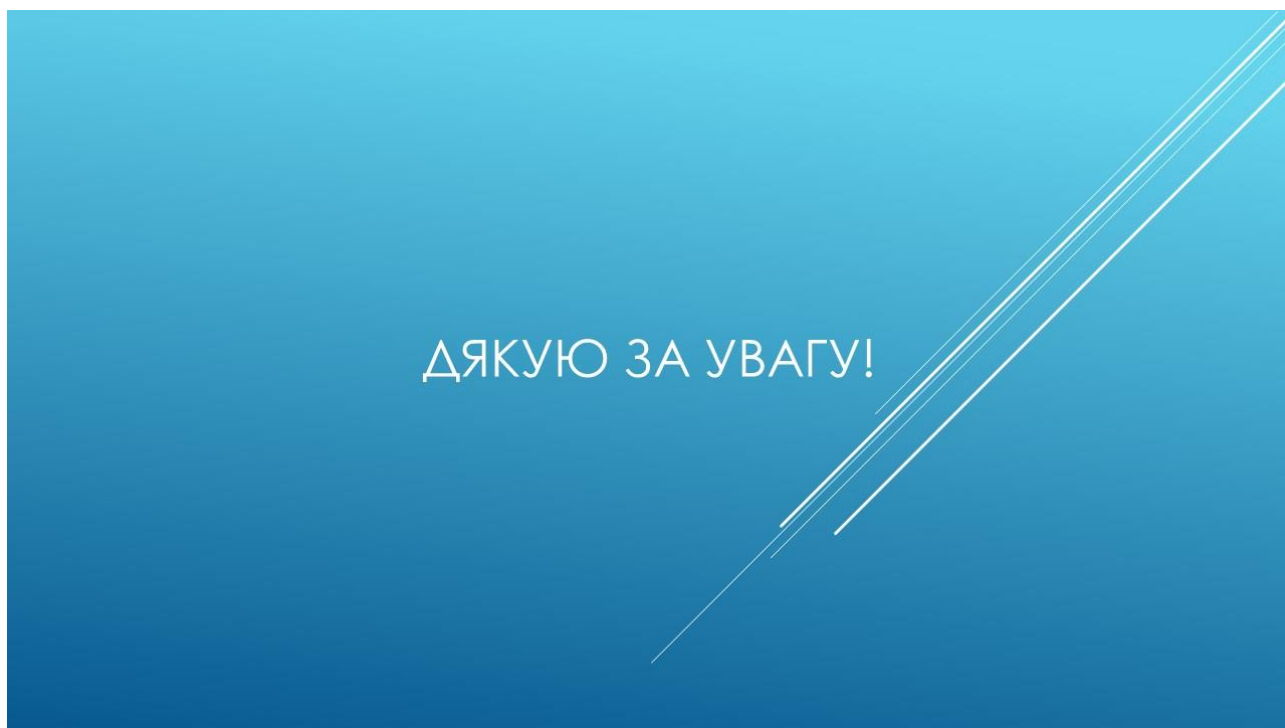


Рисунок Г.12 – Фінальний слайд