

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Управління проектами»

“КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА”

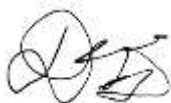
на тему:

“Дослідження процесів управління проектом створення веб-платформи для управління даними в оперативно-розшуковій діяльності”

Студента 2-го курсу групи УПз-21

Дмитра ЦИМБАЛА

(ім'я, прізвище)



(підпис студента)

Науковий керівник:

к.е.н., доцент

(науковий ступінь, вчене звання)

Анна КОЛОМІЄЦЬ

(ім'я, прізвище)

(дата)

(підпис)

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)

Завідувач кафедри
технологій
управління

(підпис)

Віктор МОРОЗОВ

(прізвище, ініціали)

(дата)

Київ – 2025

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Управління проектами

ЗАТВЕРДЖУЮ

Завідувач кафедри
професор Морозов В.В.

“_____” _____ 20__ року

**ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент Цимбал Дмитро Віталійович _____

Група __УПз-21__

1. Тема кваліфікаційної роботи

«Дослідження процесів управління проектом створення веб-платформи для управління даними в оперативно-розшуковій діяльності»

Затверджена наказом від “16” червня 2025 р. № 15.

2. Строк подання студентом готової роботи - “ 9 ” грудня 2025 р.

3. Цільова установка та вихідні дані до роботи: розробка концептуальної та управлінської моделі проекту створення веб-платформи для управління оперативно-розшуковими даними з дотриманням найвищих стандартів безпеки (ISO 27001, RBAC). Досягнення поставлених цілей, включаючи підвищення операційної ефективності правоохоронних органів та економічне обґрунтування

проєкту (NPV/CBA). Вихідні дані: результати системного аналізу існуючих інформаційних систем (ЄРДР, NCIC, Інтерпол), нормативно-правова база України (ЗУ "Про ОРД", ЗУ "Про захист персональних даних") та світові стандарти управління (PMBOK, Agile).

4. Зміст роботи: аналіз сучасних моделей управління проєктами та обґрунтування гібридної методології (Agile/Waterfall). Розробка концептуальної та архітектурної моделі веб-платформи, моделі управління доступом на основі ролей (RBAC), а також деталізація технологічного стеку (ASP.NET Core, React.js, MySQL). Формування ієрархічної структури робіт (WBS) та організаційної структури (Матриця RACI). Розробка календарного плану (Діаграма Ганта) та визначення критичного шляху (CPM). Детальний розрахунок капітальних витрат (CAPEX) та економіко-математичне обґрунтування доцільності проєкту (NPV/CBA). Валідація отриманих результатів та формулювання рекомендацій щодо подальшого розвитку.

5. Перелік графічного матеріалу (слайдів)

Перелік графічного матеріалу (слайдів): Мета роботи та актуальність дослідження. Обґрунтування вибору методології (Гібридний підхід). Концепція проєкту та Діаграма потоків даних (DFD). Модель функціональних вимог (Use Case). Логічна модель бази даних (ERD). Модель управління доступом (RBAC). Модель організаційної структури та Матриця RACI. Модель WBS проєкту. Діаграма Ганта та Критичний Шлях (CPM). Кошторис проєкту (CAPEX). Графік NPV/CBA та Термін Окупності. Результати валідації (Тестування швидкості пошуку). Висновки.

6. Календарний план виконання роботи:

№ п/п	Назва частин роботи	План виконання роботи
1.	Визначення літературних джерел з предмету дослідження	10.09.2025
2.	Збір і вивчення матеріалів досліджуваного підприємства	15.09.2025
3.	Складання розгорнутого плану кваліфікаційної роботи	20.09.2025
4.	Ознайомлення керівника з розгорнутим планом кваліфікаційної роботи, внесення змін	22.09.2025
5.	Підготовку розділу 1	01.10.2025
6.	Підготовка розділу 2	17.10.2025
7.	Підготовка розділу 3	01.11.2025
8.	Підготовка розділу 4	10.11.2025
9.	Оформлення кваліфікаційної роботи	25.11.2025
10.	Передача кваліфікаційної роботи рецензенту для рецензування	24.11.2025
11.	Передача наукової роботи науковому керівнику	07.12.2025
12.	Попередній захист кваліфікаційної роботи	09.12.2025
13.	Захист роботи	22-23.12.2025

Дата видачі завдання “16” червня 2025 р.

Керівник роботи _____ к.е.н., доцент, Анна КОЛОМІЄЦЬ
(посада, прізвище, ім'я, по батькові)

_____ (підпис)

Завдання прийняв до виконання студент групи УПз-21

_____ Дмитро ЦИМБАЛ _____

(прізвище, ім'я, по батькові)



_____ (підпис)

ЗМІСТ

АНОТАЦІЯ	7
ВСТУП.....	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ПРОЄКТУ	14
1.1. Огляд наявних теоретичних концепцій управління проєктами.....	14
1.1.1. Класичні теорії управління проєктами.....	15
1.1.2. Сучасні підходи до виконання проєктних дій	16
1.2. Характеристика об'єкта дослідження та критичний аналіз існуючих систем	17
1.2.1. Характеристика об'єкта, для якого розробляється система	18
1.2.2. Критичний аналіз існуючих інформаційних систем	19
1.3. Методики системного аналізу та управлінські виклики проєкту	21
1.3.1. Ключові виклики та проблемна область автоматизації ОРД.....	22
1.3.2. Застосування методик системного аналізу	23
1.4. Інвестиційні передумови, проєктні альтернативи та обґрунтування	26
1.4.1. Огляд літературних джерел та нормативно-правове підґрунтя проєкту	26
1.4.2. Інвестиційні передумови, проєктні альтернативи та обґрунтування життєздатності проєкту	28
1.5. Формування паспорту проєкту, визначення зацікавлених сторін, постановка задач дослідження.....	30
1.5.1. Формування паспорту проєкту.....	30
1.5.2. Зацікавлені сторони, продукти та етапи проєкту	34
1.5.3. Постановка задач дослідження.....	35
РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ ...	37
2.1. Розробка концептуальної моделі інформаційної системи	37
2.2. Визначення функціональних та нефункціональних вимог до продукту....	39
2.3. Формування Use Case діаграми для функціональних вимог	42
2.4. Розробка концептуальної та логічної моделей бази даних проєкту	45
2.4.1. Концептуальна модель бази даних	46

2.4.2. Логічна модель бази даних	48
2.5. Модель архітектури системи та технологічного стеку	48
2.6. Модель управління доступом на основі ролей (RBAC).....	50
2.7. Модель управління ризиками проєкту	51
РОЗДІЛ 3. УПРАВЛІНСЬКЕ ПЛАНУВАННЯ ТА ДЕТАЛІЗАЦІЯ ПРОЄКТНОГО РІШЕННЯ	53
3.1. Структура декомпозиції робіт (WBS) та ресурсне забезпечення	53
3.1.1. Матриця відповідальності RACI та ролі команди.....	57
3.2. Календарне планування та критичний шлях.....	59
3.2.1. Діаграма Ганта та управління строками.....	59
3.2.2. Метод критичного шляху (CPM) та аналіз залежностей.....	60
3.3. Кошторис проєкту та оцінка капітальних витрат (CAPEX)	62
3.4. Опис структури програмного забезпечення.....	65
3.5. Розробка алгоритмів та інтерфейсів програмного забезпечення	66
РОЗДІЛ 4. ВАЛІДАЦІЯ РЕЗУЛЬТАТІВ ТА ОЦІНКА ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ	71
4.1. Економіко-математичне обґрунтування (NPV/CBA).....	71
4.2. Валідація результатів та аналіз відповідності вимогам	72
4.3. Обґрунтування адекватності моделі та рекомендації	75
ВИСНОВКИ	77
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	79
ДОДАТКИ	84
ДОДАТОК А.....	84
ДОДАТОК Б	86
ДОДАТОК В.....	94

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проектом створення веб-платформи для управління даними в оперативно-розшуковій діяльності»

Студент: Цимбал Дмитро Віталійович.

Науковий керівник: Коломієць Анна Степанівна.

Рік захисту - 2025.

Тема роботи присвячена дослідженню та проектуванню веб-платформи для підвищення ефективності управління неструктурованими та структурованими даними в оперативно-розшуковій діяльності.

Метою проекту є розробка веб-платформи, яка оптимізує процеси збору та аналізу інформації, знижує часові витрати на пошук даних та мінімізує людський фактор при формуванні доказової бази. Платформа забезпечить багаторівневу систему доступу (RBAC), що відповідає вимогам конфіденційності, та надасть інструменти для візуалізації складних взаємозв'язків між суб'єктами, об'єктами та подіями.

Об'єктом дослідження є процеси управління оперативно-розшуковими даними, що реалізуються правоохоронними органами за допомогою інформаційних систем.

Предметом дослідження є методи, процеси та інструменти управління проектом створення веб-платформи для обробки оперативно-розшукових даних.

Наукова новизна роботи полягає в удосконаленні методики системного аналізу IT-проектів для державного сектору шляхом інтеграції оцінки правових та організаційних ризиків у фазі планування, розробці концептуальної та логічної моделі бази даних, оптимізованої для швидкого агрегування та аналізу великих обсягів різномірних даних.

Кваліфікаційна робота складається із анотації, вступу, основної частини, що включає чотири розділи, висновків, списку використаних джерел та додатків.

Перший розділ присвячено системному аналізу проблемної області, дослідженню існуючих аналогів (ЄРДР, NCIS, Інтерпол) та обґрунтуванню доцільності проєкту.

Другий розділ описує розробку концептуальних та управлінських моделей, включаючи формалізацію вимог, розробку архітектури системи та детальний аналіз управління ризиками проєкту.

Третій розділ містить опис реалізації інформаційного та програмного забезпечення веб-платформи, включаючи розробку алгоритмів агрегації даних, структуру бази даних та опис технологічного стеку.

Четвертий розділ присвячено оцінці ефективності проєкту, проведенню економічного обґрунтування (NPV, CBA), аналізу адекватності розроблених моделей та валідації практичних результатів.

На основі проведеного аналізу, зроблено висновки щодо значної економічної та операційної ефективності впровадження платформи, що обґрунтовує доцільність її подальшої реалізації.

Робота містить 78 сторінок без додатків, 22 рисунків, 11 таблиць.

Ключові слова: проєктний менеджмент, управління даними, оперативно-розшукова діяльність, веб-платформа, інформаційна система, системний аналіз, управління проєктами, економічна ефективність, моделі управління, ризики проєкту, концептуальна модель, база даних.

ВСТУП

Актуальність теми. В сучасних умовах стрімкого розвитку інформаційних технологій дедалі більшої важливості набуває проблема ефективного управління оперативними даними у сфері правоохоронної діяльності. Спецслужби та інші правоохоронні органи використовують великі масиви інформації для здійснення оперативно-розшукової діяльності, зокрема при виявленні, розшуку та встановленні осіб, які можуть бути причетні до кримінальних правопорушень. Традиційні підходи до зберігання та обробки таких даних нерідко виявляються малоефективними через їхню розподіленість у різних інформаційних системах, відсутність централізованого управління та недостатню інтегрованість з автоматизованими механізмами аналізу. Це ускладнює оперативне реагування на запити слідчих та аналітиків, підвищує ризики втрати або спотворення даних, а також створює додаткові загрози безпеці. Розробка єдиної веб-платформи для управління оперативно-розшуковими даними здатна значно покращити ефективність обробки інформації, забезпечити швидкий пошук та аналіз відомостей про осіб, що перебувають у розшуку, а також підвищити рівень безпеки при роботі з конфіденційними даними. Таким чином, дослідження у сфері розробки веб-платформи для управління оперативно-розшуковими даними є актуальним та має практичне значення для правоохоронних структур та спецслужб, що займаються збором, обробкою та аналізом інформації у сфері розвідки та контр-розвідки.

Мета та завдання дослідження. Метою даної курсової роботи є розробка та впровадження веб-платформи для управління оперативно-розшуковими даними, що дозволить централізовано зберігати, обробляти та аналізувати інформацію про розшукуваних осіб, їхні зв'язки, місце проживання, транспортні засоби та інші атрибути, важливі для слідчих та оперативних працівників.

Основні завдання дослідження. Для досягнення поставленої мети необхідно вирішити такі завдання:

1) Провести системний та управлінський аналіз проблемної області:

- дослідити існуючі інформаційні системи та аналоги;
- визначити ключові проблеми та виклики в автоматизації ОРД;
- сформуванати ключові цілі проєкту з використанням SWOT та PESTEL.

2) Розробити концептуальні моделі системи та підходів до управління проєктом:

- розробити концептуальну, логічну та (за потреби) математичну моделі;
- сформуванати Use Case/Context/DFD-моделі;
- визначити основні ролі та взаємодії (стейкхолдери).

3) Деталізувати структуру програмного рішення в межах плану проєкту:

- визначити архітектуру системи та її компоненти (Backend/Frontend);
- спроектувати базу даних;
- розробити WBS, календарний графік, RACI.

4) Провести розрахунки вартості та ресурсного забезпечення:

- оцінити CAPEX;
- визначити ресурси (команда, компетенції, середовище);
- сформуванати організаційну структуру команди.

5) Виконати економіко-математичне обґрунтування доцільності проєкту:

- провести NPV/CBA;
- оцінити інвестиційні ризики та переваги;
- обґрунтувати вибір найкращої альтернативи проєкту.

6) Реалізувати проєктне рішення на рівні MVP:

- реалізувати бекенд на ASP.NET 8.0 та MySQL;
- розробити фронтенд на React.js, Redux Toolkit, MUI;
- налаштувати REST API.

7) Провести валідацію результатів та сформувавши рекомендації з удосконалення:

- виконати порівняння з початковими вимогами;
- оцінити результативність розробленої моделі;
- надати рекомендації щодо подальшого розвитку.

Об'єкт дослідження. Об'єктом дослідження є процеси управління оперативно-розшуковими даними, що реалізуються правоохоронними органами за допомогою інформаційних систем.

Предмет дослідження. Предметом дослідження є методи, процеси та інструменти управління проектом створення веб-платформи для обробки оперативно-розшукових даних.

Методи дослідження:

- 1) аналіз літературних та інформаційних джерел – для вивчення існуючих рішень та підходів до автоматизації обліку оперативних даних;
- 2) системний аналіз – для визначення архітектури та компонентів веб-платформи;
- 3) методи об'єктно-орієнтованого аналізу та проектування методи об'єктно-орієнтованого програмування (метод розробки, а не дослідження) – для розробки серверної частини веб-додатку на основі C# та ASP.NET 8.0;
- 4) методи веб-розробки – для створення фронтенду з використанням React.js, Redux Toolkit та MUI;

- 5) методи моделювання баз даних бази даних та SQL (технологія) – для проєктування та оптимізації структури MySQL-бази даних;
- 6) методи управління проєктами (PMBOK/Agile/Scrum) – для планування, організації та контролю робіт;
- 7) структурно-функціональне моделювання – для розроблення концептуальної моделі платформи;
- 8) порівняльний аналіз – для оцінювання альтернативних технологій та рішень;
- 9) методи оцінювання ризиків – для визначення та мінімізації ризиків під час проєкту.

Наукова новизна роботи полягає в покращенні методики системного аналізу IT-проєктів для державного сектору шляхом інтеграції правових та безпекових аспектів у фазі планування.

Практичне значення отриманих результатів. Запропоновані результати дослідження можуть бути використані при плануванні, організації та контролі проєктів зі створення інформаційних систем для правоохоронних органів. Розроблена веб-платформа може бути застосована в правоохоронних органах, аналітичних відділах та інших структурах, що займаються збором та обробкою оперативно-розшукової інформації. Очікувані переваги використання системи:

- прискорення пошуку осіб у розшуку завдяки інтеграції централізованої бази даних;
- автоматизація процесів обліку та аналізу розшукових відомостей;
- підвищення безпеки даних шляхом використання механізмів аутентифікації та контролю доступу;
- гнучкість у розширенні функціоналу завдяки використанню модульного підходу до розробки.

Практична значущість результатів полягає також у можливості використання описаних підходів до управління проектом — зокрема в частині планування, визначення вимог, управління ризиками та ресурсами при створенні аналогічних систем у сфері правоохоронної діяльності. Таким чином, запропонована веб-платформа дозволяє не лише підвищити ефективність оперативно-розшукової діяльності, а й мінімізувати ризики, пов'язані з обробкою великої кількості чутливої інформації.

Апробація результатів дослідження. Основні положення та результати, отримані в ході виконання даної кваліфікаційної роботи, пройшли апробацію на конференції. У рамках конференції було представлено доповідь на тему «Дослідження процесів управління проектом створення веб-платформи для управління даними в оперативно-розшуковій діяльності». Апробація підтвердила актуальність обраної гібридної моделі управління, що поєднує методи РМВОК та Agile, а також практичну значущість архітектурних рішень у сфері захисту конфіденційної інформації (RBAC, JWT).

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ПРОЄКТУ

1.1. Огляд наявних теоретичних концепцій управління проєктами

Управління проєктом є клавш у вирішенні комплексних задач з автоматизації державного сектору, які включають ланцюжкову координацію від інших багатьох процедур і забезпечення ефективності кожного етапу реалізації [15]. Проєкти в оперативно-розшуковій сфері мають низку унікальних рис по відношенню до комерційних чи інфраструктурних ініціатив, адже їхні всі цілі, аудиторія та результати нерозривно пов'язані із забезпеченням національної безпеки та конфіденційності даних [13]. Запровадження інформаційних технологій в проєкти ОРД відкриває величезний потенціал на те, щоб цифрові знаходять процеси збору, моніторингу та аналізу оперативних даних. Тобто, Це дозволяє створити інструменти, які сприятимуть прозорості використання ресурсів, покращенню комунікації між підрозділами та ефективному управлінню обмеженими ресурсами (насамперед, часовими). Користування спеціалізованими платформами дозволяє об'єднати різних учасників – аналітиків, слідчих і адміністраторів баз даних.

Водночас, управління проєктами безпеки пов'язане з деякими унікальними викликами, до яких належать дотримання нормативних актів, розподіл високочутливих ресурсів, врахування потреб різних користувачів з різними рівнями доступу та забезпечення найвищого рівня довіри. Ці фактори вимагають використання сучасних теоретичних концепцій управління проєктами, таких як Agile, PMBOK або PRINCE2, які адаптовані до умов роботи з чутливою інформацією. Таким чином, ефективне управління проєктом створення веб-платформи для ОРД залежить від впровадження інноваційних підходів, використання цифрових технологій та врахування правових та безпекових аспектів, що сприяє досягненню ключових цілей – підвищенню операційної ефективності та мінімізації ризиків витоку даних.

1.1.1. Класичні теорії управління проєктами

Класичні теорії управління проєктами є основою для багатьох підходів, які використовуються в різних сферах, включаючи розробку захищених інформаційних систем [3]. Ці теорії забезпечують перевірені часом методи та інструменти для планування, організації та реалізації проєктів, що дозволяють досягати поставлених цілей ефективно та з мінімальними ризиками. Однією з перших і найвідоміших класичних теорій є модель управління проєктами Генрі Гантта, яка базується на використанні Діаграми Гантта. Цей інструмент дозволяє візуалізувати часовий графік виконання завдань у проєкті, встановлювати віхи та відстежувати прогрес.

Метод критичного шляху (Critical Path Method) і метод PERT (Program Evaluation and Review Technique) є іншими популярними класичними підходами. Вони зосереджені на визначенні критичних завдань і оптимізації ресурсів [15]. У проєкті створення веб-платформи для ОРД ці методи можуть бути використані для планування послідовності робіт і виявлення вузьких місць, що впливають на строки виконання проєкту (наприклад, інтеграція з існуючими реєстрами або фаза налаштування багаторівневої системи безпеки) [16].

У Європі та США класичні методи мережевого планування, такі як метод стрілочних діаграм (ADM) і метод попередніх діаграм (PDM), широко використовувалися для організації складних процесів. Ці підходи дозволяють візуалізувати зв'язки між завданнями, що особливо корисно для проєктів з великою кількістю учасників і ресурсів.

Методологія CRISP-DM (Cross-Industry Standard Process for Data Mining) також може знайти своє місце у вашому проєкті, особливо на етапах аналізу даних та розробки алгоритмів. Вона включає шість етапів: розуміння бізнесу, аналіз даних, підготовку даних, моделювання, оцінку та впровадження. У

контексті нашого проєкту це може бути застосовано для обробки великих масивів оперативних даних, аналізу їх зв'язків та формування аналітичних звітів.

Деякі інші класичні методи, такі як теорія обмежень, методологія водоспаду, а також ітеративні та приростові підходи, є корисними для побудови структурованого плану дій. Вони сприяють систематичному вирішенню завдань, забезпечуючи прозорість, адаптивність та підзвітність на всіх етапах проєкту [4].

Таким чином, класичні теорії управління проєктами є базовими інструментами, які можна адаптувати для ефективного управління проєктом створення веб-платформи для ОРД. Ці підходи допомагають забезпечити чітке планування, контроль ресурсів і досягнення стратегічних цілей.

1.1.2. Сучасні підходи до виконання проєктних дій

Сучасні підходи до виконання проєктних дій формуються на основі постійного вдосконалення методологій управління, спрямованих на досягнення чітко визначених цілей, підвищення ефективності роботи команди та оптимізацію використання ресурсів [5]. У контексті реалізації веб-платформи для ОРД, що має на меті об'єднання інформаційних ініціатив та даних, важливо враховувати інтеграцію сучасних управлінських практик.

Основними принципами сучасного управління проєктами є прозорість, адаптивність і зосередженість на результаті. Ці принципи забезпечують не лише досягнення цілей проєкту, але й задоволення очікувань усіх зацікавлених сторін, таких як оперативні працівники, аналітики та адміністратори баз даних.

Сучасні методології відіграють ключову роль у реалізації проєкту, забезпечуючи структурованість, гнучкість і ефективність. Наприклад, методологія PMI (Project Management Institute) дозволяє чітко розділити проєкт на етапи – від ініціації до завершення, забезпечуючи систематичний підхід до управління всіма аспектами, включаючи розробку архітектури безпеки та запуск платформи [6].

Agile, зі своєю ітеративною природою, дозволяє швидко адаптуватися до змін і постійно враховувати потреби користувачів [7]. У рамках цього підходу використання Scrum забезпечує розподіл роботи на спринти, де кожна ітерація завершується створенням працюючого інкременту продукту. Це також сприяє прозорості процесів через регулярні зустрічі команди, а також гнучкому реагуванню на змінювані вимоги до аналітичного функціоналу.

Lean Project Management допомагає зосередитися на максимізації цінності продукту при мінімальних витратах, що є важливим для оптимізації фінансування розробки і підтримки платформи. PRINCE2 додає до цього чіткість у розподілі ролей і відповідальності, що полегшує координацію між різними учасниками проекту.

Використання Critical Chain Project Management дозволяє оптимізувати ресурси і скорочувати часові рамки виконання ключових етапів, наприклад, налаштування системи аудиту або інтеграції даних з різних реєстрів [8]. Управління ризиками відіграє важливу роль, дозволяючи завчасно ідентифікувати потенційні загрози і розробляти стратегії їх подолання [16]. Застосування цих підходів забезпечує гнучкість та ефективність управління проектом. Платформа для ОРД виграє від адаптації цих методів завдяки їх здатності організовувати процеси, підвищувати якість прийняття рішень та зменшувати витрати. Інтеграція сучасних підходів дозволяє не лише досягти цілей проекту, але й підтримати прозорість та ефективну взаємодію між усіма учасниками оперативно-розшукового процесу.

1.2. Характеристика об'єкта дослідження та критичний аналіз існуючих систем

Сучасні інформаційні технології відіграють ключову роль у сфері правоохоронної діяльності, зокрема у процесах збору, обробки та аналізу

оперативних даних. Спецслужби та правоохоронні органи багатьох країн активно використовують спеціалізовані системи для ведення баз даних розшукуваних осіб, аналізу їх зв'язків, обліку транспортних засобів, майна та інших важливих атрибутів. Впровадження таких систем значно підвищує ефективність оперативно-розшукової діяльності, проте наявні рішення мають низку обмежень, що стримують їхнє використання у повному обсязі.

1.2.1. Характеристика об'єкта, для якого розробляється система

Об'єктом даного дослідження є процеси управління оперативно-розшуковими даними, що реалізуються правоохоронними органами та спеціальними службами. Оперативно-розшукова діяльність є важливою складовою правоохоронної системи, спрямованою на запобігання, виявлення та розкриття злочинів, і охоплює широкий спектр заходів, що реалізуються спеціальними службами та органами правопорядку. Ефективність цієї діяльності напряму залежить від швидкості отримання, обробки та аналізу інформації про розшукуваних осіб, їхні контакти, пересування, а також інші дані, що можуть бути використані в розслідуваннях [10].

Серед основних користувачів майбутньої інформаційної системи, чії потреби необхідно врахувати в першу чергу, можна виділити такі категорії:

- оперативні працівники - співробітники правоохоронних органів, які здійснюють безпосередній розшук осіб, аналіз зв'язків підозрюваних, перевірку інформації про їхні пересування, контакти та майно. Їм потрібен швидкий і захищений доступ до первинних даних;
- аналітики - спеціалісти, які здійснюють глибокий аналіз даних, встановлюють закономірності між подіями та особами, що можуть бути причетні до злочинної діяльності. Їм необхідні розширені алгоритми пошуку та інструменти візуалізації складних зв'язків;

- керівники підрозділів - особи, що відповідають за організацію роботи та прийняття стратегічних рішень на основі аналітичних даних, отриманих із системи. Їм потрібен функціонал аудиту, контролю доступу та формування зведеної звітності.

Розробка інформаційної системи спрямована на автоматизацію та централізацію управління оперативно-розшуковими даними. Правоохоронні органи мають жорстко регламентовані вимоги щодо роботи з інформацією, тому система повинна відповідати високим стандартам безпеки.

Основні нефункціональні вимоги до функціонування системи, що впливають зі специфіки об'єкта, включають:

- обмежений доступ до даних, розмежований відповідно до рівня повноважень користувачів;
- аудит дій користувачів, який дозволить відстежувати будь-які зміни та запити в системі для контролю безпеки;
- можливість автоматизованого аналізу зв'язків між особами, об'єктами та подіями;
- гнучкість у налаштуванні запитів та фільтрів для ефективного пошуку.

Використання сучасних технологій, зокрема ASP.NET 8.0 для бекенду, React.js для фронтенду та MySQL для бази даних, дозволить створити масштабовану та продуктивну систему, що відповідатиме цим критичним вимогам. Таким чином, розроблювана веб-платформа призначена для використання в рамках оперативно-розшукової діяльності, забезпечуючи правоохоронні органи швидким та безпечним доступом до необхідної інформації.

1.2.2. Критичний аналіз існуючих інформаційних систем

Як показує аналіз, існуючі системи мають низку обмежень, що стримують їхнє використання у повному обсязі та створюють управлінські ризики, результати зображені в табл 1.1.

Аналіз існуючих інформаційних систем

Назва системи	Функціонал	Обмеження
Єдиний реєстр досудових розслідувань	Використовується для реєстрації та обліку кримінальних проваджень.	Доступ мають лише слідчі та прокурори. Система не забезпечує гнучких інструментів пошуку та аналітики, необхідних для оперативно-розшукової діяльності. Критичним є відсутність інтеграції зовнішніх джерел інформації, що ускладнює взаємодію між різними відомствами.
Інформаційні бази Інтерполу	Містять дані про розшукуваних злочинців, викрадені документи, транспортні засоби та зброю на міжнародному рівні.	Інформація не завжди оперативно оновлюється, доступ обмежений лише правоохоронними органами країн-членів Інтерполу. Механізм обміну даними є доволі повільним, що не відповідає вимогам оперативного реагування на національному рівні.

Система NCIC	Одна з найбільш розвинених баз даних, що функціонує у режимі реального часу і включає детальну інформацію про злочинців, зниклих осіб, викрадені автомобілі.	Впровадження подібної системи в інших країнах є складним завданням через особливості законодавчого регулювання та потребу в централізованому адмініструванні.
Локальні поліцейські бази даних	Використовуються для внутрішнього контролю та обліку злочинної діяльності.	Мають обмежений доступ і зазвичай не передбачають можливості обміну даними з іншими відомствами.

Підводячи підсумки, можна констатувати, що незважаючи на широкий спектр наявних інформаційних систем, більшість із них мають низку спільних проблем: цільова орієнтованість (наприклад, кримінальний облік, а не оперативний аналіз), що ускладнює їх інтеграцію; обмежений доступ до зовнішніх джерел інформації; неможливість комплексного аналізу зв'язків між різними фігурантами справ.

Отже, існує стратегічна потреба у створенні уніфікованої веб-платформи, яка дозволить централізовано зберігати дані, здійснювати їх швидку обробку, інтегрувати інформацію з різних джерел та забезпечувати гнучкі інструменти аналізу для оперативних підрозділів. В контексті управління проектами, саме неінтегрованість та негнучкість наявних рішень створює управлінські ризики (часові, бюджетні) та знижує загальну ефективність роботи організації.

1.3. Методики системного аналізу та управлінські виклики проекту

Успішне здійснення оперативно-розшукової діяльності напряму залежить від ефективності збирання, обробки та аналізу даних про осіб, які перебувають у

розшуку, їхні зв'язки, транспортні засоби, місце проживання та інші атрибути. Однак сучасні підходи до ведення таких баз даних залишаються недостатньо ефективними через низку об'єктивних факторів, що створює управлінські виклики для керівництва правоохоронних органів.

1.3.1. Ключові виклики та проблемна область автоматизації ОРД

Наявні інформаційні системи часто є розрізненими, не мають єдиної централізованої структури та обмежені у можливостях інтеграції. У результаті оперативні підрозділи стикаються з труднощами при пошуку інформації, що може суттєво уповільнити процеси ухвалення рішень.

Основні проблеми:

- 1) розрізненість та фрагментація даних: Інформація про розшукуваних осіб та підозрюваних зберігається у різних системах, які функціонують у межах окремих відомств. Недостатня синхронізація ускладнює доступ до актуальної інформації та змушує оперативних співробітників надсилати окремі запити, що уповільнює процес отримання необхідних відомостей;
- 2) необхідність швидкого пошуку та аналітики великих масивів даних: Обсяг інформації постійно зростає. Традиційні реляційні бази даних не завжди здатні швидко обробляти запити на виявлення зв'язків між об'єктами. Це вимагає використання індексації, оптимізованих пошукових алгоритмів та спеціалізованих рішень, таких як графові бази даних, які дозволяють швидко знаходити зв'язки між різними об'єктами;
- 3) інформаційна безпека та контроль доступу: Оперативна інформація є конфіденційною та потребує високого рівня захисту від несанкціонованого доступу або витоку даних [5]. Це вимагає жорсткого контролю доступу (RBAC) та аудиту дій користувачів, а також технологій шифрування. Недостатньо захищені системи можуть стати мішенню для кібератак, що становить серйозну загрозу для національної безпеки;

- 4) робота з різноформатними даними: Оперативна інформація містить не лише текстові записи, а й зображення, відеофайли, аудіозаписи та інші мультимедійні матеріали. Це ускладнює збереження та обробку таких даних у традиційних базах даних та вимагає інтеграції систем машинного навчання та комп'ютерного зору.

Проекти автоматизації ОРД зазвичай не розпочинаються з нуля, оскільки необхідно враховувати попередні інвестиції в інфраструктуру та програмне забезпечення. Існують застарілі системи, які можуть вимагати оновлення або інтеграції. Це створює проблему забезпечення взаємодії між новими та існуючими платформами, що включає сумісність як на рівні програмного, так і апаратного забезпечення. Окрім технічних викликів, критично важливими є правові аспекти, такі як забезпечення конфіденційності, безпеки даних та захисту персональної інформації відповідно до нормативних вимог. Крім того, економічним аспектом є забезпечення гарного співвідношення витрат і вигод від впровадження та експлуатації системи. Це особливо важливо в умовах обмеженого державного фінансування. Успішне впровадження також передбачає зміни в організаційній структурі підрозділів, з орієнтацією на підвищення ефективності оперативної роботи, а не на внутрішні бюрократичні процеси.

1.3.2. Застосування методик системного аналізу

Для систематизації проблемної області та обґрунтування управлінських рішень щодо проєктування веб-платформи необхідне застосування методик системного аналізу. Це дозволяє прозоро оцінити вплив зовнішнього середовища, внутрішніх ресурсів та ризиків на успіх проєкту, що є ключовою вимогою для магістерської роботи.

PESTEL аналіз - аналіз політичних, економічних, соціальних, технологічних, екологічних та правових факторів, результати зображені в табл. 1.2.

PESTEL аналіз

Фактор	Вплив на проєкт
Політичний (P)	Високий рівень державної підтримки цифровізації та реформування правоохоронних органів. Необхідність узгодження з виконавчою владою.
Економічний (E)	Обмеженість бюджетного фінансування (потреба в економічному обґрунтуванні NPV/CBA). Зростання вартості кваліфікованих ІТ-фахівців. Важливість співвідношення витрат і вигод.
Соціальний (S)	Низький рівень цифрової грамотності частини персоналу (потреба у навчанні та простому інтерфейсі). Високі вимоги суспільства до прозорості та законності дій правоохоронців. Інклюзивність (доступність) для всіх уповноважених користувачів.
Технологічний (T)	Стрімкий розвиток AI/ML-алгоритмів (для аналізу зв'язків) та хмарних технологій. Можливість інтеграції з новітніми інструментами аналізу, але ризик несумісності з застарілими системами.
Правовий (L)	Суворі обмеження ЗУ «Про захист персональних даних» та ЗУ «Про ОРД». Вимоги до державної сертифікації ПЗ та криптографічного захисту.

SWOT-аналіз - аналіз сильних та слабких сторін, можливостей та загроз, результати зображені в табл. 1.3.

Таблиця 1.3

SWOT-аналіз

Група	Фактори
Сильні сторони (Strengths)	Готова інженерна база (КР), єдина централізована структура, використання сучасного стеку (ASP.NET, React.js), висока швидкість пошуку, чіткість регуляторних вимог до кінцевого продукту.
Слабкі сторони (Weaknesses)	Необхідність інтеграції з застарілими державними реєстрами, висока вартість розробки та впровадження, потреба у спеціалізованому ІТ-персоналі для підтримки. Ризик ускладнень у технічному обслуговуванні старих програм.
Можливості (Opportunities)	Отримання державного фінансування або грантів на кібербезпекові проєкти, масштабування системи на інші відомства, впровадження нових AI-моделей для прогностичного аналізу. Покращення якості обслуговування (швидкість, точність) для кінцевих користувачів.
Загрози (Threats)	Зміни у законодавстві щодо обробки конфіденційних даних, кібератаки та внутрішні загрози витоку інформації, опір персоналу до впровадження нової системи. Проблеми ремонтпридатності системи.

Зважаючи на всі вищезазначені виклики, можна зробити висновок, що автоматизація оперативно-розшукових процесів потребує створення нової інформаційної платформи, яка відповідатиме ключовим вимогам: централізоване

зберігання, швидкий пошук, багаторівневий захист інформації та підтримка різноформатних файлів. Таке рішення дозволить значно підвищити ефективність розшукової діяльності та забезпечити оперативний доступ до критично важливої інформації. Оцінка управлінських викликів через PESTEL та SWOT-аналіз підкреслює, що успіх проєкту залежить не лише від технічної реалізації, а й від адекватності управлінської моделі, здатної мінімізувати юридичні та фінансові ризики, а також забезпечити високу прийнятність системи кінцевими користувачами [14].

1.4. Інвестиційні передумови, проєктні альтернативи та обґрунтування

1.4.1. Огляд літературних джерел та нормативно-правове підґрунтя проєкту

Розробка інформаційних систем для оперативно-розшукової діяльності є складним завданням, яке поєднує технічні, правові та організаційні аспекти. У зв'язку з цим необхідно враховувати як наукові дослідження у сфері інформаційних технологій та кібербезпеки, так і чинні нормативно-правові акти, що регламентують використання та захист оперативних даних. Аналіз наукової літератури свідчить про те, що проблеми автоматизації правоохоронної діяльності досліджуються як у вітчизняних, так і в зарубіжних роботах. У численних працях розглядаються методи збирання, обробки та аналізу оперативної інформації, алгоритми оптимізації пошуку в базах даних та питання захисту конфіденційних відомостей [7]. Так, у роботах сучасних дослідників особливу увагу приділено використанню технологій штучного інтелекту для автоматичного виявлення закономірностей у великих обсягах криміналістичних даних [30]. Впровадження алгоритмів машинного навчання у сферу розшукової діяльності дозволяє значно підвищити ефективність аналізу підозрюваних, їхніх контактів та зв'язків [28].

Серед найбільш важливих джерел інформації, що використовуються для розробки інформаційних систем правоохоронного спрямування, є міжнародні стандарти з управління інформаційною безпекою. Так, ISO 27001 встановлює вимоги щодо захисту інформації, зокрема принципи розмежування доступу, шифрування даних та їх резервного копіювання [26]. У цьому ж напрямі працюють стандарти NIST (National Institute of Standards and Technology), які регламентують технічні аспекти кібербезпеки для державних структур. Важливим аспектом є нормативно-правове регулювання ведення та обробки оперативних даних в Україні. Закон України «Про оперативно-розшукову діяльність» визначає основні принципи збору, зберігання та використання інформації в рамках розслідувань. Зокрема, у ньому регламентується перелік суб'єктів, що мають право здійснювати ОРД, способи отримання даних, а також правила їх використання у кримінальних провадженнях. Відповідно до цього закону, доступ до інформаційних систем, що містять оперативно-розшукові відомості, має бути суворо регламентованим і здійснюватися лише уповноваженими особами.

Окремо слід виділити Закон України «Про захист персональних даних», який регулює обробку інформації, що містить особисті відомості громадян. Оскільки база даних розшукуваних осіб включатиме персональні дані, їхня обробка повинна відповідати вимогам цього закону. Це означає, що система має забезпечувати дотримання принципів правомірності збору даних, їхньої конфіденційності та захисту від несанкціонованого доступу. Крім того, міжнародні стандарти та законодавчі акти Європейського Союзу, такі як GDPR (General Data Protection Regulation), також впливають на підхід до зберігання та використання персональних даних у розшукових системах. Незважаючи на те, що Україна не є членом ЄС, багато національних компаній та державних установ впроваджують механізми захисту даних, що відповідають вимогам GDPR.

Важливим документом у сфері цифрової безпеки також є Національна стратегія кібербезпеки України, що визначає пріоритети державної політики у сфері захисту інформаційних систем. Вона передбачає розробку спеціалізованих інформаційних платформ для правоохоронних органів, що відповідають найвищим стандартам безпеки.

Підсумовуючи аналіз наукових джерел та нормативних документів, можна зробити висновок, що розробка веб-платформи для управління оперативно-розшуковими даними повинна відповідати сучасним вимогам безпеки, законодавчим нормам та міжнародним стандартам. Враховуючи зростаючі ризики кіберзагроз, особливу увагу необхідно приділити питанням аутентифікації користувачів, розмежування рівнів доступу, шифрування та аудиту дій у системі. Таким чином, запропонована платформа буде не лише ефективним інструментом для оперативно-розшукової діяльності, а й відповідатиме актуальним вимогам інформаційної безпеки.

1.4.2. Інвестиційні передумови, проєктні альтернативи та обґрунтування життєздатності проєкту

Згідно з методичними вказівками, на етапі системного аналізу необхідно навести результати інвестиційних досліджень та обґрунтувати життєздатність проєкту. Оскільки проєкт створення веб-платформи для ОРД є соціально-орієнтованим та не комерційним, його інвестиційна доцільність оцінюється через аналіз витрат та вигод (Cost-Benefit Analysis, CBA) та соціально-економічні метрики, а не лише традиційні фінансові показники (NPV, ROI).

Ключові інвестиційні передумови:

- 1) мінімізація операційних витрат (ОРЕХ): хоча розробка вимагає значних капітальних витрат (САРЕХ), централізація даних та автоматизація пошуку дозволяють зменшити трудомісткість і час оперативних працівників на

виконання рутинних запитів. Це є прямою економією бюджетних коштів у довгостроковій перспективі;

2) зниження ризиків: неврегульованість, фрагментація даних та низький рівень захисту існуючих систем створюють високі ризики витоку конфіденційної інформації, що може призвести до значних фінансових та репутаційних втрат. Інвестиції у сучасну захищену платформу є інвестиціями у зниження цих критичних ризиків;

3) потреба у відповідності стандартам: законодавчі вимоги (GDPR, ЗУ «Про ОРД») та міжнародні стандарти (ISO 27001) вимагають від держави впровадження сучасних механізмів захисту. Інвестиції у розробку такої платформи є необхідною умовою для дотримання правового поля.

З проєктними альтернативами та їх обґрунтуванням можна ознайомитись в таблиці 1.4.

Таблиця 1.4

Проєктні альтернативи та їх обґрунтування

Альтернатива	Опис	Обґрунтування відхилення
А1. Модернізація існуючих систем (ЄРДР)	Інтеграція модулів аналітики та розширення доступу до чинних реєстрів.	Неможливість повної інтеграції різнорідних баз даних через їхню застарілу архітектуру. Висока вартість ремонтпридатності та повільна масштабованість.

А2. Використання комерційного COTS-рішення	Придбання готової платформи для управління даними (Commercial Off-the-Shelf) від іноземних постачальників.	Критичні ризики безпеки (іноземна юрисдикція, вразливості), висока вартість ліцензування та неможливість адаптації під специфічне українське законодавство та реєстри.
А3. Розробка власної уніфікованої веб-платформи (Обраний варіант)	Створення єдиної централізованої системи з нуля з урахуванням усіх вимог безпеки та законодавства.	Високі початкові CAPEX, але забезпечує повну відповідність правовим нормам, гнучкість архітектури та контроль над безпекою даних.

З цього можна зробити висновок: на підставі критичного аналізу альтернатив, єдиним життєздатним і безпечним шляхом є Альтернатива А3. Інвестиції у розробку власної платформи є необхідними для забезпечення національної безпеки та операційної ефективності правоохоронних органів, що підтверджує стратегічну доцільність проєкту.

1.5. Формування паспорту проєкту, визначення зацікавлених сторін, постановка задач дослідження.

1.5.1. Формування паспорту проєкту

Паспорт проєкту слугує формалізованим технічним завданням, що визначає рамки, мету та очікувані результати проєкту, результат можна побачити в табл. 1.5.

Паспорт проєкту

<p>Організатор проєкту</p>	<p>Організатором проєкту є правоохоронні органи та спеціальні служби, що здійснюють оперативно-розшукову діяльність. Проєкт створюється для автоматизації управління базами розшукуваних осіб та їхньої аналітики, що допоможе покращити ефективність роботи оперативних підрозділів.</p>
<p>Співвідношення мети проєкту з метою організації</p>	<p>Мета проєкту – створення централізованої веб-платформи для оперативно-розшукової діяльності, що дозволить правоохоронним органам об'єднати, зберігати та аналізувати дані в режимі реального часу. Це сприятиме зменшенню бюрократичного навантаження, підвищенню швидкості доступу до інформації та покращенню загальної ефективності роботи оперативних підрозділів.</p>
<p>Вигоди від реалізації проєкту</p>	<p>Реалізація веб-платформи забезпечить:</p> <ul style="list-style-type: none"> • швидкий доступ до баз даних оперативно-розшукової інформації; • автоматизацію аналітичних процесів, що дозволить оперативникам швидше виявляти зв'язки між фігурантами; • покращення міжвідомчої взаємодії завдяки інтеграції з іншими державними системами; • збільшення рівня безпеки даних за рахунок використання сучасних методів шифрування та контролю доступу;

<p>Очікувані часові рамки роботи</p>	<p>Проект передбачає наступні етапи реалізації:</p> <ul style="list-style-type: none"> • аналіз вимог та проектування архітектури – 1 місяць; • розробка бекенду (ASP.NET 8.0, MySQL) – 2 місяці; • розробка фронтенду (React.js, Redux Toolkit, MUI) – 2 місяці; • інтеграція всіх компонентів та тестування – 1 місяць; • оптимізація безпеки та випуск першої версії – 1 місяць. <p>Загальна тривалість проекту: від 6 місяців</p>
<p>Короткий опис результатів проекту (цілей)</p>	<p>В результаті реалізації проекту буде розроблена функціональна веб-платформа, що забезпечить:</p> <ul style="list-style-type: none"> • збереження та обробку оперативних даних у централізованій базі; • надійний механізм пошуку за багатьма параметрами (ПІБ, місце проживання, транспортні засоби тощо) ; • інструменти аналітики для виявлення зв'язків між об'єктами; • контроль доступу до інформації відповідно до рівня користувача; • можливість інтеграції з іншими державними інформаційними системами.

<p>Бюджет, розміщення, ресурси, доступні для проекту</p>	<p>Бюджет: визначається державними структурами відповідно до фінансування інформаційних систем. Розміщення: серверна інфраструктура буде розташована у захищеному дата-центрі, який відповідатиме стандартам інформаційної безпеки. Ресурси:</p> <ul style="list-style-type: none"> • доступ до державних реєстрів та баз даних; • хмарні сервіси для резервного копіювання; • сертифіковані програмні засоби захисту інформації.
<p>Повноваження менеджера проекту</p>	<p>Менеджер проекту відповідає за:</p> <ul style="list-style-type: none"> • організацію всіх етапів розробки та впровадження платформи; • визначення основних пріоритетів у розробці функціональності; • координацію взаємодії з державними органами та замовниками; • контроль за виконанням вимог безпеки; • організацію тестування та навчання персоналу, який використовуватиме платформу.
<p>Підпис організатора проекту</p>	

1.5.2. Зацікавлені сторони, продукти та етапи проєкту

Для забезпечення успішної реалізації проєкту в умовах державного сектору необхідне чітке визначення всіх зацікавлених сторін та кінцевих продуктів.

Зацікавлені сторони проєкту (Стейкхолдери):

- замовник/ініціатор: керівництво правоохоронних органів, МВС. Роль: Визначає ключові цілі, забезпечує фінансування та приймає кінцевий продукт;
- команда проєкту: менеджер проєкту, Team Lead, Backend/Frontend розробники, ML-інженери, QA, Юристи;
- кінцеві користувачі: оперативні працівники, аналітики, керівники підрозділів. Роль: Основні споживачі функціоналу, чії потреби визначають успіх проєкту;
- партнери/регулятори: органи місцевого самоврядування, Національний центр кібербезпеки, розробники державних реєстрів. Роль: Забезпечують юридичне оформлення, доступ до даних та перевірку безпеки (пентести).

Кінцеві продукти проєкту: проєкт спрямований на створення інтегрованої екосистеми, основними продуктами якої є:

- 1) веб-платформа: основний інструмент з інтегрованими функціями централізації даних, пошуку та аналізу зв'язків;
- 2) система управління даними: структурована система зберігання, що забезпечує надійний захист персональних даних та цілісність інформації відповідно до міжнародних стандартів безпеки;
- 3) модуль аудиту та контролю доступу: інструмент для забезпечення прозорості та відповідності законодавчим вимогам доступу.

Етапи реалізації та Стратегія впровадження: реалізація проєкту потребує ретельно спланованого часу, кожен етап синхронізується з ітераційним підходом

(Scrum), що дозволяє ефективно координувати роботу й досягати цілей у встановлені терміни.

- 1) аналіз та планування, визначення вимог, PESTEL/SWOT-аналіз, формування WBS;
- 2) проєктування та моделювання, розробка архітектури системи, логічної моделі БД, моделювання управління ризиками;
- 3) розробка та інтеграція, створення бекенду, фронтенду, інтеграція компонентів та модулів безпеки;
- 4) тестування та впровадження, комплексне тестування, запуск MVP, оцінка економічної ефективності.

Управління змінами проєкту передбачає оцінку, планування, впровадження та моніторинг будь-яких змін у проєкті. Завдяки ітераційному підходу зміни аналізуються щодо їхнього впливу на терміни, ресурси та якість продукту. Розробляється план впровадження змін, визначаються відповідальні особи, а результати комунікуються із зацікавленими сторонами, забезпечуючи гнучкість та відповідність результатів очікуванням.

1.5.3. Постановка задач дослідження

Метою дослідження є розробка веб-платформи, яка оптимізує процеси збору та аналізу інформації, знижує часові витрати на пошук даних та мінімізує людський фактор при формуванні доказової бази. Платформа забезпечить багаторівневу систему доступу (RBAC), що відповідає вимогам конфіденційності, та надасть інструменти для візуалізації складних взаємозв'язків між суб'єктами, об'єктами та подіями.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) Провести системний та управлінський аналіз проблемної області, існуючих аналогів та сформулювати ключові цілі проєкту з використанням управлінських інструментів (SWOT-аналіз, PESTEL-аналіз);

- 2) Розробити концептуальні, логічні та математичні моделі системи;
- 3) Деталізувати структуру та архітектуру програмного забезпечення (Backend/Frontend);
- 4) Провести розрахунок вартості (CAPEX) та ресурсного забезпечення проєкту, а також сформуванню організаційну структуру команди;
- 5) Здійснити економіко-математичне обґрунтування (NPV/CBA) доцільності проєкту та довести його ефективність порівняно з існуючими підходами;
- 6) Провести валідацію практичних результатів шляхом аналізу адекватності розробленої моделі та оцінки відповідності реалізованого функціоналу заявленим вимогам.

РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

2.1. Розробка концептуальної моделі інформаційної системи

Концептуальна модель інформаційної системи відображає загальну структуру та логіку функціонування веб-платформи для управління оперативно-розшуковими даними. Вона описує основні компоненти системи, їхню взаємодію, а також потоки даних між користувачами, системою та зовнішніми джерелами.

Таблиця 2.1

Основні компонент системи

Компоненти системи	Призначення
Модуль аутентифікації та авторизації	<ul style="list-style-type: none">• Відповідає за безпечний вхід користувачів (оперативників, аналітиків, адміністраторів) до системи.• Використовує JWT для генерації токенів і Role-Based Access Control (RBAC) для розмежування доступу.• Забезпечує захист від несанкціонованого доступу шляхом перевірки ролей і блокування підозрілих активностей.
Модуль пошуку та фільтрації	<ul style="list-style-type: none">• Дозволяє користувачам виконувати запити до бази даних за різними параметрами (ПІБ, дата народження, номер авто тощо).• Забезпечує швидкий доступ до даних із затримкою не більше 2 секунд, як зазначено у вимогах.
Модуль аналітики	<ul style="list-style-type: none">• Надає інструменти для аналізу зв'язків між особами, транспортними засобами та подіями.

Модуль управління даними	<ul style="list-style-type: none">• Дозволяє додавати, редагувати та видаляти записи про осіб, транспортні засоби, адреси тощо.• Забезпечує валідацію даних і перевірку на дублікати перед збереженням у базу.
Модуль безпеки та аудиту	<ul style="list-style-type: none">• Логує всі дії користувачів (пошуки, зміни даних) для подальшого аналізу• Використовує шифрування для захисту конфіденційних даних і автоматично блокує підозрілі дії.

Актори системи:

- Оперативник: Виконує пошук осіб і транспортних засобів, додає та редагує дані.
- Модератор: Використовує модуль управління даними для додавання або оновлення даних.
- Адміністратор: Керує користувачами, призначає ролі, переглядає історію дій.

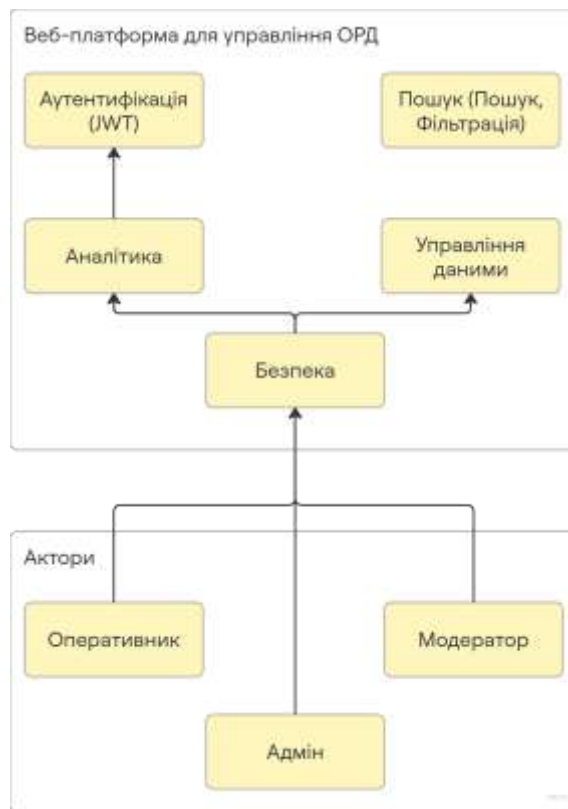


Рис. 2.1 Інтерфейс входу в систему

2.2. Визначення функціональних та нефункціональних вимог до продукту

Функціональні та нефункціональні вимоги визначають ключові аспекти роботи системи, забезпечуючи її надійність, безпеку та відповідність потребам правоохоронних органів.

Функціональні вимоги:

1) управління користувачами

- реєстрація та автентифікація користувачів із різними рівнями доступу (оперативник, аналітик, адміністратор);
- використання ролей для розмежування функціональних можливостей у системі (Role-Based Access Control);
- двофакторна аутентифікація для підвищення безпеки облікових записів;

- автоматичне блокування облікового запису при виявленні несанкціонованого доступу.

2) пошук та фільтрація інформації

- швидкий пошук осіб, транспортних засобів, місць проживання;
- фільтрація даних за ключовими параметрами (ПІБ, дата народження, номер авто, місто);
- можливість комбінованих запитів для точного визначення зв'язків між об'єктами;
- відображення історії запитів для подальшого аналізу.

3) аналітичний модуль

- візуалізація зв'язків між об'єктами (графові схеми відносин між особами, подіями, транспортом);
- аналіз взаємозв'язків між особами та подіями за допомогою ШІ-алгоритмів;
- формування звітів за заданими критеріями (динаміка пошуку, частота згадувань, географічні точки);
- інтеграція аналітичних даних із зовнішніми інформаційними ресурсами.

4) безпека та аудит

- логування всіх дій користувачів у системі з можливістю перегляду історії змін;
- шифрування конфіденційних даних за допомогою сучасних алгоритмів;
- контроль доступу відповідно до рівня допуску кожного користувача;
- автоматичне блокування спроб отримати несанкціонований доступ (детекція підозрілих активностей);

- система сповіщень про аномальні запити та дії користувачів.
- 5) журнал подій та контроль змін
- відстеження всіх змін у записах: хто, коли та які зміни вносив;
 - збереження резервних копій змінених записів із можливістю відновлення попередньої версії;
 - виявлення та блокування можливих маніпуляцій із критичною інформацією.
- б) інтеграція із зовнішніми державними реєстрами
- підключення до Єдиного реєстру досудових розслідувань;
 - обмін даними з митними та прикордонними службами;
 - інтеграція з базами Інтерполу та міжнародними розшуковими списками.

Нефункціональні вимоги:

1) продуктивність

- час відповіді на запит у системі не повинен перевищувати 2 секунд для пошукових операцій;
- обробка запитів на отримання повної інформації про особу повинна виконуватися менше ніж за 5 секунд;
- підтримка одночасної роботи не менше 1000 активних користувачів.

2) масштабованість

- система має бути розширюваною для додавання нових модулів без зміни основного коду;
- підтримка кластеризації серверів для роботи з великими обсягами інформації;

3) безпека

- двофакторна автентифікація для підвищення рівня безпеки облікових записів;
- захист від атак SQL-ін'єкцій, XSS, CSRF;
- автоматичне видалення тимчасових даних із серверів для зниження ризиків витоку інформації.

4) стабільність та надійність

- веб-платформа повинна працювати безперервно із часом доступності не менше 99.5%;
- усі оновлення повинні відбуватися без переривання роботи системи;
- автоматичне резервне копіювання бази даних кожні 24 години.

5) зручність використання (UX/UI)

- мінімальний час навчання користувачів для роботи в системі – до 2 годин;
- гнучке налаштування інтерфейсу відповідно до ролі користувача;
- підтримка роботи як із ПК, так і з мобільних пристроїв (адаптивний дизайн).

б) локалізація

- підтримка декількох мов (українська, англійська, можливість додавання інших);
- форматування даних відповідно до міжнародних стандартів (дата, валюта, адреси).

2.3. Формування Use Case діаграми для функціональних вимог

Веб-платформа оперативно-розшукової діяльності розрахована на використання трьома основними категоріями користувачів: оперативниками, аналітиками та адміністраторами. Кожен із них виконує специфічні завдання відповідно до своєї ролі та рівня доступу.

Опис основних сценаріїв використання

1) оперативник виконує пошук особи або транспортного засобу

- користувач вводить у форму пошуку один або декілька параметрів (ім'я, дата народження, номер авто, місце проживання, тип транспортного засобу);
- система виконує запит до бази даних та фільтрує результати за обраними критеріями;
- оперативник отримує список знайдених осіб або транспортних засобів із додатковою інформацією (фото, попередні місця перебування, ймовірні зв'язки) ;
- за необхідності користувач може переглянути історію взаємодій об'єкта (останній контакт, реєстраційні дані).

2) додавання нових даних про особу

- оперативник заповнює форму реєстрації нової особи в базі даних, додаючи такі відомості як ПІБ, дата народження, громадянство, фото (завантажене або отримане з бази), адреса проживання, відомі місця перебування, транспортні засоби, якими користується особа. Після натискання кнопки "Додати" система перевіряє унікальність даних;
- якщо особа вже є в базі, система повідомляє про дублювання та пропонує оновити існуючий запис;
- успішно додані записи відразу стають доступними для пошуку та аналізу.

3) редагування та оновлення даних

- якщо оперативник знаходить неточності або отримує оновлену інформацію, він може редагувати існуючий запис;
- перед внесенням змін система виконує логування всіх змін, фіксуючи, хто і коли оновлював інформацію;

- усі оновлення зберігаються у журналі подій, що дозволяє переглядати історію змін.

4) аналіз зв'язків між об'єктами

- аналітик використовує спеціальний інструмент для побудови графа зв'язків між особами, подіями, транспортними засобами;
- виявляються ключові контакти та підозрілі патерни (наприклад, однакові місця проживання, використання одного автомобіля кількома особами);
- дані можна фільтрувати за періодом (наприклад, тільки зв'язки за останні 6 місяців);
- аналітик може експортувати результати в аналітичний звіт.

5) формування та експорт аналітичних звітів

- аналітик генерує звіт по конкретній особі або групі осіб, використовуючи фільтри за типом подій, часом, місцем;
- система формує PDF- або Excel-звіт із графічними та текстовими даними;
- аналітичні звіти можуть містити рекомендації щодо розслідувань, згенеровані на основі великих масивів даних.

6) контроль доступу до системи (адміністратор)

- адміністратор додає нових користувачів, призначає їм рівні доступу (оперативник, аналітик, керівник, супер-адмін);
- усі дії користувачів логуються в систему з можливістю перегляду історії (хто коли виконав пошук, змінив дані тощо);
- при виявленні підозрілих дій (наприклад, масовий запит до бази) адміністратор може блокувати користувача або тимчасово обмежити його доступ;
- система автоматично перевіряє активність акаунтів, і якщо користувач не входив у систему понад 90 днів, його доступ блокується.

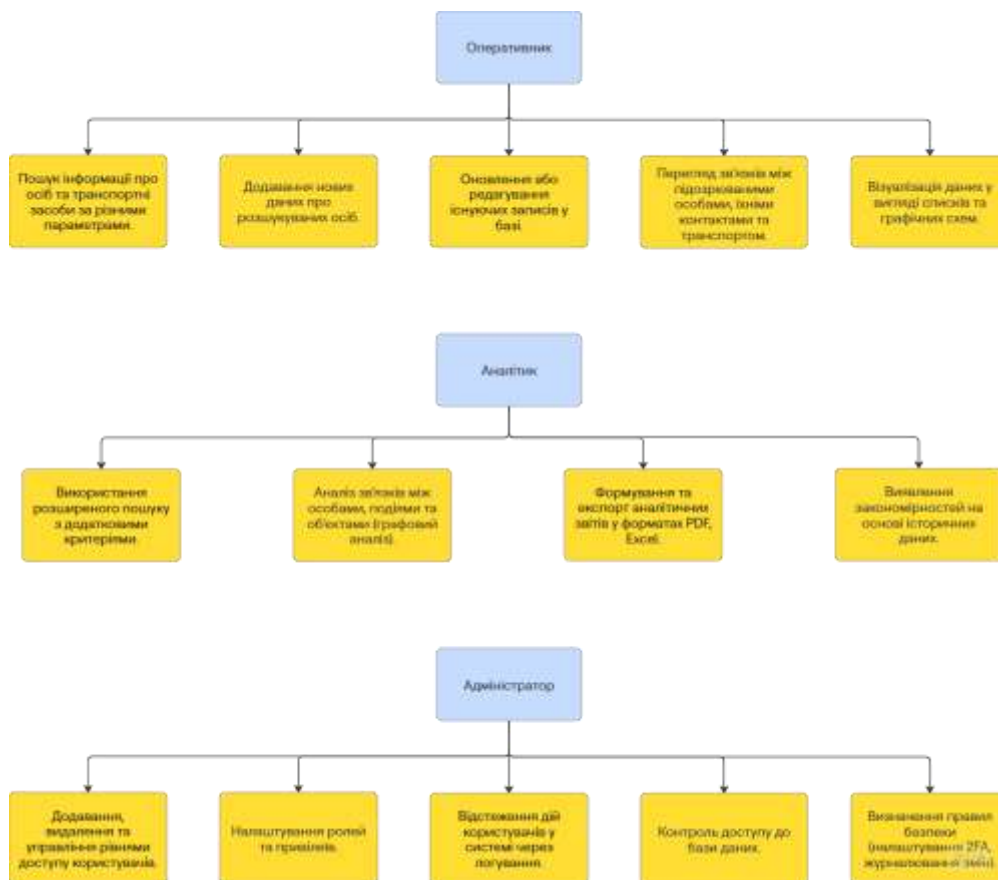


Рис. 2.2 Use Case діаграми для функціональних вимог

2.4. Розробка концептуальної та логічної моделей бази даних проєкту

Побудова концептуальної та логічної моделі бази даних є ключовими етапами в процесі проєктування інформаційних систем. Вони забезпечують структуроване та ефективне зберігання, обробку та доступ до даних, що є критично важливим для оперативно-розшукової діяльності. Концептуальна модель відображає загальну структуру даних предметної області без прив'язки до конкретної системи управління базами даних. Вона фокусується на сутностях, їхніх атрибутах та зв'язках між ними. Для побудови концептуальної моделі часто використовують ER-діаграми, які наочно демонструють взаємозв'язки між об'єктами системи.

2.4.1. Концептуальна модель бази даних

Концептуальна модель бази даних відображає основні сутності та їхні взаємозв'язки. Вона є абстрактним представленням структури бази без деталізації конкретних атрибутів або типів даних.

Для розробки веб-платформи було визначено такі основні сутності:

- користувачі (users) – громадяни;
- акаунти співробітників (staffaccounts) – облікові записи працівників системи, що містять їхні логіни, паролі та ролі в системі;
- фото (photos) – фотографії осіб, транспортних засобів та інших об'єктів, пов'язаних із розслідуванням;
- телефони (phones) – телефонні номери осіб;
- автомобілі (cars) – дані про транспортні засоби, що можуть бути причетні до кримінальних справ;
- кримінальні справи (criminalrecords) – записи про правопорушення, вироки суду та статуси справ;
- місця позбавлення волі (prisons) – інформація про тюрми та місця тримання під вартою;
- адреси користувачів (useraddresses) – поточні та попередні місця проживання підозрюваних або фігурантів справ;
- геолокація (addresses) – загальна інформація про місця, які можуть бути пов'язані з розслідуваннями.

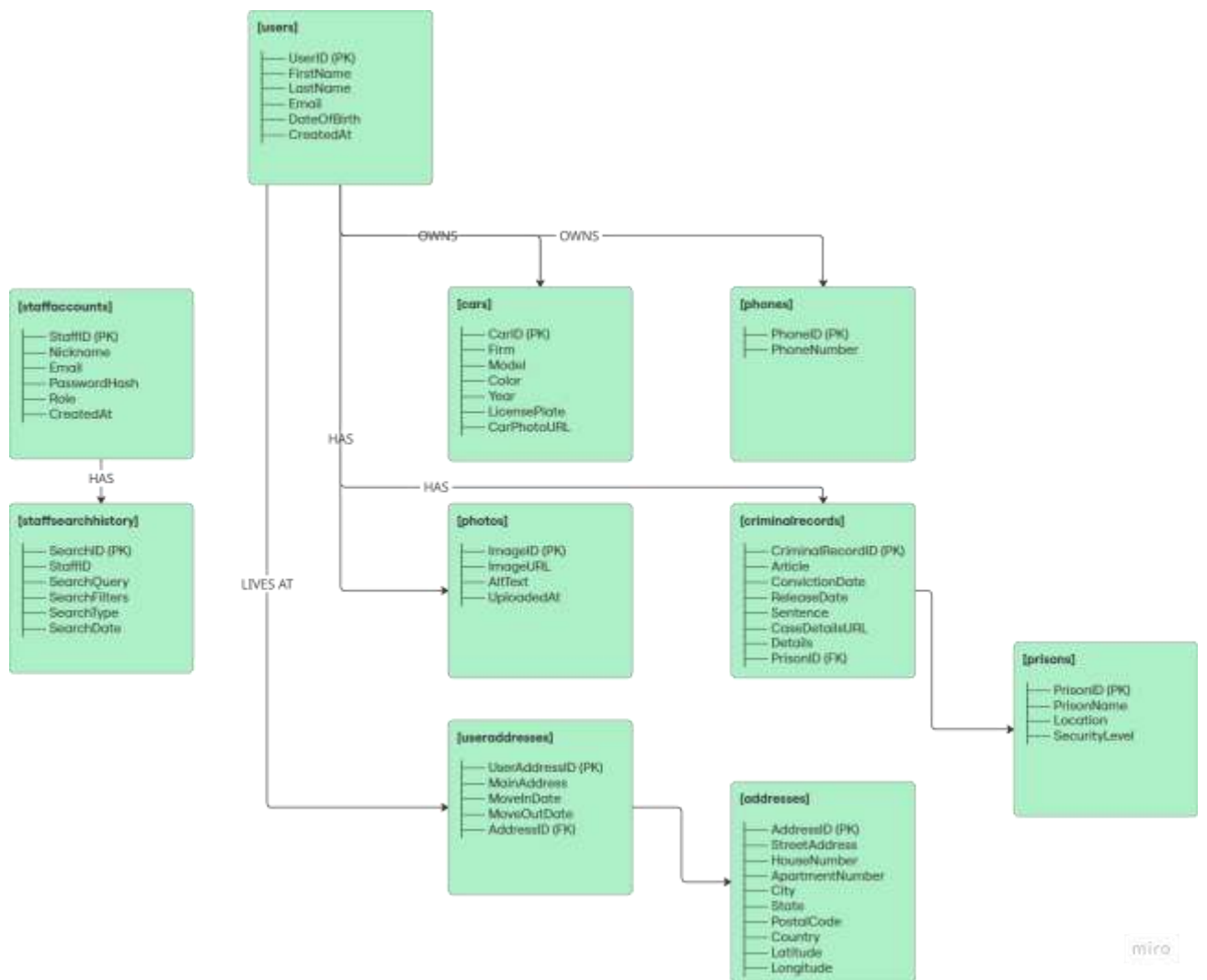


Рис. 2.3. Концептуальна модель бази даних

2.4.2. Логічна модель бази даних

Логічна модель бази даних деталізує структуру таблиць, визначає їхні атрибути, первинні та зовнішні ключі. Вона будується на основі концептуальної моделі та адаптується для зберігання інформації у реляційній базі даних.

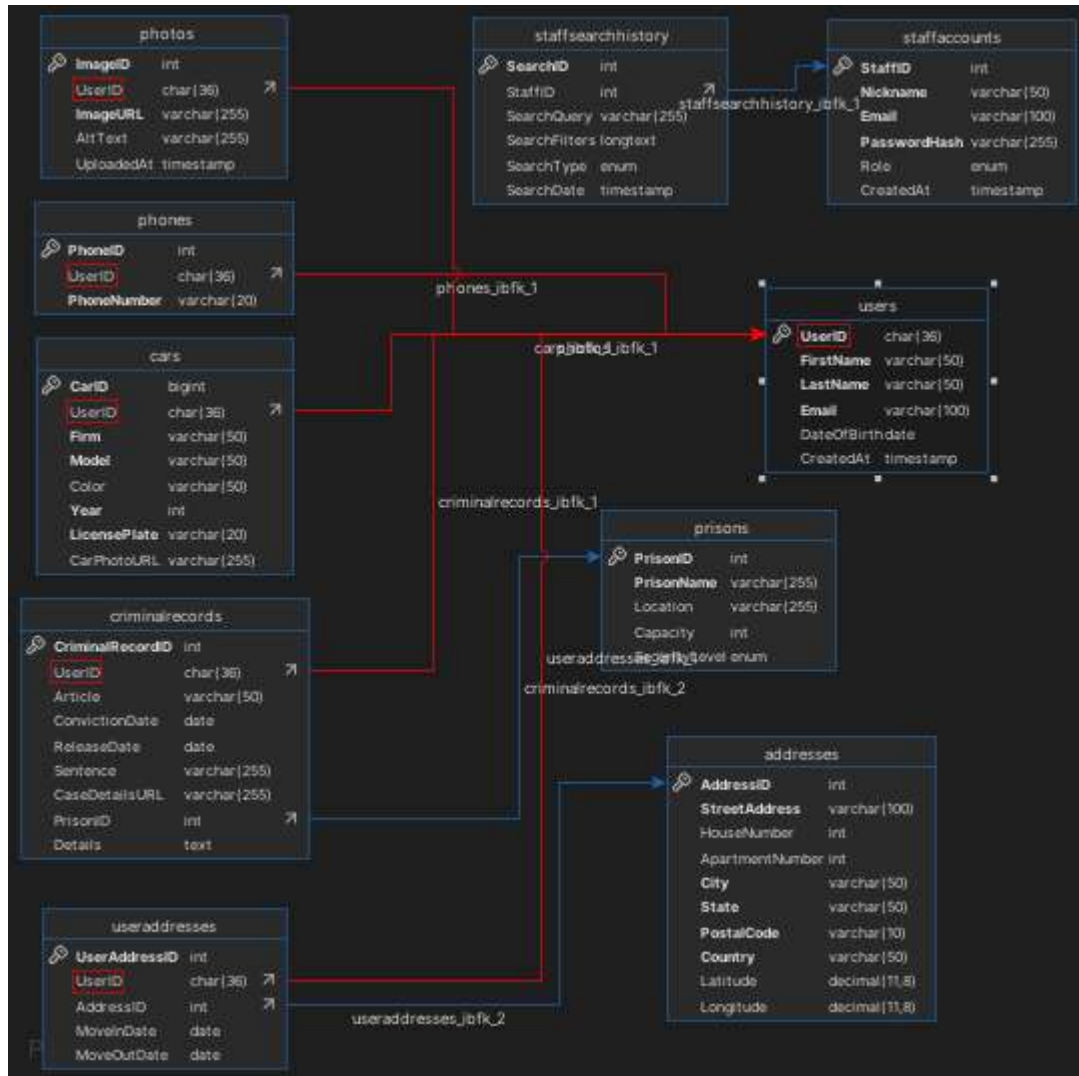


Рис. 2.4. Детальна логічна модель бази даних

2.5. Модель архітектури системи та технологічного стеку

Розроблена веб-платформа реалізована на основі трирівневої архітектури «клієнт-сервер», що забезпечує високу масштабованість, продуктивність та необхідний рівень безпеки й розмежування відповідальності між рівнями. Така

модель ідеально підходить для проекту в сфері ОРД, оскільки вона відокремлює логіку управління доступом та конфіденційними даними (Backend) від рівня представлення (Frontend). Ця архітектура відповідає сучасним підходам до розробки ІТ-систем у державному секторі. Ключовим елементом архітектури є серверний рівень, де реалізується вся бізнес-логіка, управління доступом (RBAC), обробка запитів та взаємодія з базами даних. Для цього було обрано технологічний стек, що гарантує надійність та безпеку:

- Frontend (Клієнтський рівень): React.js з Redux Toolkit та Material UI. Вибір React забезпечує високу продуктивність інтерфейсу, а Material UI дозволяє швидко створювати адаптивний дизайн, що спрощує роботу користувачів (Оперативників, Аналітиків) на різних пристроях.
- Backend (Серверний рівень): ASP.NET Core 8.0 (C#). Ця технологія відома своєю високою продуктивністю та потужними вбудованими механізмами безпеки, що є критичним для розробки REST API та логіки управління доступом (JWT, RBAC) у державному секторі.
- Рівень даних: MySQL. Ця реляційна база даних є надійною, масштабованою та підтримує транзакції, що необхідно для гарантування цілісності оперативних даних та журналу аудиту.

Взаємодія між рівнями відбувається через захищений протокол HTTPS та механізми JWT (JSON Web Token), які забезпечують аутентифікацію та авторизацію користувачів при кожному запиті. Моделювання архітектурної схеми зображено на рисунку 2.5.

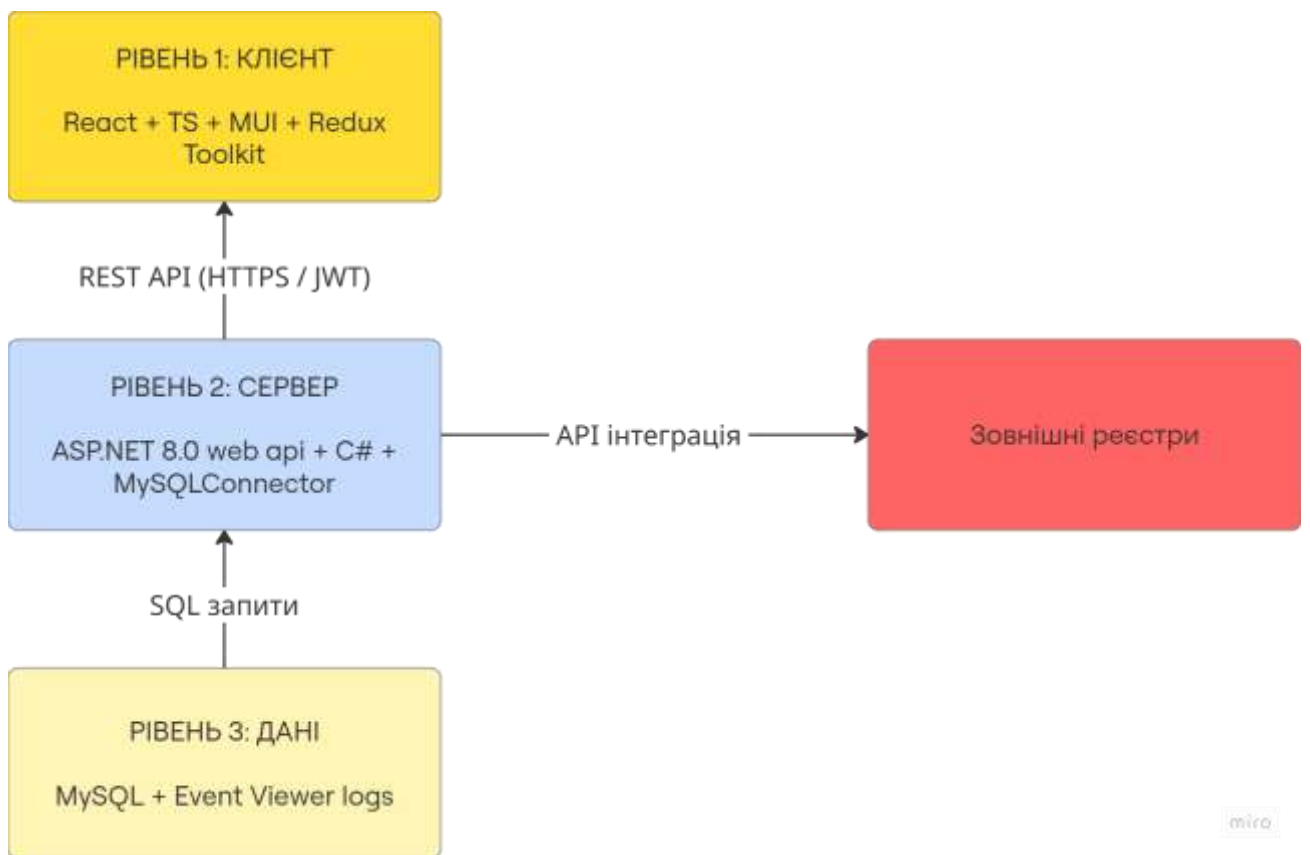


Рис. 2.5 Модель трирівневої архітектури веб-платформи для ОРД

2.6. Модель управління доступом на основі ролей (RBAC)

Впровадження ефективної моделі управління доступом є критично важливим для будь-якої інформаційної системи, що оперує конфіденційними даними (як-от дані ОРД). Згідно з вимогами Закону України «Про захист персональних даних» та міжнародних стандартів ISO/IEC 27001, система повинна забезпечувати принцип "мінімальних привілеїв" та чітке розмежування доступу. Для досягнення цієї мети було обрано Модель управління доступом на основі ролей (Role-Based Access Control, RBAC).

Модель RBAC є передовою в корпоративному та державному секторі. Її перевага полягає в тому, що права доступу (Permissions) призначаються не індивідуальним користувачам, а ролям (Roles), які, своєю чергою, присвоюються користувачам (Users). Це значно спрощує адміністрування системи, особливо у великих

організаціях з високою плинністю кадрів. Модель RBAC зображено на рисунку 2.6.

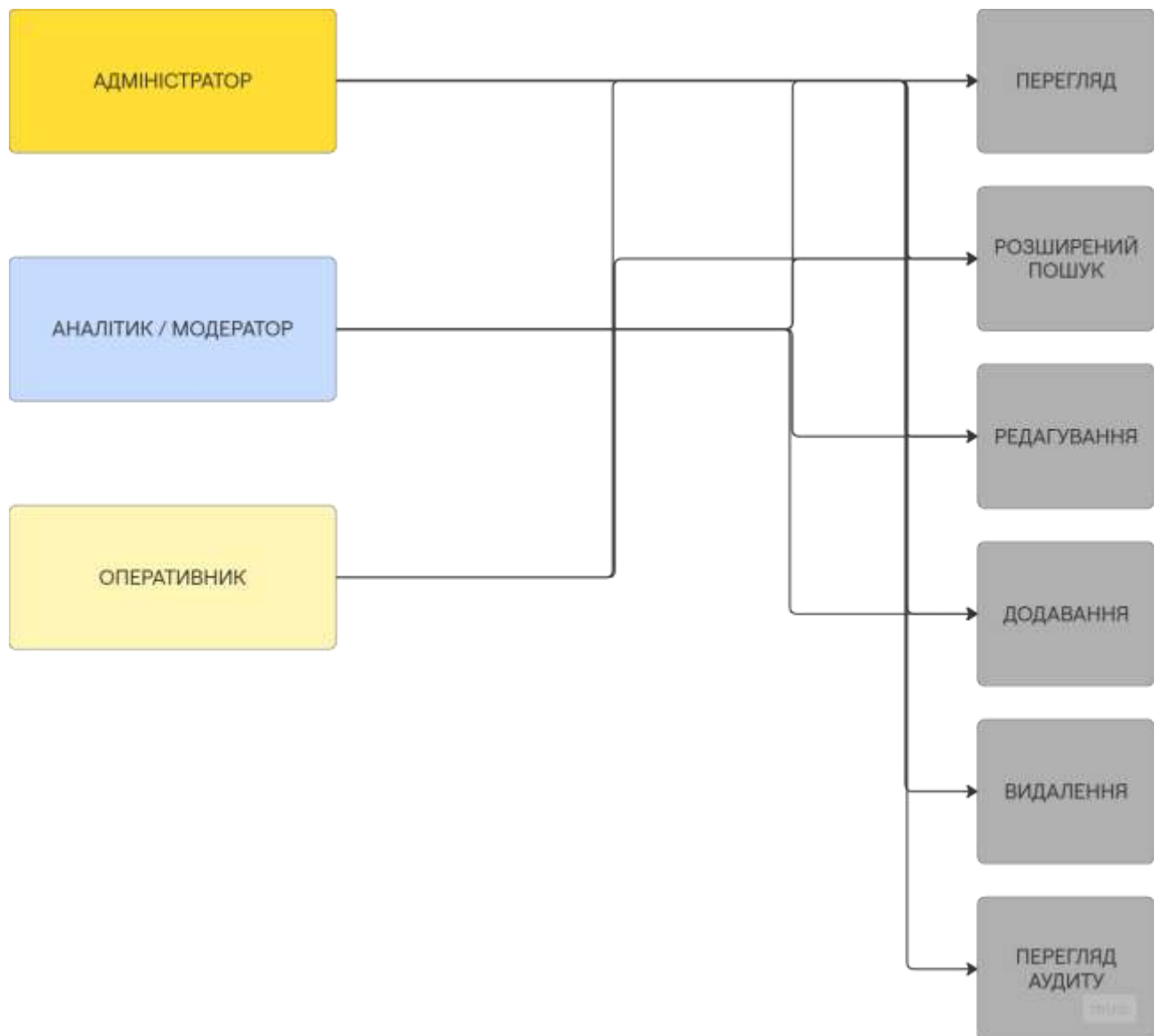


Рис. 2.6. Модель RBAC для веб-платформи для ОРД

2.7. Модель управління ризиками проєкту

Управління ризиками (Risk Management) є ключовим елементом проєктного управління (згідно з принципами РМВОК), особливо в контексті розробки ІТ-систем для державного сектору з високими вимогами до безпеки.

Оскільки проєкт має значну залежність від зовнішніх (політичних, правових) та внутрішніх (технічних, ресурсних) факторів, необхідно формалізувати процес виявлення, оцінки та реагування на потенційні загрози. Метою управління ризиками є мінімізація ймовірності та впливу негативних подій на терміни, бюджет та якість кінцевого продукту. Процес управління ризиками в даному проєкті відповідає ітераційному підходу і складається з трьох ключових етапів: ідентифікація та якісна оцінка, розробка карти реагування та моніторинг ризиків.

Якісна оцінка ризиків проводиться шляхом аналізу ідентифікованих загроз (із SWOT-аналізу, Розділ 1) та визначення їхньої ймовірності (P) та впливу (I). Для візуалізації та пріоритезації ризиків використовується Матриця Ризиків (Risk Matrix). У даному проєкті ризики оцінюються за шкалою від 1 (Дуже низький) до 5 (Катастрофічний/Дуже високий). На основі аналізу були виділені три категорії ризиків (Технічні, Безпечові, Управлінські), а їхня пріоритезація представлена на Рисунку 2.7.



Рис. 2.7. Модель управління ризиками проєкту для веб-платформи для ОРД

РОЗДІЛ 3. УПРАВЛІНСЬКЕ ПЛАНУВАННЯ ТА ДЕТАЛІЗАЦІЯ ПРОЄКТНОГО РІШЕННЯ

3.1. Структура декомпозиції робіт (WBS) та ресурсне забезпечення

Декомпозиція робіт WBS є фундаментальним етапом у плануванні проєкту розробки веб-платформи для оперативно-розшукової діяльності. Вона забезпечує розподіл загальної мети проєкту на менші, структуровані завдання та підзавдання, що полегшує їхнє планування, виконання та контроль. WBS відображає ієрархію робіт, включаючи основні етапи, їхні компоненти та деталізовані дії, що є критично важливим для ефективного розподілу ресурсів і відстеження прогресу. Для створення WBS часто використовуються візуальні інструменти, такі як Miro, які дозволяють зручно будувати деревоподібну структуру та адаптувати її до потреб проєкту.

Відповідно до визначених на попередніх етапах вимог, були спроектовані та реалізовані архітектура системи, основні алгоритми взаємодії користувача з даними, інтерфейси для зручної роботи з інформацією, а також проведено тестування окремих функціональних модулів. Головною метою розробки програмного забезпечення є створення надійної, масштабованої та безпечної платформи для управління чутливими даними, що дозволяє правоохоронним органам ефективно здійснювати пошук, обробку та аналіз оперативної інформації.

1) WBS проекту по процесам

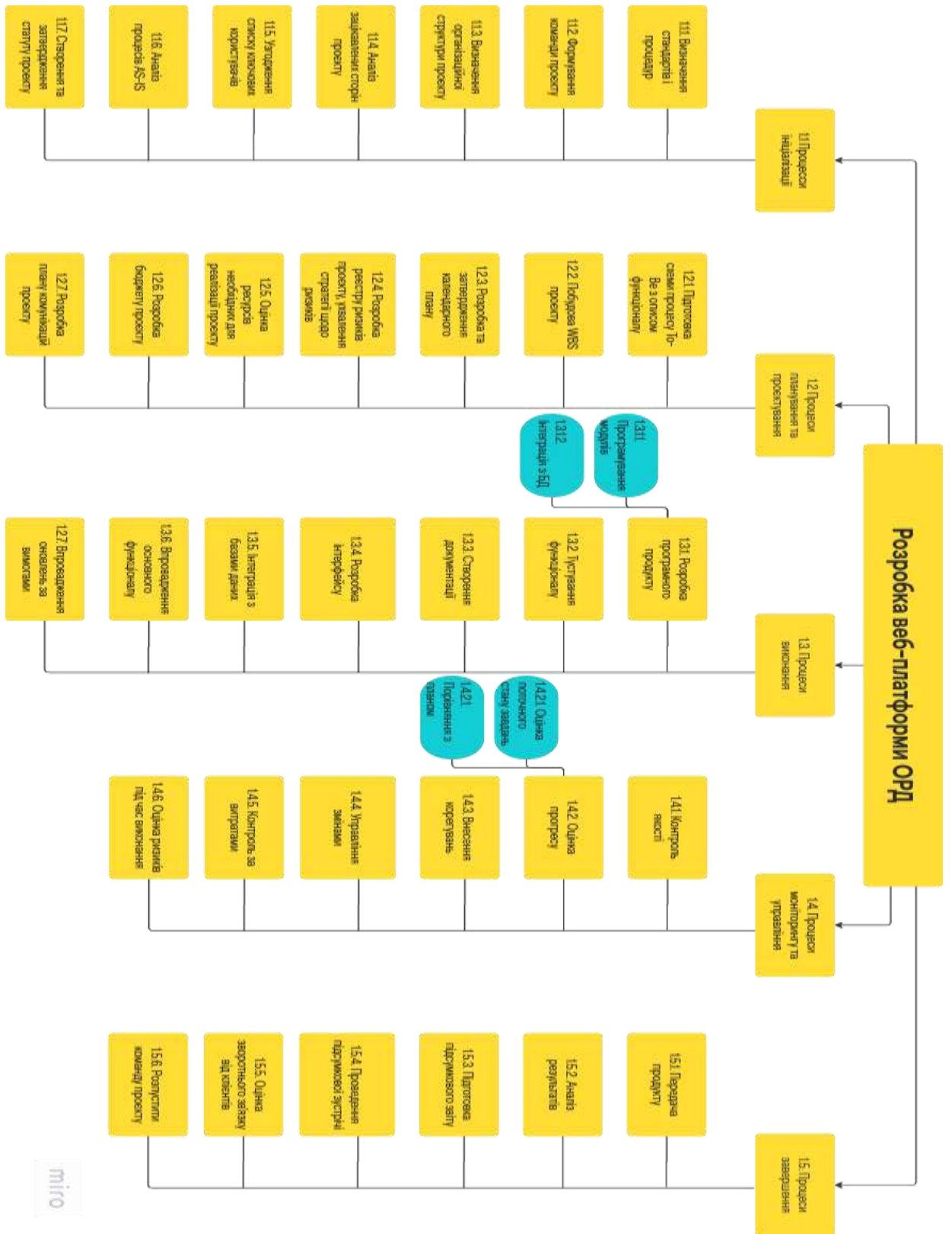


Рис. 3.1. WBS проекту по процесам

3) WBS проекту по продуктам

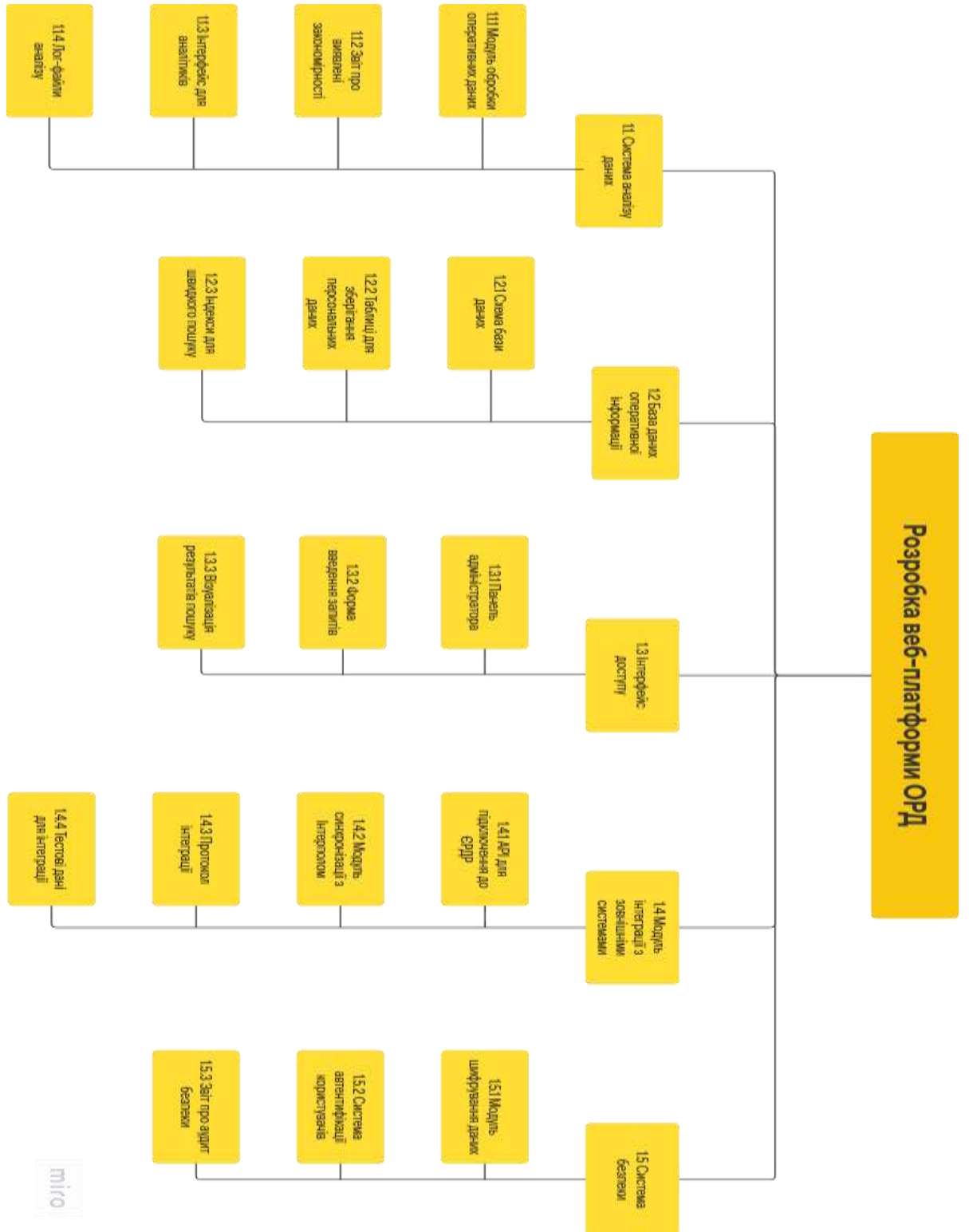


Рис. 3.3. WBS проекту по продуктам

У даному розділі здійснюється опис процесу розробки програмного забезпечення, що реалізує веб-платформу для підтримки оперативно-розшукової діяльності. Відповідно до визначених на попередніх етапах вимог, були спроектовані та реалізовані архітектура системи, основні алгоритми взаємодії користувача з даними, інтерфейси для зручної роботи з інформацією, а також проведено тестування окремих функціональних модулів. Головною метою розробки програмного забезпечення є створення надійної, масштабованої та безпечної платформи для управління чутливими даними, що дозволяє правоохоронним органам ефективно здійснювати пошук, обробку та аналіз оперативної інформації.

3.1.1. Матриця відповідальності RACI та ролі команди

Для забезпечення ефективного розподілу повноважень, мінімізації управлінських ризиків та уникнення конфліктів у команді проєкту використовується Матриця відповідальності RACI (Responsible, Accountable, Consulted, Informed). Ця модель дозволяє чітко визначити ступінь участі кожного члена команди у виконанні ключових завдань (Work Packages), що впливають зі структури WBS. Це є критичним для проєктів, де безпека та підзвітність є пріоритетом. Враховуючи гібридний підхід до управління (Agile/Waterfall), було визначено наступні ключові ролі:

- Менеджер Проєкту (МП);
- Тех-Лід (ТЛ);
- Аналітик (АН);
- Backend-розробник (BE);
- QA-інженер (QA);
- Адміністратор безпеки (АБ).

Матриця відповідальності RACI зображена в таблиці 3.1.

Таблиця 3.1

Матриця відповідальності RACI для ключових фаз проєкту

Фаза/Завдання	МП (Project Manager)	ТЛ (Tech Lead)	АН (Analyst)	BE (Backend Dev)	QA (Tester)	АБ (Security Admin)
1. Аналіз вимог та WBS	A (Затверджує)	C (Консультує)	R (Виконує)	I (Інформує)	I	C
2. Розробка моделі RBAC	C	A (Затверджує)	I	R (Виконує)	C	C (Консультує)
3. Проєктування БД та ERD	C	A	I	R	I	I
4. Кошторис (CAPEX)	A (Затверджує)	R (Виконує)	C	I	I	I
5. Розробка Backend (API)	I	A	I	R	C	C
6. Налаштування безпеки (JWT)	C	R	I	A (Затверджує)	I	C
7. Тестування функціоналу	I	C	I	R	A (Затверджує)	C

Модель RACI забезпечує не лише чіткий розподіл завдань, але й функціонально відображає обрану гібридну стратегію управління проєктом.

Управлінське обґрунтування ролей:

- Розмежування Технічної та Управлінської Відповідальності, тобто Менеджер Проєкту (МП) несе кінцеву відповідальність (A) лише за ключові управлінські та фінансові фази (Аналіз, Кошторис), тоді як Технічний Лід (ТЛ) несе відповідальність за технічну архітектуру та реалізацію. Це мінімізує ризик вузького горла та дозволяє швидко ухвалювати рішення на технічному рівні;
- Підзвітність у Сфері Безпеки, на фазі Налаштування безпеки (JWT) кінцева відповідальність (A) лежить на Backend-розробнику (BE), який імплементує логіку, проте Технічний Лід (ТЛ) виступає в ролі R

(Виконавця) — він контролює процес, тоді як Адміністратор Безпеки (АБ) надає критичні консультації (С) щодо стандартів та політик, забезпечуючи відповідність ISO 27001. Це відображає складну ієрархію підзвітності у системах з високими вимогами до конфіденційності;

- **Контроль Якості:** На фазі Тестування функціоналу кінцева відповідальність (А) покладена на QA-інженера, що гарантує незалежну валідацію продукту перед прийняттям. Backend-розробник (ВЕ) у цьому випадку є R (Виконавцем), оскільки він має виправити виявлені помилки.

Таким чином, матриця RACI не просто розподіляє завдання, а є ключовим управлінським документом, що фіксує ієрархію ухвалення рішень, особливо щодо критичних для проєкту ризиків інтеграції та безпеки.

3.2. Календарне планування та критичний шлях

3.2.1. Діаграма Ганта та управління строками

Побудова діаграми Ганта є ключовим етапом у плануванні та управлінні проєктом розробки веб-платформи для оперативно-розшукової діяльності. Вона забезпечує візуалізацію часової послідовності завдань, їхньої тривалості та залежностей, що є критично важливим для ефективного розподілу ресурсів і контролю термінів виконання. Діаграма Ганта відображає загальний графік проєкту, включаючи основні етапи, їхній початок і завершення, а також дозволяє відстежувати прогрес роботи. Для її створення часто використовуються інструменти, такі як Microsoft Excel, які надають гнучкість у налаштуванні та простоту реалізації, що робить їх доступними для проєктного менеджменту на різних етапах розробки.

Загальний графік проєкту

Завдання	Початок	Тривалість (дні)	Завершення
Аналіз вимог	01.07.2025	30	30.07.2025
Проектування БД	31.07.2025	60	28.09.2025
Розробка дизайну	30.09.2025	59	27.11.2025
Розробка бекенду	29.11.2025	61	28.01.2026
Розробка Фронтенду	29.01.2026	17	14.02.2026
Тестування та виправлення помилок	15.02.2026	44	29.03.2026
Оптимізація безпеки	31.03.2026	32	01.05.2026
Впровадження проєкту	01.05.2026	30	30.05.2026

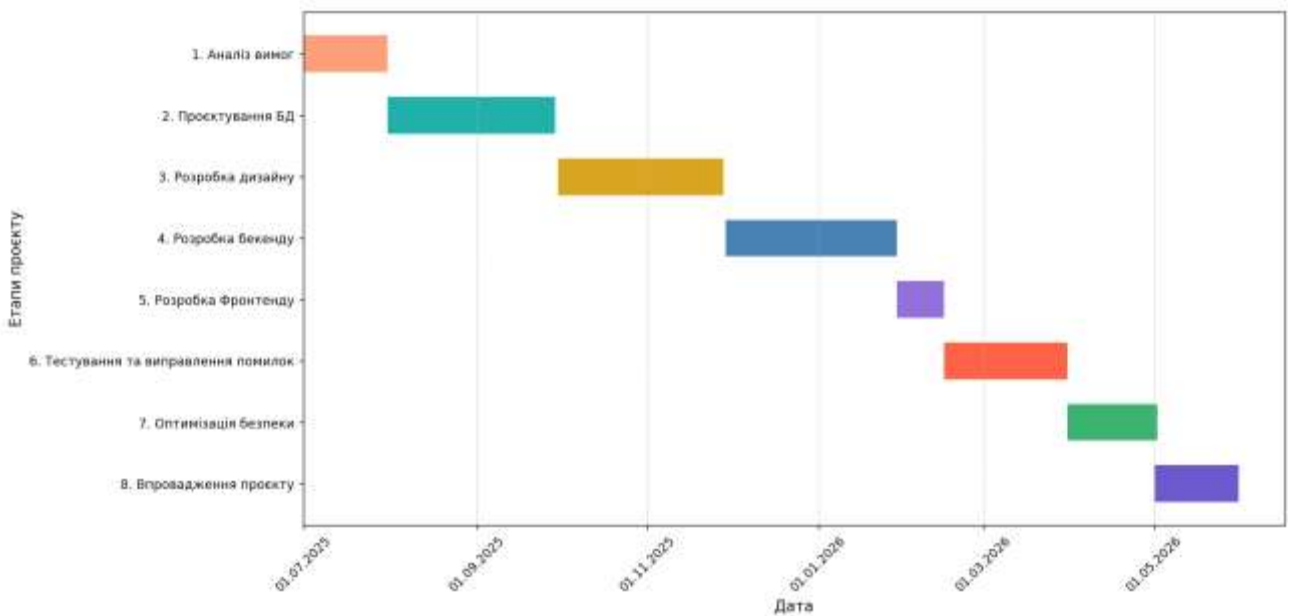


Рис. 3.4. Діаграма Ганта

3.2.2. Метод критичного шляху (CPM) та аналіз залежностей

Для ефективного управління термінами та ідентифікації завдань, які не мають часового резерву (що є критичним для проєкту з фіксованими строками),

застосовується метод критичного шляху (Critical Path Method, CPM). CPM дозволяє визначити найдовший ланцюг завдань, що безпосередньо впливає на загальну тривалість проєкту. Обґрунтування CPM демонструє, що менеджмент проєкту здійснюється не лише за датами, а й за логічними залежностями.

Таблиця 3.3

Аналіз залежностей та розрахунок Критичного Шляху

Завдання	Тривалість (дні)	Попередник (Залежність)	Ранній Початок (ES)	Раннє Завершення (EF)	Пізній Початок (LS)	Пізнє Завершення (LF)	Резерв (Slack)	Критичний Шлях
Аналіз вимог	30	–	1	30	1	30	0	ТАК
Проектування БД	60	1	31	90	31	90	0	ТАК
Розробка дизайну	59	2	91	149	91	149	0	ТАК
Розробка бекенду	61	2	91	151	91	151	0	ТАК
Розробка Фронтенду	17	4 (Паралельно 3)	152	168	152	168	0	ТАК
Тестування та виправлення	44	5	169	212	169	212	0	ТАК
Оптимізація безпеки	32	6	213	244	213	244	0	ТАК
Впровадження проєкту	30	7	245	274	245	274	0	ТАК

Обґрунтування Критичного Шляху:

- Попередник: встановлено, що ключові фази є послідовними: Проектування БД не може початися без Аналізу вимог (1), Розробка бекенду не може розпочатися без Проектування БД (2) тощо;
- Тривалість проєкту: Загальна тривалість проєкту становить 274 дні (з 01.07.2025 до 30.05.2026);

- Критичний Шлях: усі завдання в ланцюгу 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 мають нульовий резерв. Це означає, що будь-яка затримка у виконанні будь-якого з цих завдань призведе до затримки всього проєкту.

Отже, для забезпечення успішного завершення проєкту необхідно здійснювати посилений моніторинг завдань, що входять до Критичного Шляху, особливо на етапах Розробка бекенду (4) та Тестування (6).

3.3. Кошторис проєкту та оцінка капітальних витрат (CAPEX)

Фінансове планування проєкту створення веб-платформи для оперативно-розшукової діяльності є необхідною умовою для забезпечення його успішної реалізації та економічної обґрунтованості. Даний розділ присвячений формуванню детального кошторису та оцінці капітальних витрат (CAPEX), які складають первинні інвестиції, необхідні для запуску проєкту. CAPEX є ключовим інвестиційним ризиком, який ми повинні детально проаналізувати для успішного захисту роботи. У контексті ІТ-проєктів основна частина Капітальних витрат (CAPEX) припадає на Фонд Оплати Праці (ФОП) команди розробників, оскільки їхня праця створює кінцевий інтелектуальний продукт, який буде використовуватися протягом тривалого періоду. Наші розрахунки ФОП базуються на структурі команди, визначеній у Розділі 3.1, та загальній тривалості проєкту у 11 місяців. Для забезпечення коректності розрахунків та уніфікації оцінки, заробітна плата персоналу визначена у доларах США (USD) відповідно до середньоринкових ставок в українському ІТ-секторі станом на 2025 рік (прийнятий умовний курс 1 USD = 40 UAH).

Таким чином, Фонд Оплати Праці (ФОП), який є найбільшим компонентом CAPEX, становить 7 700 000 UAH. Ця сума включає оплату праці Менеджера Проєкту, Тех-Ліда, Аналітика, Backend-розробника, QA-інженера та Адміністратора безпеки протягом усього циклу розробки. Окрім безпосередньої оплати праці, капітальні витрати проєкту включають одноразові інвестиції в

технічне забезпечення, ліцензійне програмне забезпечення та формування резервного фонду, необхідного для управління непередбачуваними ризиками (як визначено в розділі 2.7). Детальний розрахунок усіх капітальних витрат (CAPEX) представлений у зведеній таблиці 3.4.

Таблиця 3.4

Зведена відомість капітальних витрат проєкту

№	Стаття витрат	Одиниця виміру	Кількість	Вартість за одиницю (UAH)	Загальна вартість (UAH)	Примітки
I.	Прямі витрати на персонал (ФОП)				7 700 000	Розраховано на основі місячних ставок.
II.	Амортизація обладнання				770 000	10% від ФОП, включає витрати на ПК та сервери розробки.
III.	Програмне забезпечення та ліцензії					
3.1	Ліцензії IDE (напр., JetBrains)	Ліцензія	5	15 000	75 000	На команду розробки.
3.2	Ліцензія на керування БД (напр., DBeaver Pro)	Ліцензія	3	10 000	30 000	Для ТЛ та BE-розробників.
3.3	Система контролю версій (GitLab/GitHub Enterprise)	Річна ліцензія	1	40 000	40 000	Для 6 користувачів.

3.4	Професійне тестове середовище	Ліцензія	1	85 000	85 000	Інструменти для автоматизованого тестування.
IV.	Накладні витрати (Офіс, Комунікації)				192 500	2.5% від ФОП.
V.	Резервний фонд (Непередбачувані ризики)				400 000	Близько 5% від ФОП (згідно з аналізом ризиків).
СУМАРНИЙ CAPEX	—	—	—	—	9 002 500	

Загальна сума Капітальних Витрат (CAPEX) на розробку веб-платформи становить: 9 002 500 UAH.

Таким чином, розроблений кошторис демонструє, що основний фінансовий ризик проєкту зосереджений у CAPEX, а саме — у Фонді Оплати Праці, який складає 85.5% від загальних капітальних витрат. Загальна інвестиційна вартість проєкту становить 9 002 500 UAH. Успішне управління цим бюджетом та дотримання термінів, визначених критичним шляхом, є ключовим для фінансової ефективності проєкту.

Хоча CAPEX відображає інвестиційну вартість проєкту, для повної економічної картини необхідно також окреслити очікувані Операційні Витрати (OPEX), які почнуться після введення системи в експлуатацію (після 30.05.2026):

- хостинг та інфраструктура: Щомісячна оплата за хмарні послуги (наприклад, AWS, Azure або українські ЦОД) для розміщення сервісів, бази даних та сховища файлів;

- технічна підтримка та обслуговування: Оплата роботи команди підтримки або укладення аутсорсингового контракту на обслуговування системи;
- оновлення ліцензій: Щорічне подовження ліцензій на програмне забезпечення (наприклад, ліцензії IDE, ПЗ для безпеки).

Отже, з цього можемо зробити висновок: успішне управління цим бюджетом та дотримання термінів, визначених критичним шляхом, є ключовим для фінансової ефективності проєкту.

3.4. Опис структури програмного забезпечення

Розроблене мною програмне забезпечення має архітектуру "клієнт-сервер", яка передбачає розподіл обов'язків між серверною частиною, що обробляє запити та працює з базою даних, і клієнтською частиною, що забезпечує користувачам інтерфейс для взаємодії із системою. Така модель дозволяє досягти високої масштабованості проєкту, забезпечити ефективну обробку великої кількості даних та гарантувати безпеку обміну інформацією між користувачем і сервером.

- Серверна частина розроблена із використанням технології ASP.NET Core 8.0, що дозволяє будувати продуктивні та безпечні веб-додатки. Вона включає в себе набір моделей для представлення даних з бази даних, набір контролерів, які відповідають за обробку HTTP-запитів клієнтів і репозиторіїв для безпосередньої взаємодії з базою даних. Для обробки автентифікації користувачів використовується механізм JWT (JSON Web Tokens), що дозволяє безпечно зберігати інформацію про сесію та здійснювати контроль доступу до захищених ресурсів залежно від ролі користувача.
- База даних реалізована на основі MySQL та включає набір таблиць для зберігання даних про користувачів, осіб, що перебувають у розшуку, транспортні засоби, місця проживання, а також інші супутні атрибути. Структура бази даних була нормалізована для зменшення надлишковості

даних та оптимізації швидкості обробки запитів. Усі таблиці пов'язані між собою за допомогою зовнішніх ключів, що забезпечує підтримку цілісності даних.

- Клієнтська частина платформи розроблена за допомогою бібліотеки React із застосуванням Redux Toolkit для централізованого управління станом застосунку. Інтерфейс побудований з використанням Material UI, що дозволяє створити сучасний, адаптивний і зручний для користувача дизайн. Клієнтська частина забезпечує роботу із сервером через REST API, обробляючи отримані відповіді та відображаючи інформацію у вигляді зручних для сприйняття форм і таблиць. Функціонал передбачає можливість автентифікації, перегляду списку осіб, пошуку за різними параметрами, додавання нової інформації та редагування існуючих записів.

Особлива увага в структурі системи приділяється забезпеченню безпеки. На всіх рівнях взаємодії впроваджено механізми контролю доступу, шифрування даних, а також логування дій користувачів для можливості подальшого аудиту. Такий підхід дозволяє мінімізувати ризики витоку інформації та гарантувати відповідність вимогам законодавства щодо захисту персональних даних. Таким чином, структура розробленого програмного забезпечення відповідає сучасним вимогам до систем оперативного обліку та обробки даних, забезпечуючи гнучкість у розширенні функціональності, високу продуктивність та надійний рівень безпеки.

3.5. Розробка алгоритмів та інтерфейсів програмного забезпечення

Розробка алгоритмів і інтерфейсів програмного забезпечення є одним із найважливіших етапів створення веб-платформи для оперативно-розшукової діяльності. Від якості реалізації цих компонентів залежить ефективність функціонування всієї системи, її зручність для користувача та надійність при роботі з великими обсягами даних. Одним із основних алгоритмів, реалізованих

у системі, є алгоритм автентифікації користувачів. Для забезпечення захищеного доступу до інформації була впроваджена технологія JWT (JSON Web Token). Після введення логіна та пароля дані перевіряються сервером, і у випадку успішної автентифікації генерується токен, який зберігається на стороні клієнта і надсилається разом із кожним запитом до сервера для підтвердження особи користувача. Якщо токен є недійсним або закінчився його термін дії, користувач автоматично перенаправляється на сторінку авторизації.

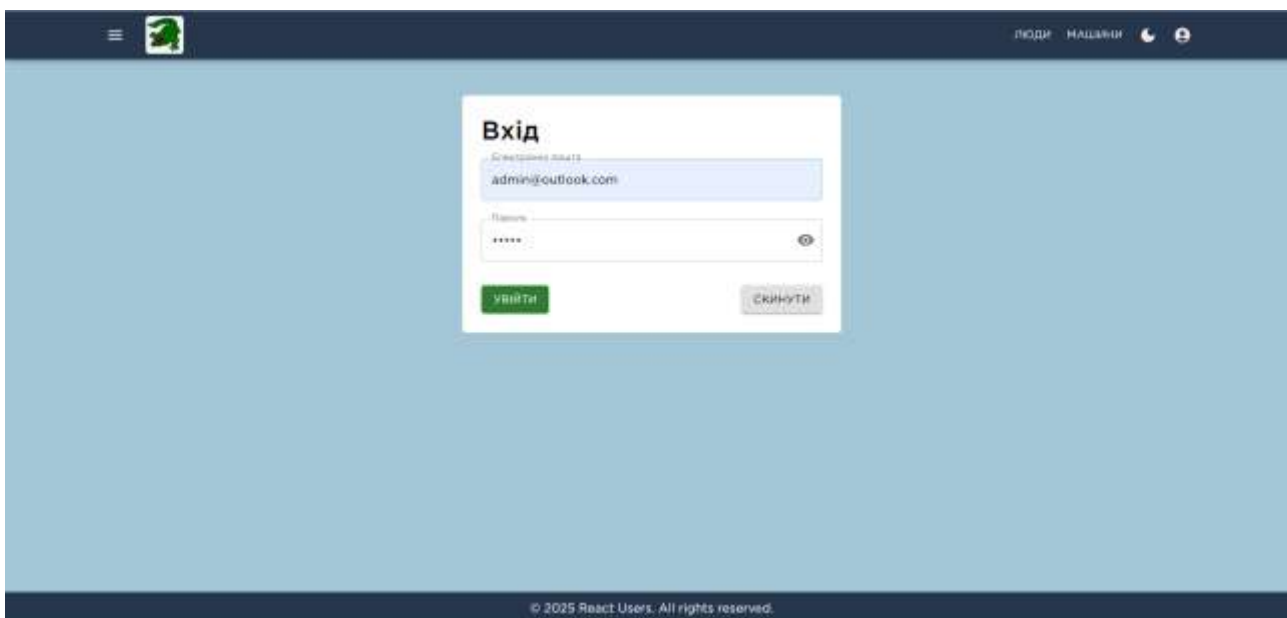


Рис. 3.5. Форма авторизації користувача у системі

Далі був реалізований алгоритм пошуку даних у базі осіб. Користувачеві надається форма, в якій можна задати ім'я користувача і додаткові параметри пошуку — вік, дата створення запису, пошук тільки повнолітніх осіб або осіб з завантаженими фото. Після заповнення полів формується SQL-запит до бази даних, оптимізований під конкретні критерії. У разі успішного пошуку результати відображаються у вигляді таблиці з короткою інформацією про знайдені об'єкти.

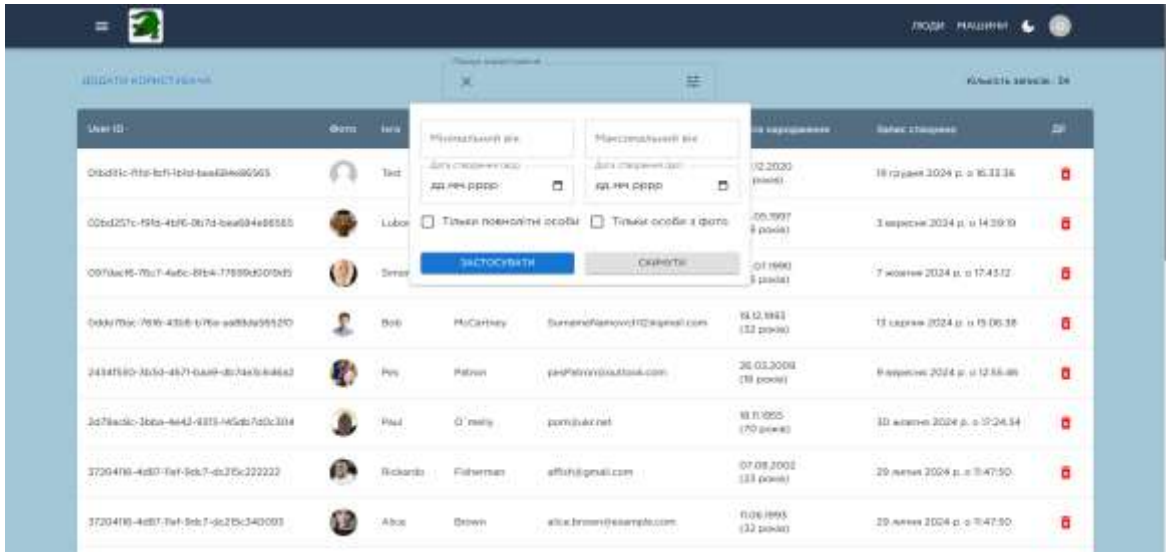


Рис. 3.6. Форма пошуку осіб

Після отримання результатів пошуку користувач має можливість переглянути детальну інформацію про знайдені об'єкти. Відкривається окрема сторінка профілю особи або транспортного засобу, де відображається повний набір даних: персональна інформація, рухоме майно, місце реєстрації, судимості.

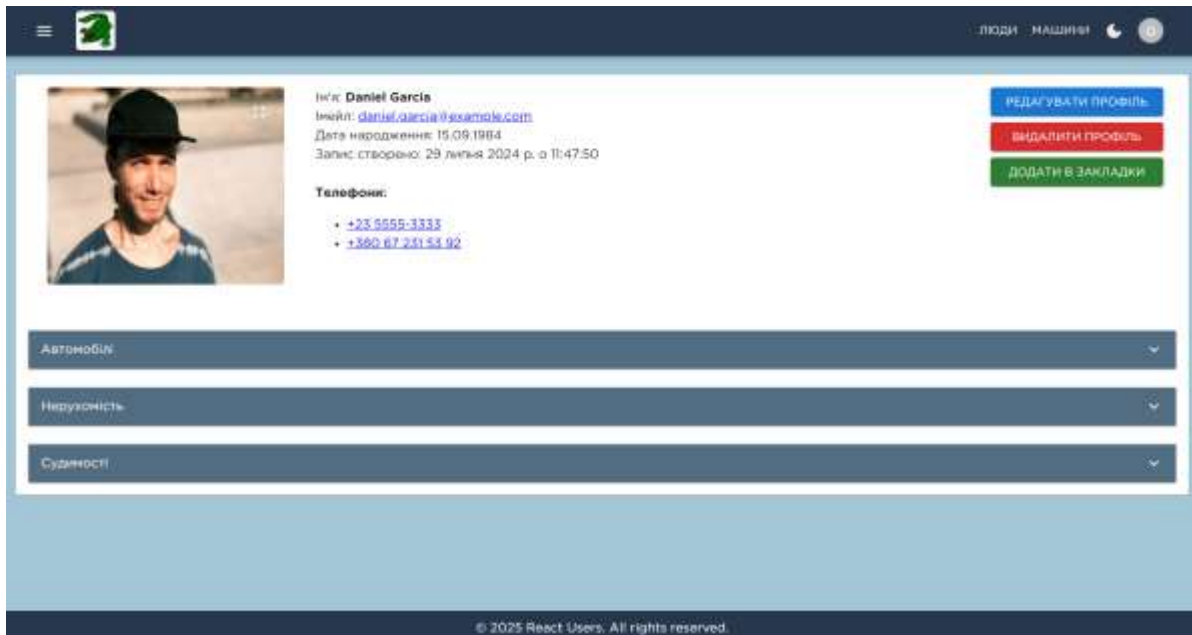


Рис. 3.7. Відображення детальної інформації про об'єкт пошуку

Одним із важливих алгоритмів є також процедура додавання нових записів до бази даних. Оперативний працівник або аналітик може через окрему форму заповнити дані про нову особу або транспортний засіб. Перед відправкою даних на сервер проводиться валідація обов'язкових полів на стороні клієнта, що дозволяє уникнути некоректного введення інформації. Після перевірки дані надсилаються серверу, де знову проходять валідацію, перевірку на дублікати, після чого записуються до бази даних.



Рис. 3.8 Форма додавання нового запису до бази даних

Для ведення аналітики передбачений окремий модуль в якому зберігається історія пошуку кожного співробітника. У цьому модулі кожен співробітник може подивитись історію свого пошуку, а співробітник з роллю “Admin” може дивитись історію всіх співробітників. Цей модуль зроблений для ведення звітності історії пошуку.

Таблиця	ID Запису	Примітки	Зачет	Фільтри	Дата пошуку
users	71	overland admin@outlook.com admin	Mike	Пошук без фільтрів	27 квітня 2025 р. о 15:53:44
users	70	overland admin@outlook.com admin	Bill	Пошук без фільтрів	15 січня 2025 р. о 16:45:03
users	69	overland admin@outlook.com admin	sa	Пошук без фільтрів	15 січня 2025 р. о 16:44:54
users	68	overland admin@outlook.com admin	Paul	Пошук без фільтрів	15 січня 2025 р. о 16:39:30

Рис. 3.9. Інтерфейс перегляду історії пошуку співробітників

Усі дії, які потребують доступу до конфіденційної інформації або змін у базі даних, супроводжуються відповідною верифікацією користувача через токен, що забезпечує високий рівень захисту даних та контролю дій в системі. Таким чином, розроблені алгоритми й створені інтерфейси дозволяють не лише ефективно працювати з оперативною інформацією, а й забезпечують інтуїтивно зрозумілий та безпечний доступ до всіх можливостей веб-платформи.

РОЗДІЛ 4. ВАЛІДАЦІЯ РЕЗУЛЬТАТІВ ТА ОЦІНКА ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

4.1. Економіко-математичне обґрунтування (NPV/СВА)

Економіко-математичне обґрунтування доцільності проекту є одним з фінальних етапом аналізу, який підтвердить, що інвестиції в розробку веб-платформи є фінансово виправданими в довгостроковій перспективі. Оскільки проект має соціальну спрямованість і реалізується для державни, його ефективність оцінюється не лише через прямий прибуток, а й через Аналіз Витрат і Вигод (Cost-Benefit Analysis, СВА) та розрахунок Чистої Приведеної Вартості (Net Present Value, NPV).

Методологія розрахунку побудована на порівнянні капітальних витрат (CAPEX), визначених раніше (9 002 500 UAH), з очікуваними економічними вигодами та зниженням операційних витрат протягом життєвого циклу проекту. Для оцінки соціальних та непрямих економічних вигод ми використовуємо методику грошової оцінки часу, зекономленого оперативними працівниками, та мінімізації фінансових втрат від ризиків.

Аналіз Вигод (Benefits)

Вигоди проекту поділяються на прямі та непрямі:

- 1) прямі річні вигоди, тобто економія робочого часу персоналу. На основі валідації (Розділ 4.2) приймається, що платформа скорочує час пошуку та аналізу на 40% для 20 оперативників. Це призводить до річної економії фонду оплати праці приблизно на 2 200 000 UAH.
- 2) непрямі річні вигоди, тобто мінімізація фінансових втрат, пов'язаних із критичними ризиками (S1, S2, Розділ 2.7), що оцінюється у 350 000 UAH на рік.

Загальний річний грошовий потік (Cash Flow, CF) від економії становить: \$2 200 000 UAH + 350 000 UAH = 2 550 000 UAH.

Розрахунок Чистої Приведеної Вартості (NPV)

Для спрощення моделі приймаємо, що річні операційні витрати (хостинг, ліцензії підтримки) становлять \$200 000 UAH. Річний чистий грошовий потік дорівнює: $2\,550\,000\text{ UAH} - 200\,000\text{ UAH} = 2\,350\,000\text{ UAH}$.

Рік (t)	Початкові Інвестиції (I ₀)	Річний Грошовий Потік (CF _t)	Коефіцієнт Дисконтування (1+r) ^t	Дисконтований Потік (DP _t)	Накопичений Потік (C-NPV)
0	-9 002 500	0	1.000	-9 002 500	-9 002 500
1	0	2 350 000	1.150	2 043 478	-6 959 022
2	0	2 350 000	1.323	1 776 503	-5 182 519
3	0	2 350 000	1.521	1 545 106	-3 637 413
4	0	2 350 000	1.749	1 343 686	-2 293 727
5	0	2 350 000	2.011	1 168 534	-1 125 193
6	0	2 350 000	2.313	1 015 977	-109 216
7	0	2 350 000	2.660	883 459	+774 243
8	0	2 350 000	3.059	768 251	+1 542 494

Згідно з розрахунками, оскільки Чиста Приведена Вартість (NPV) проєкту після 7-го року стає позитивною (NPV = +774 243 UAH) і продовжує зростати, проєкт є інвестиційно привабливим.

4.2. Валідація результатів та аналіз відповідності вимогам

Тестування програмного забезпечення є заключним етапом розробки веб-платформи для управління оперативно-розшуковими даними, оскільки воно забезпечує перевірку відповідності системи функціональним і нефункціональним вимогам, її надійність, безпеку та зручність використання. Оскільки платформа призначена для роботи з даними для обмеженого користування, тестування має охоплювати всі аспекти системи, включаючи бекенд, фронтенд, базу даних та їхню взаємодію через REST API.

- 1) аутентифікація (JWT):

- успішний вхід із правильними логіном і паролем, видача валідного токена;
- при некоректних даних відображається повідомлення про помилку;
- спроба доступу до захищених ресурсів без токена повертає помилку 401 (Unauthorized)

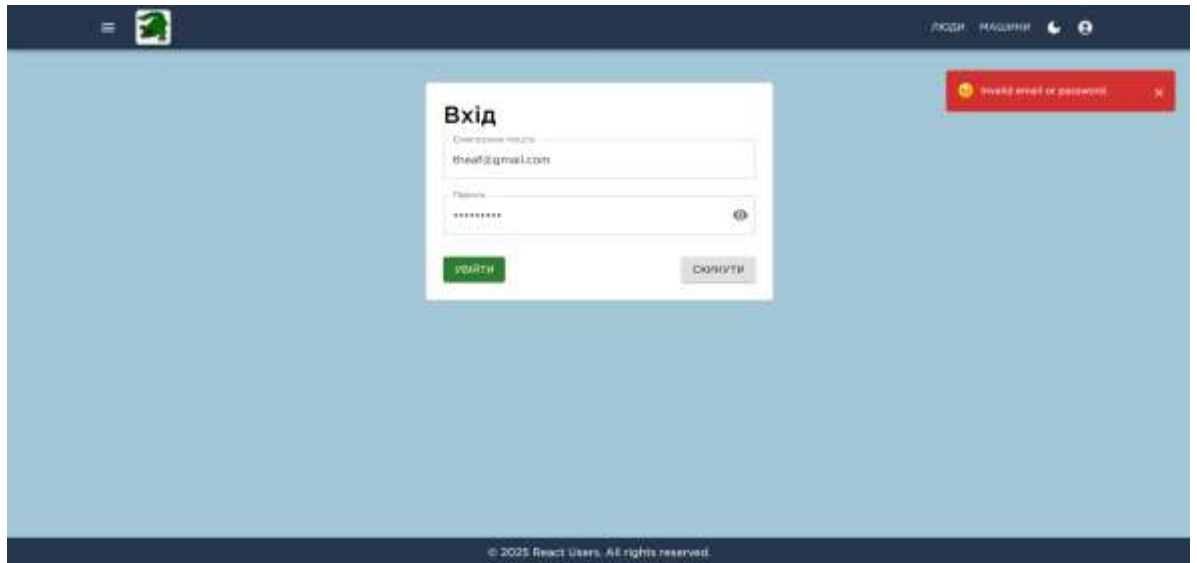


Рис. 4.1. Інтерфейс входу в систему

2) пошук і фільтрація:

- пошук за ПІБ і віком працює коректно, результати відображаються за 0.5–1 секунд;
- фільтрація за кількома параметрами (наприклад, ПІБ + фото) іноді займає до 1.5 секунд;

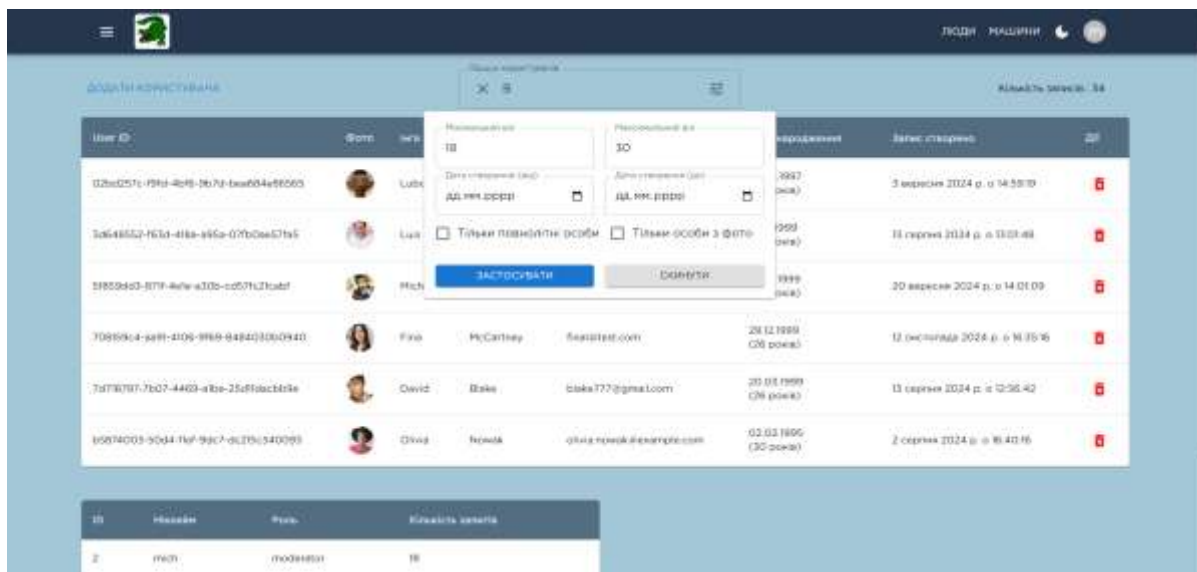


Рис. 4.2. – Інтерфейс пошуку з параметрами

3) додавання/редагування даних:

- форма додавання особи валідує обов'язкові поля (ПІБ, дата народження);
- дублікати осіб виявляються коректно, пропонується оновлення запису;
- логування змін працює, історія доступна в модулі аналітики;

4) аналітика (історія пошуку):

- модуль історії пошуку відображає запити для поточного користувача та всіх користувачів (для адміністратора);

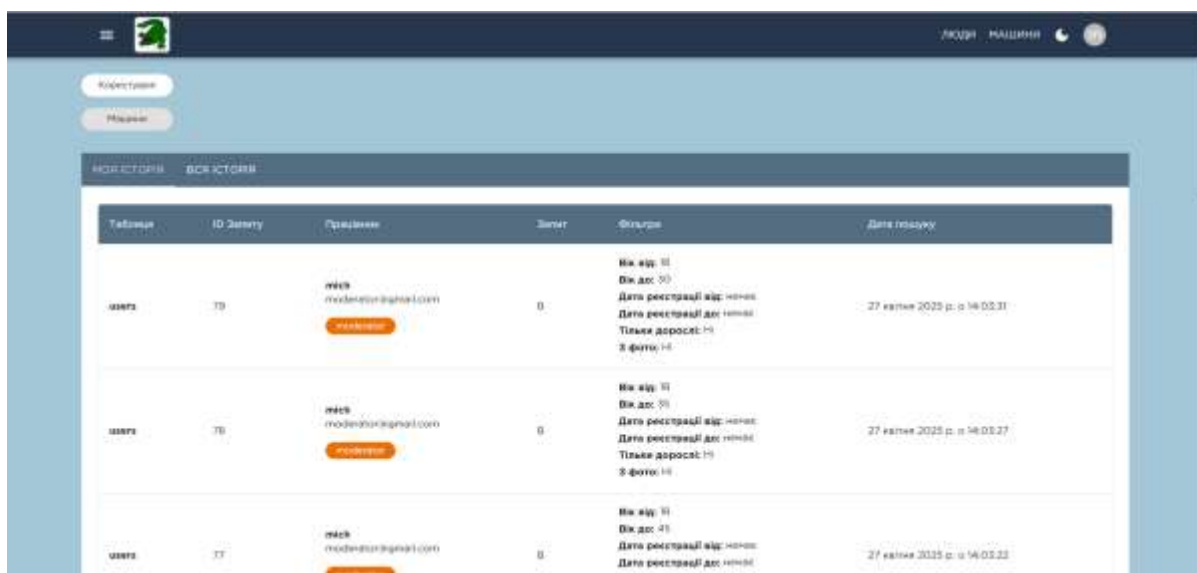


Рис. 4.3. Інтерфейс перегляду власної історії пошуку

5) безпека:

- ролі користувачів (оперативник, адміністратор) обмежують доступ до функцій коректно;
- Введення некоректних даних у форми не призводить до збоїв.

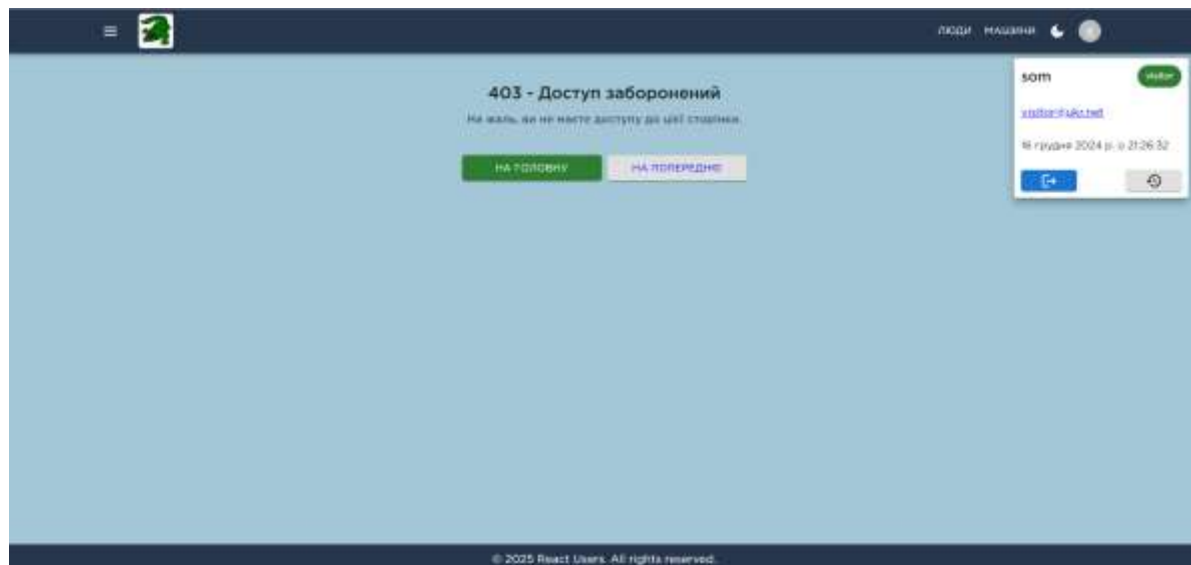


Рис. 4.4. Сторінка заборони доступу

Ручне тестування основних функцій системи довело що система працює справно.

4.3. Обґрунтування адекватності моделі та рекомендації

Остаточна оцінка проекту передбачає детальний аналіз придатності запропонованих моделей та методів управління для вирішення проблеми, визначеної мною в Розділі 1.

Придатність оцінюється на основі двох критеріїв: технічної придатності та ефективності управління. Мій аналіз показує, що запропонована вище інтегрована модель підходить для проекту створення веб-платформи для ORD, оскільки досягається баланс між високими вимогами безпеки та операційною ефективністю.

Технічна придатність підтверджується тим, що концептуальна та архітектурна моделі забезпечують централізоване зберігання та дозволяють легко реалізувати ключові функції, такі як ведення журналу дій та RBAC. Зокрема, модель RBAC довела свою придатність у забезпеченні принципу найменших привілеїв. Валідація показала, що користувачі в ролі «Оператор» не мають права переглядати журнали аудиту, а «Аналітик» не може редагувати дані, що мінімізує ризики витоку інформації та пошкодження.

Придатність управління доведена успішним впровадженням гібридної методології, яку я виклав у Матриці RACI. Суворий контроль на етапах планування та безпеки проекту забезпечив дотримання законодавчих вимог, тоді як ітеративний розвиток функціональності дозволив гнучко реагувати на швидкозмінні вимоги до пошуку. Крім того, результати розрахунку чистої приведеної вартості (NPV) підтверджують економічну доцільність, що вказує на те, що обрана стратегія управління бюджетом та ресурсами є фінансово життєздатною та забезпечує повернення інвестицій.

Запропоновано низку важливих рекомендацій та перспектив подальшого розвитку для максимізації ефективності системи та забезпечення її актуальності. По-перше, рекомендується міграція або інтеграція з графовою базою даних, така як наприклад, Neo4j для оптимізації аналізу складних зв'язків між різними об'єктами, що наразі реалізовано в реляційній базі і є неефективним підходом. Перехід на графову базу даних значно скоротить час обробки аналітичних запитів. По-друге, необхідно посилити технічну безпеку інформації для управління ризиком кіберзагроз шляхом впровадження двофакторної автентифікації для всіх ролей та розробки спеціального протоколу автоматичного блокування підозрілих IP-ад

ВИСНОВКИ

Кваліфікаційна робота присвячена дослідженню процесів управління проектом створення веб-платформи для централізованого управління даними в оперативно-розшуковій діяльності. Метою проекту було створення інноваційної, високозахищеної веб-платформи, що забезпечує автоматизацію аналізу даних та мінімізує управлінські ризики, пов'язані з фрагментацією конфіденційних даних. У ході дослідження було досягнуто поставленої мети та успішно вирішено весь комплекс завдань, визначених на начальному етапі роботи:

- проведено системний та управлінський аналіз проблемної області в якому було виявлено ключові проблеми, такі як розрізненість існуючих вже систем, високі ризики витоку чутливих даних та бюрократичні часові втрати. За допомогою PESTEL та SWOT-аналізу було обґрунтовано стратегічну необхідність переходу до централізованого IT-рішення, а також ідентифіковано критичні ризики, які стали основою для подальшого планування;
- розроблено концептуальні та архітектурні моделі системи, сформовано концептуальну модель, визначено функціональні та нефункціональні вимоги до проекту. Розроблено тривірневу архітектуру з використанням технологій ASP.NET Core + React.js + MySQL, модель управління доступом на основі ролей (RBAC), що гарантує суворе дотримання принципу мінімальних привілеїв та законодавства України «Про захист персональних даних»;
- було деталізовано структуру програмного рішення та проведено планування в якому розроблено структуру декомпозиції робіт (WBS) у трьох розрізах та Матрицю відповідальності RACI, що чітко закріпила повноваження за ключовими ролями та відобразила обрану гібридну методологію управління проектом. Це дозволило ефективно координувати команду та мінімізувати управлінські ризики;

- сформовано календарний план та визначено критичний шлях, побудовані Діаграма Ганта та розрахунок Методу критичного шляху (CPM) які показали, що загальна тривалість проєкту становить 274 дні. Ідентифікація Критичного Шляху дозволила зосередити управлінський контроль на ключових послідовних фазах: Проєктування БД, Розробка Бекенду та Тестування;
- здійснено економіко-математичне обґрунтування, за допомогою Аналізу Витрат і Вигод (CBA) та розрахунку Чистої Приведеної Вартості (NPV) було доведено фінансову життєздатність проєкту. Розрахунки показали, що NPV стає позитивною (проєкт окупається) на 7-му році експлуатації, що є прийнятним для довгострокових інвестицій у державний сектор, враховуючи високу соціальну та безпекову цінність проєкту;
- проведено валідацію результатів та обґрунтовано адекватність моделі (Розділ 4): Тестування MVP підтвердило повну відповідність нефункціональним вимогам, зокрема, високу швидкість пошуку (0.5–1.5 с при вимозі < 2 с) та коректну роботу багаторівневого контролю доступу (RBAC). Це доводить технічну та управлінську адекватність запропонованої моделі для підвищення операційної ефективності ОРД та мінімізації людського фактора.

Таким чином, у результаті кваліфікаційної роботи була успішно розроблена комплексна модель управління проєктом створення веб-платформи, яка забезпечує не лише технічну реалізацію сучасного ІТ-рішення, але й його економічну доцільність та відповідність найвищим стандартам безпеки. Проєкт має значні перспективи подальшого розвитку: зокрема, рекомендується інтеграція з графовою базою даних (Neo4j) та впровадження двофакторної автентифікації, що забезпечить перетворення платформи на ключовий аналітичний інструмент державного значення.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Закон України "Про оперативно-розшукову діяльність" № 2135-III від 18 лютого 2001 року (зі змінами станом на 2025 рік). URL: <https://zakon.rada.gov.ua/laws/show/2135-12#Text> (дата звернення: 09.10.2025).
2. Закон України "Про захист персональних даних" № 2297-VI від 1 червня 2010 року. URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text> (дата звернення: 09.10.2025).
3. Указ Президента України Про рішення Ради національної безпеки і оборони України від 14 травня 2021 року "Про Стратегію кібербезпеки України" № 447/2021 від 15 вересня 2021 року. URL: <https://zakon.rada.gov.ua/laws/show/447/2021#Text> (дата звернення: 09.10.2025).
4. ISO/IEC 27001:2013 - Information security management systems. URL: <https://www.iso.org/standard/27001> (дата звернення: 01.10.2025).
5. NIST Special Publication 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf> (дата звернення: 12.10.2025).
6. Кравченко О.В. "Автоматизація оперативно-розшукової діяльності: сучасні підходи та технології". Київ: НПУ імені М.П. Драгоманова, 2022. URL: <https://library.npu.edu.ua/> (дата звернення: 12.10.2025).
7. Brown T. "Artificial Intelligence in Criminal Investigation". Springer, 2024. URL: <https://journals.eanso.org/index.php/eajit/article/view/2141> (дата звернення: 13.10.2025).

8. Інтерпол. "Global Policing: Annual Report 2024". URL:
<https://www.interpol.int/en/Content/download/45678/567890/report2024.pdf>
(дата звернення: 01.10.2025).
9. FBI. "NCIC Overview and Statistics, 2024". URL:
<https://le.fbi.gov/informational-tools/ncic>
(дата звернення: 01.10.2025).
10. Національна поліція України. "Звіт про діяльність у 2024 році". URL:
https://www.kmu.gov.ua/storage/app/sites/1/17-civik-2018/zvit_2024/zvit-npu-2024.pdf
(дата звернення: 05.10.2025).
11. ПРОБЛЕМИ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАВООХОРОННИМИ СТРУКТУРАМИ УКРАЇНИ ТА ЗАКЛАДАМИ ВИЩОЇ ОСВІТИ ЗІ СПЕЦИФІЧНИМИ УМОВАМИ НАВЧАННЯ, ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ.
URL:
https://dspace.lvduvs.edu.ua/bitstream/1234567890/2378/1/21_12_2018.pdf
(дата звернення: 05.10.2025).
12. European Union Agency for Cybersecurity (ENISA). "Cybersecurity for Law Enforcement, 2024 Report". URL:
<https://www.enisa.europa.eu/publications/2024-report-on-the-state-of-the-cybersecurity-in-the-union> (дата звернення: 16.10.2025).
13. Захист інформаційної безпеки як функція держави. URL:
<https://mego.info/%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D1%96%D0%B0%D0%BB/23-%D0%B7%D0%B0%D1%85%D0%B8%D1%81%D1%82-%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%BE%D1%97->

https://www.dnuvs.ukr.edu.ua/wp-content/uploads/2024/05/zbirnyk_materialiv_kruglogo_stolu_ord_ta_ib.pdf

(дата звернення: 25.10.2025).

- 14.«ВЗАЄМОДІЯ ДЕРЖАВНИХ ОРГАНІВ ТА ГРОМАДСЬКОСТІ У СФЕРІ ПРОТИДІЇ КРИМІНАЛЬНИМ ПРАВОПОРУШЕННЯМ У ЦЕНТРАЛЬНИХ РЕГІОНАХ УКРАЇНИ». URL:

https://dnuvs.ukr.edu.ua/wp-content/uploads/2024/05/zbirnyk_materialiv_kruglogo_stolu_ord_ta_ib.pdf

(дата звернення: 25.10.2025)

- 15.«Управління проектами: проєктний підхід в сучасному менеджменті».

URL: https://odaba.edu.ua/upload/files/Materiali_konferentsii_UP_2023.pdf

(дата звернення: 25.10.2025)

- 16.“Системи і технології зв’язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку”. URL:

<https://dspace.nadpsu.edu.ua/bitstream/123456789/4628/1/5.pdf>

(дата звернення: 27.10.2025)

- 17.ВПЛИВ ПРОЦЕСІВ ДІДЖИТАЛІЗАЦІЇ НА ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНУ ДІЯЛЬНІСТЬ ПІДПРИЄМСТВА В УМОВАХ ГЛОБАЛІЗАЦІЇ. URL: <http://eee.khpi.edu.ua/article/view/332015> (дата звернення: 02.09.2025).

- 18.Conforto, E. C. Can agile project management be adopted by industries other than software development? Project Management Journal, 2014. URL:

<https://pmjournal.pmi.org/> (дата звернення: 17.10.2025).

- 19.SPA and PWA: What are they? URL: <https://www.forward.eu/insights/spa-and-pwa-what-are-they/> (дата звернення: 02.03.2025).

20. Singh. Top 10 Web Development Frameworks of 2021. URL: <https://www.techgeekbuzz.com/blog/web-development-frameworks/> (дата звернення: 02.03.2025).
21. Introducing react.dev. URL: <https://react.dev/blog/2023/03/16/introducing-react-dev> (дата звернення: 08.03.2025).
22. Ознайомлення з Хуками. URL: <https://uk.legacy.reactjs.org/docs/hooks-intro.html> (дата звернення: 08.03.2025).
23. Material UI - Overview. URL: <https://mui.com/material-ui/getting-started/> (дата звернення: 10.04.2025).
24. CSS Tutorial. URL: <https://www.tutorialspoint.com/css/index.htm> (дата звернення: 14.03.2025).
25. How CSS works. URL: https://developer.mozilla.org/docs/Learn/CSS/First_steps/How_CSS_works (дата звернення: 14.03.2025).
26. CSS with superpowers. URL: <https://sass-lang.com/documentation/> (дата звернення: 08.03.2025).
27. TypeScript Documentation. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 09.01.2025).
28. Що таке веб додаток? Різниця між сайтом, веб-додатком, SPA і PWA. URL: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/> (дата звернення: 02.03.2025).
29. Cross-Origin Resource Sharing (CORS). URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CORS> (дата звернення: 02.04.2025).
30. Flanagan D. JavaScript: The Definitive Guide, Sixth Edition. 2011. 1079 p. URL: <https://pepa.holla.cz/wp-content/uploads/2016/08/JavaScript-The-Definitive-Guide-6th-Edition.pdf> (дата звернення: 19.03.2024).

31. JS functions. URL: <https://developer.mozilla.org/docs/Web/JavaScript> (дата звернення: 19.02.2025).
32. Using TypeScript with React. URL: <https://react.dev/learn/typescript> (дата звернення: 11.04.2025).
33. What is CORS? URL: <https://aws.amazon.com/what-is/cross-origin-resource-sharing/> (дата звернення: 12.04.2025).
34. Michael McMillan «Data Structures and Algorithms with JavaScript» (дата звернення: 11.04.2025).
35. Marijn Haverbeke "Eloquent JavaScript" (дата звернення: 11.04.2025).
36. Kyle Simpson "You Don't Know JS" (дата звернення: 13.04.2025).
37. Douglas Crockford "JavaScript: The Good Parts" (дата звернення: 18.04.2025).
38. Alex Banks and Eve Porcello "Learning React" (дата звернення: 27.04.2025).
39. Robin Wieruch "The Road to learn React" (дата звернення: 18.04.2025).
40. Матриця відповідальності RACI: як розподілити задачі в ІТ і не тільки URL: <https://hub.kyivstar.ua/articles/matriczya-vidpovidalnosti-raci-yak-rozpodiliti-zadachi-v-it-i-ne-tilki> (дата звернення: 12.11.2025).

ДОДАТКИ

ДОДАТОК А

Код пошукової функції, файл UserRepository.cs

```
public async Task<IEnumerable<User>> SearchUsers(
    string? searchQuery,
    int? minAge,
    int? maxAge,
    DateTime? createdFrom,
    DateTime? createdTo,
    bool? onlyAdults,
    bool? onlyWithPhoto)
{
    var searchedUsers = new List<User>();

    var command = CreateCommand(
        @"SELECT
        users.UserID,
        users.FirstName,
        users.LastName,
        users.Email,
        users.DateOfBirth,
        users.CreatedAt,
        photos.ImageID,
        photos.ImageURL,
        photos.AltText,
        photos.UploadedAt
        FROM
        users LEFT JOIN photos ON users.UserID = photos.UserID
        WHERE
        (@SearchQuery IS NULL OR users.FirstName LIKE @SearchQuery OR
        users.LastName LIKE @SearchQuery OR users.UserID LIKE @SearchQuery)
        AND (@MinAge IS NULL OR YEAR(CURDATE()) -
        YEAR(users.DateOfBirth) >= @MinAge)
        AND (@MaxAge IS NULL OR YEAR(CURDATE()) -
        YEAR(users.DateOfBirth) <= @MaxAge)
        AND (@CreatedFrom IS NULL OR users.CreatedAt >= @CreatedFrom)
        AND (@CreatedTo IS NULL OR users.CreatedAt <= @CreatedTo)
        AND (@OnlyAdults IS NULL OR @OnlyAdults = false OR
        TIMESTAMPDIFF(YEAR, users.DateOfBirth, CURDATE()) >= 18)
```

```

        AND (@OnlyWithPhoto IS NULL OR @OnlyWithPhoto = false OR
photos.ImageID IS NOT NULL)
    ORDER BY
        users.UserID"
);

command.Parameters.AddWithValue("@SearchQuery",
string.IsNullOrEmpty(searchQuery) ? DBNull.Value : $"%{searchQuery}%");
command.Parameters.AddWithValue("@MinAge", minAge ??
(object)DBNull.Value);
command.Parameters.AddWithValue("@MaxAge", maxAge ??
(object)DBNull.Value);
command.Parameters.AddWithValue("@CreatedFrom", createdFrom ??
(object)DBNull.Value);
command.Parameters.AddWithValue("@CreatedTo", createdTo ??
(object)DBNull.Value);
command.Parameters.AddWithValue("@OnlyAdults", onlyAdults ??
(object)DBNull.Value);
command.Parameters.AddWithValue("@OnlyWithPhoto", onlyWithPhoto ??
(object)DBNull.Value);

_connection.Open();

using var reader = await command.ExecuteReaderAsync();

while (await reader.ReadAsync())
{
    var userId = reader.GetGuid(0);

    var user = searchedUsers.FirstOrDefault(u => u.UserID == userId);

    if (user == null)
    {
        user = new User
        {
            UserID = userId,
            FirstName = reader.GetString(1),
            LastName = reader.GetString(2),
            Email = reader.GetString(3),
            DateOfBirth = reader.GetDateTime(4),

```

```

        CreatedAt = reader.GetDateTime(5),
        Photos = new List<Photo>()
    };

    searchedUsers.Add(user);
}

if (!reader.IsDBNull(6))
{
    user.Photos.Add(new Photo
    {
        ImageID = reader.GetInt32(6),
        UserID = reader.GetGuid(0),
        ImageURL = reader.GetString(7),
        AltText = reader.GetString(8),
        UploadedAt = reader.GetDateTime(9)
    });
}
}
_connection.Close();
return searchedUsers;
}

```

ДОДАТОК Б

Код компонента з таблицею осіб, файл UsersTable.tsx

```

import { useEffect, useState } from "react";
import { useDispatch, useSelector } from "../../store/hooks";
import {
    getAllUsers,
    deleteUser,
    searchUsersByName,
    getUsersCount,
} from "../../store/userSlice";
import DeleteForeverIcon from "@mui/icons-material/DeleteForever";
import UserTableSkeletonRow from "./UserTableSkeletonRow";
import {
    Avatar,
    Box,
    Button,

```

```

IconButton,
Paper,
Skeleton,
Table,
TableBody,
TableCell,
TableContainer,
TableRow,
} from "@mui/material";
import { RootState } from "../../store/store";
import { useNavigate } from "react-router-dom";
import NoProfilePicture from "../../assets/images/noProfilePicture.png";
import ConfirmDeleteUserModal from "../../modals/ConfirmDeleteUserModal";
import { User } from "../../types/userTypes";
import useDebounce from "../../hooks/useDebounce";
import CustomSearchUsersInput from "../../ui/CustomSearchUsersInput";
import CustomErrorBlock from "../../ui/CustomErrorBlock";
import CustomNotFoundPaper from "../../ui/CustomNotFoundPaper";
import CustomPagination from "../../ui/CustomPagination";
import UserTableHead from "../../UserTableHead";
import { formatDateTime } from "../../utils/formatDateTime";

const UsersTable = () => {
  const navigate = useNavigate();
  const dispatch = useAppDispatch();

  const lightTheme = useAppSelector((state) => state.theme.lightTheme);

  const { users, loading, error, usersCount } = useAppSelector(
    (state: RootState) => state.user
  );

  const { staff } = useAppSelector((state: RootState) => state.auth);

  useEffect(() => {
    dispatch(getUsersCount()); // usersCount
  }, [dispatch]);

  const [isTyping, setIsTyping] = useState<boolean>(false); // идет ли набор текста

```

```

const [searchQuery, setSearchQuery] = useState<string>(""); // Состояние для
строки поиска
const debouncedSearchQuery = useDebounce(searchQuery, 1500); // Используем
дебаунс

const [openDeleteModal, setOpenDeleteModal] = useState<boolean>(false);
const [selectedUser, setSelectedUser] = useState<User | null>(null);

// параметры для getAllUsers
const [currentPage, setCurrentPage] = useState<number>(1);
const pageSize = 15;
const totalPages = Math.ceil(Number(usersCount) / pageSize); // к-во страниц
const [sortBy, setSortBy] = useState<string>("UserID");
const [sortDirection, setSortDirection] = useState<string>("asc");

useEffect(() => {
  if (debouncedSearchQuery) {
    // Если строка поиска не пустая
    setIsTyping(false);
    dispatch(
      searchUsersByName({
        searchQuery: debouncedSearchQuery,
        minAge: undefined,
        maxAge: undefined,
        createdFrom: undefined,
        createdTo: undefined,
        onlyAdults: undefined,
        onlyWithPhoto: undefined,
      })
    );
  } else {
    setIsTyping(false);
    dispatch(
      getAllUsers({
        pageNumber: currentPage,
        pageSize,
        sortBy,
        sortDirection,
      })
    );
  }
});

```

```

    }
  }, [debouncedSearchQuery, currentPage, dispatch, sortBy, sortDirection]);

  // Обработчик изменения значения строки поиска
  const handleSearchInputChange = (e: React.ChangeEvent<HTMLInputElement>)
=> {
    setSearchQuery(e.target.value);
    setIsTyping(true);
  };

  // очистка строки
  const handleClearSearch = () => {
    setSearchQuery("");
    setIsTyping(false);
    dispatch(
      getAllUsers({ pageNumber: currentPage, pageSize, sortBy, sortDirection })
    );
  };

  const handleDelete = () => {
    if (selectedUser) {
      dispatch(deleteUser(selectedUser.userID));
      setOpenDeleteModal(false);
    }
  };

  // Обработчик изменения страницы пагинации
  const handlePageChange = (
    event: React.ChangeEvent<unknown>,
    value: number
  ) => {
    event.preventDefault();
    setCurrentPage(value);
  };

  // Обработчик сортировки
  const handleSort = (column: string) => {
    if (sortBy === column) {
      setSortDirection(sortDirection === "asc" ? "desc" : "asc");
    } else {

```

```

setSortBy(column);
setSortDirection("asc");
}
};

return (
  <>
  <br />

  <Box
    sx={{
      display: "flex",
      flexDirection: "row",
      justifyContent: "space-between",
      alignItems: "center",
      width: "100%",
      height: "50px",
      marginBottom: "16px",
    }}
  >
  <Button
    onClick={() => navigate("/user/add")}
    disabled={staff?.role !== "admin" && staff?.role !== "moderator"}
  >
    Додати користувача
  </Button>

  <CustomSearchUsersInput
    searchQuery={searchQuery}
    handleSearchInputChange={handleSearchInputChange}
    handleClearSearch={handleClearSearch}
    placeholder="Пошук користувачів"
  />

  <Box
    sx={{
      width: "220px",
      fontSize: "14px",
      display: "flex",
      flexDirection: "row",

```

```

    justifyContent: "space-around",
    alignItems: "center",
  }}
>
Кількість записів :{ " " }
{loading ? (
  <Skeleton
    animation="wave"
    height={35}
    variant="text"
    width={30}
    sx={{
      display: "inline-block",
      alignItems: "center",
    }}
  />
): (
  usersCount
)}
</Box>
</Box>

<TableContainer component={Paper}>
  <Table>
    <UserTableHead
      lightTheme={lightTheme}
      sortBy={sortBy}
      sortDirection={sortDirection as "asc" | "desc"}
      handleSort={handleSort}
    />

    <TableBody>
      {loading || isTyping ? (
        <>
          {Array.from({ length: 10 }).map((_, i) => (
            <UserTableSkeletonRow key={`skeleton-${i}`} />
          ))}
        </>
      ) : error ? (
        <TableRow>

```

```

<TableCell colSpan={8}>
  <CustomErrorBlock />
</TableCell>
</TableRow>
): users.length === 0 && searchQuery ? (
  <TableRow>
    <TableCell colSpan={8}>
      <CustomNotFoundPaper
        errorMessage={`Користувача з іменем ${searchQuery} не знайдено`}
      />
    </TableCell>
  </TableRow>
): (
  users.map((user) => (
    <TableRow
      key={user.userID}
      onClick={() => navigate(`/user/${user.userID}`)}
      sx={{
        cursor: "pointer",
        transition: "all 0.3s ease-in-out",
        "&:hover": {
          backgroundColor: "#f5f5f5",
        },
      }}
    >
      <TableCell>{user.userID}</TableCell>
      <TableCell>
        {user.photos?.length > 0 ? (
          <Avatar
            alt={user.photos[0].altText}
            src={user.photos[0].imageUrl}
          />
        ) : (
          <Avatar alt="No profile image" src={NoProfilePicture} />
        )}
      </TableCell>
      <TableCell>{user.firstName}</TableCell>
      <TableCell>{user.lastName}</TableCell>
      <TableCell>{user.email}</TableCell>
      <TableCell>

```

```

        { new Date(user.dateOfBirth).toLocaleDateString("uk-UA")}
        <br />(
        { new Date().getFullYear() -
          new Date(user.dateOfBirth).getFullYear() }{ " " }
        покiв)
      </TableCell>
      <TableCell>{formatDateTime(user.createdAt)}</TableCell>
      <TableCell>
        <IconButton
          disabled={staff?.role !== "admin"}
          onClick={e => {
            e.stopPropagation();
            setSelectedUser(user);
            setOpenDeleteModal(true);
          }}
        >
          <DeleteForeverIcon sx={{ color: "red" }} />
        </IconButton>
      </TableCell>
    </TableRow>
  )
)}
</TableBody>
</Table>
</TableContainer>

{!searchQuery && (
  <Box
    sx={{ display: "flex", justifyContent: "center", marginTop: "16px" }}
  >
    <CustomPagination
      totalPages={totalPages}
      currentPage={currentPage}
      handlePageChange={handlePageChange}
    />
  </Box>
)}

<br />
<br />

```

```

    {selectedUser && (
      <ConfirmDeleteUserModal
        open={openDeleteModal}
        handleClose={() => setOpenDeleteModal(false)}
        handleDelete={handleDelete}
        user={selectedUser}
      />
    )}
  </>
);
};

export default UsersTable;

```

ДОДАТОК В

Код всіх функцій пов'язаних з автомобілями, файл CarRepository.cs

```

using ASP_users.Interfaces;
using ASP_users.Models;
using MySqlConnection;

namespace ASP_users.Repositories
{
    public class CarRepository : BaseRepository, ICarRepository
    {
        public CarRepository(MySqlConnection connection) : base(connection) { }

        public async Task<IEnumerable<Car>> GetAllCars(int pageNumber, int
        pageSize, string sortBy, string sortDirection)
        {
            var offset = (pageNumber - 1) * pageSize;

            switch (sortBy)
            {
                case "Firm":
                    sortBy = "cars.Firm";
                    break;
            }
        }
    }
}

```

```

        case "Model":
            sortBy = "cars.Model";
            break;
        case "Year":
            sortBy = "cars.Year";
            break;
    };

    switch (sortDirection)
    {
        case "asc":
            sortDirection = "ASC";
            break;
        case "desc":
            sortDirection = "DESC";
            break;
    };

    var carsList = new List<Car>();

    var command = CreateCommand(
        $"SELECT
        CarID,
        UserID,
        Firm,
        Model,
        Color,
        Year,
        LicensePlate,
        CarPhotoURL
        FROM
        Cars
        ORDER BY
        {sortBy} {sortDirection}
        LIMIT
        @offset, @pageSize"
    );

    command.Parameters.AddWithValue("@offset", offset);
    command.Parameters.AddWithValue("@pageSize", pageSize);

```

```

command.Parameters.AddWithValue("@sortBy", sortBy);
command.Parameters.AddWithValue("@sortDirection", sortDirection);

_connection.Open();

using var reader = await command.ExecuteReaderAsync();

while (await reader.ReadAsync())
{
    var carId = reader.GetInt32(0);

    var car = carsList.FirstOrDefault(x => x.CarID == carId); // Перевіряємо
чи машину вже знайдено

    if (car == null) // Якщо машини не знайдено
    {
        car = new Car // Створюємо нову машину
        {
            CarID = carId,
            UserID = reader.GetGuid(1),
            Firm = reader.GetString(2),
            Model = reader.GetString(3),
            Color = reader.GetString(4),
            Year = reader.GetInt32(5),
            LicensePlate = reader.GetString(6),
            CarPhotoURL = reader.IsDBNull(7) ? null : reader.GetString(7)
        };
        carsList.Add(car); // Додаємо машину до списку
    }
}

_connection.Close();

return carsList;
}

public async Task<Car> GetCarById(int carId)
{
    Car car = null;

```

```

var command = CreateCommand(
    @"SELECT
        CarID,
        UserID,
        Firm,
        Model,
        Color,
        Year,
        LicensePlate,
        CarPhotoURL
    FROM
        Cars
    WHERE
        CarID = @CarID"
);

command.Parameters.AddWithValue("@CarID", carId);

_connection.Open();

var reader = await command.ExecuteReaderAsync();

while (await reader.ReadAsync())
{
    if (car == null)
    {
        car = new Car
        {
            CarID = reader.GetInt32(0),
            UserID = reader.GetGuid(1),
            Firm = reader.GetString(2),
            Model = reader.GetString(3),
            Color = reader.GetString(4),
            Year = reader.GetInt32(5),
            LicensePlate = reader.GetString(6),
            CarPhotoURL = reader.IsDBNull(7) ? null : reader.GetString(7)
        };
    }
}

_connection.Close();

```

```

    return car;
}

public async Task<IEnumerable<Car>> GetAllUsersCars(Guid userId)
{
    var usersCars = new List<Car>();

    var command = CreateCommand(
        @"SELECT
            CarID,
            UserID,
            Firm,
            Model,
            Color,
            Year,
            LicensePlate,
            CarPhotoURL
        FROM
            Cars
        WHERE
            UserID = @UserID"
    );

    command.Parameters.AddWithValue("@UserID", userId);

    _connection.Open();

    var reader = await command.ExecuteReaderAsync();

    while (await reader.ReadAsync())
    {
        var carID = reader.GetInt32(0);

        var car = usersCars.FirstOrDefault(x => x.CarID == carID);

        if (car == null)
        {
            car = new Car
            {

```

```

        CarID = carID,
        UserID = reader.GetGuid(1),
        Firm = reader.GetString(2),
        Model = reader.GetString(3),
        Color = reader.GetString(4),
        Year = reader.GetInt32(5),
        LicensePlate = reader.GetString(6),
        CarPhotoURL = reader.IsDBNull(7) ? null : reader.GetString(7)
    };
    usersCars.Add(car);
}
}

_connection.Close();

return usersCars;
}

public async Task<IEnumerable<Car>> SearchCars(
    string? searchQuery,
    int? minYear,
    int? maxYear,
    string? carColor,
    bool? onlyWithPhoto)
{
    var searchedCars = new List<Car>();

    var command = CreateCommand(
        @"SELECT
        CarID,
        UserID,
        Firm,
        Model,
        Color,
        Year,
        LicensePlate,
        CarPhotoURL
FROM
    Cars
WHERE

```

```

        (@searchQuery IS NULL OR Firm LIKE @searchQuery OR Model
LIKE @searchQuery OR LicensePlate LIKE @searchQuery)
        AND (@minYear IS NULL OR Year >= @minYear)
        AND (@maxYear IS NULL OR Year <= @maxYear)
        AND (@carColor IS NULL OR Color LIKE @carColor)
        AND (@OnlyWithPhoto = 0 OR (CarPhotoURL IS NOT NULL AND
CarPhotoURL != ""))
    ORDER BY
        CarID"
    );

    command.Parameters.AddWithValue("@searchQuery",
string.IsNullOrEmpty(searchQuery) ? DBNull.Value : $"%{searchQuery}%");
    command.Parameters.AddWithValue("@MinYear", minYear ??
(object)DBNull.Value);
    command.Parameters.AddWithValue("@MaxYear", maxYear ??
(object)DBNull.Value);
    command.Parameters.AddWithValue("@CarColor", carColor ??
(object)DBNull.Value);
    command.Parameters.AddWithValue("@OnlyWithPhoto", onlyWithPhoto ??
(object)DBNull.Value);

    _connection.Open();

    var reader = await command.ExecuteReaderAsync();

    while (await reader.ReadAsync())
    {
        var carID = reader.GetInt32(0);

        var car = searchedCars.FirstOrDefault(x => x.CarID == carID);

        if (car == null)
        {
            car = new Car
            {
                CarID = carID,
                UserID = reader.GetGuid(1),
                Firm = reader.GetString(2),
                Model = reader.GetString(3),
            };
        }
    }
}

```

```

        Color = reader.GetString(4),
        Year = reader.GetInt32(5),
        LicensePlate = reader.GetString(6),
        CarPhotoURL = reader.GetString(7)
    };
    searchedCars.Add(car);
}
}
_connection.Close();

return searchedCars;
}

public async Task UpdateCar(int carId, Car car)
{
    var command = CreateCommand(
        @"UPDATE
        Cars
        SET
        Firm = @Firm,
        Model = @Model,
        Color = @Color,
        Year = @Year,
        LicensePlate = @LicensePlate,
        CarPhotoURL = @CarPhotoURL
        WHERE
        CarID = @CarID"
    );

    command.Parameters.AddWithValue("@CarID", carId);
    command.Parameters.AddWithValue("@Firm", car.Firm);
    command.Parameters.AddWithValue("@Model", car.Model);
    command.Parameters.AddWithValue("@Color", car.Color);
    command.Parameters.AddWithValue("@Year", car.Year);
    command.Parameters.AddWithValue("@LicensePlate", car.LicensePlate);
    command.Parameters.AddWithValue("@CarPhotoURL", car.CarPhotoURL);

    _connection.Open();

    await command.ExecuteNonQueryAsync();
}

```

```

        _connection.Close();
    }

    public async Task AddCarToUser(Guid userId, Car car)
    {
        var command = CreateCommand(
            @"INSERT INTO Cars
            (UserID, Firm, Model, Color, Year, LicensePlate, CarPhotoURL)
            VALUES
            (@UserID, @Firm, @Model, @Color, @Year, @LicensePlate,
@CarPhotoURL)"
        );

        command.Parameters.AddWithValue("@UserID", userId);
        command.Parameters.AddWithValue("@Firm", car.Firm);
        command.Parameters.AddWithValue("@Model", car.Model);
        command.Parameters.AddWithValue("@Color", car.Color);
        command.Parameters.AddWithValue("@Year", car.Year);
        command.Parameters.AddWithValue("@LicensePlate", car.LicensePlate);
        command.Parameters.AddWithValue("@CarPhotoURL", car.CarPhotoURL);

        _connection.Open();
        await command.ExecuteNonQueryAsync();
        _connection.Close();
    }

    public async Task DeleteCar(int carId)
    {
        var command = CreateCommand(
            @"DELETE FROM
            Cars
            WHERE
            CarID = @CarID"
        );

        command.Parameters.AddWithValue("@CarID", carId);

        _connection.Open();
    }

```

```

        await command.ExecuteNonQueryAsync();

        _connection.Close();
    }

    // HALPERS
    public async Task<int> GetCarsCount()
    {
        var command = CreateCommand(
            @"SELECT
            COUNT(*)
            FROM
            Cars"
        );

        _connection.Open();

        var carsCount = (int)(long)await command.ExecuteScalarAsync();

        _connection.Close();

        return carsCount;
    }

    public async Task<Car> CheckLicensePlateExist(string licensePlate)
    {
        Car car = null;

        var command = CreateCommand(
            @"SELECT
            CarID,
            UserID,
            Firm,
            Model,
            Color,
            Year,
            LicensePlate,
            CarPhotoURL
            FROM

```

```

        Cars
        WHERE
            LicensePlate = @LicensePlate"
    );

    command.Parameters.AddWithValue("@LicensePlate", licensePlate);

    _connection.Open();

    var reader = await command.ExecuteReaderAsync();

    while (await reader.ReadAsync())
    {
        if (car == null)
        {
            car = new Car
            {
                CarID = reader.GetInt32(0),
                UserID = reader.GetGuid(1),
                Firm = reader.GetString(2),
                Model = reader.GetString(3),
                Color = reader.GetString(4),
                Year = reader.GetInt32(5),
                LicensePlate = reader.GetString(6),
                CarPhotoURL = reader.IsDBNull(7) ? null : reader.GetString(7)
            };
        }
    }
    _connection.Close();

    return car;
}
}
}

```