

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАДИМА ГЕТЬМАНА  
Навчально-науковий інститут  
«Інститут інформаційних технологій в економіці»  
Кафедра інформаційних систем в економіці**

**ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»**

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

**КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА**

на тему: «Розробка платформи для онлайн-тестування»

здобувача Заболотіна Андрія Олександровича

\_\_\_\_\_

*(Підпис)*

**Науковий керівник:**

к.т.н., доцент

\_\_\_\_\_Кривошеєв К. В.

**Робота допущена до захисту перед**

**екзаменаційною комісією з**

**атестації здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент

\_\_\_\_\_Тішков Б.О.

**Київ 2025**

Київ 2025

Міністерство освіти і науки України  
Київський національний економічний університет імені Вадима  
Гетьмана  
Навчально-науковий інститут «Інститут інформаційних технологій в  
економіці»  
Кафедра інформаційних систем в економіці  
**ОСВІТНЬО\_ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»**  
галузь знань 12 «Інформаційні технології»  
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)

освітньо-професійної програми

\_\_\_\_\_ Помазун О.М.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

\_\_\_\_\_ Тішков Б.О.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

здобувача вищої освіти *Заболотіна Андрія Олександровича*

*очної (денної) форми навчання*

на підготовку кваліфікаційної бакалаврської роботи  
на тему: **«Розробка платформи для онлайн-тестування»**

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р.  
№ 466-ст.

**Кваліфікаційна бакалаврська робота виконується на матеріалах  
отриманих з мережі Інтернет, відкритих джерел інформації**

**План кваліфікаційної бакалаврської роботи**

**Розділ I** *Характеристика та аналіз предметної галузі*

**Розділ II** *Розробка вимог і моделювання інформаційної системи/підсистеми*

**Розділ III** *Проектування та реалізація компонентів системи/підсистеми*

**Об'єкт дослідження** *є процес створення інформаційної системи для онлайн-тестування знань користувачів.*

**Предмет дослідження** *є методи, моделі та засоби розробки платформи для створення, проходження й оцінювання тестових завдань у веб-середовищі.*

**Мета кваліфікаційної бакалаврської роботи** *розробити платформу для онлайн-тестування, яка забезпечує автоматизоване створення, адміністрування тестів і обробку результатів зручним і надійним способом.*

**Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:**

**У розділі I** необхідно: *описати предметну галузь і об'єкт дослідження, проаналізувати завдання, які належить вирішити, в тому числі й їх економічний зміст, а також проводиться аналіз існуючих програмних засобів в цій області, наводячи недоліки й переваги кожної з них.*

**У розділі II** необхідно: *розробити вимоги до системи і обґрунтувати методи моделювання системи, провести опис поставленої задачі та пояснити алгоритм її розв'язання.*

**У розділі III** необхідно: *обґрунтувати структуру інформаційного масиву, описати формат вхідних та вихідних даних, пояснити вибір технічних засобів та програмного забезпечення, що використовуються для створення мобільних додатків та продемонструвати результати використання технологій.*

**Завдання підготував  
науковий керівник** \_\_\_\_\_

**Кривошесв Константин В.**

“ \_\_\_\_\_ ” \_\_\_\_\_ 2025 р.

**Завдання одержав  
студент** \_\_\_\_\_

**Заболотін Андрій Олександрович**

“ \_\_\_\_\_ ” \_\_\_\_\_ 2025 р.

**Відгук**  
про кваліфікаційну бакалаврську роботу  
здобувача навчально-наукового інституту  
«Інститут інформаційних технологій в економіці»  
освітньо-професійної програми  
«Комп'ютерні науки»

*Заболотіна Андрія Олександровича*  
на тему  
**«Розробка платформи для онлайн-тестування»**

1. Актуальність теми:

Робота присвячена розробці платформи для онлайн-тестування – актуальному завданню в умовах цифровізації освіти та зростання потреби у дистанційному оцінюванні знань. Вибрана тема є сучасною, прикладною і має значний практичний потенціал.

2. Позитивні риси кваліфікаційного бакалаврської роботи:

Автор роботи, здійснив проектування та розробку веб-платформи з використанням сучасного стеку технологій, проаналізував аналоги та визначив напрями подальшого удосконалення продукту.

3. Наявність самостійних розробок автора:

У процесі роботи студент самостійно сформував вимоги до системи, виконав проектування структури, моделі даних та реалізував веб-прототип системи з можливістю створення й проходження тестів, що свідчить про його практичну підготовку.

4. Цінність теоретичних висновків та практичних рекомендацій:

У роботі подано обґрунтовані теоретичні підходи до проектування систем онлайн-тестування, що можуть бути використані при створенні подібних освітніх платформ.

5. Наявність недоліків:

До недоліків можна віднести обмежений рівень автоматичного аналізу результатів тестування. У майбутньому систему доцільно доопрацювати шляхом впровадження автоматизованого тестування та розширеної аналітики.

6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:

Представлена робота відповідає встановленим вимогам, що висуваються до кваліфікаційних бакалаврських робіт. Виявлені запозичення не містять ознак плагіату та необхідні для більш зрозумілого сприйняття загальної суті дослідження. Кваліфікаційна бакалаврська робота Заболотіна Андрія Олександровича рекомендується до захисту перед комісією з атестації здобувачів першого (бакалаврського) рівня вищої освіти з позитивною оцінкою.

Науковий керівник

\_\_\_\_\_ доцент, к.т.н. Кривошесєв К.В.  
(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**РЕЦЕНЗІЯ**  
на кваліфікаційну бакалаврську роботу  
здобувача вищої освіти  
**Заболотіна Андрія Олександровича**

**Тема: «Розробка платформи для онлайн-тестування»**

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення

У сучасних умовах цифровізації освіти розробка платформ для онлайн-тестування набуває особливої актуальності. Вибір теми є обґрунтованим з огляду на потребу в дистанційних засобах контролю знань як у навчальних закладах, так і в корпоративному секторі. Запропонована система відповідає поточним викликам у галузі електронного навчання.

Якість проведеного дослідження

Робота виконана на високому рівні. Автор послідовно здійснив аналіз предметної галузі, сформулював вимоги до системи, виконав моделювання структури та інтерфейсу з використанням UML-діаграм, реалізував функціональний прототип веб-платформи. Дослідження охоплює як теоретичні аспекти, так і практичну реалізацію завдання.

Позитивні риси кваліфікаційної бакалаврської роботи

Студентом було проведено аналіз існуючих цифрових рішень та сформовано власну концепцію платформи, яка об'єднує кращі практики. Реалізований функціонал забезпечує створення тестів, збереження результатів і базову аналітику, що робить систему зручною для авторів тестів і учасників тестування.

Зауваження

У роботі варто було б приділити більше уваги розширеному аналізу результатів тестування та інтеграції з зовнішніми LMS-системами. Деякі діаграми могли б бути деталізовані для повнішого розкриття взаємодій між компонентами.

Практична значимість висновків і рекомендацій

Результати, отримані в ході виконання кваліфікаційної роботи, можуть бути використані для створення або модернізації онлайн-сервісів з тестування. Розроблена платформа придатна до впровадження для проведення навчальних тестів, іспитів, а також у корпоративному середовищі для внутрішньої атестації персоналу. Рекомендації щодо розширення функціональності є реалістичними та можуть бути втілені при подальшій розробці системи.

Місце роботи та посада рецензента

ТОВ «ПАТРІОТ РЕНТАЛ»  
Team-Lead IT Відділу

**Метлін Віктор**



## АНОТАЦІЯ

кваліфікаційної бакалаврської роботи  
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,  
виконаної на тему: «**Розробка платформи для онлайн-тестування**»  
Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена створенню веб-платформи для онлайн-тестування, що забезпечує автоматизоване створення, адміністрування та перевірку тестових завдань у веб-середовищі.

У першому розділі виконано аналіз предметної галузі, досліджено сучасні підходи до онлайн-тестування, здійснено огляд існуючих платформ і виявлено їх переваги та недоліки.

У другому розділі розроблено вимоги до інформаційної системи, сформовано математичне забезпечення, постановку задачі та алгоритм функціонування системи. Виконано моделювання структури, поведінки та інтерфейсу системи засобами UML.

У третьому розділі описано інформаційне, технічне та програмне забезпечення, реалізовано прототип веб-додатку, продемонстровано результати його функціонування.

Робота завершується узагальненням результатів та наданням рекомендацій щодо подальшого вдосконалення системи.

## РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 61 сторінок, 14 таблиць, 36 рисунків, список використаних джерел з 30 найменувань, 2 додатків.

### **«Розробка платформи для онлайн-тестування»**

*Перелік ключових слів:* онлайн-тестування, інформаційна система, веб-платформа, автоматизація оцінювання, інтерфейс, адаптивне тестування, моделювання, розробка.

*Предметом дослідження* є методи, моделі та засоби розробки платформи для створення, проходження й оцінювання тестових завдань у веб-середовищі.

*Об'єкт дослідження* – процес створення інформаційної системи для онлайн-тестування знань користувачів.

*Мета роботи* полягає в розробці платформи, яка забезпечить зручне адміністрування, складання тестів та автоматизоване оцінювання результатів у сучасному веб-середовищі.

*Завдання дослідження:*

- проаналізувати предметну галузь та оглянути існуючі системи онлайн-тестування;
- визначити вимоги до інформаційної системи, сформулювати функціональні та нефункціональні характеристики;
- розробити інформаційну модель та алгоритми функціонування системи;
- здійснити моделювання структури, поведінки та інтерфейсу веб-платформи;
- реалізувати прототип платформи та провести тестування її основних функцій.

Застосовані інструменти розробки: Python, фреймворк REACT, фреймворк Django, HTML/CSS, система управління базами даних PostgreSQL.

*Практична цінність:* система може бути використана в закладах освіти та корпоративному навчанні для проведення тестування, сертифікацій та оцінки знань.

Рік виконання – 2025.

Рік захисту – 2025

## ЗМІСТ

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....                                                          | 3  |
| ВСТУП .....                                                                              | 4  |
| РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ.....                                 | 5  |
| 1.1. Сутність та значення онлайн-тестування.....                                         | 5  |
| 1.2. Аналіз сучасних підходів до онлайн-тестування .....                                 | 6  |
| 1.3. Огляд існуючих платформ для онлайн-тестування.....                                  | 7  |
| 1.4. Аналіз літературних джерел та практичного досвіду використання ІС і технологій..... | 11 |
| РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....                         | 13 |
| 2.1 Аналіз і специфікація вимог до інформаційної системи .....                           | 13 |
| 2.2 Постановка та алгоритм розв’язання задачі .....                                      | 19 |
| 2.3 Моделювання інформаційної системи .....                                              | 24 |
| РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ .....                            | 39 |
| 3.1 Інформаційне забезпечення.....                                                       | 39 |
| 3.2 Технічне забезпечення .....                                                          | 45 |
| 3.3 Програмне забезпечення.....                                                          | 47 |
| 3.4 Результати реалізації інформаційної системи .....                                    | 49 |
| ВИСНОВКИ .....                                                                           | 53 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....                                                         | 55 |
| ДОДАТКИ .....                                                                            | 57 |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІС – Інформаційна система

ІТ – Інформаційні технології

СКБД – Система керування базами даних

ІІІ – Штучний інтелект

API – Application Programming Interface (інтерфейс прикладного програмування)

UI – User Interface (користувацький інтерфейс)

UML – Unified Modeling Language (уніфікована мова моделювання)

HTTPS – Hypertext Transfer Protocol Secure (протокол захищеної передачі гіпертексту)

SQL – Structured Query Language (структурована мова запитів)

CSRF – Cross-Site Request Forgery (міжсайтове підроблення запиту)

XSS – Cross-Site Scripting (міжсайтове скриптування)

UUID – Universally Unique Identifier (універсальний унікальний ідентифікатор)

LMS – Learning Management System (система управління навчанням)

CRUD – Create, Read, Update, Delete (створення, читання, оновлення, видалення)

UX – User Experience (досвід користувача)

## ВСТУП

*Актуальність теми дослідження* зумовлена стрімким розвитком цифрових технологій та зростанням потреби в гнучких і доступних інструментах для дистанційного навчання та оцінювання знань. Онлайн-тестування стає невід'ємною частиною сучасної освіти, корпоративного навчання та сертифікаційних процесів. Більшість існуючих рішень мають обмежений функціонал, не забезпечують достатнього рівня зручності, що створює потребу у створенні більш сучасних і універсальних платформ.

*Об'єктом дослідження* кваліфікаційного бакалаврської роботи є процес створення інформаційної системи для онлайн-тестування знань користувачів.

*Предметом дослідження* є методи, моделі та засоби розробки платформи для створення, проходження й оцінювання тестових завдань у веб-середовищі.

*Метою і завданням дослідження* є розробка платформи для онлайн-тестування, яка забезпечує автоматизоване створення, адміністрування тестів і обробку результатів зручним і надійним способом.

Для досягнення поставленої мети передбачено вирішення таких завдань:

- дослідити предметну область онлайн-тестування;
- провести аналіз теоретичних аспектів створення інформаційних систем для тестування знань;
- визначити функціональні, нефункціональні та бізнес-вимоги до системи;
- розробити постановку задачі та інформаційну модель;
- створити алгоритм функціонування системи;
- здійснити моделювання поведінки та структури майбутньої системи;
- спроектувати інформаційне, технічне та програмне забезпечення;
- реалізувати прототип веб-платформи для онлайн-тестування.

*Теоретична, методична та практична значущість отриманих результатів* полягає у створенні комплексу підходів до розробки освітньої платформи, яку можна адаптувати до потреб навчальних закладів і корпоративного сектору.

Практичні результати дослідження полягають у розробленні інформаційної системи, яка може бути використана для онлайн тестування.

## РОЗДІЛ 1 ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

### 1.1. Сутність та значення онлайн-тестування

Онлайн-тестування – це метод перевірки знань, навичок або компетенцій користувачів у цифровому середовищі за допомогою спеціалізованих програмних засобів [15]. Такий формат тестування набув широкого поширення завдяки розвитку інформаційних технологій, що дозволяють автоматизувати процеси оцінювання, спрощувати доступ до тестів та знижувати витрати на їх проведення.

Однією з головних переваг онлайн-тестування є його доступність. Користувачі можуть проходити тести з будь-якої точки світу, використовуючи персональні комп'ютери, планшети чи смартфони [18]. Це особливо важливо для дистанційного навчання, сертифікаційних програм, рекрутингу та корпоративного навчання.

Іншою важливою перевагою є автоматизація перевірки результатів. Більшість платформ для онлайн-тестування дозволяють налаштовувати автоматичне оцінювання відповідей, що значно зменшує витрати часу на перевірку та виключає суб'єктивний фактор у виставленні оцінок. Крім того, деякі системи використовують адаптивні алгоритми, які підбирають рівень складності запитань відповідно до рівня знань тестованого [22].

Онлайн-тестування широко застосовується у різних сферах:

- **Освіта** – навчальні заклади використовують тести для перевірки знань студентів та школярів.
- **Професійна сертифікація** – багато компаній та організацій проводять онлайн-іспити для підтвердження кваліфікації спеціалістів.
- **Рекрутинг** – тестування допомагає оцінити рівень знань і навичок кандидатів на посади.
- **Корпоративне навчання** – компанії використовують тести для оцінки ефективності тренінгів та підвищення кваліфікації працівників.

Таким чином, онлайн-тестування є ефективним інструментом для оцінювання знань та компетенцій у різних галузях. Його гнучкість, масштабованість та автоматизація роблять його незамінним у сучасному цифровому світі.

## 1.2. Аналіз сучасних підходів до онлайн-тестування

Розвиток онлайн-тестування базується на різних підходах, які враховують специфіку оцінювання знань, технологічні можливості та методики навчання. Основні сучасні підходи включають:

- **Класичне тестування** – передбачає використання закритих і відкритих запитань, на які користувачі дають відповіді в обмежений проміжок часу. Такий підхід широко застосовується у стандартизованих іспитах та сертифікаційних програмах.
- **Адаптивне тестування** – використовує алгоритми, які змінюють складність запитань залежно від відповідей користувача. Це дозволяє більш точно оцінити рівень знань тестованого та забезпечити персоналізований досвід проходження тесту.
- **Гейміфікація тестування** – включає використання ігрових механік, таких як бали, досягнення, рейтинги та нагороди. Це підвищує мотивацію користувачів і робить процес тестування більш захопливим, що особливо ефективно у навчанні дітей і корпоративних тренінгах.
- **Використання штучного інтелекту (ШІ) та аналізу даних** – сучасні системи тестування можуть застосовувати алгоритми машинного навчання для персоналізації тестових завдань, виявлення прогалин у знаннях та прогнозування успішності користувачів.
- **Проектне тестування та кейсовий підхід** – замість стандартних запитань користувачам пропонуються реальні кейси або практичні завдання, що вимагають глибокого аналізу та розв'язання проблем. Такий підхід активно використовується у бізнесі, медицині та інженерії.
- **Онлайн-прокторинг** – застосовується у високостандартних іспитах та сертифікаційних тестуваннях. Він включає використання веб-камер, моніторингу активності на екрані та аналізу поведінки користувача для запобігання шахрайству під час тестування.

Таким чином, сучасні підходи до онлайн-тестування спрямовані на підвищення точності оцінювання, персоналізацію навчання та забезпечення чесності результатів. Використання адаптивних алгоритмів, штучного інтелекту та ігрових механік

дозволяє зробити тестування більш ефективним та зручним як для користувачів, так і для організацій, які проводять оцінювання [28].

### 1.3. Огляд існуючих платформ для онлайн-тестування

На сучасному ринку представлено безліч платформ для онлайн-тестування, кожна з яких має свої особливості та переваги. Серед найбільш популярних можна виділити наступні:

- **Quizlet** – орієнтована на створення карток із запитаннями та відповідями, що дозволяє користувачам легко засвоювати матеріал. Використовується у сфері освіти для навчання та самоперевірки.
  - Переваги: проста у використанні, підтримує різні формати навчальних матеріалів.
  - Недоліки: обмежена інтерактивність тестів, відсутність складних механізмів оцінювання.

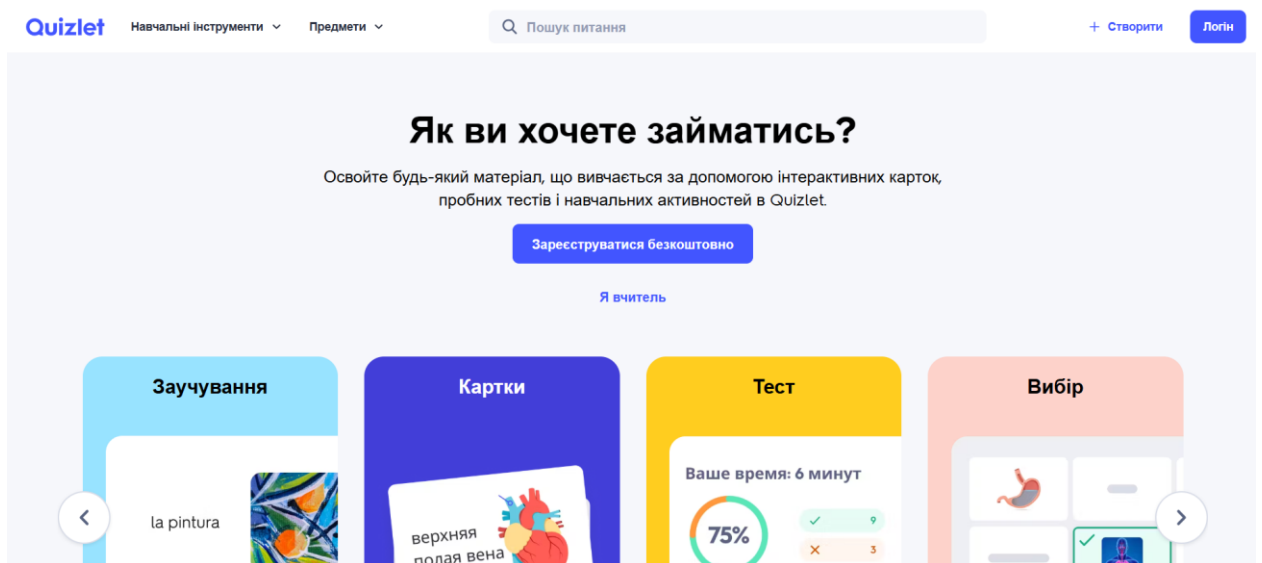


Рисунок 1.1 – Платформа Quizlet

Джерело: [14]

- **Kahoot!** – платформа, яка використовує гейміфікацію для тестування. Дозволяє проводити інтерактивні вікторини в реальному часі, що робить її популярною серед освітніх установ.
  - Переваги: інтерактивність, залучення учасників через елементи гри.
  - Недоліки: потребує синхронного проходження тестів, що не завжди зручно.

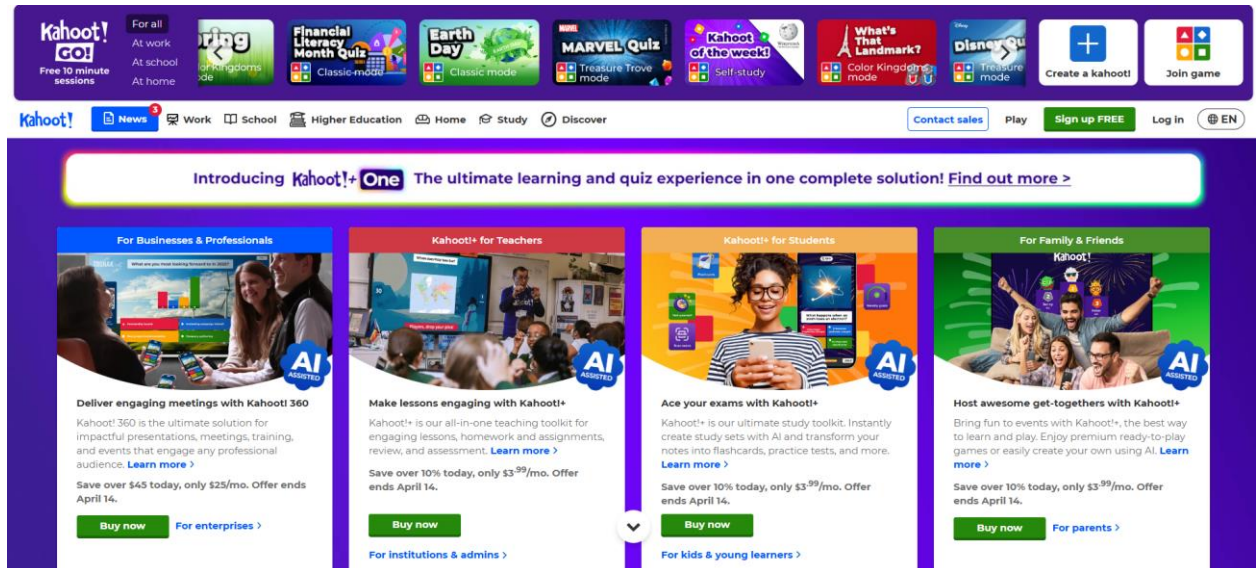


Рисунок 1.2 – Платформа Kahoot!

Джерело: [8]

- **Google Forms** – простий інструмент для створення тестів і опитувань, що інтегрується з Google Workspace. Використовується як в освіті, так і в корпоративному середовищі.
  - Переваги: безкоштовний, легко інтегрується з іншими сервісами Google.
  - Недоліки: відсутність розширених аналітичних можливостей, обмежена візуалізація тестів.

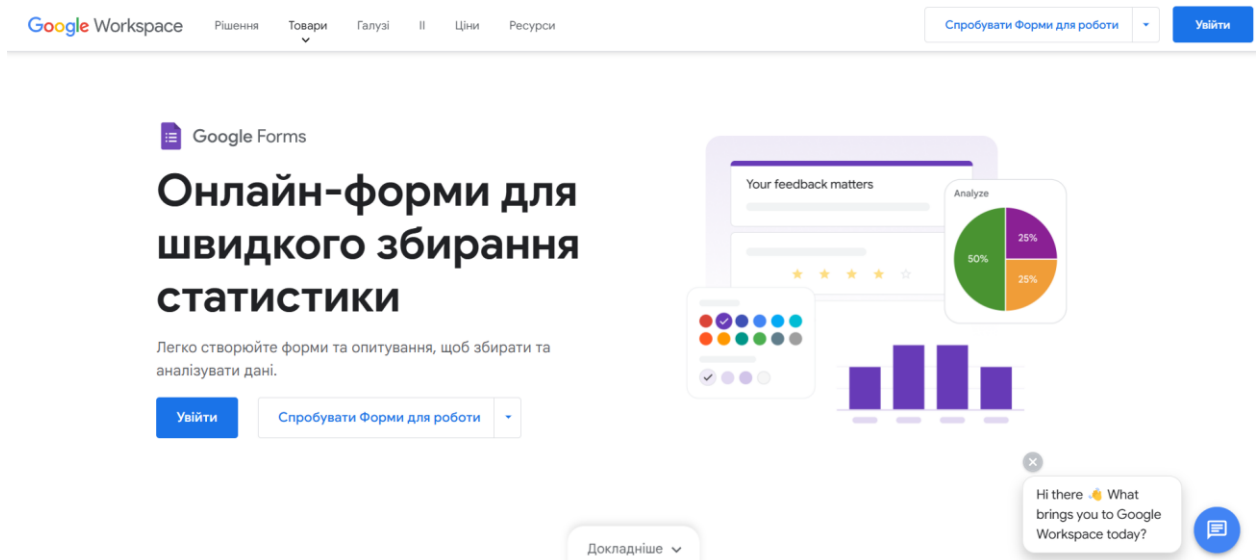


Рисунок 1.3 – Інструмент Google Forms

Джерело: [4]

- **Moodle** – потужна система управління навчанням, яка містить інструменти для створення тестів, аналізу результатів і адміністрування навчального процесу.
  - Переваги: гнучкі налаштування, підтримка складних тестових завдань.

- Недоліки: складний інтерфейс для нових користувачів, потребує серверних ресурсів.

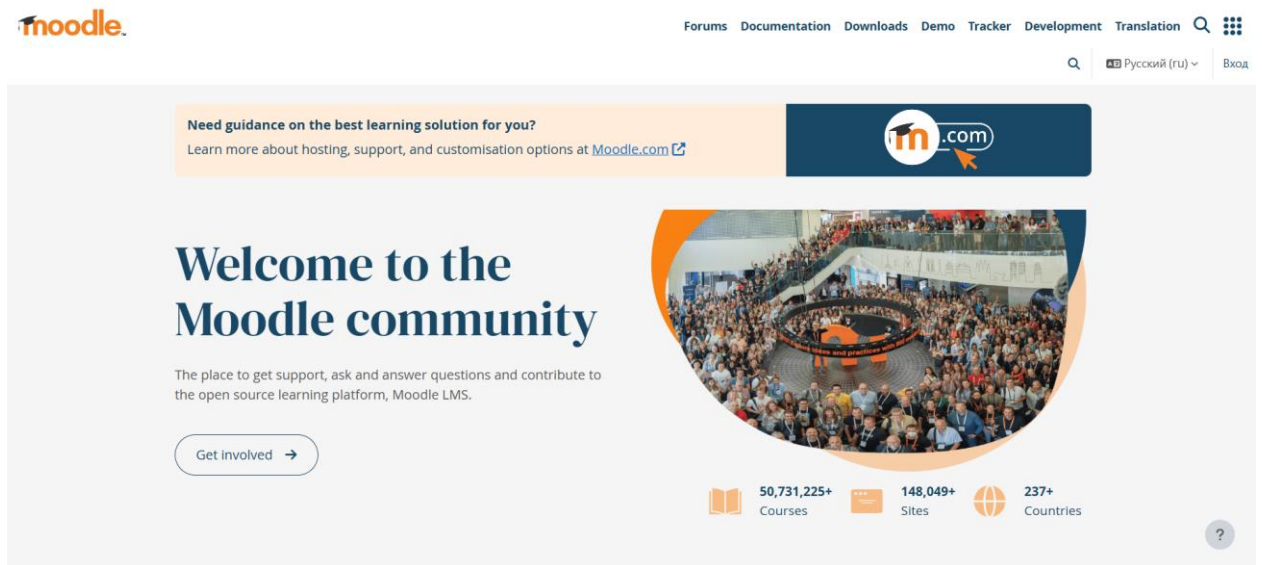


Рисунок 1.4 – Система Moodle

Джерело: [10]

- **Typeform** – платформа для створення інтерактивних опитувань і тестів із гнучкими налаштуваннями.
  - Переваги: гарний дизайн, інтерактивність, можливість кастомізації.
  - Недоліки: платна модель, менша орієнтація на освітні потреби.

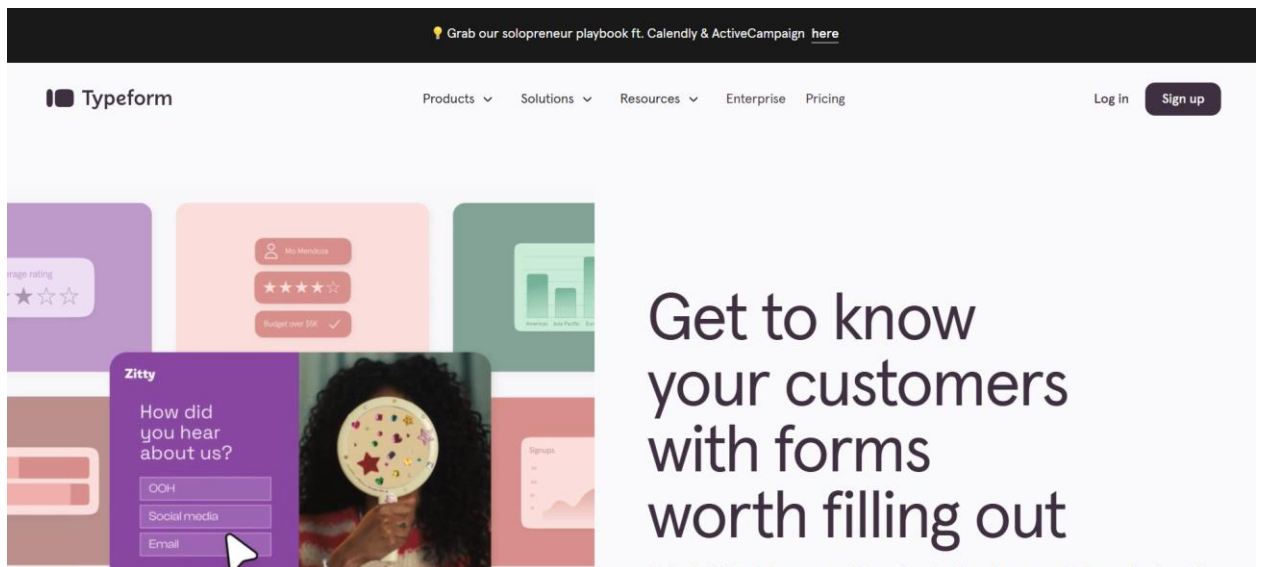


Рисунок 1.5 – Платформа Typeform

Джерело: [16]

- **ProProfs Quiz Maker** – дозволяє створювати тести з автоматичним оцінюванням та аналітикою результатів.
  - Переваги: потужні аналітичні інструменти, автоматизація перевірки.

- Недоліки: деякі функції доступні лише у платній версії.

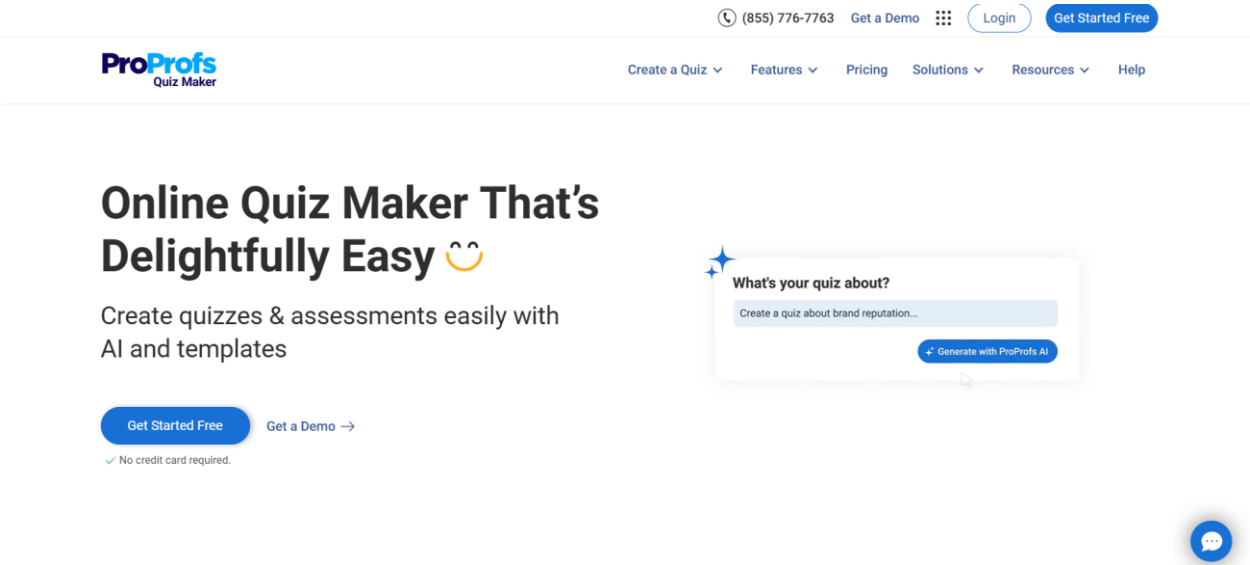


Рисунок 1.6 – Платформа ProProfs Quiz Maker

Джерело: [13]

Таблиця 1.1 – Аналіз існуючих систем

| Критерій           | Quizlet                 | Kahoot!                        | Google Forms                | Moodle                        | Typeform                    | ProProfs Quiz Maker              |
|--------------------|-------------------------|--------------------------------|-----------------------------|-------------------------------|-----------------------------|----------------------------------|
| Призначення        | Освіта, самонавчання    | Освіта, інтерактивні вікторини | Опитування, тести           | Управління навчанням          | Бізнес, опитування          | Освіта, бізнес                   |
| Типи запитань      | Картки, тестові питання | Вікторини, гейміфіковані тести | Відкриті, закриті запитання | Різноманітні тестові завдання | Відкриті та закриті питання | Тести з автоматичним оцінюванням |
| Гейміфікація       | Відсутня                | Висока                         | Відсутня                    | Низька                        | Середня                     | Низька                           |
| Аналіз результатів | Обмежений               | Обмежений                      | Мінімальний                 | Розширений                    | Обмежений                   | Розширений                       |

| Критерій                      | Quizlet                | Kahoot!                         | Google Forms   | Moodle                          | Typeform   | ProProfs Quiz Maker |
|-------------------------------|------------------------|---------------------------------|----------------|---------------------------------|------------|---------------------|
| Автоматичне оцінювання        | Так                    | Так                             | Так            | Так                             | Так        | Так                 |
| Інтеграція з іншими сервісами | Часткова               | Часткова                        | Google-сервіси | Висока                          | API-доступ | API-доступ          |
| Вартість                      | Безкоштовно            | Безкоштовно (платні розширення) | Безкоштовно    | Безкоштовно (платні розширення) | Платно     | Платно              |
| Доступність                   | Веб, мобільний додаток | Веб, мобільний додаток          | Веб            | Веб                             | Веб        | Веб                 |

*Джерело: сформовано автором на основі виконаного дослідження*

#### **1.4. Аналіз літературних джерел та практичного досвіду використання ІС і технологій**

Дослідження літературних джерел і практичного використання інформаційних систем (ІС) у сфері онлайн-тестування дозволяє визначити ключові тенденції та виклики у цій галузі. Одним із важливих аспектів є використання адаптивних алгоритмів, які дозволяють персоналізувати процес тестування відповідно до рівня знань користувачів. Це сприяє більш точному оцінюванню та підвищенню ефективності навчання [7].

Також значну увагу приділяють розвитку технологій онлайн-прокторингу, що забезпечують чесність іспитів та запобігають порушенням під час проходження тестування. Різні платформи використовують веб-камери, моніторинг активності на екрані та аналіз поведінки тестованого.

Ще одним напрямом досліджень є гейміфікація тестування, яка включає використання ігрових механік, таких як бали, рейтинги та нагороди. Це дозволяє підвищити мотивацію користувачів і зробити процес оцінювання більш цікавим.

Автоматизовані системи оцінювання також відіграють важливу роль, оскільки знижують навантаження на викладачів та забезпечують об'єктивність оцінювання. Завдяки штучному інтелекту можна аналізувати відповіді учасників і навіть надавати рекомендації щодо подальшого навчання.

Інтеграція онлайн-тестування із системами управління навчанням (LMS) є ще одним ключовим аспектом, який дозволяє ефективно поєднувати тестування з іншими освітніми процесами. Це сприяє створенню єдиного цифрового середовища для навчання та оцінювання.

Практичний досвід компаній та освітніх установ підтверджує, що успішне впровадження онлайн-тестування залежить від зручності інтерфейсу, рівня аналітики та можливості інтеграції з іншими сервісами [27]. Важливою залишається проблема захисту даних та ідентифікації користувачів під час проходження тестів.

Аналіз сучасних платформ та підходів до онлайн-тестування показав, що найважливішими аспектами для розробки є автоматизація оцінювання, адаптивність тестових завдань та аналітика результатів. У рамках бакалаврської роботи буде зроблено акцент на розробці системи автоматичного оцінювання відповідей користувачів. Це дозволить персоналізувати процес проходження тестів та підвищити точність оцінювання знань.

## РОЗДІЛ 2 РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Аналіз і специфікація вимог до інформаційної системи

#### 2.1.1 Бізнес-вимоги до системи

Бізнес-вимоги визначають цілі та задачі, які повинна виконувати інформаційна система для підтримки навчально-освітнього центру розвитку. Вони формують основу для подальшого технічної розробки системи. Для розробленої системи, було визначено основних стейкхолдерів, а саме: автор тестів, учасник тестування.

Під час визначення бізнес вимог, варто виокремити основні задачі:

- Створити веб-платформу, яка дозволяє користувачам проходити та перевіряти онлайн-тести.
- Підвищити ефективність і зручність навчального процесу через інтерактивне тестування.
- Забезпечити підтримку різних типів питань (один правильний варіант, множинний вибір).

Розглянемо бізнес вимоги кожного стейкхолдера до проектованої інформаційної системи.

До бізнес-цілей користувача, який є автором тестів варто віднести:

#### 1. Створення якісного навчального контенту

Автор має можливість створювати тематичні тести з варіативними типами питань для формування знань або перевірки рівня підготовки.

#### 2. Організація та структурування тестів

Автор може групувати тести за предметами, темами або курсами для зручного використання та поширення.

#### 3. Поширення знань серед користувачів

Завдяки публікації тестів, автор ділиться знаннями з іншими учасниками платформи.

#### 4. Аналіз результатів проходження

Автор може переглядати статистику проходження тестів користувачами для оцінки складності питань та якості навчального матеріалу.

## **5. Розвиток особистого бренду або спільноти**

За допомогою відкритих груп тестів автор може залучати підписників або створювати власну навчальну спільноту.

До бізнес-вимог користувача, який є автором тестів варто віднести:

1. Можливість створювати тести з різними типами питань (одна правильна відповідь, множинний вибір)

### **2. Формування груп тестів**

Автор може об'єднувати тести у групи або курси для структурованого подання навчального матеріалу.

### **3. Редагування та оновлення тестів**

Автор має змогу змінювати питання, відповіді, налаштовувати час проходження, кількість спроб тощо.

### **4. Публікація груп тестів у відкритий або приватний доступ**

Автор самостійно визначає, хто зможе приєднатись до групи з тестами.

### **5. Статистика та результати проходження**

Система повинна надавати автору доступ до перегляду результатів користувачів, які пройшли його тести.

### **6. Захист від несанкціонованого редагування**

Тільки автор має повний контроль над тестами, які створив. Інші користувачі не можуть вносити зміни.

До бізнес-цілей користувача, який є учасником тестування варто віднести:

### **1. Перевірка власних знань**

Користувач проходить тести, щоб оцінити рівень засвоєння теми чи предмета.

### **2. Підготовка до контрольних, іспитів або сертифікацій**

Тести можуть використовуватись як форма підготовки до реальних перевірок знань.

### **3. Отримання зворотного зв'язку та аналітики**

Учасник отримує оцінку, бачить правильні відповіді, пояснення та динаміку результатів.

### **4. Гнучке та самостійне навчання**

Учасник може проходити тести у зручний час, з будь-якого пристрою.

## 5. Участь у навчальних групах або спільнотах

Користувач може приєднуватися до груп тестів або курсів за інтересами чи навчальною програмою.

До бізнес-вимог користувача, який є учасником тестування варто віднести:

### 1. Можливість переглядати доступні тести та групи тестів

Пошук і фільтрація за темою, автором, складністю, датою.

### 2. Отримання результату після завершення тесту

Кількість правильних відповідей, оцінка у %, пояснення до запитань (якщо надано).

### 3. Перегляд історії проходжень і результатів

Статистика за окремими тестами або за періодами.

### 4. Захист результатів користувача

Ніхто, окрім користувача (і, за потреби, автора тесту), не має доступу до його результатів без дозволу.

## 2.1.2 Функціональні вимоги до системи

Функціональні вимоги, які були визначені для розробленої інформаційної системи, її складові представлені на рис.2.1, опис функціональних вимог у таблиці 2.1.

Таблиця 2.1 – Функціональні вимоги до системи

| Функціональна вимога    | Опис функціональності                                          |
|-------------------------|----------------------------------------------------------------|
| Реєстрація користувача  | Користувач має можливість створити обліковий запис через email |
| Авторизація користувача | Система забезпечує вхід у систему через логін і пароль         |
| Відновлення пароля      | Можливість скидання пароля через email                         |
| Редагування профілю     | Користувач може змінювати особисті дані, фото, мову тощо       |
| Створення груп          | Користувач може створити групу для розташування в ній тестів   |

| <b>Функціональна вимога</b>      | <b>Опис функціональності</b>                                  |
|----------------------------------|---------------------------------------------------------------|
| Налаштування доступу до груп     | Автор визначає рівень доступу: публічний, приватний           |
| Створення тесту                  | Автор може створювати тести з різними типами запитань         |
| Редагування/видалення тесту      | Автор має змогу редагувати та видаляти створені ним тести     |
| Проходження тесту                | Користувач проходить тест в інтерактивному інтерфейсі         |
| Перегляд результатів тестування  | Користувач отримує оцінку після проходження тесту             |
| Аналітика результатів для автора | Автор тестів бачить зведену статистику проходжень його тестів |
| Пошук та фільтрація тестів       | Можливість знаходити тести за темою, автором, складністю      |

*Джерело: сформовано автором*

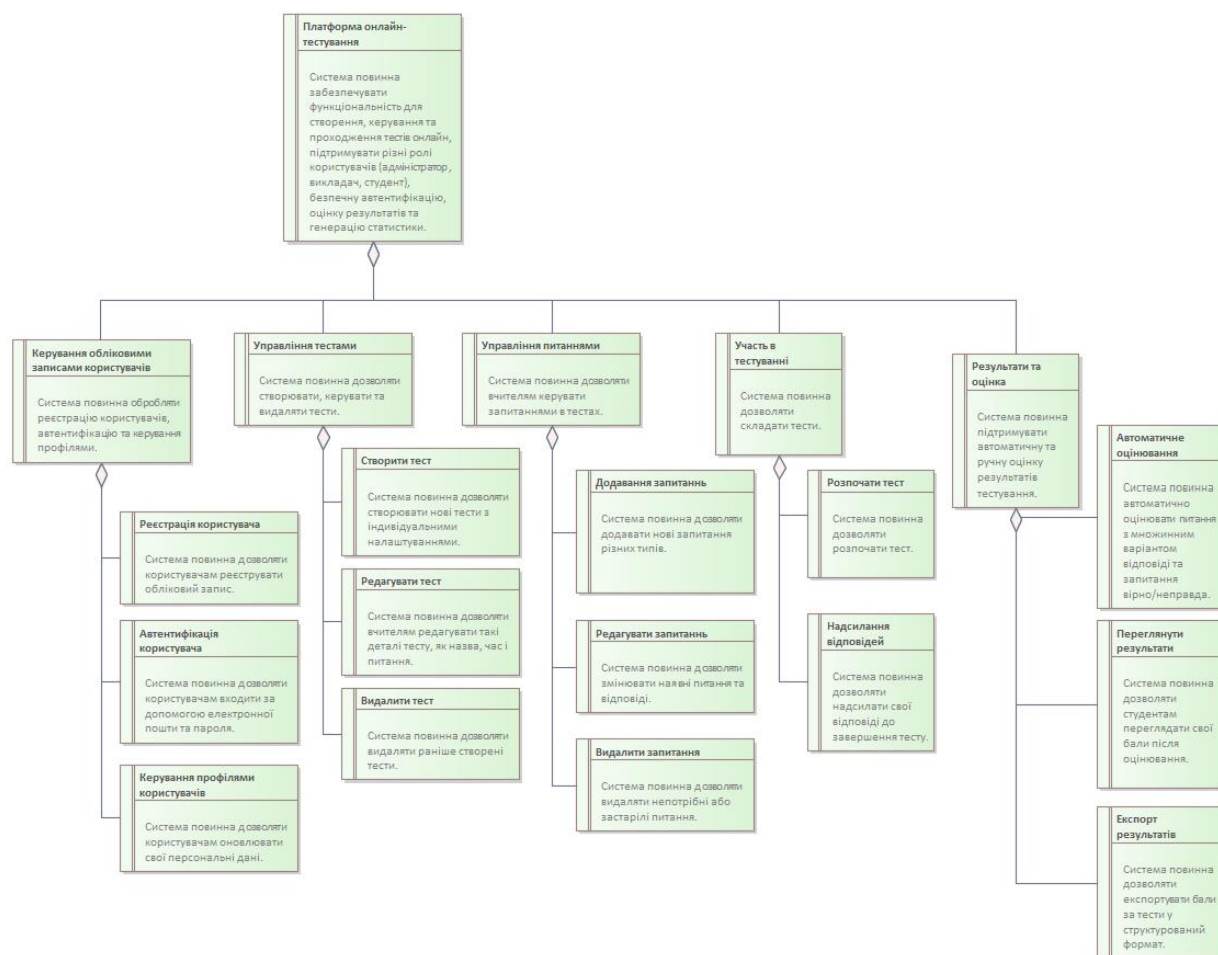


Рисунок 2.1 – Функціональні вимоги до розробленої інформаційної системи  
Джерело: розроблено самостійно

### 2.1.3 Нефункціональні вимоги до системи

Нефункціональні вимоги орієнтовані на забезпечення стабільної та безперебійної роботи платформи, збереження конфіденційності користувачів та ефективне використання ресурсів. У елементах нефункціональних вимог було виділено наступні елементи:

#### Продуктивність

- Система повинна забезпечувати швидке завантаження інтерфейсу — не більше 2 секунд при середньому навантаженні.
- Час відповіді сервера на основні запити (авторизація, проходження тесту, збереження результату) не повинен перевищувати 1 секунди.
- Система повинна підтримувати щонайменше 100 одночасних користувачів без втрати продуктивності.

#### Безпека

- Всі паролі повинні зберігатися у зашифрованому вигляді (хешування + сіль).
- Доступ до персональних даних повинен бути обмежений лише авторизованими користувачами.
- Тестові результати мають бути доступні тільки автору тесту (для аналітики) та учаснику (особисто).
- Захист від SQL-ін'єкцій, XSS та CSRF-атак повинен бути реалізований.

#### Надійність

- Усі дані користувачів та результати тестування мають зберігатися у надійній СУБД з регулярним резервним копіюванням.
- Система має автоматично обробляти нештатні ситуації (обрив з'єднання, повторне відправлення відповідей).

#### Масштабованість і підтримка

- Архітектура платформи повинна забезпечувати легке додавання нових функцій (наприклад, нові типи запитань або модулів аналітики).
- Код програми має бути задокументованим та підтримуваним для подальшого розширення роботи іншими розробниками.

#### Локалізація

- Інтерфейс платформи повинен підтримувати українську мову. У перспективі — можливість додати інші мови.

Оболонка portalу повинна підтримувати роботу на різноманітних пристроях, включаючи персональні комп'ютери, ноутбуки, планшети та смартфони. Інтерфейс системи повинен бути доступний через веб-браузер, що дозволить користувачам працювати з порталом без необхідності встановлення додаткового програмного забезпечення. Для забезпечення робочої функціональності, сервіс має бути сумісний із сучасними веб-браузерами, такими як Google Chrome, Edge та Opera тощо. Це забезпечить універсальність доступу та стабільну роботу незалежно від платформи або операційної системи пристрою.

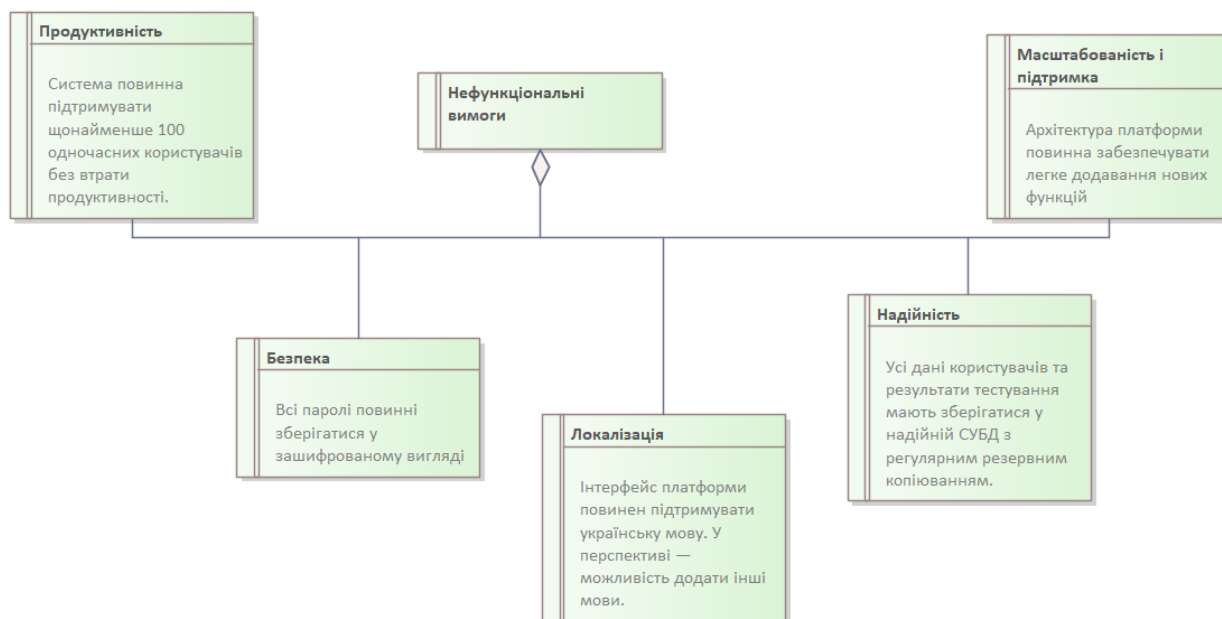


Рисунок 2.2 – Нефункціональні вимоги до розробленої інформаційної системи  
Джерело: розроблено самостійно

## 2.2 Постановка та алгоритм розв’язання задачі

### 2.2.1 Характеристика задачі

Онлайн-платформи для тестування є важливим інструментом у сфері освіти, корпоративного навчання та самоперевірки. З огляду на зростання попиту на дистанційне навчання та гнучкі форми перевірки знань, виникає потреба в розробці інтуїтивно зрозумілої, функціональної та надійної системи для створення й проходження тестів у веб-середовищі.

Основною задачею даного дипломної роботи є створення інформаційної системи, яка дозволить:

- користувачам створювати тести з різними типами запитань;
- об’єднувати тести у тематичні групи;
- поширювати тести для проходження іншими користувачами;
- проходити тести з отриманням результатів у реальному часі;
- переглядати аналітику відповідей (для авторів) та історію тестування (для учасників).

Платформа повинна мати адаптивний веб-інтерфейс, підтримувати обробку результатів, зберігати історію дій, забезпечувати базовий рівень безпеки даних та дозволяти масштабування функціональності у майбутньому.

Розробка такої системи передбачає реалізацію наступних підзадач:

- розробка логіки реєстрації, аутентифікації та управління користувачами;
- проектування та реалізація механізму створення тестів і груп;
- реалізація модуля проходження тестів та обробки відповідей;
- створення інтерфейсу перегляду результатів та аналітики;
- забезпечення захисту персональних даних користувачів;
- створення зручного та сучасного інтерфейсу.

Розв’язання задачі потребує використання інструментів веб-розробки, зокрема фреймворків для front-end та back-end, систем управління базами даних, а також сучасних підходів до моделювання інформаційних систем.

### 2.2.2 Вихідна інформація

Перелік і опис вихідних повідомлень показаний в таблиці 2.2

Таблиця 2.2 – Перелік та опис вихідних повідомлень

| Назва вихідного повідомлення  | Ідентифікатор | Форма подання   | Періодичність та видання | Термін видання і допустимий час затримки | Користувачі інформації      |
|-------------------------------|---------------|-----------------|--------------------------|------------------------------------------|-----------------------------|
| Результати тестування         | result_test   | Екран           | Після завершення тесту   | Миттєво                                  | Учасник                     |
| Статистика проходження тестів | stat_author   | Екран, документ | За вимогою               | За вимогою                               | Автор тесту                 |
| Історія проходження тестів    | history_user  | Екран           | За вимогою               | За вимогою                               | Учасник                     |
| Загальна аналітика системи    | sys_analytics | Екран, документ | За вимогою               | За вимогою                               | Адміністратор, автор тестів |

*Джерело: розроблено самостійно*

### 2.2.3 Вхідна інформація

Перелік і опис вхідних повідомлень показаний в таблиці 2.3

Таблиця 2.3 – Перелік та опис вхідних повідомлень

| Назва вхідного повідомлення          | Ідентифікатор     | Форма подання | Періодичність подання     | Термін подання та допустимий час затримки | Джерело інформації |
|--------------------------------------|-------------------|---------------|---------------------------|-------------------------------------------|--------------------|
| Дані для створення облікового запису | user_reg_data     | Екран         | Один раз при реєстрації   | Після підтвердження email                 | Користувач         |
| Вхід у систему                       | user_login        | Екран         | Кожного входу             | Миттєво                                   | Користувач         |
| Відповіді на тестові запитання       | test_answers      | Екран         | Під час проходження тесту | Миттєво                                   | Користувач         |
| Дані для створення тесту             | test_create_data  | Екран         | За вимогою                | Миттєво                                   | Автор тесту        |
| Дані для створення групи             | group_create_data | Екран         | За вимогою                | Миттєво                                   | Автор тесту        |
| Зміна профілю                        | profile_update    | Екран         | За потреби                | Миттєво                                   | Користувач         |

*Джерело: розроблено самостійно*

## 2.2.4 Математичне забезпечення та алгоритм функціонування системи

### Використовувана інформація

Перелік масивів використаної інформації показаний в таблиці 2.4.

Таблиця 2.4 – Перелік масивів використаної інформації

| Назва масиву           | Ідентифікатор | Максимальна кількість записів |
|------------------------|---------------|-------------------------------|
| Довідник користувачів  | users         | 100 000                       |
| Тести                  | tests         | 50 000                        |
| Питання у тестах       | questions     | 500 000                       |
| Варіанти відповідей    | answers       | 2 000 000                     |
| Групи тестів           | groups        | 10 000                        |
| Результати проходження | results       | 1 000 000                     |
| Історія відповідей     | user_answers  | 5 000 000                     |

*Джерело: розроблено самостійно*

## Результати розв'язання

Результатом виконання задачі є:

- проходження тестів користувачами;
- збереження та аналіз результатів;
- побудова статистики;
- відображення інтерфейсу з оцінкою та відгуком.

### Математичний опис

Вхідні дані:

$Q = \{q_1, q_2, \dots, q_n\}$  – множина запитань тесту;

$A_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$  – множина варіантів відповіді для запитання  $q_i$ ;

$R = \{r_1, r_2, \dots, r_n\}$ , де  $r_i \subseteq A_i$  – правильні відповіді;

$U = \{u_1, u_2, \dots, u_n\}$ , де  $u_i \subseteq A_i$  – відповіді користувача;

Визначимо функцію правильності відповіді:

$$f_i = \begin{cases} 1, & \text{якщо } u_i = r_i \\ 0, & \text{інакше} \end{cases} \quad (2.1)$$

Загальна кількість правильних відповідей:

$$S = \sum_{i=1}^n f_i \quad (2.2)$$

Максимальна кількість балів:

$$B_{max} = n \quad (2.3)$$

Відсоткова оцінка:

$$P = \frac{S}{B_{max}} * 100\% \quad (2.4)$$

### Додаткові розрахунки

Статистика по всіх користувачах для одного тесту (наприклад, середній бал):

Нехай  $P_1, P_2, \dots, P_k$  – відсоткові результати  $k$  користувачів, тоді:

$$\bar{P} = \frac{1}{k} \sum_{j=1}^k P_j \quad (2.5)$$

Алгоритм функціонування системи

#### 1. Реєстрація користувача

- Користувач вводить email, пароль та інші дані.
- Система створює запис у масиві users.

#### 2. Аутентифікація

- Користувач входить у систему.
  - Проводиться перевірка відповідності email/пароля у базі даних.
3. Створення групи
- Створюється запис у масиві groups.
  - Тести додаються до цієї групи.
4. Створення тесту (Автор тесту)
- Автор створює новий тест у масиві tests.
  - Додає питання до тесту (questions) та варіанти відповідей (answers).
5. Проходження тесту (Учасник тестування)
- Користувач обирає тест із каталогу або з групи.
  - Питання подаються по черзі з варіантами.
  - Відповіді зберігаються у user\_answers.
6. Підрахунок результату
- Після завершення тесту система підраховує кількість правильних відповідей.
  - Формується оцінка та зберігається у results.
7. Аналіз результатів (для автора тесту)
- Автор переглядає загальну статистику: середній бал, кількість спроб, найскладніші питання.
8. Відображення результатів (для користувача)
- Учасник тестування бачить кількість правильних відповідей.

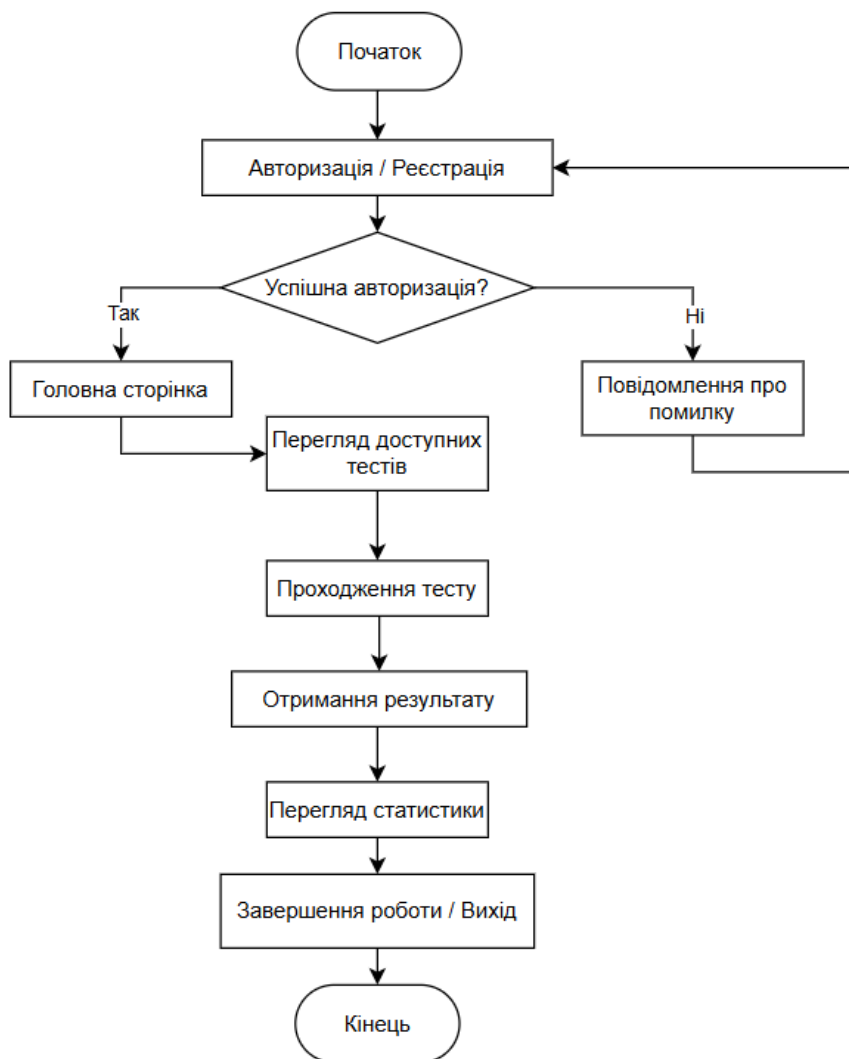


Рисунок 2.3 – Алгоритм роботи з системою  
*Джерело: розроблено самостійно*

## 2.3 Моделювання інформаційної системи

### 2.3.1 Моделювання користувацького інтерфейсу

Прототипування фреймів програми відбувається за допомогою онлайнпрограми Figma (векторний графічний редактор, в якому можна створювати макети, елементи інтерфейсу, зображення, ілюстрації, прототипування, проектування та багато інших креативів та полотен). Було створено клікабельний прототип сайту для онлайн тестування за допомогою інструменту Figma.

Головна сторінка. Вона є стартовою точкою взаємодії користувача з платформою онлайн-тестування. Вона реалізована з дотриманням принципів простоти, доступності та фокусування уваги на ключовому заклику до дії. Містить навігацію по сайту, та пропонує увійти на сайт зареєстрованим користувачам, а якщо користувач незареєстрований – виконати процес реєстрації.

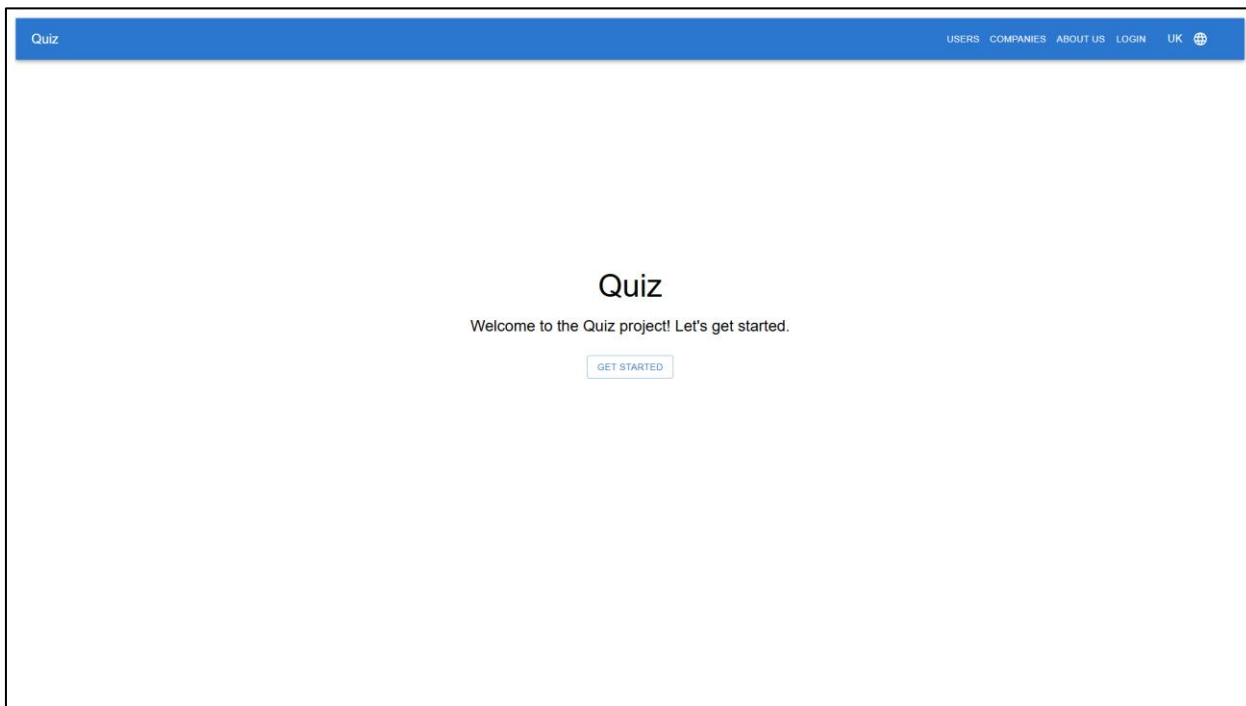


Рисунок 2.4 – Прототип інтерфейсу головної сторінки  
*Джерело: розроблено самостійно за допомогою Figma [2]*

Сторінка реєстрації. Вона є частиною модуля аутентифікації та забезпечує первинний доступ користувача до системи. Реєстрація необхідна для створення облікового запису, який надає можливість користувачеві створювати тести, проходити їх, переглядати результати та зберігати прогрес.

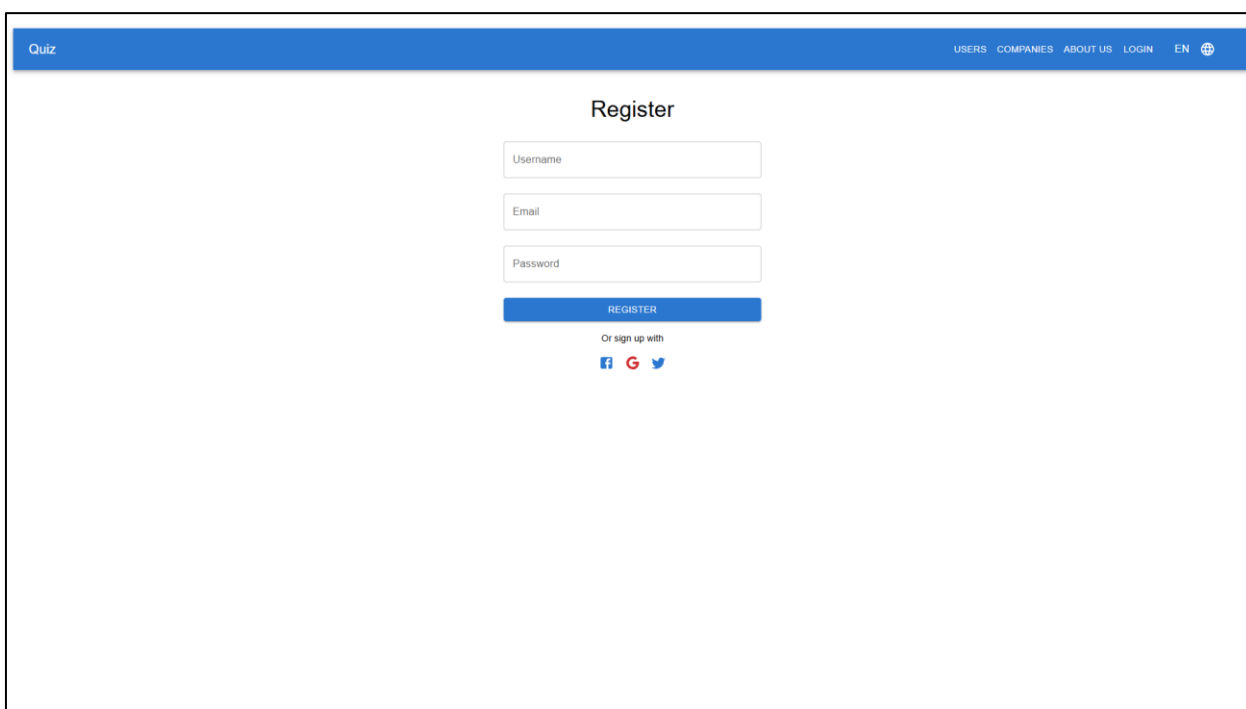


Рисунок 2.5 – Прототип інтерфейсу сторінки реєстрації  
*Джерело: розроблено самостійно за допомогою Figma [2]*

Сторінка входу. Призначена для надання зареєстрованим користувачам доступу до особистого кабінету, створених тестів, результатів проходження та інших персоналізованих функцій системи.

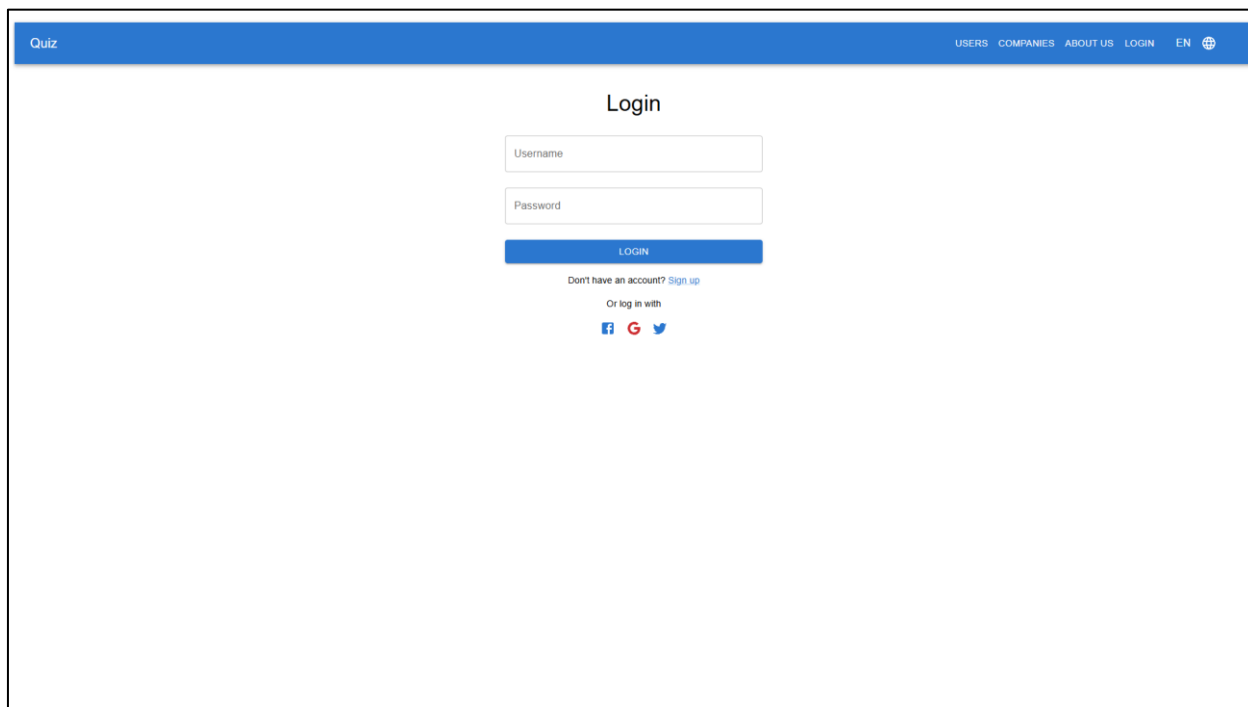
A wireframe of a login page. At the top is a blue header bar with the word 'Quiz' on the left and a navigation menu on the right containing 'USERS', 'COMPANIES', 'ABOUT US', 'LOGIN', 'EN', and a globe icon. The main content area is white and centered. It features the title 'Login' at the top. Below the title are two input fields: 'Username' and 'Password'. Under these fields is a blue button with the text 'LOGIN'. Below the button is a link that says 'Don't have an account? Sign up'. Further down is the text 'Or log in with' followed by three social media icons: Facebook, Google+, and Twitter.

Рисунок 2.6 – Прототип інтерфейсу сторінки входу  
*Джерело: розроблено самостійно за допомогою Figma [2]*

Профіль користувача. Сторінка профілю користувача є центральним елементом персоналізованого інтерфейсу системи онлайн-тестування. Вона містить основну інформацію про користувача, надає доступ до створених ним тестів, груп, а також історії проходжень. Залежно від ролі користувача (автор тестів або учасник), набір доступного функціоналу може змінюватися.

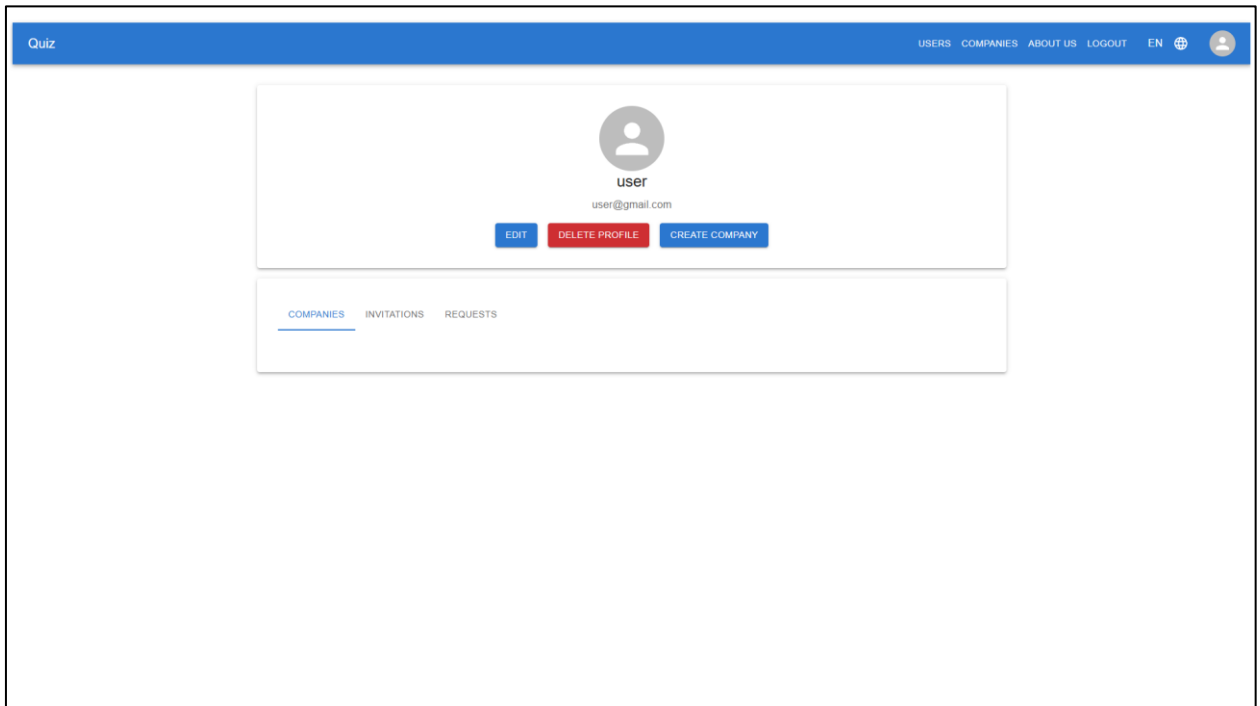


Рисунок 2.7 – Прототип інтерфейсу профілю користувача  
*Джерело: розроблено самостійно за допомогою Figma [2]*

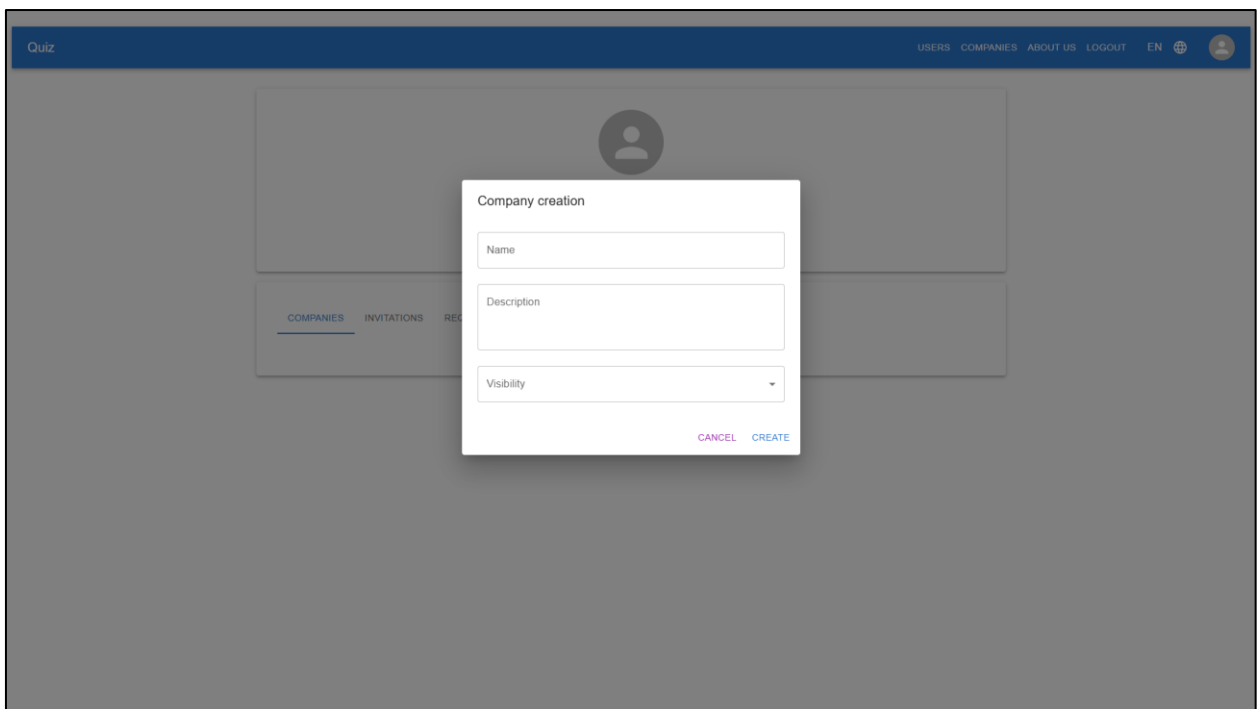


Рисунок 2.8 – Форма створення групи  
*Джерело: розроблено самостійно за допомогою Figma [2]*

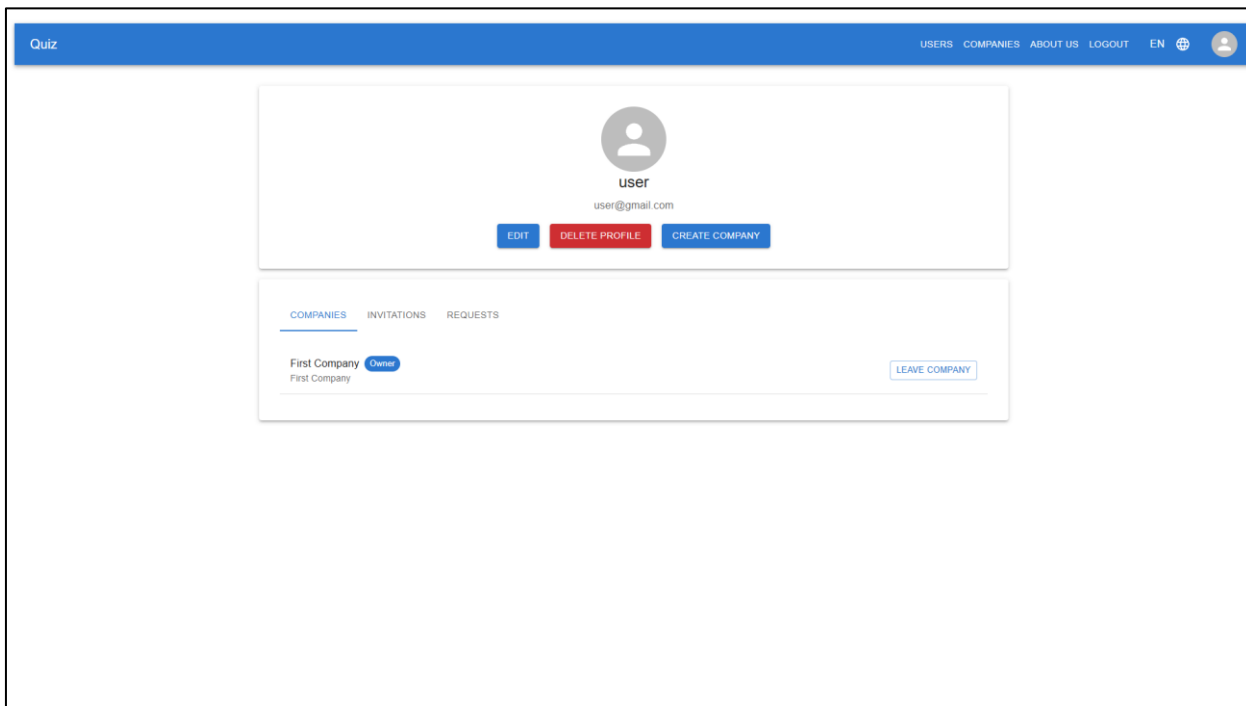


Рисунок 2.9 – Профіль користувача з створеною групою  
*Джерело: розроблено самостійно за допомогою Figma [2]*

Профіль групи. Сторінка профілю групи тестів є логічним центром об'єднання контенту на платформі онлайн-тестування. Вона дозволяє автору групи (та, за потреби, учасникам) управляти колекцією тестів, налаштовувати доступ до них, взаємодіяти з учасниками й переглядати статистику використання.

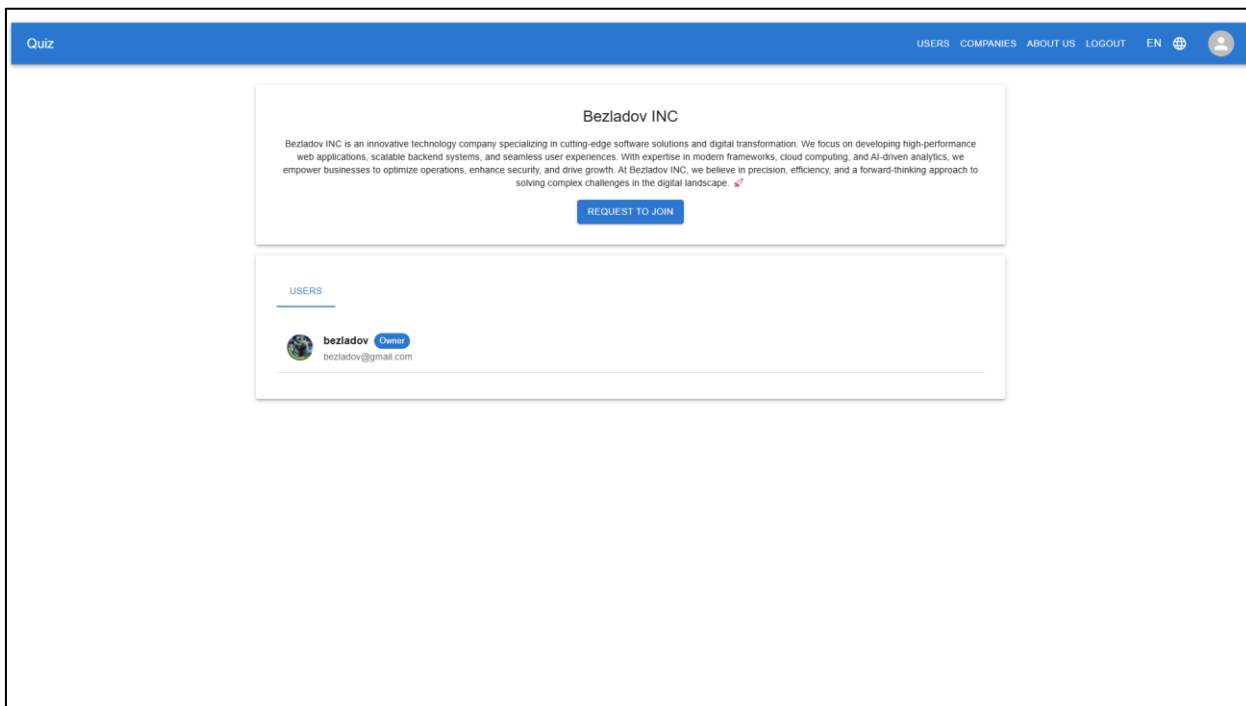


Рисунок 2.10 – Профіль групи  
*Джерело: розроблено самостійно за допомогою Figma [2]*

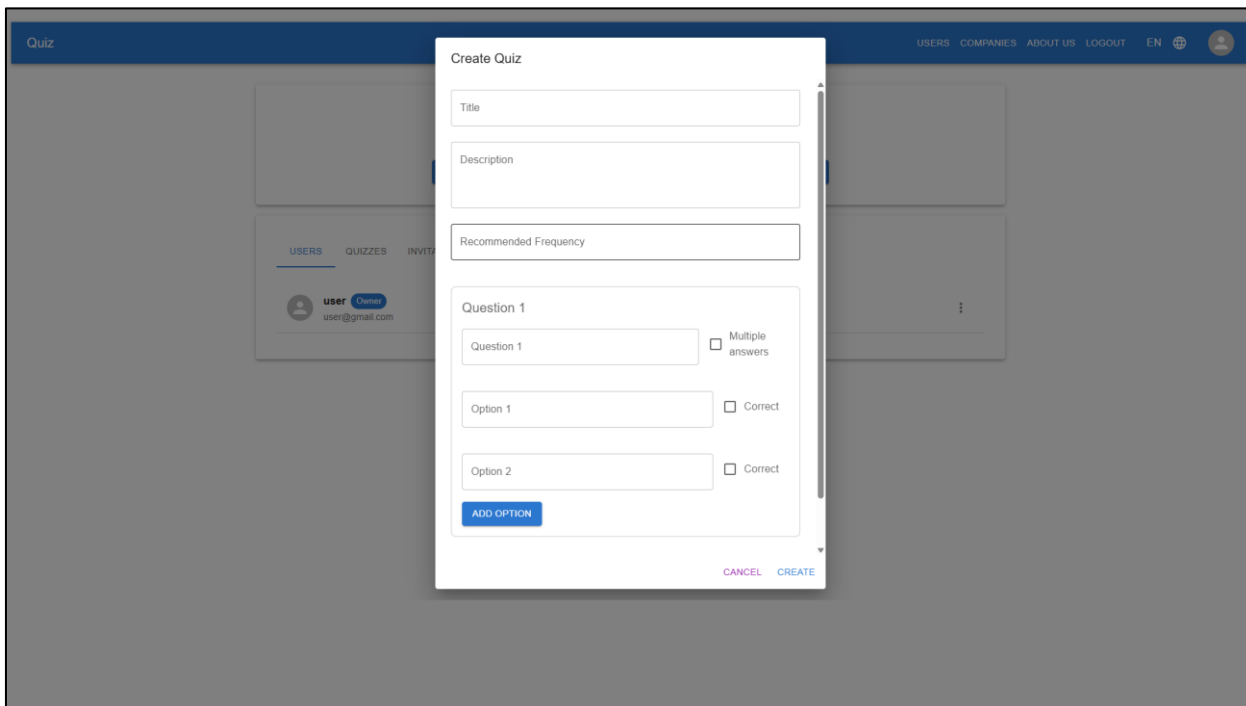


Рисунок 2.11 – Прототип інтерфейсу створення тесту  
Джерело: розроблено самостійно за допомогою Figma [2]

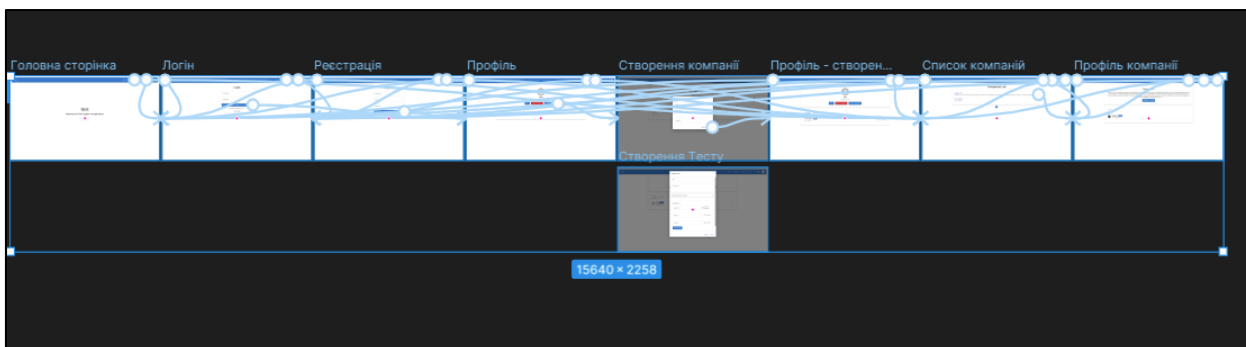


Рисунок 2.12 – Ілюстрація навігації по веб-сайту  
Джерело: розроблено самостійно за допомогою Figma [2]

Розроблений прототип інтерфейсу платформи онлайн-тестування відповідає основним принципам зручності, простоти та доступності. Користувацькі сценарії логічно структуровані, а візуальні елементи забезпечують інтуїтивну навігацію. Може бути використаний для подальшого доопрацювання і розробки повноцінного сайту.

### 2.3.2 Моделювання поведінки системи

Було розроблено діаграму прецедентів (Use Case diagram), яка відображає основні сценарії взаємодії користувачів із системою онлайн-тестування. Діаграма демонструє типові дії, які можуть виконуватись у межах системи.

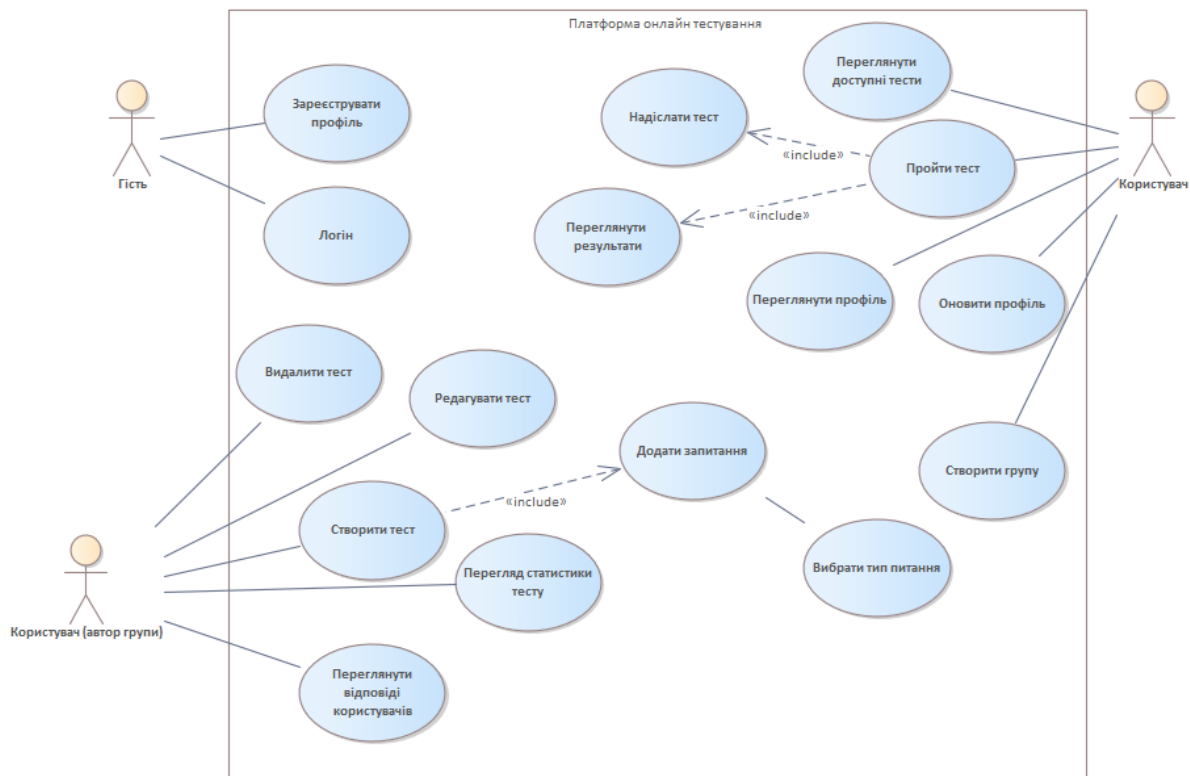


Рисунок 2.13 – Діаграма Use Case

*Джерело: розроблено самостійно*

### Створення Діаграм послідовності

Було побудовано діаграма послідовності, які відображають процес авторизації, реєстрації, створення групи, створення та проходження тесту в даній системі (рисунки 2.17 – 2.21).

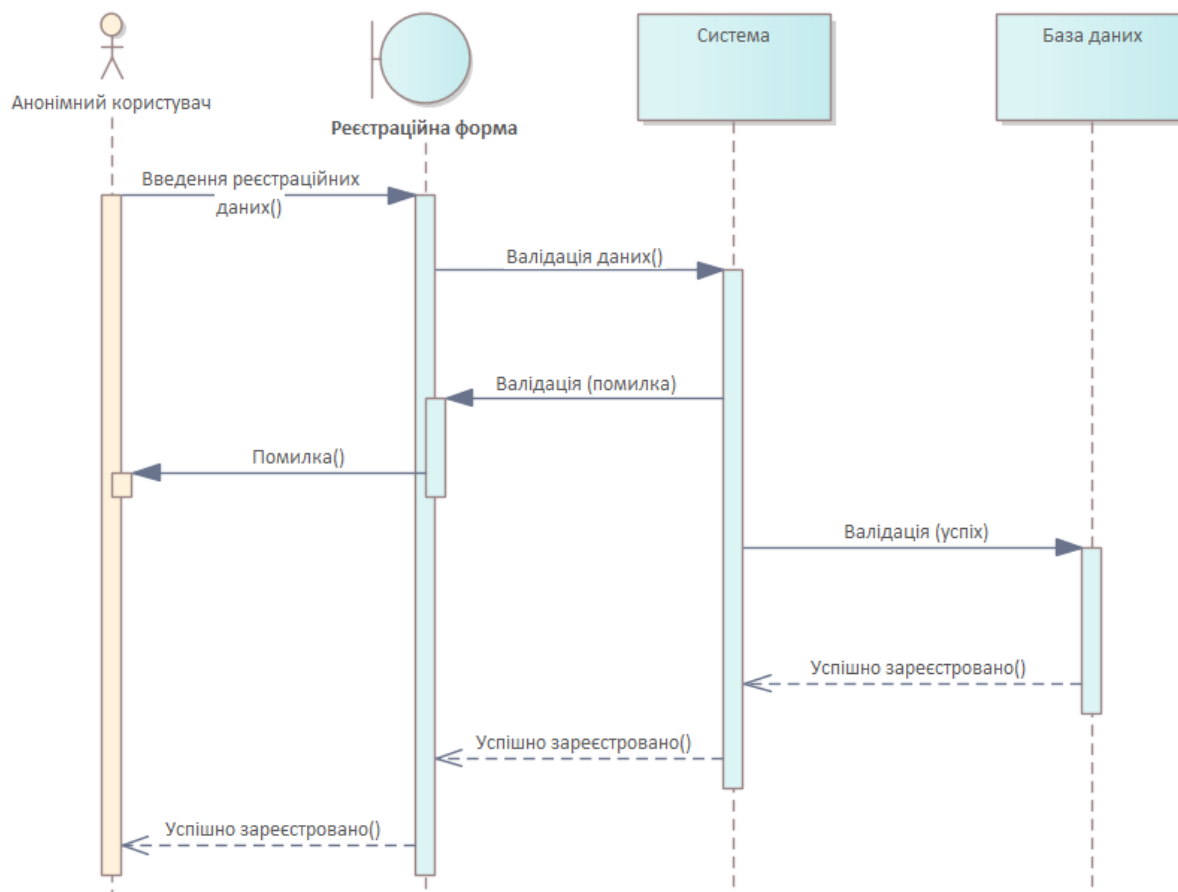


Рисунок 2.14 – Діаграма послідовності реєстрації

Джерело: розроблено самостійно

1. Анонімний користувач заповнює форму — *Введення реєстраційних даних()*.
2. Реєстраційна форма передає ці дані системі для перевірки — *Валідація даних()*.
3. Далі можливі два сценарії:
  - Якщо дані некоректні:
    - Система повертає результат з помилкою — *Валідація (помилка)*.
    - Реєстраційна форма повідомляє користувача — *Помилка()*.
  - Якщо дані коректні:
    - Система надсилає їх до бази даних — *Валідація (успіх)*.
    - База даних реєструє нового користувача — *Успішно зареєстровано()*.
    - Система інформує форму про успіх — *Успішно зареєстровано()*.
    - Реєстраційна форма сповіщає користувача про успішну реєстрацію — *Успішно зареєстровано()*.

Цей процес ілюструє умовну логіку валідації, обробку помилок і успішну реєстрацію нового користувача.

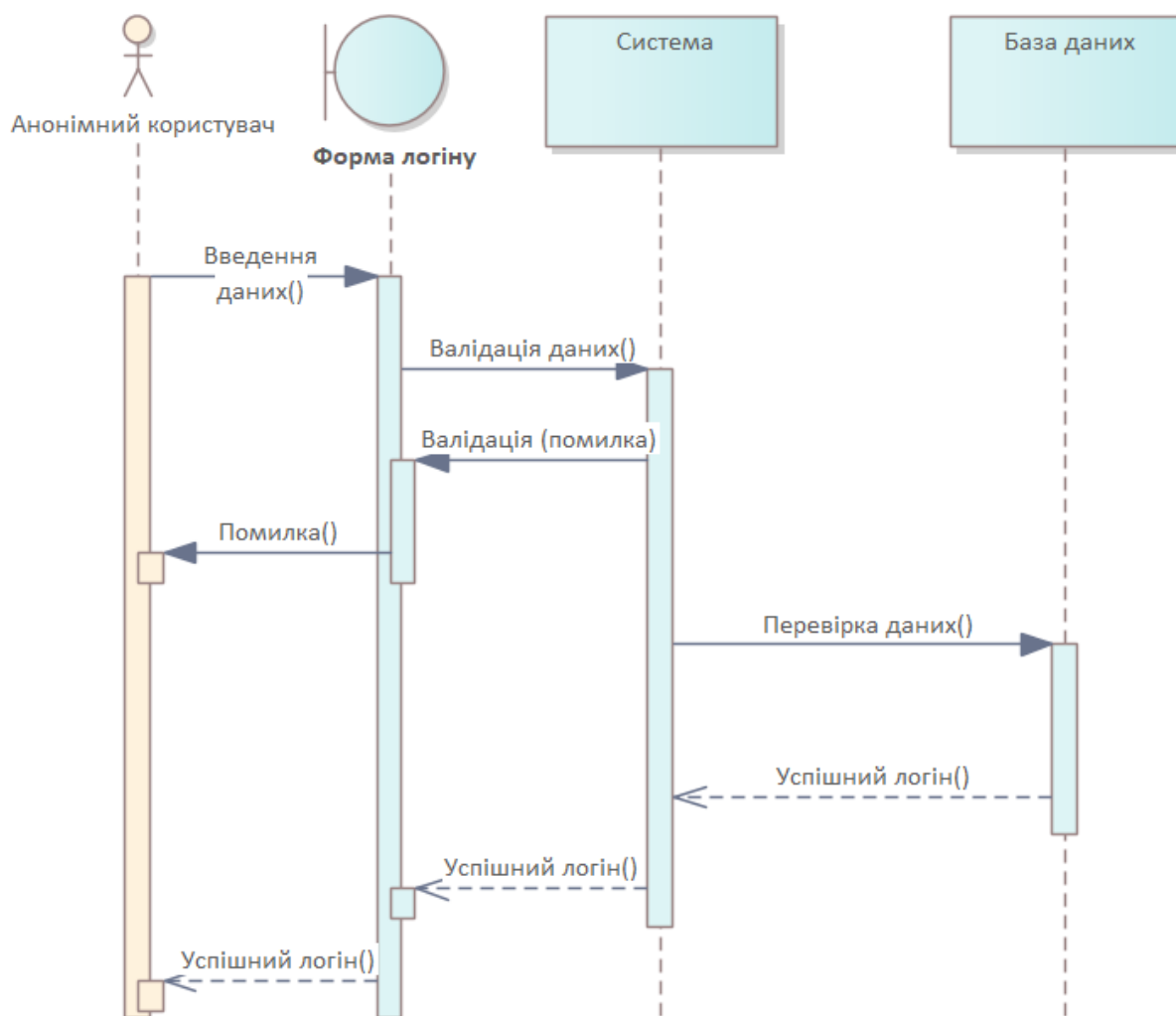


Рисунок 2.15 – Діаграма послідовності авторизації

Джерело: розроблено самостійно

1. Анонімний користувач вводить свої дані у форму логіну – *Введення даних()*.
2. Форма логіну надсилає ці дані в систему для перевірки – *Валідація даних()*.
3. Можливі два сценарії:
  - Якщо введені дані некоректні (наприклад, відсутній email чи пароль не відповідає вимогам):
    - Система надсилає відповідь з помилкою – *Валідація (помилка)*.
    - Форма повідомляє користувача – *Помилка()*.
  - Якщо дані валідні:
    - Система виконує запит до бази даних – *Перевірка даних()*.
    - Якщо облікові дані коректні (користувач існує, пароль правильний):
      - База даних повертає підтвердження – *Успішний логін()*.
      - Система пересилає результат формі логіну – *Успішний логін()*.

- Форма повідомляє користувача про успішний вхід – *Успішний логін()*.

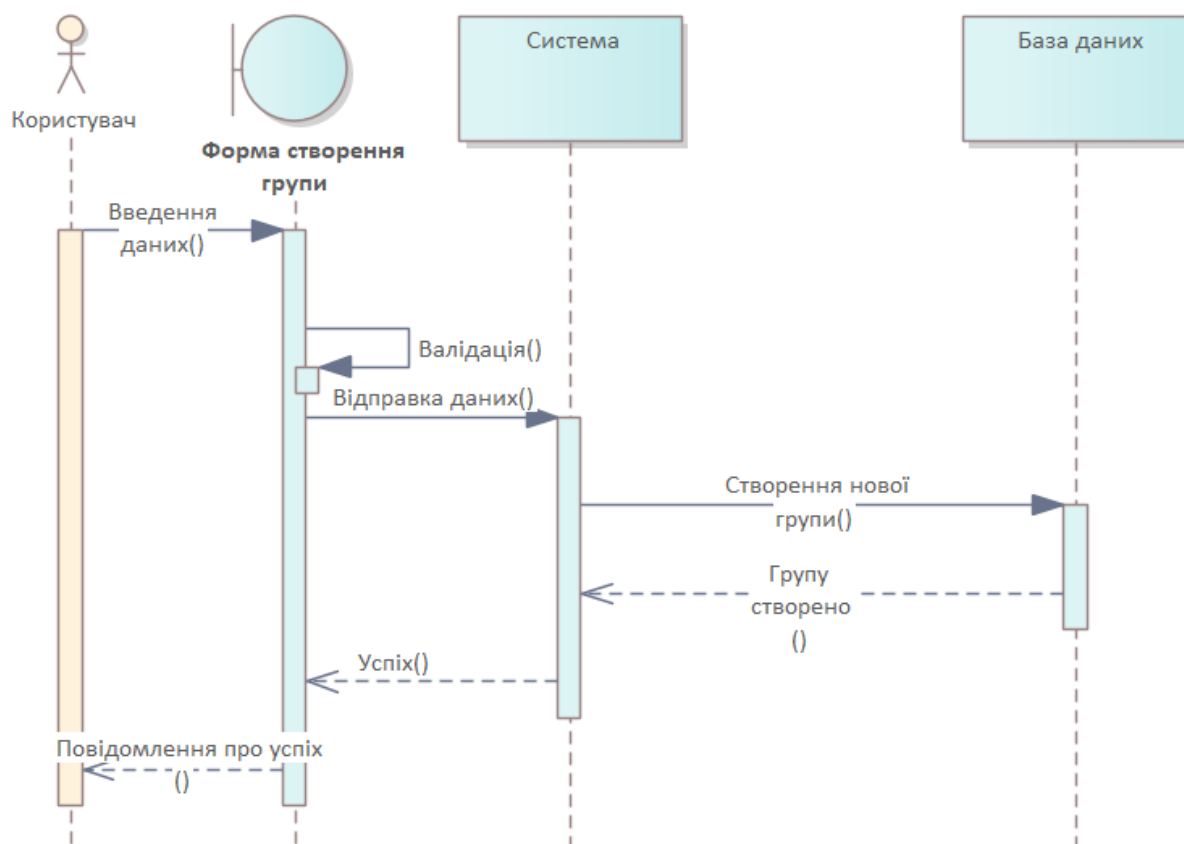


Рисунок 2.16 – Діаграма послідовності створення групи

Джерело: розроблено самостійно

1. Користувач вводить необхідні дані у форму створення групи – *Введення даних()*.
2. Форма створення групи виконує валідацію даних – *Валідація()*:
  - Якщо дані некоректні, користувач отримує повідомлення про помилку.
  - Якщо дані валідні, вони надсилаються в систему (*Відправка даних()*).
3. Система ініціює створення нової групи у базі даних – *Створення нової групи()*.
4. База даних успішно створює запис про нову групу і надсилає відповідь – *Групу створено()*.
5. Система надсилає результат у форму – *Успіх()*.
6. Форма створення групи повідомляє користувача про успішне створення групи – *Повідомлення про успіх()*.

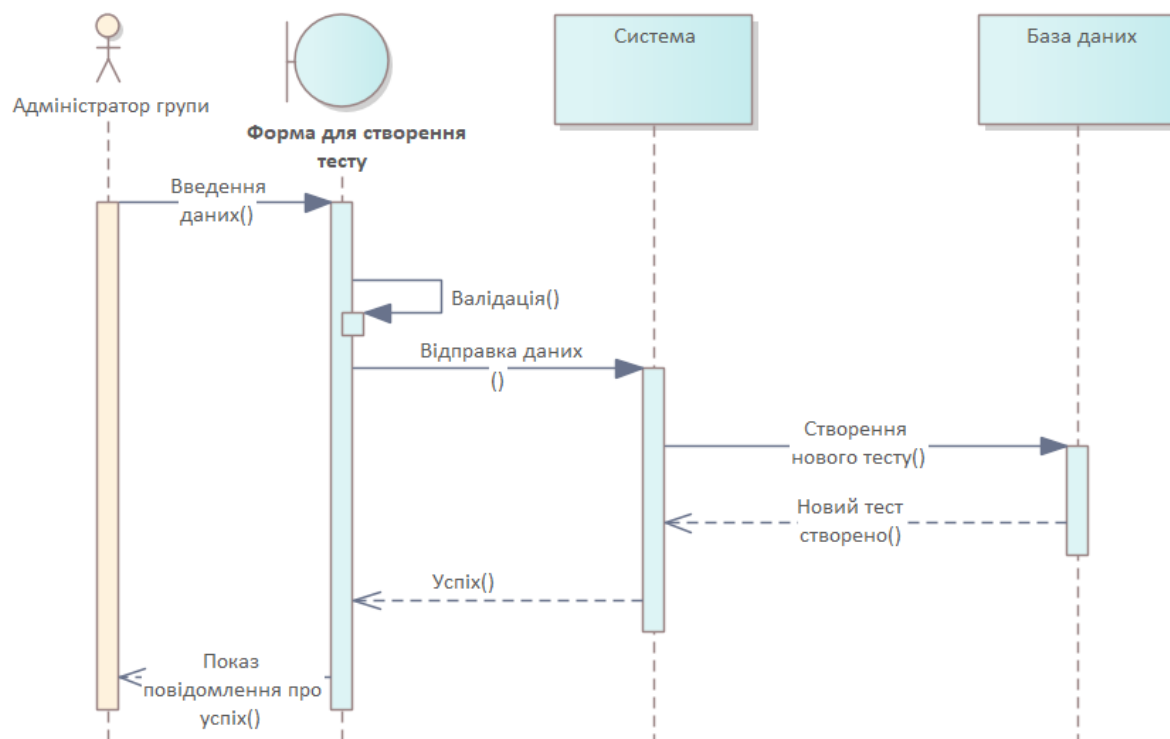


Рисунок 2.17 – Діаграма послідовності створення тесту

Джерело: розроблено самостійно

1. Адміністратор групи заповнює необхідні поля у формі створення тесту – *Введення даних()*.
2. Форма створення тесту виконує перевірку коректності введеної інформації – *Валідація()*:
  - Якщо дані некоректні, користувач отримує повідомлення про помилку.
  - Якщо дані валідні, форма надсилає їх до системи – *Відправка даних()*.
3. Система приймає дані та ініціює створення нового тесту в базі даних – *Створення нового тесту()*.
4. База даних створює новий запис про тест і надсилає підтвердження до системи – *Новий тест створено()*.
5. Система передає результат у форму – *Успіх()*.
6. Форма створення тесту відображає повідомлення для адміністратора про успішне створення тесту – *Показ повідомлення про успіх()*.

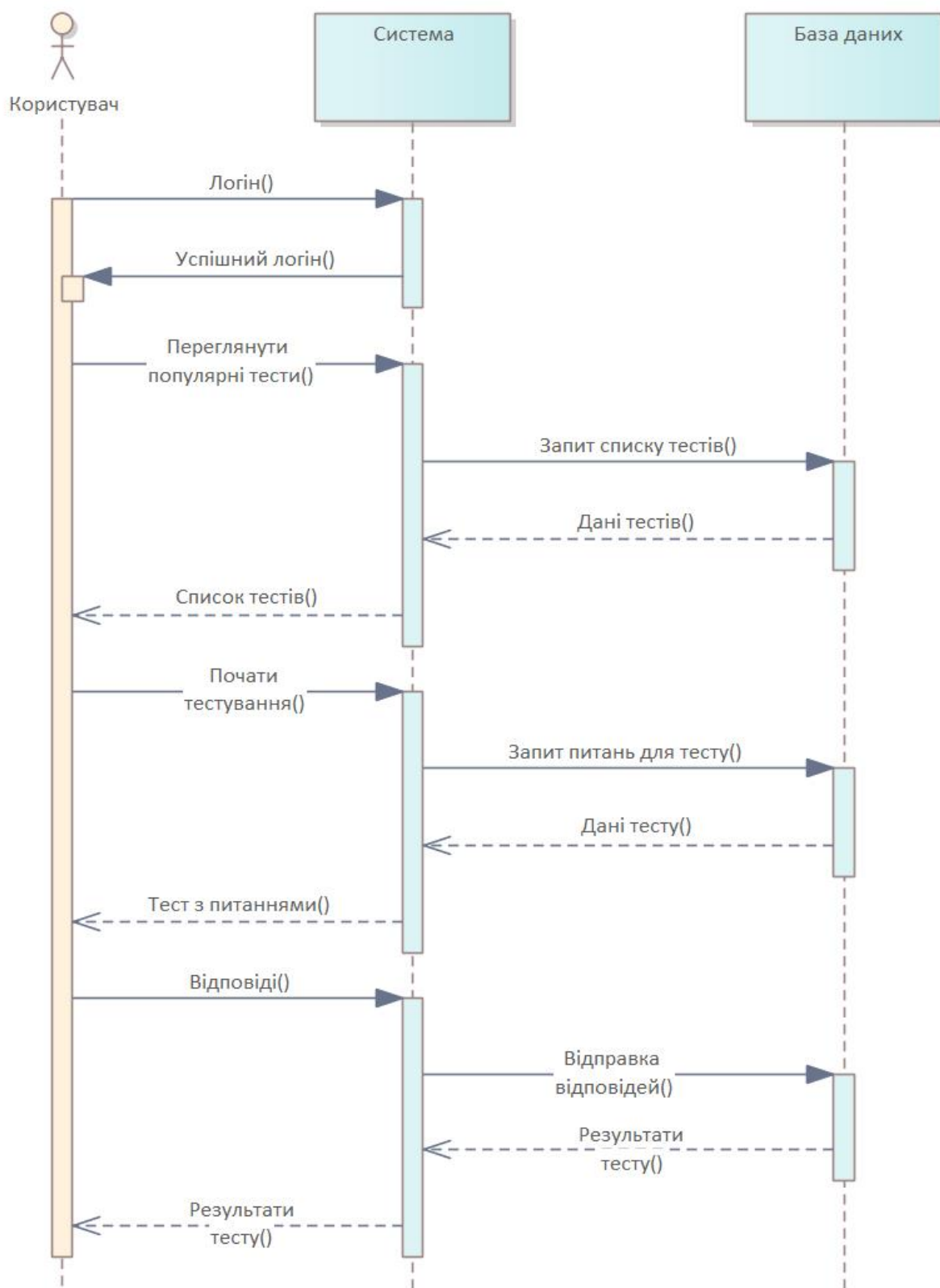


Рисунок 2.18 – Діаграма послідовності проходження тесту

Джерело: розроблено самостійно

1. Користувач здійснює вхід у систему – *Логін()*.
2. Система перевіряє облікові дані та повідомляє про успішну авторизацію – *Успішний логін()*.

3. Після входу користувач ініціює перегляд доступних тестів – *Переглянути популярні тести()*.
4. Система формує запит до бази даних на отримання списку тестів – *Запит списку тестів()*.
5. База даних повертає список тестів – *Дані тестів()*.
6. Система передає ці дані користувачу – *Список тестів()*.
7. Користувач обирає конкретний тест і розпочинає його проходження – *Почати тестування()*.
8. Система формує запит до бази даних на отримання запитань тесту – *Запит питань для тесту()*.
9. База даних повертає відповідні запитання – *Дані тесту()*.
10. Система надсилає тест із питаннями користувачу – *Тест з питаннями()*.
11. Після проходження тесту користувач надсилає відповіді – *Відповіді()*.
12. Система приймає відповіді й надсилає їх до бази даних – *Відправка відповідей()*.
13. База даних обробляє відповіді та повертає результат – *Результати тесту()*.
14. Система надсилає результат користувачу – *Результати тесту()*.

### 2.3.3 Модель структури системи

Для подання внутрішньої структури системи була створена діаграма визначення блоків у вигляді ієрархії блоків різних типів.

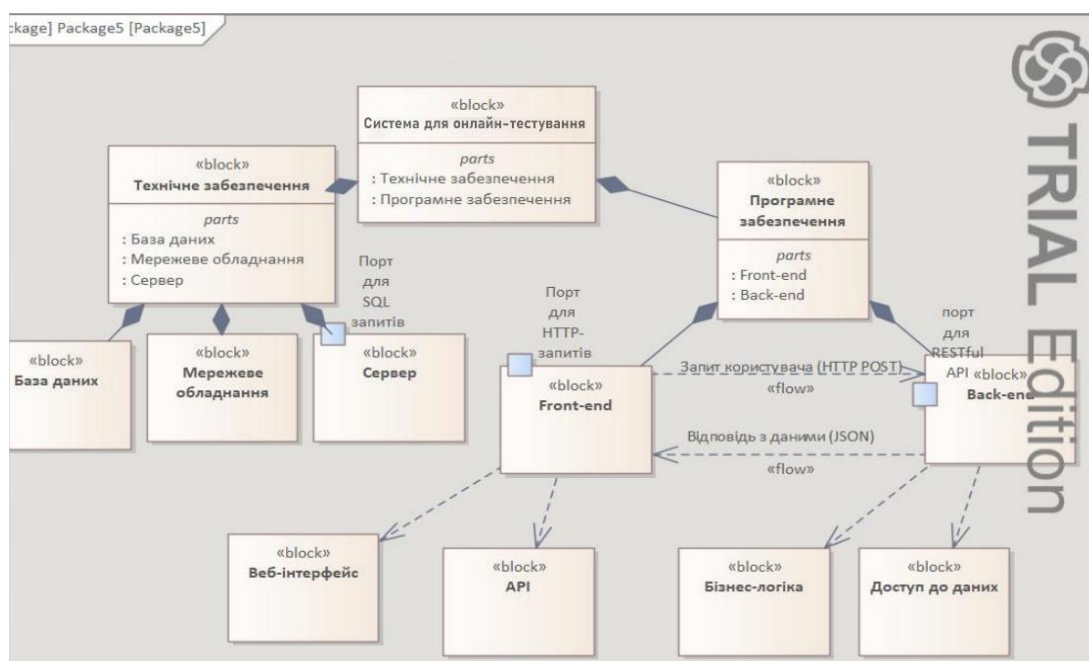


Рисунок 2.19 – Діаграма визначення блоків

Джерело: розроблено самостійно

Побудована діаграма внутрішніх блоків зображена на рисунку 2.20.

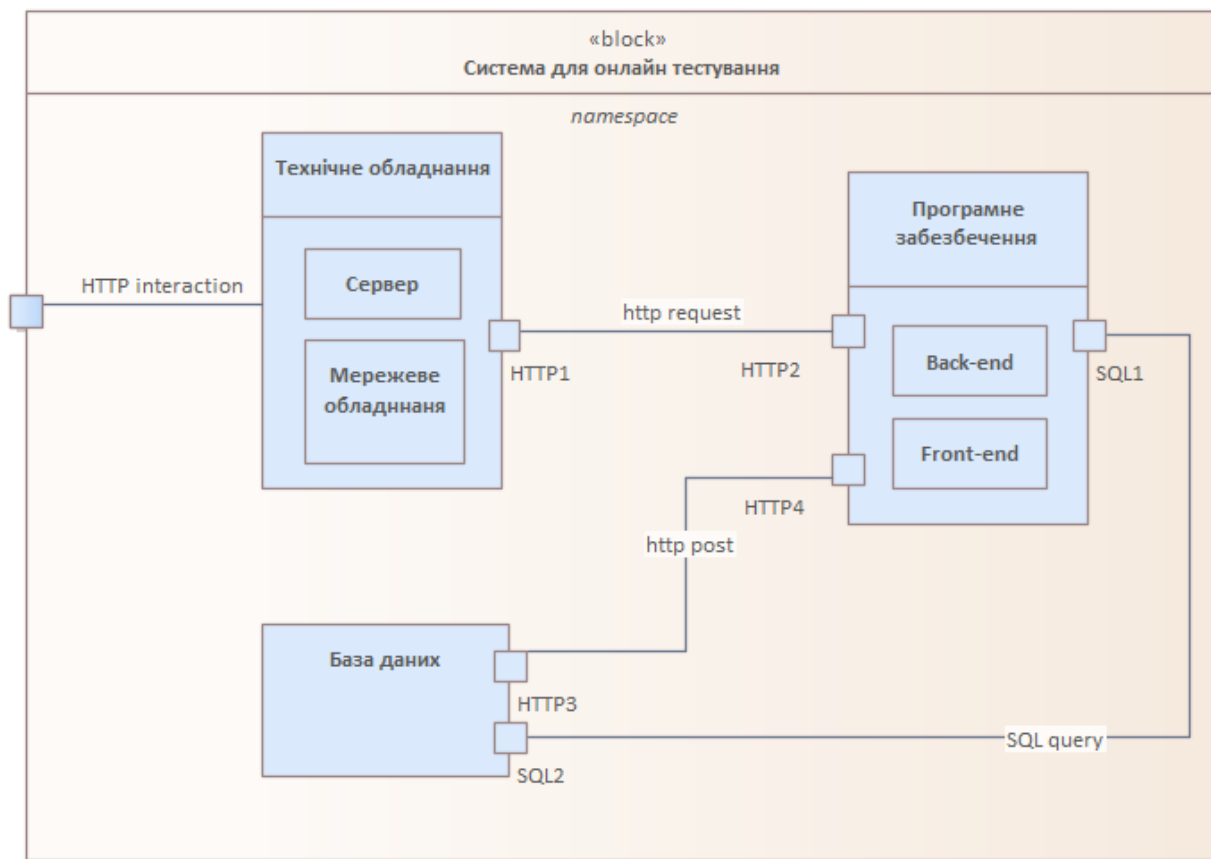


Рисунок 2.20 – Діаграма внутрішніх блоків

Джерело: розроблено самостійно

Побудована діаграма класів, яка описує класи і атрибути для даної системи, зображена на рисунку 2.21.

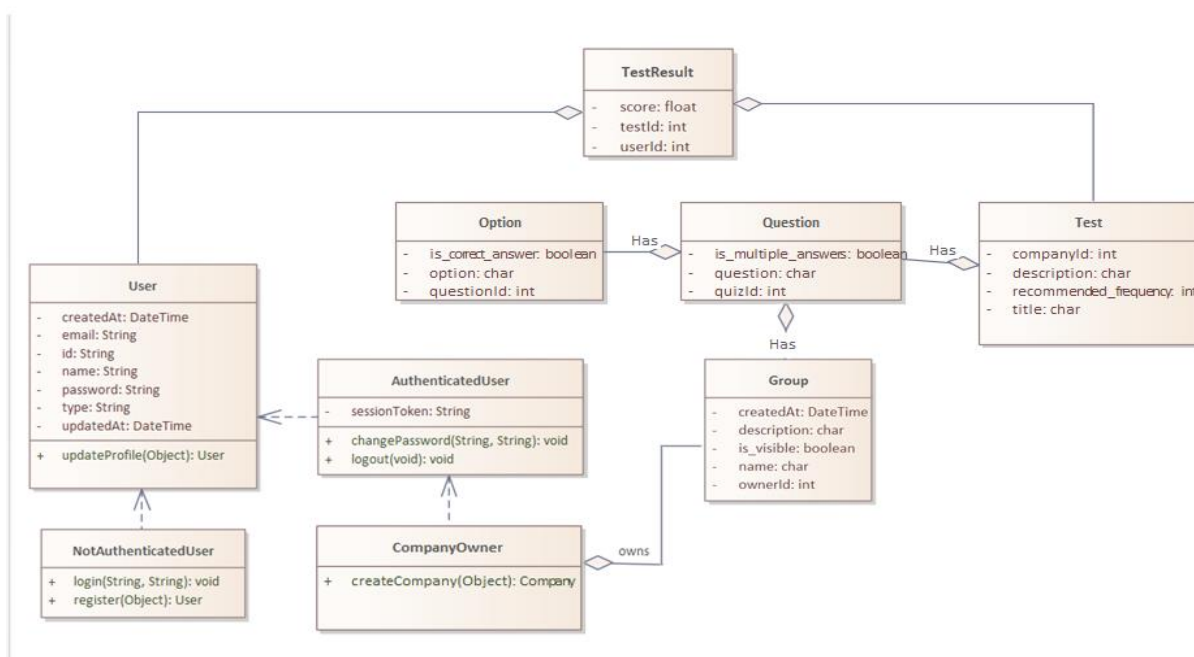


Рисунок 2.21 – Діаграма класів

Джерело: розроблено самостійно

## **Висновки до Розділу 2**

У даному розділі було проведено аналіз функціональних та нефункціональних вимог до системи онлайн-тестування, визначено бізнес-цілі ключових користувачів, а також сформовано математичну модель задачі. Детально описано алгоритм функціонування системи, структуру вхідних і вихідних даних, а також способи обробки результатів тестування. Було змодельовано інтерфейс користувача, діаграми прецедентів та послідовностей, що дозволило формалізувати основні сценарії взаємодії з платформою. Отримані результати створюють цілісну основу для подальшого проєктування та реалізації компонентів системи.

## РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

### 3.1 Інформаційне забезпечення

#### 3.1.1 Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення платформи для онлайн-тестування включає сукупність класифікованої та структурованої інформації, яка використовується для підтримки процесів створення, управління та проходження тестів. Система орієнтована на дві основні категорії користувачів: авторів тестів (викладачів, тренерів, адміністраторів) та учасників тестування (студентів, слухачів, співробітників тощо).

Основу інформаційного забезпечення складають наступні типи даних:

- Дані про користувачів (реєстраційна інформація, роль у системі, історія активності);
- Дані про групи (назва, опис, список учасників, автор групи);
- Дані про тести (назва тесту, дата створення, список питань, правильні відповіді, тощо);
- Питання та відповіді (тип питання, текст, варіанти відповідей, позначення правильної відповіді);
- Результати тестування (отримані бали, оцінка, аналітика відповідей);
- Службова інформація (логи, повідомлення, налаштування користувача).

Інформація у системі зберігається в реляційній базі даних, що дозволяє забезпечити її цілісність, структурованість та ефективне обслуговування запитів. Дані пов'язані між собою за допомогою ключів, що дає змогу підтримувати логічні зв'язки між користувачами, тестами та результатами.

Також система передбачає розмежування прав доступу, яке реалізується за допомогою механізмів автентифікації та авторизації. Це гарантує, що автори тестів мають доступ до редагування та перегляду результатів, а учасники — лише до проходження призначених їм тестів.

Таким чином, інформаційне забезпечення системи виконує ключову роль у забезпеченні її функціональності, підтримуючи всі процеси онлайн-тестування від створення тестів до аналізу результатів.

### 3.1.2 Організація збору і передавання первинної інформації

У системі онлайн-тестування первинна інформація формується безпосередньо користувачами під час взаємодії з інтерфейсом. Джерелами первинних даних є:

- **Автори тестів**, які створюють групи, тести, питання та варіанти відповідей;
- **Учасники тестування**, які проходять тести, обирають відповіді та надсилають їх на перевірку;
- **Система автентифікації**, яка реєструє дії користувача (вхід, вихід, час активності).

Збір первинної інформації відбувається за допомогою веб-форм на відповідних сторінках інтерфейсу: реєстрації, створення тестів, редагування профілю, проходження тесту тощо. Введені користувачем дані валідуються на клієнтській стороні та передаються на серверну частину через API-запити.

Передача даних здійснюється через захищені канали з використанням протоколу HTTPS, що забезпечує конфіденційність інформації. Серверна частина обробляє запити, зберігає інформацію у базі даних, а також забезпечує зворотний зв'язок з користувачем у вигляді повідомлень, результатів тестування або підтвердження виконаної дії.

Усі дані зберігаються централізовано в базі даних, що дає змогу уникнути дублювання інформації, забезпечити її актуальність, точність і доступність. Додатково передбачено кешування та логування критичних дій користувачів для підвищення продуктивності та безпеки системи.

### 3.1.3 Структура інформаційних масивів

Структури інформаційних масивів описана у вигляді таблиць (Табл.: 3.1 – 3.9)

Таблиця 3.1 – Опис структури масиву «Користувачі»

| Найменування     | Ідентифікатор у програмі | Формат    | Бізнес правила |                  |               |
|------------------|--------------------------|-----------|----------------|------------------|---------------|
|                  |                          |           | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор    | id                       | UUID      | РК             | Так              | ІД            |
| Ім'я користувача | username                 | text      | -              | Так              | ІД            |
| Електронна пошта | email                    | text      | -              | Так              | ІД            |
| Аватар           | avatar                   | image URL | -              | Ні               | -             |
| Час створення    | created_at               | timestamp | -              | Так              | -             |
| Час оновлення    | updated_at               | timestamp | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.2 – Опис структури масиву «Компанії (групи)»

| Найменування   | Ідентифікатор у програмі | Формат    | Бізнес правила |                  |               |
|----------------|--------------------------|-----------|----------------|------------------|---------------|
|                |                          |           | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор  | id                       | int       | РК             | Так              | ІД            |
| Власник        | owner                    | FK (User) | -              | Ні               | -             |
| Назва компанії | name                     | text      | -              | Так              | -             |
| Опис           | description              | text      | -              | Ні               | -             |
| Видимість      | is_visible               | boolean   | -              | Так              | -             |
| Час створення  | created_at               | timestamp | -              | Так              | -             |
| Час оновлення  | updated_at               | timestamp | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.3 – Опис структури масиву «Компанія (група) - користувач»

| Найменування  | Ідентифікатор у програмі | Формат       | Бізнес правила |                  |               |
|---------------|--------------------------|--------------|----------------|------------------|---------------|
|               |                          |              | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор | id                       | int          | РК             | Так              | ІД            |
| Компанія      | company                  | FK (Company) | -              | Так              | -             |
| Користувач    | user                     | FK (User)    | -              | Так              | -             |
| Адміністратор | is_administrator         | boolean      | -              | Так              | -             |
| Час створення | created_at               | timestamp    | -              | Так              | -             |
| Час оновлення | updated_at               | timestamp    | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.4 – Опис структури масиву «Тести»

| Найменування          | Ідентифікатор у програмі | Формат       | Бізнес правила |                  |               |
|-----------------------|--------------------------|--------------|----------------|------------------|---------------|
|                       |                          |              | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор         | id                       | int          | PK             | Так              | ІД            |
| Компанія              | company                  | FK (Company) | -              | Так              | -             |
| Назва                 | title                    | text         | -              | Так              | -             |
| Опис                  | description              | text         | -              | Так              | -             |
| Рекомендована частота | recommended_frequency    | int          | -              | Так              | -             |
| Час створення         | created_at               | timestamp    | -              | Так              | -             |
| Час оновлення         | updated_at               | timestamp    | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.5 – Опис структури масиву «Запитання»

| Найменування       | Ідентифікатор у програмі | Формат    | Бізнес правила |                  |               |
|--------------------|--------------------------|-----------|----------------|------------------|---------------|
|                    |                          |           | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор      | id                       | int       | PK             | Так              | ІД            |
| Тест               | quiz                     | FK (Quiz) | -              | Так              | -             |
| Запитання          | question                 | text      | -              | Так              | -             |
| Багатоваріантність | is_multiple_answers      | boolean   | -              | Так              | -             |
| Час створення      | created_at               | timestamp | -              | Так              | -             |
| Час оновлення      | updated_at               | timestamp | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.6 – Опис структури масиву «Варіанти відповідей»

| Найменування        | Ідентифікатор у програмі | Формат        | Бізнес правила |                  |               |
|---------------------|--------------------------|---------------|----------------|------------------|---------------|
|                     |                          |               | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор       | id                       | int           | PK             | Так              | ІД            |
| Запитання           | question                 | FK (Question) | -              | Так              | -             |
| Варіант відповіді   | option                   | text          | -              | Так              | -             |
| Правильна відповідь | is_correct_answer        | boolean       | -              | Так              | -             |
| Час створення       | created_at               | timestamp     | -              | Так              | -             |
| Час оновлення       | updated_at               | timestamp     | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.7 – Опис структури масиву «Результат тесту»

| Найменування          | Ідентифікатор у програмі | Формат       | Бізнес правила |                  |               |
|-----------------------|--------------------------|--------------|----------------|------------------|---------------|
|                       |                          |              | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор         | id                       | int          | PK             | Так              | ІД            |
| Користувач            | user                     | FK (User)    | -              | Так              | -             |
| Компанія              | company                  | FK (Company) | -              | Так              | -             |
| Тест                  | quiz                     | FK (Quiz)    | -              | Так              | -             |
| Результат (%)         | score                    | float        | -              | Так              | -             |
| Усього запитань       | total_questions          | int          | -              | Так              | -             |
| Правильних відповідей | correct_answers          | int          | -              | Так              | -             |
| Час створення         | created_at               | timestamp    | -              | Так              | -             |
| Час оновлення         | updated_at               | timestamp    | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.8 – Опис структури масиву «Статус тестування користувача»

| Найменування  | Ідентифікатор у програмі | Формат    | Бізнес правила |                  |               |
|---------------|--------------------------|-----------|----------------|------------------|---------------|
|               |                          |           | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор | id                       | int       | PK             | Так              | ІД            |
| Користувач    | user                     | FK (User) | -              | Так              | -             |

| Найменування        | Ідентифікатор у програмі | Формат    | Бізнес правила |                  |               |
|---------------------|--------------------------|-----------|----------------|------------------|---------------|
|                     |                          |           | Первинний ключ | Обов'язкове поле | Індексне поле |
| Тест                | quiz                     | FK (Quiz) | -              | Так              | -             |
| Останнє проходження | last_taken_time          | datetime  | -              | Ні               | -             |
| Середній бал        | average_score            | float     | -              | Так              | -             |
| Кількість спроб     | total_attempts           | int       | -              | Так              | -             |
| Час створення       | created_at               | timestamp | -              | Так              | -             |
| Час оновлення       | updated_at               | timestamp | -              | Так              | -             |

*Джерело: розроблено самостійно*

Таблиця 3.9 – Опис структури масиву «Взаємодії»

| Найменування  | Ідентифікатор у програмі | Формат       | Бізнес правила |                  |               |
|---------------|--------------------------|--------------|----------------|------------------|---------------|
|               |                          |              | Первинний ключ | Обов'язкове поле | Індексне поле |
| Ідентифікатор | id                       | int          | РК             | Так              | ІД            |
| Тип взаємодії | type                     | enum (3)     | -              | Так              | -             |
| Статус        | status                   | enum (3)     | -              | Так              | -             |
| Компанія      | company                  | FK (Company) | -              | Так              | -             |
| Користувач    | user                     | FK (User)    | -              | Так              | -             |
| Ініціатор     | initiator                | FK (User)    | -              | Так              | -             |
| Час створення | created_at               | timestamp    | -              | Так              | -             |
| Час оновлення | updated_at               | timestamp    | -              | Так              | -             |

*Джерело: розроблено самостійно*

### 3.1.4 Вибір СКБД

Для реалізації роботи було обрано систему керування базами даних PostgreSQL. Цей вибір обумовлений її високою продуктивністю, стабільністю та відкритою ліцензією, що дозволяє використовувати базову збірку без додаткових витрат. PostgreSQL повністю сумісна з фреймворком Django, який використовується на серверній частині платформи, але на відміну від багатьох інших СКБД, вона забезпечує розширені можливості обробки даних. Зокрема, підтримка збережених процедур з використанням мови PL/pgSQL дозволяє виконувати складну логіку обробки безпосередньо на сервері бази даних, знижуючи навантаження на сервер

застосунку та пришвидшуючи доступ до інформації. Крім того, PostgreSQL відома своєю здатністю ефективно працювати з великими обсягами даних, підтримує транзакції, складні SQL-запити, масштабування в хмарному середовищі, а також забезпечує цілісність і надійність даних. Активна спільнота та детальна документація роблять її зручною у використанні для створення сучасних веб-застосунків.

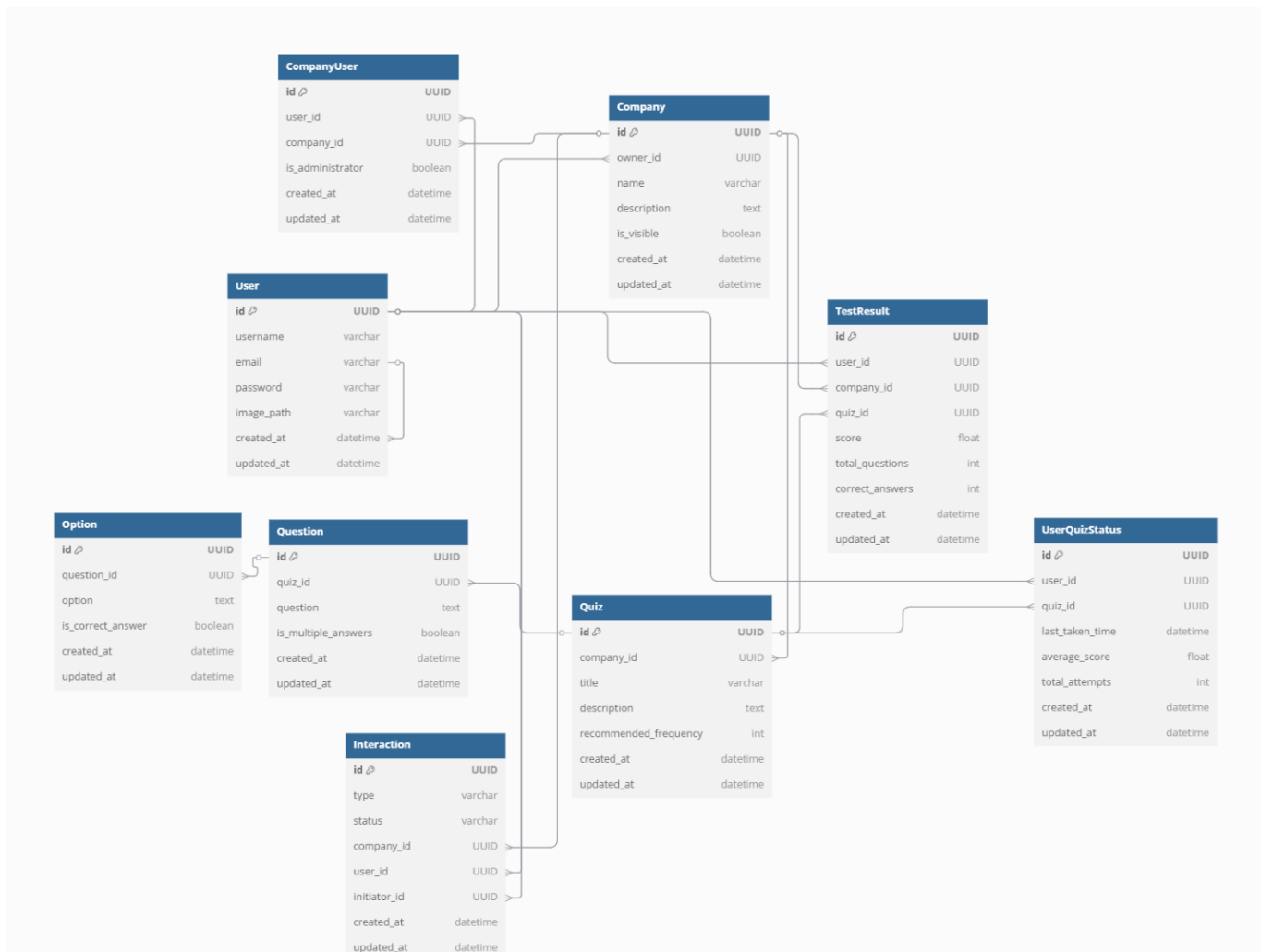


Рисунок 3.1 – Даталогічна модель

*Джерело: розроблено самостійно*

Детальний опис структури моделей бази даних наведено в Додатку А

### 3.2 Технічне забезпечення

#### Архітектурний підхід

У роботі реалізовано клієнт-серверну архітектуру, яка є однією з найбільш ефективних та гнучких моделей для створення сучасних інформаційних систем. Вона передбачає чітке розділення обов'язків між клієнтом і сервером. Клієнтська частина забезпечує інтерфейс користувача та взаємодію з ним, у той час як серверна частина відповідає за обробку запитів, доступ до бази даних та реалізацію бізнес-логіки.

Таке розділення дозволяє спростити підтримку й розвиток системи, забезпечити її масштабованість та підвищити рівень безпеки. Усі дані зберігаються централізовано на сервері, що мінімізує ризики втрати або несанкціонованого доступу до інформації. Крім того, клієнт-серверна архітектура полегшує інтеграцію з іншими системами та дає змогу одночасно працювати багатьом користувачам у реальному часі.

Система підтримує віддалений доступ до функціоналу через мережу Інтернет, що є особливо важливим для користувачів, які можуть працювати з нею з різних пристроїв — комп'ютерів, ноутбуків або мобільних телефонів. Завдяки використанню відкритих стандартів обміну даними система може бути інтегрована з іншими зовнішніми сервісами, що також відкриває перспективи для її подальшого розвитку.

### **Апаратне забезпечення**

Для повноцінної роботи над системою на етапах розробки, тестування та експлуатації необхідне відповідне апаратне забезпечення.

Розробка системи здійснювалася на персональному комп'ютері, обладнаному процесором Intel Core i5 (або аналогічним за продуктивністю), об'ємом оперативної пам'яті не менше 8 ГБ та твердотілим накопичувачем (SSD) із вільним простором від 20 ГБ. Такі технічні характеристики дозволяють одночасно працювати з середовищем розробки, запускати локальний сервер, тестувати клієнтську частину та здійснювати доступ до бази даних без затримок.

Для кінцевих користувачів достатньо мати будь-який пристрій із встановленим сучасним веб-браузером і доступом до Інтернету. Це може бути настільний комп'ютер, ноутбук або мобільний пристрій. Такий підхід значно спрощує процес доступу до системи і не вимагає встановлення додаткового програмного забезпечення на клієнтській стороні.

### **Інфраструктура розгортання**

У рамках реалізації дипломної роботи передбачено можливість розгортання системи на сучасних хмарних платформах або віртуальних приватних серверах. Серверна частина системи може бути розміщена на хостингу з підтримкою Python-середовища, де здійснюється обробка запитів, авторизація користувачів та робота з

базою даних. Клієнтська частина, що реалізована як веб-застосунок, може бути розміщена на спеціалізованих сервісах для фронтенд-розгортання.

Наявність стабільного інтернет-з'єднання, резервного копіювання та системи моніторингу забезпечує надійність та безперервність роботи системи, а також дозволяє своєчасно реагувати на технічні несправності або перевантаження.

### **3.3 Програмне забезпечення**

#### **Серверна частина**

Серверна частина онлайн-порталу чи back-end, є головним елементом, який забезпечує функціональність веб-додатка. Вона відповідає за обробку запитів, управління базами даних, сценаріями і виконання операцій, що відбуваються на сервері. Коли клієнт виконує дію, таку як вхід у систему, заповнення форми чи проходження тесту, браузер передає відповідний запит на сервер, де відбувається його обробка. Бекенд дозволяє сервісу надавати взаємодію з базами даних, зберігаючи інформацію.

Серверну логіку реалізовано на мові програмування Python з використанням Django — потужного вебфреймворку, що дозволяє швидко створювати масштабовані вебзастосунки. Для побудови REST API використано розширення Django REST Framework, яке забезпечує серіалізацію даних, маршрутизацію запитів та механізми авторизації.

Розробка серверної частини здійснювалася в середовищі PyCharm IDE, яке надає зручні інструменти для написання, налагодження та тестування коду, включаючи підтримку Django, інтеграцію з Git та автоматичне завершення коду.

На етапі тестування серверної частини інформаційної системи використовувався локальний сервер, що дозволяє перевіряти роботу скриптів та їхню взаємодію з базою даних. Зокрема, було перевірено:

- коректність обробки запитів клієнтів;
- взаємодію серверної частини з базою даних;
- генерацію динамічних сторінок і передачу їх користувачу;
- реакцію системи на різні типи запитів і помилок.

Тестування проводилося як на етапі розробки, так і під час інтеграції нових функціональних можливостей, що дозволяє своєчасно виявляти та виправляти помилки.

### **Клієнтська частина**

Одним із головних аспектів розробки клієнтської частини є забезпечення постійної взаємодії клієнта із системою. Для цього створюються механізми, які дозволяють швидко та легко виконувати основні операції, такі як реєстрація, проходження тестування, перегляд результатів та інші функції, що доступні в інформаційній системі. Окрім того, важливим є дотримання принципів зручності, що сприяє ефективному виконанню завдань користувачами незалежно від їх рівня підготовки. Візуальна привабливість та простота навігації забезпечують позитивний досвід користувача, що сприяє його задоволеності системою.

Клієнтська частина веб-сайту створюється із врахуванням потреб користувачів і розробляється на основі макетів та прототипів. Її було реалізовано за допомогою JavaScript-бібліотеки React.js, яка дає змогу створювати динамічні та зручні у використанні вебінтерфейси. Для візуального оформлення застосовано бібліотеку компонентів Material UI, що забезпечує сучасний дизайн відповідно до принципів Material Design та містить велику кількість готових UI-елементів.

Розробка клієнтської частини здійснювалася у середовищі WebStorm, яке надає потужні інструменти для роботи з фронтенд-технологіями: підсвічування синтаксису, інтеграцію з системами контролю версій, автоматичне завершення коду, а також зручні інструменти для дебагу.

### 3.4 Результати реалізації інформаційної системи

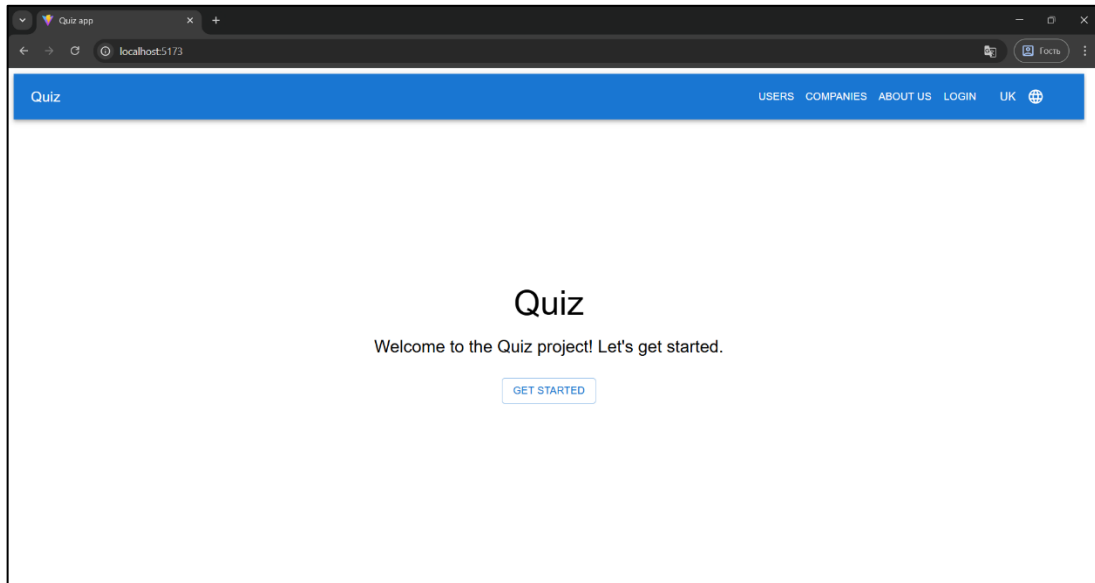


Рисунок 3.2 – Інтерфейс головної сторінки

*Джерело: розроблена платформа*

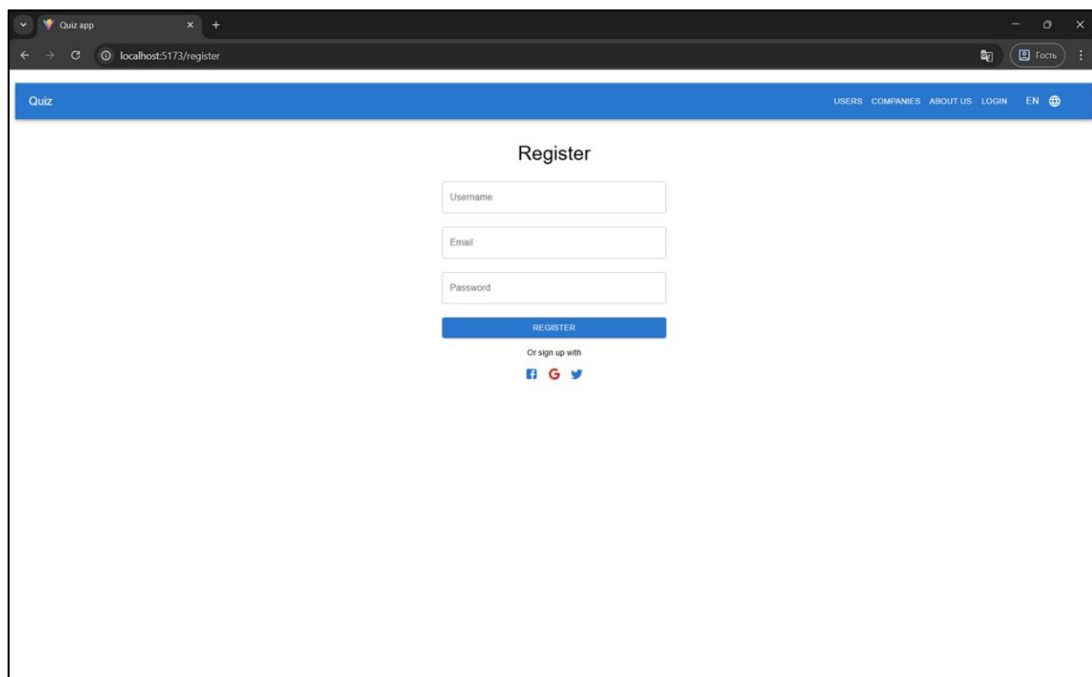


Рисунок 3.3 – Інтерфейс сторінки реєстрації

*Джерело: розроблена платформа*

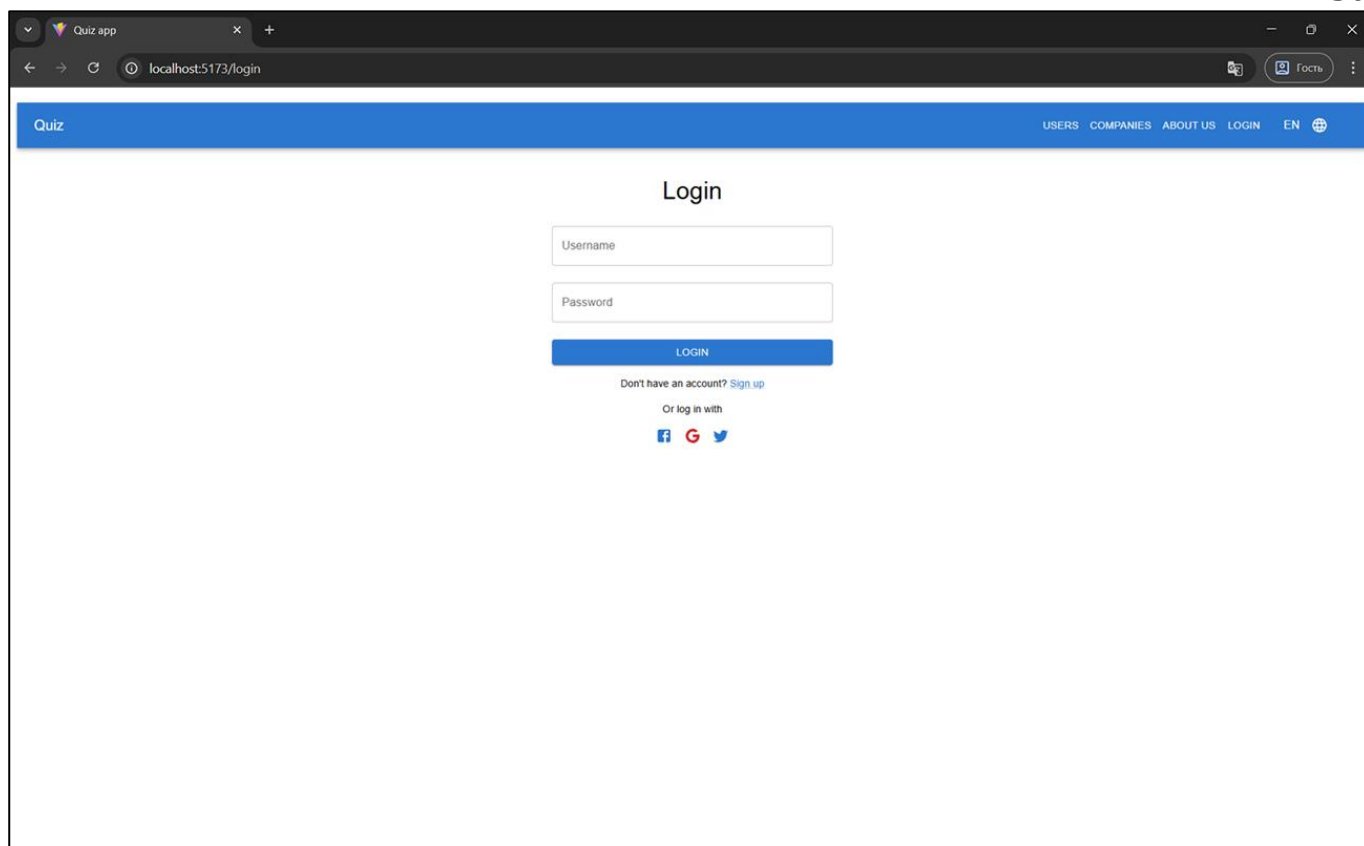


Рисунок 3.4 – Інтерфейс сторінки входу

*Джерело: розроблена платформа*

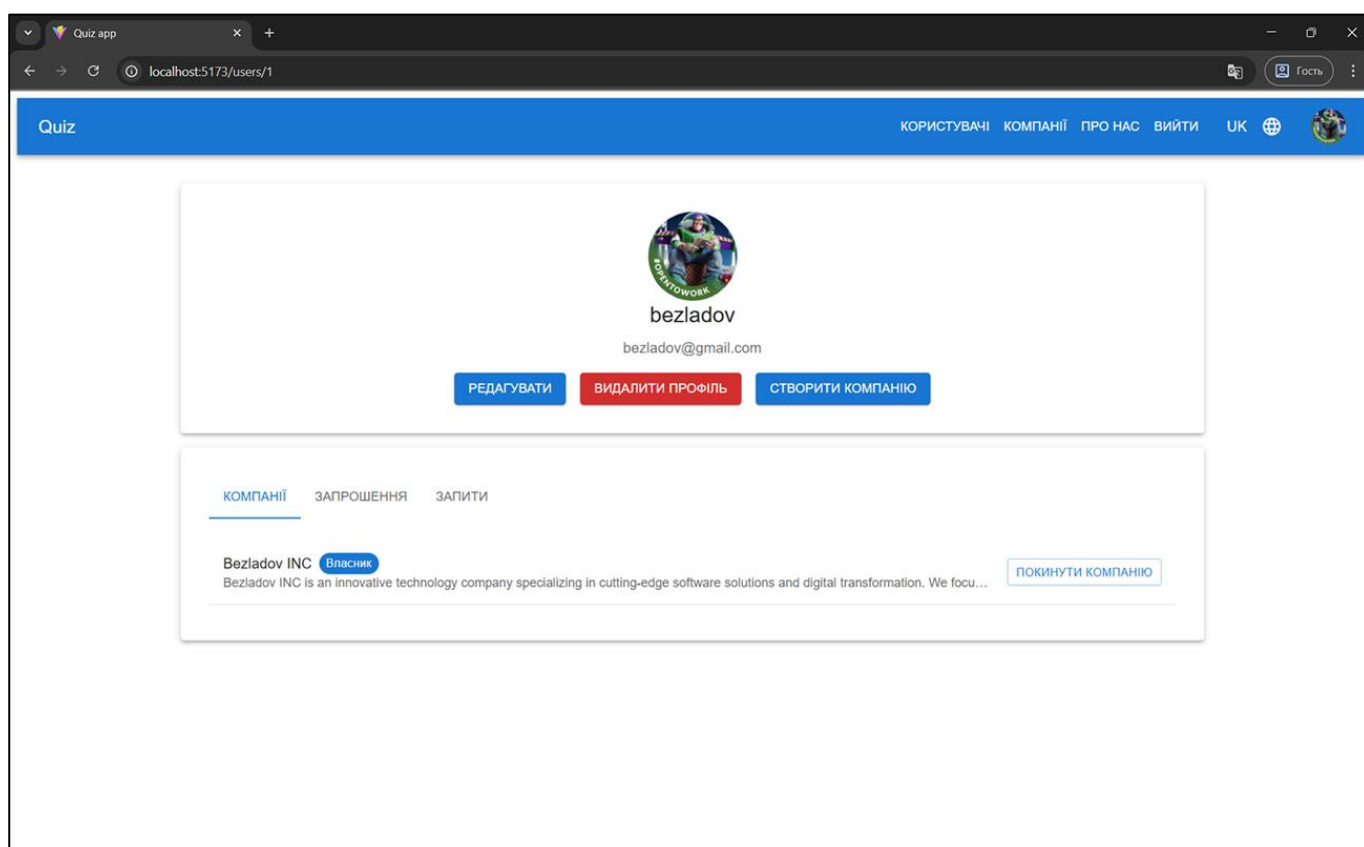


Рисунок 3.5 – Інтерфейс профілю користувача

*Джерело: розроблена платформа*

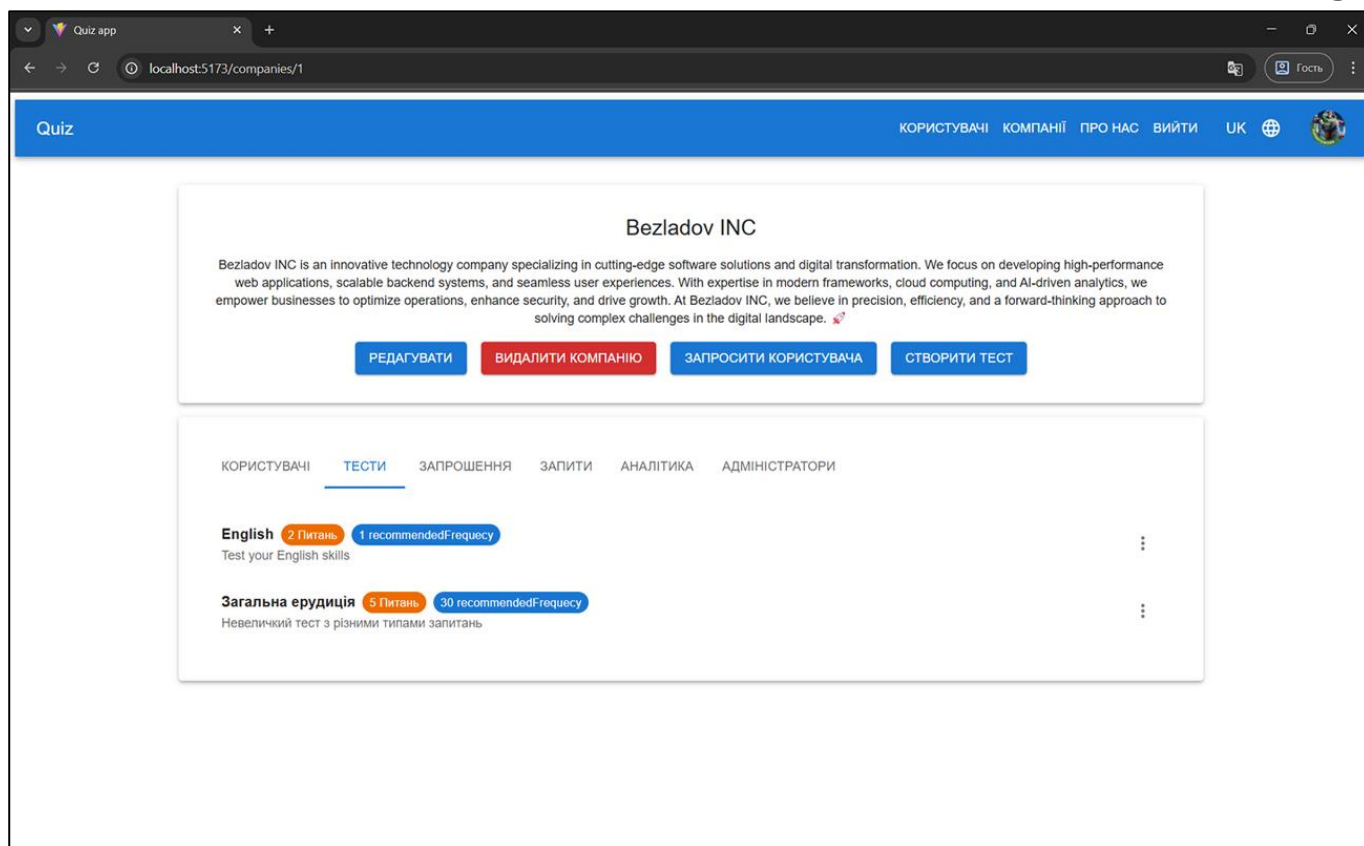


Рисунок 3.6 – Профіль групи

*Джерело: розроблена платформа*

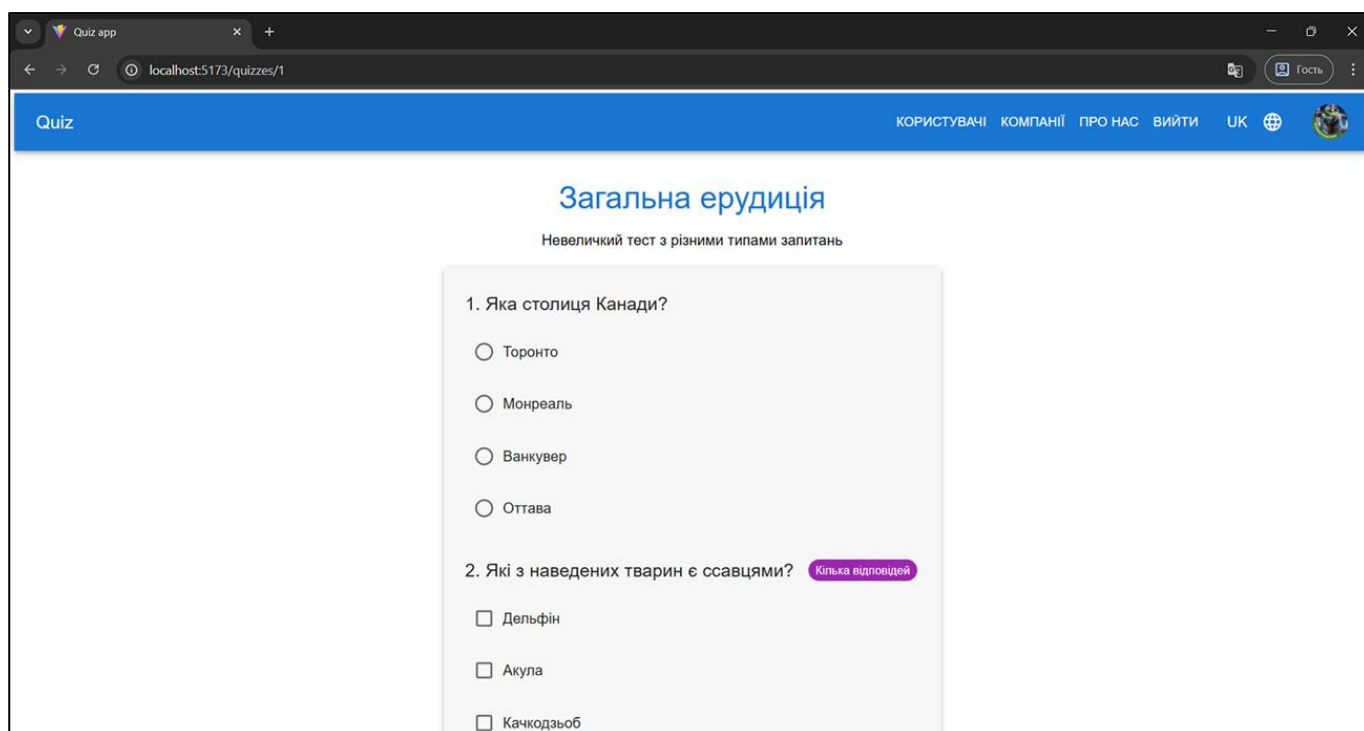


Рисунок 3.7 – Проходження тесту

*Джерело: розроблена платформа*

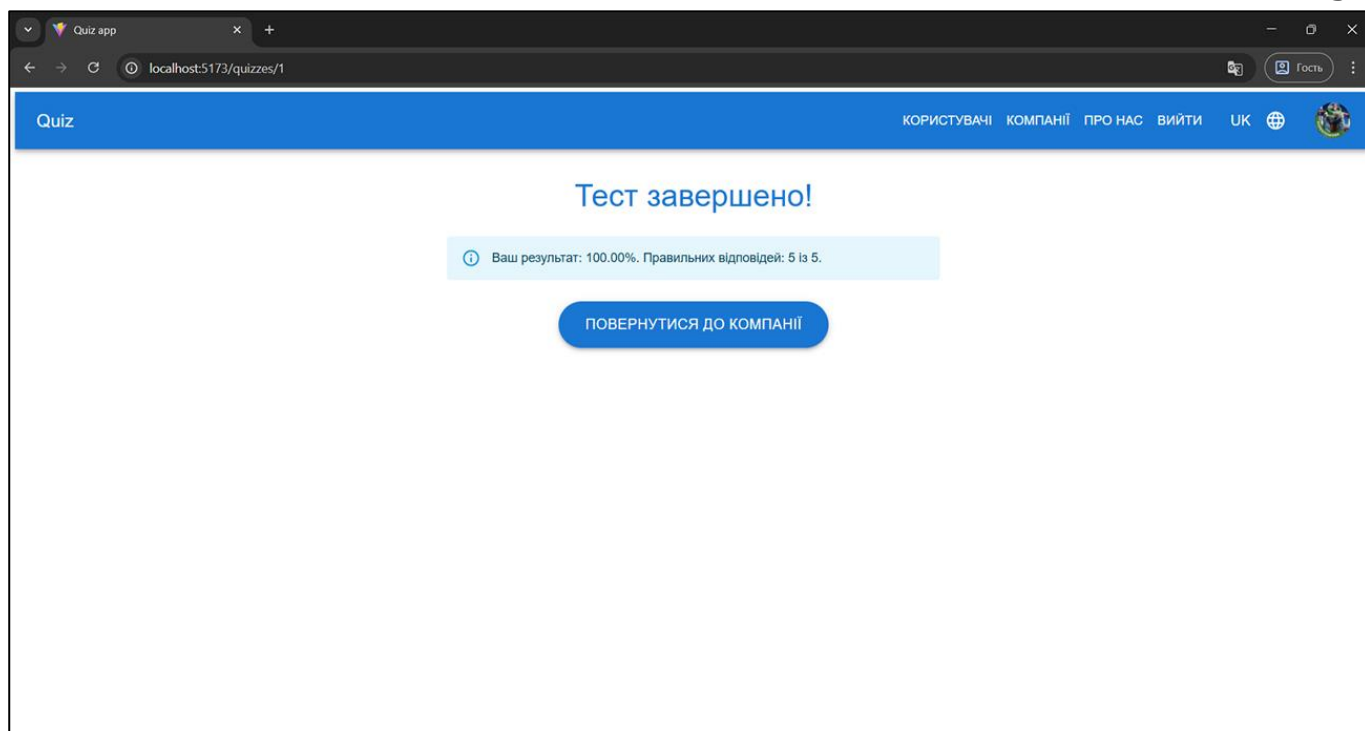


Рисунок 3.8 – Результати тесту

*Джерело: розроблена платформа*

## ВИСНОВКИ

У процесі виконання дипломної роботи було реалізовано повнофункціональну інформаційну систему для онлайн-тестування, що дозволяє користувачам створювати, проходити та аналізувати результати тестів. Розроблена система є сучасною веб-платформою, орієнтованою на потреби навчальних закладів, курсів підвищення кваліфікації, корпоративного навчання та самостійного засвоєння знань.

На етапі аналітики було детально досліджено предметну область, проведено аналіз існуючих рішень, зокрема таких платформ, як Quizlet, Kahoot!, Google Forms, Moodle, Typeform тощо. На основі вивчених матеріалів було визначено основні функціональні та нефункціональні вимоги до системи. Зокрема, акцент зроблено на автоматизації оцінювання, простоті використання, забезпеченні безпеки даних користувачів та можливості масштабування в майбутньому.

Під час роботи над дипломною роботою було створено моделі поведінки, структури та інтерфейсу системи. Було застосовано засоби візуального моделювання, побудовано діаграми прецедентів, послідовностей, класів, блоків та внутрішніх зв'язків, що дозволило формалізувати логіку функціонування системи та спростити етап розробки. Також створено інтерактивний прототип у середовищі Figma, який допоміг спроектувати зручний інтерфейс користувача.

Технічна реалізація дипломної роботи охоплює повний стек сучасних технологій. Серверна частина була реалізована за допомогою мови програмування Python та фреймворку Django з використанням REST API. Клієнтську частину створено на основі бібліотеки React.js із застосуванням компонентів Material UI, що забезпечило високу інтерактивність та привабливий вигляд веб-інтерфейсу. Для зберігання даних обрано надійну систему керування базами даних PostgreSQL.

Проведене ручне тестування підтвердило стабільну роботу всіх модулів системи. Сервіс дозволяє користувачам реєструватися, створювати профілі, створювати групи тестів, проходити тести з різними типами запитань (одна правильна відповідь, множинний вибір), переглядати результати та аналітику. Автори тестів отримують доступ до статистики, а учасники — до історії проходження та детального зворотного зв'язку. Забезпечено базовий рівень безпеки (аутентифікація, шифрування паролів, захист від типових атак).

Результатом роботи є готова до використання онлайн-платформа, яка може бути легко розгорнута в навчальному середовищі або доопрацьована до повноцінного комерційного продукту.

Подальше вдосконалення платформи доцільно здійснювати за такими напрямками:

- реалізація адаптивного тестування для персоналізації складності запитань;
- розширення аналітики результатів (середній бал, динаміка, слабкі теми);
- підтримка нових типів запитань, зокрема інтерактивних;
- створення мобільної версії платформи;

Таким чином, усі поставлені у роботі цілі були досягнуті, а задачі — успішно реалізовані. Дипломна робота має практичну цінність і може бути використаний у навчальному процесі, підготовці до іспитів, оцінюванні персоналу та в інших сферах, де потрібне швидке й ефективне тестування знань.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Django documentation. URL: <https://docs.djangoproject.com/> (дата звернення: 30.03.2025).
2. Figma. URL: <https://www.figma.com/> (дата звернення: 10.03.2025).
3. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. — Addison-Wesley, 1994. — 395 p.
4. Google Forms. URL: <https://workspace.google.com/products/forms/> (дата звернення: 09.03.2025).
5. IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. — IEEE, 1998. — 25 p.
6. ISO/IEC 12207:2017. Systems and software engineering — Software life cycle processes. — Geneva: ISO, 2017.
7. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. — Geneva: ISO, 2011.
8. Kahoot!. URL: <https://kahoot.com/> (дата звернення: 09.03.2025).
9. McConnell S. Code Complete: A Practical Handbook of Software Construction. — Microsoft Press, 2004. — 960 p.
10. Moodle. URL: <https://moodle.org/> (дата звернення: 09.03.2025).
11. Mozilla Developer Network (MDN Web Docs). URL: <https://developer.mozilla.org/> (дата звернення: 29.03.2025).
12. OMG Unified Modeling Language (OMG UML), Version 2.5.1. URL: <https://www.omg.org/spec/UML> (дата звернення: 14.03.2025).
13. ProProfs Quiz Maker. URL: <https://www.proprofs.com/quiz-school/> (дата звернення: 09.03.2025).
14. Quizlet. URL: <https://quizlet.com/ua> (дата звернення: 09.03.2025).
15. Sommerville I. Software Engineering. 10th ed. — Pearson Education, 2015. — 792 p.
16. Typeform. URL: <https://www.typeform.com/> (дата звернення: 09.03.2025).
17. W3C. Web Content Accessibility Guidelines (WCAG) 2.1. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 28.03.2025).

18. Андресюк О.М. Інформаційні технології в освіті: навч. посіб. — К.: Центр учбової літератури, 2012. — 224 с.
19. Бондаренко С.М. Основи розробки веб-застосунків: навч. посіб. — Харків: ХНУ імені В.Н. Каразіна, 2018. — 172 с.
20. Васильєва С.О. Освітні інформаційні технології: навч. посіб. — Х.: ХНУ імені В.Н. Каразіна, 2020. — 312 с.
21. Виговська Г.І. Використання онлайн-тестування у навчальному процесі: переваги та виклики // Інформаційні технології і засоби навчання. — 2020. — №3(77). — С. 24–33.
22. Глушко В.В., Биков В.Ю. Хмаро орієнтовані платформи навчання: стан і перспективи розвитку // Інформаційні технології і засоби навчання. — 2017. — №1(57). — С. 1–13.
23. Денісова О.О. Методичні матеріали з написання та захисту курсової роботи з дисципліни "Системи автоматизованого інжинірингу". — К.: КНЕУ, 2022. — 28 с.
24. Жалдак М.І., Молчанов С.В. Основи інформатики. — К.: Вища школа, 2002. — 340 с.
25. Крамаренко Т.Г., Гринюк О.І., Ткаченко І.М. Інформаційні технології навчання: навч. посібник. — К.: Педагогічна думка, 2016. — 192 с.
26. Мартін Р. Чистий код: створення, аналіз та рефакторинг. — К.: Видавнича група BHV, 2021. — 448 с.
27. Ситник В.Ф., Писаревська Т.А., Єрьоміна Н.В., Краєва О.С. Основи інформаційних систем: навч. посіб. — 2-ге вид., перероб. і доп. — К.: КНЕУ, 2001. — 420 с.
28. Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методичний посібник / За заг. ред. І.А. Зязюна, Н.Г. Ничкало. — К.: Міленіум, 2012. — 396 с.
29. Тішков Б.О. Системний аналіз інформаційних технологій. — К.: КНЕУ, 2019. — 280 с.
30. Фаулер М. Архітектура корпоративних програм. — К.: Діалектика, 2020. — 496 с.

## ДОДАТКИ

### Додаток А

#### Абстрактна модель

```
class TimeStampedModel(models.Model):

    class Meta:
        abstract = True

    created_at = models.DateTimeField(auto_now_add=True, editable=False)
    updated_at = models.DateTimeField(auto_now=True)
```

#### Моделі Тестів

```
class Quiz(TimeStampedModel):
    company = models.ForeignKey(Company, on_delete=models.CASCADE, blank=True)
    title = models.CharField(max_length=255)
    description = models.TextField()
    recommended_frequency = models.PositiveIntegerField()

class Question(TimeStampedModel):
    quiz = models.ForeignKey(Quiz, on_delete=models.CASCADE,
related_name="questions")
    question = models.TextField()
    is_multiple_answers = models.BooleanField(default=False)

class Option(TimeStampedModel):
    question = models.ForeignKey(Question, on_delete=models.CASCADE,
related_name="options")
    option = models.TextField()
    is_correct_answer = models.BooleanField(default=False)

    def save(self, *args, **kwargs):
        if self.is_correct_answer and not self.question.is_multiple_answers:
            options = Option.objects.filter(question=self.question,
is_correct_answer=True)
            if options.exists() and not self.pk:
                raise ValidationError(
                    _("This question is not multiple answer. There is already a
correct answer for this question."))

        super().save(*args, **kwargs)

class TestResult(TimeStampedModel):
    user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="test_results")
    company = models.ForeignKey(Company, on_delete=models.CASCADE,
related_name="test_results")
    quiz = models.ForeignKey(Quiz, on_delete=models.CASCADE,
related_name="test_results")
    score = models.FloatField()
    total_questions = models.PositiveIntegerField()
    correct_answers = models.PositiveIntegerField()

    class Meta:
        ordering = ['-created_at']

    @staticmethod
    def average_score_within_company(user, company):
        average_score = TestResult.objects.filter(user=user,
```

```

company=company).aggregate(Avg('score'))['score__avg']
    return average_score or 0

    @staticmethod
    def average_score_across_system(user):
        average_score =
        TestResult.objects.filter(user=user).aggregate(Avg('score'))['score__avg']
        return average_score or 0

class UserQuizStatus(TimeStampedModel):
    user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="quiz_statuses")
    quiz = models.ForeignKey(Quiz, on_delete=models.CASCADE,
related_name="user_statuses")
    last_taken_time = models.DateTimeField(null=True, blank=True)
    average_score = models.FloatField(default=0.0)
    total_attempts = models.PositiveIntegerField(default=0)

    class Meta:
        unique_together = ("user", "quiz")

```

## Моделі груп тестів

```

class Company(TimeStampedModel):
    class Meta:
        verbose_name = _("Company")
        verbose_name_plural = _("Companies")

    owner = models.ForeignKey(User, on_delete=models.CASCADE, blank=True)
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True, null=True)
    is_visible = models.BooleanField(default=True)

    def __str__(self):
        return self.name

class CompanyUser(TimeStampedModel):
    company = models.ForeignKey(Company, on_delete=models.CASCADE,
related_name="company_users")
    user = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="user_companies")
    is_administrator = models.BooleanField(default=False)

    class Meta:
        unique_together = ("company", "user")

    def __str__(self):
        return f"{self.user.username} in {self.company}"

class Interaction(TimeStampedModel):
    class Type(models.TextChoices):
        INVITATION = "INV", _("Invitation")
        REQUEST = "REQ", _("Request")

    class Status(models.TextChoices):
        PENDING = "PEN", _("Pending")
        APPROVED = "APP", _("Approved")
        DECLINED = "DEC", _("Declined")
        REVOKED = "REV", _("Revoked")

    type = models.CharField(max_length=3, choices=Type.choices)
    status = models.CharField(max_length=3, choices=Status.choices,
default=Status.PENDING, blank=True)

```

```
company = models.ForeignKey(Company, on_delete=models.CASCADE)
user = models.ForeignKey(User, on_delete=models.CASCADE)
initiator = models.ForeignKey(User, on_delete=models.CASCADE,
related_name="initiator", blank=True)

def __str__(self):
    return f"{self.user.username} {self.get_type_display()} to {self.company} "
    ({self.get_status_display()})"
```



Дата звіту 6/10/2025  
Дата редагування ---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації  
**Kyiv National Economic University named after Vadym Hetman KNEU**  
Заголовок  
**Розробка платформи для онлайн-тестування**  
Автор  
Науковий керівник / Експерт  
**Заболотін Андрій ОлександровичКривошеєв К.В.**  
підрозділ  
**кафедра інформаційних систем в економіці**

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

Довжина фрази для коефіцієнта подібності 2



КП 2

7107

Кількість слів



КЦ

55404

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

|                        |  |    |
|------------------------|--|----|
| Заміна букв            |  | 1  |
| Інтервали              |  | 0  |
| Мікропробіли           |  | 1  |
| Білі знаки             |  | 0  |
| Парафрази (SmartMarks) |  | 35 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

| 10 найдовших фраз   |                                                                                                                                                                                             |                                           | Колір тексту |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------------|
| ПОРЯДКОВИЙ<br>НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)                                                                                                                                                    | КІЛЬКІСТЬ ІДЕНТИЧНИХ<br>СЛІВ (ФРАГМЕНТІВ) |              |
| 1                   | +ДОНЕЦЬ_ogurman72@gmail.com_Євтушенко МП 2024<br>12/3/2024<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (IATE, К-ра цифрових технології в енергетиці) | 74                                        | 1.04 %       |
| 2                   | +ДОНЕЦЬ_ogurman72@gmail.com_Євтушенко МП 2024<br>12/3/2024<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (IATE, К-ра цифрових технології в енергетиці) | 73                                        | 1.03 %       |

|    |                                                                                                                                                                                                             |           |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 3  | +ДОНЕЦЬ_ogurman72@gmail.com_Євтушенко МП 2024<br>12/3/2024<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (IATE, К-ра цифрових технологій в енергетиці)                 | 65 0.91 % |
| 4  | +ДОНЕЦЬ_ogurman72@gmail.com_Євтушенко МП 2024<br>12/3/2024<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (IATE, К-ра цифрових технологій в енергетиці)                 | 61 0.86 % |
| 5  | <a href="https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content">https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content</a> | 53 0.75 % |
| 6  | <a href="https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download">https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download</a>                               | 43 0.61 % |
| 7  | <a href="https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download">https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download</a>                               | 40 0.56 % |
| 8  | <a href="https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download">https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download</a>                               | 29 0.41 % |
| 9  | <a href="https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content">https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content</a> | 26 0.37 % |
| 10 | <a href="https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content">https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content</a> | 26 0.37 % |

## з бази даних RefBooks (0.00 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|-----------|----------------------------------------|
|------------------|-----------|----------------------------------------|

## з домашньої бази даних (0.00 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|-----------|----------------------------------------|
|------------------|-----------|----------------------------------------|

## з програми обміну базами даних (7.87 %)

| ПОРЯДКОВИЙ НОМЕР | ЗАГОЛОВОК                                                                                                                                                                                         | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 1                | +ДОНЕЦЬ_ogurman72@gmail.com_Євтушенко МП 2024<br>12/3/2024<br>National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (IATE, К-ра цифрових технологій в енергетиці)       | 289 (5) 4.07 %                         |
| 2                | Проектування та програмування інформаційної системи управління ОСББ<br>12/8/2023<br>Kryvyi Rih National University (Кафедра автоматизації, комп'ютерних наук і технологій)                        | 230 (26) 3.24 %                        |
| 3                | ФКПІ_2024_121_Собко_Я.С<br>7/11/2024<br>Ukrainian national aviation university (Ukrainian national aviation university)                                                                           | 23 (3) 0.32 %                          |
| 4                | Поліщук О. С. група_242з дипломна робота _Інженерія програмного забезпечення_ ф. _Інформаційних систем та технологій н.кер. Фролов І.М.<br>4/15/2025<br>European University (European University) | 17 (3) 0.24 %                          |

## з Інтернету (8.70 %)

| ПОРЯДКОВИЙ НОМЕР | ДЖЕРЕЛО URL                                                                                                                                                                                                 | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 1                | <a href="https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content">https://ir.kneu.edu.ua/server/api/core/bitstreams/1b807d60-145f-44df-a17e-e46667c96544/content</a> | 159 (8) 2.24 %                         |
| 2                | <a href="https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download">https://ir.kneu.edu.ua/bitstreams/b20c497c-320d-46ce-8d1d-a56f3c2eec3f/download</a>                               | 153 (10) 2.15 %                        |
| 3                | <a href="https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download">https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download</a>                               | 143 (8) 2.01 %                         |
| 4                | <a href="https://ela.kpi.ua/jspui/bitstream/123456789/51915/1/Strukov_magistr.docx">https://ela.kpi.ua/jspui/bitstream/123456789/51915/1/Strukov_magistr.docx</a>                                           | 82 (8) 1.15 %                          |