

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПРОВЕДЕННЯ ОПИТУВАНЬ І
ГОЛОСУВАНЬ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ З АНАЛІЗОМ
РЕЗУЛЬТАТІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ**

**DEVELOPMENT OF A WEB APPLICATION FOR REAL-TIME SURVEYS
AND VOTING WITH RESULT ANALYSIS USING NEURAL NETWORKS**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ПЗс-21
Мартинюк Богдан Ярославович

Керівник:
Падалко Анатолій Михайлович

Кваліфікаційну роботу
допущено до захисту

«10» 06 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри


«20» 12 2024р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Мартинюку Богдану Ярославовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: «Розробка вебдодатку для проведення опитувань і голосувань у режимі реального часу з аналізом результатів за допомогою нейронних мереж».

Керівник роботи: к.фіз.-мат.н., доцент Падалко А. М.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

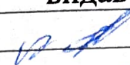
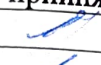
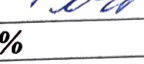
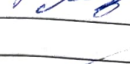
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: PHP, MySQL, RedBeanPHP, JavaScript, jQuery, HTML/CSS, AnyChart, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз предметної області та постановка задачі, визначення вимог до вебдодатку, специфікація функціоналу, вибір технологій та інструментів реалізації, проєктування структури бази даних, розробка та тестування функціональних модулів, оцінка безпеки системи.

5. Перелік графічного матеріалу: 9 рисунків, 3 таблиці.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Падалко А. М.		
Специфікація вимог до розробленої системи	Падалко А. М.		
Розробка об'єкта проектування	Падалко А. М.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	0,67 %		
Академічна доброчесність	Падалко А. М.		

7. Дата видачі завдання «20» грудня 2024 р.

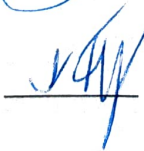
№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкта проектування	до 01.03.2025 р.	вик.
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	вик.
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	вик.
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	вик.

Здобувач вищої освіти



Богдан МАРТИНЮК

Керівник кваліфікаційної роботи



Анатолій ПАДАЛКО

АНОТАЦІЯ

Мартинюк Б. Я. Розробка вебдодатку для проведення опитувань і голосувань у режимі реального часу з аналізом результатів за допомогою нейронних мереж. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі здійснено аналіз предметної області, розглянуто існуючі аналоги систем опитування та сформульовано завдання на кваліфікаційну роботу. У другому розділі наведено специфікацію вимог, UML-моделі та обґрунтовано вибір засобів і методів реалізації. У третьому розділі описано практичну реалізацію вебдодатку, проведено тестування, розроблено базу даних і проаналізовано захист системи. У висновках підбито підсумки та окреслено напрями подальшого вдосконалення.

Ключові слова: вебдодаток, опитування, голосування, PHP, MySQL, RedBeanPHP, JavaScript.

ABSTRACT

Martyniuk B. Development of a Web Application for Real-time Surveys and Voting with Result Analysis Using Neural Networks. Manuscript. Qualification work of the bachelor's program "Software Engineering" in the specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions and a list of references.

The first chapter presents the analysis of the subject area, reviews existing solutions for online surveys, and formulates the objectives of the project. The second chapter provides the software requirements specification, UML models, and justification of the selected tools and methods. The third chapter describes the implementation of the web application, including testing, database development, and system security analysis. The conclusions summarize the results and outline future improvement directions.

Keywords: web application, surveys, polls, PHP, MySQL, RedBeanPHP, JavaScript.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СИСТЕМ ОНЛАЙН-ОПИТУВАНЬ І ГОЛОСУВАНЬ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	12
Висновки до розділу 1	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОГО МОБІЛЬНОГО ДОДАТКУ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	18
Висновки до розділу 2	22
РОЗДІЛ 3 РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПРОВЕДЕННЯ ОПИТУВАНЬ І ГОЛОСУВАНЬ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ	24
3.1 Практична реалізація об'єкта проектування	24
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	28
3.3 Розробка бази даних	30
3.4 Захист інформаційно-комп'ютерної системи	32
Висновки до розділу 3	34
ВИСНОВКИ	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

ВСТУП

Актуальність теми. У сучасному цифровому середовищі, де інтерактивність, оперативність і масове залучення користувачів стають основними чинниками ефективної комунікації, зростає потреба у вебдодатках, що дозволяють проводити опитування та голосування в режимі реального часу. Такі системи широко використовуються в освіті, наукових дослідженнях, громадських ініціативах, медіа та бізнесі для швидкого збору зворотного зв'язку, ухвалення рішень або оцінки думки аудиторії. Водночас більшість існуючих платформ мають обмеження у функціональності, масштабованості, адаптації під мобільні пристрої або забезпеченні захисту даних. Тому розробка нового, адаптивного і захищеного вебдодатку є актуальним напрямом у сфері інформаційних технологій.

Оцінка сучасного стану проблеми свідчить, що ринок пропонує чимало інструментів для голосувань (Google Forms, Slido, Kahoot, Poll Everywhere тощо), однак більшість з них не підтримують режим реального часу на рівні обробки великих обсягів даних, не дозволяють гнучко управляти правами доступу та мають обмежений функціонал інтеграції з внутрішніми системами організацій. Також спостерігається відсутність українських продуктів з відкритим вихідним кодом, що орієнтовані на потреби навчальних закладів, громадських ініціатив і бізнесу. У світовій практиці поширеним є підхід до використання вебтехнологій з WebSocket, REST API, шифруванням і адаптивною версткою для вирішення аналогічних задач, що демонструє потенціал цієї галузі для подальших розробок.

Отже, метою даної кваліфікаційної роботи є створення вебдодатку для проведення опитувань і голосувань у режимі реального часу, який дозволяє в зручному інтерфейсі керувати опитуваннями, збирати та аналізувати відповіді, забезпечуючи інтерактивність, швидкість і безпеку даних.

Об'єктом кваліфікаційної роботи є процес організації та реалізації веборієнтованих інформаційних систем для збору та обробки думок користувачів.

Предметом кваліфікаційної роботи виступають програмні методи, інструменти, архітектури та технології, що забезпечують функціонування вебдодатку для голосувань у режимі реального часу.

Для досягнення поставленої мети були визначені наступні завдання:

- 1) аналіз існуючих рішень у сфері онлайн-голосувань і опитувань;
- 2) формулювання функціональних і нефункціональних вимог до системи, проєктування архітектури вебдодатку та інформаційних потоків;
- 3) вибір інструментів розробки;
- 4) розробка інтерфейсу користувача з урахуванням принципів UX/UI-дизайну, реалізація серверної та клієнтської частин програмного забезпечення;
- 5) тестування працездатності системи;
- 6) створення бази даних та реалізація логіки збереження результатів;
- 7) забезпечення безпеки системи та контроль доступу.

Розроблений вебдодаток призначений для використання в закладах освіти, громадських організаціях, державних структурах і приватному бізнесі як ефективний інструмент для збирання думок, проведення голосувань, швидкого аналізу результатів та прийняття рішень на основі зібраної інформації. Сфера застосування охоплює онлайн-опитування серед студентів, аудиторій під час вебінарів, клієнтів компаній, працівників підприємств тощо.

Взаємозв'язок з іншими роботами полягає у використанні сучасних підходів веброзробки, інтеграції з базами даних, реалізації авторизації користувачів та забезпечення інформаційної безпеки, що має практичне значення для побудови інших інформаційно-комунікаційних систем у реальному часі.

РОЗДІЛ 1

АНАЛІЗ СИСТЕМ ОНЛАЙН-ОПИТУВАНЬ І ГОЛОСУВАНЬ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасних умовах цифровізації все більше сфер суспільного життя потребують ефективних засобів для оперативного збирання, обробки та аналізу думок великої кількості користувачів. Одним із найбільш зручних інструментів для таких завдань є вебдодатки для проведення опитувань і голосувань у режимі реального часу. Ці рішення знаходять широке застосування в освіті, політиці, громадських ініціативах, бізнесі, HR-сфері, маркетингових дослідженнях і не тільки.

Проведене дослідження спирається на сучасні наукові джерела, присвячені технологічним, правовим та організаційним аспектам електронного голосування, що дозволяє сформулювати комплексне уявлення про сучасний стан і перспективи розвитку таких систем.

У роботах S. Al-Maaitah та співавторів [1] і U. Jafar та співавторів [2] акцент зроблено на потенціалі використання блокчейн-технологій у системах електронного голосування. Зокрема, підкреслюється висока стійкість до фальсифікацій, забезпечення прозорості та можливість перевірки результатів голосування без порушення анонімності. Водночас автори звертають увагу на складність масштабування таких рішень та потребу в гармонізації технічних і правових аспектів впровадження. Аналогічну позицію поділяють M.-V. Vladucu та співавтори [3], які в оглядовій роботі систематизують переваги і виклики інтеграції блокчейну в e-voting, підкреслюючи важливість підвищення довіри до технологій серед користувачів.

У національному контексті, А. А. Котобан [4] представив власну реалізацію клієнт-серверного застосунку для локального голосування, який ілюструє практичну реалізацію архітектури з базовим функціоналом. Його розробка вказує на технічну доцільність створення децентралізованих систем

голосування навіть у масштабах муніципального рівня, проте водночас засвідчує обмеження щодо масштабованості та захисту від зовнішніх втручань. У статті М. В. Кухаренка [5] розглядаються трансформації публічного адміністрування під впливом цифровізації, зокрема впровадження цифрових інструментів участі громадян у демократичних процесах. Автор підкреслює, що електронне голосування є важливим інструментом диджитал-демократії, здатним посилити прозорість та інклюзивність управлінських рішень.

Питання безпеки, зокрема забезпечення анонімності користувачів під час онлайн-опитувань, ґрунтовно розкриті у статті І. В. Олешка та К. О. Папазова [6]. Дослідники наголошують на загрозах deanonymization-атак та пропонують підходи до розробки систем, які мінімізують ризики витоку персональних даних за рахунок використання криптографічних протоколів та технологій тунелювання. Коротко у дисертації М. А. Приймак [7] акцентується увага на ролі цифрових платформ у підвищенні участі громадськості в управлінні публічними фінансами. Автор вказує на потенціал інтерактивних вебрішень, у тому числі систем онлайн-опитувань, як інструментів електронної демократії.

В цілому, проаналізовані джерела демонструють, що впровадження сучасних систем голосування є перспективним і затребуваним напрямом, що поєднує інтереси користувачів, потребу у прозорості рішень та розвиток інформаційних технологій. Водночас зазначаються значні виклики у сфері безпеки, масштабування, а також збереження анонімності, що має бути враховано при проектуванні нових рішень.

На ринку представлені низка популярних рішень, зокрема:

1) Google Forms – проста у використанні платформа для опитувань, однак не підтримує режиму реального часу при зборі відповідей [8];

2) Slido – інтерактивний інструмент для голосувань і Q&A на заходах, проте обмежена функціональність у безкоштовній версії [9];

3) Kahoot – зручна система для освітніх цілей, орієнтована переважно на гейміфіковані тести, не пристосована для серйозних аналітичних досліджень [10];

4) Mentimeter – підтримує візуалізацію даних у реальному часі, але має закритий код і обмеження на кількість запитань [11];

5) StrawPoll, Poll Everywhere, Typeform – мають обмежені сценарії використання, часто не адаптовані під специфічні потреби організацій.

З патентного погляду, аналіз баз даних USPTO, EPO та Укрпатенту показує наявність зареєстрованих технічних рішень, які фокусуються на системах анонімного голосування, використанні блокчейну або біометрії, однак практичне їх застосування часто ускладнене високою вартістю реалізації або недостатньою інтеграцією з вебтехнологіями.

Серед основних переваг сучасних вебсистем для проведення опитувань і голосувань варто відзначити зручність у користуванні, наявність адаптивного інтерфейсу, базову інтеграцію з хмарними сервісами, а також можливість візуалізації результатів у зрозумілому вигляді. Водночас ці рішення мають і низку суттєвих недоліків, зокрема відсутність повної локалізації українською мовою, обмеження у налаштуванні логіки формування питань, недостатню підтримку розширених прав доступу для різних ролей користувачів, зменшений функціонал у безкоштовних версіях, а також відсутність можливості роботи в офлайн-режимі або при нестабільному інтернет-з'єднанні.

У зв'язку з цим виникає об'єктивна потреба у створенні власного вебдодатку, орієнтованого на зручне адміністрування, адаптацію до різних платформ, гнучке налаштування структури опитувань, захист даних, обробку результатів у режимі реального часу та збереження результатів у захищеній базі даних. Це дозволить забезпечити гнучкий та ефективний інструмент для широкого кола користувачів: від викладачів і студентів до організаторів публічних заходів та представників малого бізнесу.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

З урахуванням актуальності теми, аналізу предметної області та недоліків існуючих рішень, у межах кваліфікаційної роботи визначено низку завдань, кожне з яких відповідає відповідному етапу реалізації проєкту та підрозділу пояснювальної записки:

1) провести аналіз сучасного стану розробок у сфері веборієнтованих систем для проведення опитувань та голосувань;

2) сформулювати технічні та функціональні вимоги до системи, визначити архітектуру, сценарії використання та специфікацію програмного продукту;

3) провести обґрунтований вибір інструментів, бібліотек і технологій реалізації вебдодатку;

4) розробити вебдодаток, який дозволяє створювати та адмініструвати опитування, приймати голоси та відображати результати в режимі реального часу;

5) здійснити тестування та налагодження програмного продукту з урахуванням функціональних і нефункціональних вимог;

6) розробити схему та реалізацію бази даних для зберігання інформації про користувачів, опитування, варіанти відповідей і результати голосування;

7) реалізувати базові механізми захисту інформаційно-комп'ютерної системи.

Усі ці завдання є частинами комплексного процесу створення повноцінної вебсистеми і повинні бути виконані у межах кваліфікаційної роботи бакалавра для досягнення поставленої мети.

Висновки до розділу 1

У результаті проведеного аналізу предметної області було встановлено, що вебдодатки для проведення опитувань і голосувань у режимі реального часу

мають високу актуальність та широкий спектр застосування в освітньому, соціальному, політичному та комерційному середовищі. Огляд літературних джерел, патентних розробок і існуючих аналогів (Google Forms, Slido, Mentimeter, Kahoot тощо) дозволив окреслити основні функціональні можливості таких систем, а також виявити їх переваги та недоліки.

До позитивних характеристик сучасних вебсистем слід віднести простоту у використанні, адаптивний інтерфейс, підтримку хмарних сервісів та зручну візуалізацію результатів. Водночас, значними недоліками залишаються відсутність повної локалізації українською мовою, обмеження у налаштуванні логіки питань, обмежена підтримка ролей користувачів, урізаний функціонал безкоштовних версій, а також неможливість стабільної роботи при слабкому або відсутньому інтернет-з'єднанні.

Загалом, усе це свідчить про наявність вільної ніші на ринку вебзастосунків для опитувань, що відкриває перспективи для розробки нового функціонального продукту. Обґрунтована доцільність створення власного вебдодатку полягає у реалізації розширених можливостей адміністрування, гнучкого управління структурою питань, інтерактивного відображення результатів, підтримки режиму реального часу, високого рівня захисту та повної адаптації під україномовного користувача. Результатом цього етапу стало формулювання конкретних завдань, які необхідно реалізувати в межах кваліфікаційної роботи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОГО МОБІЛЬНОГО ДОДАТКУ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування

Для моделювання архітектури вебдодатку для проведення опитувань і голосувань у режимі реального часу було обрано модель MVC (Model–View–Controller). Вона забезпечує логічне розділення бізнес-логіки, користувацького інтерфейсу та доступу до даних. Це дозволяє ефективно підтримувати, розширювати та масштабувати систему. Для візуального представлення компонентів програмної системи було використано UML-нотацію, зокрема діаграми прецедентів, класів та послідовностей.

Аналіз вимог здійснювався шляхом:

- 1) вивчення аналогічних рішень (Google Forms, Slido, Mentimeter);
- 2) визначення потреб цільових груп користувачів (адміністратор/відвідувач);
- 3) проведення умовного SWOT-аналізу функціоналу;
- 4) визначення обмежень середовища (веб-доступність, підтримка мобільних пристроїв, базова безпека).

В таблиці 2.1 наведено формалізацію функціональних вимог.

Таблиця 2.1 – Формалізація функціональних вимог

Функція	Короткий опис
Реєстрація користувача	Заповнення анкети з демографічними даними, збереження в базі даних
Авторизація	Перевірка облікових даних, вхід до особистого кабінету
Перегляд доступних опитувань	Вивід списку активних тестів з можливістю їх проходження
Проходження опитування	Вибір варіантів відповіді, збереження результату у форматі JSON
Перегляд історії результатів	Персоналізована статистика користувача
Візуалізація зведених результатів	Побудова діаграм на основі всіх зібраних відповідей
Адміністративне керування тестами та запитаннями	Створення, редагування, видалення тестів та запитань

Продовження таблиці 2.1

Функція	Короткий опис
Прогнозування типів користувачів	Алгоритмічна класифікація на основі даних при реєстрації
Завантаження зображень	Додавання графіки до запитань у форматі .jpg/.png

Також для проєктування системи створені відповідні UML-моделі. Діаграма прецедентів (рис. 2.1) охоплює два актори: Користувач та Адміністратор. Вона ілюструє основні сценарії взаємодії: реєстрація, проходження тестів, перегляд результатів, створення запитань тощо.

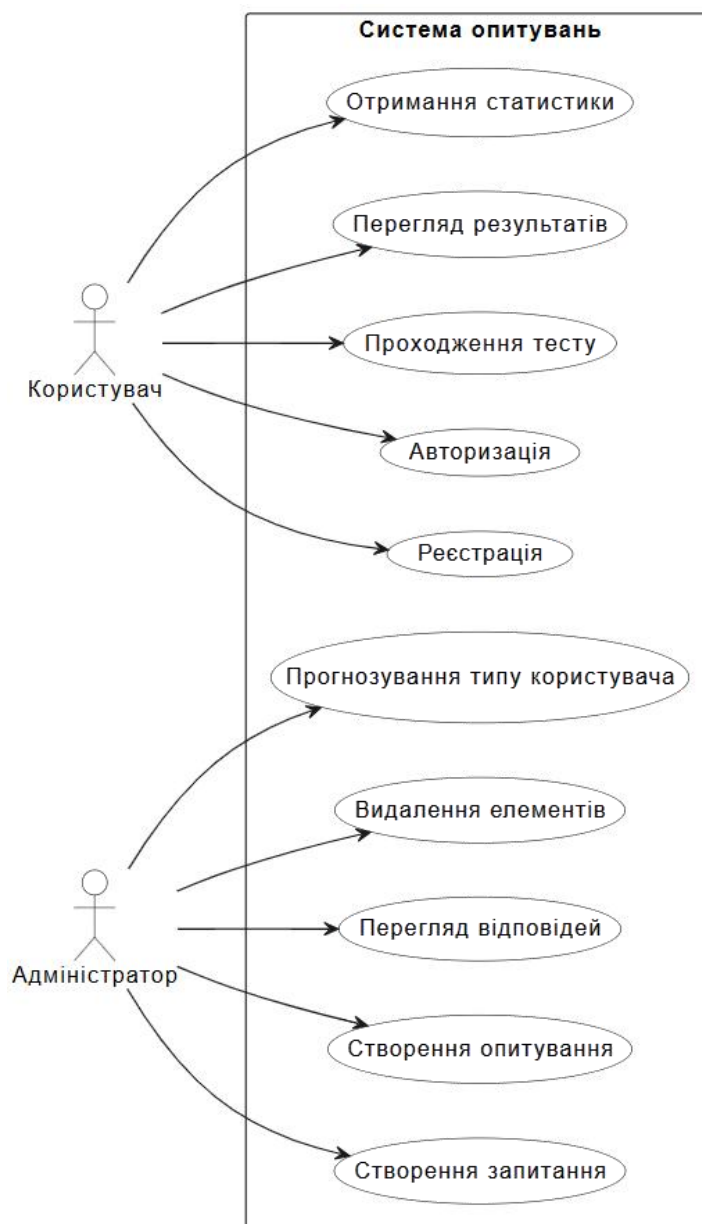


Рисунок 2.1 – Діаграма прецедентів

Діаграма класів (рис. 2.2) демонструє об'єкти User, Ticket, Question, Stat, включаючи їх атрибути та зв'язки типу «один-до-багатьох».

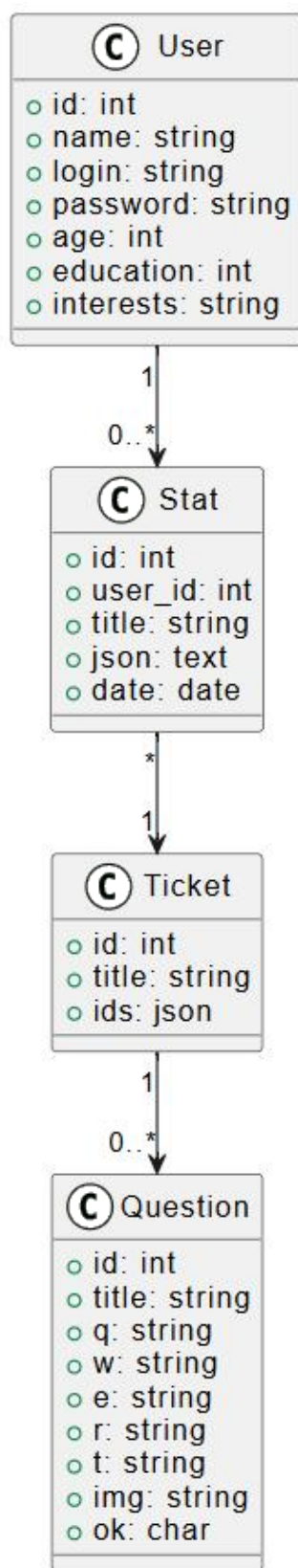


Рисунок 2.2 – Діаграма класів

Діаграма послідовностей (рис. 2.3) відображає процес проходження тесту: від відкриття списку до збереження відповіді через AJAX.

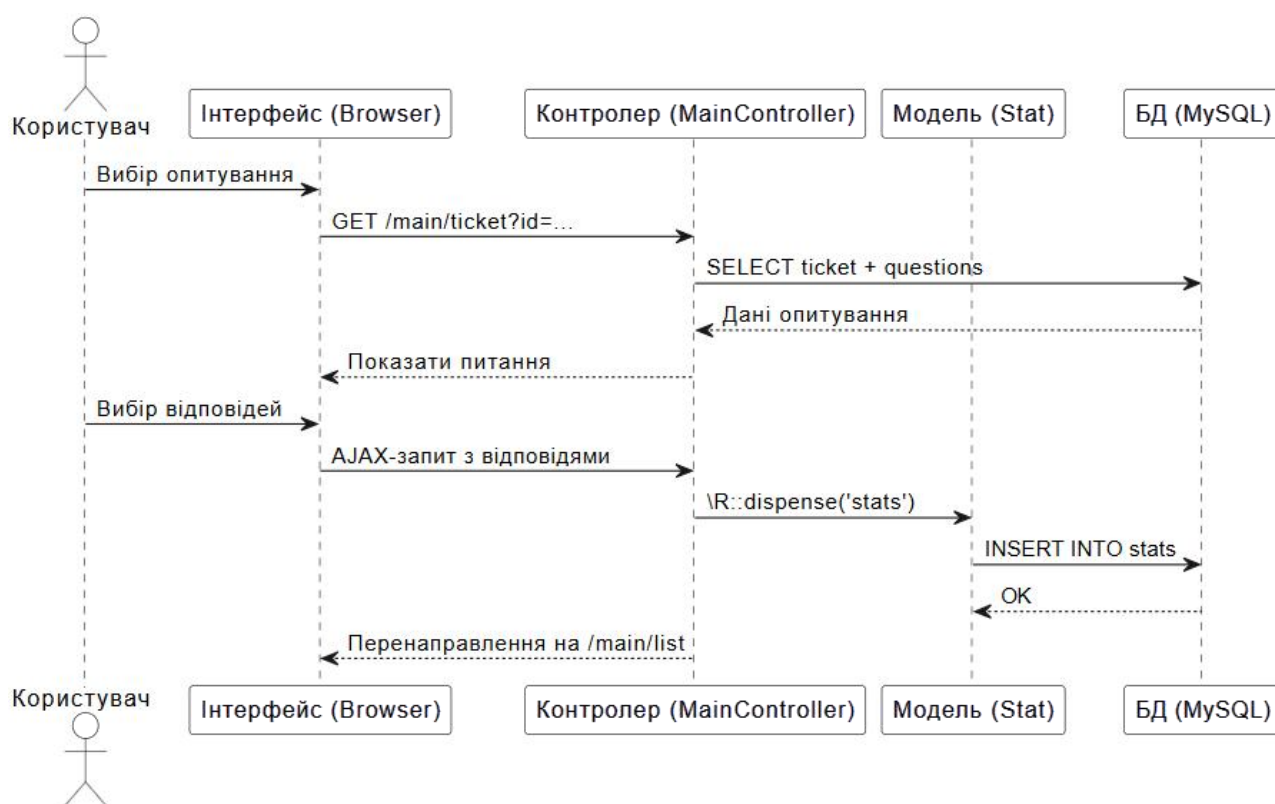


Рисунок 2.3 – Діаграма послідовностей

В таблиці 2.2 наведено специфікацію вимог до програмного забезпечення.

Таблиця 2.2 – Специфікація вимог до ПЗ

Категорія	Вимоги
Платформа	Працює у браузері з підтримкою JavaScript. Серверна частина на PHP 7.4+
Інтерфейс	Інтуїтивно зрозумілий, адаптивний, двомовний (укр/англ), побудований з HTML + CSS + JS
Безпека	Авторизація, хешування паролів, захист від SQL-ін'єкцій, розмежування прав доступу
Інформаційна цілісність	Валідація введення, перевірка унікальності логінів, контроль допустимих форматів даних
Збереження даних	База MySQL, взаємодія через ORM RedBeanPHP
Інформаційне обслуговування	Побудова графіків у форматі AnyChart, персоналізована статистика

Інтерфейс проєкту реалізовано за принципами простоти, логічної структури та доступності. Навігаційне меню розміщене у верхній частині,

відображаючи поточний статус користувача (авторизований/гість). Всі форми реєстрації, входу, створення тестів мають підказки, валідацію, стилізовані кнопки. Потоки даних (від введення до збереження і візуалізації) розділено на:

- 1) вхідний потік – дані, які вводяться користувачем (відповіді, анкета);
- 2) обробка – логіка контролера, обробка POST-запиту;
- 3) збереження – ORM-запис у базу даних;
- 4) виведення – відображення на сторінці користувача/адміністратора або діаграма.

У результаті аналізу вимог до розроблюваного ПЗ було обґрунтовано вибір архітектурної моделі MVC, застосовано UML для структурування системи, визначено повний перелік функціональних характеристик та сформульовано вимоги до інтерфейсу, логіки та безпеки. Розроблена специфікація забезпечує чітке бачення цілей і завдань майбутньої системи, а також сприяє її ефективному впровадженню.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для реалізації вебдодатку для проведення опитувань і голосувань у режимі реального часу було здійснено детальний огляд сучасних інструментів розробки програмного забезпечення, засобів візуалізації, зберігання даних та організації взаємодії між клієнтом і сервером. На основі аналізу можливостей, сумісності, гнучкості та швидкості реалізації було обрано найдоцільніші технології для кожного етапу створення інформаційної системи.

Для реалізації серверної логіки обрано мову PHP 7.4+ завдяки її широкій підтримці, простоті використання та ефективності у роботі з базами даних [12]. PHP добре підходить для створення MVC-додатків, взаємодії з MySQL та обробки HTTP-запитів. Переваги:

- 1) поширеність і велика кількість бібліотек;
- 2) просте підключення до БД;

3) висока продуктивність при коректному кешуванні [13].

В якості системи управління базами даних обрано MySQL, яка є стабільною, масштабованою та підтримується більшістю хостингів [14]. Створення та керування базою здійснювалося за допомогою phpMyAdmin [15] та ORM-бібліотеки. Переваги:

- 1) сумісність із PHP;
- 2) підтримка ACID;
- 3) легка інтеграція через ORM RedBeanPHP.

Для зручної роботи з БД використовується RedBeanPHP – ORM-фреймворк, що автоматично створює таблиці, якщо вони відсутні [16]. Це дозволяє значно знизити кількість SQL-коду, зосередившись на логіці.

Для реалізації інтерактивної поведінки інтерфейсу використовується JavaScript, зокрема бібліотека jQuery [17]. Вона полегшує обробку подій, виконання AJAX-запитів та динамічну зміну DOM. Функціональні можливості:

- 1) вибір варіантів відповіді в опитуванні;
- 2) відправлення результатів без перезавантаження сторінки;
- 3) рандомне заповнення тестів для прогнозів.

Для побудови діаграм використано AnyChart – потужну JavaScript-бібліотеку для візуалізації статистичних даних [18]. Додаток будує кругові, стовпчикові та лінійні діаграми на основі відповідей користувачів. Фрагмент використання наведено в лістингу 2.1.

Лістинг 2.1 – Побудова кругової діаграми з використанням бібліотеки AnyChart

```
const chart = anychart.pie(data.ageGroups);
chart.container('chart1');
chart.draw();
```

кінець лістингу 2.2

В системі реалізовано умовний алгоритм прогнозування типу користувача, що базується на значеннях віку, освіти, сімейного стану та

інтересів. Цей алгоритм дозволяє призначити користувача до певного профілю (A, B, C, D, E) з метою подальшого прогнозу його відповідей.

Також розроблено деякі UML-діаграми. Діаграма компонентів (рис. 2.4) відображає логічну структуру вебдодатку у вигляді компонентів: інтерфейс користувача, контролери, модель доступу до даних та зовнішні бібліотеки (RedBeanPHP, AnyChart). Вона демонструє, як модулі пов'язані між собою та які залежності між ними існують.

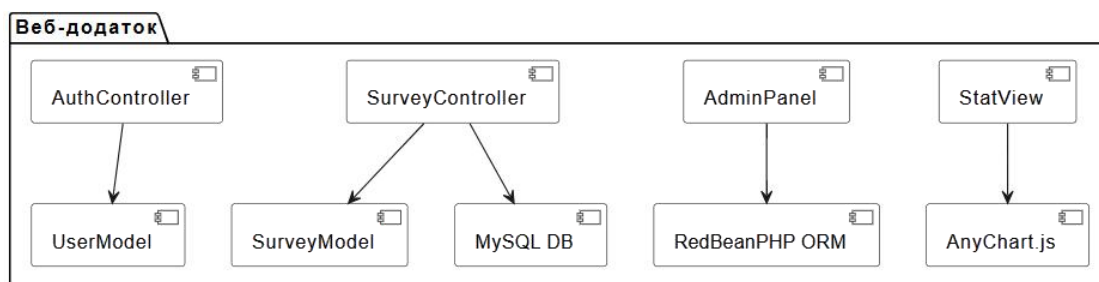


Рисунок 2.4 – Діаграма компонентів

На діаграмі розгортання (рис. 2.5) показано фізичну архітектуру системи: клієнт (браузер), вебсервер (Apache з PHP) і сервер бази даних (MySQL). Вона ілюструє, як різні вузли взаємодіють між собою у процесі опитування.

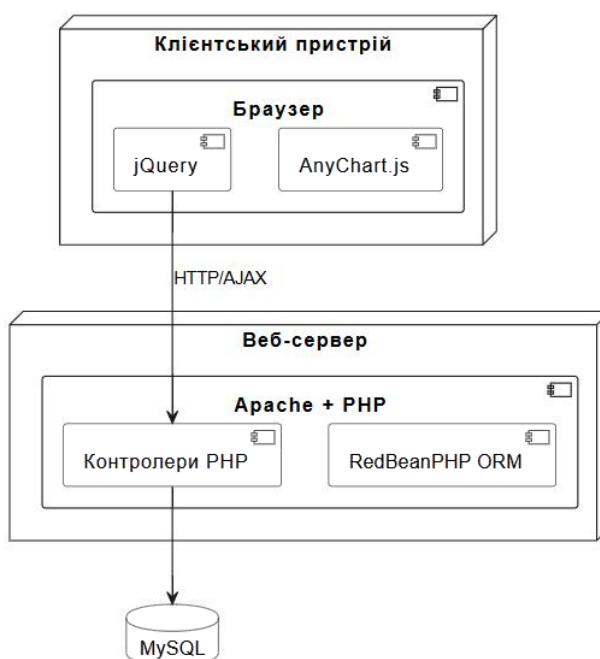


Рисунок 2.5 – Діаграма розгортання

Блок-схема алгоритму проходження опитування зображена на рисунку 2.6. Ця схема описує логіку дій користувача при проходженні опитування: від відкриття списку доступних опитувань до обробки відповіді через AJAX і збереження у базу. Вона дозволяє побачити загальну послідовність кроків і перевірок.

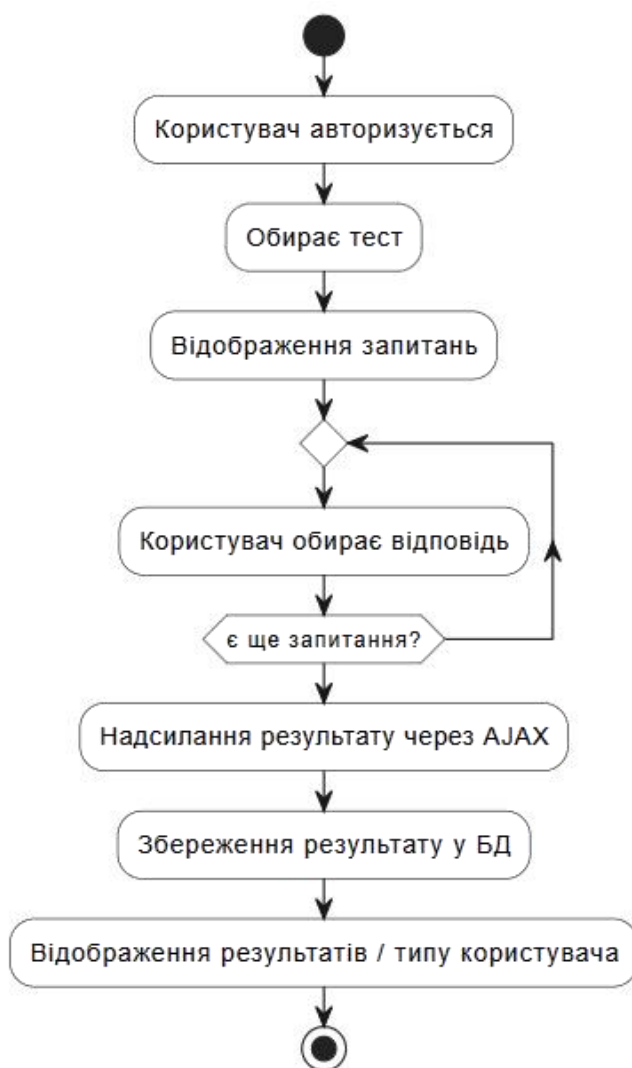


Рисунок 2.6 – Блок-схема алгоритму проходження опитування

Діаграма на рисунку 2.7 відображає процес створення нового опитування адміністратором: заповнення полів форми, вибір запитань, надсилання на сервер та формування запису в БД. Ця модель корисна для відображення логіки взаємодії інтерфейсу з бекендом.



Рисунок 2.7 – Діаграма активностей (створення опитування)

Отже, у процесі розробки було обґрунтовано вибір оптимальних технологій і засобів. Комбінація PHP, RedBeanPHP, JavaScript, AnyChart та MySQL дозволила створити повноцінну інформаційну систему з підтримкою реального часу, інтерактивного UX/UI та гнучкої обробки опитувань. Розроблені алгоритми і засоби наочно демонструють ефективність обраного технічного стеку для реалізації завдань.

Висновки до розділу 2

У другому розділі було здійснено систематичний аналіз функціональних, технічних і проектних вимог до створюваного вебдодатку для проведення опитувань і голосувань у режимі реального часу. Визначено доцільність

використання архітектурної моделі MVC, яка забезпечує ефективне розділення логіки, представлення і взаємодії з базою даних. Проведено специфікацію функціональних можливостей програмного забезпечення, зокрема реєстрацію користувачів, створення та проходження опитувань, обробку результатів і візуалізацію статистики.

Було обґрунтовано вибір технологічного стеку: PHP як основна серверна мова програмування, MySQL для збереження даних, RedBeanPHP як ORM, а також JavaScript із бібліотекою jQuery для динамічного клієнтського функціоналу. Візуалізація результатів реалізована за допомогою AnyChart, що забезпечує сучасну подачу аналітичних даних.

Для моделювання логіки та структури системи застосовано UML-нотацію, зокрема діаграми прецедентів, класів та послідовностей. Це дозволило формалізувати компоненти та взаємодії всередині системи. Описано також базовий алгоритм класифікації користувачів за поведінковими профілями, що підвищує інтелектуальність сервісу.

Загалом, розділ закладає теоретичну та методологічну основу для практичної реалізації інформаційної системи в наступному етапі розробки.

РОЗДІЛ 3

РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПРОВЕДЕННЯ ОПИТУВАНЬ І ГОЛОСУВАНЬ У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

3.1 Практична реалізація об'єкта проєктування

У процесі реалізації програмного забезпечення було створено вебдодаток із підтримкою динамічного опитування та збереження результатів у режимі реального часу. Для розробки було використано мову програмування PHP, бібліотеку RedBeanPHP для ORM-роботи з базою даних MySQL, а також мову JavaScript (jQuery) на стороні клієнта. Середовище розробки – Visual Studio Code, локальний сервер – XAMPP.

Архітектура додатку побудована за шаблоном MVC (Model-View-Controller), що забезпечує чітке розділення логіки обробки даних, відображення інформації та маршрутизації. У кореневому файлі Router.php реалізовано логіку маршрутизації запитів (ліст. 3.1).

Лістинг 3.1 – Маршрутизація

```
Router::add('^$', ['controller' => 'Main', 'action' => 'index']);
Router::add('^(?P<controller>[a-z-]+)/?(?P<action>[a-z-]+)?$');
```

кінець лістингу 3.1

Це дозволяє автоматично визначати контролер та метод, що має бути викликаний на основі URL, наприклад, /main/stat викликає метод statAction() у MainController.

Контролери зберігаються у папці controllers/. Базовий клас Controller містить загальні методи ініціалізації, передачі змінних у в'юшки та перевірку доступу. Приклад наведено в лістингу 3.2.

Лістинг 3.2 – Реалізація контролерів

```
public function getView() {
    $vObj = new View($this->route, $this->layout, $this->view);
```

```
$vObj->render($this->vars);
}
```

кінець лістингу 3.2

У AdminController реалізовані функції створення, редагування, видалення опитувань (ліст. 3.3).

Лістинг 3.3 – Фрагмент AdminController

```
public function addTicketAction() {
    $title = clear($_POST['title'], 'text');
    $arr = array_filter($_POST['arr']); // відбір обраних запитань
    $ticket = \R::dispense('tickets');
    $ticket->title = $title;
    $ticket->ids = json_encode($arr);
    \R::store($ticket);
    redirect('/admin');
}
```

кінець лістингу 3.3

Для взаємодії з базою даних використовується RedBeanPHP – легкий ORM-фреймворк. У Model.php підключено бібліотеку та налаштовано з'єднання (ліст. 3.4).

Лістинг 3.4 – Робота з базою даних

```
\R::setup('mysql:host=localhost;dbname=poll;charset=utf8', 'root', '');
\R::freeze(true); // заборона автоматичних змін структури
```

кінець лістингу 3.4

Зберігання та вибірка даних здійснюються через методи \R::dispense(), \R::store(), \R::findAll(), \R::findOne().

У файлі js.js реалізовано перевірку та відправку результатів голосування у форматі JSON (ліст. 3.5).

Лістинг 3.5 – Обробка відповідей у браузері через JavaScript

```
$(".send-res").on("click", function() {
    let answers = [];
    $(".list .item").each(function() {
        let selected = $(this).find(".ok");
```

```

    if (selected.length) {
        answers.push(selected.data("answer"));
    }
});

let listId = $(".list").data("id");
$.ajax({
    url: `/main/stat`,
    method: "GET",
    data: {
        id: listId,
        json: JSON.stringify(answers)
    }
});

window.location.href = '/main/list';
});

```

кінець лістингу 3.5

Цей скрипт забезпечує динамічний збір відповідей та надсилання їх без перезавантаження сторінки.

На сторінці `/main/chart` використовується бібліотека `AnyChart`, яка виводить діаграми (ліст. 3.6).

Лістинг 3.6 – Візуалізація даних

```

fetch('/main/json')
    .then(response => response.json())
    .then(data => {
        const chart1 = anychart.pie(data.ageGroups);
        chart1.container('chart1').draw();
    });

```

кінець лістингу 3.6

Це дозволяє отримати статистику у вигляді діаграм за віком, доходом, інтересами тощо. В свою чергу, форма підтримує збір соціодемографічних даних (ліст. 3.7).

Лістинг 3.7 – Форма реєстрації

```

<select name="age" required>
  <option value="1">До 18 років</option>

```

```
<option value="2">18-25 років</option>
</select>
```

кінець лістингу 3.7

Дані передаються до WelcomeController::registerAction(), де відбувається збереження нового користувача з автоматичним призначенням профілю за допомогою методу tCheck() (ліст. 3.8).

Лістинг 3.8 – Визначення типу користувача «Аналізатор»

```
if ($education >= 4 && strpos($interests, 'Наука') !== false) {
    return 'A'; // Аналізатор
}
```

кінець лістингу 3.8

На рисунку 3.1 представлено головний екран вебдодатку, доступний після входу користувача. Інтерфейс містить логотип системи, контактні дані, навігаційне меню з кнопками переходу до розділів «Головна», «Опитування», «Історія», «Статистика» та «Вихід». Нижче подано короткий опис функціонування системи, інструкцію для користувача та принцип роботи платформи. Дизайн адаптований для зручного ознайомлення з можливостями ресурсу.

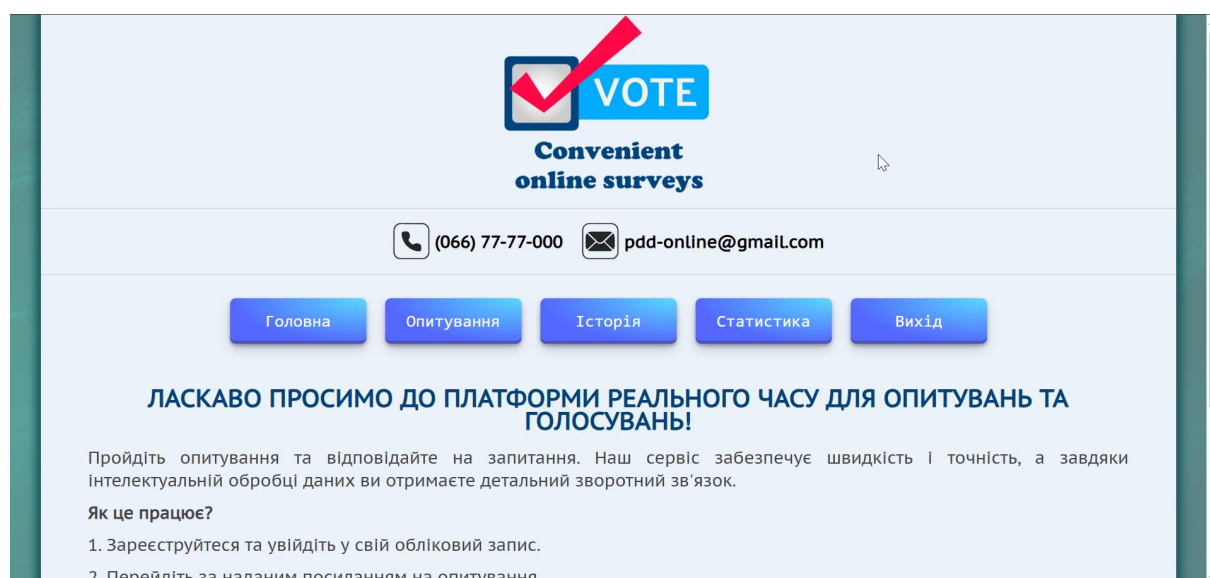


Рисунок 3.1 – Головна сторінка платформи для опитувань

Рисунок 3.2 демонструє сторінку адміністратора, на якій представлено перелік створених запитань. Кожне запитання виведено разом із варіантами відповідей. Під кожним блоком присутнє посилання «Видалити», що дозволяє адміністратору редагувати контент. Вгорі розташовано навігаційне меню до основних адміністративних функцій: перегляд опитувань, створення нових опитувань і запитань, вибір типу прогнозу та вихід із кабінету.

Опитування	Нове опитування	Запитання	Нове запитання	Тип прогнозу	Вихід
Ваш поточний рівень доходу? - Менше 10 тисяч грн - 10-20 тисяч грн - Більше 20 тисяч грн - Не хочу відповідати Видалити	Ваша основна зайнятість? - Студент - Працюю - Безробітний - На пенсії Видалити	Чи є у вас заощадження? - Так - Ні - Маю невеликі заощадження - Не хочу відповідати Видалити	Чи маєте кредити? - Так, іпотека - Так, споживчий кредит - Ні - Не хочу відповідати Видалити		
Чи задоволені ви своїм фінансовим становищем? - Цілком задоволений - Частково задоволений - Не задоволений - Не хочу відповідати Видалити	На що ви найчастіше витрачаєте кошти? - Продукти харчування - Розваги - Подорожі - Навчання - Інше Видалити	Ваш сімейний стан? - Не одружений/Не заміжня - Одружений/Заміжня - Розлучений/Розлучена - Вдовець/Вдова Видалити	Чи маєте дітей? - Ні - Одна дитина - Двоє дітей - Троє або більше Видалити		
Скільки часу ви проводите з родиною щодня? - Менше 1 години - 1-3 години	Яке основне джерело конфліктів у родині? - Фінанси - Розподіл обов'язків	Чи задоволені ви своїми сімейними стосунками? - Цілком задоволений - Частково задоволений	Ваш рівень освіти? - Середня освіта - Бакалавр		

Рисунок 3.2 – Інтерфейс керування запитаннями адміністратора

Загалом, було реалізовано вебдодаток із повним циклом обробки опитувань: створення, проходження, збереження результатів та візуалізація статистики. Система використовує можливості PHP, JavaScript, AJAX, ORM, маршрутизації та MVC, демонструючи приклад ефективної клієнт-серверної взаємодії в реальному часі.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У процесі розробки вебдодатку для проведення опитувань і голосувань було здійснено комплексне тестування функціональності, юзабіліті та стійкості системи в умовах реального використання. Основними цілями тестування були:

перевірка коректної роботи логіки опитувань, обробки відповідей користувача, правильності збереження даних у базі, а також адаптивності інтерфейсу.

Для забезпечення якості ПЗ було застосовано відповідні методи тестування. Модульне тестування виконувалося для окремих компонентів – зокрема, методів контролерів (addTicketAction, registerAction, statAction). Тестування відбувалося з використанням спеціально сформованих вхідних даних, включаючи граничні значення, порожні поля та недопустимі символи. На етапі функціонального тестування перевірялося, як система обробляє послідовність дій користувача – створення опитування, проходження опитування, отримання результатів. Тестування охоплювало як інтерфейс адміністратора, так і публічну частину.

Під час тестування безпеки окремо перевірялась стійкість до SQL-ін'єкцій через форми, коректна робота з сесіями, обмеження прав доступу. Кросбраузерне тестування проводилося у середовищах Chrome, Firefox та Microsoft Edge. Здійснювалася перевірка відображення елементів інтерфейсу, коректної роботи форм і скриптів. В таблиці 3.1 описано основні системні вимоги для роботи додатку.

Таблиця 3.1 – Системні вимоги

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Операційна система	Windows 7/10/11, Linux	Windows 10/11, Ubuntu 22.04+
ОЗП (RAM)	2 ГБ	4 ГБ і більше
Процесор	Intel Core i3 або аналог	Intel Core i5/i7 або AMD Ryzen
Дисковий простір	100 МБ	500 МБ (для бази та логів)
Веб-браузер	Chrome, Firefox, Edge	Остання версія Chrome
Сервер	Apache з PHP 7.4+	Apache/Nginx з PHP 8.0+, MySQL 8
ПЗ для локального запуску	XAMPP 7.4+, або LAMP-стек	XAMPP 8.0+, або Docker

У ході тестування були виявлені наступні недоліки:

1) некоректна обробка порожніх форм під час реєстрації – усунуто шляхом додаткової перевірки required на рівні HTML та серверної валідації;

2) неможливість створення опитування без вибраних запитань – реалізовано перевірку довжини масиву \$_POST['arr'] перед записом;

3) неправильна обробка сесій при виході користувача – усунуто через явне знищення сесії у `WelcomeController::logoutAction()`;

4) помилки відображення діаграм у браузерях Safari – додано Fallback для JSON-даних, перевірка на порожні значення перед передачею у `anychart`.

Здійснене тестування підтвердило працездатність системи в рамках заявленого функціоналу. Своєчасне налагодження забезпечило стабільність вебдодатку та коректну роботу на різних пристроях. Програма готова до розгортання на продакшн-сервері з урахуванням зазначених системних вимог.

3.3 Розробка бази даних

Для зберігання даних у вебдодатку, розробленому з метою проведення опитувань і голосувань у режимі реального часу, було використано систему управління базами даних MySQL, яка є однією з найпопулярніших СУБД з відкритим кодом. Процес створення бази даних реалізовано із використанням інструменту `phpMyAdmin`, а також автоматично – через ORM-бібліотеку `RedBeanPHP`, що дозволяє динамічно створювати структуру таблиць у процесі взаємодії додатку з базою.

База даних має назву `poll`. Основні логічні сутності – це користувачі, опитування, запитання та результати. Кожна таблиця має унікальний ідентифікатор (`id`) та набір відповідних атрибутів. Нижче наведено короткий опис основних таблиць:

- 1) `users` – зберігає дані зареєстрованих користувачів;
- 2) `questions` – містить запитання та варіанти відповідей;
- 3) `tickets` – представляє опитування, які об'єднують кілька запитань;
- 4) `stats` – містить результати опитувань для кожного користувача.

Наведемо приклади SQL-запитів до основних таблиць. Запит на отримання списку всіх користувачів наведено в лістингу 3.9.

Лістинг 3.9 – Отримання списку користувачів

```
SELECT * FROM users;
```

кінець лістингу 3.9

Як отримати всі запитання, пов'язані з конкретним опитуванням наведено в лістингу 3.10.

Лістинг 3.10 – Запит на отримання запитань, пов'язаних з конкретним опитуванням

```
// PHP-код із використанням RedBeanPHP
$ticket = \R::findOne('tickets', 'id = ?', [ $id ]);
$arr = json_decode($ticket->ids);
$questions = \R::find('questions', 'id IN (' . \R::genSlots($arr) . ')', $arr);
```

кінець лістингу 3.10

Код збереження результатів опитування наведено в лістингу 3.11.

Лістинг 3.11 – Збереження результатів опитування

```
$stats = \R::dispense('stats');
$stats->json = $json;
$stats->title = $ticket->title;
$stats->date = date("Y-m-d");
$stats->user_id = (int)$_SESSION['id'];
\R::store($stats);
```

кінець лістингу 3.11

Запит на отримання всіх статистичних результатів конкретного користувача наведено в лістингу 3.12.

Лістинг 3.12 – Отримання всіх статистичних результатів конкретного користувача

```
SELECT * FROM stats
WHERE user_id = 5
ORDER BY date DESC;
```

кінець лістингу 3.12

ORM RedBeanPHP забезпечує автоматичне створення та оновлення таблиць на основі об'єктів, які використовуються в коді. Наприклад, при першому виклику (ліст. 3.13).

Лістинг 3.13 – Автоматичне створення запису в таблиці tickets за допомогою RedBeanPHP

```
$ticket = \R::dispense('tickets');
$ticket->title = "Новий тест";
\R::store($ticket);
```

кінець лістингу 3.13

Внаслідок виконання запиту буде автоматично створено таблицю tickets, якщо вона ще не існує, з колонкою title типу TEXT.

Основні переваги використаної схеми:

- 1) гнучкість – завдяки RedBeanPHP, не потрібно вручну створювати таблиці на початковому етапі.
- 2) швидка розробка – структура формується відповідно до логіки бізнес-процесів.
- 3) можливість міграції – можна легко експортувати структуру в SQL через phpMyAdmin.

У результаті розробки бази даних було реалізовано ефективну схему зберігання інформації про користувачів, структуру опитувань, запитання та відповіді, що дає змогу системі зберігати, обробляти та виводити дані у реальному часі. Комбінація MySQL та RedBeanPHP забезпечила високу швидкість розробки та масштабованість застосунку.

3.4 Захист інформаційно-комп'ютерної системи

Інформаційна безпека вебдодатку є важливим фактором його надійності та стійкості до зовнішніх загроз. У розробленому додатку реалізовано елементарні механізми захисту, зокрема автентифікацію адміністратора,

сесійну авторизацію, а також часткову інкапсуляцію доступу до бази даних через ORM RedBeanPHP. Система передбачає використання PHP-сесій для збереження типу користувача після входу (ліст. 3.14).

Лістинг 3.14 – Використання PHP-сесій для збереження типу користувача

```
if ($_SESSION['type'] == "A") {
    // Доступ для аналітика
}
```

кінець лістингу 3.14

Це дозволяє здійснювати розмежування доступу до окремих сторінок та функціоналу в інтерфейсі адміністратора. Хоча повноцінна реєстрація/вхід користувачів відсутні, система підтримує авторизацію адміністратора через механізм сесій, що зберігаються у змінній `$_SESSION`.

Завдяки використанню ORM-фреймворку RedBeanPHP, який підтримує автоматичне екранування даних, система мінімізує ризик SQL-ін'єкцій.

Наприклад, збереження нового опитування реалізовано наступним чином (ліст. 3.15).

Лістинг 3.15 – Збереження нового опитування

```
$ticket = \R::dispense('tickets');
$ticket->title = "Новий тест";
\R::store($ticket);
```

кінець лістингу 3.15

RedBeanPHP самостійно формує безпечні SQL-запити, зберігаючи дані в базі без необхідності прямого використання INSERT.

У формах завантаження зображень використано атрибут `enctype="multipart/form-data"` та обмеження на розмір.

Це забезпечує базовий контроль розміру файлів, однак відсутня валідація MIME-типу чи фільтрація потенційно шкідливих розширень (наприклад, `.php`).

Усі критично важливі дії обмежені через HTML-атрибут `required`, що забезпечує початкову перевірку полів на стороні клієнта. Проте ця перевірка не гарантує захисту без додаткової серверної валідації, яку необхідно впровадити.

Посилання на адміністративні дії (створення опитувань, видалення запитань тощо) винесені в окремі маршрути `/admin/*`, які захищено перевіркою сесії. Проте на рівні маршрутизатора або бекенду бракує централізованої перевірки прав доступу.

Загалом, для реального розгортання системи у відкритому середовищі необхідно додатково реалізувати повноцінну автентифікацію, фільтрацію даних, захист форм, а також впровадити серверні механізми перевірки прав доступу.

Висновки до розділу 3

У третьому розділі представлено практичну реалізацію вебдодатку для проведення опитувань і голосувань у режимі реального часу. Було створено функціональну систему з підтримкою створення опитувань, формування та редагування запитань, збереження відповідей користувачів, а також перегляду статистики у вигляді діаграм. Розробка велася з використанням PHP, JavaScript (зокрема бібліотеки jQuery) та бази даних MySQL, що дало змогу забезпечити інтерактивність інтерфейсу та ефективне збереження даних.

Для зручної роботи з базою даних впроваджено ORM-фреймворк RedBeanPHP, який дозволив автоматично створювати таблиці та суттєво зменшити обсяг SQL-коду. Реалізовано динамічну побудову інтерфейсу, використання AJAX-запитів і інтерактивну візуалізацію відповідей за допомогою бібліотеки AnyChart. Проведено тестування всіх основних функцій, виправлено виявлені помилки та забезпечено стабільну роботу в більшості сучасних браузерів.

Особливу увагу приділено базовим аспектам захисту: застосовано механізм авторизації на основі сесій, обмежено доступ до адміністративної

частини, використано ORM для запобігання SQL-ін'єкціям. Це дозволило підвищити рівень безпеки при обробці та зберіганні даних.

Загалом, вебдодаток відповідає поставленим функціональним вимогам, має гнучку архітектуру та може бути адаптований до різних сценаріїв використання. Його модульна структура забезпечує легке масштабування та доповнення новими можливостями в майбутньому.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Розробка вебдодатку для проведення опитувань і голосувань у режимі реального часу» було повністю реалізовано поставлене завдання, що охоплювало всі етапи проєктування: від аналізу предметної області до практичної реалізації системи з використанням сучасних інструментів та технологій.

Світові тенденції у сфері збору зворотного зв'язку свідчать про зростаючий попит на інструменти, які забезпечують реальну інтерактивність, персоналізацію та аналітику в реальному часі. Розроблений додаток відповідає цим вимогам та може бути адаптований до потреб онлайн-освіти, маркетингових досліджень, соціології, державного управління тощо.

Проведений аналіз сучасного стану проблеми показав, що на ринку вже існують рішення для створення онлайн-опитувань, однак вони мають ряд недоліків, таких як обмежена підтримка української мови, відсутність можливості адаптації логіки питань та обмежений доступ до розширеного функціоналу без підписки, що обґрунтовує доцільність створення власного вебдодатку з урахуванням локальних потреб.

У процесі аналізу та проєктування програмного забезпечення визначено вимоги до системи, обґрунтовано застосування моделі MVC, побудовано UML-діаграми (прецедентів, класів, послідовностей), а також здійснено формалізацію функціональних сценаріїв – що дозволило структурувати архітектуру додатку та підготувати основу для його реалізації.

Для вирішення поставлених завдань було обрано інструменти, що відповідають вимогам гнучкості та швидкої розробки: PHP у поєднанні з RedBeanPHP для роботи з БД, JavaScript та бібліотеку AnyChart для візуалізації, jQuery – для динаміки та AJAX, що забезпечило ефективну реалізацію ключових функцій системи.

У процесі реалізації системи створено повнофункціональний вебдодаток з підтримкою обліку користувачів, динамічного формування опитувань, вибору

запитань, збереження відповідей, перегляду історії та формування графічної статистики, що підтверджує виконання усіх вимог до проєкту.

В ході тестування було перевірено коректність роботи всіх основних сценаріїв використання системи, усунуто виявлені помилки, уточнено вимоги до середовища розгортання, що дозволило забезпечити стабільність роботи додатку та його сумісність із типовими серверними конфігураціями.

База даних реалізована за допомогою MySQL з використанням ORM RedBeanPHP, що дозволило створити гнучку структуру даних без ручного написання SQL-запитів; система підтримує автоматичне створення таблиць, зберігання відповідей користувачів, їх анкетних даних і статистики.

Щодо безпеки, у вебдодатку реалізовано сесійну авторизацію, обмеження доступу до адміністративного інтерфейсу та роботу з БД через ORM для мінімізації SQL-ін'єкцій, проте відсутні більш складні механізми – такі як шифрування, CSRF-захист і розмежування ролей, що визначає напрям подальшого вдосконалення системи.

Результати розробки можуть бути впроваджені в навчальний процес на кафедрі програмної інженерії для демонстрації основ MVC-архітектури, клієнт-серверної взаємодії та побудови UI/UX логіки. Вони також можуть бути використані для подальших досліджень у галузі адаптивних систем опитування, інтелектуального аналізу даних, машинного класифікування користувачів.

У процесі роботи отримано нові практичні результати, які засвідчують здатність інтегрувати опитування з базовими методами сегментації користувачів на основі їхніх анкетних даних. Це створює підґрунтя для побудови персоналізованих опитувальних систем у майбутньому.

Надалі роботу доцільно розвивати в напрямі:

- 1) удосконалення системи безпеки (впровадження реєстрації з підтвердженням, шифрування даних, CSRF/XSS-захисту);
- 2) створення повноцінної системи ролей і розмежування доступу;
- 3) розширення функціоналу за рахунок аналітичного модуля з машинним навчанням;

- 4) адаптації до мобільних платформ;
- 5) інтеграції з зовнішніми API та месенджерами.

Отже, кваліфікаційна робота має практичну цінність, демонструє відповідність сучасним технологічним тенденціям і може слугувати основою для подальших науково-дослідних розробок та прикладних рішень у сфері цифрової взаємодії з користувачами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Maaitah S., Qatawneh M., Quzmar A. E-voting system based on blockchain technology: A survey. In 2021 International Conference on Information Technology (ICIT). 2021. P. 200-205. URL: <https://doi.org/10.1109/ICIT52682.2021.9491734> (date of access: 07.03.2025).
2. Jafar U., Aziz M. J. A., Shukur Z. Blockchain for Electronic Voting System—Review and Open Research Challenges. *Sensors*. 2021. Vol. 21, no. 17. P. 5874. URL: <https://doi.org/10.3390/s21175874> (date of access: 07.03.2025).
3. E-voting Meets Blockchain: A Survey / M.-V. Vladucu et al. *IEEE Access*. 2023. P. 1. URL: <https://doi.org/10.1109/access.2023.3253682> (date of access: 10.03.2025).
4. Котобан А. А. Дистрибутивний клієнт-серверний застосунок для віддаленого проведення локальних голосувань та референдумів : Одеса, 2021. 48 с.
5. Кухаренко М. В. Вплив цифровізації на сучасні демократичні процеси та публічне адміністрування. Аналітичне-порівняльне правознавство/ редкол.: Ю. М. Бисага (голов. ред.), В. В. Заборовський, Д. М. Бєлов, С. Б. Булеца та ін.; ДВНЗ «УжНУ» Ужгород, 2024. №4. С. 390-394. Бібліогр.: с. 393-394 (8 назв). URL: <http://journal-app.uzhnu.edu.ua/article/view/311024/302324> (дата звернення: 11.03.2025).
6. Олешко І. Забезпечення анонімності користувачів при проведенні онлайн-опитувань. *Радіотехніка*. 2024. № 219. С. 28-35. URL: <https://doi.org/10.30837/rt.2024.4.219.03> (дата звернення: 13.03.2025).
7. Приймак М. А. Можливості цифровізації у посиленні участі громадськості в управлінні бюджетними ресурсами. Doctoral dissertation, 2025. С. 244-245.
8. Google Forms. Sign-in. URL: <https://docs.google.com/forms/u/0/> (date of access: 15.03.2025).

9. Home. Slido. Audience Interaction Made Easy.
URL: <https://www.slido.com/> (date of access: 21.03.2025).
10. Kahoot! Learning games. Make learning awesome!
URL: <https://kahoot.com/> (date of access: 21.03.2025).
11. Mentimeter. Interactive presentation software.
URL: <https://www.mentimeter.com/> (date of access: 23.03.2025).
12. PHP. Hypertext Preprocessor. URL: <https://www.php.net/> (date of access: 04.04.2025).
13. Tatroe K., MacIntyre P. Programming PHP. Creating dynamic web pages. " O'Reilly Media, Inc.", 2020. 544 p.
14. Grippa V. M., Kuzmichev S. Learning MySQL. " O'Reilly Media, Inc.", 2021. 632 p.
15. phpMyAdmin. URL: <https://www.phpmyadmin.net/> (date of access: 06.04.2025).
16. RedBeanPHP. Welcome. URL: <https://redbeanphp.com/index.php> (date of access: 10.04.2025).
17. Comparative Analysis of JavaScript Frameworks. Interantional journal of scientific research in engineering and management. 2024. Vol. 08, no. 05. P. 1-5.
URL: <https://doi.org/10.55041/ijsrem34297> (date of access: 16.04.2025).
18. AnyChart is a lightweight and robust JavaScript charting library. AnyChart.
URL: <https://www.anychart.com/> (date of access: 22.04.2025).