

**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 ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СПРОЩЕНОГО
УПРАВЛІННЯ ПРОЄКТАМИ НА ОСНОВІ PHP ТА MYSQL**

**SOFTWARE DEVELOPMENT FOR SIMPLIFIED PROJECT
MANAGEMENT BASED ON PHP AND MYSQL**

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

Виконав: здобувач вищої освіти
групи ІПЗ-41
Сятковський Владислав Олександрович

Керівник:
Суринович Олена Миколаївна

Кваліфікаційну роботу
допущено до захисту
«__» _____ 20__ р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

Луцьк – 2026 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__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Сятковському Владиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка програмного забезпечення для спрощеного управління проєктами на основі PHP та MySQL»

Керівник роботи: к.т.н., доц. Суринович О. М.

затверджені наказом закладу вищої освіти від «31» грудня 2025 р. № 541/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «29» травня 2026 р.

3. Вихідні дані до роботи: предметна область управління проєктами та завданнями, технології розробки веборієнтованого програмного забезпечення (PHP, JavaScript, HTML, CSS), засоби роботи з базами даних (MySQL, phpMyAdmin), локальне серверне середовище XAMPP, методи проєктування інформаційних систем (UML-діаграми, ER-моделювання), вимоги до веборієнтованих інформаційних систем та забезпечення інформаційної безпеки.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз сучасного стану систем управління проєктами та веборієнтованих інформаційних систем, постановка завдання на розробку програмного забезпечення для спрощеного управління проєктами, формування функціональних і нефункціональних вимог до системи, обґрунтування вибору технологій та засобів розробки, проєктування архітектури програмного забезпечення та структури бази даних, побудова UML- та ER-діаграм, практична реалізація веборієнтованої інформаційної системи, розробка структури бази даних та SQL-запитів, реалізація механізмів авторизації, автентифікації та захисту інформації, тестування та налагодження програмного забезпечення, розробка документації для програмного продукту.

5. Перелік графічного матеріалу: 17 рисунків, 3 лістинги.

6. Консультанти розділів роботи

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

7. Дата видачі завдання «3» лютого 2026 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 14.02.2026 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 03.03.2026 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 14.03.2026 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 14.04.2026 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 20.04.2026 р.	
6	Тестування та налагодження об'єкта проектування	до 04.05.2026 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 29.05.2026 р.	

Здобувач вищої освіти _____

Владислав СЯТКОВСЬКИЙ

Керівник кваліфікаційної роботи _____

Олена СУРИНОВИЧ

АНОТАЦІЯ

Сятковський В. О. Розробка програмного забезпечення для спрощеного управління проєктами на основі PHP та MySQL. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2026.

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

У першому розділі проведено дослідження предметної області, пов'язаної з управлінням проєктами у веборієнтованих інформаційних системах, проаналізовано сучасні програмні рішення для управління проєктами, зокрема Jira, Trello, Asana та Monday.com, визначено їхні переваги та недоліки, а також обґрунтовано необхідність створення спрощеної системи управління проєктами для невеликих команд і організацій. У другому розділі сформовано функціональні та нефункціональні вимоги до програмного забезпечення, виконано проєктування системи із використанням UML-діаграми класів та ER-діаграми бази даних, а також обґрунтовано вибір технологій, методів і засобів розробки, зокрема PHP, MySQL та XAMPP. У третьому розділі описано процес практичної реалізації веборієнтованої інформаційної системи, розроблено структуру бази даних, реалізовано механізми авторизації, автентифікації та захисту інформації, а також наведено результати тестування, налагодження та розробки документації для програмного продукту. У висновках підсумовано результати виконаної роботи, підтверджено досягнення поставленої мети та повне виконання визначених завдань кваліфікаційної роботи.

Ключові слова: управління проєктами, PHP, MySQL, XAMPP, вебсистема, база даних, UML-діаграма.

ABSTRACT

Siatkovskiy V. O. Software Development for Simplified Project Management Based on PHP and MySQL. Manuscript. Qualification work for bachelor's degree in Software Engineering, specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2026.

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

The first chapter presents a study of the subject area related to project management in web-oriented information systems. Modern project management software solutions, including Jira, Trello, Asana, and Monday.com, were analyzed, their advantages and disadvantages were identified, and the necessity of developing a simplified project management system for small teams and organizations was substantiated. The second chapter defines the functional and non-functional requirements for the software product, describes the system design using a UML class diagram and an ER database diagram, and substantiates the choice of technologies, methods, and development tools, including PHP, MySQL, and XAMPP. The third chapter describes the process of practical implementation of the web-oriented information system, the development of the database structure, the implementation of authorization, authentication, and information security mechanisms, as well as the results of testing, debugging, and documentation development for the software product. The conclusions summarize the results of the completed work, confirm the achievement of the stated goal, and verify the full completion of the tasks defined for the qualification thesis.

Keywords: Project Management, PHP, MySQL, XAMPP, Web-System, Database, UML Diagram.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СПРОЩЕНОГО УПРАВЛІННЯ ПРОЄКТАМИ І 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.....	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	19
Висновки до розділу 2.....	22
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СПРОЩЕНОГО УПРАВЛІННЯ ПРОЄКТАМИ НА ОСНОВІ PHP ТА MYSQL	24
3.1 Практична реалізація об'єкта проєктування	24
3.2 Розробка бази даних	28
3.3 Захист інформаційно-комп'ютерної системи	30
3.4 Тестування та налагодження інформаційно-комп'ютерної системи.....	32
3.5 Розробка документації для програмного продукту	34
Висновки до розділу 3.....	36
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій сприяє активному впровадженню програмних систем у діяльність підприємств, організацій та установ різного типу. Одним із найважливіших напрямів автоматизації є управління проєктами, яке забезпечує планування, координацію, контроль виконання завдань та взаємодію між учасниками команди. Використання спеціалізованого програмного забезпечення дозволяє значно підвищити ефективність управління ресурсами, зменшити кількість помилок та покращити процес моніторингу виконання робіт.

У сучасних умовах значна кількість компаній використовує веборієнтовані системи для організації роботи над проєктами. Такі системи забезпечують доступ до інформації через мережу Інтернет, дозволяють централізовано зберігати дані та спрощують процес комунікації між користувачами. Особливо актуальними є системи спрощеного управління проєктами, які мають мінімалістичний інтерфейс, швидке налаштування та не потребують складної інфраструктури.

Мова програмування PHP та система керування базами даних MySQL є одними з найпоширеніших технологій для створення вебзастосунків. PHP забезпечує реалізацію серверної логіки, обробку запитів користувачів та взаємодію з базою даних, а MySQL відповідає за зберігання та обробку інформації. Використання цих технологій дозволяє створити гнучке, масштабоване та ефективне програмне забезпечення.

Актуальність теми кваліфікаційної роботи бакалавра полягає у необхідності розробки доступної та функціональної системи управління проєктами, яка забезпечить ефективну організацію роботи користувачів, контроль виконання завдань та збереження інформації у централізованій базі даних.

Метою кваліфікаційної роботи бакалавра є розробка програмного забезпечення для спрощеного управління проєктами із використанням технологій PHP та MySQL.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні системи управління проєктами;
- визначити функціональні та нефункціональні вимоги до програмного забезпечення;
- обрати технології та інструменти розробки;
- спроектувати архітектуру системи та структуру бази даних;
- реалізувати програмне забезпечення;
- провести тестування та налагодження системи;
- реалізувати механізми захисту інформації;
- розробити документацію користувача.

Об'єктом дослідження є процес управління проєктами у веборієнтованих інформаційних системах.

Предметом дослідження є методи та засоби розробки програмного забезпечення для спрощеного управління проєктами на основі PHP та MySQL.

Практичне значення отриманих результатів полягає у створенні веборієнтованої системи, яка може використовуватись невеликими компаніями, стартапами та командами розробників для організації роботи над проєктами.

РОЗДІЛ 1

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СПРОЩЕНОГО УПРАВЛІННЯ ПРОЄКТАМИ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасному світі управління проєктами є невід’ємною частиною діяльності більшості організацій. Незалежно від галузі, компанії потребують ефективних інструментів для планування, контролю виконання завдань, моніторингу ресурсів та взаємодії між учасниками команди.

Існує велика кількість програмних продуктів для управління проєктами, серед яких найбільш відомими є Jira, Trello, Asana, Monday.com, ClickUp та Basecamp [1]. Кожна із зазначених систем має власні особливості, функціональні можливості та підходи до організації роботи [2].

Система Jira широко використовується у сфері розробки програмного забезпечення та підтримує Agile-методології, Scrum та Kanban [3]. Водночас дана система є досить складною для невеликих команд та потребує значних ресурсів для адміністрування (рис. 1.1).



Рисунок 1.1 – Інтерфейс системи тренінгу Jira для бізнесу [4]

Trello орієнтована на простоту використання та базується на концепції Канбан-дошок [5]. Система дозволяє організовувати завдання у вигляді карток та списків. Однак функціональність Trello є обмеженою для складних проєктів (рис. 1.2).

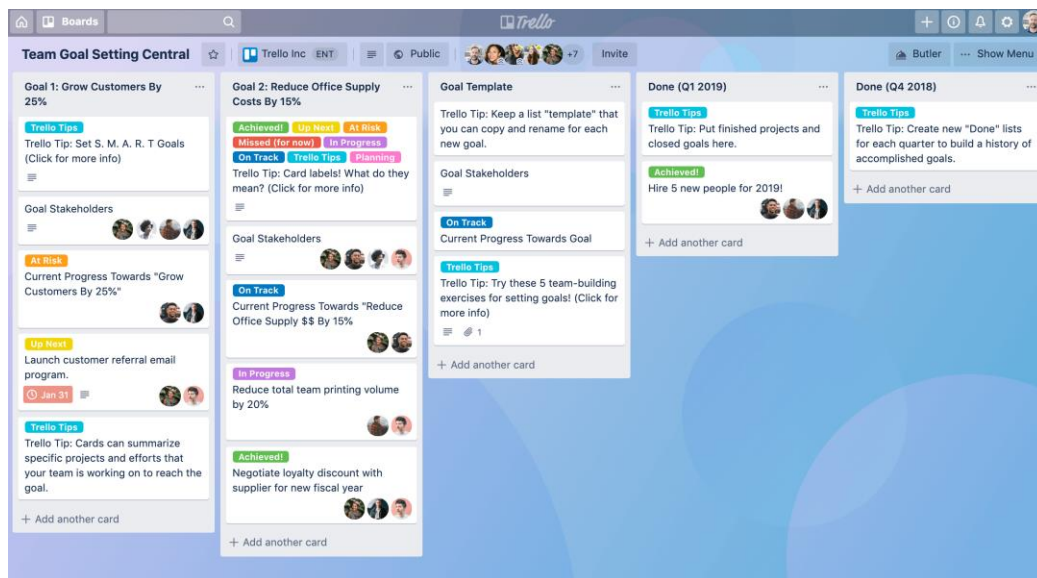


Рисунок 1.2 – Інтерфейс системи Trello [6]

Asana [7] та Monday.com [8] забезпечують розширені можливості управління проєктами, включаючи аналітику, інтеграції та автоматизацію процесів. Разом із тим використання таких систем часто потребує платної підписки (рис. 1.3-1.4).

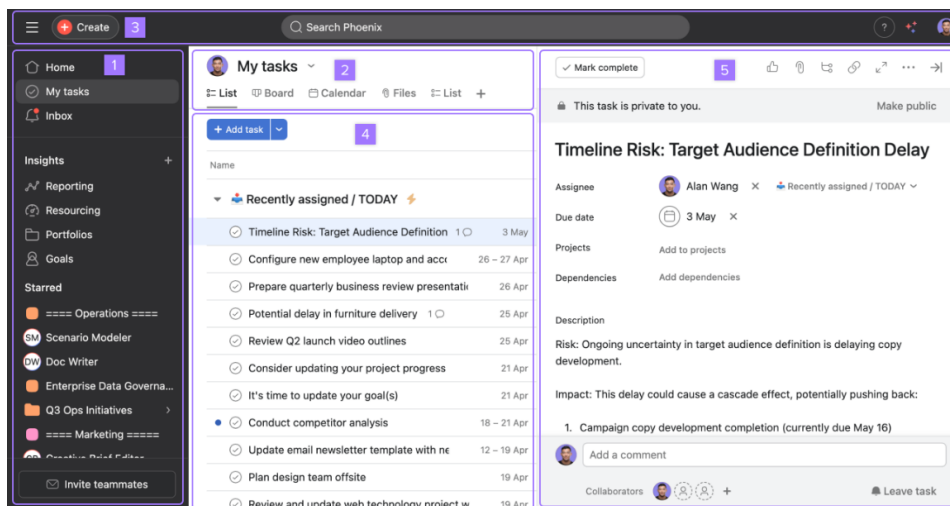


Рисунок 1.3 – Інтерфейс системи Asana [9]

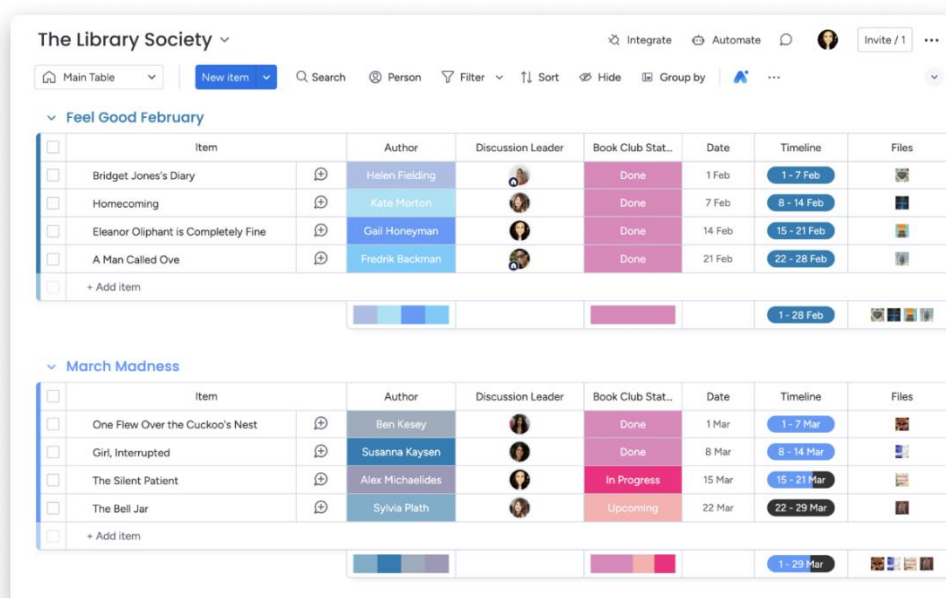


Рисунок 1.4 – Інтерфейс системи Monday.com [10]

Аналіз сучасних систем показав, що значна частина програмного забезпечення для управління проєктами має надлишковий функціонал для невеликих організацій. Це створює потребу у розробці спрощених інформаційних систем, які забезпечують базові функції управління проєктами та не потребують значних витрат.

Основними функціями систем управління проєктами є організація роботи користувачів, створення та редагування проєктів, управління завданнями, контроль виконання робіт, ведення клієнтської бази та забезпечення зручної взаємодії між учасниками команди. Важливим аспектом є також забезпечення захисту інформації та централізованого зберігання даних.

При розробці веборієнтованих інформаційних систем важливе значення мають технології реалізації. Однією з найпоширеніших мов програмування для серверної розробки є PHP [11]. Дана мова підтримується більшістю хостингів, має велику кількість бібліотек та характеризується простотою інтеграції із базами даних.

Система керування базами даних MySQL використовується для зберігання структурованої інформації [12]. Вона забезпечує швидке виконання SQL-запитів, підтримує реляційну модель даних та має високий рівень продуктивності.

Після аналізу існуючих рішень можна зробити висновок, що спрощена система управління проєктами повинна мати:

- зрозумілий інтерфейс;
- веборієнтовану архітектуру;
- систему авторизації;
- централізоване зберігання даних;
- можливість роботи із клієнтами та проєктами;
- механізми захисту інформації.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою кваліфікаційної роботи є розробка програмного забезпечення для спрощеного управління проєктами із використанням технологій PHP та MySQL.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні системи управління проєктами;
- визначити функціональні та нефункціональні вимоги до програмного забезпечення;
- обрати технології та інструменти розробки;
- спроектувати архітектуру системи та структуру бази даних;
- реалізувати програмне забезпечення;
- провести тестування та налагодження системи;
- реалізувати механізми захисту інформації;
- розробити документацію користувача.

Система повинна забезпечувати:

- реєстрацію та авторизацію користувачів;
- створення, редагування та видалення проєктів;
- управління списком клієнтів;
- створення та редагування завдань;
- облік часу виконання робіт;
- захист даних користувачів;

- взаємодію із базою даних MySQL.

Основними вимогами до програмного забезпечення є:

- простота використання;
- веборієнтована архітектура;
- надійність зберігання інформації;
- захист від несанкціонованого доступу;
- масштабованість;
- кросбраузерність.

Висновки до розділу 1

У першому розділі проведено аналіз сучасних систем управління проєктами та визначено основні проблеми існуючих програмних рішень. Встановлено, що багато систем мають надлишковий функціонал та складний інтерфейс, що ускладнює їх використання невеликими командами.

Сформульовано мету та основні завдання кваліфікаційної роботи. Визначено основні функціональні можливості майбутньої системи, а також обґрунтовано доцільність використання PHP та MySQL для реалізації веборієнтованої інформаційної системи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення

На етапі проєктування інформаційної системи необхідно визначити функціональні та нефункціональні вимоги до програмного забезпечення.

До функціональних вимог системи належить:

- забезпечення авторизації користувачів;
- створення та редагування облікових записів;
- управління проєктами, завданнями та клієнтами;
- збереження інформації у базі даних.

Система повинна забезпечувати можливість редагування та видалення записів, а також швидку взаємодію користувача із серверною частиною вебзастосунку.

До нефункціональних вимог належать:

- безпечність;
- продуктивність;
- надійність;
- масштабованість;
- зручність використання;
- адаптивність інтерфейсу.

Архітектура системи реалізована за клієнт-серверним принципом [13]. Користувач взаємодіє із системою через веббраузер, а сервер обробляє запити та взаємодіє із базою даних.

Для реалізації системи було використано:

- PHP;
- MySQL;
- HTML;
- CSS;

- JavaScript;
- Apache Server.

Для кращого розуміння структури та принципів функціонування розроблюваної інформаційної системи доцільно використати UML-діаграму класів [14]. Дана діаграма дозволяє наочно відобразити основні компоненти програмного забезпечення, їх атрибути, методи та взаємозв'язки між окремими модулями системи. Використання UML-моделювання значно спрощує процес проєктування програмного забезпечення, допомагає визначити архітектуру системи та забезпечує логічну організацію взаємодії між її елементами. На рисунку 2.1 наведено діаграму класів програмного забезпечення для спрощеного управління проєктами.

Проведемо опис діаграми. Клас «User» описує користувача системи. До основних атрибутів належать ідентифікатор користувача, ім'я користувача, електронна пошта, пароль та роль у системі. Методи класу забезпечують авторизацію користувача, вихід із системи та оновлення особистих даних. Даний клас є одним із головних елементів системи, оскільки через нього реалізовується доступ до функціональних можливостей вебзастосунку.

Клас «Customer» призначений для зберігання інформації про клієнтів. Він містить атрибути, які описують ім'я клієнта, назву компанії, електронну пошту та номер телефону. Методи класу дозволяють створювати, редагувати та видаляти інформацію про клієнтів. Наявність окремого класу «Customer» дає можливість систематизувати інформацію про замовників та забезпечити зв'язок між клієнтами і проєктами.

Клас «Project» використовується для управління проєктами. Він містить основні характеристики проєкту, зокрема назву, опис, дату початку, дату завершення та статус виконання. Методи класу забезпечують створення, редагування та видалення проєктів. Один клієнт може мати декілька проєктів, тому між класами «Customer» та «Project» реалізовано зв'язок типу «один до багатьох».

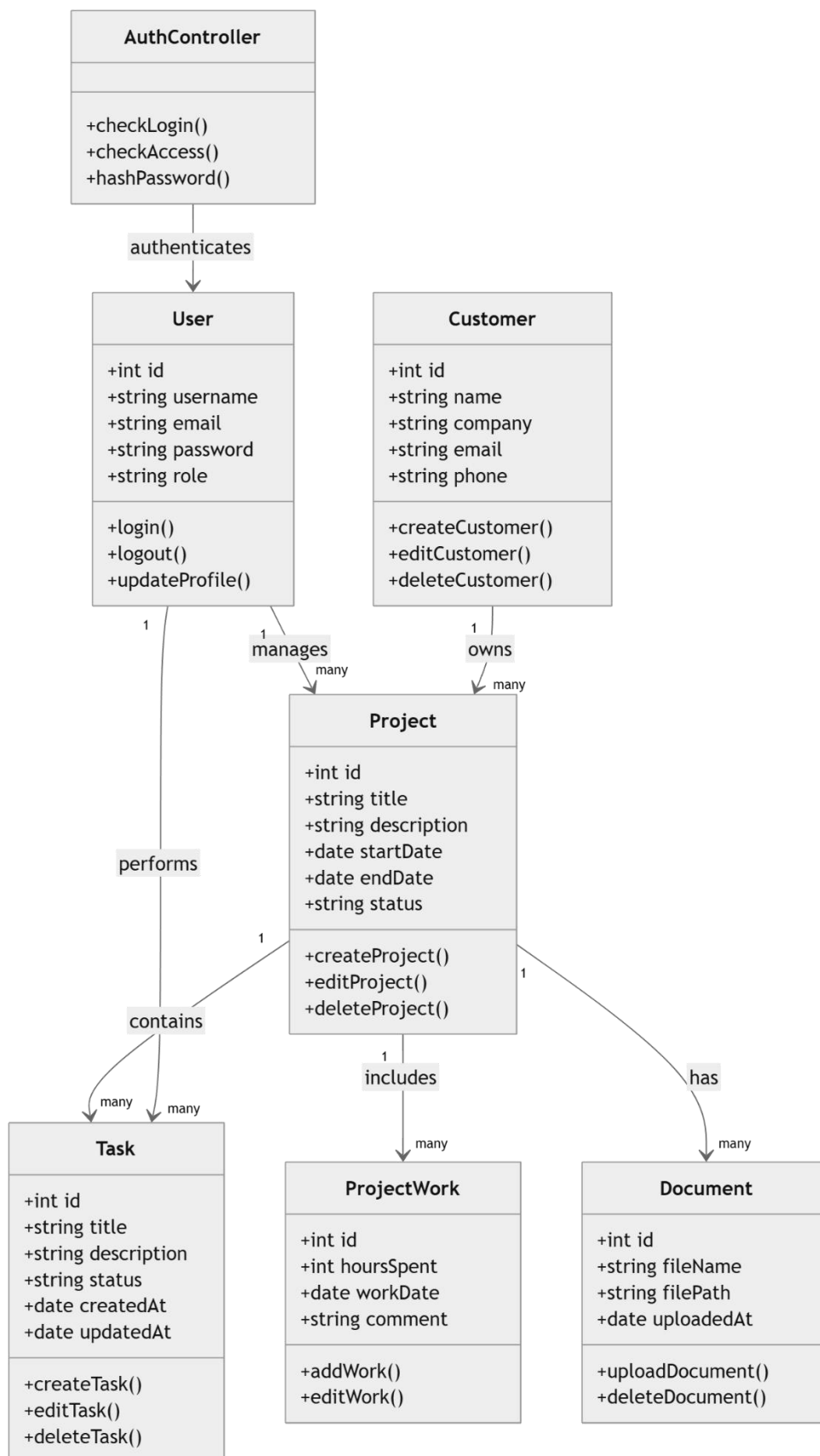


Рисунок 2.1 – Діаграма класів програмного забезпечення для спрощеного управління проєктами

Клас «Task» відповідає за управління завданнями в межах проєкту. До атрибутів класу належать назва завдання, опис, статус виконання, дата створення та дата оновлення. Методи класу забезпечують створення, редагування та видалення завдань. Один проєкт може містити багато завдань, тому між класами «Project» і «Task» також реалізовано зв'язок «один до багатьох».

Клас «ProjectWork» використовується для обліку виконаної роботи над проєктом. У класі зберігається інформація про кількість витрачених годин, дату виконання роботи та коментарі. Методи класу дозволяють додавати нові записи про виконану роботу та редагувати їх. Даний клас забезпечує можливість контролю часу, витраченого на виконання проєктів.

Клас «Document» призначений для роботи із файлами та документами, які належать до певного проєкту. Атрибути класу містять назву файлу, шлях до нього та дату завантаження. Методи класу забезпечують завантаження та видалення документів. Один проєкт може містити декілька документів, тому між класами «Project» та «Document» існує зв'язок «один до багатьох».

Клас «AuthController» реалізує механізми авторизації та автентифікації користувачів. Основними методами класу є перевірка логіна та пароля, контроль доступу користувачів до функціоналу системи та хешування паролів. Даний клас взаємодіє із класом «User» і забезпечує захист інформаційної системи від несанкціонованого доступу.

Між класами реалізовані логічні взаємозв'язки. Один користувач може керувати декількома проєктами, один клієнт може мати декілька проєктів, а кожен проєкт може містити багато завдань, документів та записів про виконану роботу. Така структура забезпечує гнучкість системи, спрощує масштабування програмного забезпечення та дозволяє ефективно організувати процес управління проєктами.

Після побудови діаграми класів доцільно перейти до проєктування структури бази даних, оскільки саме вона забезпечує збереження, обробку та взаємозв'язок інформації у системі. Діаграма класів дозволяє визначити основні об'єкти програмного забезпечення та взаємодію між ними, тоді як ER-діаграма

деталізує структуру таблиць бази даних, їх атрибути, ключі та типи зв'язків між сутностями [15]. Використання ER-моделювання дає можливість забезпечити цілісність даних, уникнути дублювання інформації та оптимізувати процес виконання SQL-запитів у системі управління проєктами. На рисунку 2.2 наведено ER-діаграму бази даних системи управління проєктами.

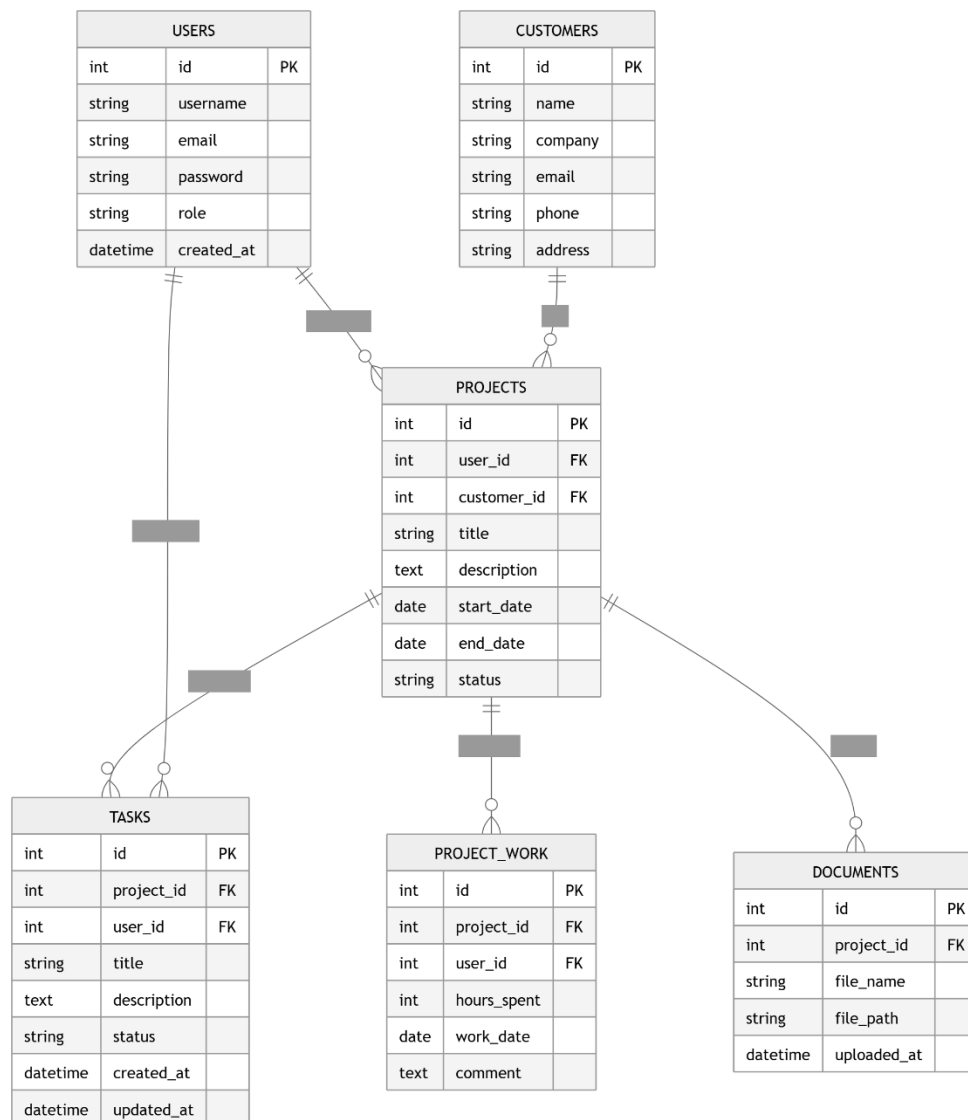


Рисунок 2.2 – ER-діаграма бази даних системи управління проєктами

Отже, було проведено аналіз предметної області та визначено основні функціональні й нефункціональні вимоги до розроблюваної системи управління проєктами. Визначено основні модулі програмного забезпечення, структуру взаємодії користувачів із системою та особливості організації даних у базі даних.

Також було виконано проєктування архітектури програмного забезпечення та побудовано UML- і ER-діаграми, які дозволили наочно відобразити структуру системи та взаємозв'язки між її компонентами. Отримані результати створюють основу для подальшої практичної реалізації програмного забезпечення та вибору ефективних методів і засобів розробки системи.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для реалізації серверної частини системи було обрано мову програмування PHP, оскільки вона характеризується простотою використання, широкими можливостями інтеграції із базами даних та підтримкою більшості сучасних вебсерверів. Крім того, PHP має велику кількість готових бібліотек і фреймворків, що значно спрощує процес створення веборієнтованих інформаційних систем.

Для зберігання інформації використовувалася система керування базами даних MySQL, яка забезпечує ефективне виконання SQL-запитів, підтримує реляційну модель даних та дозволяє організувати надійне централізоване зберігання інформації.

Для створення користувацького інтерфейсу використовувались HTML, CSS та JavaScript. Дані технології забезпечують формування адаптивного та зручного вебінтерфейсу для роботи користувачів із системою управління проєктами.

Для локального розгортання та тестування програмного забезпечення використовувалось середовище XAMPP, яке забезпечує підтримку Apache, PHP та MySQL [16]. Використання XAMPP дозволяє швидко налаштувати локальний сервер для розробки та тестування вебсистеми без необхідності використання окремого серверного обладнання. Середовище включає Apache як вебсервер, PHP для обробки серверної логіки та MySQL для роботи із базою даних (рис. 2.3).

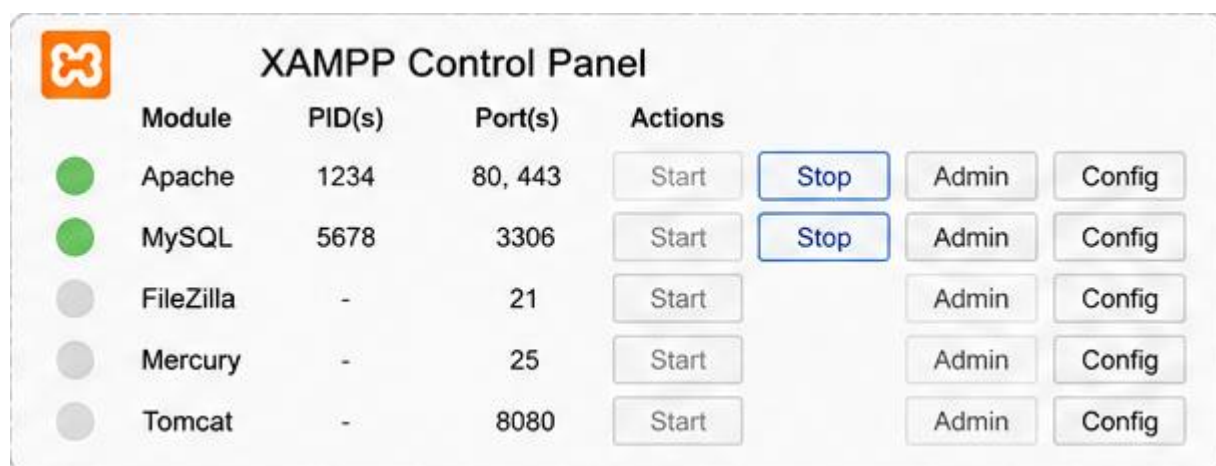


Рисунок 2.3 – Використання XAMPP для локального розгортання системи управління проєктами

Для взаємодії із базою даних у системі використовуються SQL-запити, які забезпечують отримання, додавання, редагування та видалення інформації. Приклад SQL-запитів наведено у лістингу 2.1.

Лістинг 2.1 – SQL-запит взаємодії з базою даних

```
SELECT * FROM users;
```

```
INSERT INTO projects(name, description)
VALUES('CRM', 'Project CRM');
```

```
UPDATE todos
SET status='done'
WHERE id=1;
```

```
DELETE FROM customers
WHERE id=2;
```

кінець лістингу 2.1

Для захисту паролів використовувався сучасний алгоритм bcrypt, який реалізується за допомогою функції password_hash() [17]. Такий підхід забезпечує високий рівень захисту облікових даних користувачів та значно перевищує рівень безпеки застарілого алгоритму MD5, що використовувався у ранніх версіях подібних систем.

У процесі розробки програмного забезпечення для спрощеного управління проєктами використовувалися різні алгоритми, які забезпечують коректну роботу системи, безпечне зберігання даних та взаємодію користувачів із вебсистемою. Використання таких алгоритмів дозволяє реалізувати основні функціональні можливості системи та забезпечити стабільну роботу програмного забезпечення.

Алгоритм авторизації використовувався для перевірки прав доступу користувача після успішного входу в систему. Після введення логіна та пароля система визначає роль користувача та надає доступ лише до тих функцій, які дозволені відповідним рівнем доступу. Наприклад, адміністратор має можливість керувати користувачами системи, тоді як звичайний користувач може працювати лише із проєктами та завданнями. Реалізація авторизації дозволяє обмежити доступ до окремих модулів системи та запобігти несанкціонованому використанню функціоналу.

Алгоритм автентифікації використовувався для підтвердження особи користувача під час входу в систему. Після введення логіна та пароля введені дані порівнюються із записами у базі даних. У випадку успішного співпадіння система створює сесію користувача та надає доступ до системи. Якщо введені дані є неправильними, система виводить повідомлення про помилку авторизації. Використання механізму автентифікації забезпечує контроль доступу до інформаційної системи.

Алгоритм хешування паролів використовувався для безпечного зберігання облікових даних користувачів у базі даних. У системі використовується алгоритм `bcrypt`, реалізований за допомогою функції `password_hash()`. Під час реєстрації або зміни пароля введений пароль не зберігається у відкритому вигляді, а перетворюється у спеціальний хеш-код. Під час входу в систему введений пароль перевіряється за допомогою функції `password_verify()`. Такий підхід значно підвищує рівень безпеки системи та ускладнює можливість отримання паролів сторонніми особами.

Алгоритм валідації форм використовувався для перевірки правильності введених користувачем даних перед їх обробкою сервером. У процесі валідації перевіряється коректність заповнення полів, формат електронної пошти, довжина пароля, наявність обов'язкових даних та допустимість введених символів. Валідація може виконуватись як на стороні клієнта за допомогою JavaScript, так і на стороні сервера засобами PHP. Використання алгоритмів валідації дозволяє уникнути помилок введення та підвищити надійність роботи системи.

Алгоритми SQL-запитів забезпечують взаємодію системи із базою даних MySQL. За допомогою SQL-запитів виконується отримання, додавання, редагування та видалення інформації. Для підвищення рівня безпеки у системі використовувався prepared statements, які дозволяють захистити програмне забезпечення від SQL-ін'єкцій. SQL-запити використовуються під час роботи із таблицями користувачів, проєктів, клієнтів та завдань, забезпечуючи централізоване зберігання та обробку інформації у системі управління проєктами [18].

Висновки до розділу 2

У другому розділі було проведено аналіз та визначення основних вимог до розроблюваної системи управління проєктами. Встановлено функціональні та нефункціональні вимоги до програмного забезпечення, які забезпечують стабільну, безпечну та ефективну роботу вебсистеми. Визначено основні модулі системи, зокрема модулі авторизації, управління користувачами, проєктами, клієнтами та завданнями.

У процесі проєктування програмного забезпечення було побудовано UML-діаграму класів та ER-діаграму бази даних, які дозволили відобразити структуру системи, взаємозв'язки між компонентами та особливості організації даних. Використання UML- та ER-моделювання спростило процес проєктування архітектури програмного забезпечення та структури бази даних.

Також було обґрунтовано вибір засобів та технологій розробки програмного забезпечення. Для реалізації серверної частини системи обрано PHP, для зберігання інформації – MySQL, а для створення користувацького інтерфейсу – HTML, CSS та JavaScript. Для локального розгортання та тестування системи використовувалось середовище XAMPP, яке забезпечує підтримку Apache, PHP та MySQL.

У розділі також розглянуто основні алгоритми, що використовуються у системі, зокрема алгоритми авторизації, автентифікації, хешування паролів, валідації форм та виконання SQL-запитів. Використання prepared statements та алгоритму bcrypt дозволило підвищити рівень безпеки системи та забезпечити захист даних користувачів.

Отримані результати створюють основу для подальшої практичної реалізації програмного забезпечення, тестування системи та впровадження механізмів захисту інформації.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СПРОЩЕНОГО УПРАВЛІННЯ ПРОЄКТАМИ НА ОСНОВІ PHP ТА MYSQL

3.1 Практична реалізація об'єкта проєктування

Практична реалізація програмного забезпечення для спрощеного управління проєктами виконувалась із використанням мови програмування PHP, системи керування базами даних MySQL та вебтехнологій HTML, CSS і JavaScript. Основною метою реалізації було створення веборієнтованої інформаційної системи, яка забезпечує управління проєктами, завданнями, клієнтами та користувачами через вебінтерфейс.

Архітектура програмного забезпечення реалізована за клієнт-серверним принципом. Користувач взаємодіє із системою через веббраузер, надсилаючи HTTP-запити до вебсервера Apache. Серверна частина системи, реалізована на PHP, обробляє запити користувачів, виконує взаємодію із базою даних MySQL та повертає сформовані HTML-сторінки.

Для локального розгортання та тестування системи використовувалось середовище XAMPP, яке забезпечує підтримку Apache, PHP та MySQL. Усі файли програмного забезпечення розміщувались у директорії htdocs локального сервера (рис. 3.1).

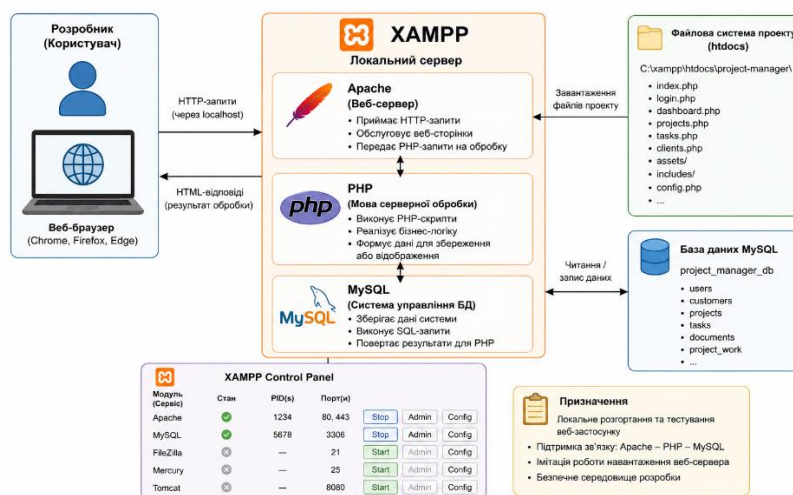


Рисунок 3.1 – Структура локального середовища розробки XAMPP

У процесі реалізації системи було створено окремі модулі для роботи із користувачами, проектами, клієнтами та завданнями. Головна сторінка системи забезпечує доступ до навігаційного меню, списку проектів та основних функцій керування (рис. 3.2).

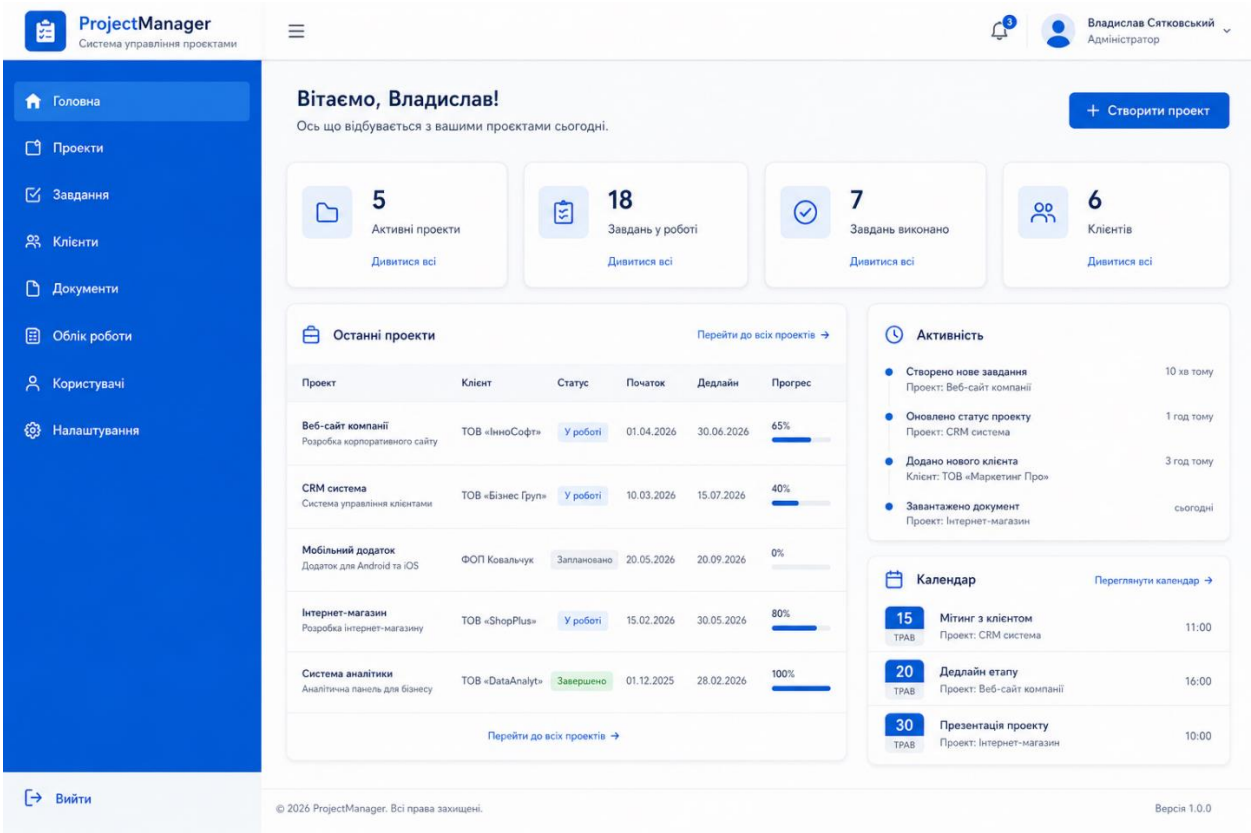


Рисунок 3.2 – Головна сторінка системи управління проектами

Модуль авторизації забезпечує вхід користувача до системи. Після введення логіна та пароля система виконує перевірку введених даних у базі даних та створює сесію користувача.

Для реалізації механізму авторизації використовувався PHP-код із використанням функцій password_verify() та session_start(). Приклад реалізації авторизації наведено у лістингу 3.1.

Лістинг 3.1 – Реалізація механізму авторизації користувача

```
<?php
session_start();
require 'config.php';
```

```

$email = $_POST['email'];
$password = $_POST['password'];

$query = $pdo->prepare("SELECT * FROM users WHERE email = ?");
$query->execute([$email]);

$user = $query->fetch();

if($user && password_verify($password, $user['password'])) {
    $_SESSION['user_id'] = $user['id'];
    header("Location: dashboard.php");
} else {
    echo "Неправильний логін або пароль";
}
?>

```

кінець лістингу 3.1

Для управління проектами реалізовано функції створення, редагування та видалення проектів. Користувач може створювати нові проекти, задавати назву, опис, дату початку та статус виконання (рис. 3.3).

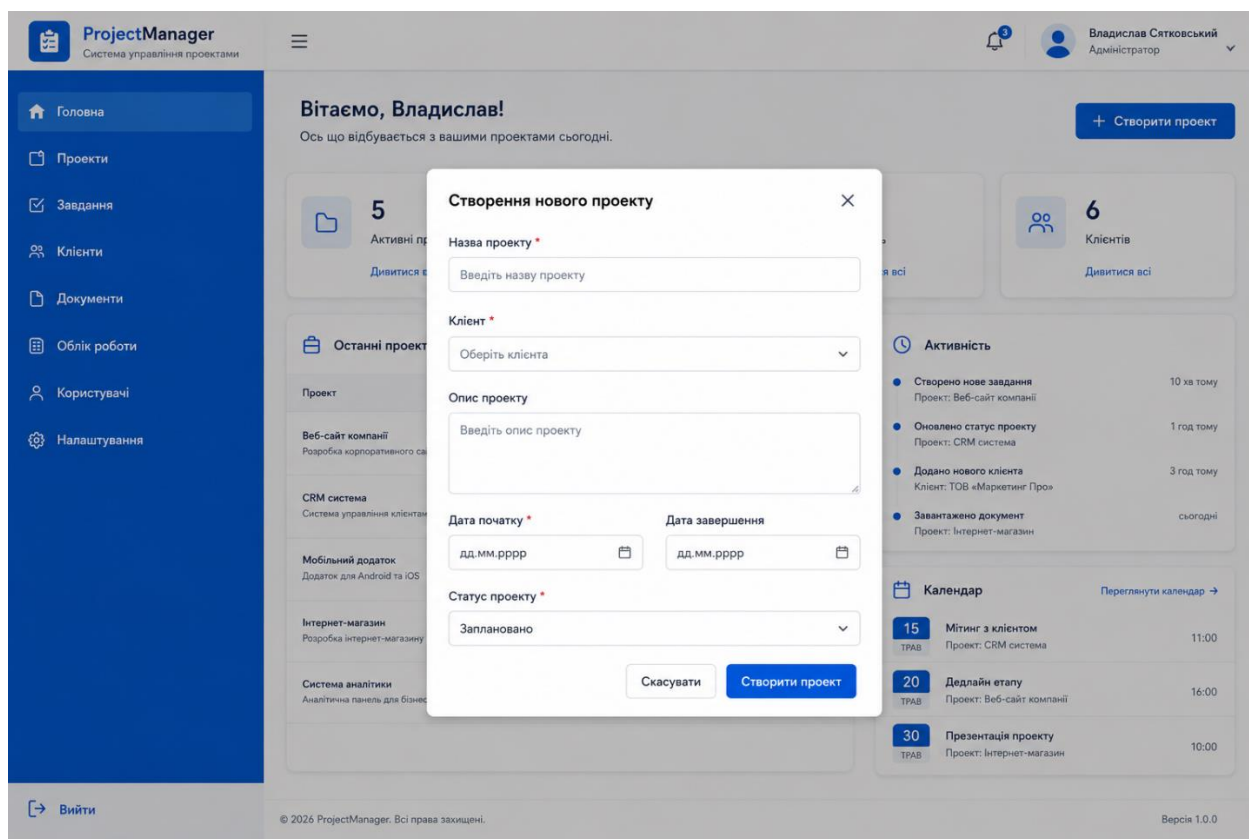


Рисунок 3.3 – Форма створення нового проекту

Після створення проєкту інформація автоматично зберігається у базі даних MySQL. Для взаємодії із базою даних використовувались prepared statements, що дозволило підвищити рівень захисту від SQL-ін'єкцій.

У системі також реалізовано модуль управління завданнями. Кожен проєкт може містити необмежену кількість завдань, які мають власний статус виконання, опис та дату створення (рис. 3.4).

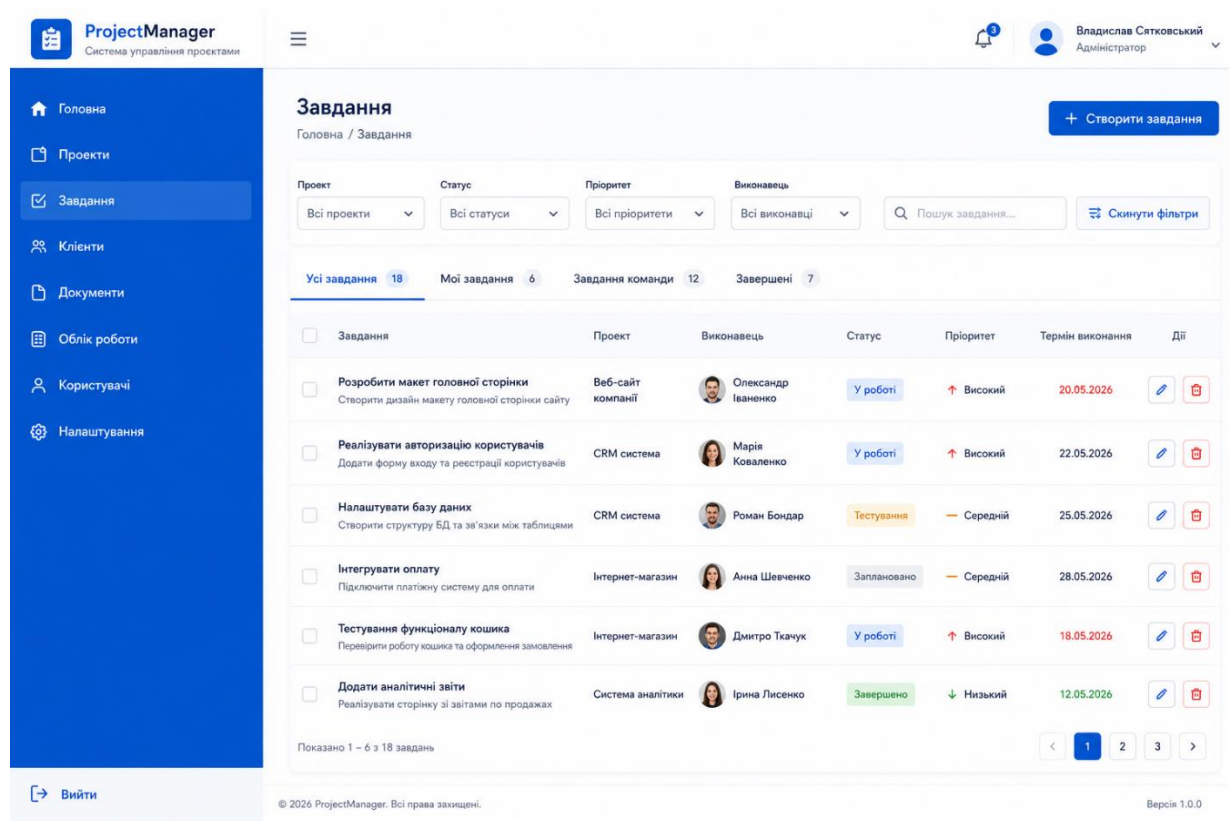


Рисунок 3.4 – Інтерфейс управління завданнями

Для покращення зручності використання системи використовувались технології HTML та CSS. Адаптивність вебінтерфейсу забезпечується використанням Flexbox та Bootstrap-компонентів. Це дозволяє коректно відображати систему на різних пристроях та екранах.

Для динамічного оновлення окремих елементів сторінки використовувався JavaScript. Зокрема, JavaScript застосовувався для перевірки правильності введення даних у формах та реалізації модальних вікон (рис. 3.5).

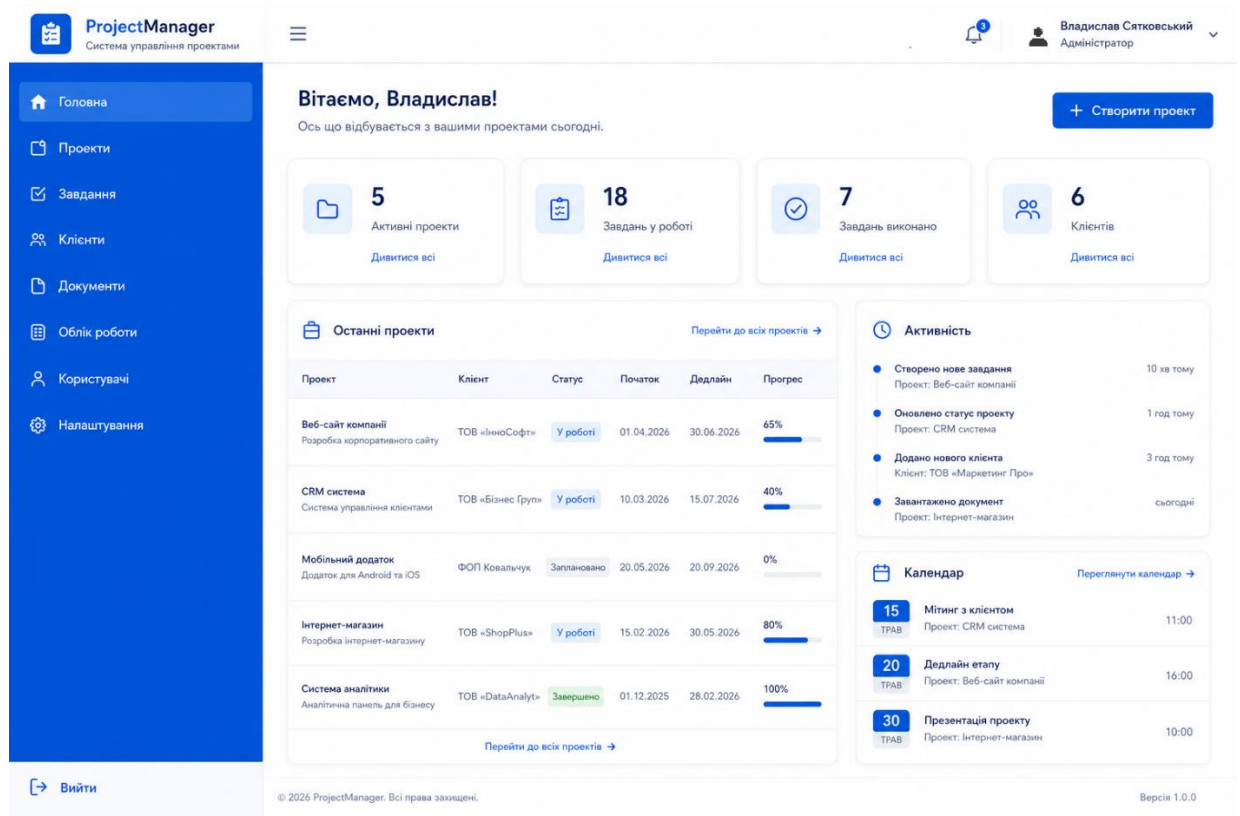


Рисунок 3.5 – Інтерфейс системи після авторизації користувача

У результаті практичної реалізації було створено веборієнтовану систему управління проектами, яка забезпечує централізоване зберігання інформації, взаємодію користувачів із проектами та ефективне управління завданнями.

3.2 Розробка бази даних

База даних інформаційної системи була реалізована із використанням системи керування базами даних MySQL. Основною метою проєктування бази даних було забезпечення централізованого зберігання інформації про користувачів, клієнтів, проекти, завдання та документи.

Структура бази даних розроблялась на основі ER-діаграми, наведеної у другому розділі роботи. Для створення таблиць використовувався інструмент phpMyAdmin, який входить до складу XAMPP (рис. 3.6).

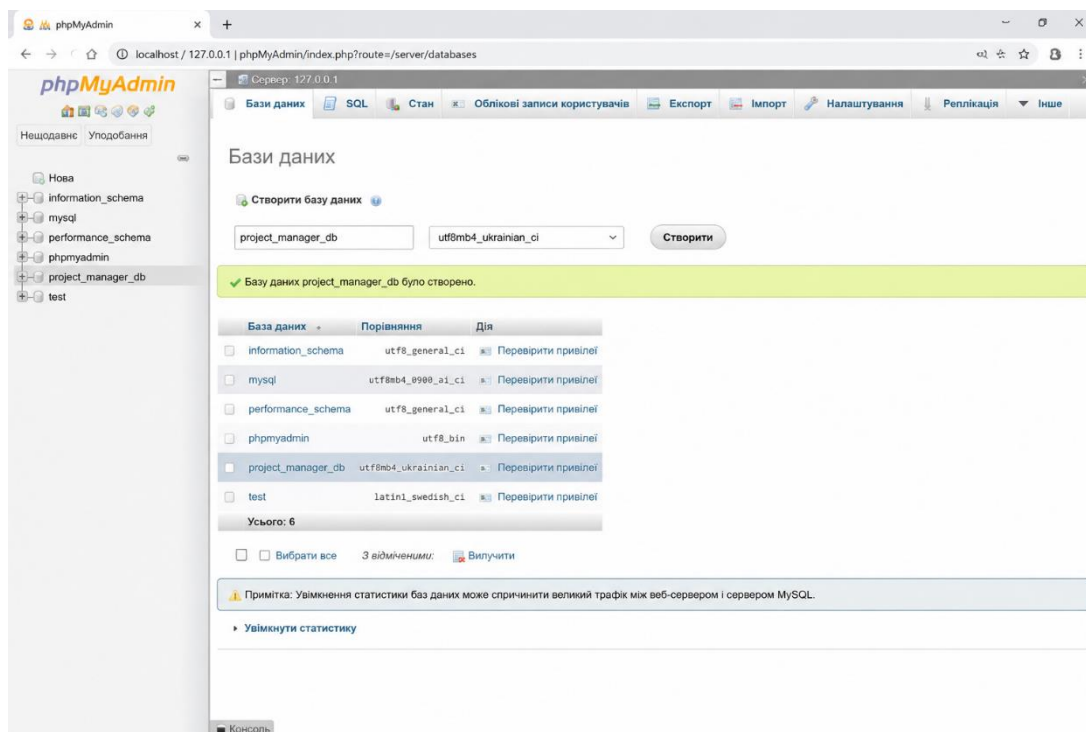


Рисунок 3.6 – Створення бази даних у phpMyAdmin

У системі було створено таблиці users, customers, projects, tasks, documents та project_work. Таблиця users містить інформацію про користувачів системи, їх логіни, електронні адреси, паролі та ролі доступу.

Таблиця projects використовується для зберігання інформації про проекти. Вона містить зовнішні ключі user_id та customer_id, що забезпечують зв'язок із таблицями users та customers.

Для створення таблиці users використовувався SQL-запит (ліст. 3.2).

Лістинг 3.2 – SQL-запит створення таблиці users

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100),
  email VARCHAR(150),
  password VARCHAR(255),
  role VARCHAR(50),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

кінець лістингу 3.2

Для забезпечення цілісності даних використовувались первинні та зовнішні ключі. Це дозволяє уникнути дублювання інформації та забезпечує правильність взаємозв'язків між таблицями.

Під час роботи системи активно використовуються SQL-запити SELECT, INSERT, UPDATE та DELETE. SQL-запити забезпечують отримання інформації із бази даних, додавання нових записів та оновлення існуючих даних (рис. 3.7).

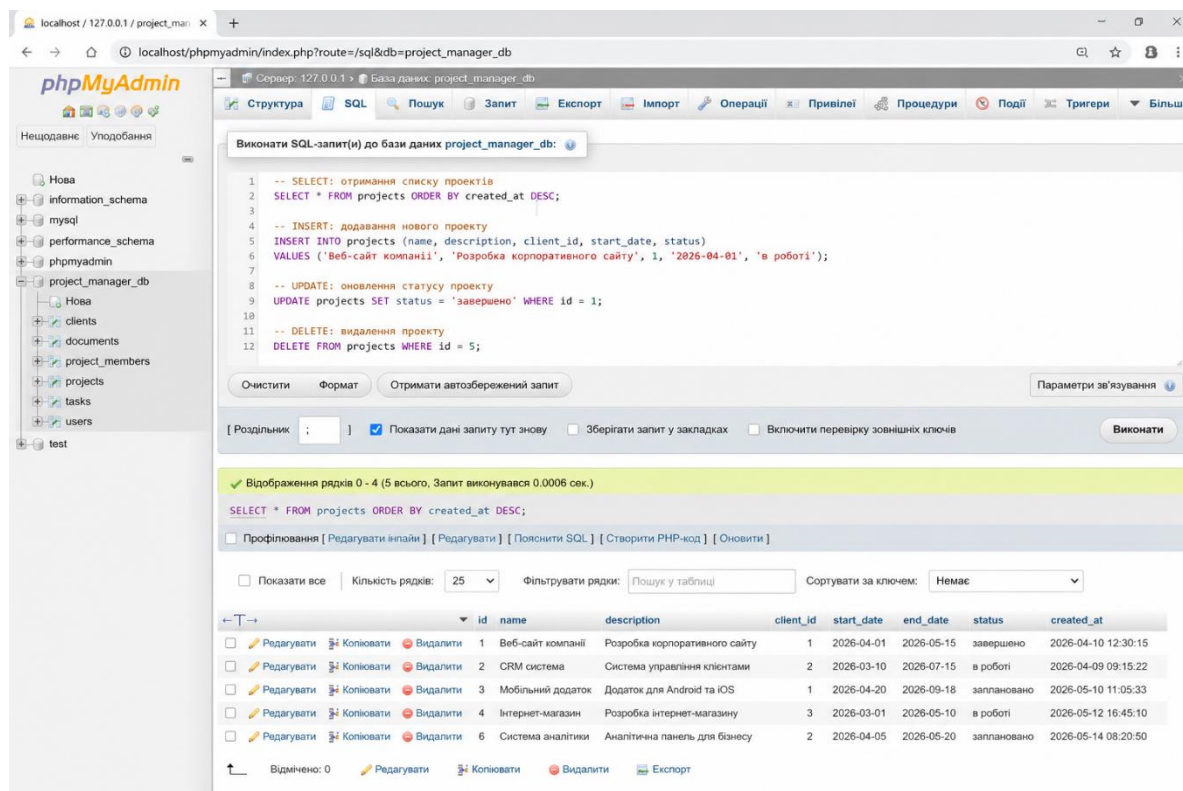


Рисунок 3.7 – Виконання SQL-запитів у phpMyAdmin

Розроблена структура бази даних забезпечує ефективне зберігання інформації, масштабованість системи та швидке виконання запитів.

3.3 Захист інформаційно-комп'ютерної системи

Одним із важливих етапів розробки програмного забезпечення є забезпечення захисту інформації та безпеки даних користувачів. У розробленій

системі реалізовано комплекс механізмів захисту, спрямованих на запобігання несанкціонованому доступу та витоку інформації (рис. 3.8).

Для захисту облікових записів користувачів використовувався алгоритм хешування bcrypt. Паролі не зберігаються у відкритому вигляді, а перетворюються у хеш-код за допомогою функції `password_hash()`.

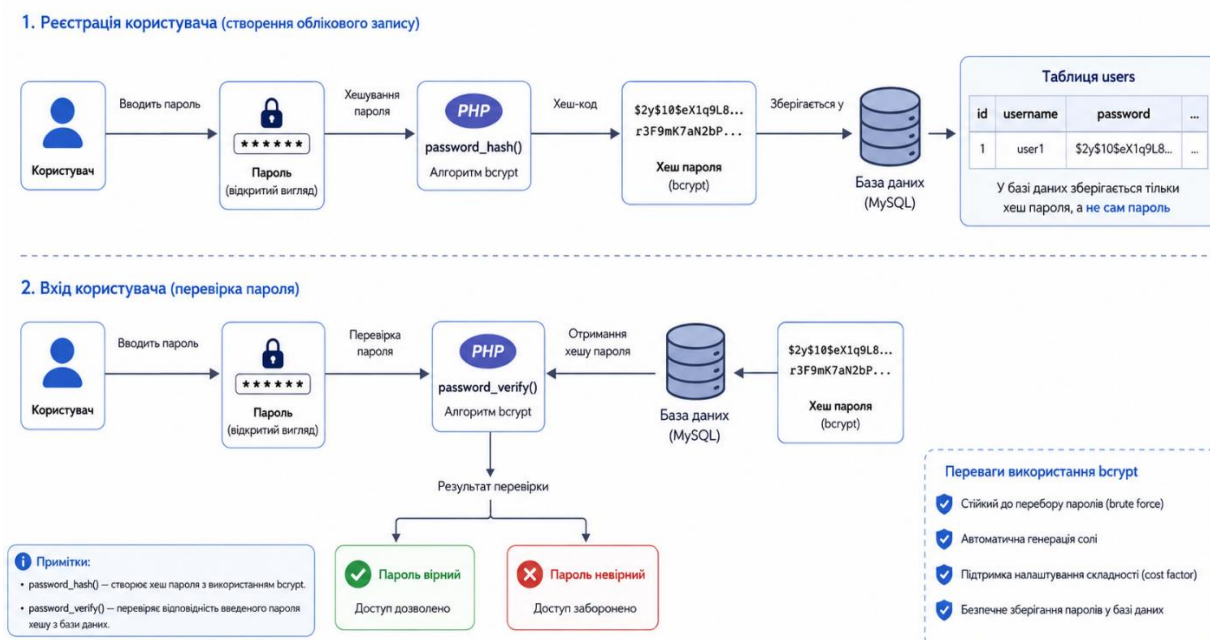


Рисунок 3.8 – Схема хешування паролів у системі

Для перевірки паролів під час авторизації використовувалась функція `password_verify()`, яка дозволяє порівнювати введений пароль із хешем, що зберігається у базі даних.

У системі реалізовано механізм авторизації користувачів на основі сесій PHP. Після успішного входу система створює унікальну сесію користувача, яка використовується для контролю доступу до сторінок системи.

Для захисту від SQL-ін'єкцій використовувались `prepared statements` (рис. 3.9). Даний механізм дозволяє відокремити SQL-код від користувацьких даних та значно підвищує рівень безпеки вебзастосунку.

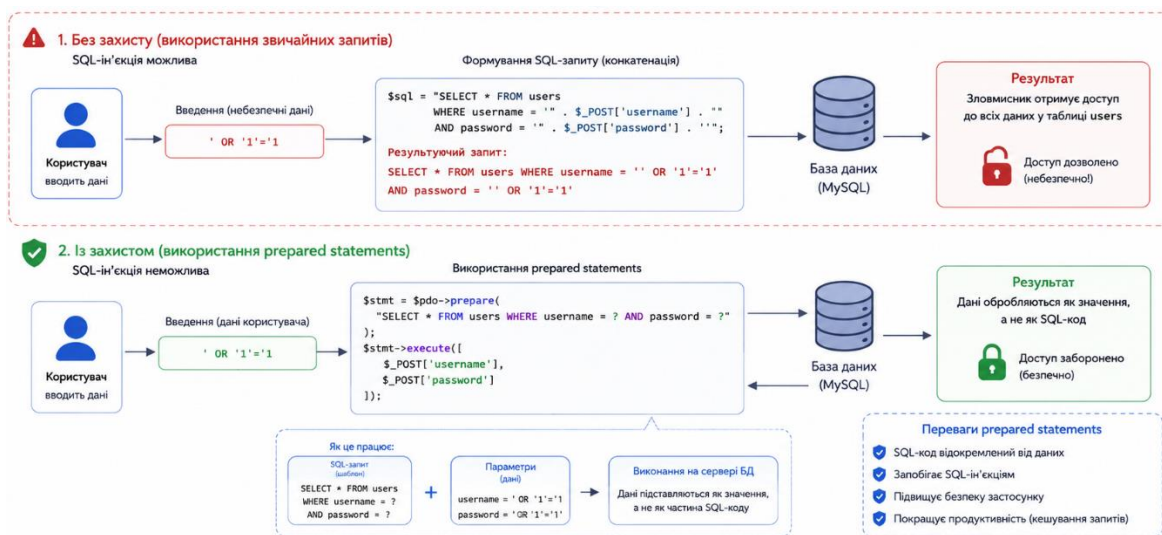


Рисунок 3.9 – Механізм захисту від SQL-ін'єкцій

Також у системі реалізовано серверну та клієнтську валідацію форм. Перевіряється правильність введення електронної пошти, довжина пароля та наявність обов'язкових полів.

Для захисту файлів та документів використовувалось обмеження типів завантажуваних файлів та перевірка MIME-типів. Це дозволяє уникнути завантаження потенційно небезпечних файлів на сервер.

У результаті реалізації механізмів захисту було забезпечено безпечне зберігання даних користувачів, контроль доступу до функціоналу системи та захист від основних типів вебзагроз.

3.4 Тестування та налагодження інформаційно-комп'ютерної системи

Після завершення реалізації програмного забезпечення було проведено тестування та налагодження інформаційної системи. Основною метою тестування була перевірка працездатності функціональних модулів, коректності взаємодії із базою даних та стабільності роботи системи (рис. 3.10).

Тестування проводилось у локальному середовищі XAMPP із використанням веббраузерів Google Chrome та Mozilla Firefox. Під час тестування перевірялись:

- робота авторизації;
- створення проєктів;
- редагування завдань;
- взаємодія із базою даних;
- коректність SQL-запитів;
- валідація форм;
- механізми захисту інформації.

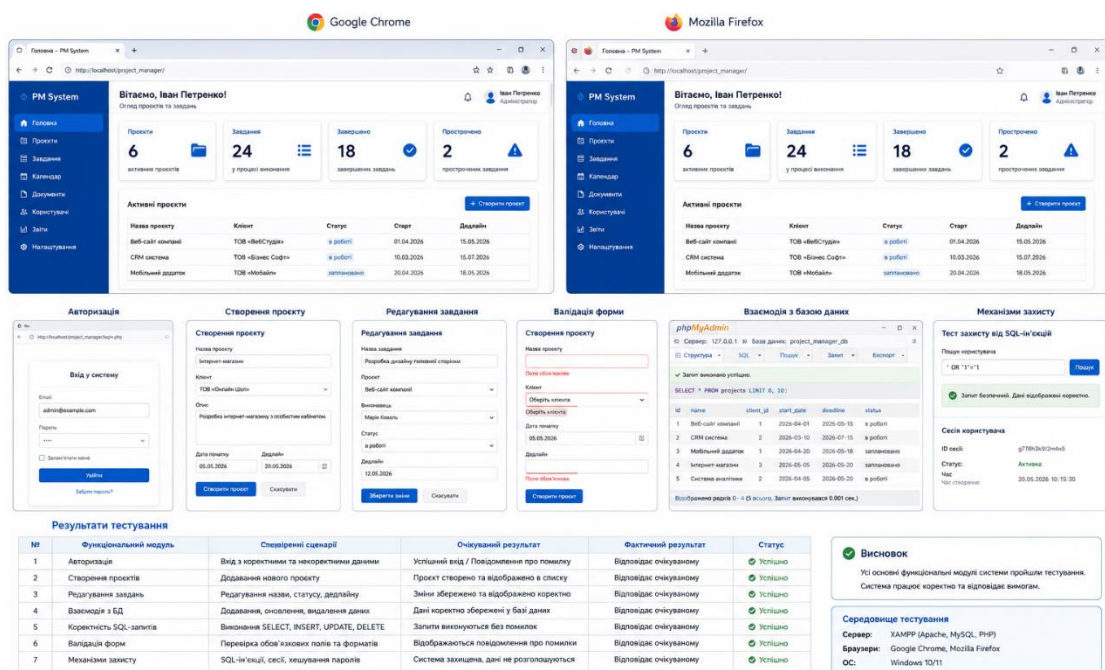


Рисунок 3.10 – Результати тестування системи у веббраузерах

Під час тестування було виявлено окремі помилки, пов'язані із некоректною обробкою порожніх полів форм та неправильним відображенням окремих елементів інтерфейсу. Після виявлення недоліків було виконано налагодження програмного забезпечення та внесено відповідні зміни до програмного коду.

Для перевірки працездатності системи використовувалось функціональне тестування. Основна увага приділялась правильності роботи CRUD-операцій та стабільності взаємодії із базою даних MySQL.

Мінімальні системні вимоги для роботи програмного забезпечення:

- операційна система Windows 10 або новіша;
- вебсервер Apache;
- PHP версії 8.0 або новішої;
- MySQL версії 8.0;
- XAMPP;
- веббраузер із підтримкою JavaScript.

Для встановлення системи необхідно:

- встановити XAMPP;
- запустити Apache та MySQL;
- скопіювати файли вебзастосунку у директорію htdocs;
- імпортувати SQL-файл бази даних через phpMyAdmin;
- налаштувати файл підключення до бази даних config.php;
- відкрити систему у браузері за адресою localhost.

У результаті тестування було підтверджено працездатність усіх основних модулів системи, коректність взаємодії із базою даних та стабільність роботи програмного забезпечення.

3.5 Розробка документації для програмного продукту

Важливим етапом розробки програмного забезпечення є створення документації для програмного продукту. Документація забезпечує спрощення процесу встановлення, налаштування та використання інформаційної системи користувачами та адміністраторами. Наявність документації дозволяє швидше ознайомитись із функціональними можливостями системи, а також забезпечує правильне розгортання вебзастосунку на локальному або віддаленому сервері.

Для розробленої системи управління проєктами було створено інструкцію користувача, яка містить опис основних функцій програмного забезпечення, порядок авторизації, створення проєктів, роботу із завданнями та управління клієнтами. Документація орієнтована на користувачів, які не мають глибоких

технічних знань, тому всі етапи роботи із системою описані максимально просто та зрозуміло.

Перед початком роботи із системою необхідно виконати встановлення програмного забезпечення та налаштування локального середовища розробки. Для цього використовується середовище XAMPP, яке забезпечує підтримку Apache, PHP та MySQL.

Для встановлення системи необхідно виконати такі дії:

- встановити середовище XAMPP;
- запустити модулі Apache та MySQL;
- скопіювати файли вебзастосунку у директорію htdocs;
- створити базу даних у phpMyAdmin;
- імпортувати SQL-файл структури бази даних;
- налаштувати параметри підключення у файлі config.php;
- відкрити систему через веббраузер за адресою localhost.

Після запуску системи користувач переходить на сторінку авторизації, де необхідно ввести логін та пароль. Після успішного входу відкривається головна сторінка системи із доступом до управління проєктами, завданнями та клієнтами.

Для створення нового проєкту користувач повинен перейти до розділу «Проекти» та натиснути кнопку створення нового запису. Після цього відкривається форма введення даних проєкту, де необхідно вказати назву, опис, дату початку та статус виконання.

Після збереження інформації дані автоматично додаються до бази даних та відображаються у загальному списку проєктів. Аналогічним чином реалізовано створення та редагування завдань, клієнтів і документів.

У документації також наведено мінімальні системні вимоги для роботи програмного забезпечення:

- операційна система Windows 10 або новіша;
- Apache Server;
- PHP версії 8.0 або новішої;
- MySQL версії 8.0;

- ХАМРР;
- сучасний веббраузер із підтримкою JavaScript.

Для забезпечення стабільної роботи системи рекомендується використовувати:

- процесор Intel Core i3 або вище;
- не менше 4 ГБ оперативної пам'яті;
- SSD-накопичувач;
- стабільне підключення до мережі Інтернет.

У процесі розробки документації також було підготовлено опис структури каталогів вебзастосунку, призначення основних РНР-файлів та порядок резервного копіювання бази даних.

Для резервного копіювання бази даних може використовуватись експорт SQL-файлів через phpMyAdmin. Це дозволяє зберегти структуру таблиць та всі дані системи для подальшого відновлення.

Таким чином, розроблена документація забезпечує спрощення процесу встановлення, налаштування та використання програмного забезпечення, а також дозволяє користувачам ефективно працювати із системою управління проєктами.

Висновки до розділу 3

У третьому розділі було виконано практичну реалізацію програмного забезпечення для спрощеного управління проєктами на основі РНР та MySQL. У процесі розробки створено веборієнтовану інформаційну систему, яка забезпечує управління проєктами, завданнями, клієнтами та користувачами через вебінтерфейс. Для реалізації серверної частини системи використовувалась мова програмування РНР, а для зберігання інформації – система керування базами даних MySQL.

У ході роботи було реалізовано основні функціональні модулі системи, зокрема модулі авторизації, управління проєктами, завданнями та клієнтами. Для

створення користувацького інтерфейсу використовувались HTML, CSS та JavaScript, що дозволило забезпечити зручну взаємодію користувача із вебсистемою.

Також було розроблено структуру бази даних та реалізовано механізми взаємодії із MySQL за допомогою SQL-запитів і prepared statements. Використання реляційної бази даних забезпечило централізоване зберігання інформації та підтримку взаємозв'язків між основними сутностями системи.

Особливу увагу було приділено захисту інформаційно-комп'ютерної системи. У програмному забезпеченні реалізовано механізми авторизації та автентифікації користувачів, алгоритм хешування паролів bcrypt, серверну та клієнтську валідацію форм, а також захист від SQL-ін'єкцій. Це дозволило підвищити рівень безпеки системи та забезпечити захист облікових даних користувачів.

У процесі тестування та налагодження було перевірено працездатність основних функціональних модулів системи, правильність взаємодії із базою даних та стабільність роботи розробки. Виявлені помилки були усунені, що дозволило забезпечити коректне функціонування програмного забезпечення.

Крім того, було розроблено документацію для програмного продукту, яка містить опис процесу встановлення, налаштування та використання системи. Наявність документації спрощує процес впровадження програмного забезпечення та забезпечує зручність його подальшого використання користувачами.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було розроблено програмне забезпечення для спрощеного управління проєктами на основі PHP та MySQL. У ході дослідження було проаналізовано сучасні системи управління проєктами, зокрема Jira, Trello, Asana та Monday.com. Аналіз показав, що значна частина існуючих програмних продуктів має складний інтерфейс, надлишковий функціонал та потребує додаткових фінансових витрат для повноцінного використання. Це підтвердило доцільність створення спрощеної веборієнтованої системи, яка забезпечує базові функції управління проєктами та може використовуватись невеликими командами й організаціями.

Було визначено основні функціональні та нефункціональні вимоги до програмного забезпечення. До функціональних можливостей системи віднесено авторизацію користувачів, управління проєктами, завданнями та клієнтами, а також взаємодію із базою даних. Серед нефункціональних вимог особливу увагу приділено безпечності, продуктивності, масштабованості, адаптивності інтерфейсу та зручності використання вебсистеми.

У процесі розробки було обґрунтовано вибір технологій та інструментальних засобів. Для реалізації серверної частини системи використано мову програмування PHP, яка забезпечує ефективну обробку запитів користувачів та інтеграцію із базою даних MySQL. Для створення користувацького інтерфейсу використовувались HTML, CSS та JavaScript. Для локального розгортання та тестування програмного забезпечення застосовувалось середовище XAMPP, що забезпечує підтримку Apache, PHP та MySQL.

Було виконано проєктування архітектури системи та структури бази даних. Для опису структури програмного забезпечення побудовано UML-діаграму класів, яка дозволила відобразити взаємозв'язки між основними компонентами системи. Також було розроблено ER-діаграму бази даних, яка визначає структуру таблиць, атрибуту та зв'язки між сутностями. Використання ER-

моделювання дозволило забезпечити цілісність даних та оптимізувати процес взаємодії із базою даних.

У ході практичної реалізації створено веборієнтовану інформаційну систему, яка забезпечує управління проєктами, завданнями, клієнтами та користувачами. Реалізовано механізми авторизації та автентифікації користувачів, функції створення, редагування та видалення проєктів і завдань, а також взаємодію із базою даних MySQL за допомогою SQL-запитів.

Під час виконання роботи було проведено тестування та налагодження програмного забезпечення. У процесі тестування перевірялась працездатність функціональних модулів системи, коректність виконання SQL-запитів, стабільність роботи системи та правильність взаємодії із базою даних. Виявлені недоліки були усунені, що дозволило забезпечити стабільну та коректну роботу програмного забезпечення.

Особливу увагу у роботі приділено реалізації механізмів захисту інформації. Для безпечного зберігання паролів використано алгоритм bcrypt та функції password_hash() і password_verify(). Для захисту від SQL-ін'єкцій застосовувались prepared statements. Також реалізовано серверну та клієнтську валідацію форм, що дозволило підвищити рівень безпеки інформаційної системи та захистити дані користувачів від несанкціонованого доступу.

У межах кваліфікаційної роботи бакалавра було розроблено документацію користувача, яка містить опис процесу встановлення, налаштування та використання програмного забезпечення. Документація включає мінімальні системні вимоги, порядок запуску вебсистеми у середовищі XAMPP та основні інструкції для роботи із системою.

Отже, у результаті виконання кваліфікаційної роботи бакалавра було досягнуто поставленої мети та реалізовано програмне забезпечення для спрощеного управління проєктами на основі PHP та MySQL, яке забезпечує ефективну організацію роботи користувачів, централізоване зберігання інформації та підтримку основних процесів управління проєктами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Capterra. Best Project Management Software. URL: https://www.capterra.com/project-management-software/?utm_source=chatgpt.com (date of access: 16.02.2026).
2. Gartner. Magic Quadrant for Collaborative Work Management. URL: <https://www.gartner.com/en> (date of access: 16.02.2026).
3. Atlassian. Jira Software Documentation. URL: <https://support.atlassian.com/jira-software-cloud/resources/> (date of access: 16.02.2026).
4. Jira для бізнесу. URL: <https://pmdoc.ua/trainings/pm-jira/> (дата звернення: 21.02.2026).
5. Trello Guide and Features. URL: https://trello.com/guide?utm_source=chatgpt.com (date of access: 21.02.2026).
6. How to use Trello to manage and inspire your team. URL: <https://www.atlassian.com/blog/trello/trello-manage-team> (date of access: 21.02.2026).
7. The only work management platform built for scale. URL: <https://asana.com/product> (дата звернення: 21.02.2026).
8. How can we help you? URL: https://support.monday.com/hc/en-us?utm_source=chatgpt.com (date of access: 21.02.2026).
9. Navigating Asana. URL: https://help.asana.com/s/article/navigating-asana?language=en_US (date of access: 21.02.2026).
10. The basics of a board. URL: <https://support.monday.com/hc/en-us/articles/115005317249-The-basics-of-a-board> (date of access: 21.02.2026).
11. PHP. Documentation. URL: https://www.php.net/docs.php?utm_source=chatgpt.com (date of access: 07.03.2026).
12. MySQL 8.0 Reference Manual. URL: https://dev.mysql.com/doc/refman/8.0/en/?utm_source=chatgpt.com (date of access: 07.03.2026).

13. Sommerville I. Software Engineering. 11th Edition. Pearson, 2021. P.187-195.
14. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 4th Edition, Addison-Wesley. URL: <https://surl.li/tpfmjl> (date of access: 17.04.2026).
15. Coronel C., Morris S. Database Systems: Design, Implementation, & Management. Cengage Learning, 2022. 816 p.
16. XAMPP Apache + MariaDB + PHP + Perl. URL: https://www.apachefriends.org/index.html?utm_source=chatgpt.com (date of access: 20.04.2026).
17. Password Storage Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html?utm_source=chatgpt.com (date of access: 24.04.2026).
18. SQL Injection Prevention Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html?utm_source=chatgpt.com (date of access: 01.05.2026).