МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКА ДЛЯ УПРАВЛІННЯ ОСОБИСТИМИ ТА КОМАНДНИМИ ЗАДАЧАМИ»

Виконав: здобувач освіти 4 курсу, групи К121-22
Спеціальність 121 Інженерія
програмного забезпечення
(шифр і спеціальність)
Тимофій ВАЛУЄВ
(ім'я та ПРІЗВИЩЕ)
Керівник Олена ШИТІКОВА
(ім'я та ПРІЗВИЩЕ)
Рецензент викладач Запорізького авіаційного фахового коледжу ім. О.Г. Івченка Артур ТІМЕНКО
(посада, вчене звання, науковий ступінь, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2026

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо–професійний ступінь фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Т.М. Смолянкова
(підпис)

“ 05 ” червня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ВАЛУЄВУ Тимофію Олександровичу
(ПРІЗВИЩЕ, ім’я та по– батькові)

1. Тема роботи «Проєктування та розробка вебдодатка для управління особистими та командними задачами»

Керівник роботи к.т.н. ШИТІКОВА Олена Вікторівна
(науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від «01» грудня 2025 р. № 1728-с

2. Строк подання студентом роботи 05.06.2026

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Сучасні інформаційні системи для управління задачами та проєктами
2. Розробка проєкту інформаційної системи Worklume
3. Програмна реалізація вебдодатка Worklume
4. Тестування та аналіз результатів роботи системи

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту, UML-діаграма прецедентів, схема бази даних, схема архітектури MVC

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.12.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	постановка задачі	Грудень 2025	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2026	виконано
3.	Розробка першого розділу:		
	огляд інформаційних систем та технологій для їхнього створення	Лютий 2026	виконано
4.	Розробка другого розділу:		
	розробка проєкту інформаційної системи та вибір технологій	Квітень 2026	виконано
5.	Розробка третього розділу:		
	розробка застосунку	Травень 2026	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2026	виконано
7.	Захист кваліфікаційної роботи	16.06.2026	виконано

Здобувач освіти _____ Тимофій ВАЛУЄВ
(підпис) (ім’я ПРІЗВИЩЕ)

Керівник роботи _____ Олена ШИТІКОВА
(підпис) (ім’я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім’я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 43 сторінки, 9 рисунків, 3 таблиці, 15 джерел

Кваліфікаційна робота містить теоретичне обґрунтування, проєктування, практичну реалізацію та тестування вебдодатка Worklume, призначеного для управління задачами, проєктами, робочими просторами та командною взаємодією. Робота складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел і додатків.

Об’єкт дослідження – процес управління задачами, проєктами та командною роботою у вебсередовищі.

Предмет дослідження – методи, засоби та програмні рішення для створення вебдодатка управління задачами, проєктами, дедлайнами, користувачами та аналітикою.

Мета роботи – розробити вебдодаток Worklume, який забезпечує створення проєктів, задач, робочих просторів, призначення виконавців, контроль дедлайнів, використання Kanban-дошки, коментарів, календаря, аналітики, профілю користувача та багатомовного інтерфейсу.

У роботі використано методи аналізу предметної області, порівняння аналогів, структурного проєктування, моделювання бази даних, розробки вебдодатків, тестування функціональності та перевірки безпеки користувацьких даних.

Практичним результатом роботи є вебдодаток Worklume, реалізований із використанням PHP, MySQL, PDO, HTML, CSS, JavaScript та архітектурного підходу MVC. Система підтримує реєстрацію та авторизацію користувачів, створення робочих просторів, проєктів і задач, зміну статусів задач на Kanban-дошці, коментарі, вкладення, теги, календар, аналітику, сповіщення, адміністрування користувачів, зміну мови інтерфейсу та покращений профіль користувача.

ВЕБДОДАТОК, TASK MANAGER, WORKLUME, PHP, MYSQL, MVC, KANBAN, ПРОЄКТ, ЗАДАЧА, АВТОРИЗАЦІЯ, БАЗА ДАНИХ, ПРОФІЛЬ КОРИСТУВАЧА

SUMMARY

Qualification work: 43 pages, 9 figures, 3 table, 15 sources

The qualification work contains the theoretical substantiation, design, practical implementation and testing of the Worklume web application intended for task, project, workspace and team collaboration management.

The object of the research is the process of task, project and team work management in a web environment. The subject of the research is the methods, tools and software solutions for creating a web application for managing tasks, projects, deadlines, users and analytics.

The purpose of the work is to develop the Worklume web application that provides project creation, task management, workspace organization, assignment of performers, deadline control, Kanban board, comments, calendar, analytics, user profile and a multilingual interface.

The work uses methods of subject area analysis, comparison of similar systems, structural design, database modeling, web application development, functionality testing and verification of user data security.

The practical result is the Worklume web application implemented using PHP, MySQL, PDO, HTML, CSS, JavaScript and the MVC architectural approach. The system supports user registration and authorization, workspaces, projects and tasks, changing task statuses on the Kanban board, comments, attachments, tags, calendar, analytics, notifications, user administration, interface language switching and an improved user profile.

KEYWORDS: WEB APPLICATION, TASK MANAGER, WORKLUME, PHP, MYSQL, MVC, KANBAN, PROJECT, TASK, AUTHORIZATION, DATABASE, USER PROFILE

ЗМІСТ

Завдання на кваліфікаційну роботу студенту 2
Реферат 4
Summary 6
Вступ 8
1 Аналіз предметної області та постановка задачі 10
1.1 Актуальність розробки вебдодатка для управління задачами 10
1.2 Аналіз предметної області 11
1.3 Огляд аналогічних програмних рішень 12
1.4 Визначення вимог до вебдодатка Worklume 13
1.5 Постановка задачі 15
2 Проєктування вебдодатку Worklume 16
2.1 Загальна характеристика системи 16
2.2 Ролі користувачів системи 16
2.3 Функціональні можливості вебдодатка 17
2.4 Проєктування структури бази даних 18
2.5 Архітектура вебдодатка 20
2.6 Проєктування інтерфейсу користувача 21
3 Реалізація вебдодатку Worklume 23
3.1 Обґрунтування вибору технологій 23
3.2 Реалізація структури проєкту 24
3.3 Реалізація реєстрації та авторизації користувачів 25
3.4 Реалізація робочих просторів і проєктів 26
3.5 Реалізація задач та Kanban-дошки 27
3.6 Реалізація коментарів, вкладень і тегів 28
3.7 Реалізація календаря, аналітики та повідомлень 29
3.8 Реалізація профілю користувача 30
3.9 Захист даних користувачів 31
4 Тестування та аналіз результатів 32
4.1 Мета і підхід до тестування 32
4.2 Тестування реєстрації, авторизації та профілю 32
4.3 Тестування проєктів, задач і Kanban-дошки 33
4.4 Тестування календаря, аналітики та повідомлень 33
4.5 Тестування безпеки 34
4.6 Аналіз отриманих результатів 34
Висновки 36
Перелік використаних джерел 38
Додаток А 40
Додаток Б 42
Додаток В 43

ВСТУП

У сучасних умовах цифровізації навчальної, професійної та підприємницької діяльності особливого значення набувають програмні засоби, які допомагають організовувати робочі процеси, планувати завдання, контролювати терміни виконання та забезпечувати командну взаємодію. Велика кількість користувачів щоденно працює з різними типами задач: навчальними, робочими, особистими, проєктними, командними. Без спеціалізованої системи такі задачі часто зберігаються у нотатках, месенджерах або таблицях, що ускладнює контроль виконання, пошук інформації та аналіз прогресу [13].

Системи управління задачами дозволяють структурувати роботу, розподіляти відповідальність між користувачами, встановлювати пріоритети, контролювати дедлайни та зберігати історію взаємодії в межах конкретного проєкту. Такі інструменти є корисними як для окремих користувачів, так і для студентів, фрилансерів, малих команд та IT-спеціалістів [14].

Актуальність теми кваліфікаційної роботи полягає в необхідності створення зручного вебдодатка для управління задачами та проєктами, який поєднує функції особистого планування, командної роботи, Kanban-дошки, календаря, коментарів, сповіщень, аналітики та профілю користувача. Такий застосунок може бути використаний у навчальному процесі, для організації роботи малих команд або керування особистими задачами.

Об’єкт дослідження – процес управління задачами, проєктами та командною роботою у вебсередовищі.

Предмет дослідження – методи та засоби розробки вебдодатка для створення, розподілу, контролю та аналізу виконання задач.

Мета кваліфікаційної роботи – розробити вебдодаток Worklume для управління особистими та командними задачами, робочими просторами, дедлайнами та взаємодією користувачів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область управління задачами та проєктами;
- розглянути існуючі аналоги програмних систем;
- визначити функціональні та нефункціональні вимоги до вебдодатка;
- спроєктувати структуру бази даних;
- розробити архітектуру вебдодатка;
- реалізувати модулі реєстрації, авторизації та профілю користувача;
- реалізувати робочі простори, проєкти, задачі та Kanban-дошку;
- додати коментарі, теги, вкладення, календар, статистику та повідомлення;
- забезпечити базовий рівень захисту даних користувачів;
- провести тестування основних функцій системи.

Практичне значення роботи полягає у створенні працездатного вебдодатка, який може бути використаний для організації навчальних, особистих або командних проєктів. Розроблена система демонструє застосування баз даних, серверної логіки, авторизації користувачів, MVC-структури, CRUD-операцій, обробки форм, інтерактивного інтерфейсу та адаптивного дизайну [12].

Під час розробки використано такі технології: PHP, MySQL, PDO, HTML, CSS, JavaScript, архітектурний підхід MVC, а також локальне середовище XAMPP для запуску та тестування вебдодатка. Бібліографічні записи у роботі оформлено з урахуванням вимог ДСТУ 8302:2015 [15].

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел і додатків. У першому розділі розглянуто предметну область і аналоги систем управління задачами. У другому розділі виконано проєктування системи, описано ролі користувачів, функціональні можливості, структуру бази даних та архітектуру. У третьому розділі наведено практичну реалізацію вебдодатка Worklume. У четвертому розділі подано тестування та аналіз результатів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність розробки вебдодатка для управління задачами

Планування та контроль виконання задач є важливими складовими ефективної організації роботи. У навчанні, професійній діяльності та командній розробці програмного забезпечення часто виникає потреба відстежувати велику кількість завдань, контролювати терміни, розподіляти відповідальність між учасниками та аналізувати прогрес виконання [13].

Традиційні способи планування, такі як паперові списки, нотатки або електронні таблиці, мають обмежені можливості. Вони не завжди дозволяють швидко змінювати статус задач, фільтрувати інформацію, переглядати прогрес, залишати коментарі або працювати з командою. Крім того, при збільшенні кількості задач такі способи стають незручними та можуть призводити до втрати важливої інформації [14].

Вебзастосунок для управління задачами дозволяє вирішити ці проблеми шляхом централізованого збереження даних, розподілу задач між користувачами, візуалізації робочого процесу та автоматичного формування статистики. Оскільки вебдодаток доступний через браузер, користувач може працювати з ним із різних пристроїв без встановлення додаткового програмного забезпечення [3].

Розробка системи для управління особистими та командними задачами є актуальною, оскільки вона спрямована на створення універсального інструменту для організації задач і проєктів. Система може використовуватися студентами для планування навчання, фрилансерами для управління замовленнями, малими командами для розподілу роботи та IT-спеціалістами для ведення проєктних задач.

1.2 Аналіз предметної області

Предметна область управління задачами охоплює процеси створення, планування, виконання, контролю та аналізу задач. Основною одиницею такої системи є задача. Задача може мати назву, опис, статус, пріоритет, дедлайн, виконавця, автора, теги, коментарі, вкладення та історію змін [13].

Управління задачами часто пов’язане з управлінням проєктами. Проєкт об’єднує набір задач, які мають спільну мету, терміни виконання та відповідальних осіб. Для зручної організації проєкти можуть розміщуватися в робочих просторах. Робочий простір дозволяє відокремлювати особисті, навчальні та командні проєкти [14].

Одним із популярних підходів до візуалізації задач є Kanban-дошка. Вона представляє задачі у вигляді карток, розміщених у колонках відповідно до їхнього статусу. Наприклад, задача може переходити між станами: «Нова», «Запланована», «В роботі», «На перевірці», «Виконана». Такий підхід дозволяє користувачам швидко бачити поточний стан роботи [7].

Важливою частиною систем управління задачами є дедлайни. Завдяки датам виконання користувач може бачити задачі на сьогодні, найближчі дедлайни та прострочені задачі. Для зручності такі задачі можуть відображатися у календарі [9].

Командна взаємодія передбачає можливість залишати коментарі до задач, призначати виконавців, додавати учасників до робочих просторів і контролювати активність користувачів. У системі також доцільно реалізувати сповіщення, які інформують користувача про зміну статусу задачі, новий коментар або наближення дедлайну [10].

1.3 Огляд аналогічних програмних рішень

На ринку існує багато програмних рішень для управління задачами та проєктами. Серед найвідоміших можна виділити Trello, Jira, Asana та ClickUp. Кожна із цих систем має власні особливості, переваги та недоліки [7-10].

Trello є зручним інструментом для організації задач у вигляді дошок, списків і карток. Його основна перевага – простота використання та наочність. Користувачі можуть створювати картки задач і переміщувати їх між колонками. Недоліком може бути обмеженість складної аналітики та розширених функцій без додаткових інтеграцій [7].

Jira частіше використовується IT-командами для управління розробкою програмного забезпечення. Вона підтримує agile-дошки, backlog, спринти, звіти, ролі користувачів і налаштування робочих процесів. Перевагою Jira є гнучкість і потужність, однак для початківців система може бути складною через велику кількість налаштувань [8].

Asana орієнтована на командну роботу, планування задач, перегляд проєктів у різних форматах і контроль дедлайнів. Система має зручний інтерфейс і підходить для бізнес-команд, але частина функцій доступна лише в платних тарифах [9].

ClickUp намагається об’єднати в одному середовищі задачі, документи, чати, дошки, календарі, dashboards та автоматизацію. Це робить систему універсальною, але одночасно досить насиченою функціями, що може ускладнювати її використання для простих задач [10].

Порівняння аналогів показує, що сучасний вебдодаток для управління задачами повинен мати зручний інтерфейс, підтримку проєктів, Kanban-дошку, дедлайни, коментарі, фільтрацію, сповіщення, аналітику та можливість командної роботи. Саме такі функції покладено в основу розробки майбутньої системи Worklume. Порівняння переваг та недоліків всіх аналогів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння аналогічних систем

Система	Основне призначення	Переваги	Недоліки
Trello	Візуальне управління задачами через дошки та картки	Простота, зрозуміла Kanban-логіка	Менше можливостей для складної аналітики
Jira	Управління розробкою ПЗ та agile-процесами	Гнучкі робочі процеси, звіти, backlog	Складніший інтерфейс для початківців
Asana	Командне планування задач і проєктів	Зручний інтерфейс, календар, проєктні перегляди	Частина можливостей доступна у платних тарифах
ClickUp	Комплексне робоче середовище для задач, документів і аналітики	Багато функцій в одному продукті	Може бути перевантаженим для простих задач
Worklume	Навчальний вебдодаток для задач, проєктів і робочих просторів	Оптимальний набір функцій, зрозумілий інтерфейс, локальна реалізація	Потребує подальшого розвитку для комерційного використання

1.4 Визначення вимог до вебдодатка Worklume

Під час аналізу предметної області було визначено основні вимоги до вебдодатка Worklume. Система повинна забезпечувати реєстрацію та авторизацію користувачів, створення робочих просторів, проєктів і задач, призначення виконавців, зміну статусів задач, перегляд Kanban-дошки, коментування задач, роботу з тегами та вкладеннями, перегляд календаря, отримання сповіщень і перегляд статистики [13].

Функціональні вимоги до системи:

- реєстрація користувача за іменем, email і паролем;

- авторизація користувача та завершення сеансу роботи;

- перевірка складності пароля під час реєстрації та зміни пароля;

- створення та редагування профілю користувача;

- вибір мови інтерфейсу: українська, російська та англійська версія інтерфейсу;

- вибір світлої або темної теми оформлення;

- створення робочих просторів для різних напрямів діяльності;

- додавання користувачів до робочого простору за email;

- створення, редагування, перегляд і видалення проєктів;

- створення, редагування, перегляд і видалення задач;

- призначення виконавців задач із числа учасників робочого простору;

- встановлення дедлайнів, статусів і пріоритетів задач;

- перегляд задач на Kanban-дошці;

- переміщення задач між колонками Kanban-дошки;

- додавання коментарів до задач;

- прикріплення файлів до задач;

- використання тегів для класифікації задач;

- перегляд календаря задач;

- перегляд аналітики та статистики;

- отримання внутрішніх повідомлень;

- адміністрування користувачів системи.

Нефункціональні вимоги до системи:

- інтерфейс повинен бути зрозумілим, сучасним і не перевантаженим зайвими елементами [11];

- сторінки мають коректно відображатися на комп’ютерах, планшетах і смартфонах [4];

- паролі користувачів мають зберігатися тільки у хешованому вигляді [6];

- SQL-запити повинні виконуватися через підготовлені запити PDO [1, 2];

- усі форми повинні мати CSRF-захист [6];

- користувач не повинен мати доступ до чужих проєктів і задач без відповідних прав [6];

- завантаження файлів повинно обмежуватися допустимими типами та розмірами;

- код проєкту має бути структурованим і придатним до подальшого розширення [12];

- система повинна мати можливість подальшого розвитку: API, email-сповіщення, мобільна версія, інтеграції.

1.5 Постановка задачі

На основі проведеного аналізу необхідно розробити вебдодаток Worklume, який забезпечує управління задачами, проєктами та командною взаємодією. Система повинна мати зрозумілий інтерфейс, підтримувати різні ролі користувачів, дозволяти створювати робочі простори, проєкти та задачі, відстежувати дедлайни, використовувати Kanban-дошку та переглядати статистику.

Для реалізації поставленої задачі необхідно створити базу даних, розробити серверну логіку, реалізувати контролери, моделі та представлення, створити адаптивний дизайн, додати JavaScript-функціональність для Kanban-дошки та забезпечити базовий рівень безпеки [1-6, 12].

2 ПРОЄКТУВАННЯ ВЕБДОДАТКУ WORKLUME

2.1 Загальна характеристика системи

Worklume – це вебдодаток для управління задачами та проєктами. Система орієнтована на студентів, фрилансерів, малий бізнес і невеликі команди, які потребують простого та зручного інструменту для організації роботи.

Основна ідея системи полягає в тому, щоб надати користувачеві єдине середовище для створення робочих просторів, проєктів і задач. Користувач може планувати задачі, призначати виконавців, контролювати дедлайни, залишати коментарі, переглядати календар і аналізувати прогрес.

Система має назву Worklume. Назва утворена від англійських слів «work» і «lume», що асоціюється з освітленням або ясністю. Таким чином, бренд підкреслює ідею впорядкування роботи, прозорого прогресу та зрозумілого керування задачами.

2.2 Ролі користувачів системи

Гість – користувач, який не авторизований у системі. Він може переглядати головну сторінку, перейти на сторінку входу або реєстрації.

Зареєстрований користувач – користувач, який має обліковий запис і може створювати робочі простори, проєкти та задачі, редагувати профіль, переглядати календар і статистику.

Учасник робочого простору – користувач, доданий до робочого простору. Він може переглядати проєкти, задачі, коментувати задачі та виконувати призначену роботу.

Менеджер робочого простору – користувач, який має розширені права в межах робочого простору. Він може керувати учасниками, створювати проєкти, призначати задачі та контролювати прогрес.

Адміністратор системи – користувач із найвищими правами доступу. Він може переглядати список користувачів, блокувати або розблоковувати облікові записи та контролювати загальний стан системи. Основні ролі та прецеденти системи наведено на рисунку 2.1.

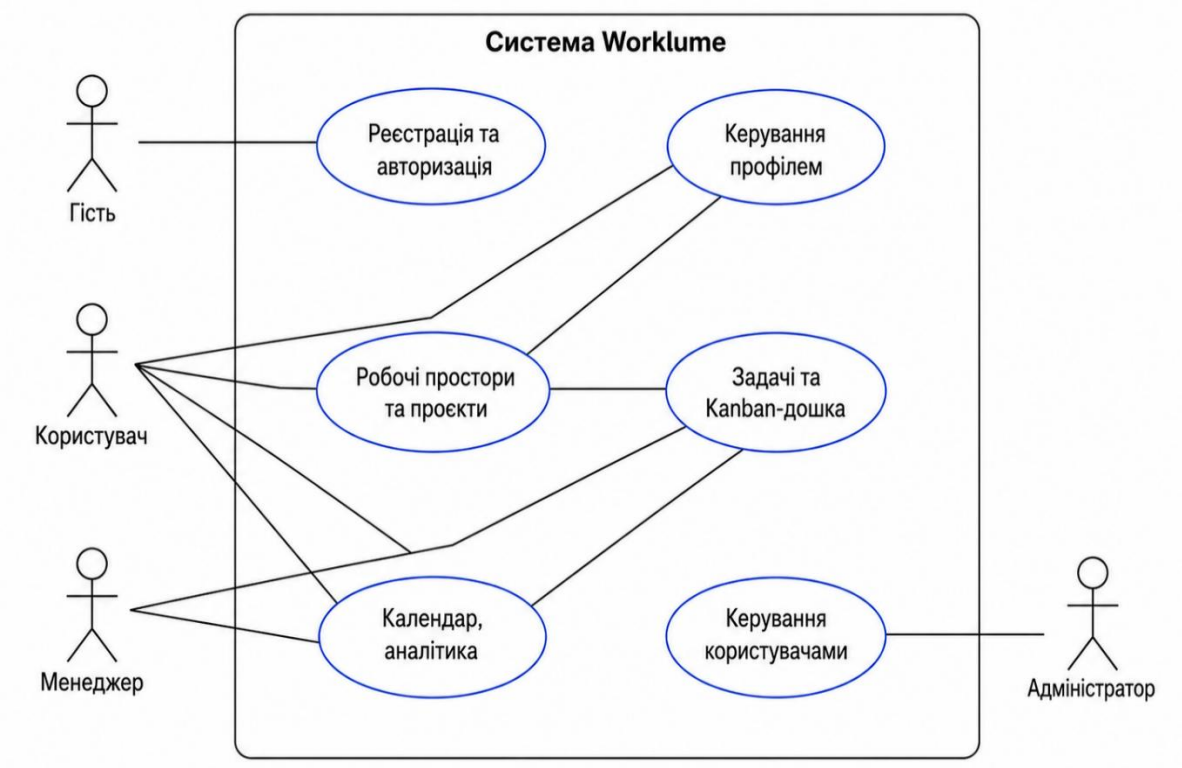


Рисунок 2.1 – UML-діаграма прецедентів

2.3 Функціональні можливості вебдодатка

Система Worklume реалізує такі основні функціональні можливості:

1) Реєстрація та авторизація користувачів. Користувач може створити обліковий запис, увійти в систему та вийти з неї. Паролі зберігаються у хешованому вигляді.

2) Робочі простори. Користувач може створювати окремі робочі простори для різних сфер діяльності: навчання, роботи, особистих задач або командних проєктів.

3) Проєкти. У межах робочого простору можна створювати проєкти, задавати назву, опис, статус і дедлайн.

4) Задачі. Задача містить назву, опис, статус, пріоритет, виконавця, дату початку, дедлайн, теги, коментарі та вкладення.

5) Kanban-дошка. Задачі відображаються у вигляді карток у колонках відповідно до статусу. Користувач може переміщувати задачі між колонками.

6) Коментарі. У задачах можна залишати коментарі, що дозволяє обговорювати виконання роботи без використання сторонніх месенджерів.

7) Вкладення. До задач можна прикріплювати файли, що дає змогу зберігати додаткові матеріали.

8) Теги. Задачі можна позначати тегами, наприклад bug, urgent, backend, frontend, documentation.

9) Календар. Користувач може переглядати задачі за датами, бачити дедлайни та прострочені задачі.

10) Аналітика. Система показує кількість задач, виконані задачі, прострочені задачі, прогрес проєктів і розподіл задач за статусами.

11) Профіль користувача. У профілі відображається інформація про користувача, його роль, мова інтерфейсу, тема оформлення, статистика задач, робочі простори та активність.

12) Адмін-панель. Адміністратор може переглядати користувачів і змінювати статус облікових записів.

2.4 Проєктування структури бази даних

Для збереження даних вебдодатка використовується реляційна база даних MySQL. Основними сутностями системи є користувачі, робочі

простори, учасники робочих просторів, проєкти, задачі, коментарі, теги, вкладення, сповіщення та історія активності [2].

Зв’язки між таблицями дозволяють організувати логіку роботи системи. Один користувач може мати багато робочих просторів. Один робочий простір може містити багато проєктів. Один проєкт може містити багато задач. Кожна задача може мати багато коментарів, тегів і вкладень. Перелік основних таблиць БД наведено в таблиці 2.1, а загальну схему бази даних показано на рисунку 2.2.

Таблиця 2.1 – Основні таблиці бази даних

Таблиця	**Призначення**	**Основні поля**
users	Зберігання користувачів	id, name, email, password_hash, role, status, language, theme
workspaces	Робочі простори	id, name, owner_id, created_at, updated_at
workspace_members	Учасники робочих просторів	id, workspace_id, user_id, role
projects	Проєкти користувачів	id, workspace_id, name, description, status, deadline
tasks	Задачі проєктів	id, project_id, title, status, priority, creator_id, assignee_id, due_date
comments	Коментарі до задач	id, task_id, user_id, comment_text
tags	Теги для задач	id, name, color, workspace_id
attachments	Файли, прикріплені до задач	id, task_id, user_id, file_name, file_path
notifications	Внутрішні повідомлення	id, user_id, type, message, is_read
activity_logs	Історія дій користувачів	id, user_id, action, entity_type, entity_id

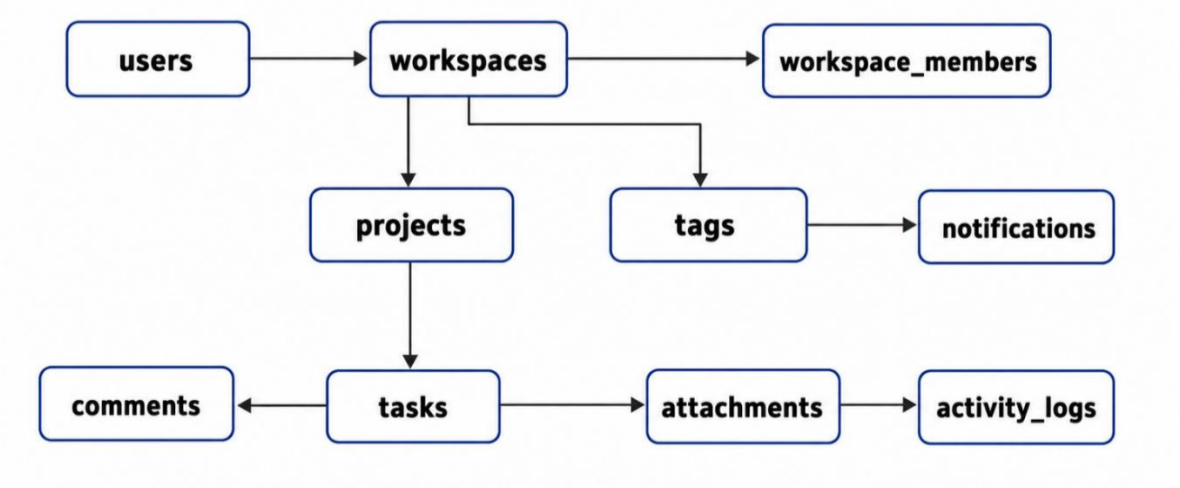


Рисунок 2.2 – Схема бази даних

2.5 Архітектура вебдодатка

Для розробки Worklume використано архітектурний підхід MVC, який передбачає поділ застосунку на три основні частини: модель, представлення та контролер [12]. Схему взаємодії компонентів показано на рисунку 2.3.

Модель відповідає за роботу з даними та базою даних. У системі створено моделі User, Workspace, Project, Task, Comment, Attachment, Tag, Notification, ActivityLog та Analytics.

Контролер відповідає за обробку запитів користувача, перевірку даних, виклик моделей і передавання результатів у представлення. У системі реалізовано контролери AuthController, DashboardController, WorkspaceController, ProjectController, TaskController, ProfileController, NotificationController, CalendarController, AnalyticsController та AdminController.

Представлення відповідає за відображення сторінок користувачу. Воно містить HTML-структуру, підключення стилів, форми, таблиці, картки задач, Kanban-дошку та інші елементи інтерфейсу.

Використання MVC дозволяє зробити структуру проєкту зрозумілою, розділити логіку роботи з даними, обробку запитів і відображення інтерфейсу. Це спрощує підтримку та подальший розвиток системи.

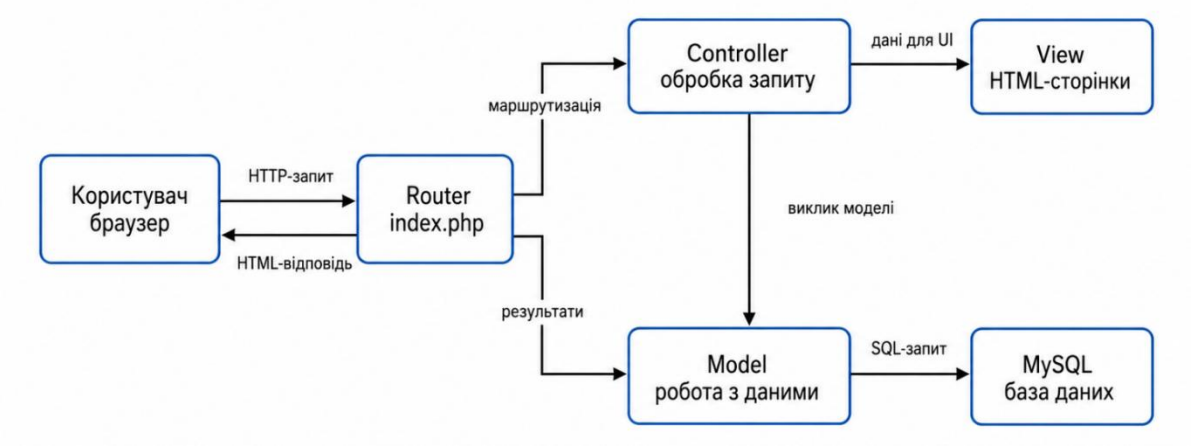


Рисунок 2.3 – Схема взаємодії компонентів

2.6 Проєктування інтерфейсу користувача

Інтерфейс Worklume спроєктовано з урахуванням зручності використання та сучасного вигляду. Основними елементами інтерфейсу є бокове меню, верхня панель, картки статистики, списки задач, Kanban-дошка, форми створення та редагування, таблиці й профіль користувача [11].

Бокове меню містить посилання на основні розділи системи: Dashboard, Workspaces, Projects, Calendar, Analytics, Notifications, Profile та Admin Panel. Верхня панель відображає назву поточної сторінки, ім’я користувача, мову інтерфейсу та кнопку виходу.

Для дизайну використано сучасний стиль із м’якими градієнтами, округленими кутами, тінями, картками та контрастними кнопками. Кнопки мають різні візуальні стани, що дозволяє користувачеві швидко розуміти їх призначення. Основний акцентний колір – фіолетово-синій, який відповідає бренду Worklume.

Система підтримує світлу та темну тему, що покращує комфорт користування. Інтерфейс адаптовано для різних розмірів екранів, тому застосунок можна використовувати як на комп’ютері, так і на мобільних пристроях [4].

Під час проєктування вебдодатка важливо було забезпечити баланс між функціональністю та простотою використання. Надмірна кількість налаштувань може ускладнити роботу користувача, тому в першій версії системи реалізовано функції, які є найбільш важливими для щоденного управління задачами.

Важливим принципом розробки стала модульність. Кожна частина системи відповідає за окремий набір функцій: авторизація відповідає за доступ до системи, модуль проєктів організовує задачі, модуль профілю зберігає налаштування користувача, а модуль аналітики надає підсумкову інформацію про стан роботи.

3 РЕАЛІЗАЦІЯ ВЕБДОДАТКУ WORKLUME

3.1 Обґрунтування вибору технологій

Для реалізації вебдодатка Worklume було обрано технології PHP, MySQL, HTML, CSS та JavaScript. Такий набір технологій є поширеним для створення динамічних вебдодатків і дозволяє реалізувати як серверну, так і клієнтську частину системи [1-5].

PHP використовується для реалізації серверної логіки, обробки форм, роботи з сесіями, перевірки прав доступу та взаємодії з базою даних. Перевагою PHP є простота використання, підтримка більшістю хостингів і зручність для навчальних вебпроєктів [1].

MySQL використовується як система керування базами даних. Вона дозволяє зберігати користувачів, проєкти, задачі, коментарі, теги, вкладення та інші дані системи. Для взаємодії з базою даних використовується PDO, що забезпечує підготовлені SQL-запити та підвищує захищеність від SQL-ін’єкцій [1, 2, 6].

HTML застосовується для побудови структури сторінок. CSS використовується для оформлення інтерфейсу, адаптивності, кольорової схеми, карток, кнопок і Kanban-дошки. JavaScript використовується для інтерактивних елементів, зокрема переміщення задач між колонками Kanban-дошки [3-5].

Для локальної розробки та тестування використано середовище XAMPP, яке містить Apache, PHP, MySQL і phpMyAdmin. Це дозволяє запускати вебдодаток на локальному комп’ютері без необхідності налаштування віддаленого сервера.

3.2 Реалізація структури проєкту

Структура проєкту Worklume організована відповідно до MVC-підходу. Основні каталоги проєкту наведено нижче на рисунку 3.1, а фрагменти програмного коду подано у додатку А.

```
task-manager-php-mysql/
├── app/
│   ├── controllers/
│   ├── core/
│   ├── lang/
│   ├── models/
│   └── views/
├── config/
├── database/
└── public/
├── assets/
├── uploads/
└── index.php
```

Рисунок 3.1 – Структура каталогу проєкту

Каталог app/controllers містить контролери, які обробляють запити користувачів. Каталог app/models містить моделі для роботи з базою даних. Каталог app/views містить файли представлень, які формують HTML-сторінки. Каталог app/core містить допоміжні класи, зокрема Database, Auth, CSRF, I18n та View.

Каталог config містить конфігураційний файл, у якому задаються параметри підключення до бази даних і назва застосунку. Каталог database містить SQL-файли для створення структури бази даних і міграцій. Приклад SQL-структури бази даних наведено у додатку Б. Каталог public є публічною частиною застосунку, через яку користувач відкриває сайт.

Після запуску застосунку користувач потрапляє на головну сторінку системи. Загальний вигляд головної сторінки наведено на рисунку 3.2.

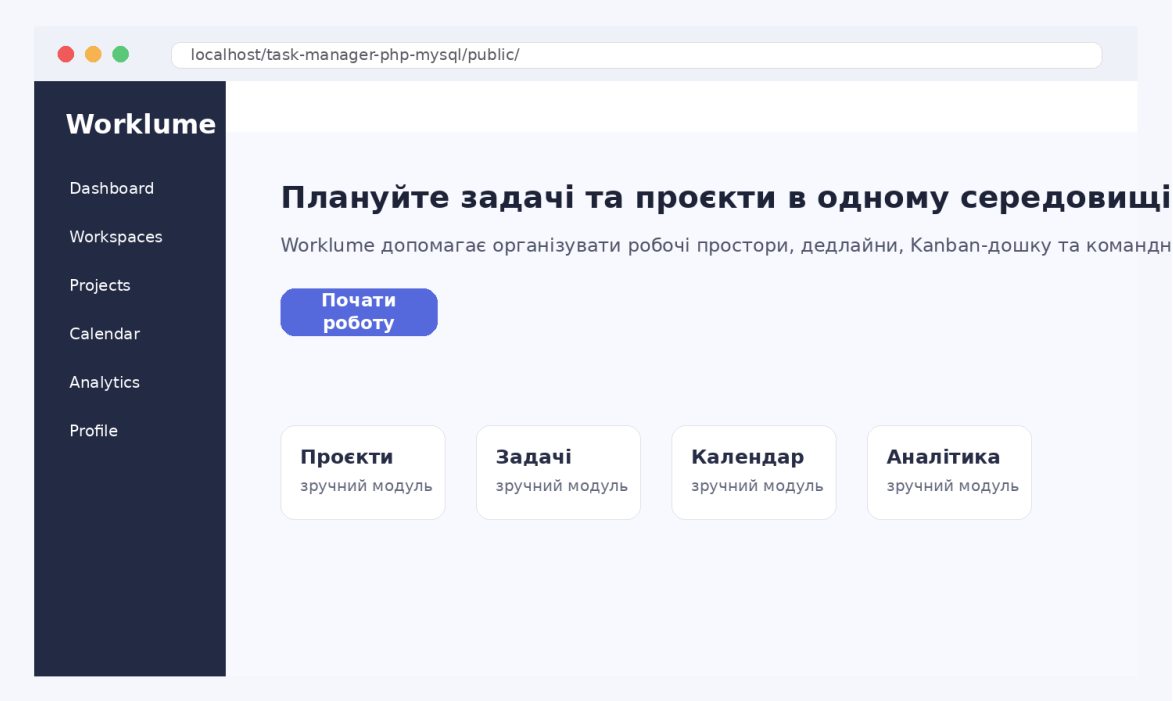


Рисунок 3.2 – Головна сторінка вебдодатка Worklume

3.3 Реалізація реєстрації та авторизації користувачів

Модуль авторизації дозволяє користувачам створювати облікові записи та входити до системи. Реєстрація передбачає введення імені, email, пароля та підтвердження пароля. Перед створенням користувача система перевіряє, чи заповнені всі поля, чи правильно введено email, чи збігаються паролі та чи не використовується email іншим користувачем.

Пароль користувача не зберігається у базі даних у відкритому вигляді. Для його збереження використовується хешування. Це підвищує безпеку системи, оскільки навіть у разі отримання доступу до бази даних неможливо одразу побачити пароль користувача [6].

Також реалізовано перевірку складності пароля. Система не дозволяє створити пароль з одного символу або дуже простий пароль. Пароль повинен містити достатню кількість символів, цифри, великі та малі літери, а також спеціальні символи.

Після успішного входу в систему в сесії зберігається ідентифікатор користувача. Це дозволяє системі визначати поточного користувача, його права доступу, мову інтерфейсу та тему оформлення. Особистий кабінет користувача після входу до системи наведено на рисунку 3.3.

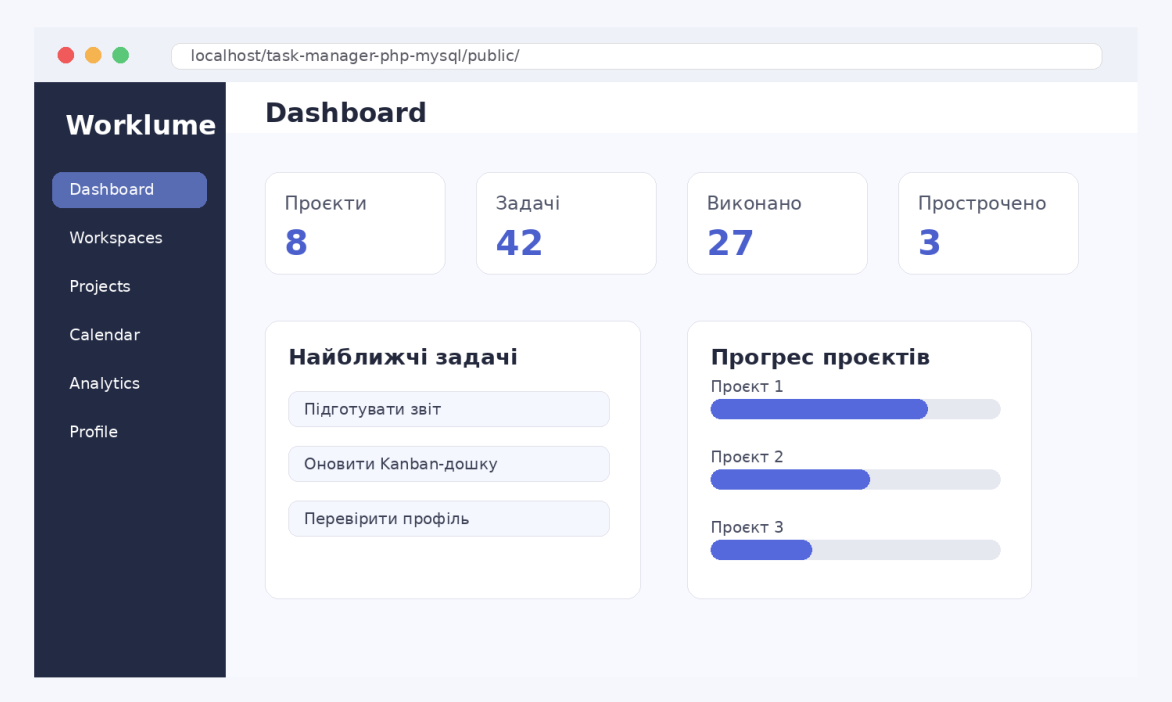


Рисунок 3.3 – Особистий кабінет користувача Worklume

3.4 Реалізація робочих просторів і проєктів

Робочі простори призначені для групування проєктів за напрямами діяльності. Наприклад, користувач може створити окремі простори «Навчання», «Особисті задачі», «Команда розробки» або «Дипломний проєкт». Кожен робочий простір має власника та список учасників.

Учасники робочого простору можуть мати різні ролі: власник, менеджер або учасник. Це дозволяє обмежувати доступ до певних дій і забезпечувати контроль над змінами.

Проєкти створюються всередині робочих просторів. Кожен проєкт має назву, опис, статус, дедлайн, автора та список задач. Статус проєкту може

відображати його поточний стан: новий, у роботі, на паузі, завершений або архівований.

На сторінці проєкту користувач може переглядати задачі, прогрес виконання, учасників і Kanban-дошку. Прогрес проєкту обчислюється на основі кількості виконаних задач відносно загальної кількості задач.

3.5 Реалізація задач та Kanban-дошки

Задача є основною сутністю системи. Вона містить назву, опис, статус, пріоритет, автора, виконавця, дату початку, дедлайн, дату завершення, теги, коментарі, вкладення та історію змін.

У Worklume передбачено такі статуси задач: нова, запланована, в роботі, на перевірці, виконана та скасована. Пріоритети задач поділяються на низький, середній, високий і критичний. Завдяки цьому користувач може швидко визначати важливість задачі та її поточний стан.

Kanban-дошка реалізована у вигляді кількох колонок. Кожна колонка відповідає певному статусу задачі. Задачі відображаються у вигляді карток, які містять назву, короткий опис, пріоритет, виконавця та дедлайн. Приклад роботи Kanban-дошки наведено на рисунку 3.4.

Для переміщення задач між колонками використовується JavaScript. Під час перетягування задачі система відправляє запит на сервер, де оновлюється статус задачі в базі даних. Такий підхід робить інтерфейс інтерактивним і зручним для користувача [5].

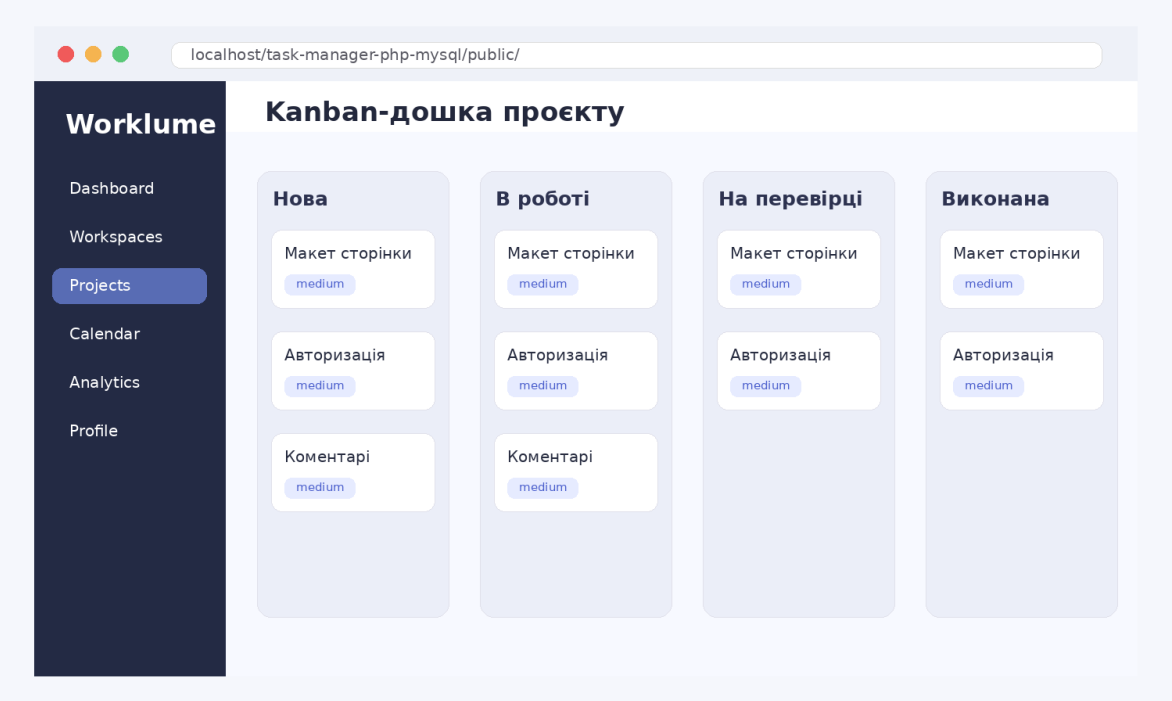


Рисунок 3.4 – Kanban-дошка проєкту у Worklume

3.6 Реалізація коментарів, вкладень і тегів

Коментарі дозволяють користувачам обговорювати конкретну задачу без використання сторонніх месенджерів. Кожен коментар має автора, текст і дату створення. Коментарі відображаються на сторінці задачі в хронологічному порядку.

Вкладення дозволяють прикріплювати до задач файли, наприклад зображення, документи або архіви. Дані про файл зберігаються у базі даних, а сам файл розміщується у спеціальному каталозі public/uploads/tasks. Для безпеки необхідно перевіряти тип і розмір файлу [6].

Теги використовуються для класифікації задач. Наприклад, задача може мати теги bug, urgent, frontend, backend або documentation. Теги мають назву та колір, що робить їх зручними для візуального сприйняття.

3.7 Реалізація календаря, аналітики та повідомлень

Календар у системі Worklume дозволяє переглядати задачі за датами. На календарній сітці відображаються дедлайни задач, що допомагає користувачу планувати роботу та не пропускати важливі терміни. Приклад календаря задач наведено на рисунку 3.5.

Аналітика показує загальну кількість задач, кількість виконаних задач, прострочені задачі, задачі за статусами та пріоритетами. Ці дані допомагають оцінити продуктивність і стан проєкту.

Повідомлення інформують користувача про важливі події в системі: створення задачі, зміну статусу, додавання коментаря або наближення дедлайну. У поточній версії реалізовано внутрішні повідомлення в інтерфейсі системи.

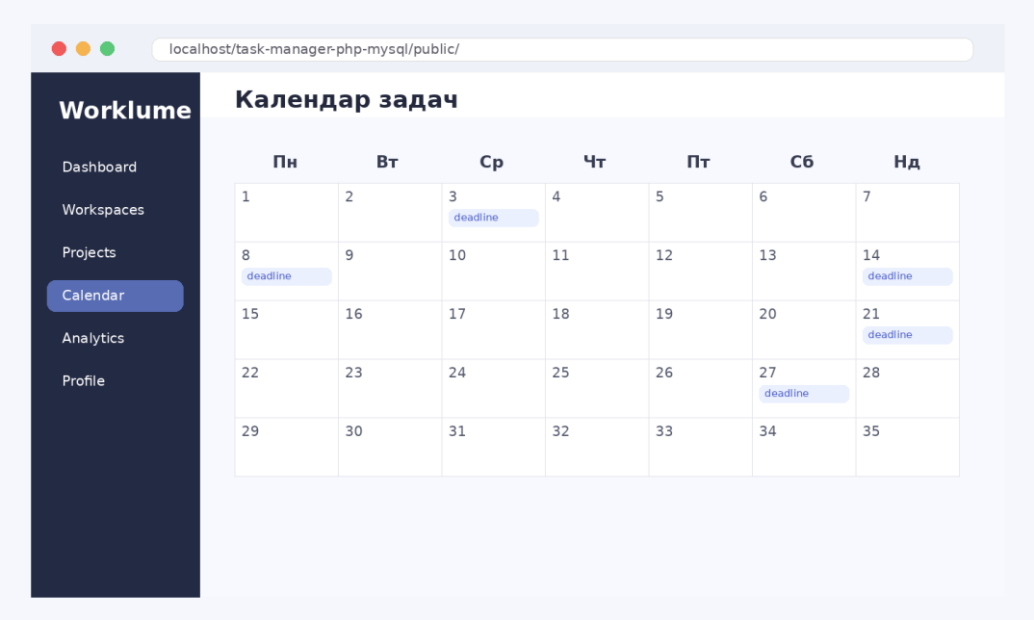


Рисунок 3.5 – Календар задач у вебдодатка Worklume

3.8 Реалізація профілю користувача

Профіль користувача є важливою частиною системи, оскільки він дозволяє користувачу переглядати та змінювати особисту інформацію. У Worklume профіль оформлено не як набір окремих полів, а як повноцінну сторінку користувача.

У профілі відображається картка користувача з аватаром у вигляді першої літери імені, ім’ям, email, роллю, статусом акаунта, мовою інтерфейсу, темою оформлення та датою реєстрації. Також відображається статистика: кількість робочих просторів, проєктів, призначених задач, створених задач, виконаних задач, прострочених задач, коментарів і вкладень.

Окремий блок профілю відповідає за безпеку. У ньому користувач може змінити пароль. Під час зміни пароля також використовується перевірка складності, щоб запобігти використанню слабких паролів.

Покращений профіль робить систему більш завершеною та зручною для користувача, а також демонструє персоналізацію та аналітичні можливості вебдодатка. Приклад сторінки профілю наведено на рисунку 3.6.

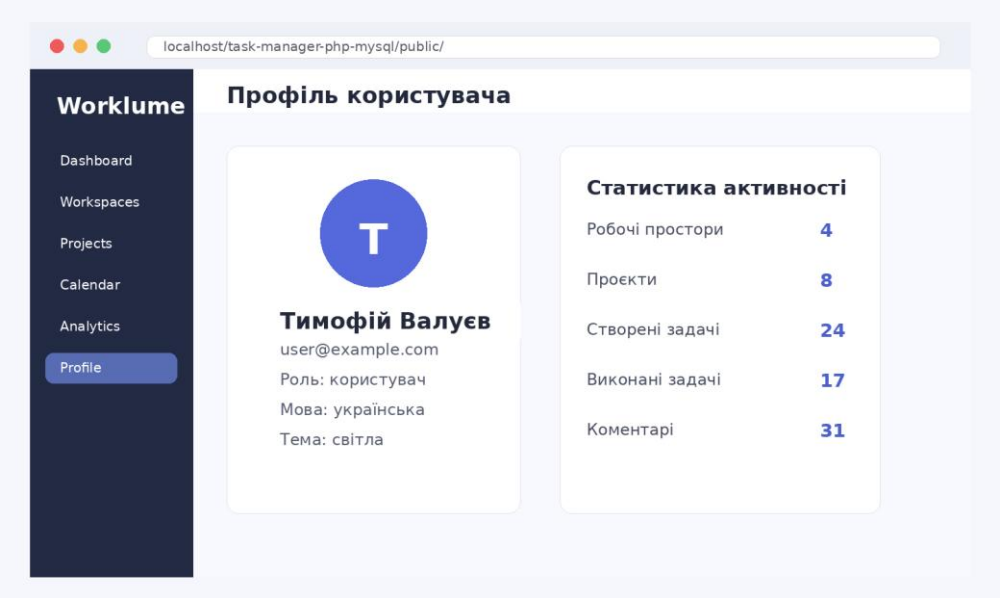


Рисунок 3.6 – Профіль користувача у Worklume

3.9 Захист даних користувачів

Безпека є одним із важливих аспектів розробки вебдодатка. У Worklume реалізовано базові механізми захисту даних користувачів [6].

Для захисту паролів використовується хешування. Це означає, що у базі даних не зберігається реальний пароль користувача, а лише його хеш. Для перевірки пароля під час входу використовується спеціальна функція порівняння.

Для захисту від SQL-ін’єкцій використовується PDO та підготовлені запити. Дані користувача не вставляються безпосередньо в SQL-рядок, а передаються як параметри [1, 2, 6].

Для захисту форм використовується CSRF-токен. Це дозволяє переконатися, що запит на зміну даних був сформований саме із сайту Worklume, а не зі стороннього ресурсу [6].

Для запобігання XSS-атакам текстові дані, які виводяться в інтерфейсі, проходять екранування. Це не дозволяє користувачу вставити шкідливий JavaScript-код у поля введення [5, 6].

Система також перевіряє права доступу. Користувач не повинен мати можливість переглядати або змінювати дані робочих просторів, проєктів і задач, до яких він не має відношення.

Інструкцію користувача для запуску та роботи із системою наведено у додатку В.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Мета і підхід до тестування

Метою тестування є перевірка працездатності основних функцій вебдодатка Worklume, відповідності реалізованих можливостей поставленим вимогам, а також виявлення помилок у роботі системи [13].

Тестування проводилося у локальному середовищі XAMPP із використанням браузера. Перевірялися основні сценарії роботи користувача: реєстрація, вхід, створення робочого простору, створення проєкту, створення задачі, зміна статусу задачі, робота з коментарями, перегляд календаря, аналітики та профілю.

Особлива увага приділялася перевірці введення некоректних даних, наприклад порожніх полів, неправильного email, слабкого пароля, відсутності обов’язкових значень або спроби доступу до чужих даних [6].

4.2 Тестування реєстрації, авторизації та профілю

Під час тестування реєстрації було перевірено створення нового користувача з коректними даними. Система успішно створює обліковий запис, хешує пароль і перенаправляє користувача до системи.

Також було перевірено введення неправильних даних. Якщо користувач вводить порожнє ім’я, неправильний email або пароль, який не відповідає вимогам складності, система показує повідомлення про помилку. Якщо email уже використовується, реєстрація не виконується.

Під час тестування авторизації перевірено вхід із правильним email і паролем, а також спроби входу з неправильними даними. У разі правильних

даних користувач потрапляє на dashboard, у разі помилки система повідомляє про неправильний email або пароль.

Профіль користувача перевірено на відображення особистої інформації, статистики та можливість зміни налаштувань. Сторінка профілю коректно показує ім’я користувача, email, роль, статус, мову, тему, кількість задач, проєктів і робочих просторів.

4.3 Тестування проєктів, задач і Kanban-дошки

Було перевірено створення робочого простору, створення проєкту в межах робочого простору та створення задачі в межах проєкту. Усі дані коректно зберігаються в базі даних і відображаються в інтерфейсі.

Перевірено редагування проєктів і задач. Після зміни назви, опису, статусу, дедлайну або пріоритету дані оновлюються та відображаються на відповідних сторінках.

Kanban-дошка перевірялася шляхом переміщення задач між колонками. Після переміщення статус задачі змінюється в базі даних, а задача відображається в новій колонці. Це підтверджує правильну роботу JavaScript-функціональності та серверного обробника зміни статусу.

4.4 Тестування календаря, аналітики та повідомлень

Під час тестування календаря було перевірено відображення задач за дедлайнами. Задачі з датою виконання відображаються у відповідних клітинках календаря. Це дозволяє користувачу бачити навантаження за днями та контролювати терміни.

Аналітика перевірялася шляхом створення задач із різними статусами та пріоритетами. Система правильно підраховує загальну кількість задач, виконані задачі, прострочені задачі та відсоток виконання проєкту.

Повідомлення перевірялися під час створення задач і зміни їхнього стану. Система формує записи повідомлень, які користувач може переглядати на окремій сторінці.

4.5 Тестування безпеки

Під час тестування безпеки перевірено, що пароль користувача не зберігається у відкритому вигляді. У таблиці users зберігається лише password_hash.

Перевірено роботу валідації форм. Система не дозволяє надсилати критично важливі форми без CSRF-токена. Це зменшує ризик підробки запитів.

Також перевірено захист від SQL-ін’єкцій. Оскільки запити виконуються через PDO та підготовлені параметри, введення спеціальних символів у поля форми не порушує роботу SQL-запиту.

Для перевірки XSS-захисту в текстові поля вводилися HTML-теги та JavaScript-код. Під час виведення в інтерфейсі ці дані екрануються, тому код не виконується в браузері.

4.6 Аналіз отриманих результатів

За результатами тестування можна зробити висновок, що вебдодаток Worklume виконує поставлені функціональні вимоги. Користувач може зареєструватися, увійти в систему, створити робочий простір, проєкт і задачу, змінити статус задачі, переглянути календар, статистику та профіль.

Система має сучасний інтерфейс, підтримує адаптивність, світлу й темну тему, багатомовність і базові механізми безпеки. Реалізована структура коду дозволяє в майбутньому розширювати функціональність застосунку.

До перспектив розвитку можна віднести додавання повноцінного email-підтвердження, відновлення пароля, REST API, push-сповіщень, інтеграції з Google Calendar, GitHub, Telegram-ботом, а також створення мобільного застосунку.

У таблиці 4.1 наведено приклади тестових сценаріїв.

Таблиця 4.1 – Приклади тестових сценаріїв

№	Тестовий сценарій	Очікуваний результат	Результат
1	Реєстрація з коректними даними	Обліковий запис створено	Успішно
2	Реєстрація зі слабким паролем	Система показує помилку	Успішно
3	Вхід із правильним паролем	Користувач потрапляє в dashboard	Успішно
4	Створення проєкту	Проєкт збережено та відображено	Успішно
5	Створення задачі	Задача збережена в проєкті	Успішно
6	Переміщення задачі на Kanban-дошці	Статус задачі змінено	Успішно
7	Додавання коментаря	Коментар відображено на сторінці задачі	Успішно
8	Перегляд календаря	Задачі показано за датами	Успішно
9	Перегляд профілю	Відображено картку та статистику користувача	Успішно

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто процес проєктування та розробки вебдодатка Worklume для управління особистими та командними задачами. Під час виконання роботи проаналізовано предметну область, розглянуто аналогічні програмні рішення, визначено вимоги до системи, спроєктовано структуру бази даних, архітектуру застосунку та інтерфейс користувача.

Практичним результатом роботи є вебдодаток Worklume, реалізований із використанням PHP, MySQL, PDO, HTML, CSS та JavaScript. Система має MVC-структуру, що забезпечує логічний поділ коду на моделі, контролери та представлення.

У вебдодатка реалізовано реєстрацію та авторизацію користувачів, перевірку складності пароля, робочі простори, проєкти, задачі, Kanban-дошку, коментарі, вкладення, теги, календар, статистику, повідомлення, адмін-панель і покращений профіль користувача. Також передбачено підтримку кількох мов інтерфейсу та світлої/темної теми.

Під час розробки враховано базові вимоги безпеки: хешування паролів, використання підготовлених SQL-запитів, CSRF-захист форм, екранування даних під час виведення та перевірку прав доступу. Це підвищує надійність системи та демонструє правильний підхід до створення вебдодатків.

Проведене тестування підтвердило працездатність основних функцій системи. Worklume дозволяє користувачу створювати проєкти, керувати задачами, змінювати їхні статуси, переглядати календар і статистику, а також працювати з профілем користувача.

Розроблений вебдодаток може бути використаний як навчальний, особистий або командний інструмент для організації роботи. У майбутньому систему можна розширити шляхом додавання email-підтвердження,

відновлення пароля, API, push-сповіщень, інтеграції з календарями, Telegram-ботом і мобільним застосунком.

Отже, поставлену мету кваліфікаційної роботи досягнуто, а всі основні завдання виконано.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP Manual. Офіційна документація мови PHP. URL: https://www.php.net/manual/ (дата звернення: 20.04.2026).

2. MySQL 8.0 Reference Manual. Офіційна документація MySQL. URL: https://dev.mysql.com/doc/ (дата звернення: 20.04.2026).

3. MDN Web Docs. HTML: HyperText Markup Language. URL: https://developer.mozilla.org/ (дата звернення: 20.04.2026).

4. MDN Web Docs. CSS: Cascading Style Sheets. URL: https://developer.mozilla.org/ (дата звернення: 20.04.2026).

5. MDN Web Docs. JavaScript Guide. URL: https://developer.mozilla.org/ (дата звернення: 20.04.2026).

6. OWASP Foundation. Web Security Testing Guide. URL: https://owasp.org/ (дата звернення: 20.04.2026).

7. Trello. Офіційний вебсайт. URL: https://trello.com/ (дата звернення: 20.04.2026).

8. Jira. Офіційний вебсайт Atlassian Jira. URL: https://www.atlassian.com/software/jira (дата звернення: 20.04.2026).

9. Asana. Офіційний вебсайт. URL: https://asana.com/ (дата звернення: 20.04.2026).

10. ClickUp. Офіційний вебсайт. URL: https://clickup.com/ (дата звернення: 20.04.2026).

11. Bootstrap Documentation. Офіційна документація. URL: https://getbootstrap.com/ (дата звернення: 20.04.2026).

12. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002. 560 p.

13. Pressman R. S. Software Engineering: A Practitioner’s Approach. New York: McGraw-Hill, 2014. 976 p.

14. Sommerville I. Software Engineering. Boston: Pearson, 2016. 816 p.

15. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ: ДП «УкрНДНЦ», 2016. 16 с.

ДОДАТОК А

Фрагменти програмного коду

Нижче наведено приклади фрагментів програмного коду, які ілюструють реалізацію окремих частин вебдодатка Worklume. Приклад підключення до бази даних через PDO:

```
class Database
{
   private static ?PDO $connection = null;

   public static function connection(): PDO
   {
      if (self::$connection === null) {
         self::$connection = new PDO(
            'mysql:host=' . DB_HOST . ';dbname=' . DB_NAME . ';charset=utf8mb4',
            DB_USER,
            DB_PASS,
            [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
         );
      }
      return self::$connection;
   }
}
```

Приклад перевірки складності пароля:

```
function validatePasswordStrength(string $password): array
{
   $errors = [];
   if (mb_strlen($password) < 8) {
      $errors[] = 'Пароль повинен містити мінімум 8 символів.';
   }
   if (!preg_match('/[A-Z]/', $password)) {
      $errors[] = 'Пароль повинен містити велику літеру.';
   }
   if (!preg_match('/[a-z]/', $password)) {
      $errors[] = 'Пароль повинен містити малу літеру.';
   }
   if (!preg_match('/[0-9]/', $password)) {
      $errors[] = 'Пароль повинен містити цифру.';
   }
```

```
    if (!preg_match('/[^A-Za-z0-9]/', $password)) {
        $errors[] = 'Пароль повинен містити спеціальний символ.';
    }
    return $errors;
}
```

Приклад маршрутизації в публічному файлі index.php:

```
$route = $_GET['route'] ?? 'home';

match ($route) {
    'home' => Auth::check() ? redirect('dashboard') : View::render('auth/home'),
    'login' => isPost() ? AuthController::login() : AuthController::showLogin(),
    'dashboard' => DashboardController::index(),
    'projects' => ProjectController::index(),
    'profile' => ProfileController::show(),
    default => http_response_code(404),
};
```

ДОДАТОК Б

Структура бази даних

У додатку наведено скорочений SQL-фрагмент створення основних таблиць бази даних вебдодатка Worklume.

```
CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   email VARCHAR(150) NOT NULL UNIQUE,
   password_hash VARCHAR(255) NOT NULL,
   role VARCHAR(30) DEFAULT 'user',
   status VARCHAR(30) DEFAULT 'active',
   language VARCHAR(5) DEFAULT 'uk',
   theme VARCHAR(10) DEFAULT 'light',
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);

CREATE TABLE projects (
   id INT AUTO_INCREMENT PRIMARY KEY,
   workspace_id INT NOT NULL,
   name VARCHAR(150) NOT NULL,
   description TEXT,
   status VARCHAR(30) DEFAULT 'new',
   deadline DATE,
   created_by INT NOT NULL,
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);

CREATE TABLE tasks (
   id INT AUTO_INCREMENT PRIMARY KEY,
   project_id INT NOT NULL,
   title VARCHAR(180) NOT NULL,
   description TEXT,
   status VARCHAR(30) DEFAULT 'new',
   priority VARCHAR(30) DEFAULT 'medium',
   creator_id INT NOT NULL,
   assignee_id INT,
   start_date DATE,
   due_date DATE,
   completed_at DATETIME,
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

ДОДАТОК В

Інструкція користувача

Для запуску вебдодатка Worklume необхідно встановити XAMPP, скопіювати папку проєкту до каталогу htdocs, запустити Apache і MySQL, створити базу даних task_manager та імпортувати файл database/schema.sql через phpMyAdmin.

Після запуску локального сервера застосунок відкривається за адресою: http://localhost/task-manager-php-mysql/public/.

Порядок роботи користувача із системою:

1) Відкрити головну сторінку Worklume.
2) Зареєструвати новий обліковий запис.
3) Увійти до системи за email і паролем.
4) Створити робочий простір.
5) Створити проєкт у межах робочого простору.
6) Додати задачі до проєкту.
7) Призначити виконавців і дедлайни.
8) Керувати задачами через Kanban-дошку.
9) Переглядати календар і статистику.
10) За потреби змінити налаштування у профілі користувача.

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничий фаховий коледж ЗНУ»

Я, Валуєв Тимофій Олександрович, здобувач освіти 4 курсу, спеціальності 121 Інженерія програмного забезпечення, освітньо-професійної програми Розробка програмного забезпечення, групи К121-22, адреса електронної пошти 30@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Проєктування та розробка вебдодатка для управління особистими та командними задачами» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 05.06.2026 Підпис  Тимофій ВАЛУЄВ

Дата _____ Підпис _____ Олена ШИТІКОВА