ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
(повне найменування вищого навчального закладу)
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
(повне найменування інституту, назва факультету (відділення))
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра
(освітньо-кваліфікаційний рівень)

на тему: «Розробка розподіленої REST API системи для багатокористувацького управління завданнями»

Виконав: здобувач *2* курсу, групи *2ПРс/1*
напряму підготовки (спеціальності)
121 «Інженерія програмного забезпечення»
(шифр і назва напряму підготовки, спеціальності)
Білий Володимир Вікторович
(прізвище та ініціали)
Керівник к.т.н, доцент Хохлов В. А.
(прізвище та ініціали)
Рецензент к.т.н., доцент М.О. Вороненко
(прізвище та ініціали)

Хмельницький – 2025 року

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення *Інформаційних технологій та дизайну*
Кафедра *Програмних засобів і технологій*
Освітньо-кваліфікаційний рівень *бакалавр*
ОПП *Програмна інженерія*
(шифр і назва)
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
програмних засобів і технологій
к.т.н., доц. О.Є.Огнєва
«___» _____ 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Білий Володимир Вікторович
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) «Розробка розподіленої REST API системи для багатокористувацького управління завданнями»

керівник проекту (роботи) доцент Хохлов Вадим Анатолійович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «_20_» січня 2025 року №_99-с_

2. Строк подання здобувачем проекту(роботи) 17.06.2025

3. Вихідні дані до проекту (роботи)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної області
2. Аналіз вимог та проєктних специфікацій
3. Проєктування програмного продукту
4. Реалізація та впровадження системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Комп'ютерна презентація

6. Консультанти розділів проекту (роботи)

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	20.01.2025	Виконано
2	Підбір літератури	21.01.2025 – 31.01.2025	Виконано
3	Аналіз предметної області та наявних рішень	01.02.2025 – 10.02.2025	Виконано
4	Формулювання вимог і проєктних специфікацій	11.02.2025 – 20.02.2025	Виконано
5	Архітектурне проєктування системи	21.02.2025 – 05.03.2025	Виконано
6	Проєктування бази даних	06.03.2025 – 20.03.2025	Виконано
7	Розробка REST-API	21.03.2025 – 30.04.2025	Виконано
8	Тестування та налаштування інфраструктури	01.05.2025 – 15.05.2025	Виконано
9	Оформлення пояснювальної записки	16.05.2025- 23.05.2025	Виконано
10	Захист кваліфікаційної роботи	17.06.2025	Виконано

Здобувач _____ Білий В.В.
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ Хохлов В. А.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 89 сторінок, 27 рисунків, 2 додатки, 25 джерел.

Мета роботи – розробка розподіленої REST API системи для багатокористувацького управління завданнями, основне завдання якої полягає в забезпеченні надійного, масштабованого та безпечного сервісу для спільної роботи над тасками з використанням сучасних вебтехнологій і протоколів обміну даними.

Об'єкт дослідження – розподілені REST API системи для багатокористувацького управління завданнями.

Предмет дослідження – проєктування архітектури, безпека, реалізація та тестування розподіленої REST API системи з підтримкою ролей, запрошень, фільтрації та взаємодії з базою даних. Вибір і застосування сучасних фреймворків, методів аутентифікації та авторизації, а також механізмів синхронізації стану між мікросервісами.

Методи дослідження – аналіз сучасних підходів і рішень до побудови REST API та розподілених систем; моделювання системних вимог і потоків даних (DFD, UML); проєктування ER-моделі бази даних; експериментальна реалізація мікросервісної архітектури; функціональне та навантажувальне тестування.

Результат роботи:

– Реалізовано REST API систему для багатокористувацького управління завданнями на основі модульної монолітної архітектури;

– Базу даних проєктовано і використано через ORM, що забезпечує абстракцію від фізичного рівня;

– Розроблено ключові ендпоінти для CRUD-операцій із завданнями, управління ролями та аутентифікації через JWT;

– Проведено тестування функціональне, продуктивності та безпеки з обґрунтованими результатами.

Новизна роботи – реалізація модульної монолітної REST API системи для багатокористувацького управління завданнями з підтримкою розмежування

доступу на основі ролей. У роботі впроваджено гнучку модель авторизації з використанням JWT, яка дозволяє масштабувати систему під різні сценарії використання. Архітектурні рішення забезпечують чисту модульність, розширюваність та придатність до подальшого поділу на мікросервіси за потреби.

Ключові слова: REST API, багатокористувацька система, управління завданнями, JWT, авторизація, модульна архітектура, проєктування, розробка.

АНОТАЦІЯ

Кваліфікаційна робота бакалавра має наступну структуру: вступ, чотири розділи, висновки, список використаних джерел та додатки.

Перший розділ «Аналіз предметної області» присвячено дослідженню наявних архітектурних підходів (монолітного та мікросервісного), бізнес-процесів управління завданнями та огляду сучасних рішень. У розділі сформульовано функціональні й нефункціональні вимоги до системи, а також обґрунтовано доцільність створення нової модульно-монолітної системи з відкритим REST API.

Другий розділ «Аналіз вимог та проєктних специфікацій» включає формалізацію функціональних сценаріїв роботи системи, побудову Use Case-діаграм і діаграм активностей, опис обраного стека технологій (Django, Django REST Framework, PostgreSQL, SimpleJWT). У цьому розділі також представлено API-прототип системи у Swagger-документації, що забезпечує прозору взаємодію з клієнтськими застосунками.

Третій розділ «Проєктування програмного продукту» охоплює архітектурне проєктування системи у вигляді UML-діаграм компонентів, побудову ER-діаграми бази даних, опис класів та їх взаємодій через Sequence-діаграми. Також подано механізми забезпечення безпеки: авторизація через JWT, кастомні дозволи доступу до проєктів, throttle-захист та централізоване логування. Окрему увагу приділено ролям користувачів і моделі запрошень через share links.

Четвертий розділ «Реалізація та впровадження системи» містить опис REST API, реалізованого за допомогою Django REST Framework, з прикладами ендпоінтів і ключових модулів. Розглянуто інфраструктурну частину: CI/CD-процеси на GitHub Actions, контейнеризація через Docker. Надано інструкції для користувача й адміністратора щодо встановлення, налаштування та використання системи.

Завершується розділ оглядом можливих напрямів розвитку – реалізація push-сповіщень, перехід до мікросервісної архітектури та інше

ABSTRACT

The bachelor's thesis has the following structure: introduction, four chapters, conclusions, references, and appendices.

The first chapter, "Analysis of the Subject Area," is devoted to the study of existing architectural approaches (monolithic and microservice), the business processes of task management, and an overview of modern solutions. It formulates the functional and non-functional requirements of the system and provides the rationale for developing a new modular-monolithic system with an open REST API.

The second chapter, "Analysis of Requirements and Project Specifications," includes the formalization of system use cases, the development of use-case and activity diagrams, and a description of the selected technology stack (Django, Django REST Framework, PostgreSQL, SimpleJWT). This chapter also presents the API prototype documented with Swagger, enabling transparent interaction with client applications.

The third chapter, "Design of the Software Product," covers the system's architectural design using UML component diagrams, the database ER diagram, and class descriptions with sequence diagrams illustrating their interactions. It also describes security mechanisms: JWT-based authorization, custom project access permissions, request throttling, and centralized logging. Special attention is given to user role management and the invitation model implemented via shareable links.

The fourth chapter, "System Implementation and Deployment," describes the REST API implemented with Django REST Framework, including endpoint examples and core modules. The infrastructure section details CI/CD pipelines using GitHub Actions and Docker containerization. User and administrator instructions for installation, configuration, and usage are provided. The chapter concludes with potential future enhancements, such as push notifications and a transition to a microservice architecture.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Скорочення	Позначення
API	Application Programming Interface, спосіб взаємодії комп'ютерних програм між собою
CRUD	Create Read Update Delete, 4 основні функції управління даними «створення, читання, оновлення та вилучення»
Django	Високорівневий Python-фреймворк розробки вебсистем
DRF	Django REST Framework – інструмент побудови Web API
ER-Diagram	Entity-Relationship Diagram, Модель «сутність – зв'язок»
Git	Система керування версіями файлів та спільної роботи
HTTP	Hypertext Transfer Protocol, протокол передачі даних
IDE	Integrated Development Environment, Інтегроване середовище розробки програмного забезпечення
JSON	JavaScript Object Notation, текстовий формат обміну даними між комп'ютерами
JWT	JSON Web Tokens, стандарт токена доступу
ORM	Object-relational mapping, технологія програмування, яка зв'язує бази даних з концепціями об'єктноорієнтованих мов програмування
Replication	Техніка створення копій даних з однієї бази даних в іншій
Sharding	Розподіл даних між фізичними БД або серверами
SQL	Structured Query Language, мова структурованих запитів для взаємодії користувача з базами даних
URL	Uniform Resource Locator, уніфікований локатор ресурсів
VS Code	Visual Studio Code, Редактор початкового коду і дебагер
БД	База даних
ЕОМ	Електронна обчислювальна машина
ОС	Операційна система
ПЗ	Програмне забезпечення
ПК	Персональний комп’ютер
СКБД\СУБД	Система керування\управління базами даних

ВСТУП

У сучасних умовах цифрової трансформації особливого значення набувають засоби автоматизації управління проєктами, завданнями та спільною роботою в команді. Особливої актуальності ці процеси набули з розвитком віддаленої праці, DevOps-культури та зростанням кількості технологічних стартапів. Ефективне управління завданнями вимагає не лише гнучкої системи планування, але й надійного бекенд-рішення, що забезпечить масштабовану, безпечну та доступну взаємодію між користувачами. Саме такі цілі покладені в основу даної роботи.

REST API – це сучасний стандарт організації взаємодії між фронтендом і сервером. У поєднанні з фреймворком Django REST Framework він дозволяє створювати розширювані й підтримувані серверні рішення. У цій кваліфікаційній роботі розглядається побудова backend-системи TaskManagerSystem – модульного застосунку для багатокористувацького управління завданнями з підтримкою прав доступу, запрошень до проєктів та ролей.

Актуальність теми. Тема розробки систем управління завданнями залишається однією з найбільш потрібних у сфері програмної інженерії. Попри наявність великої кількості готових рішень (Jira, Trello, Asana), вони часто є платними, надто складними або не дозволяють гнучко кастомізувати архітектуру під конкретні потреби. У цьому контексті побудова власного серверного ядра з відкритим API є як практичною задачею, так і корисним досвідом для подальшої професійної діяльності.

Об’єкт дослідження - розподілені системи управління задачами на основі REST-архітектури.

Предмет дослідження - модульна розробка серверної частини таск-менеджера з використанням Django, Django REST Framework, JWT-автентифікації та PostgreSQL.

Методи дослідження – розробка REST API системи для багатокористувацького управління завданнями з можливістю групування по проєктах, ролями доступу, фільтрацією задач і підтримкою захищеного підключення клієнтів.

Основною задачею кваліфікаційної роботи бакалавра розробка розподіленої REST API системи для багатокористувацького управління завданнями.

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз предметної області та оцінити існуючі системи управління завданнями;

– сформулювати функціональні й нефункціональні вимоги до системи;

– побудувати проєктні специфікації, діаграми взаємодій і структури даних;

– розробити архітектуру додатку та спроєктувати базу даних;

– реалізувати REST API для управління користувачами, проєктами, завданнями та категоріями;

– реалізувати систему контролю доступу, ролей і запрошень через share links;

– провести модульне та інтеграційне тестування;

– підготувати інфраструктуру CI/CD (GitHub Actions), Docker-контейнери;

– підготувати документацію API (Swagger, ReDoc) та інструкції користувача

Наукова новизна одержаних результатів. Новизна полягає в поєднанні модульно-монолітної архітектури з реалізацією гнучкої рольової моделі та системи динамічних запрошень до проєктів. Особливістю реалізації є суворе дотримання принципів розділення відповідальностей (views.py → services.py → models.py), використання кастомних дозволів (permissions.py) та централізоване логування з обробкою помилок. Крім того, система масштабована та легко адаптується під мікросервісну архітектуру.

Практичне значення одержаних результатів. Розроблена система може бути використана в малих і середніх командах для управління проєктами, а також

як база для створення повноцінного таск-менеджера з клієнтським інтерфейсом. Вона також демонструє принципи тестування, CI/CD, безпечної авторизації та масштабованого дизайну, що є актуальним у сучасному розробницькому середовищі.

Апробація результатів бакалаврської кваліфікаційної роботи. Результати дипломної роботи можуть бути використані в реальних умовах при створенні внутрішніх систем обліку задач або MVP SaaS-продуктів для малого бізнесу.

Структура: робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, додатків. Кваліфікаційна робота бакалавра складається з 89 сторінок, 27 рисунків, 2 додатків, 25 джерел.