Специфікація програмного забезпечення

В.1 ВСТУП

В.1.1 Огляд продукту

Розроблене програмне забезпечення є вебзастосунком для візуального управління задачами в командній або персональній роботі за допомогою канбан-дощок. Система надає можливість створення проєктів, задач, управління їхнім статусом, а також перегляду календаря й аналітики. У межах командного проєкту мною було реалізовано серверну частину застосунку на основі NestJS та структуру бази даних MongoDB. Рішення побудоване на сучасному технологічному стеку: Next.js (frontend), NestJS (backend), MongoDB (NoSQL база даних).

В.1.2 Мета

Метою створення ПЗ є реалізація зручного, масштабованого серверного застосунку, що забезпечує стабільну роботу всіх ключових функцій — автентифікації, збереження задач і проєктів, а також інтеракції з клієнтською частиною через REST API.

В.1.3 Межі

Система орієнтована на веб-використання через браузер. Мобільна версія, офлайн-режим або повна інтеграція зі сторонніми сервісами (окрім email-повідомлень) наразі не підтримуються. Вся обробка даних відбувається через бекенд-застосунок, розгорнутий із використанням NestJS і MongoDB.

В.1.4 Посилання

GitHub: <https://github.com/NureMossurDaria/2025_B_KKP_PZPI-22-2_Mossur_D_Y>

В.1.5 Означення та абревіатури

ПЗ – програмне забезпечення

API – інтерфейс прикладного програмування

JWT – JSON Web Token, токен авторизації

CRUD – створення, читання, оновлення, видалення

MongoDB – документоорієнтована NoSQL база даних

NestJS – backend-фреймворк для Node.js

DTO – об’єкти передачі даних

Kanban – метод візуального управління задачами

В.2 ЗАГАЛЬНИЙ ОПИС

В.2.1 Перспективи продукту

Серверна частина побудована з урахуванням масштабованості. Надалі можливо впровадження WebSocket-з’єднання для роботи в реальному часі, додавання ролей користувачів, розширення аналітичного модуля, а також створення мобільного клієнта.

В.2.2 Функції продукту

Модулі backend-системи забезпечують:

* створення, редагування, видалення задач, проєктів і колонок;
* зберігання даних у структурованій базі MongoDB;
* автентифікацію та авторизацію користувачів з використанням JWT;
* оновлення статусів задач (у т.ч. drag-and-drop);
* отримання аналітичної інформації про виконані задачі.

В.2.3 Характеристики користувачів

Цільова аудиторія: користувачі, які потребують зручного інструменту для управління задачами через браузер. Не вимагається спеціальна технічна підготовка.

В.2.4 Загальні обмеження

Система працює лише за наявності інтернет-з’єднання, підтримки JavaScript і сучасного браузера. Доступ до функцій здійснюється через API-запити по HTTPS.

В.2.5 Припущення й залежності

* клієнт розроблено з використанням NextJS;
* сервер розгорнуто з підтримкою NestJS;
* база MongoDB постійно доступна;
* дані передаються через HTTPS;
* використовується email-сервіс для верифікації.

В.3 КОНКРЕТНІ ВИМОГИ

В.3.1 Вимоги до зовнішніх інтерфейсів

В.3.1.1 Програмний інтерфейс

Клієнт надсилає запити до backend-системи за допомогою HTTP. REST API, реалізоване через контролери NestJS, обробляє запити CRUD до задач, проєктів, колонок, а також забезпечує авторизацію.

В.3.1.2 Комунікаційний протокол

Уся взаємодія з backend здійснюється по HTTPS. Дані передаються у форматі JSON.

В.3.1.3 Операції

* вхід/реєстрація користувача;
* створення та редагування задач;
* зміна статусу задач;
* обробка запитів до MongoDB;
* робота з аналітичними запитами.

В.3.1.4 Припущення

Очікується, що користувач працює з фронтендом, що коректно формує запити. База MongoDB забезпечує швидкий доступ до колекцій users, projects, tasks, columns.

В.3.2 Властивості програмного продукту

Програмна архітектура backend-системи побудована за принципом модульності. Кожен функціональний блок (auth, tasks, projects, analytics) реалізовано в окремому модулі NestJS. Вся логіка рознесена між контролерами, сервісами та DTO, що забезпечує зручність підтримки.

В.3.3 Атрибути програмного продукту

Надійність: дані перевіряються через DTO й валідацію class-validator; помилки обробляються глобальними фільтрами.

Безпека: доступ до маршрутів обмежено через Guard-и та JWT.

Супроводжуваність: код розбитий на модулі, легко розширюється.

Продуктивність: використовуються індекси MongoDB для швидкого доступу до даних.

Переносимість: серверна частина може бути розгорнута у будь-якому середовищі з підтримкою Node.js.

В.3.4 Вимоги до бази даних

Для зберігання даних використовується документоорієнтована база MongoDB. Реалізовано колекції: users, projects, columns, tasks, analytics. Кожна сутність має унікальний \_id, поля з прив’язками (projectId, userId) та автоматичні мітки createdAt, updatedAt. Структура бази підтримує вкладені зв’язки та забезпечує зручну модифікацію без зміни схеми.