Специфікація програмного забезпечення

**1. ВСТУП**

### 1.1 Огляд продукту

Даний вебзастосунок реалізує систему Kanban-дошки для управління задачами у форматі drag-and-drop. Система призначена для покращення особистої або командної продуктивності шляхом візуального відображення процесів проєктного менеджменту.

### 1.2 Мета

Метою розробки є створення зручного та безпечного інструменту для створення Kanban-дошок, управління задачами, відстеження їх статусу та збереження даних користувача.

### 1.3 Межі

Програмне забезпечення працює у веббраузері. Кожен користувач бачить лише власні дані після авторизації. Система не підтримує одночасне редагування задач декількома користувачами в режимі реального часу.

### 1.4 Посилання

GitHub: <https://github.com/NureNesmiianovaOlha/2025_B_KKP_PZPI-22-2_Nesmiianova_O_O>

### 1.5 Означення та абревіатури

* **API** – Application Programming Interface
* **JWT** – JSON Web Token
* **CRUD** – Create, Read, Update, Delete
* **UI** – User Interface
* **DB** – Database

## **2.ЗАГАЛЬНИЙ ОПИС**

### 2.1 Перспективи продукту

Застосунок створений як самостійна вебсистема з можливістю розгортання на будь-якому хостингу. Подальший розвиток передбачає підтримку командної роботи та коментарів.

### 2.2 Функції продукту

* авторизація користувача (через email/пароль);
* створення, редагування, видалення дошок;
* робота з колонками та задачами;
* перетягування задач між колонками;
* збереження даних у MongoDB.

### 2.3 Характеристики користувачів

Цільова аудиторія — індивідуальні користувачі або студенти, які ведуть власні проєкти. Технічні знання не є обов’язковими.

### 2.4 Загальні обмеження

* доступ лише після авторизації;
* працює лише за стабільного інтернет-з’єднання;
* підтримується сучасними браузерами.

### 2.5 Припущення й залежності

* сервер розгорнуто та налаштовано;
* база даних підключена;
* email користувача унікальний.

## **3. КОНКРЕТНІ ВИМОГИ**

### 3.1 Вимоги до зовнішніх інтерфейсів

#### 3.1.1 Інтерфейс користувача

UI розроблено з використанням TailwindCSS. Інтерфейс адаптивний, включає drag-and-drop взаємодію.

#### 3.1.2 Апаратний інтерфейс

Вимог до специфічного обладнання немає, достатньо комп’ютера або мобільного пристрою з браузером.

#### 3.1.3 Програмний інтерфейс

RESTful API реалізовано у Next.js App Router. Основні ендпоїнти:

* GET /api/boards;
* POST /api/boards;
* PATCH /api/tasks;
* DELETE /api/tasks.

#### 3.1.4 Комунікаційний протокол

HTTP/HTTPS, передача даних у форматі JSON.

#### 3.1.5 Обмеження пам'яті

Не більше 10MB на одного користувача (можна змінити залежно від хостингу).

#### 3.1.6 Операції

* створення об'єктів (дошка, колонка, задача);
* перетягування задач;
* видалення або редагування елементів.

#### 3.1.7 Функції продукту

* авторизація користувача (через email/пароль);
* створення, редагування, видалення дошок;
* робота з колонками та задачами;
* перетягування задач між колонками;
* збереження даних у MongoDB.

#### 3.1.8 Припущення й залежності

Застосунок залежить від NextAuth для автентифікації та MongoDB для зберігання даних.

### 3.2 Властивості програмного продукту

Розроблене програмне забезпечення має низку важливих властивостей, що забезпечують його ефективність, зручність використання та надійність. Надійність системи гарантується стабільною обробкою помилок: у разі виникнення проблеми API повертає чітке повідомлення з відповідним кодом помилки, що дозволяє користувачу або розробнику оперативно реагувати. Доступність застосунку забезпечується цілодобовим функціонуванням через вебінтерфейс, що дозволяє користувачам отримувати доступ до системи в будь-який момент із будь-якого пристрою, підключеного до інтернету.

Безпека реалізована шляхом автентифікації користувачів і захисту маршруту за допомогою сесій, що унеможливлює доступ до даних стороннім особам. Супроводжуваність досягається завдяки модульній структурі коду: логіка роботи API, база даних, авторизація та UI поділені на окремі частини, що спрощує їхню підтримку та оновлення. Переносимість системи забезпечується тим, що вона може бути розгорнута на будь-якому сервері, який підтримує Node.js і має підключення до MongoDB, що робить проєкт придатним для масштабування та перенесення в інші середовища.

Продуктивність системи полягає у швидкій роботі API — час відповіді на запити за нормального навантаження становить менше однієї секунди, що задовольняє вимоги більшості користувачів. Таким чином, застосунок відповідає сучасним стандартам розробки веборієнтованих систем, забезпечуючи стабільну, безпечну й ефективну роботу.

### Вимоги бази даних

* База: MongoDB;
* Колекції: boards, tasks, users;
* Зв'язок: один користувач – багато дошок – багато задач;
* Індексація по email користувача.

### Інші вимоги

* підтримка локалізації у майбутніх версіях;
* сумісність з мобільними пристроями;
* підготовка до CI/CD.