# ДОДАТОК Г

Специфікація програмного забезпечення

**Vision and Scope Document**

**for**

**“**СИСТЕМА КЕРУВАННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ, MERN, MONGODB, EXPRESS.JS, REACT, NODE.JS, JWT, MONGOOSE”

**Version 1.0**

**Prepared by**

**Орлеан Максим Едуардович**

**ХНУРЕ**

**11.05.2025**

**Зміст (Table of Contents)**

Історія ревізій

1. Вимоги до бізнесу

1.1 Передумови (Background)

1.2 Можливості для бізнесу (Business Opportunity)

1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)

1.4 Потреби клієнтів або ринку (Customer or Market Needs)

1.5 Бізнес-ризики (Business Risks)

2. Образ рішення

2.1 Судження про бачення проекту (Vision Statement)

2.2 Основний функціонал (Major Features)

2.3 Припущення та залежності (Assumptions and Dependencies)

3. Обсяг і обмеження

3.1 Обсяг першого випуску (Scope of Initial Release)

3.2 Обсяг подальших випусків (Scope of Subsequent Release)

3.3 Обмеження та виключення (Limitations and Exclusions)

4 Бізнес-контекст

4.1 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)

4.2 Робоче середовище (Operating Environment)

Історія ревізій

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Chages** | **Version** |
| Орлеан М. Е. | 11.05.2025 | Розробка документу | Version 1.0 |

1 ВИМОГИ ДО БІЗНЕСУ

**1.1 Передумови (Background)**  
У процесі розробки інструменту генерації REST API одним з важливих компонентів є клієнтська частина, яка забезпечує зручну взаємодію користувача з серверною логікою. Основна функція — надати графічний інтерфейс для завантаження JSON-схем, реєстрації, авторизації, запуску генерації коду API та перегляду результатів у зрозумілому форматі.  
Передумови для створення клієнтського інтерфейсу:

- Потреба в доступному веб-інтерфейсі для не технічних користувачів.

- Забезпечення взаємодії з бекендом через HTTP-запити.

- Спрощення процесу завантаження схем та отримання результату.

- Автоматичне відображення документації до згенерованого API.

1.2 Можливості для бізнесу (Business Opportunity)  
Інтерфейс користувача підвищує доступність системи та значно спрощує її використання:

1. Простота для стартапів — можливість без знань CLI завантажити JSON-схему й отримати API.
2. Можливість масштабування через хостинг (наприклад, Vercel).
3. Освітній потенціал — студенти бачать результат своєї структури бази у вигляді API.
4. Уніфікований інтерфейс для роботи з документацією та генерацією запитів.

1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)  
BO1. Розробка адаптивного інтерфейсу на HTML/CSS/JS:

- SC1.1: Інтерфейс працює у всіх основних браузерах.

- SC1.2: Всі форми валідні, взаємодія з сервером через fetch/Axios.

BO2. Інтеграція з бекендом:

- SC2.1: Можливість авторизації/реєстрації через UI.

- SC2.2: Підтримка завантаження JSON-схеми.

- SC2.3: Вивід результатів та можливість завантаження архіву.

BO3. Інтерфейс для перегляду документації:

- SC3.1: Інтеграція з Swagger/Faker UI через iframe.

**1.4 Потреби клієнтів або ринку (Customer or Market Needs)**

1. Простий інтерфейс без необхідності вивчати CLI або бекенд.
2. Можливість швидко переглянути та перевірити згенероване API.
3. Зрозумілий UX з поясненням кроків.
4. Мобільна адаптивність (на майбутнє).

1.5 Бізнес-ризики (Business Risks)

* Застарілий дизайн або низька UX-якість.
* Залежність від зовнішніх бібліотек (наприклад, Swagger UI).
* Ризик некоректної обробки помилок з боку бекенду.
* Недостатнє тестування клієнтської частини в різних браузерах.

**2 ОБРАЗ РІШЕННЯ**

2.1 Судження про бачення проекту (Vision Statement)  
Клієнтська частина має забезпечити користувачам простий шлях до отримання REST API з JSON-схеми без складних налаштувань. Інтерфейс має бути інтуїтивним, чистим, і легким у підтримці. Можливості будуть постійно розширюватися.

* 1. Основний функціонал (Major Features)

- Реєстрація та вхід користувача.

- Завантаження JSON-схеми через drag-and-drop або file input.

- Відображення результатів генерації та лінк для завантаження архіву.

- Вивід документації у вбудованому фреймі (Swagger/Faker UI).

- Адаптивний дизайн.

- Повідомлення про помилки/успіхи.

2.3 Припущення та залежності (Assumptions and Dependencies)

- HTML/CSS — як основа візуальної частини.

- JavaScript — для логіки та запитів.

- Інтеграція з бекендом через REST.

- Усі API відповідають стандарту JSON.

- Користувач має сучасний браузер.

**3 ОБСЯГ І ОБМЕЖЕННЯ**

* 1. Обсяг першого випуску (Scope of Initial Release)

- Статичний інтерфейс із формами для авторизації, завантаження схем і кнопкою запуску генерації.

- Інтеграція через fetch/Axios для API-запитів.

- Інтерфейс перегляду документації.

- Вивід статусу операцій та помилок.

3.2 Обсяг подальших випусків (Scope of Subsequent Release)

- Темна тема.

- Підтримка мобільних пристроїв.

- Завантаження декількох схем одночасно.

- Drag-and-drop UI для конфігурації параметрів.

- Локалізація інтерфейсу (multilang).

3.3 Обмеження та виключення (Limitations and Exclusions)

- UI не містить побудованих діаграм або візуалізацій бази.

- Не підтримується offline-режим.

- Немає автоматичного збереження сесії.

- Swagger UI інтегрований як iframe, без контролю над його вмістом.

**4 БІЗНЕС-КОНТЕКСТ**

* 1. Профілі зацікавлених сторін

- Індивідуальні розробники — очікують простоти й мінімуму кроків.

- Освітні заклади — демонструють студентам процес генерації API через зрозумілий інтерфейс.

- Команди — використовують як MVP-інструмент.

* 1. Робоче середовище

- Браузери: Chrome, Firefox, Edge, Safari (останні версії).

- Сумісність із бекендом через HTTPS.

- Хостинг — Vercel/Netlify.

- Підтримка CORS та обробка токенів авторизації через локальне сховище (localStorage).

Інтерфейс має стати зручним містком між технічною логікою серверної частини та кінцевим користувачем, який хоче просто й швидко отримати готове API без залучення спеціалістів з бекенду.