



## Специфікація Вимог до Онлайн Магазину PowerZone

### Загальні Вимоги

- У магазині продаються різні види товарів незалежно від обраної категорії (у нашому випадку товари для автономного освітлення). Інтерфейс повинен бути адаптивним (responsive), щоб забезпечити зручне користування на різних пристроях: комп'ютерах, планшетах та мобільних телефонах.

### Функціональні Вимоги

#### Основний функціонал включає:

Сортування за ціною:

- можливість сортувати товари від дешевших до дорожчих та навпаки.

Пошук за назвою:

- інтерактивний пошук товарів за ключовими словами у назві.

Нумерація сторінок:

- розбиття товарів на сторінки для кращого навігаційного досвіду.

Кошик для покупок та створення замовлення:

- додавання товарів у кошик, видалення товарів з кошика, оформлення замовлення через систему.

## Додатковий Функціонал

### Фільтрування за ціновим діапазоном або категорією:

- користувачі можуть вибирати товари відповідно до встановлених фільтрів.

### Огляди та сортування за рейтингом:

- відображення відгуків покупців та можливість сортувати товари за середнім рейтингом.

### CMS для адміністратора для перегляду замовлень:

- адмін-панель для перегляду деталей замовлень, їх управління, зміна статусів.

### Додавання нових товарів:

- можливість адміністраторам додавати нові позиції до магазину.

### Надсилання змін статусу замовлення покупцеві електронною поштою або SMS:

- автоматизовані сповіщення для інформування покупців про статус замовлення.

## Орієнтовний Терміни Реалізації

- Орієнтовний термін реалізації проєкту становить 2 місяці. Точна тривалість залежатиме від складу та ефективності команди, а також від можливих змін у вимогах під час розробки.

## Вибір Технологій та Їх Взаємодія

### Фронтенд

#### Технології:

- React.js, Redux Toolkit, SCSS, TypeScript для створення динамічного інтерфейсу з підтримкою адаптивного дизайну.

#### Взаємодія:

- Фронтенд буде взаємодіяти з бекендом через RESTful API.

## Бекенд

### Технології:

- Nest.js для Node.js, та Ruby.

### Комунікація:

- RabbitMQ буде використовуватися для передачі даних між мікросервісами.

### Бази даних:

- обирається між PostgreSQL і MySQL.

## База Даних (на даний час невідомо яка саме)

### Вид бази даних:

- реляційна (PostgreSQL/MySQL).

### Зберігання даних:

- інформація про товари, користувачів, замовлення, відгуки.

### Інтеграція:

- бекенд буде взаємодіяти з базою даних через RabbitMQ

- Адмін-панель (CMS)

### Реалізація:

- використання готових рішень, таких як Strapi, або створення кастомної панелі на [React.js/Angular](#).

### Функції:

- управління замовленнями, додавання та оновлення товарів, управління користувачами.

### Повідомлення:

#### Технології:

- інтеграція з сервісами, такими як Twilio для SMS та Nodemailer для відправки email-повідомлень.

#### Взаємодія:

- бекенд буде відповідати за відправку повідомлень про статус замовлень.

## Серверна Частина та Хостинг

### Хостинг:

- використання хмарних рішень,.

### Серверне середовище:

- Docker для контейнеризації та Kubernetes для оркестрації, щоб забезпечити масштабованість. (або інший стек, на даний час невідомо)



## Безпека

### Авторизація та аутентифікація:

- використання JWT для захисту даних користувачів.

### Шифрування даних:

- SSL/TLS для безпечного обміну даними.  
Також сайт повинен бути захищений від SQL-ін'єкцій та XSS-атак

## Мікросервісна Архітектура

Проект реалізується з використанням мікросервісної архітектури, що дозволяє масштабувати окремі компоненти та спростити управління.

