

# Вимоги до проєкту

## Частина 1. Аутентифікація та авторизація користувачів

### Вимоги до бекенду:

1. Реалізувати автентифікацію на основі JWT за допомогою Django REST Framework.
2. Кінцеві точки:
  1. Логін (повертає токени доступу та оновлення).
  2. Оновлення токена (отримання нового токена доступу).
  3. Отримання профілю користувача (лише зареєстрований користувач може переглядати свої дані).
  4. Скидання паролю електронною поштою (MailHog для локального тестування).

## Частина 2. Система управління бібліотекою (ролі адміністратора та користувача)

### Вимоги до бекенду:

1. Реалізувати API для управління книгами з...:
  1. Відображенням всіх книг.
  2. Можливістю отримувати детальну інформацію про книгу.
  3. Фільтрувати книги за:
    1. Жанром
    2. Діапазоном років (від-до)
    3. Пошук за назвою
  4. Функції адміністратора:
    1. Додавання/редагування/видалення книг
2. Користувацькі дозволи:
  1. Адміністратори можуть змінювати книги.
  2. Звичайні користувачі можуть лише переглядати книги.
3. Можлива додаткова функціональність (наприклад, рейтинги книг, рецензії) (необов'язково).

## Частина 3. Парсинг та експорт даних в Excel

### Вимоги до бекенду:

1. Реалізувати скрипт для парсингу даних про книги з веб-сайту (<https://books.toscrape.com/>).
2. Зберігати отримані книги в базі даних.
3. При повторних запитах:
  1. Оновлення існуючих записів.
  2. Додавати нові книги.
4. Надати кінцеву точку для завантаження книг у форматі Excel з можливістю фільтрації:
  1. Роком
  2. Автором
  3. Жанр
  4. Пошук за назвою

## Технології, що будуть використовуватися

### Бекенд:

1. Django REST Framework (DRF)
2. JWT-аутентифікація (SimpleJWT)
3. PostgreSQL
4. Requests + BeautifulSoup4 (для веб-скрепінгу)
5. Pandas + XLSXWriter (для експорту даних)
6. Swagger (документація по API)

### Інфраструктура та DevOps:

1. Docker (контейнеризація)
2. Docker Compose (налаштування мультиконтейнерів)
3. MailHog (тестування електронної пошти)

### Структура коду та кращі практики:

1. Використовуйте Generic Views + Mixins для оптимізації кінцевих точок DRF.
2. Впроваджуйте кастомні дозволи для контролю доступу.
3. Дотримуйтесь принципів проектування RESTful API.