**Лабораторная работа 0: Проектирование приложения**

1. **Концепция приложения и сценарии использования**

* Концепция:

Книжный интернет-магазин, позволяющий пользователям регистрироваться, просматривать каталог книг, добавлять их в корзину и оформлять заказы. Также должна быть возможность просматривать историю покупок и обработки заказов.

* Сценарии использования:

Регистрация и вход: Пользователь заходит на сайт. Если у него нет аккаунта, он регистрируется (вводит email, пароль, имя). Если аккаунт есть, он входит, вводя email и пароль.

Просмотр каталога книг: Пользователь переходит на страницу каталога. Ознакамливается с ассортиментом, видит названия, авторов, жанры. Выбирает.

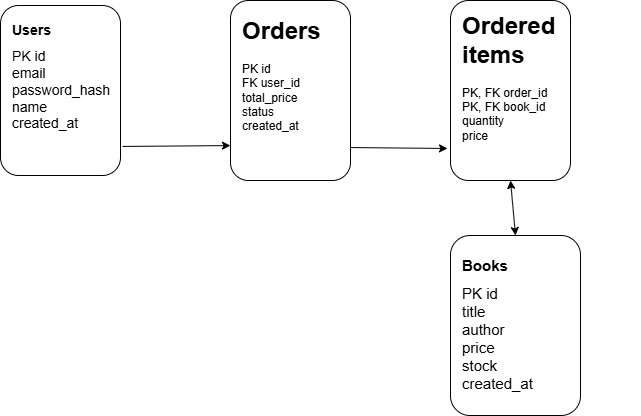
Добавление книги в корзину и покупка: Пользователь выбирает книги и нажимает «Добавить в корзину». Получает уведомление об успешной покупке.

Просмотр истории покупок: Пользователь заходит в личный кабинет. Видит список своих заказов (дата, книги, статус заказа).

Обработка ошибок: Некорректный пароль/email при входе. Ошибка при оформлении заказа (недоступность книги, проблемы с оплатой).

1. **Схема БД**

* Клиент (Frontend): веб-интерфейс, через который пользователи регистрируются, авторизуются, просматривают каталог книг и совершают покупки.
* Сервер (Backend): обрабатывает запросы от клиента, взаимодействует с базой данных, выполняет аутентификацию.
* База данных: хранит данные о пользователях, книгах и заказах.



1. **Схема API**

Аутентификация:

* POST /register

Запрос: { "email": "[user@example.com](mailto:user@example.com)", "password": "securepassword", "name": "User Name" }

Ответ: { "message": "User registered successfully" }

* POST /login

Запрос: { "email": "[user@example.com](mailto:user@example.com)", "password": "securepassword" } Ответ: { "token": "jwt\_token\_here" }

Работа с книгами

* GET /books — получить список книг

Ответ: [ { "id": 1, "title": "Book 1", "author": "Author A", "price": 19.99 }, { "id": 2, "title": "Book 2", "author": "Author B", "price": 24.99 } ]

* GET /books/{id} — получить информацию о конкретной книге

Ответ: { "id": 1, "title": "Book 1", "author": "Author A", "price": 19.99, "description": "Book description here" }

Работа с заказами

* POST /orders — создать заказ

Запрос: { "user\_id": 1, "items": [ { "book\_id": 1, "quantity": 2 }, { "book\_id": 3, "quantity": 1 } ] }

Ответ: { "order\_id": 123, "message": "Order created successfully" }

* GET /orders — получить историю заказов пользователя

Ответ: [ { "id": 123, "total\_price": 59.97, "status": "paid", "created\_at": "2025-02-15" } ]

1. **Список используемых технологий**

* Frontend (шаблоны страниц и стили):

Bootstrap — CSS-фреймворк для стилизации и адаптивной верстки

* Backend (архитектура и серверная логика)

Flask — микрофреймворк для разработки веб-приложений на Python

Jinja2 — встроенный в Flask шаблонизатор для рендеринга HTML

* База данных и управление данными

MySQL — реляционная база данных

MySQL Workbench — инструмент для проектирования и администрирования базы данных Регистрация и авторизация

OAuth — безопасная аутентификация через сторонние сервисы Flask-Login — управление сессиями пользователей в Flask