Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики, математики и электроники

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе №0

Дисциплина: «Технологии сетевого программирования»

Тема «Проектирование приложения»

Выполнили: Залевская В.Ю.,

Ващенков А. .

Группа: 6302-010302D

Самара, 2025

Задание на лабораторную работу.

1. Проектирование архитектуры:
   * Разработать схему взаимодействия компонентов (сервер, БД, клиент);
   * Разработать логическую схему базы данных (использование, как минимум, 3-х сущностей и 1-ой связи many-to-many);
   * Определить структуру API (REST, методы, URL, параметры, форматы запросов и ответов).
2. Создание Git-репозитория:
   * Инициализировать репозиторий (рекомендуется использовать GitHub);
   * Настроить базовую структуру проекта.
   * Добавить .gitignore для исключения ненужных файлов (виртуальное окружение, логи, артефакты сборки и т. д.).
3. Подготовка окружения:
   * Установить Java/Python;
   * Настроить виртуальное окружение и зависимости.
4. Оформление документации:
   * Подготовить отчет с результатами работы.
5. **Архитектура**

Клиент Сервер База данных

1. **Структура API**

#### Аутентификация и пользователи:

* **POST /auth/register** — Регистрация нового пользователя.
* **POST /auth/login** — Вход в систему.
* **POST /auth/refresh** — Обновление токена доступа.

#### Корзины (Basket):

* **POST /baskets** — Создать новую корзину.
* **GET /baskets/{id}** — Получить информацию о корзине по ID.
* **DELETE /baskets/{id}** — Удалить корзину.
* **GET /users/{userId}/baskets** — Получить все корзины пользователя.

#### Товары (Product):

* **POST /products** — Создать новый товар.
* **GET /products** — Получить список всех товаров (с фильтрами по категории: ?category={id}).
* **GET /products/{id}** — Получить информацию о товаре по ID.
* **PUT /products/{id}** — Обновить данные товара.
* **DELETE /products/{id}** — Удалить товар.

#### Категории товаров (Product\_category):

* **POST /categories** — Создать новую категорию.
* **GET /categories** — Получить список всех категорий.
* **GET /categories/{id}** — Получить информацию о категории по ID.
* **PUT /categories/{id}** — Обновить данные категории.
* **DELETE /categories/{id}** — Удалить категорию.

#### Управление товарами в корзине (buffer):

* **POST /baskets/{basketId}/items** — Добавить товар в корзину.
* **GET /baskets/{basketId}/items** — Получить список всех товаров в корзине.
* **PUT /baskets/{basketId}/items/{itemId}** — Изменить количество товара в корзине.
* **DELETE /baskets/{basketId}/items/{itemId}** — Удалить товар из корзины.
* **GET /baskets/{basketId}/total** — Рассчитать общую стоимость товаров в корзине.

1. **Схема взаимодействия**

* **Frontend**
  + Отображает интерфейс пользователя (каталог товаров, корзину, формы регистрации).
  + Отправляет HTTP-запросы к Backend API при действиях пользователя (например, добавление товара в корзину).
  + Получает данные от Backend в формате JSON и обновляет UI.
* **Backend**
  + Принимает запросы от Frontend.
  + Проверяет аутентификацию (JWT-токены, сессии).
  + Валидирует данные (например, наличие товара на складе).
  + Формирует SQL-запросы к Базе данных через ORM (Sequelize, Hibernate) или нативными методами.
  + Возвращает ответы в формате JSON (успех/ошибка).
* **База данных**
  + Хранит данные в таблицах: Basket, Product, Product\_category, buffer, User (если добавлена).
  + Выполняет CRUD-операции (Create, Read, Update, Delete) по запросам от Backend.

1. **Схема БД**

**Сущности**

1. **Корзина (Basket)**:  
   Хранит информацию о корзинах пользователей.  
   Поля:

* id (Primary Key)
* created (дата и время создания корзины)
* user (Foreign Key, ссылается на id в таблице User)

1. **Товар (Product)**:  
   Хранит данные о товарах.  
   Поля:

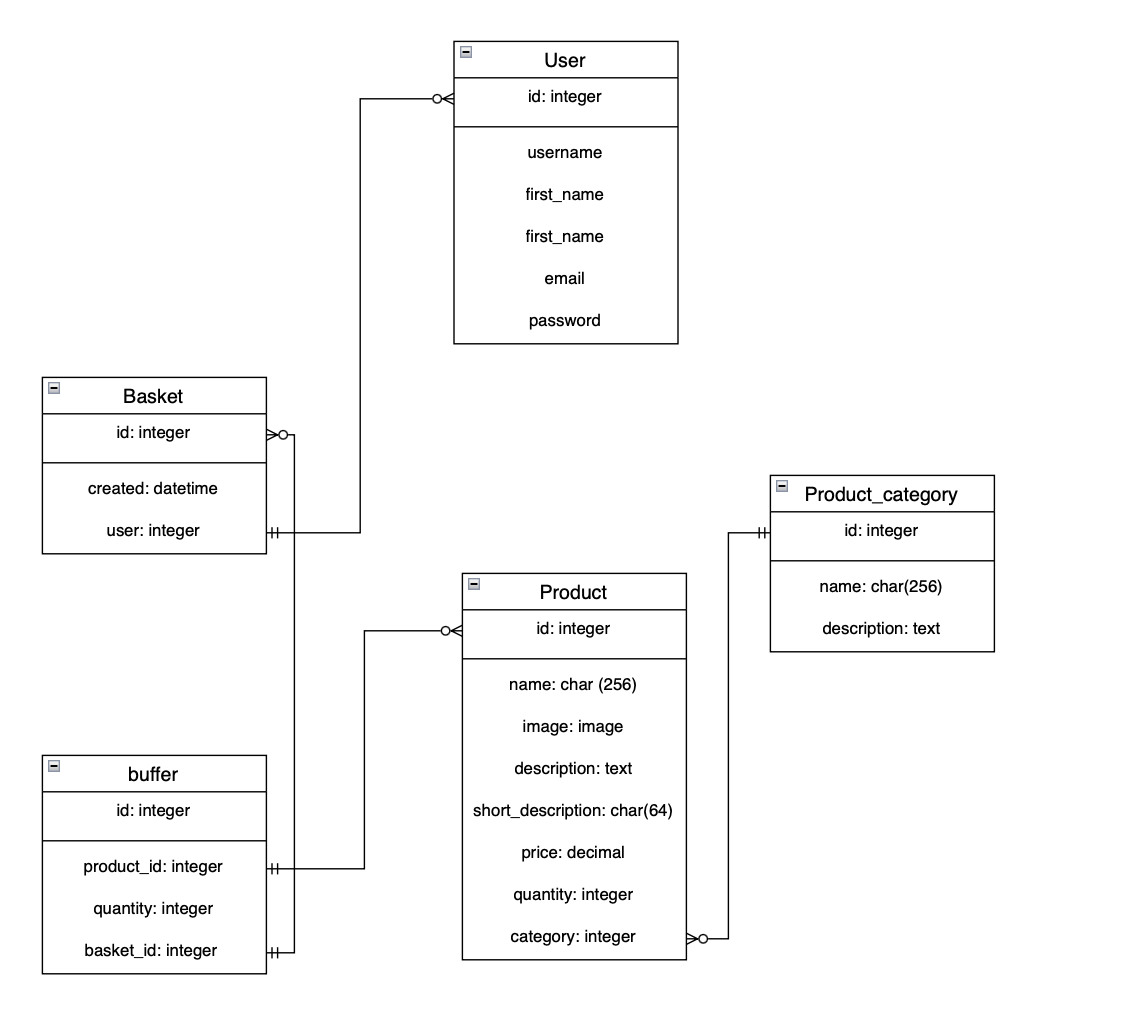
* id (Primary Key)
* name (название товара, до 256 символов)
* image (изображение товара)
* description (подробное описание товара, текст)
* short\_description (краткое описание, до 64 символов)
* price (цена товара, десятичное число)
* quantity (количество товара на складе)
* category (Foreign Key, ссылается на id в таблице Product\_category)

1. **Категория товара (Product\_category)**:  
   Хранит категории для группировки товаров.  
   Поля:

* id (Primary Key)
* name (название категории, до 256 символов)
* description (описание категории, текст)

1. **Буфер корзины (buffer)**:  
   Связывает корзины с товарами и хранит количество каждого товара в корзине.  
   Поля:

* id (Primary Key)
* product\_id (Foreign Key, ссылается на id в таблице Product)
* quantity (количество товара в корзине)
* basket\_id (Foreign Key, ссылается на id в таблице Basket)



1. **Стек технологий**

**1. Frontend (Клиентская часть)**

* Язык программирования: JavaScript.
* Фреймворк: React.js (или Vue.js, Angular)

**2. Backend (Серверная часть)**

* Язык программирования: Python
* Фреймворк: Django

**3. База данных**

* Реляционная СУБД: PostgreSQL

Стек технологий будет меняться и обновляться по мере выполнения проекта