

Отчет по разработке приложения Интернет-магазина

Разработка программного обеспечения

УДК 004.658:004.738.5

Интернет магазин “КК”

**Отчет по дисциплине “Основы проектирования баз данных”
направление 09.03.01 - Информатика и вычислительная техника**

**Работу выполнили студенты
группы 9/2-РПО-24/4**

**Контиев Владислав Алексеевич,
Казаков Александр Олегович**

**Руководитель:
Кузнецов Данила Алексеевич**

Пермь 2025

СОДЕРЖАНИЕ

1. ОБЩАЯ ИНФОРМАЦИЯ.....	3
2. СТЕК ТЕХНОЛОГИЙ.....	4
3. ОПИСАНИЕ БАЗЫ ДАННЫХ (SQLite).....	4
4. МОДЕЛИ ДАННЫХ.....	5
5. CRUD ОПЕРАЦИЙ.....	5
6. ОПИСАНИЕ API.....	7
7. БИЗНЕС–ЛОГИКА.....	7
8. МИГРАЦИЯ.....	8
9. ДОКУМЕНТАЦИЯ.....	8
10. ВЫВОД.....	8
11. БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	9

1. ОБЩАЯ ИНФОРМАЦИЯ

Название проекта: Интернет-магазин “КК”

Назначение: Онлайн-магазин с функционалом регистрации пользователей, корзины, заказов, промокодов и админки для управления категориями и товарами.

Основной функционал проекта:

- 1. Регистрация пользователей через Telegram ID.**
- 2. Добавление и удаление товаров в корзине.**
- 3. Создание заказа и перенос товаров из корзины в заказ.**
- 4. История заказов для пользователя.**
- 5. Применение промокодов.**
- 6. Админ-функции:**
 - Добавление категорий и карточек товара.
 - Обновление статусов заказов.
 - Просмотр промокодов.

2. СТЕК ТЕХНОЛОГИЙ

- **Python 3.13** — основной язык разработки.
FastAPI — веб-фреймворк для построения API, документация через **Swagger**.
- **SQLite** — реляционная база данных для хранения данных.
- **SQLAlchemy** — ORM для работы с базой данных, поддержка асинхронных сессий.
- **Pydantic** — схемы для валидации и сериализации данных.
- **AsyncIO** — асинхронная работа с базой данных и запросами.
- **Swagger UI** — автоматическая документация API (`/docs`).

3. ОПИСАНИЕ БАЗЫ ДАННЫХ (SQLite)

Основные таблицы:

1. user — пользователи

- **id — PK**
- **tg_id — Telegram ID, уникальный**
- **username — имя пользователя**
- **first_name — имя**
- **phone — телефон, уникальный**

2. categories — категории товаров

- **id, name_category**

3. card — товары

- **id, name_card, price, photo, category_id (FK)**

4. korzina — корзина пользователя

- **id, user_id, card_id**

5. promocode — промокоды

- **id, code, discount, is_active**

6. tochka_biz — точки выдачи

- **id, city, tochka_addr**

7. order — заказы

- **id, user_id, promocode_id, tochka_id, total_price**

8. order_info — позиции заказа

- **id, order_id, card_id, order_info_status**

9. review — отзывы на товары

- **id, card_id, rating, comment**

10.history_orders — история заказов

- **id, user_id, order_id**

4. МОДЕЛИ ДАННЫХ

Pydantic модели (для API и валидации):

- **UserCreate / UserResponse** — создание и отображение пользователя
- **CartItemAdd / CartItem / CartResponse** — корзина
- **OrderCreate / OrderItem / OrderResponse** — заказы
- **HistoryOrderItem / HistoryOrderResponse** — история заказов
- **PromocodeApply / PromocodeResponse** — промокоды
- **CategoryResponse / PickupPointResponse** — категории и точки выдачи
- **ReviewCreate** — отзывы
- **Enum OrderStatus** — статусы заказов (оформлен, выполнен, отказан)

SQLAlchemy модели:

Все таблицы из базы данных представлены классами с ORM

MappedColumn, включая связи через ForeignKey.

5. CRUD ОПЕРАЦИИ

Все операции вынесены в crud.py:

Пользователи

- **create_user(db, data)** — добавить нового пользователя
- **get_user_by_tg_id(db, tg_id)** — получить пользователя по Telegram ID

Корзина

- **add_to_cart(db, user_id, card_id)** — добавить товар
- **get_cart(db, user_id)** — получить корзину
- **remove_from_cart(db, user_id, card_id)** — удалить товар
- **clear_cart(db, user_id)** — очистить корзину

Заказы

- **create_order(db, user_id, promocode_id, tochka_id, total)** — создать заказ и перенести позиции из корзины

Админка

- **add_category(db, name)** — добавить категорию
- **add_card(db, name, price, photo, category_id)** — добавить карточку товара
- **update_order_status(db, order_info_id, status)** — изменить статус позиции заказа
- **get_promocode(db, code)** — получить активный промокод

6. ОПИСАНИЕ API

Документирование реализовано через FastAPI и Pydantic.

Примеры эндпойнтов:

- **POST /users/** — регистрация пользователя
- **GET /users/{tg_id}** — получить пользователя
- **POST /cart/** — добавить товар в корзину
- **GET /cart/{user_id}** — получить корзину
- **DELETE /cart/{user_id}/{card_id}** — удалить товар
- **DELETE /cart/{user_id}** — очистка корзины
- **POST /orders/** — создать заказ
- **POST /admin/category** — добавить категорию
- **POST /admin/card** — добавить товар
- **PUT /admin/order-status** — изменить статус заказа
- **GET /admin/promocode/{code}** — получить промокод

Все эндпойнты имеют **summary** и **description** для Swagger UI /docs

7. БИЗНЕС-ЛОГИКА

1. Регистрация пользователя: проверка уникальности **tg_id** и телефона.
2. Корзина: добавление, удаление, очистка товаров.

3. Создание заказа:

- Перенос всех товаров из корзины в `order_info` с `status=оформлен.`
- Расчет `total_price` (сумма товаров).
- Применение промокода (если активен).

4. Статусы заказа:

- По умолчанию оформлен.
- После выдачи или отказа → выполнен или отказан.

5. Админка: управление категориями, товарами, промокодами и статусами заказов.

6. Отзывы: пользователь может оценивать товары с рейтингом 1–5 и оставлять комментарий.

8. МИГРАЦИЯ

- Используется асинхронный SQLAlchemy с функцией `init_db()` для создания всех таблиц.
- Таблицы соответствуют структуре SQL из задания.
- В будущем можно интегрировать Alembic для версионирования схем.

9. ДОКУМЕНТАЦИЯ

- Все Pydantic модели имеют `description`.
- Все функции CRUD имеют `docstring` с описанием аргументов и результата.
- FastAPI автоматически генерирует документацию на `/docs`.

10. ВЫВОД

Проект реализует современный стек: FastAPI + AsyncSQLAlchemy + SQLite, с MVC-подходом:

- **Пользовательский функционал: регистрация, корзина, заказ, история заказов, промокоды.**
- **Админский функционал: добавление категорий, карточек товаров, управление статусами и промокодами.**
- **Полная документация все модели описаны.**
- **Асинхронная работа с базой данных для высокой производительности.**

10. БИБЛИОТИЧЕСКИЙ СПИСОК

- 1. Курсы IT “sudo teach IT”: Python Pro - полный курс - URL: <https://sudo.school/course/1> (дата обращения 01.03.2025)**

- 2. Курсы IT “sudo teach IT”: Простой ООП до PRO - URL: <https://sudoteach.com/course/oop> (дата обращения 25.11.2025)**

- 3. Курсы IT “sudo teach IT”: Основы SQL - URL: <https://sudo.school/course/5> (дата обращения 10.08.2025)**

- 4. Курсы IT “sudo teach IT”: Продвинутый SQL - URL: <https://sudo.school/course/6> (дата обращения 28.08.2025)**

- 5. Известная медиа платформа: Изучение SQLite3 за 30 минут! Практика на основе языка Python URL: <https://youtu.be/bmQPy89IZNk?si=n0uPqTjuSqnVDJbs> (дата обращения 19.09.2025)**

- 6. Курсы IT “sudo teach IT Python School”: Новостной портал на базе FastApi - URL: <https://sudoteach.com/course/fastapi> (дата обращения 20.12.2025)**

- 7. Курсы IT “Pytex”: Backend разработка на Python - URL: <https://pytex.school/courses/backend> (дата обращения 11.06.2025)**