

Презентация по проекту

Telegram-bot YLStore_Bot



Цель проекта

Цель проекта — создание интерактивного Telegram-бота для интернет-магазина, который автоматизирует обслуживание клиентов, повысит продажи и улучшит пользовательский опыт. Бот обеспечивает удобный доступ к каталогу товаров, информации об акциях, оформлению заказов и обратной связи.

Реализация позволяет снизить нагрузку на службу поддержки, ускорить оформление заказов и повысить уровень сервиса за счет персонализированного взаимодействия с клиентами в Telegram.

Основные цели и задачи проекта

Цели проекта

- ❖ Автоматизация продаж через Telegram
- ❖ Упрощение взаимодействия с ассортиментом
- ❖ Сбор и анализ данных о предпочтениях клиентов
- ❖ Интеграция с внешними сервисами модных трендов

Задачи проекта

- ❖ Интерактивное меню для навигации
- ❖ Добавление и просмотр товаров через бота
- ❖ Интеграция MoskAPI для трендов
- ❖ Управление клиентской базой
- ❖ Настройка локального сервера для вебхуков
- ❖ Обработка фотографий товаров

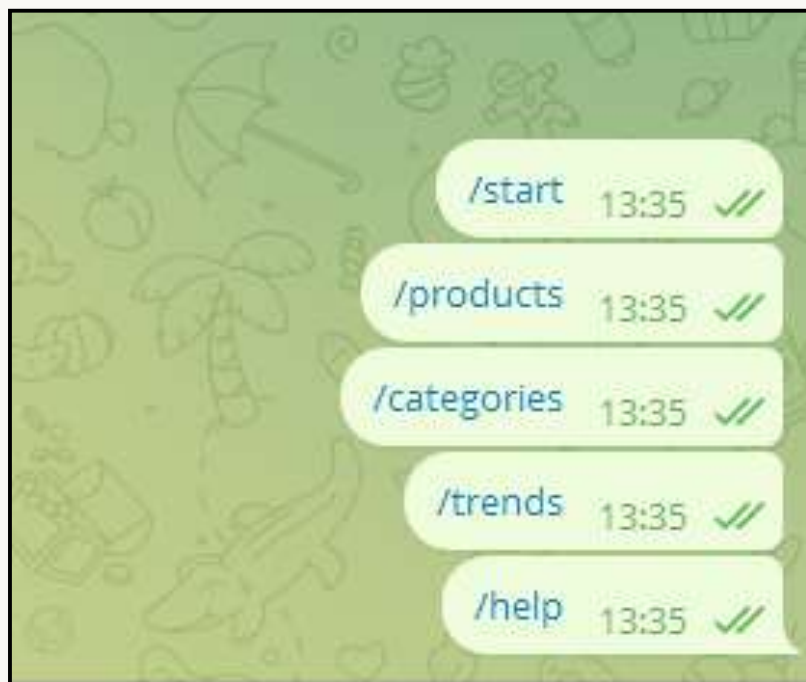
Функциональные возможности бота

Приветствие и меню

- ❖ Приветственное сообщение с кратким описанием возможностей и главное меню с командами /start, /trends, /products, /help.
- ❖ Интерактивное меню включает разделы: Товары, Тренды, Клиенты, О боте.

Навигация по каталогу

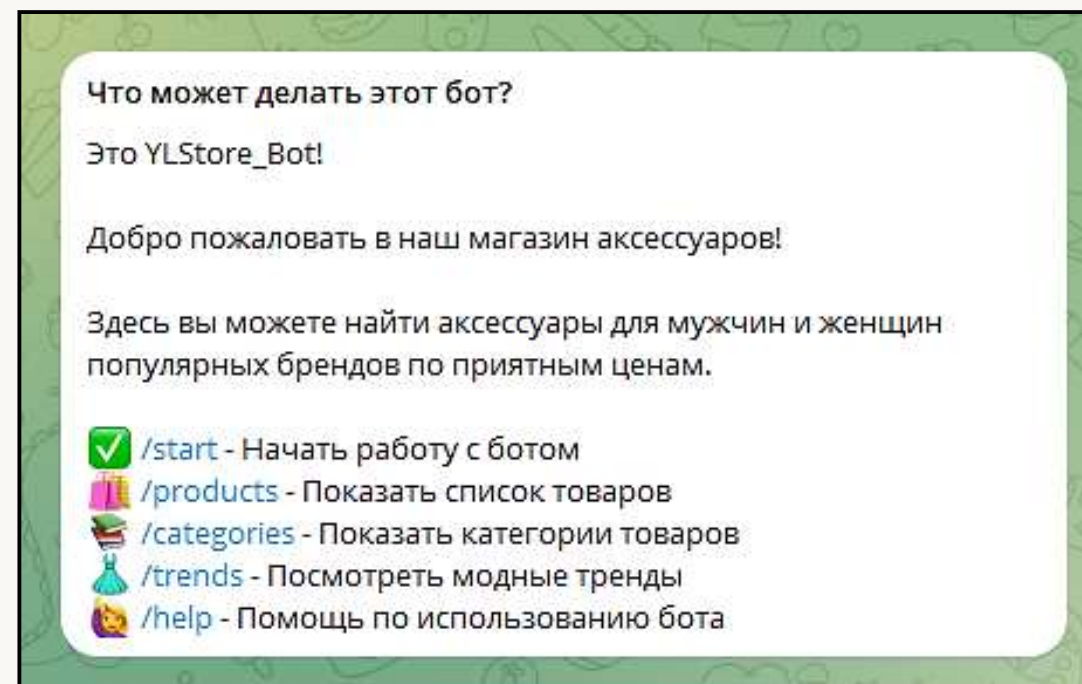
- ❖ Многоуровневое меню категорий и подкатегорий с кнопкой возврата в предыдущее меню.
- ❖ Просмотр товаров с пагинацией и фильтрацией по категориям через Inline-кнопки.



Команды



Список команд с описанием функционала



Приветственное сообщение бота

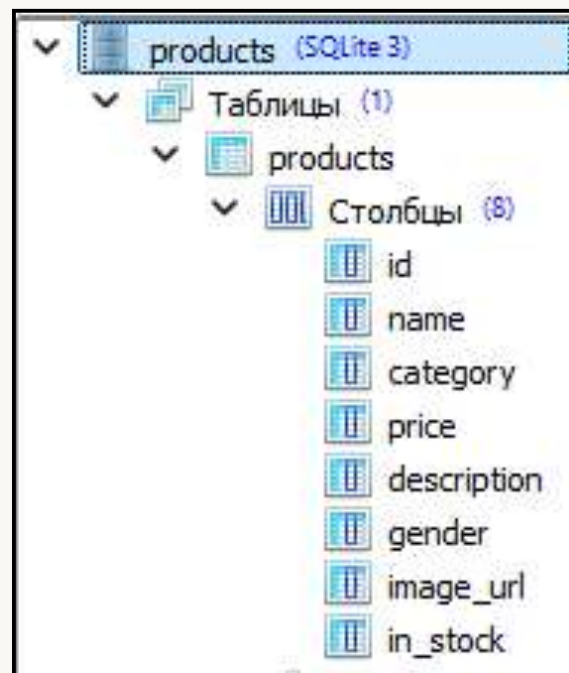
Работа с базой данных и товарами

Структура базы данных

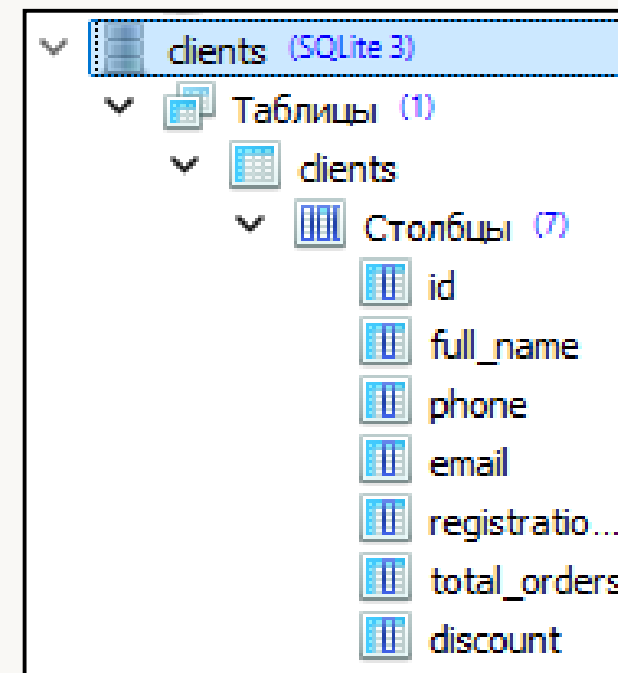
- ❖ Таблицы products и clients с полями для хранения информации о товарах и клиентах.

Управление товарами

- ❖ Добавление товаров через загрузку фото с автоматическим созданием записи в БД.
- ❖ Хранение изображений в папке uploads.



БД products.db
со списком товаров



БД clients.db
с данными о клиентах

Интеграции и API

Интеграция с MockAPI

- ❖ Получение данных о модных трендах с [MockAPI](#) и обработка ошибок подключения.

Локальный сервер и API

- ❖ Настройка вебхука <http://localhost:8080/webhook> и API-эндпоинтов для проверки работоспособности и получения списка товаров в JSON.

Технические требования и безопасность

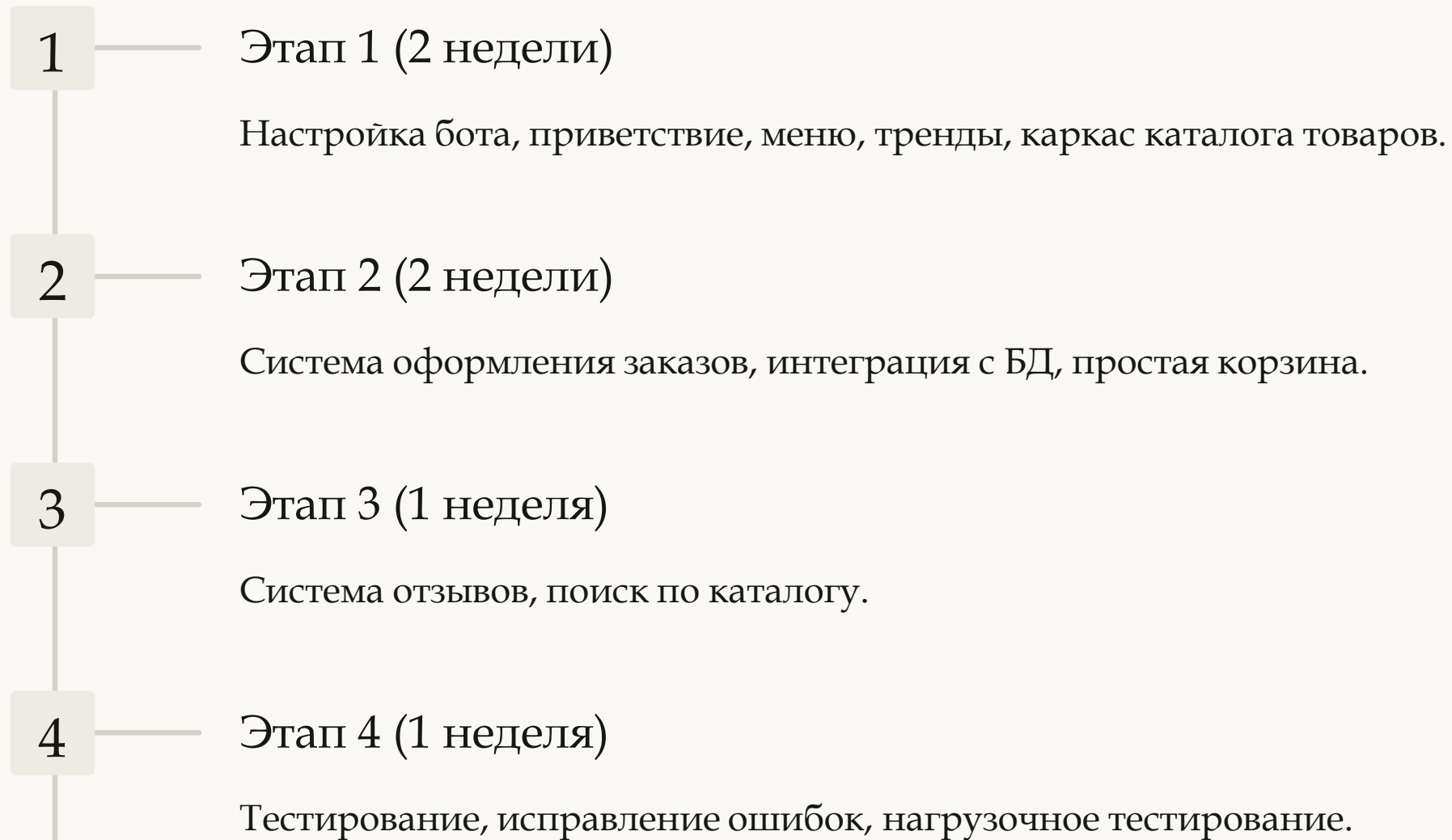
Платформа и библиотеки

- ❖ Используется Python 3.11+, библиотеки aiogram, aiohttp, SQLAlchemy, pydantic.
- ❖ Интерактивные кнопки, многоуровневое меню, поддержка медиафайлов.

Безопасность и конфигурация

- ❖ Запрет доступа к API без авторизации, валидация данных, хранение токенов и ключей в .env.
- ❖ Токен бота хранится в переменных окружения, используется python-dotenv для конфигурации.

Этапы разработки и критерии приемки



Критерии приемки включают работоспособность функций, удобный интерфейс, устойчивость к нагрузкам, логирование ошибок и документацию.

Спасибо за внимание!