**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на разработку Telegram-бота YLStore\_Bot**

**для магазина аксессуаров**

**Оглавление**

[Введение 3](#_Toc195650420)

[Функциональные требования 4](#_Toc195650421)

[Технические требования 6](#_Toc195650422)

[Этапы разработки 7](#_Toc195650423)

[Критерии приемки 8](#_Toc195650424)

[Дополнительные требования 8](#_Toc195650425)

[Сроки и бюджет 8](#_Toc195650426)

# **Введение**

Целью данного проекта является разработка интерактивного Telegram-бота YLStore\_Bot (https://t.me/YLStore\_Bot) для интернет-магазина, предназначенного для автоматизации обслуживания клиентов, повышения продаж и улучшения пользовательского опыта. Бот будет предоставлять покупателям удобный доступ к каталогу товаров, информации об акциях, возможности оформления заказов и обратной связи, а также другим функциям, способствующим увеличению вовлеченности и лояльности аудитории.

Реализация бота позволит магазину сократить нагрузку на службу поддержки, ускорить процесс оформления заказов и повысить уровень сервиса за счет персонализированного взаимодействия с клиентами в Telegram.

Проект предусматривает следующие **цели**:

* Автоматизация продаж через Telegram.
* Упрощение процесса взаимодействия клиентов с ассортиментом магазина.
* Сбор и анализ данных о предпочтениях клиентов.
* Интеграция с внешними сервисами для получения актуальных модных трендов.

Также реализуются следующие **задачи:**

* Реализовать интерактивное меню для навигации по функционалу бота.
* Обеспечить возможность добавления и просмотра товаров через бота.
* Интегрировать MockAPI для отображения модных трендов.
* Реализовать простое управление клиентской базой (добавление/просмотр).
* Настроить локальный сервер для обработки вебхуков и API-запросов.
* Обеспечить обработку фотографий товаров с сохранением в файловую систему.

# **Функциональные требования**

**Основные функции**

**1. Приветствие пользователя:**

* Краткое описание возможностей бота.
* Главное меню с кнопками:

*Основные команды:*

* **/start** – приветственное сообщение с главным меню.
* **/trends** – вывод актуальных трендов из MockAPI.
* **/products** – список товаров с пагинацией.
* **/help** – помощь в работе с ботом.

*Интерактивное меню:*

* **Товары** – просмотр каталога.
* **Тренды** – актуальные модные тенденции.
* **Клиенты** – управление клиентской базой.
* **О боте** – информация о функционале.

**2. Навигация по каталогу:**

* Многоуровневое меню категорий товаров.
* Возможность просмотра подкатегорий.
* Кнопка возврата в предыдущее меню.

*База данных (SQLite + SQLAlchemy):*

Таблицы:

* products (товары): id, name, price, category, description, image\_path, size, color.
* clients (клиенты): id, full\_name, phone, email, address.

**3. Работа с товарами:**

* Добавление товара через загрузку фото (автоматически создается запись в БД).
* Просмотр товаров с фильтрацией по категориям (реализуется через Inline-кнопки).
* Хранение изображений в папке uploads.

**5. Интеграции и API:**

*Интеграция с MockAPI:*

* Запрос к https://6824dee00f0188d7e72b3020.mockapi.io/fashion-trends для получения данных о трендах.
* Обработка ошибок подключения.

*Локальный сервер (aiohttp):*

* Вебхук: http://localhost:8080/webhook.

*API-эндпоинты:*

* GET / — проверка работоспособности (health\_check).
* GET /api/products — список товаров в JSON.

**6. Дополнительные функции, которые могут быть реализованы в дальнейшем:**

* Поиск товаров по ключевым словам.
* Корзина для сбора товаров перед заказом.
* История заказов для авторизованных пользователей.
* Система лояльности (бонусные баллы).
* Push-уведомления о статусе заказа.

# **Технические требования**

**1. Платформа:**

* Язык: Python 3.11+.
* Библиотеки:
  + aiogram (Telegram Bot Framework).
  + aiohttp (HTTP-клиент и сервер).
  + SQLAlchemy (ORM для работы с БД).
  + pydantic (валидация данных, опционально).

**2. Интерфейс:**

* Интерактивные кнопки в чате.
* Многоуровневое меню.
* Поддержка медиафайлов (фото товаров).

**3. Безопасность:**

* Запрет доступа к API без авторизации (если требуется).
* Валидация входящих данных (на стороне сервера).
* Хранение чувствительных данных (токены, API-ключи) в .env.

**4. Конфигурация:**

* Токен бота хранится в переменных окружения (BOT\_TOKEN).
* API-ключи вынесены в .env (используется python-dotenv).

**5. Логирование:**

* Запись ошибок и событий в консоль (logging.INFO).

# **Этапы разработки**

Этап 1: базовый функционал (2 недели)

* Настройка бота.
* Система приветствия и главное меню.
* Раздел с трендами.
* Каркас каталога товаров.

Этап 2: работа с заказами (2 недели)

* Система оформления заказов.
* Интеграция с базой данных.
* Простая корзина товаров.

Этап 3: дополнительные функции (1 неделя)

* Система отзывов.
* Поиск по каталогу.

Этап 4: тестирование и доработки (1 неделя)

* Юзабилити-тестирование.
* Исправление ошибок.
* Нагрузочное тестирование.

**Требования к тестированию:**

* Юнит-тесты для API-эндпоинтов (pytest).
* Ручное тестирование команд бота.

# **Критерии приемки**

1. Все основные функции работают без ошибок.

2. Интерфейс интуитивно понятен пользователям.

3. Бот выдерживает нагрузку 50+ одновременных запросов.

4. Реализована система логирования ошибок.

5. Написана документация для администратора.

# **Дополнительные требования**

1. Возможность масштабирования функционала.

2. Поддержка мультиязычности (опционально).

3. Интеграция с платежными системами (опционально).

4. API для подключения к CRM системе (опционально).

# **Сроки и бюджет**

Срок разработки: 6 недель.

Техническое сопровождение: 1 месяц после запуска.

Бюджет: определяется после уточнения всех требований.