

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку сервиса для анализа цен и ассортимента на маркетплейсах РФ

# 1. Общие сведения

**1.1. Наименование проекта:** Parser\_telegram\_bot — сервис парсинга данных о товарах, анализа цен и ассортимента в Telegram. **1.2. Контекст:** Пользователям сложно и трудозатратно самостоятельно проанализировать текущие товарные предложения на разных маркетплейсах. Данный сервис автоматизирует сбор данных о товарах, анализ цен и итоговые релевантные предложения через удобный Telegram-интерфейс. **1.3. Используемые технологии (стек):**

- **Backend (Парсинг & API):** Python, FastAPI, библиотеки для парсинга (BeautifulSoup4, Scrapy, Selenium/Playwright - по необходимости).
- **Работа с данными и задачи по расписанию:** PostgreSQL (данные о пользователях, хранение запросов и итоговые выданные результаты в абсолютах (цена, рейтинг, бренды)).
- **Внешний интерфейс:** Telegram Bot (библиотека python-telegram-bot).
- **Инфраструктура:** Docker, Docker Compose.

# 2. Цели и задачи проекта

## 2.1. Цели

1. Создать автоматизированный сервис для анализа цен на товары с выбранных маркетплейсов. Сервис находит релевантное товарное предложение с самой низкой ценой или лучшим соотношением "цена/рейтинг", информирует о сроках доставки, сравнивает цены, что дает полную картину для выбора.

## 2.2. Задачи

1. Разработать модули парсеров для целевых маркетплейсов (Wildberries, OZON).
2. Спроектировать и реализовать базу данных для хранения информации о авторизованных пользователях, истории запросов и итоговых данных в абсолютах (средняя цена, рейтинг, доставка).
3. Обеспечить базовое логирование и обработку ошибок.

# 3. Требования к системе

## 3.1. Функциональные требования (ФТ)

1. **ФТ-01. Базовый запрос на поиск и анализ товара:**
  - Пользователь выбирает один или все предложенные площадки по продаже товаров для анализа
  - Пользователь вводит наименование товара или категорию товара для поиска по множеству брендов в рамках категории
2. **ФТ-02. Парсинг данных:**

- Система должна извлекать из страницы товара: наименование, текущую цену, старую цену (если есть), рейтинг, наличие, срок доставки, кол-во отзывов.

#### 3. **ФТ-03. Анализ:**

- Система выдает результирующим ответом анализ по выбранных площадках по запрашиваемому товару в разрезе: распределение цены, топ-10 брендов, если запрос на уровне категории, кол-во предложений
- Система отправляет пользователю визуализацию анализа (matplotlib, seaborn)
- Система выдает топ-5 лучших предложений с лучшим соотношением "цена/рейтинг" и кол-во отзывов

#### 4. **ФТ-04. Администрирование:**

- (Опционально)** Существует панель или команды для администратора для просмотра состояния системы, логов, количества пользователей.

## 3.2. Нефункциональные требования (НФТ)

- НФТ-01. Производительность:** Парсинг одного товара не должен занимать более 1 минуты.
- НФТ-02. Надежность:** Система должна корректно обрабатывать ошибки парсинга (изменение структуры сайта, недоступность) и не прекращать работу целиком.
- НФТ-03. Безопасность парсинга:** Парсер должен использовать задержки между запросами и корректные HTTP-заголовки (User-Agent), чтобы минимизировать риск блокировки IP.
- НФТ-04. Масштабируемость:** Архитектура должна позволять добавлять парсеры для новых маркетплейсов без переписывания основной логики.

## 4. Описание системы (Архитектура)

Система состоит из следующих основных модулей:

### 4.1. Модуль парсинга (Parser Core)

- Отвечает за сбор данных с маркетплейсов.
- Реализуется как набор отдельных "спайдеров" (по одному на каждый маркетплейс).
- Система выполняет запрос к странице, извлекает структурированные данные и сохраняет в БД с привязкой к пользователю.

### 4.2. База данных (PostgreSQL)

Хранит:

- Пользователи (users):** ID Telegram, имя, дата регистрации.
- Товары (products):** Название, ссылка, текущая цена, дата запроса.

### 4.3. Модуль Telegram-бота (Bot Core)

- Взаимодействует с Telegram API.

- Обрабатывает команды (/start, /help, /list) и текстовые сообщения (ссылки).
- При получении данных от модуля анализа отправляет пользователям уведомления.

## 5. План работ (Roadmap)

### Фаза 1: Прототип (Недели 1-2)

1. Настройка окружения (Python, виртуальное окружение, Docker).
2. Разработка простейшего парсера для одного маркетплейса (BeautifulSoup/Requests).
3. Создание схемы БД и ее реализация.
4. Написание базовой логики для сохранения данных.

### Фаза 2: Ядро системы (Недели 3-4)

1. Разработка базового Telegram-бота с командой /start и добавлением возможности множественного выбора площадок для парсинга и анализа.
2. Привязка пользователей и товаров в БД.

### Фаза 3: Логика и анализ (Недели 5-6)

1. Реализация логики анализа полученных данных
2. Реализация логики выдачи итогов анализа
3. Добавление парсера для второго маркетплейса.
4. Улучшение обработки ошибок и логирования.

### Фаза 4: Доработка и развертывание (Недели 7-8)

1. Создание docker-compose.yml для запуска всех сервисов.
2. Написание базовой документации в README.md.
3. Тестирование и отладка.

## 6. Критерии приемки

Проект считается успешно завершенным, если:

1. Пользователь может через Telegram-бота получить релевантны пул товаров и базовый анализ на выбранном маркетплейсе(-ах) по своему запросу.
2. Система сохраняет запрос товар и его текущую цену в базу данных.
3. Пользователь может запросить список своих запросов товаров.
4. Система может работать непрерывно в фоновом режиме (запущенная через docker-compose up).