

# DemoProject

## 1. Функциональные требования

### Общая идея:

Создание веб-сервиса, который позволяет пользователям подключать внешние API, отслеживать события и настраивать уведомления.

---

### Основные модули:

#### 1.1 Аутентификация и авторизация

- Регистрация/логин
- JWT-токены / сессии
- Роли и права доступа

#### 1.2 Источники данных

- CRUD API для добавления источников
- Шифрование токенов
- Проверка связи

#### 1.3 Подписки и триггеры

- CRUD для триггеров
- Условия (напр., температура > 25)
- Расписание cron-похожее

#### 1.4 Уведомления

- Email через SMTP
- Telegram Bot API
- Webhook
- Лог отправки

#### 1.5 Дашборд

- Список триггеров, уведомлений
  - CRUD для подписок
- 

## 2. Нефункциональные требования

- PostgreSQL

- Docker + docker-compose
- Логирование (файлы/база)
- .env-конфигурация
- Swagger/OpenAPI для документации
- Pytest + coverage  $\geq 60\%$
- CI/CD (GitHub Actions/или GitLab CI)
- Black / flake8

## **Этап 1: Проектирование и подготовка окружения**

- Проработка архитектуры и ER-диаграммы
- Настройка репозитория, CI/CD пайплайна
- Docker + docker-compose базовая конфигурация
- Инициализация проекта Django и/или FastAPI

## **Этап 2: Базовая авторизация и CRUD**

- Аутентификация (JWT / сессии)
- Пользователи и роли
- CRUD для источников данных

## **Этап 3: Интеграция внешних API**

- Выбор 1–2 внешних API (например, OpenWeather, GitHub)
- Асинхронные запросы (для FastAPI)
- Хранение и обновление данных из внешних API

## **Этап 4: Настройка триггеров и подписок**

- Интерфейс настройки условий (температура, коммиты, и пр.)
- Расписание через APScheduler / Celery
- Тестирование логики триггеров

## **Этап 5: Отправка уведомлений**

- Email через SMTP
- Telegram через Bot API
- Webhook
- Логирование событий и уведомлений

## **Этап 6: Расширение и оптимизация**

- Кеширование (Redis)
- Очереди задач (Celery)
- Покрывание тестами (pytest)
- Добавление Swagger-документации

## **Этап 8: Завершение и защита**

- Подготовка документации

- Демонстрация API и функционала