

## ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ ПРОЕКТА: ECOORBIT AI

**Наименование продукта:** Интеллектуальная система спутникового мониторинга лесных ресурсов.

**Версия:** 1.0.0 (MVP)

**Язык разработки:** Python 3.10+, HTML5/CSS3, JavaScript (ES6).

### 1. Описание архитектуры (System Architecture)

Система построена на базе микрофреймворка **Flask** и состоит из трех независимых уровней:

**Слой данных:** Интеграция с API спутников Sentinel-2 (ESA).

**Аналитический слой (Core):** Скрипты на языке Python, использующие библиотеки NumPy и OpenCV для расчета вегетационного индекса NDVI и семантической сегментации зон горения.

**Интерфейсный слой:** Веб-приложение с использованием библиотеки Leaflet.js для интерактивной визуализации и Telegram-API для оперативного оповещения.

### 2. Спецификация оборудования и ПО

Для корректной работы системы требуются следующие зависимости:

Библиотека	Назначение
Flask	Организация веб-сервера и API-запросов
Requests	Передача данных в Telegram-бот
OpenCV	Предварительная обработка спутниковых снимков
Leaflet.js	Отрисовка интерактивных карт в браузере

### 3. Инструкция по установке (Deployment Guide)

#### Шаг 1: Подготовка окружения

1. Убедитесь, что у вас установлен Python. Выполните команду в терминале для установки библиотек:
2. Bash
3. `pip install flask requests opencv-python numpy`

#### Шаг 2: Настройка Telegram-бота

1. Получите токен у @BotFather.
2. Узнайте свой Chat ID через @userinfobot.
3. Вставьте полученные данные в файл app.py в соответствующие переменные TOKEN и CHAT\_ID.

#### Шаг 3: Запуск сервера

1. Запустите главный файл приложения:
2. Bash
3. `python app.py`
4. После запуска откройте браузер по адресу: `http://127.0.0.1:5000`.

#### 4. Руководство пользователя (User Manual)

1. **Загрузка данных:** На главной панели нажмите кнопку "Загрузить спутниковый снимок". Выберите файл формата .jpg или .png.
2. **Запуск ИИ:** Нажмите кнопку "Запустить анализ ИИ".
3. **Визуализация:** Система автоматически переместит карту в зону обнаруженного инцидента и подсветит границы пожара.
4. **Оповещение:** В течение 3-5 секунд на привязанное устройство в Telegram придет уведомление с координатами и расчетной площадью ущерба.

#### 5. Методология разработки (SDLC)

Проект реализован в соответствии с классическим циклом разработки ПО:

**Анализ:** Определение экологических проблем региона.

**Дизайн:** Проектирование архитектуры взаимодействия Python и Web.

**Кодинг:** Написание модулей обработки снимков и ботов.

**Тестирование:** Валидация точности ИИ на архивных снимках лесного фонда РК.

#### 6. Безопасность и масштабируемость

**Безопасность:** Доступ к управлению системой защищен уникальным токеном API.

**Масштабируемость:** Архитектура позволяет добавлять новые модули (например, детектирование паводков или незаконных карьеров) без изменения ядра системы.