**АВТОНОМАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ**

**«АКАДЕМИЯ ТОП»**

**ДОКУМЕНТАЦИЯ ИНДИВИДУАЛЬНОГО ПРОЕКТА**

**На тему** “Веб-приложение для управления проектами и задачами с использованием Python и Django”

Выполнили:

студент гр. РПО 9/23/2

Сухинин А.В

Проверил:

Алхимов В.А

г. Санкт-Петербург

2025

Оглавление

[**I. Введение** 3](#_Toc197950701)

[**II. Функциональные требования** 4](#_Toc197950702)

[**III. Нефункциональные требования** 5](#_Toc197950703)

[**IV. Архитектура и структура проекта** 6](#_Toc197950704)

[**V. Установка и запуск проекта** 7](#_Toc197950705)

[**VI. Использование** 8](#_Toc197950706)

[**VII. Заключение** 9](#_Toc197950707)

# **I. Введение**

Цель данного проекта – разработать веб-приложение для управления проектами и задачами, которое позволит пользователям эффективно организовать рабочий процесс. Приложение предназначено для индивидуального проекта и командного использования: пользователи могут создавать проекты, разбивать их на задачи, назначать ответственных исполнителей, устанавливать сроки и отслеживать статус выполнения.

Проект реализован с использованием языка программирования Python и фрейм-ворка Django, что обеспечивает быструю разработку, гибкость и масштабируемость. В качестве архитектурного подхода используется MVC, а для взаимодействия с внешними интерфейсами реализован REST API на базе Django REST Fraemework.

Приложение может быть использовано как внутри команды разработчиков, так и в других сферах, где необходима организация проектов: образование, управление персоналом, планирование мероприятий и другие.

# **II. Функциональные требования**

1. Управление пользователями

Регистрация и авторизация пользователей.

Разграничение прав доступа

Профиль пользователя

2. Управление проектами

Создание нового проекта с название, описание и сроком выполнения.

Редактирование и удаление проектов.

Просмотр списка всех проектов.

3. Управление задачами

Добавление задач в проекте

Указание описания, приоритета, статуса, дедлайна.

Назначение задачи конкретному пользователю

Изменение и удаление задач.

4. Комментарии и взаимодействие

Возможность комментировать задачи

Уведомлять о новых комментариях или изменениях в задаче

5. Фильтрация и поиск

Поиск задач по названию, статусу, исполнителю, сроку.

Фильтрация задач по проекту.

6. REST API

Доступ к функциям через API (для интеграции с другими сервисами).

Защита API через аутентификацию (например, JWT или сессии).

7. Административная панель

Возможность администратору управлять пользователями, проектами и задачами через Django Admin.

# **III. Нефункциональные требования**

Они определяют характеристики системы, не связанные напрямую с её функциями, но влияющие на качество, удобство и производительность. Для данного проекта устанавливаются следующие нефункциональные требования:

1. Удобство использования

Интерфейс должен быть интуитивно понятным.

Доступ к основным функциям — в 2–3 клика.

Поддержка тёмной темы

2. Надёжность и стабильность

Система должна обрабатывать ошибки

Приложение не должно падать при высокой нагрузке или вводе некорректной информации.

3. Безопасность

Защита от несанкционированного доступа

Хэширование паролей

Валидация данных на серверной стороне.

4. Масштабируемость

Возможность расширения функциональности

Поддержка подключения других баз данных

5. Совместимость

Поддержка современных браузеров

Работа на всех основных операционных системах.

6. Производительность

Загрузка страниц не более чем за 2 секунды при стандартной нагрузке.

Оптимизированные SQL-запросы и кеширование.

# **IV. Архитектура и структура проекта**

1. Models - описывают структуру данных (пользователи, проекты, задачи).
2. Views - логика обработки запросов и взаимодействие с данными.
3. Templates - HTML-страницы, которые видит пользователь.
4. Serializers - преобразуют данные между
5. Python - объектами и JSON

**Структура проекта**:



# **V. Установка и запуск проекта**

Клонируйте репозиторий:  
git clone https://github.com/valtureso788/web-application  
Перейдите в папку проекта:  
cd project-manager  
Установите зависимости:  
pip install -r requirements.txt  
Выполните миграции:  
python [manage.py](https://manage.py/) migrate  
Создайте суперпользователя:  
python [manage.py](https://manage.py/) createsuperuser  
Запустите сервер:  
python [manage.py](https://manage.py/) runserver

# **VI. Использование**

 После запуска — перейти на <http://127.0.0.1:8000/>  
 Зарегистрироваться или войти  
 Создать проект, затем задачи  
 Назначать исполнителей, менять статусы, следить за дедлайнами

# **VII. Заключение**

В рамках данного проекта было разработано веб-приложение для управления проектами и задачами с использованием Python и Django. Система позволяет удобно создавать проекты, распределять задачи между участниками, устанавливать сроки и отслеживать статус выполнения. Это делает приложение полезным инструментом для командной работы и личной организации.

Приложение выполнено с упором на простоту и удобств, что позволяет адаптировать его под разные цели и команды.