**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Информационные технологии и программирование»

**Отчет по проектному практикуму**

на тему:

Kanban desk: управление задачами

Выполнил:

студент группы БВТ2302

Корепанов А.

Проверил:

Мокин А.

Москва

2025

Содержание

Введение..........................................3

Ключевые функциональные блоки.....................4

Технологии и инструменты..........................5

Реализация........................................6

Система безопасности и доступ.....................7

Развертывание и использование.....................8

Заключение........................................9

# Введение

Данный проектный практикум посвящен разработке системы управления задачами в формате Kanban. Методология Kanban активно применяется в современных компаниях для оптимизации процессов разработки программного обеспечения, а также в других сферах, где необходимо отслеживать выполнение большого количества задач в рамках различных проектов. Система позволяет наглядно представлять рабочий процесс, контролировать загрузку сотрудников и своевременно реагировать на изменения. В рамках данного проекта реализована минимально жизнеспособная версия такой системы с возможностью регистрации пользователей, создания проектов, распределения задач по колонкам и их отслеживания во времени.

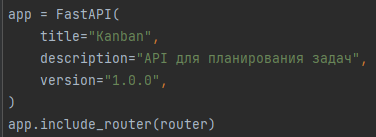


Рисунок 1 - запуск FastAPI и подключение роутеров

# Ключевые функциональные блоки

Проект содержит несколько ключевых блоков, каждый из которых отвечает за выполнение конкретной функции. Первым блоком выступает управление проектами, где пользователи могут создавать новые проекты, редактировать их описание и удалять при необходимости. Следующий блок — это система регистрации и аутентификации, где реализованы механизмы защиты данных пользователей и разграничение прав доступа. Колонки представляют собой этапы в рамках Kanban-доски, по которым перемещаются задачи в процессе выполнения. Каждый проект имеет набор колонок, и каждая задача принадлежит конкретной колонке. Сами задачи содержат информацию о заголовке, описании, сроке выполнения и порядке отображения. Также реализован механизм логирования действий, что позволяет отслеживать изменения, происходящие с задачами, например, изменение названия, описания, даты или перемещение между колонками.

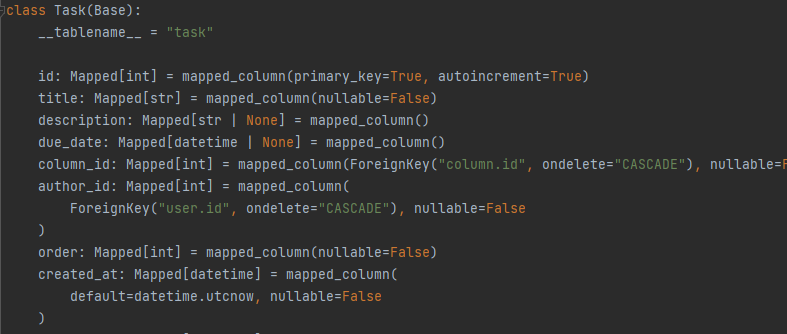


Рисунок - модели данных

# Технологии и инструменты

Для реализации проекта была выбрана современная стек технологий. В качестве основы серверной части использован FastAPI — высокопроизводительный асинхронный фреймворк на языке Python. Он позволяет быстро создавать RESTful API с поддержкой OpenAPI документации. Для работы с базой данных используется SQLAlchemy, обеспечивающий ORM-взаимодействие с PostgreSQL. База данных PostgreSQL выбрана как одна из наиболее надежных и производительных реляционных СУБД. Валидация и сериализация данных осуществляется с помощью Pydantic. Аутентификация пользователей реализована на основе JWT токенов, а конфигурация параметров осуществляется через .env файл с помощью библиотеки pydantic-settings. Также проект легко контейнеризуется при помощи Docker.

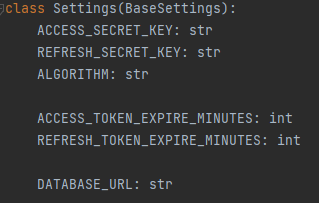


Рисунок 3 - использование Pydantic Settings и .env-файлов

# Реализация

Структура проекта разделена на модули по функциональному назначению. Модели данных описывают такие сущности, как пользователь, проект, задача, колонка и лог действий. Для каждой модели предусмотрены соответствующие схемы сериализации и валидации данных. Реализовано множество эндпоинтов, позволяющих выполнять CRUD операции над данными, включая регистрацию и вход пользователей, создание проектов и колонок, добавление задач и отслеживание логов изменений. Каждое действие пользователя логируется, что дает возможность в дальнейшем анализировать активность и производительность. Работа с базой данных осуществляется асинхронно, что повышает отклик сервиса и масштабируемость.

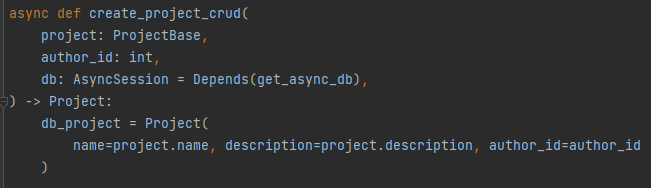


Рисунок 4 - взаимодействие с базой и использование логирования

# Система безопасности и доступ

Особое внимание в проекте уделено безопасности пользовательских данных и контролю доступа. Все пользователи должны пройти процедуру регистрации, после чего получают доступ к функциям системы через механизм JWT токенов. Access и Refresh токены хранятся в cookie с флагами безопасности, исключающими возможность XSS-атак. Доступ к задачам и проектам ограничен в зависимости от участия пользователя в соответствующих проектах. Проверка токена выполняется для каждого защищенного маршрута, и только авторизованные пользователи могут выполнять действия в рамках своих прав.

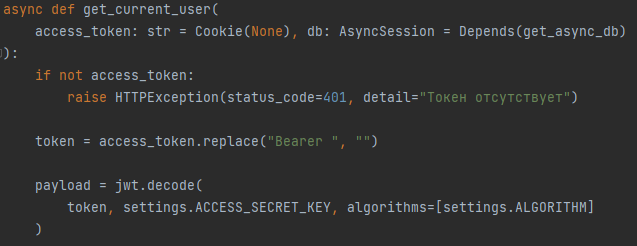


Рисунок 5 - проверка JWT токена из cookie и защиту эндпоинтов

# Развертывание и использование

Проект легко разворачивается как в локальной среде, так и в продакшн-среде при помощи Docker. Все зависимости указаны в requirements.txt, а переменные среды настраиваются через .env файл. FastAPI предоставляет документацию по всем эндпоинтам на пути /docs. После запуска пользователи могут регистрироваться, создавать проекты, добавлять задачи, перемещать их между колонками, отслеживать изменения и контролировать сроки исполнения.

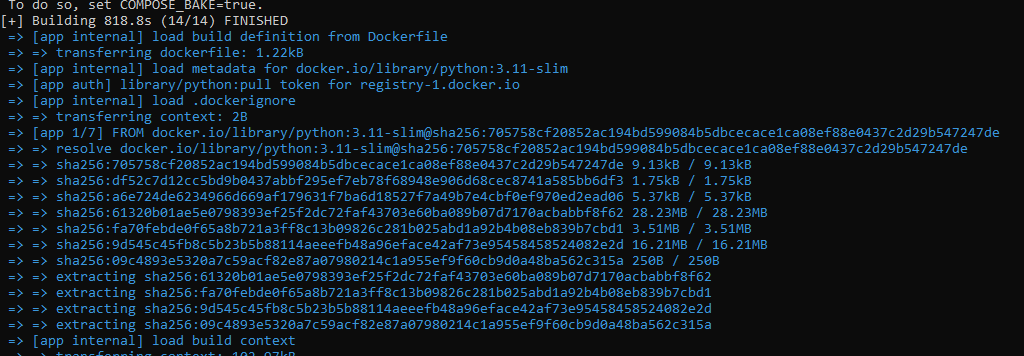


Рисунок - практическая часть развертывания

# Заключение

Разработанная система управления задачами в формате Kanban позволяет эффективно организовать рабочий процесс, улучшить взаимодействие внутри команды и повысить прозрачность выполнения задач. Проект может служить основой для дальнейшего расширения, включая внедрение уведомлений, аналитики и интеграции с другими сервисами. Полученные в ходе практикума знания и навыки нашли свое практическое применение в создании функционального и масштабируемого решения.

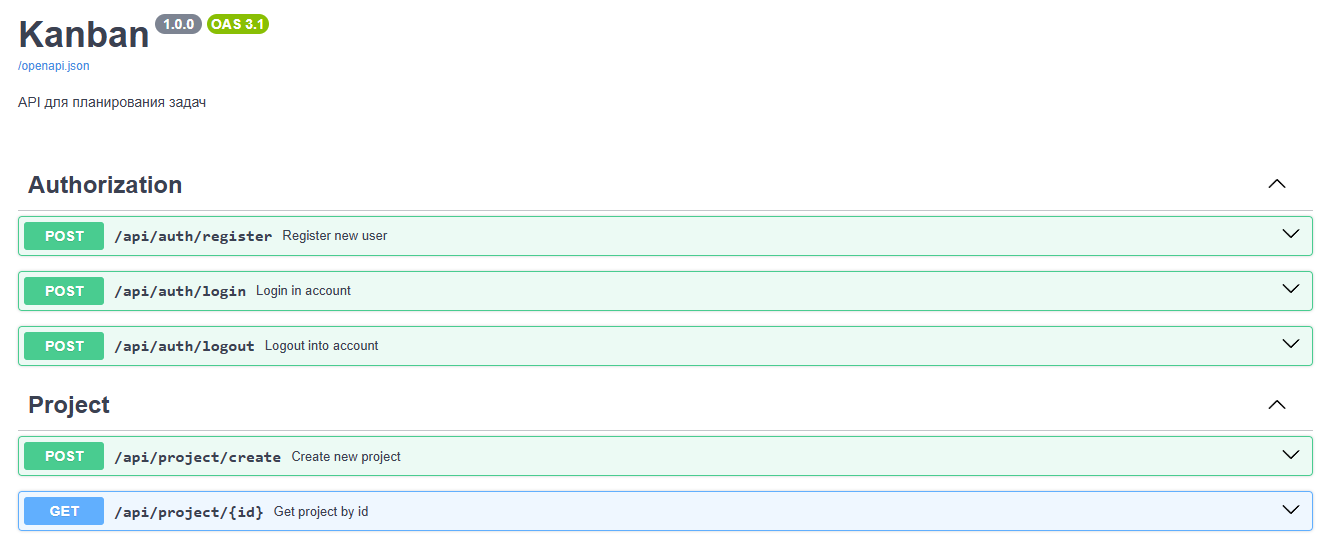


Рисунок - удобство взаимодействия с API