Министерство науки и высшего образования РФ  
ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСИС»

Институт компьютерных наук (ИКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**Отчет по лабораторной работе №7**

по дисциплине «Разработка сетевых приложений

на языке программирования Python»

на тему «Создание простого REST API на Flask»

Выполнил:  
студент группы БИСТ-22-1

Медведев Д. Р.

Проверил:   
доц. каф. ИКТ

Стучилин В.В.

Москва, 2025

**Цель работы**: приобрести навыки создания веб-приложений на Python с использованием микрофреймворка Flask.

Научиться:

обрабатывать HTTP-запросы;

формировать ответы в формате JSON;

организовывать структуру проекта;

понимать разницу между хранением данных в памяти и в базах данных

**Задание:** Разработать простое REST API для управления списком задач. Задачи должны храниться в

оперативной памяти и иметь следующую структуру:

"id": 1,

"title": "Сделать лабораторную работу"

Требования к API:

1. Реализовать маршруты:

GET /tasks — вернуть список всех задач

POST /tasks — добавить новую задачу (тело запроса:

GET /tasks/<id> — получить задачу по ID

DELETE /tasks/<id> — удалить задачу

2. Реализовать автоматическую генерацию

3. Обрабатывать ошибки:

404 Not Found — если задача не найдена по ID

400 Bad Request — если

title не передан или пустой

{ "title": "..." } )

id (инкрементный счётчик).

4. Организовать структуру проекта по папкам и модулям (

app.py ,

views.py ,

data.py ).

5. Протестировать работу API с помощью Postman или curl:

curl -X POST -H "Content-Type: application/json" -d '{"title": "Buy milk"}'

<http://localhost:5000/tasks>

Решение

Структура проекта:

App.py (Главный файл приложения)

Data.py (Работа с данными)

Views.py (HTTP-обработчики)

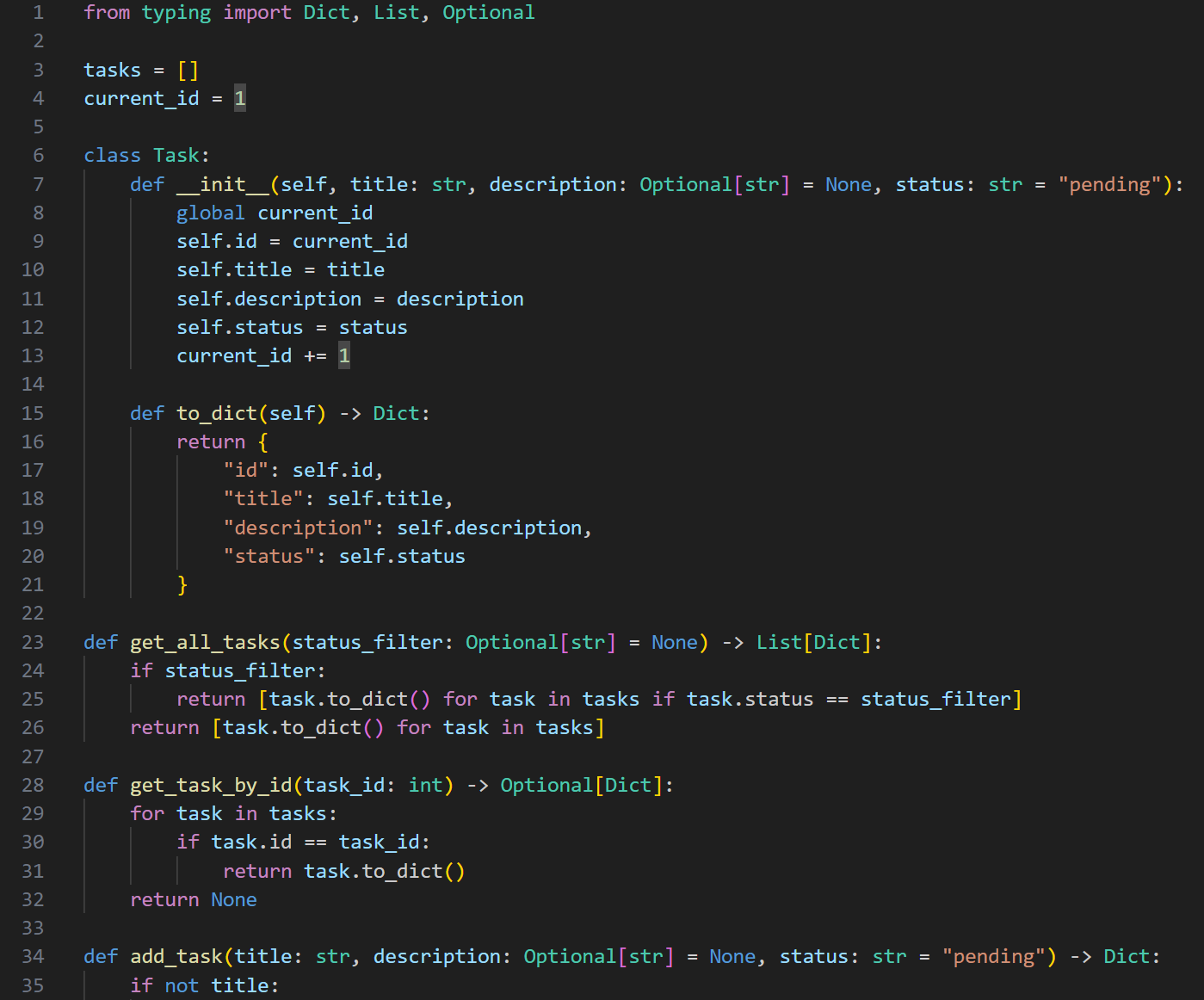
Test\_api.py (Тесты)

App.py

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Data.py



Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Views.py

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Test\_api.py

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

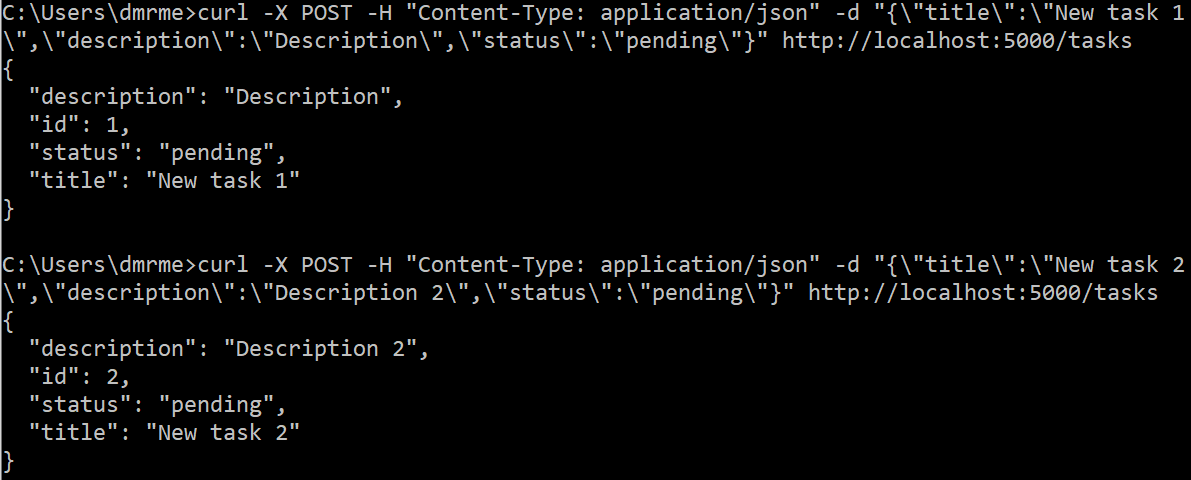


Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Результат

Добавление задач



Получение всех задач

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Получение по id

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Удаление

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Дополнительные задания:**

Изменение задачи по id

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Фильтрация по статусу

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Статистика по статусу

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

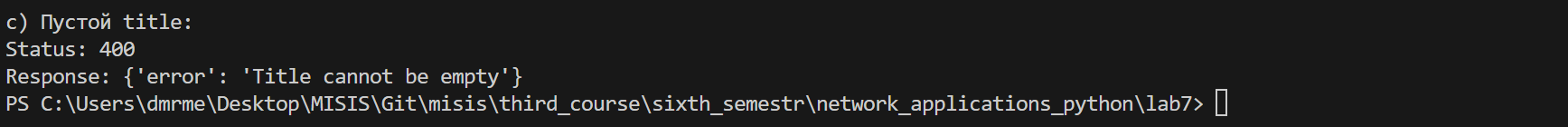
Python скрипт тестирования:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



Вывод: я приобрел навыки создания веб-приложений на Python с использованием микрофреймворка Flask.