**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

# МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

***Факультет информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки 09.03.02 «Информационные системы и технологии»**

# ЛАБОРАТОРНАЯ РАБОТА № 8

**Дисциплина: Backend-разработка**

**Тема: FastAPI**

**Выполнил(а):**

студент(ка) группы: 221-379

Сеидов М.М.

(Фамилия И.О.)

**Проверил:**

(Фамилия И.О., степень, звание)

**Москва2024 Листинг:**

from dataclasses import dataclass from fastapi import FastAPI, Request, Form, Cookie from fastapi.responses import JSONResponse, FileResponse, RedirectResponse from pydantic import BaseModel from typing import Optional

app = FastAPI()

# Определите маршрут / с помощью декоратора @app.get("/"), который будет возвращать простое сообщение "Hello, World!".

@app.get("/") def read\_root():

return "Hello, World!"

# Создайте маршрут /greet/{name}, который принимает параметр пути name и возвращает приветствие "Hello, {name}!". @app.get("/greet/{name}") def greet\_name(name: str): return f"Hello, {name}!"

# ??? Создайте маршрут /search, который принимает параметр строки запроса query и возвращает сообщение "You searched for: {query}".

@app.get("/search") def search\_query(query: Optional[str] = None): if query:

return f"You searched for: {query}" return "No query"

# Создайте маршрут /json, который возвращает JSON-ответ с данными о вас (имя, возраст, хобби).

@app.get("/json") def return\_json():

return JSONResponse({"Имя": "Тагир", "Возраст": 20, "Хобби": ["Спорт", "Чтение"]})

# Создайте маршрут /file, который отправляет текстовый файл с произвольным содержимым.

@app.get("/file") def send\_file(): with open("textfile.txt", "w") as f:

f.write("Текстовый файл") return FileResponse("textfile.txt", media\_type="text/plain", filename="textfile.txt")

# Создайте маршрут /redirect, который выполняет перенаправление на маршрут /.

@app.get("/redirect") def redirect\_to\_root():

return RedirectResponse(url="/")

# Создайте маршрут /headers, который возвращает все заголовки запроса в виде JSON.

@app.get("/headers") def return\_headers(request: Request):

return JSONResponse(content=dict(request.headers))

# Создайте маршрут /set-cookie, который устанавливает куку с именем username и значением your\_name.

@app.get("/set-cookie") def set\_cookie():

response = JSONResponse(content="Cookie set") response.set\_cookie(key="username", value="your\_name") return response

# Создайте маршрут /get-cookie, который возвращает значение куки username.

@app.get("/get-cookie") def get\_cookie(username: Optional[str] = Cookie(None)):

if username:

return f"Cookie value: {username}" return "No cookie found"

# Создайте маршрут /login, который принимает данные формы с полями username и password и возвращает сообщение "Welcome, {username}!".

@app.post("/login") def login(username: str = Form(...), password: str = Form(...)):

return f"Welcome, {username}!"

# Создайте маршрут /register, который принимает JSON-данные с полями username, email и password и возвращает сообщение "User {username} registered successfully!".

@app.post("/register") def register(user: dict):

username = user.get("username") return f"User {username} registered successfully!"

# Создайте класс User с полями id, username, email и password.

@dataclass class User:

id: int username: str email: str password: str

users = [

User(id=1, username="kate", email="kate@gmail.com", password="1234"), User(id=2, username="alex", email="alex@gmail.com", password="5678"),

]

# Создайте маршрут /users, который возвращает список объектов класса User в формате JSON.

@app.get("/users") def get\_users(): return users

# Создайте маршрут /users/{id}, который возвращает объект класса User с указанным id.

@app.get("/users/{id}") def get\_user\_by\_id(id: int): for user in users: if user.id == id: return user

return "User not found"