

Міністерство освіти і науки України

Національний технічний університет України „КПІ імені

Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки Кафедра
інформатики і програмної інженерії

ЗВІТ

лабораторної роботи № 6

з курсу «Основи WEB - технологій»

Тема: «Створення власного API на Python (FastAPI).»

Перевірів:

Викл. Альбрехт Й.О.

Виконав:

Студент ІП-15 Мешков
А.І

Київ 2024

Завдання

Створити web-додаток з використанням технології FastAPI, який буде працювати з інформацією відповідно вашому варіанту.

Варіант 7

Варіант 7. Спроекувати базу даних про оцінки, отриманих студентами на іспитах: прізвище, група, предмет, номер квитка, оцінка, викладач.

Хід роботи

Було програму python з використанням fastapi, pydantic, typing.

main.py

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from typing import List

app = FastAPI()

class Grade(BaseModel):
    id: int
    lastName: str
    group: str
    subject: str
    ticketNumber: int
    grade: int
    professor: str

grades = []

@app.get("/grades", response_model=List[Grade])
async def get_all_grades():
    return grades

@app.get("/grades/{grade_id}", response_model=Grade)
async def get_grade(grade_id: int):
    for grade in grades:
        if grade.id == grade_id:
            return grade
    raise HTTPException(status_code=404, detail="Grade not found")

@app.post("/grades", response_model=Grade)
async def create_grade(grade: Grade):
    for existing_grade in grades:
        if existing_grade.id == grade.id:
            raise HTTPException(status_code=400, detail="Grade with this ID already exists")
    grades.append(grade)
    return grade

@app.put("/grades/{grade_id}", response_model=Grade)
async def update_grade(grade_id: int, updated_grade: Grade):
    for i, grade in enumerate(grades):
        if grade.id == grade_id:
            grades[i] = updated_grade
            return updated_grade
    raise HTTPException(status_code=404, detail="Grade not found")
```

```
@app.delete("/grades/{grade_id}")
async def delete_grade(grade_id: int):
    for i, grade in enumerate(grades):
        if grade.id == grade_id:
            grades.pop(i)
            return {"message": f"Grade with ID {grade_id} deleted successfully"}
    raise HTTPException(status_code=404, detail="Grade not found")
```

1. Отримані результати

На рис 6.1-6.6 можна побачити запити.

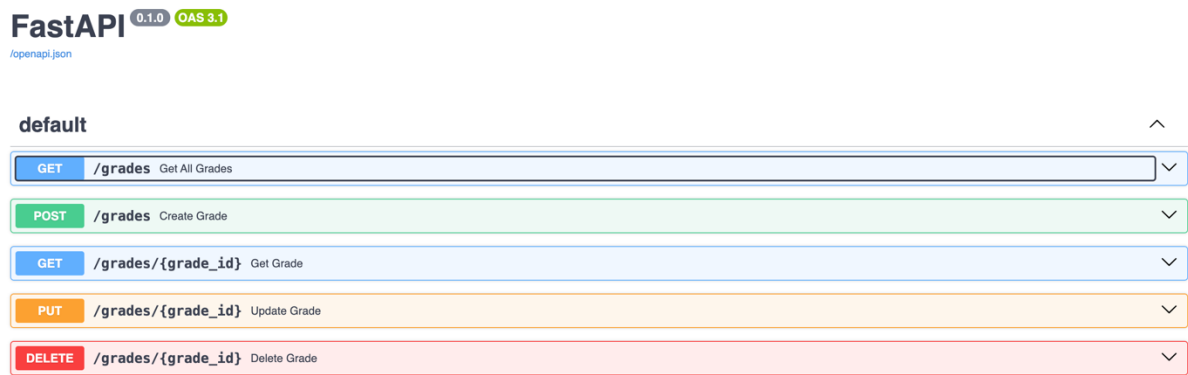


Рис. 6.1. Запити

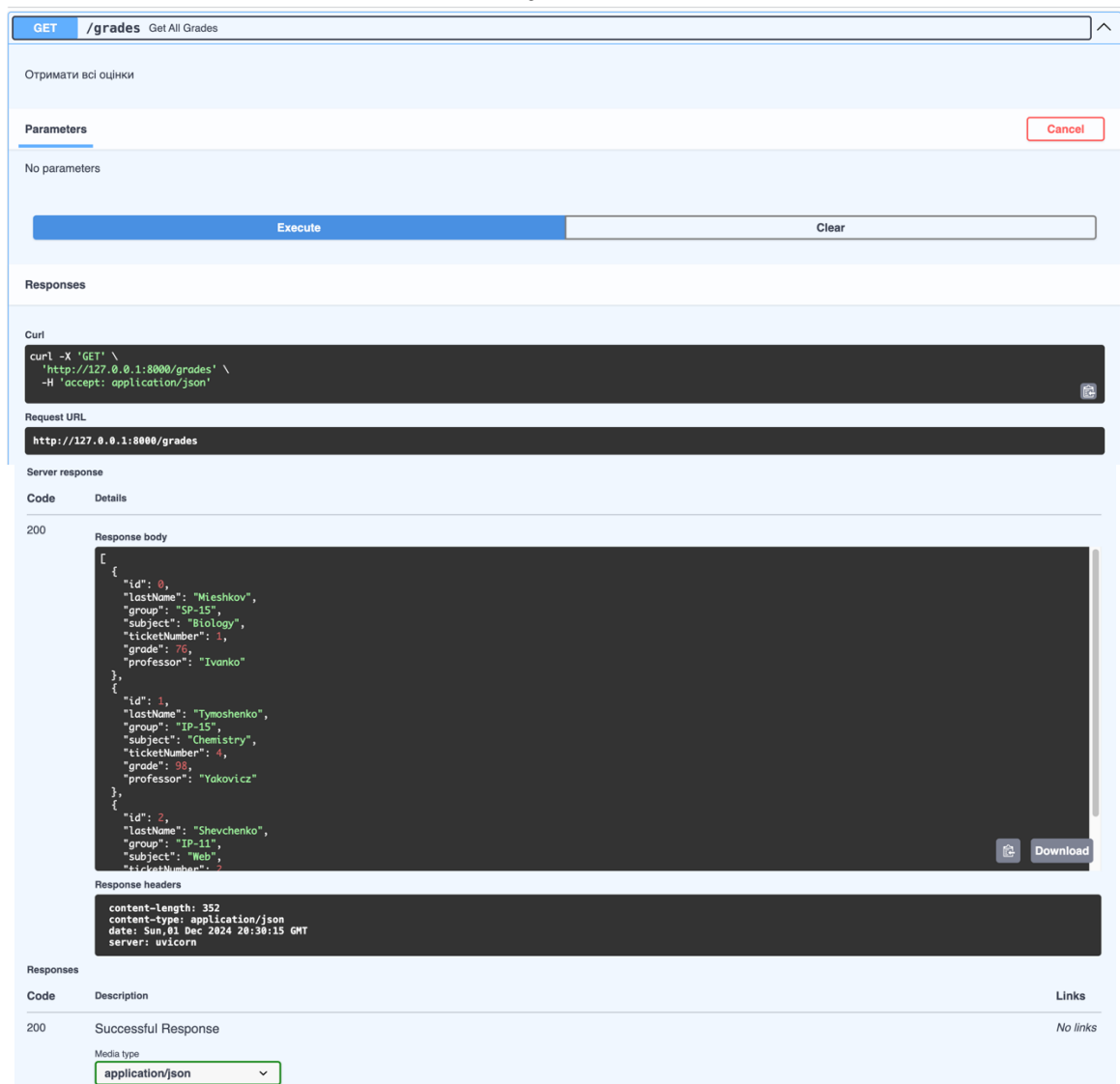


Рис. 6.2. Запит для виведення всіх документів в БД

GET

/grades/{grade_id}

Get Grade

^

Отримати оцінку за ID

Parameters

Cancel

Name	Description
grade_id required integer <small>(path)</small>	<input type="text" value="0"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/grades/0' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/grades/0
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{ "id": 0, "lastName": "Mieshkov", "group": "SP-15", "subject": "Biology", "ticketNumber": 1, "grade": 76, "professor": "Ivanko" }</pre></div> <div>Download</div> <div>Response headers</div> <div><pre>content-length: 115 content-type: application/json date: Sun, 01 Dec 2024 20:27:52 GMT server: uvicorn</pre></div>

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": 0, "lastName": "string", "group": "string", "subject": "string", "ticketNumber": 0, "grade": 0, "professor": "string" }</pre></div>	No links
422	Validation Error	No links

Рис. 6.3. Запит для виведення заданого документу в БД

POST

/grades

Create Grade

⌵

Додати нову оцінку

Parameters

Cancel

Reset

No parameters

Request body

required

application/json

⌵

```
{
  "id": 2,
  "lastName": "Shevchenko",
  "group": "IP-11",
  "subject": "Web",
  "ticketNumber": 2,
  "grade": 95,
  "professor": "Pawel"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/grades' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 2,
    "lastName": "Shevchenko",
    "group": "IP-11",
    "subject": "Web",
    "ticketNumber": 2,
    "grade": 95,
    "professor": "Pawel"
  }'
```

Request URL

http://127.0.0.1:8000/grades

Server response

Code

Details

200

Response body

```
{
  "id": 2,
  "lastName": "Shevchenko",
  "group": "IP-11",
  "subject": "Web",
  "ticketNumber": 2,
  "grade": 95,
  "professor": "Pawel"
}
```

⌵ Download

Response headers

```
content-length: 112
content-type: application/json
date: Sun, 01 Dec 2024 20:30:10 GMT
server: uvicorn
```

Responses

Code

Description

Links

Рис. 6.4. Запит для додавання документу в БД

PUT

/grades/{grade_id} Update Grade

Оновити оцінку

Parameters

Cancel

Reset

Name	Description
grade_id required	
integer	
(path)	1

Request body required

application/json

```
{
  "id": 1,
  "lastName": "Tymoshenko",
  "group": "IP-11",
  "subject": "Chemistry",
  "ticketNumber": 14,
  "grade": 100,
  "professor": "Yakovicz"
}
```

ExecuteClear

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/grades/1' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "id": 1,
    "lastName": "Tymoshenko",
    "group": "IP-11",
    "subject": "Chemistry",
    "ticketNumber": 14,
    "grade": 100,
    "professor": "Yakovicz"
  }'
```

Request URL

```
http://127.0.0.1:8000/grades/1
```

Server response

CodeDetails

200

Response body

```
{
  "id": 1,
  "lastName": "Tymoshenko",
  "group": "IP-11",
  "subject": "Chemistry",
  "ticketNumber": 14,
  "grade": 100,
  "professor": "Yakovicz"
}
```

Download

Response headers

```
content-length: 123
content-type: application/json
date: Sun, 01 Dec 2024 20:34:22 GMT
server: uvicorn
```

Responses

Рис. 6.5. Запит для оновлення документу в БД

DELETE

/grades/{grade_id}

Delete Grade

⌵

Видалити оцінку

Parameters

Cancel

Name	Description
grade_id required	
integer	
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:8000/grades/2' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/grades/2
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "message": "Grade with ID 2 deleted successfully" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 50 content-type: application/json date: Sun, 01 Dec 2024 20:36:27 GMT server: uvicorn</pre></div></div>

Responses

Code	Description	Links
200	<div>Successful Response</div> <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><div><pre>"string"</pre></div></div>	No links
422	<div>Validation Error</div> <div><div>Media type</div><div>application/json</div><div>Example Value Schema</div><div><pre>{ "detail": [{ "loc": [</pre></div></div>	No links

Рис. 6.6 Запит для видалення документу в БД

ВИСНОВОК

Під час виконання даної лабораторної роботи було створено web-додаток на базі технології FastAPI для реалізації API, що дозволяє працювати з інформацією про оцінки студентів.

Було розроблено схему даних за допомогою Pydantic, створено маршрути для CRUD-операцій (додавання, редагування, видалення та перегляду даних), а також протестовано функціональність API.

В результаті:

- Реалізовано ефективну модель зберігання даних у пам'яті додатка.
- Отримано практичний досвід роботи з FastAPI, Pydantic та тестуванням HTTP-запитів.
- Набуто розуміння основ розробки RESTful API, яке може бути масштабоване на складніші проекти з використанням баз даних.

Додаток виконує поставлені завдання відповідно до варіанту роботи, демонструючи основні можливості FastAPI для створення сучасних веб-сервісів.