📊 Financial_Api_Project Report

prepared by:

-Name: [Asala Shawky]

-Date: [4/22/2025]

Project Overview:

Managing financial transactions using

Python

Django

Django Rest Framework (DRF)

Testing on

Postman

Implemented Endpoints:

Endpoint

Methods

Description

| `server url` /api/transactions/` | GET, POST | عرض وإنشاء معاملات مالية

| `server url` /api/transactions/<id>/` | GET, PUT, DELETE | عرض، تعديل، أو حذف معاملة حسب الـ ID |

| `server url` /api/users/` | GET | عرض جميع المستخدمين

| `server url` /api/users/<user_id>/transactions/` | GET | عرض المعاملات الخاصة بمستخدم معين

| `server url `/api/reports/?user_id=&year=&month=` | GET | تقرير إجمالي المعاملات لمستخدم معين بالسنة أو الشهر|

🧪 Testing Strategy:

✅ **Tools Used:**

- **Postman**: لتنفيذ واختبار كل الـendpoints.

🧪 **Test Cases Covered:**

- إنشاء معاملة مالية(POST)
- عرض جميع المعاملات(GET)
- عرض معاملة واحدة(GET by ID)
- تعديل معاملة(PUT)
- حذف معاملة(DELETE)
- عرض المستخدمين(GET)
- عرض معاملات مستخدم(GET)
- استخراج تقرير سنوي/شهري(GET with params)

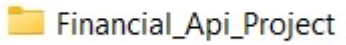
📁 Deliverables:

1. **Source code** (`.zip` أو GitHub repo).
2. **Postman Collection** (`.json` export).
3. **Development Report** (.pdf).

Now, I will show you step-by-step

How to create a comprehensive financial API

1. Create a folder of the project



2. Open folder in <vsc< go to terminal< make virtual environment
    - Pip install virtualenv
    - python -m virtualenv venv
    - 

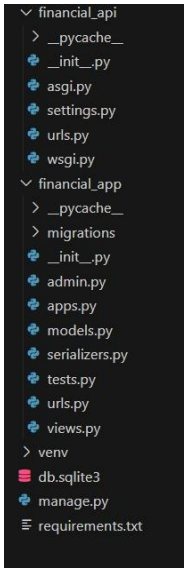
3. Activate the virtual environment
    - venv\ scripts\activate
    - if didn` t work use : **set-ExecutionPolicy RemoteSigned –scope CurrentUser**

4. download frameworks
    - pip install Django djangorestframework

```
PS D:\asala\finance_API> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
PS D:\asala\finance_API> venv\Scripts\activate
(venv) PS D:\asala\finance_API> pip install django djangorestframework
Collecting django
  Downloading Django-5.2-py3-none-any.whl (8.3 MB)
     ---------------------------------------- 8.3/8.3 MB 1.9 MB/s eta 0:00:00Collecting djangorestframework
  Downloading djangorestframework-3.16.0-py3-none-any.whl (1.1 MB)
     ---------------------------------------- 1.1/1.1 MB 1.6 MB/s eta 0:00:00Collecting asgiref>=3.8.1
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
     ---------------------------------------- 44.4/44.4 kB 2.3 MB/s eta 0:00:00Collecting tzdata
  Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
     ---------------------------------------- 347.8/347.8 kB 1.8 MB/s eta 0:00:00Installing collected packages: tzdata, sqlparse, asgiref, django, djangorestframework
Successfully installed asgiref-3.8.1 django-5.2 djangorestframework-3.16.0 sqlparse-0.5.3 tzdata-2025.2

[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

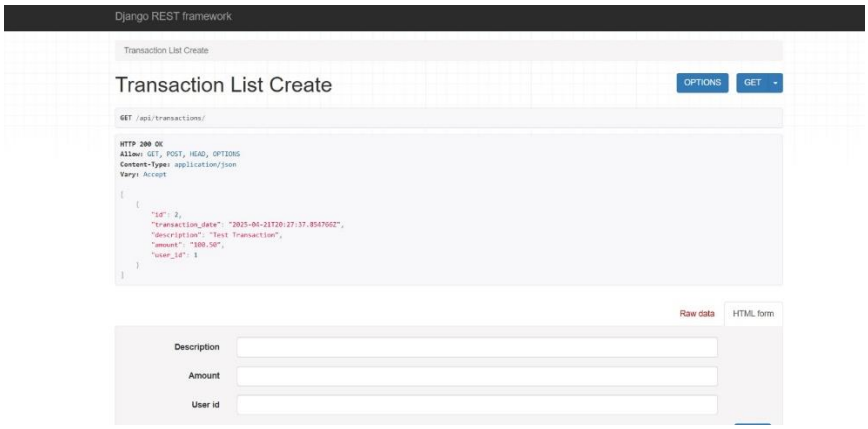
5. pip freeze > requirements.txt
6. make apps, Api, migration
    - Django-admin startproject financial_app
    - Django-admin startapp financial_project
    - Python manage.py makemigrations
    - Python manage.py migrate
7. apps will be appear here :



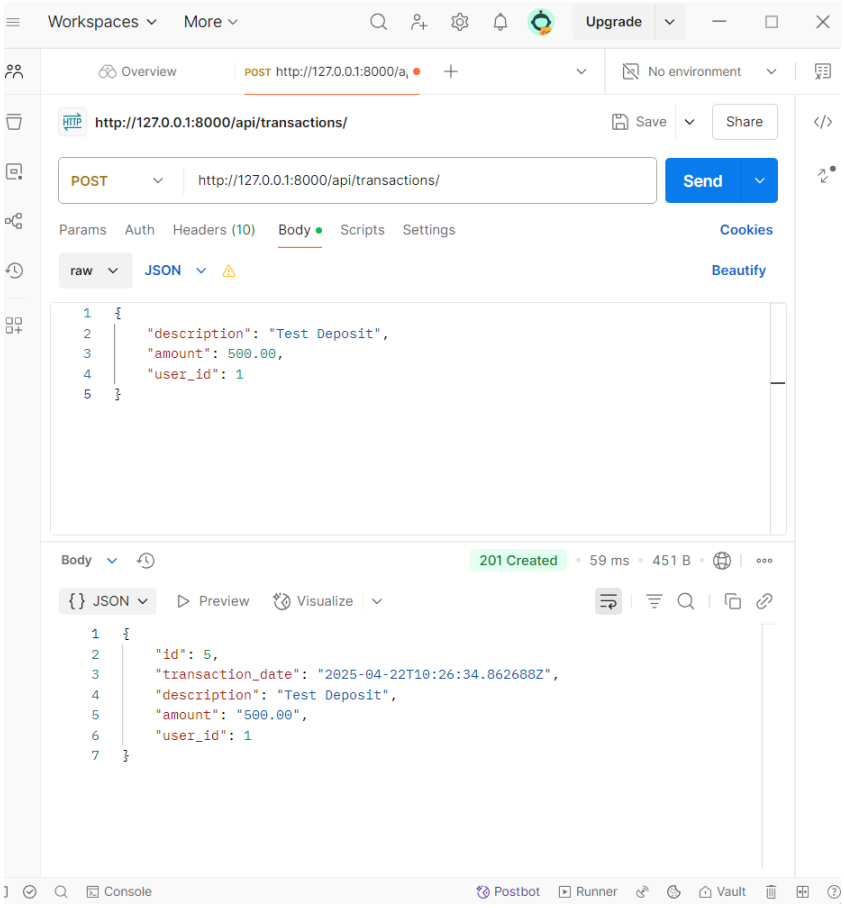
8. create superuser (admin)
    - python manage.py createsuperuser
    - put : username,email,password
9. go to< financial.api<settings.py < INSTALLED_APPS
    - put : ‘rest_framework’ , ‘financial_app’
10. https://github.com/asala810/financial_api_project.git
11. Run server:
    - Python manage.py runserver
    - Apply live server
12. Time to test :
    - Put << http://127.0.0.1:8000/api/transactions/
    - Put << http://127.0.0.1:8000/admin/
    - Put << http://127.0.0.1:8000/api/users/

13.

14.

15. 

16. Time to test on postman

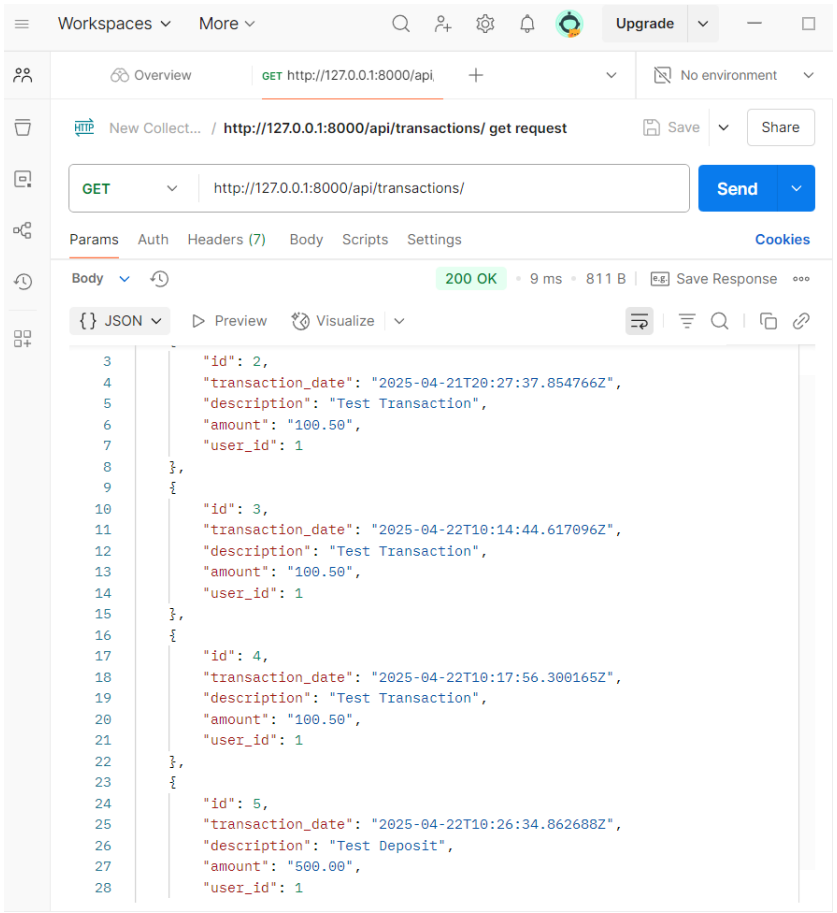


Post. Request

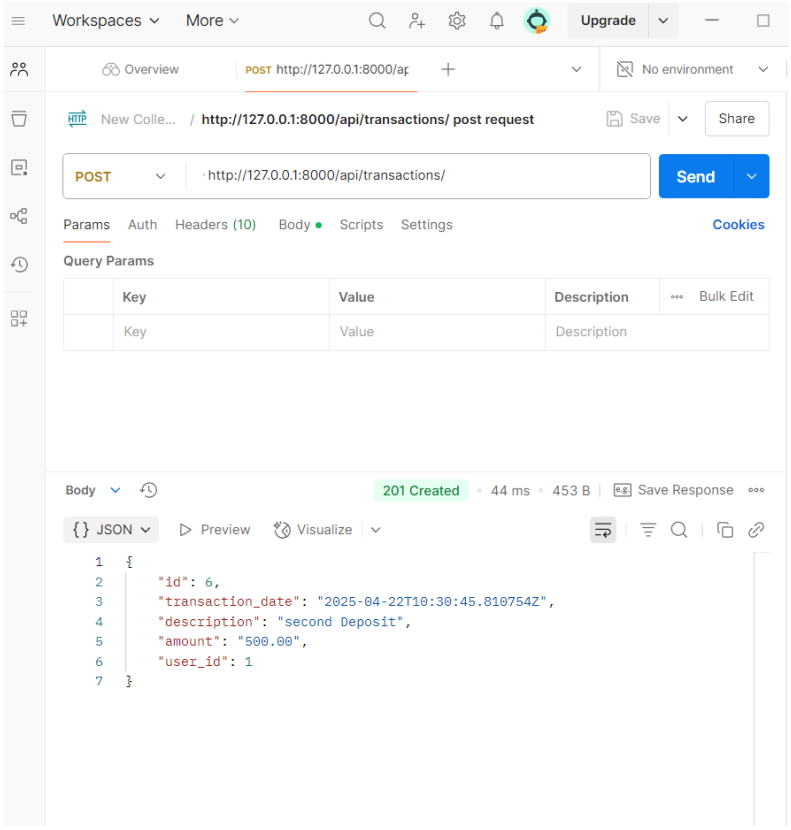
- Headers >> Content-Type: application/json

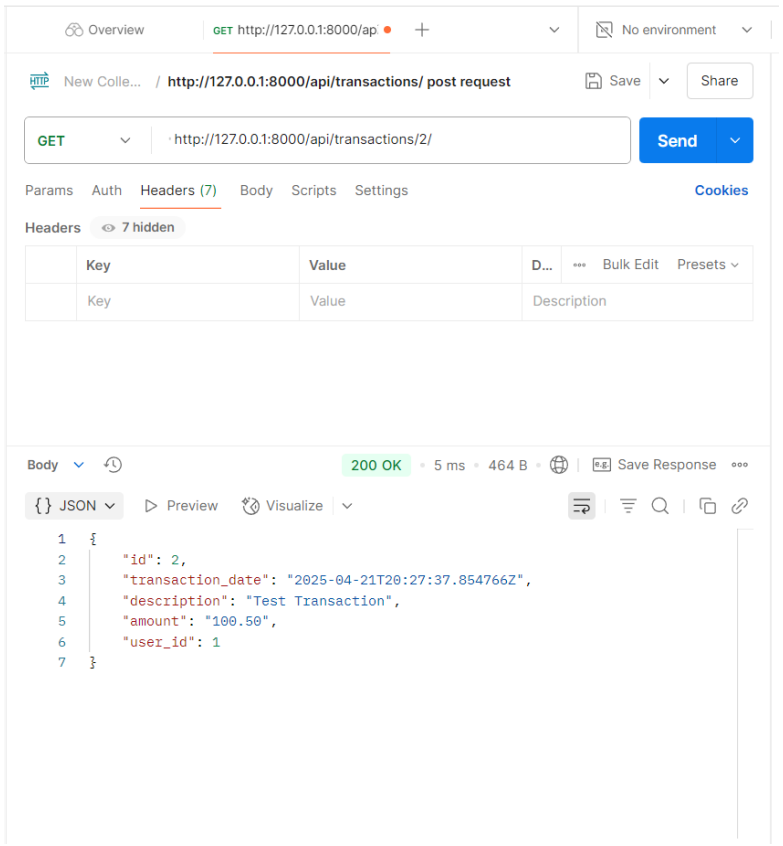
➢ **Body** (raw + JSON):

```
{
  "description": "Test Deposit",
  "amount": 500.00,
  "user_id": 1
}
```

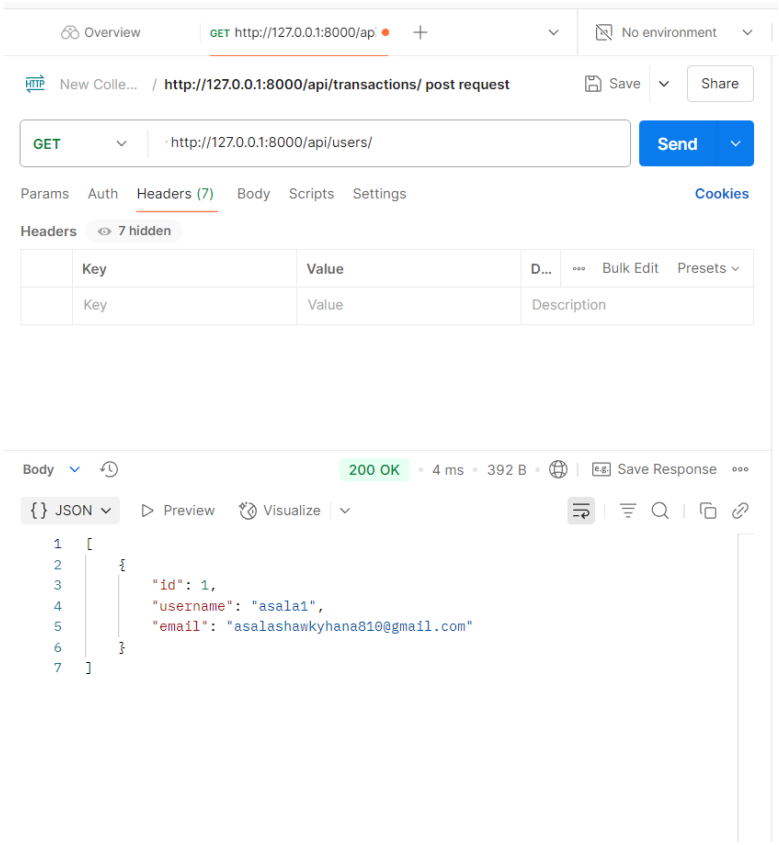


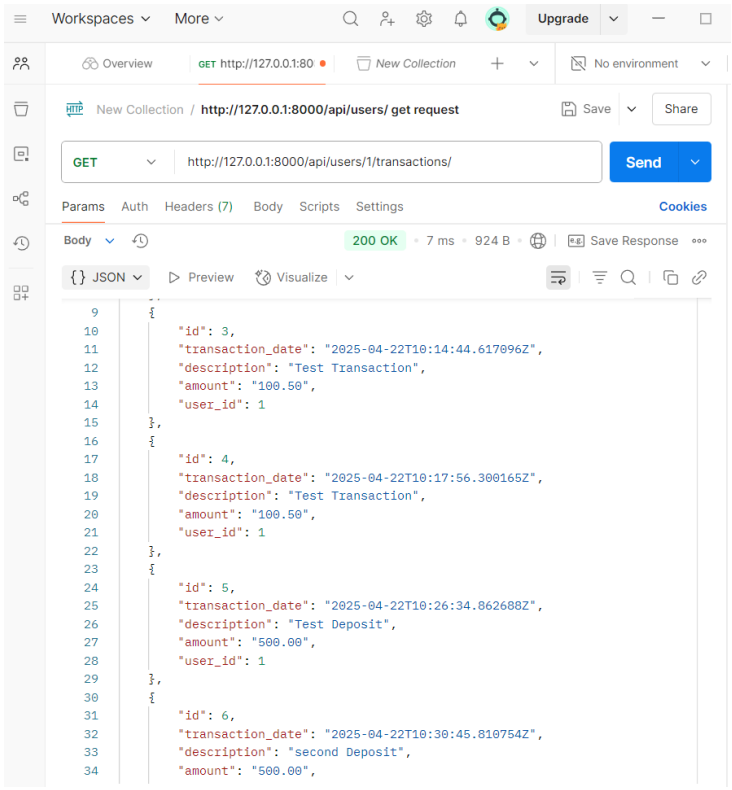
➢ Get. Request



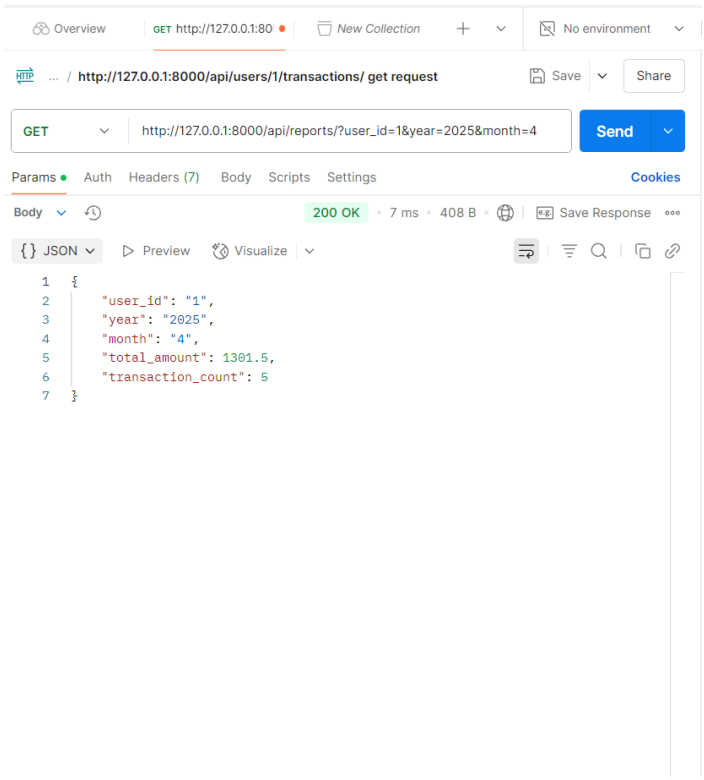


To get specific data about user(1or2or3or,etc)





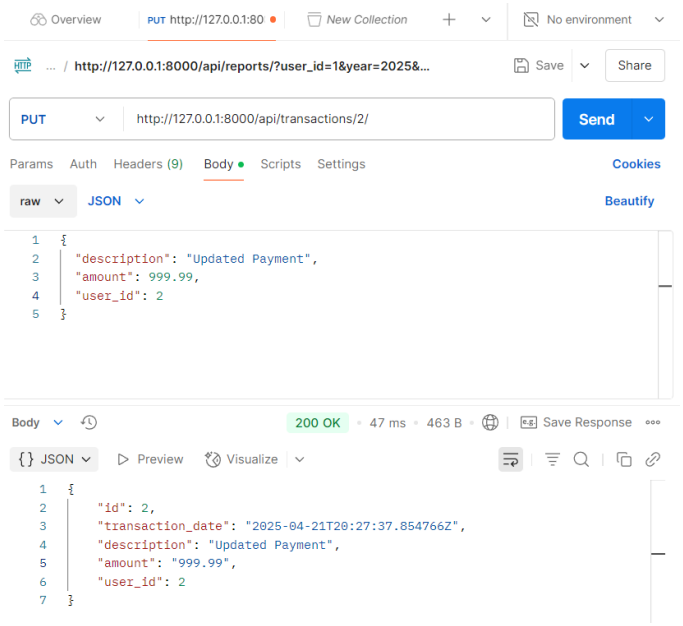
Every data we post , stored in the ‘get request’



➤

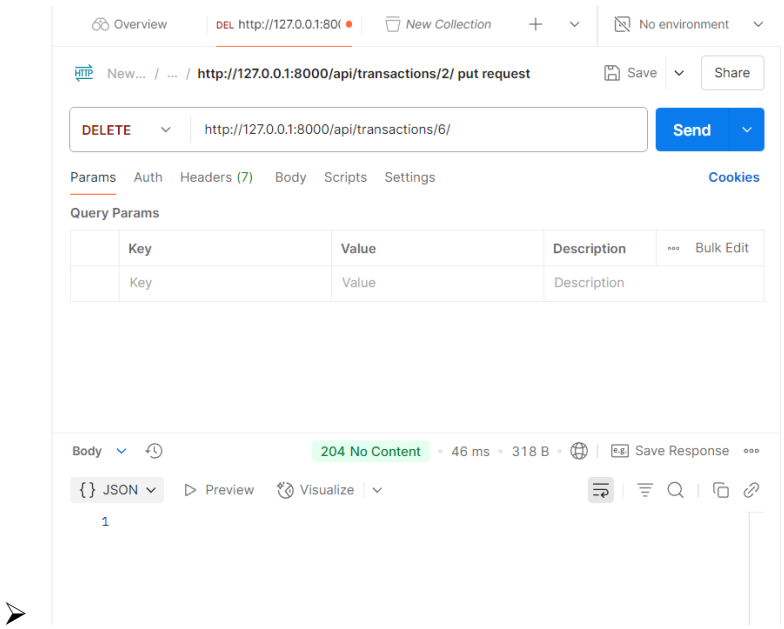
To report all transactions of specific user along the year, month ,

Report total amount



➤

If you want to update data of specific user, with his id



Delete data of specific user

Now you have all details

Challenges that faced me : how to upload folders on GitHub

And make a link between GitHub and visual studio code