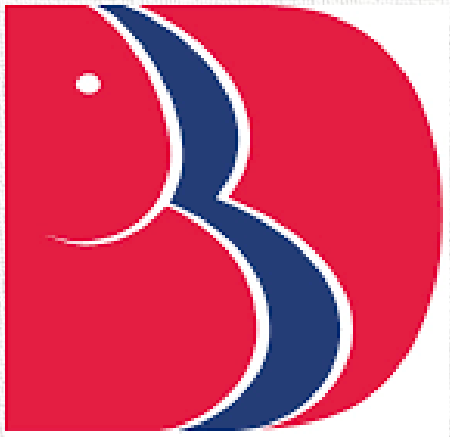
SCHOOL OF COMPUTER APPLICATION

CLOUD APPLICATION DEVELOPMENT

BCADSN12101



Submitted By:

Name: Shantanu Pandey

Section: BCA DS 16

Roll No: 05

Semester: 2nd

TO

Mr. Nikhil Sharma

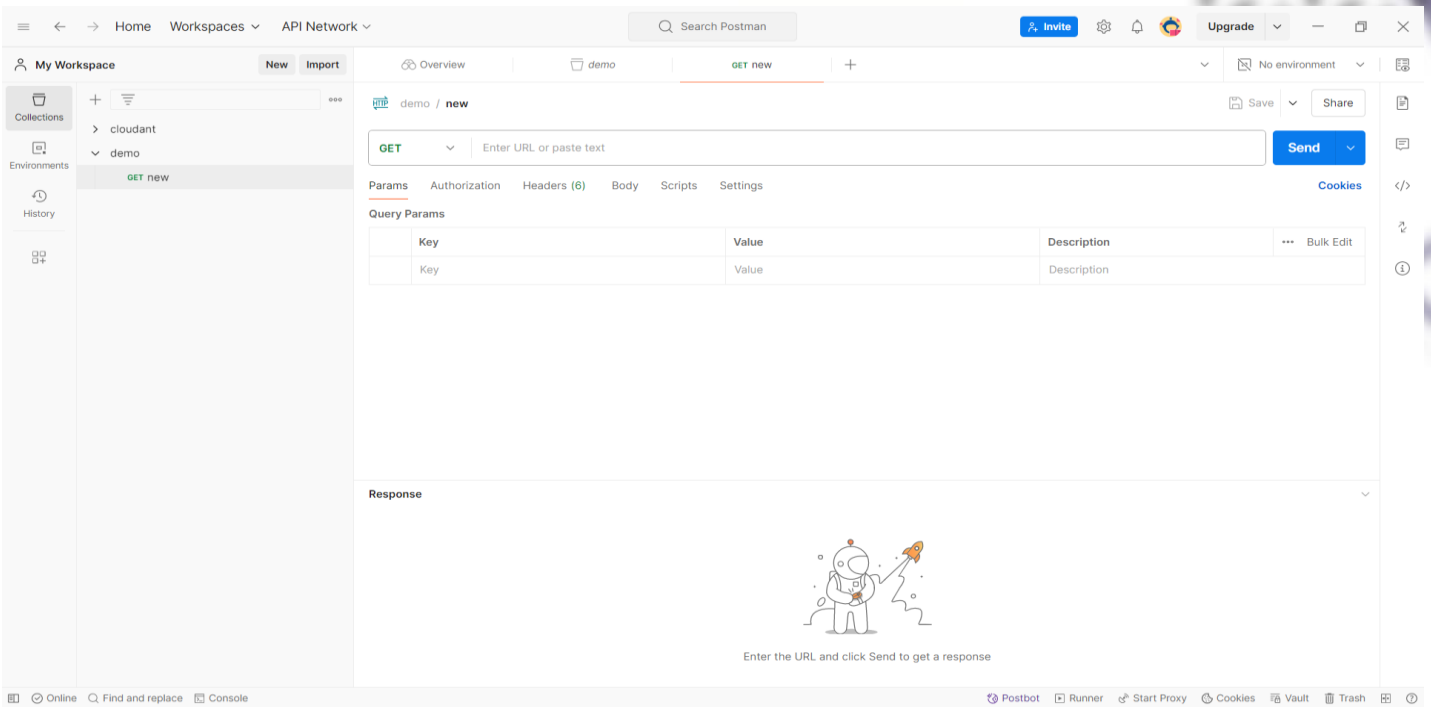
Babu Banarasi Das University, Lucknow

BBD City, Ayodhya Road, Lucknow Uttar Pradesh- 226028 Bharat

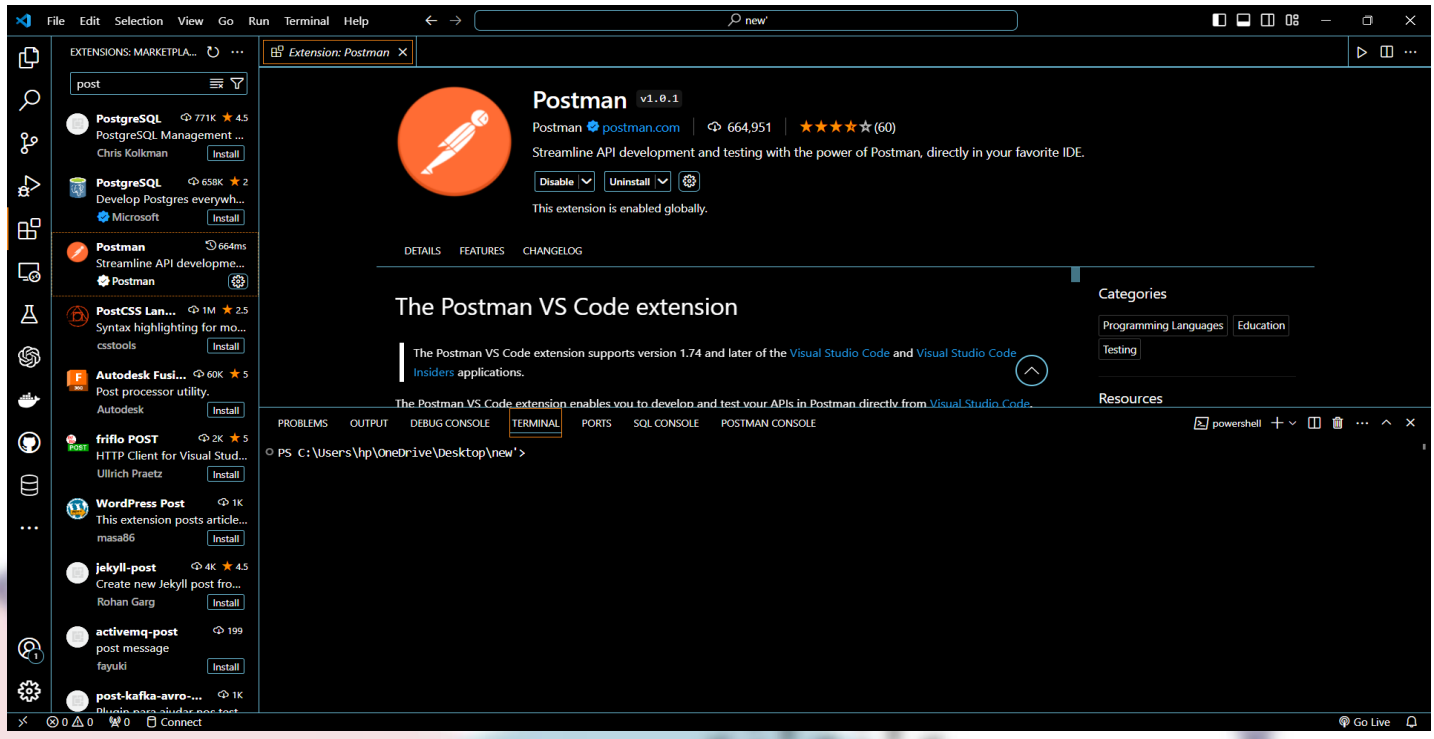
INDEX

REST API IMPLEMENTAION USING FLASK

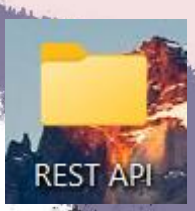
STEP 1 – Install POSTMAN application (https://www.postman.com/downloads/) and ensure that you have installed Postman correctly and running on your system. Sign-in with your credentials.

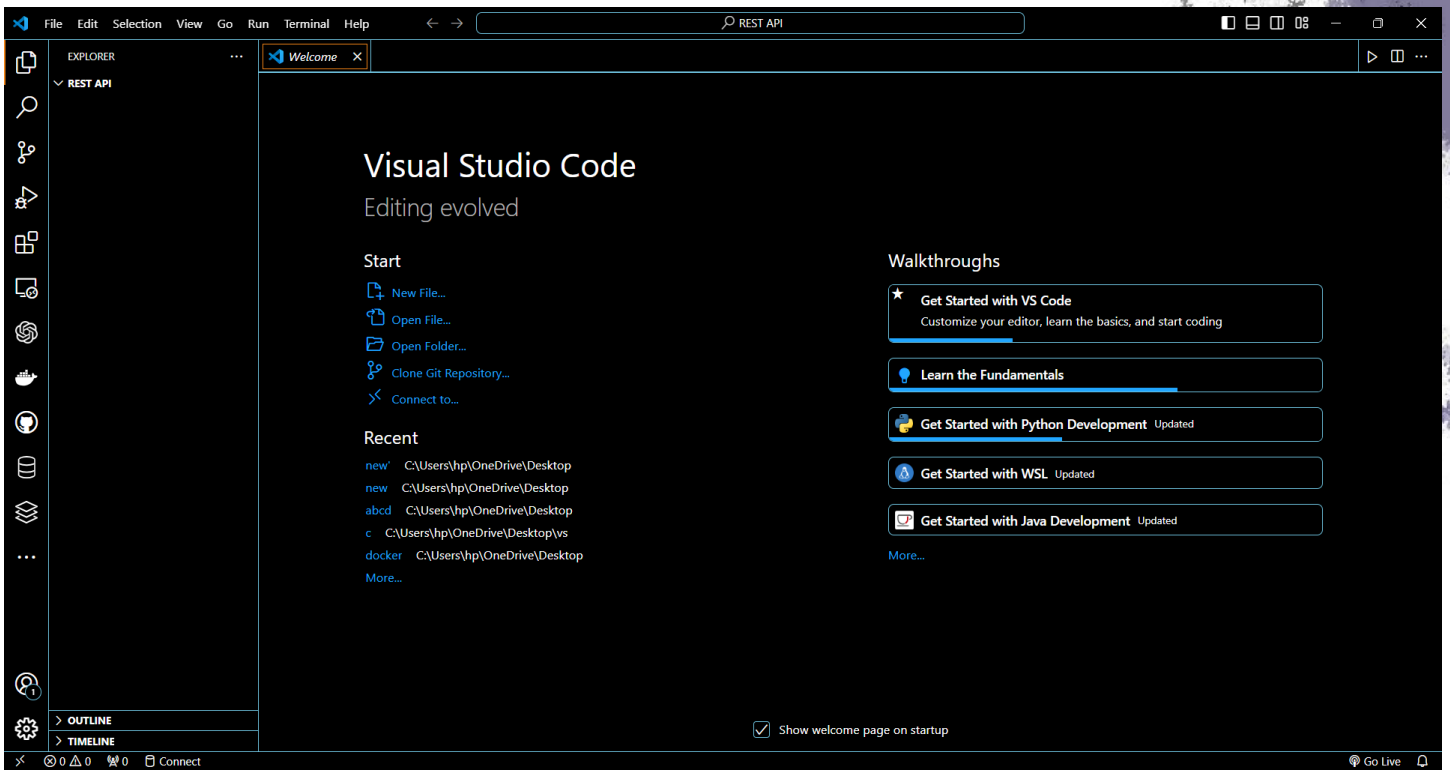


STEP 2 - Install VS Code and the official Postman extension from the VS Code Marketplace. (https://code.visualstudio.com/docs/containers/overview).

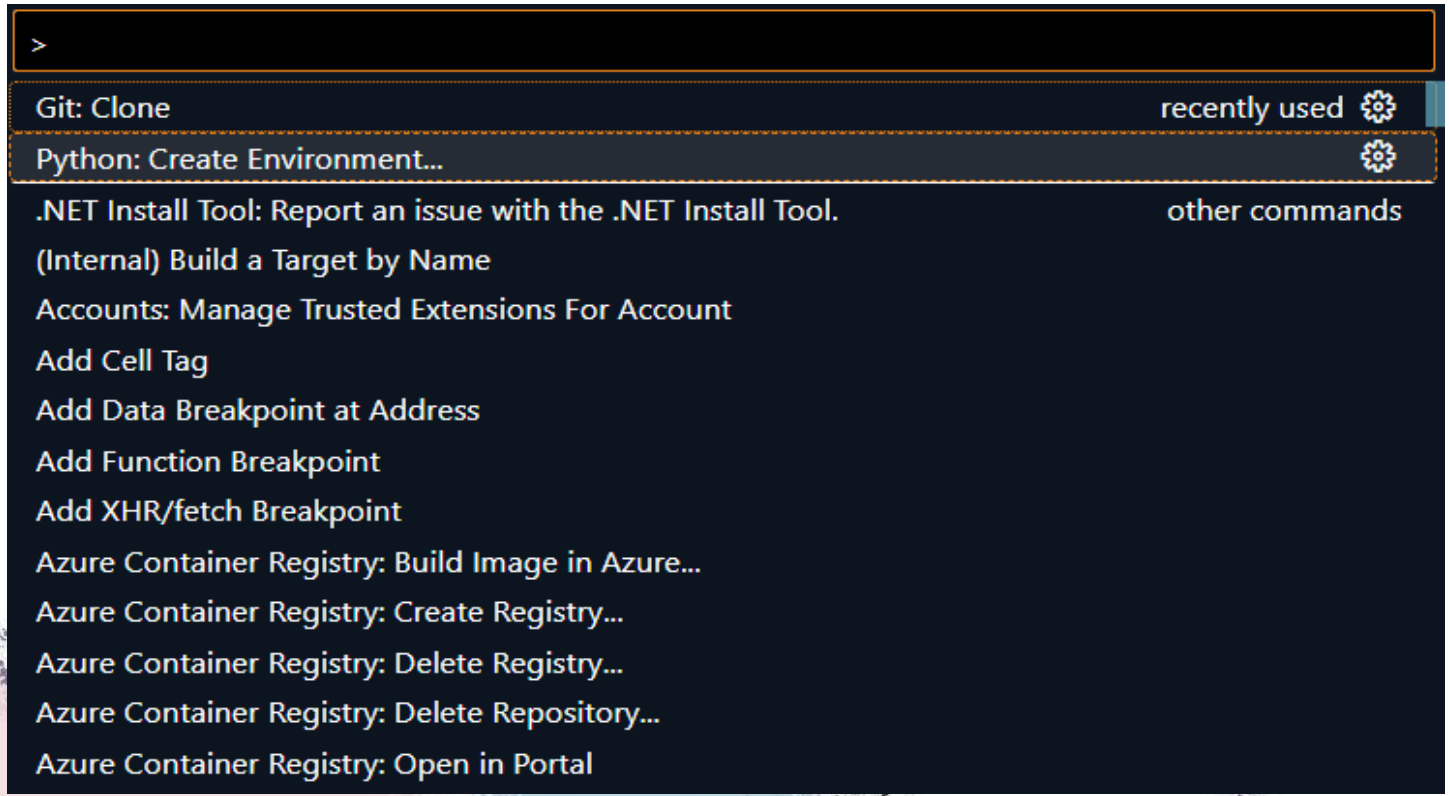


STEP 3 - Create a new folder named “REST API” and open it in VS Code.



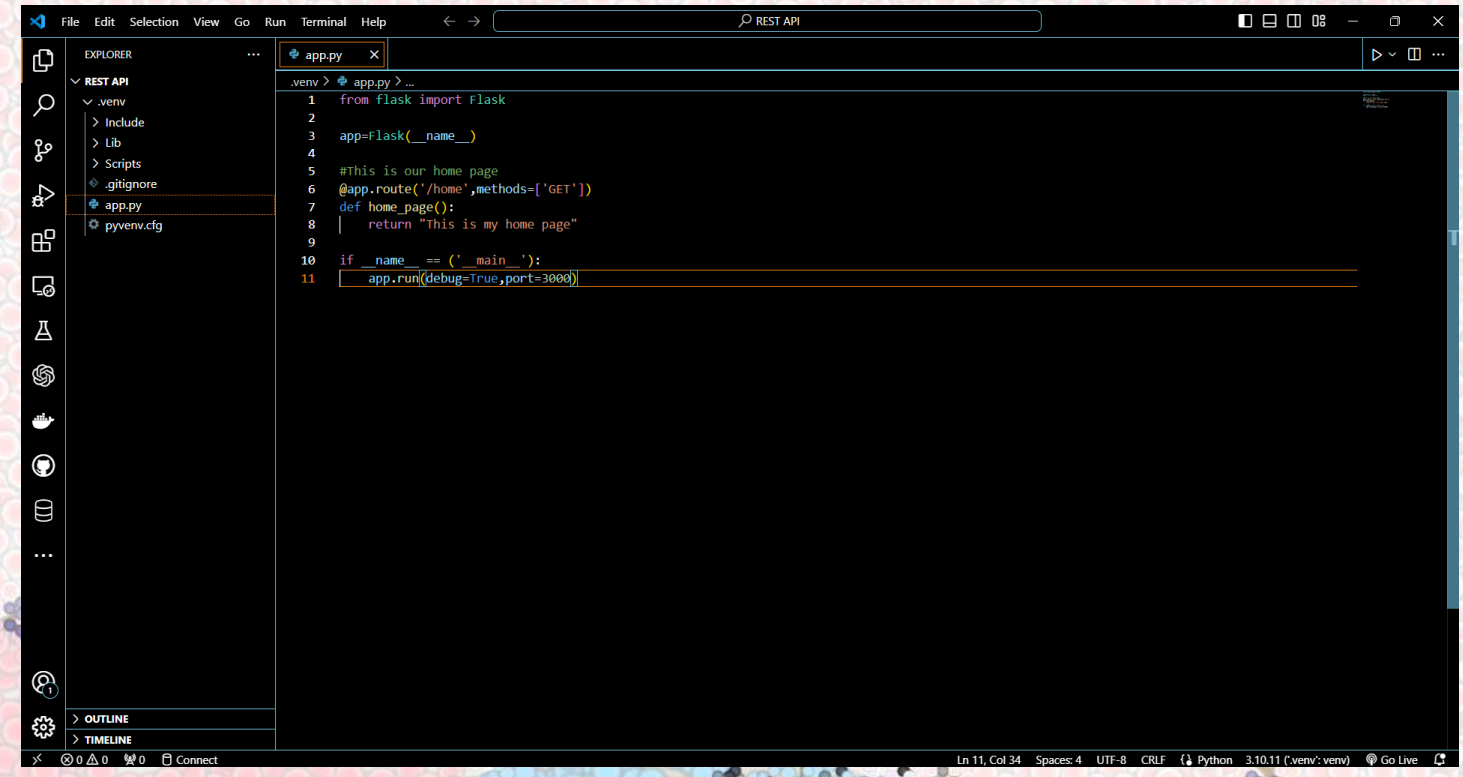


STEP 4 - Open the Command Palette (Ctrl+Shift+P) and choose "Python: Create Environment". Select the appropriate app type (Python: Flask) and provide the entry point path for your application.



STEP 5 - Create a Python file named "app.py" inside a venv folder.

STEP 6 - Write a Python Flask code for providing a localhost for the application and also define route to 'home' using "GET Methods"

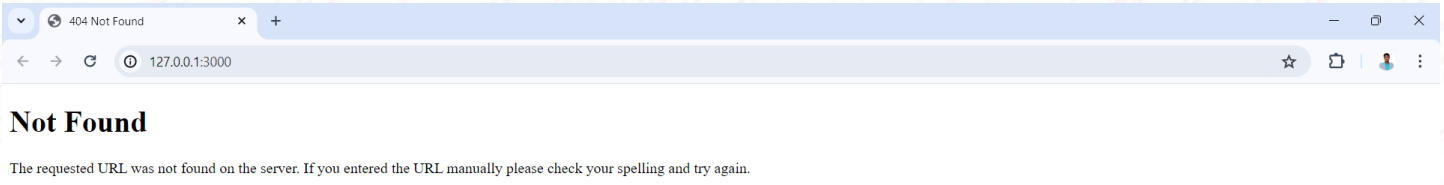


STEP 7 – Run the file to your localhost and use "http://127.0.0.1:3000/home" to redirect on your home page that you created in code.

STEP 8 - At the execution time of code you will see HTTP error which is called "404" error in which "NOT FOUND" will we displayed.

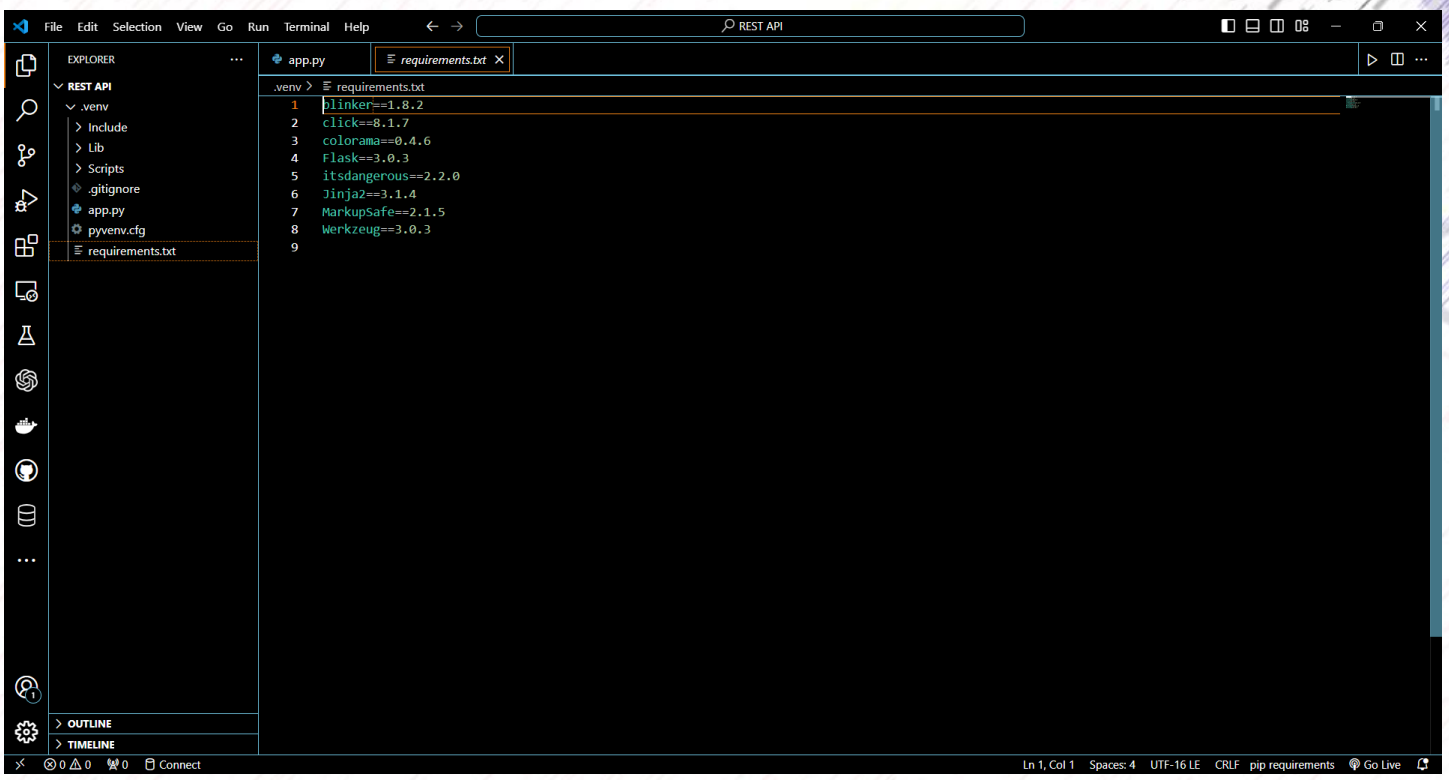
Which defines that you will not decide any route to this location.

The 404 code means that a server could not find a client-requested webpage.



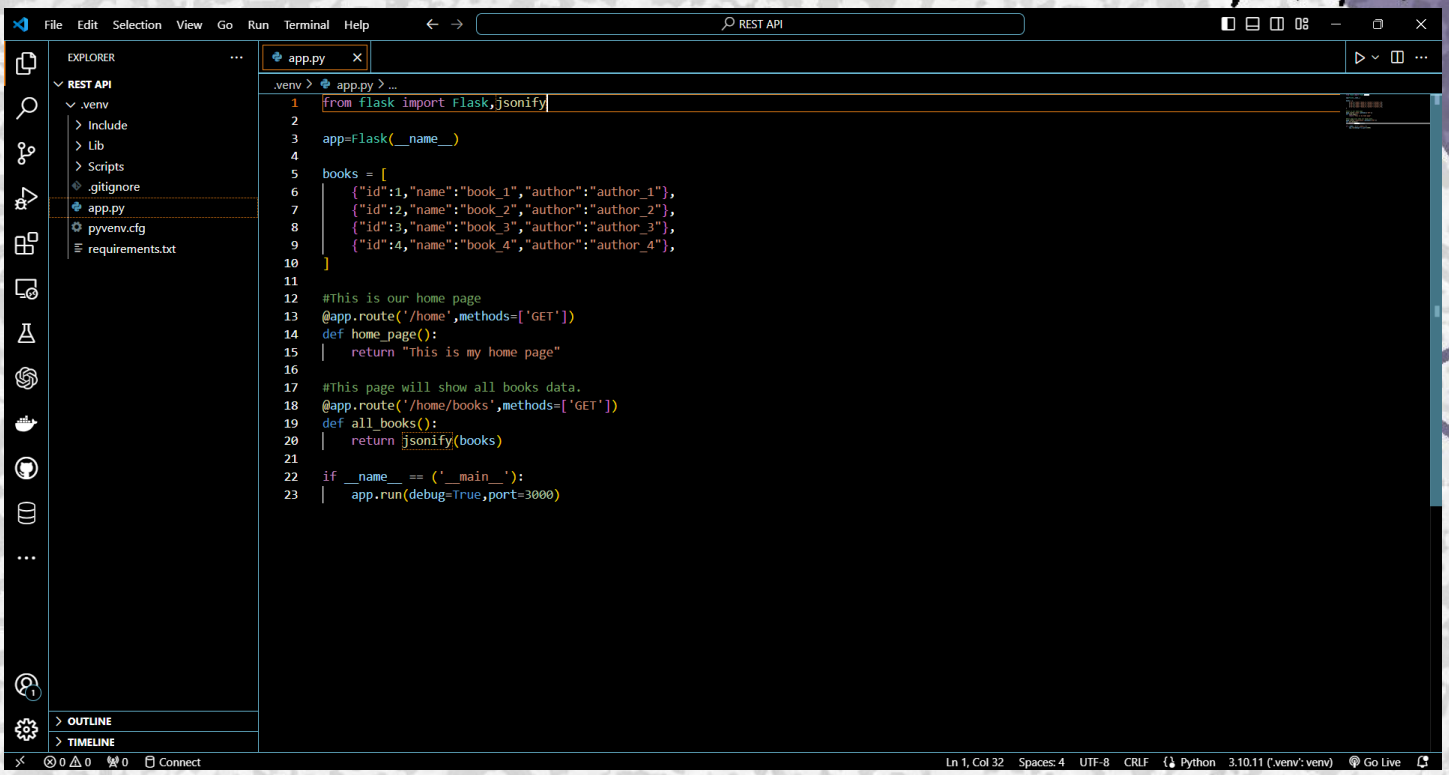
STEP 9 - Open Terminal and get inside the venv folder. Run the command (`pip freeze > requirements.txt`) to create requirements.txt file which contains all python packages for your application.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL CONSOLE  POSTMAN CONSOLE
PS C:\Users\hp\OneDrive\Desktop\REST API> cd .venv
PS C:\Users\hp\OneDrive\Desktop\REST API\.venv> pip freeze > requirements.txt
PS C:\Users\hp\OneDrive\Desktop\REST API\.venv>
```

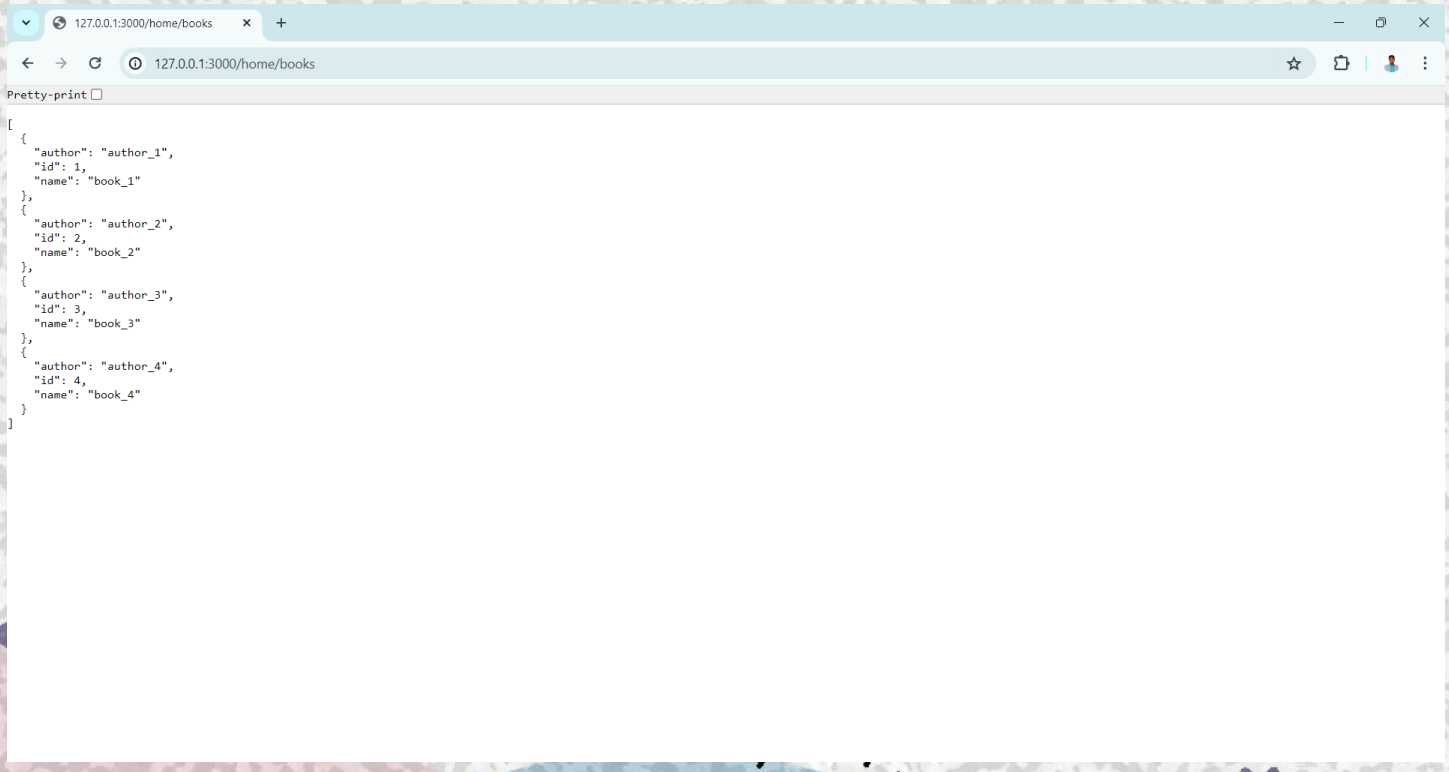


STEP 10 - Create a json file named "books" and store 4 books data.

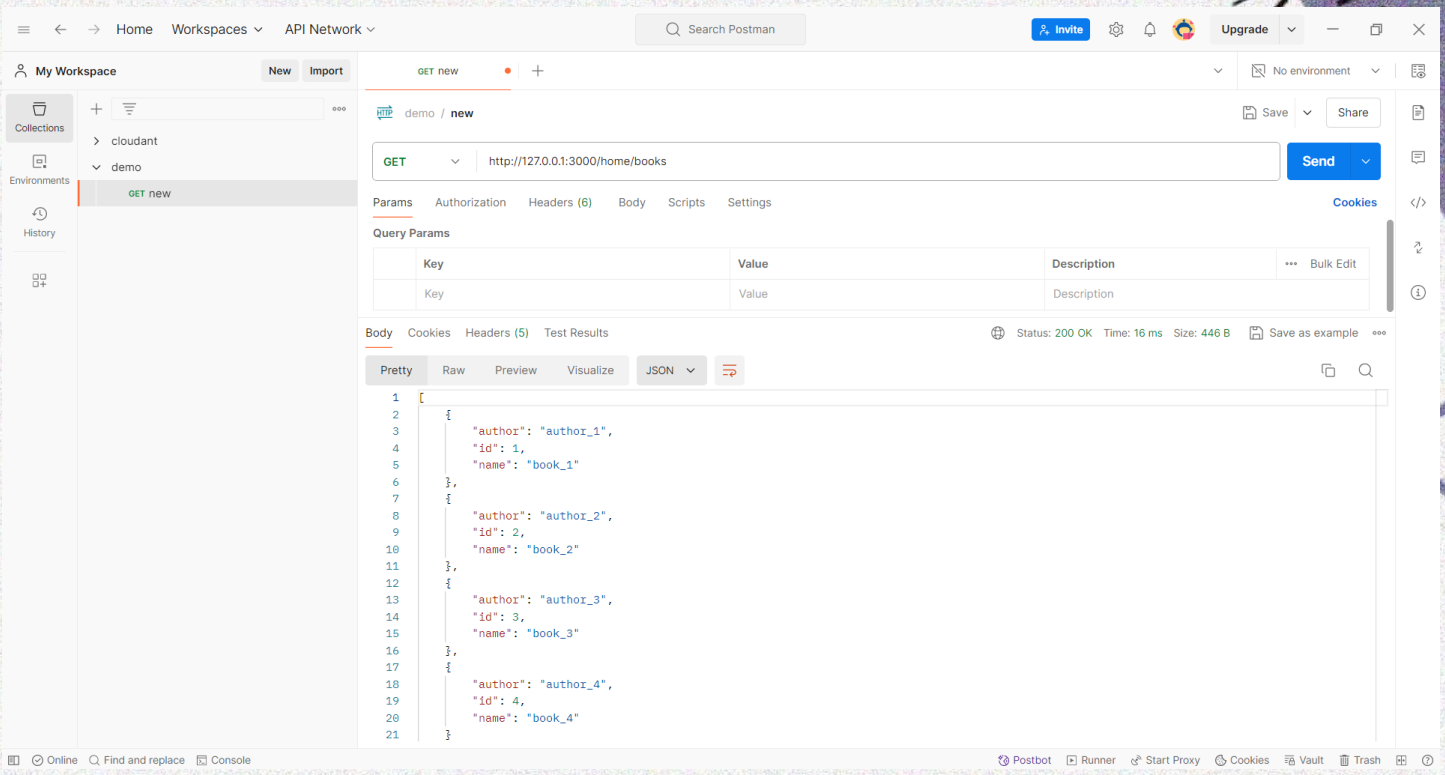
STEP 11 - Import jsonify function(It sets the correct response headers and content type for JSON responses) and create a new app route named "/books" for displaying books data using 'GET' methods.



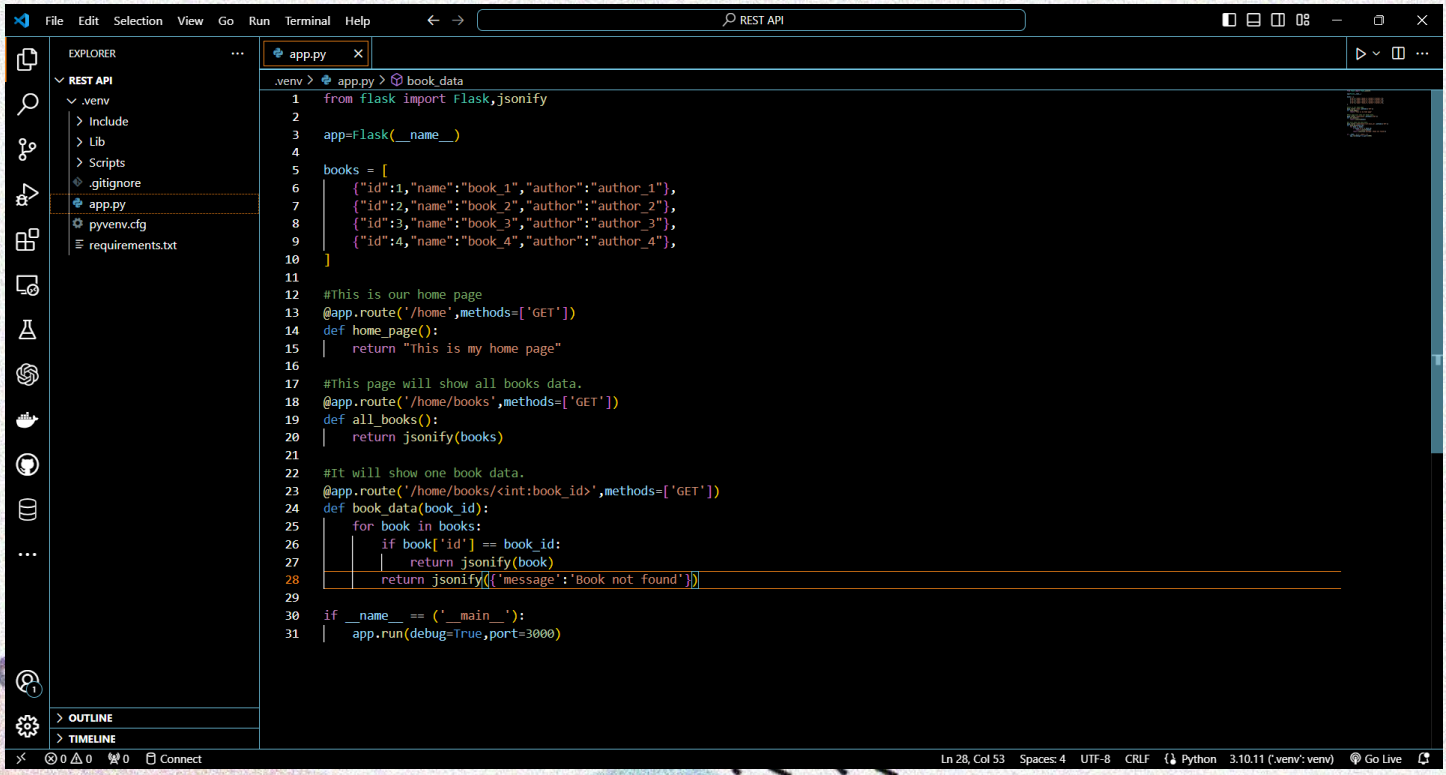
STEP 12 - Run the code and after opening the route of "/home/books" you will see that all books data which are stored in your file that all should be displayed.



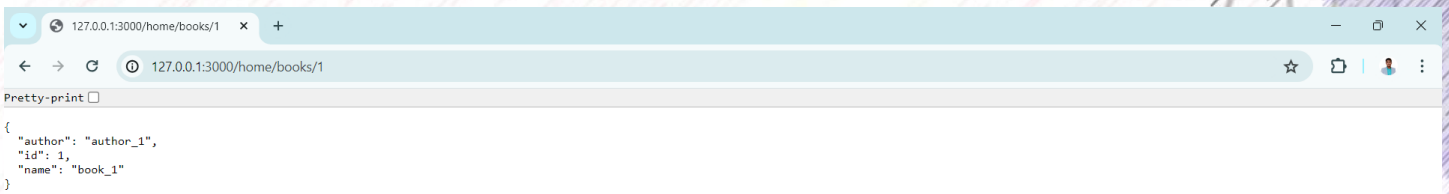
STEP 13 - Copy the URL (http://127.0.0.1:3000/home/books) and paste it in Postman with "GET" method, and you will see that all the books data has been shown here.



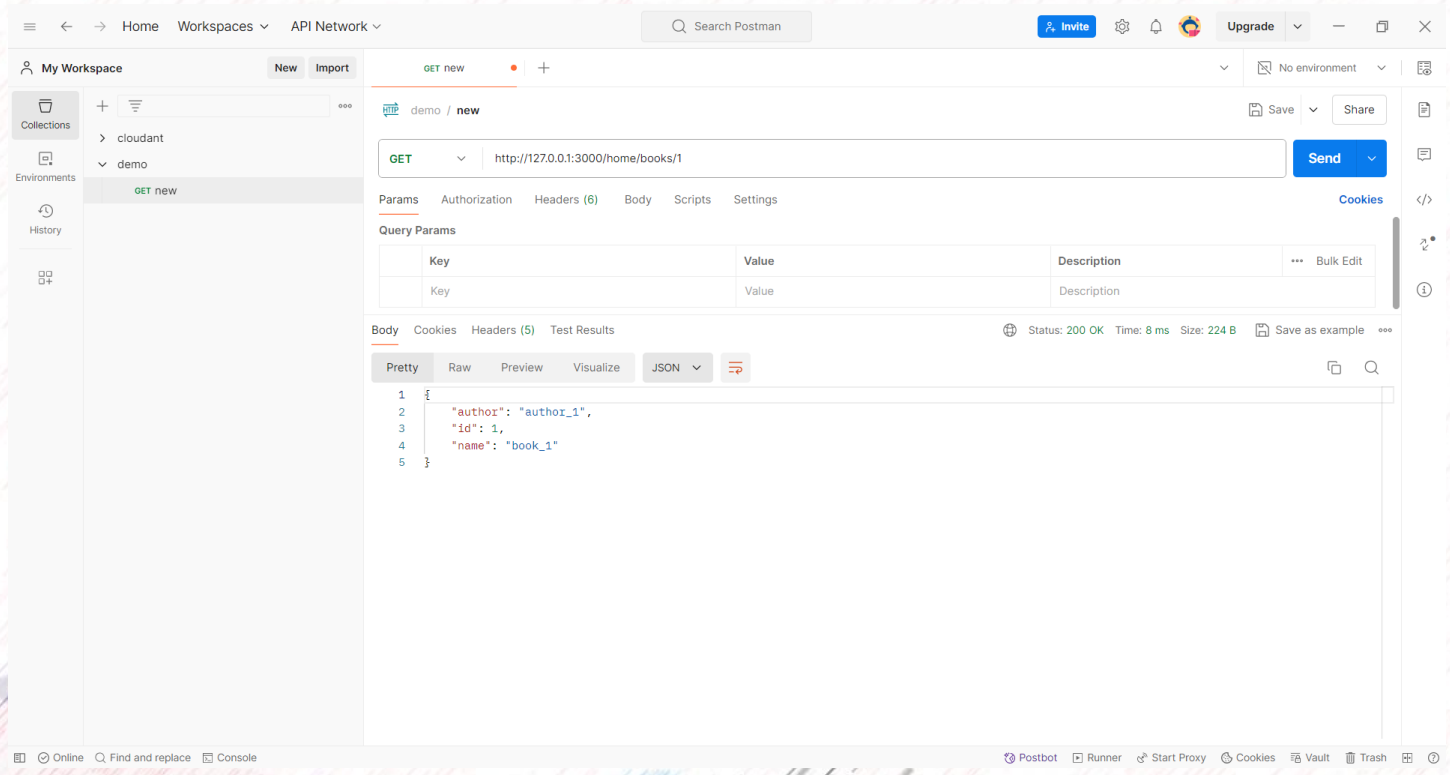
STEP 14 – After getting the request in Postman, we need to create a new app route (/home/books/1) named "/books/1" for getting extracting single book data and we use message for invalid book id.



STEP 15 - Run the code and after opening the route of "/home/books/1" you will see that single books data which are stored in your file that all should be displayed.

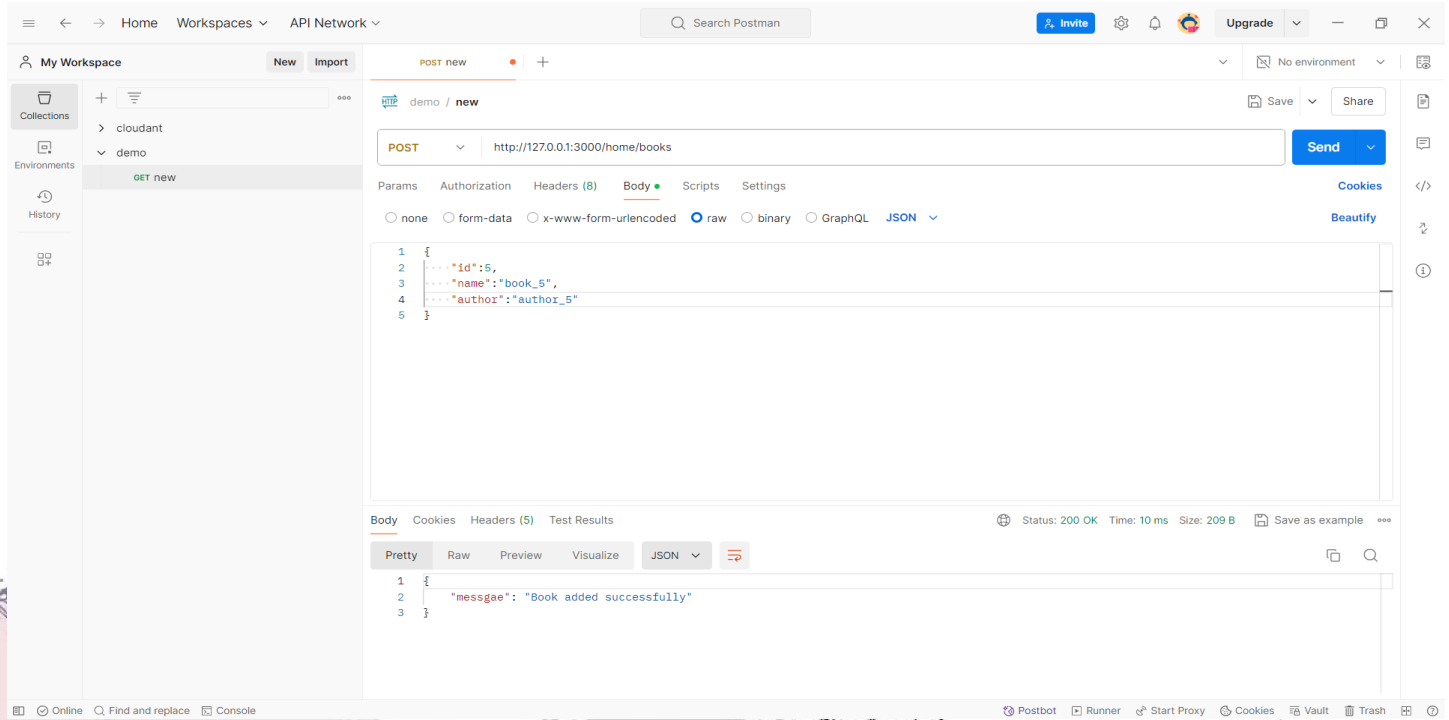


STEP 16 - Copy the URL (http://127.0.0.1:3000/home/books/1) and paste it in Postman with "GET" method, and you will see that the single books data has been shown here.

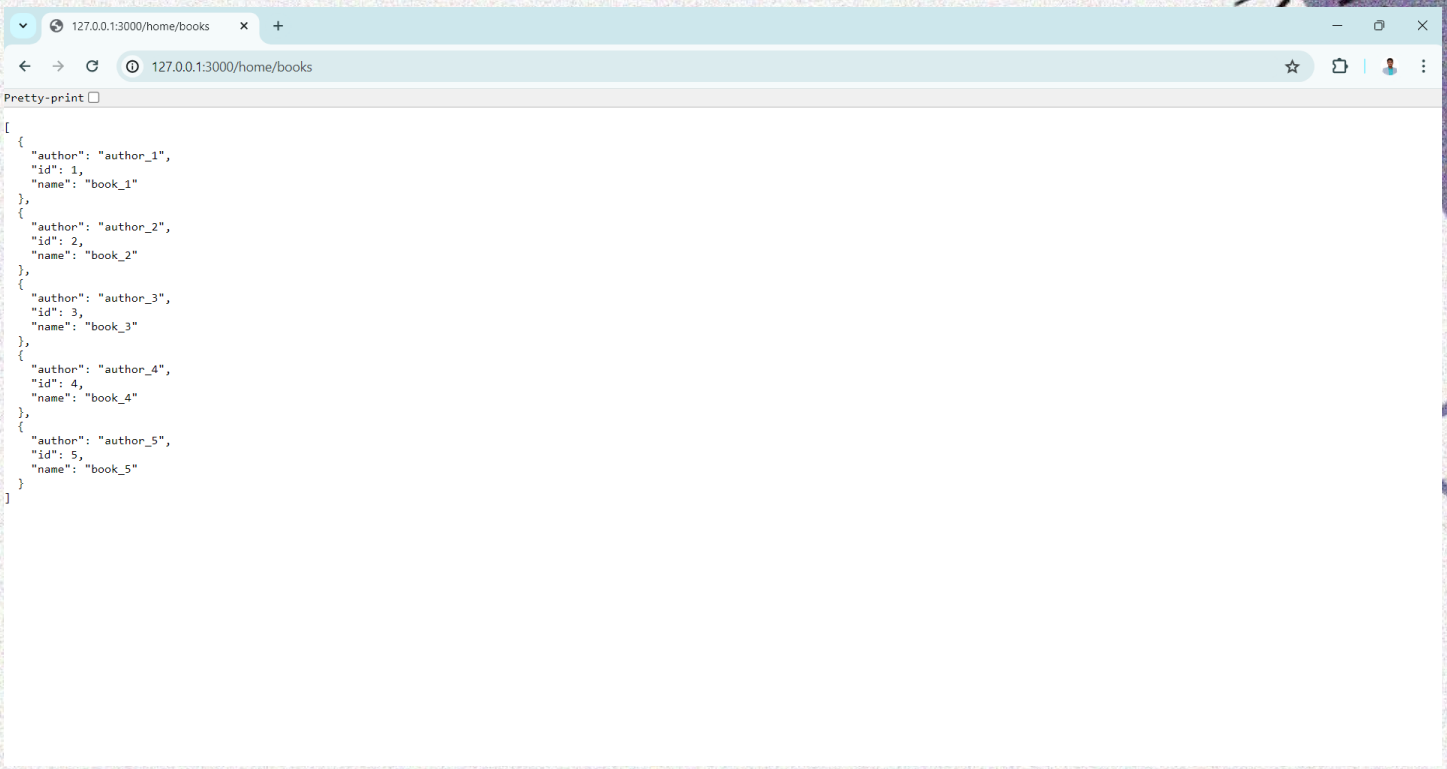


STEP 17 - After getting the output in Postman, we need to import request function (It sends seven main kinds of request to a web server) and create a structure for adding new book data from external.

STEP 18 - Run the code at localhost and copy the URL and in this we add a new book data ("author": "author_5","id": 5, "name":"book_5") with book id "/books/5" using 'POST' methods in Postman.

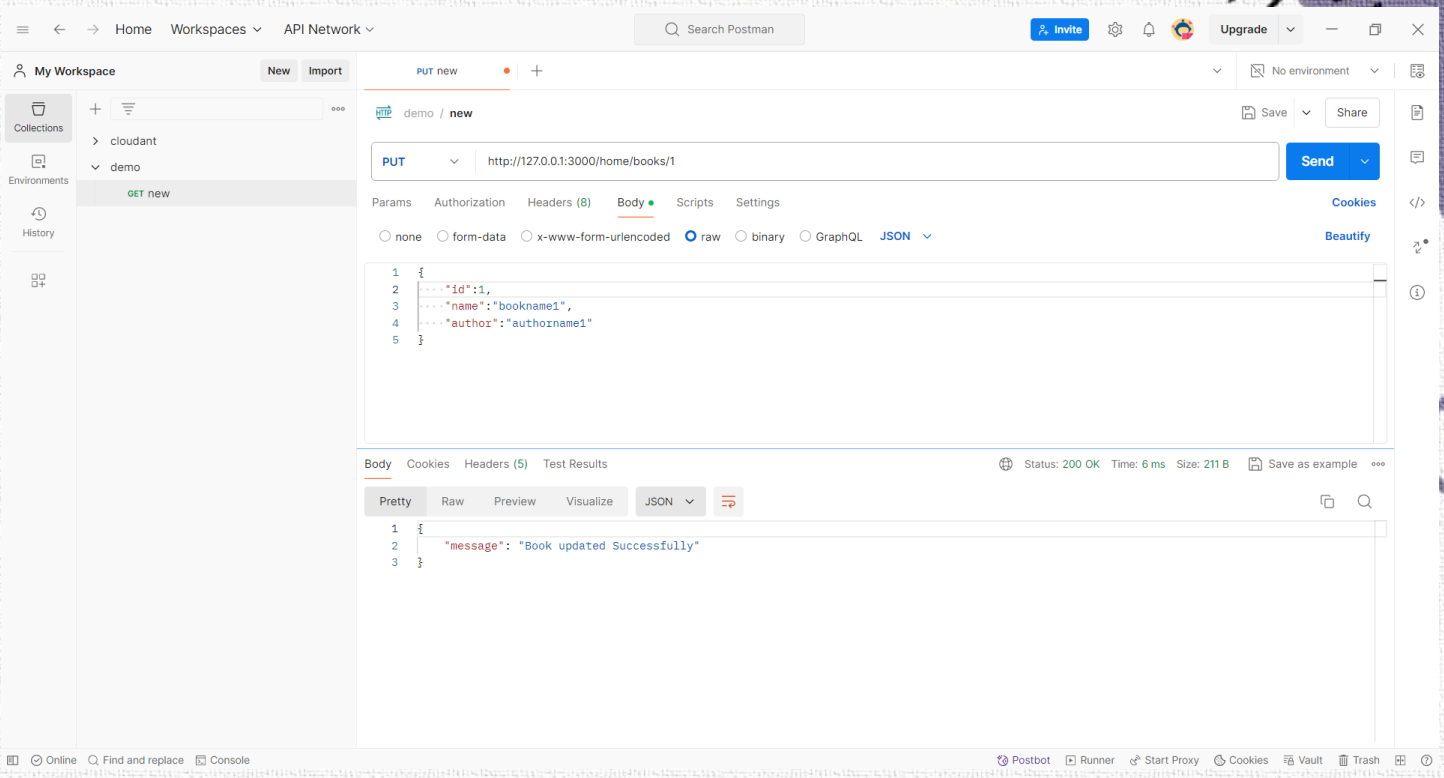


STEP 19 - After running the code the message prints that "Book added successfully", Let's check the update in localhost by refreshing the link (You will see that new book data added successfully)

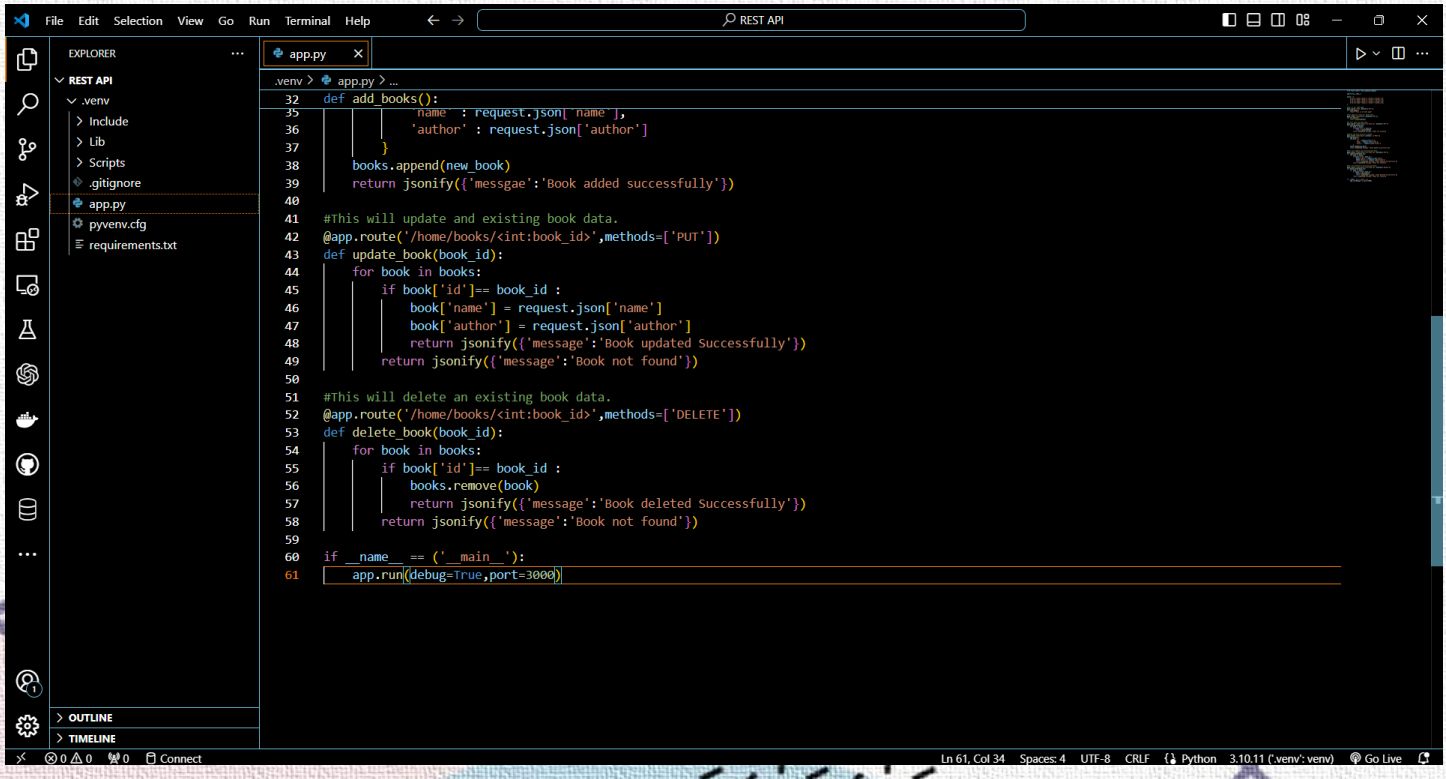


STEP 20 – For updating an existing book data we use "PUT" method. Create a new structure for using PUT methods by giving the condition of matching Book id of particular book data.

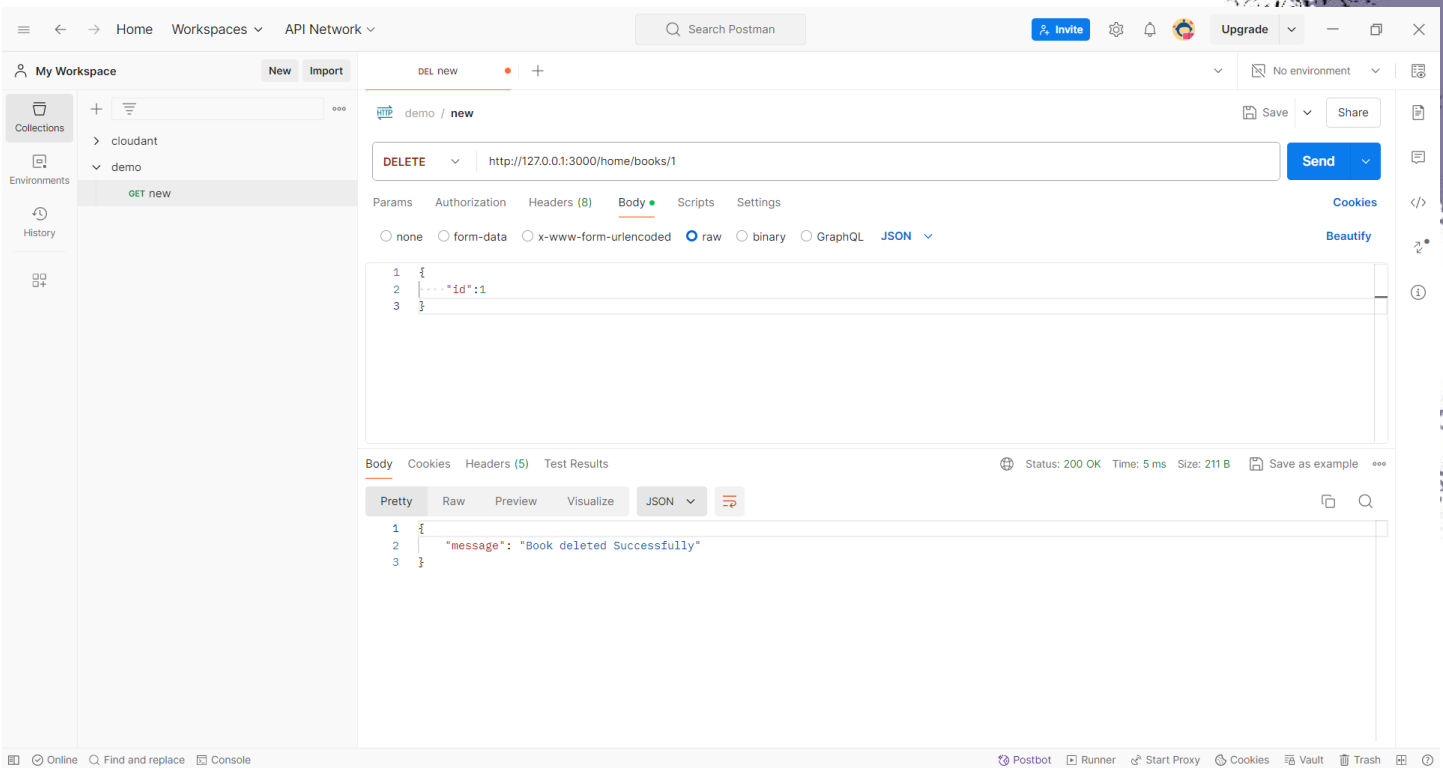
STEP 21 - Run the code and select the book which you want to change or update and copy the URL. Paste it in Postman with "PUT" method and change the data of selected book.



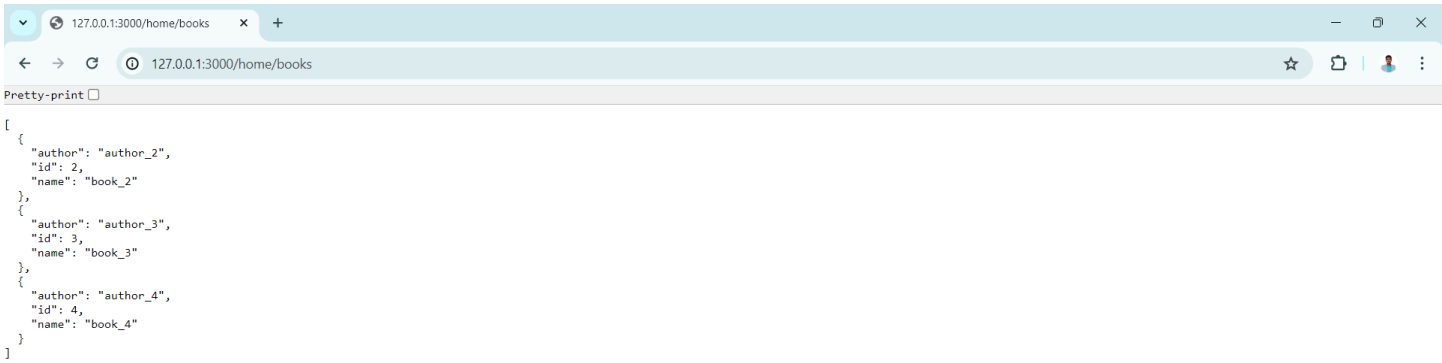
STEP 22 - For deleting an existing book data we use "DELETE" method. Create a new structure for using DELETE method by giving the condition of matching Book id of particular book data and by using remove function we delete the existing book data.



STEP 23 - Run the code and select the book which you want to delete and copy the URL. Paste it in Postman with "DELETE" method and provide the book id to it.



STEP 24 - Your book has been deleted successfully.



GIT HUB REPOSITORY LINK :

"https://github.com/shantanupandey22/HTTPS-methods/"