Assignment-4

Create an API to collect data from the URL and you have to store that data in MongoDB server.

Procedure:

Step-1:

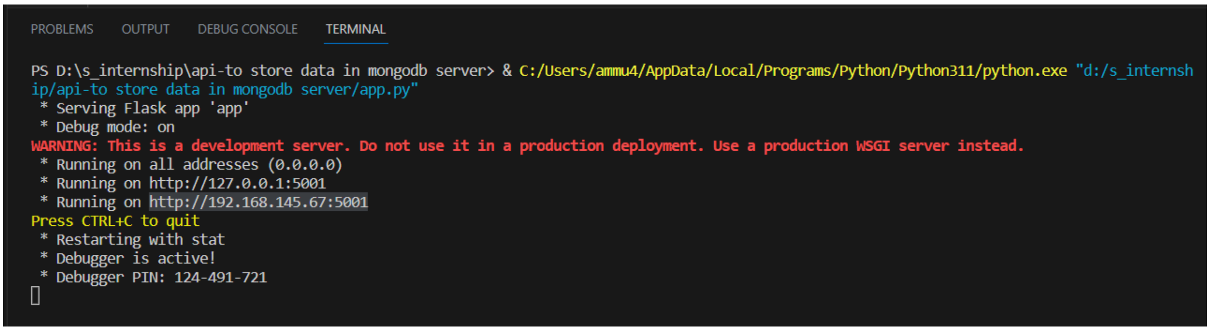
Create a folder in the respective directory.And create a file naming as “app.py”.

Step-2:

Write a python script to collect data and store the data in mongoDb server from URL.

```
from flask import Flask,request
from pymongo import MongoClient
client=MongoClient('127.0.0.1',27017)
db=client['Ammu']
collection=db['mss']
api=Flask(__name__)
@api.route('/')
def home():
    return ('API server is online,you can call APIs')
@api.route('/message',methods=['GET'])
def message():
    name=request.args.get('name')
    rollno=request.args.get('rollno')
    print(name,rollno)
    k={}
    k['name']=name
    k['rollno']=rollno
    collection.insert_one(k)
    return ('succesfully inserted into mongodb')
if(__name__)=="__main__":
    api.run(host='0.0.0.0',port=5001,debug=True)
```

now, run the code in the terminal.



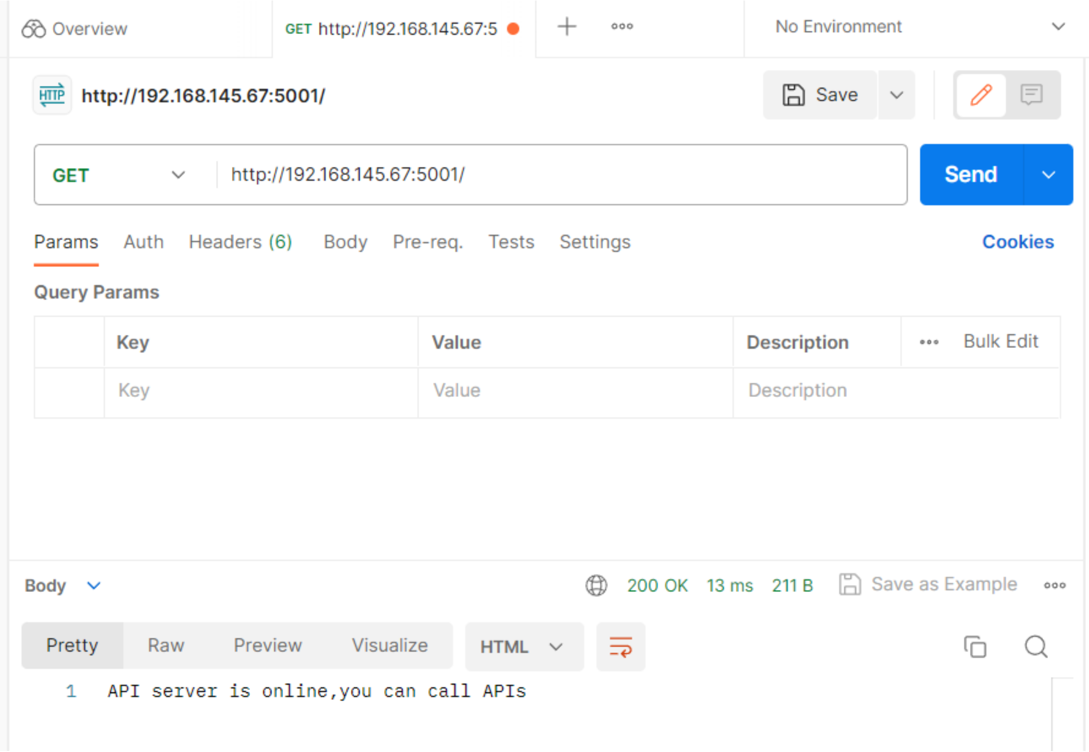
```
PS D:\s_internship\api-to store data in mongodb server> & C:/Users/ammu4/AppData/Local/Programs/Python/Python311/python.exe "d:/s_internsh
ip/api-to store data in mongodb server/app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5001
 * Running on http://192.168.145.67:5001
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 124-491-721
```

Copy the server link .

Step-3:

Open the postman tool app ,and click on + symbol.

Paste the server link and click on “send” option.



http://192.168.145.67:5001/

GET http://192.168.145.67:5001/

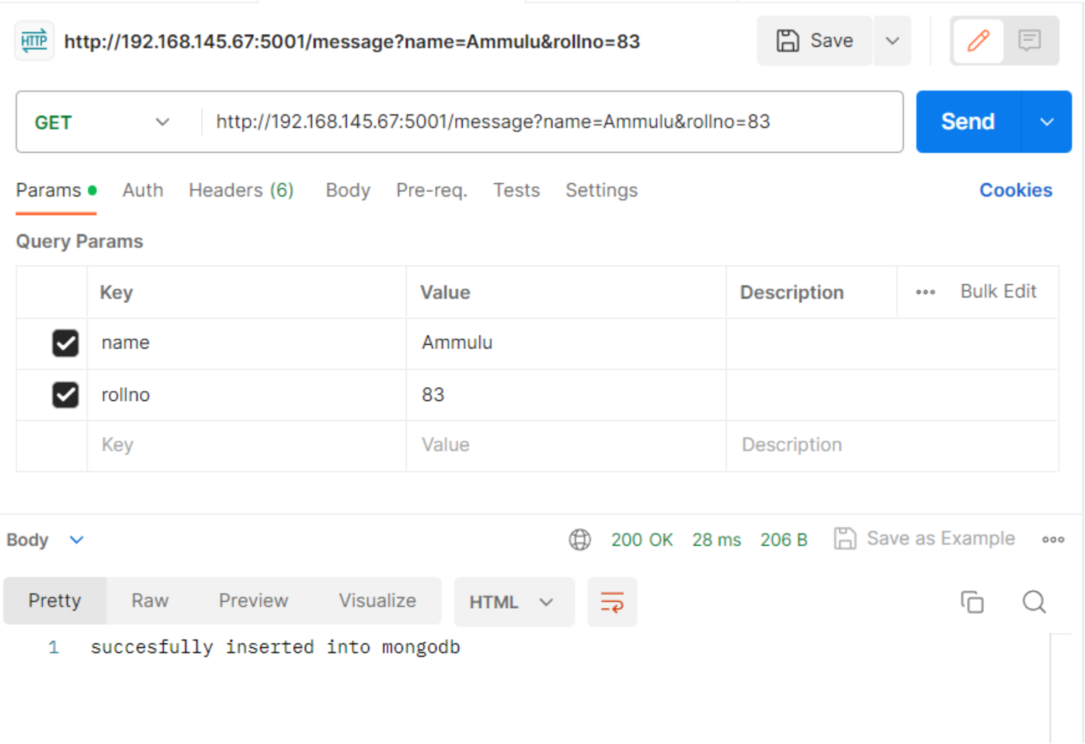
Query Params

Key	Value	Description
Key	Value	Description

200 OK 13 ms 211 B

```
API server is online,you can call APIs
```

Now ,call the “/message” handler.insert the data “key:value” format and click on send option.



http://192.168.145.67:5001/message?name=Ammulu&rollno=83

GET http://192.168.145.67:5001/message?name=Ammulu&rollno=83

Query Params

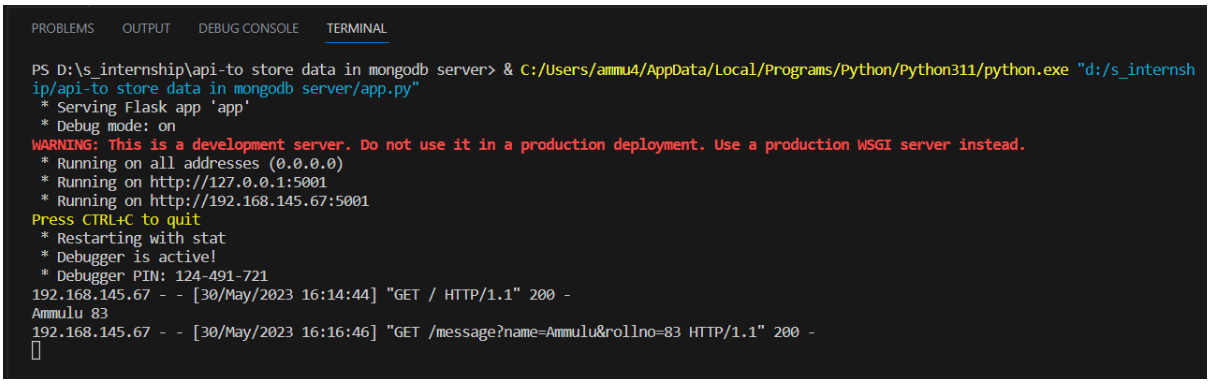
	Key	Value	Description
☑	name	Ammulu	
☑	rollno	83	
	Key	Value	Description

200 OK 28 ms 206 B

```
succesfully inserted into mongodb
```

Now the output displayed that “successfully inserted the data into MongoDB.

Now ,we also could able to see the response from the server side also.

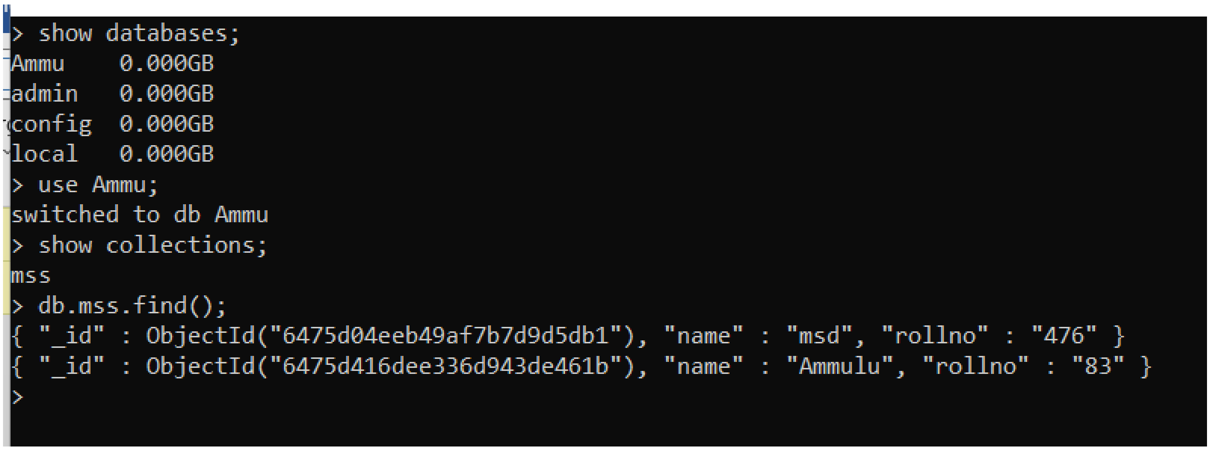


```
PS D:\s_internship\api-to store data in mongodb server> & C:/Users/ammu4/AppData/Local/Programs/Python/Python311/python.exe "d:/s_internsh
ip/api-to store data in mongodb server/app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5001
 * Running on http://192.168.145.67:5001
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 124-491-721
192.168.145.67 - - [30/May/2023 16:14:44] "GET / HTTP/1.1" 200 -
Ammulu 83
192.168.145.67 - - [30/May/2023 16:16:46] "GET /message?name=Ammulu&rollno=83 HTTP/1.1" 200 -
```

Now we should check the data that has been inserted in the mongodb server or not.

For that open the command prompt.

Use the command: db.collection_name.find() method …the terminal displays the inserted data in the collection.



```
> show databases;
Ammu     0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
> use Ammu;
switched to db Ammu
> show collections;
mss
> db.mss.find();
{ "_id" : ObjectId("6475d04eeb49af7b7d9d5db1"), "name" : "msd", "rollno" : "476" }
{ "_id" : ObjectId("6475d416dee336d943de461b"), "name" : "Ammulu", "rollno" : "83" }
>
```