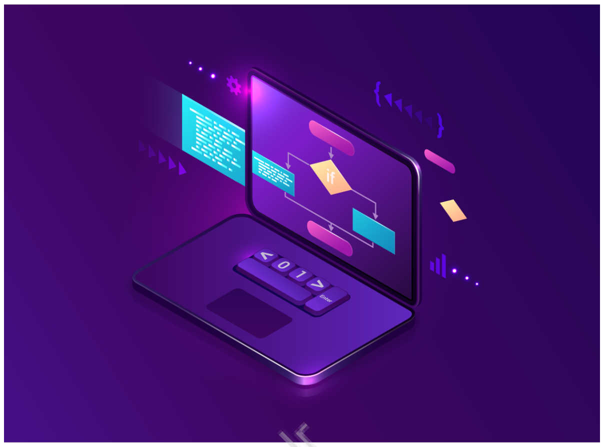
Flask



What is our GOAL for this MODULE?

The goal of this module is to explore the python flask framework to create an API.

What did we ACHIEVE in the class TODAY?

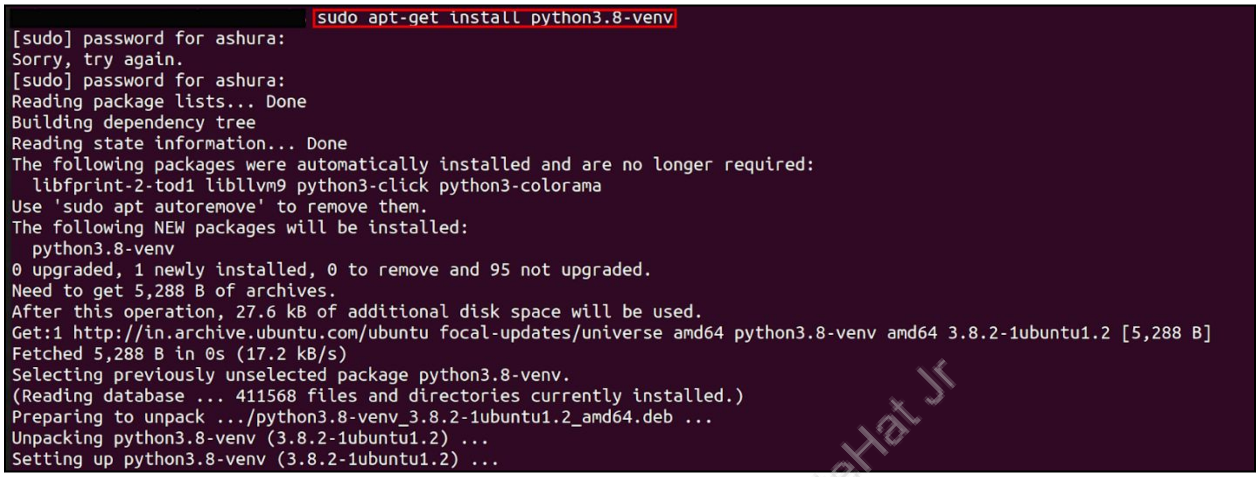
- We created an API using flask with GET and POST Methods.

Which CONCEPTS/CODING BLOCKS did we cover today?

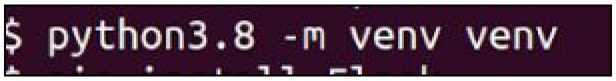
- Flask framework
- Using postman
- Creating API

How did we DO the activities?

1. We created a new folder and created a new virtual environment.

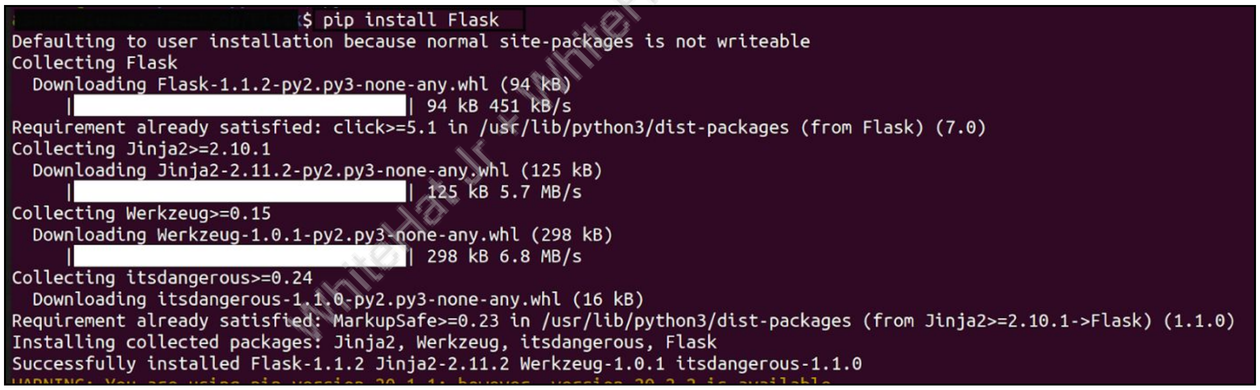


```
[redacted] sudo apt-get install python3.8-venv
[sudo] password for ashura:
Sorry, try again.
[sudo] password for ashura:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfprint-2-tod1 libllvm9 python3-click python3-colorama
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  python3.8-venv
0 upgraded, 1 newly installed, 0 to remove and 95 not upgraded.
Need to get 5,288 B of archives.
After this operation, 27.6 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python3.8-venv amd64 3.8.2-1ubuntu1.2 [5,288 B]
Fetched 5,288 B in 0s (17.2 kB/s)
Selecting previously unselected package python3.8-venv.
(Reading database ... 411568 files and directories currently installed.)
Preparing to unpack .../python3.8-venv_3.8.2-1ubuntu1.2_amd64.deb ...
Unpacking python3.8-venv (3.8.2-1ubuntu1.2) ...
Setting up python3.8-venv (3.8.2-1ubuntu1.2) ...
```



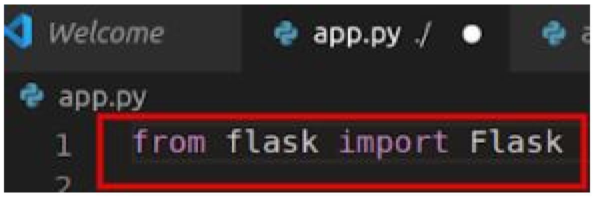
```
$ python3.8 -m venv venv
```

2. Installed Flask inside the virtual environment.

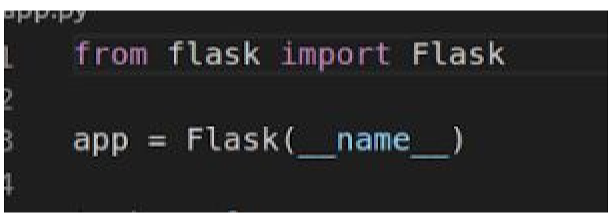


```
[redacted]$ pip install Flask
Defaulting to user installation because normal site-packages is not writeable
Collecting Flask
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 451 kB/s
Requirement already satisfied: click>=5.1 in /usr/lib/python3/dist-packages (from Flask) (7.0)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
     |████████████████████████████████| 125 kB 5.7 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
     |████████████████████████████████| 298 kB 6.8 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/lib/python3/dist-packages (from Jinja2>=2.10.1->Flask) (1.1.0)
Installing collected packages: Jinja2, Werkzeug, itsdangerous, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 Werkzeug-1.0.1 itsdangerous-1.1.0
```

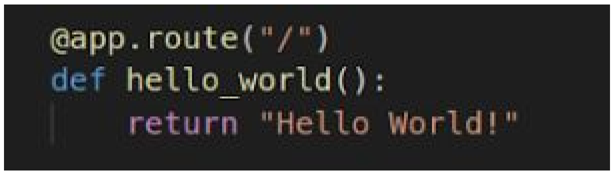
3. We created a simple hello world application.



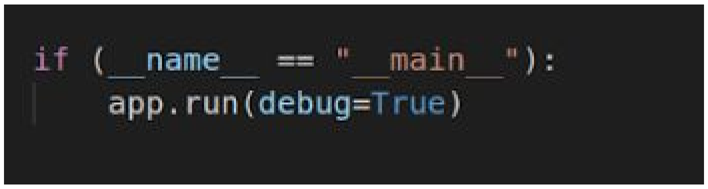
```python
from flask import Flask
```



```
app.py
from flask import Flask

app = Flask(__name__)
```

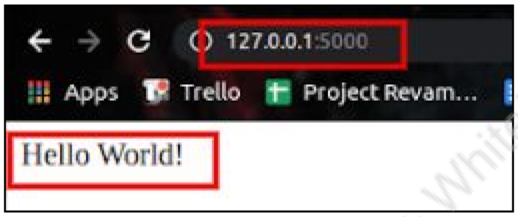


```
@app.route("/")
def hello_world():
    return "Hello World!"
```



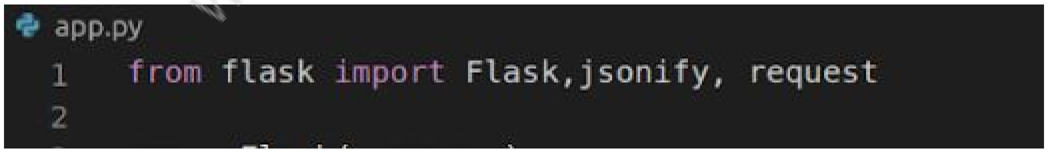
```
if (__name__ == "__main__"):
    app.run(debug=True)
```

4. We ran the code and saw the output on the web browser.

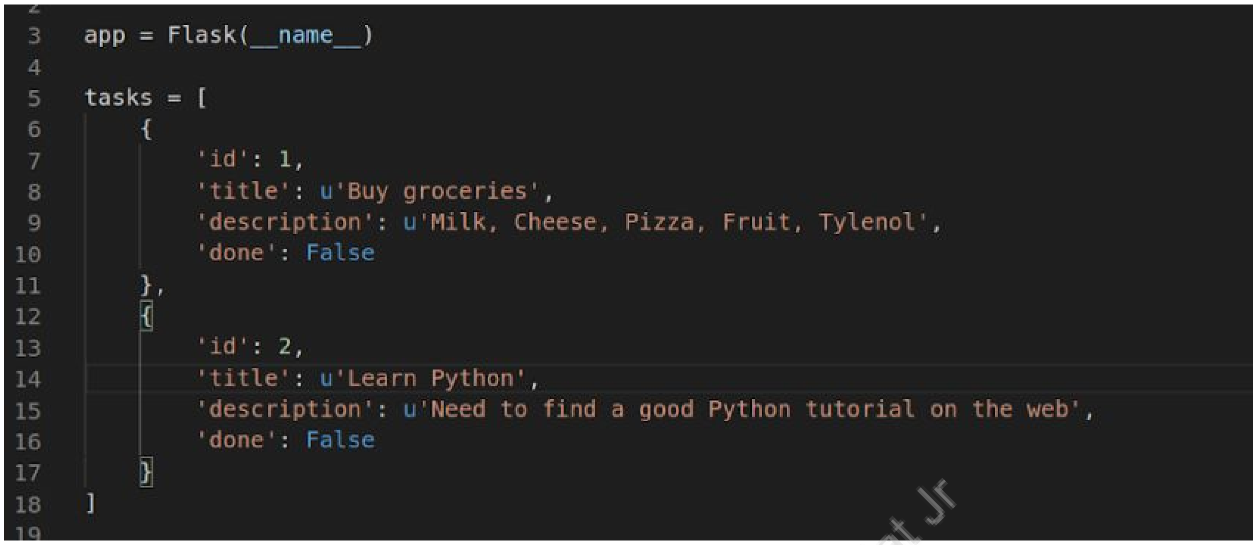


```
Hello World!
```

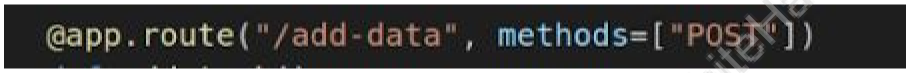
5. Then we learned about the GET ,POST, PUT and DELETE methods.
6. We also created a web application which used GET and POST methods. And allowed us to view and add tasks to the list.



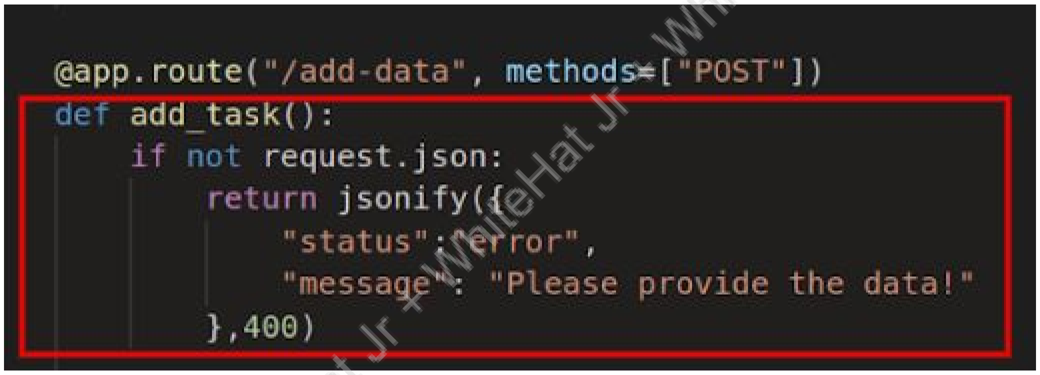
```
app.py
from flask import Flask,jsonify, request
```



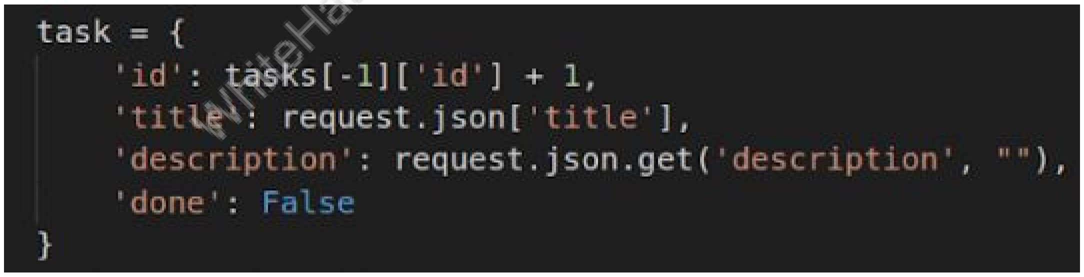
```
app = Flask(__name__)

tasks = [
    {
        'id': 1,
        'title': u'Buy groceries',
        'description': u'Milk, Cheese, Pizza, Fruit, Tylenol',
        'done': False
    },
    {
        'id': 2,
        'title': u'Learn Python',
        'description': u'Need to find a good Python tutorial on the web',
        'done': False
    }
]
```



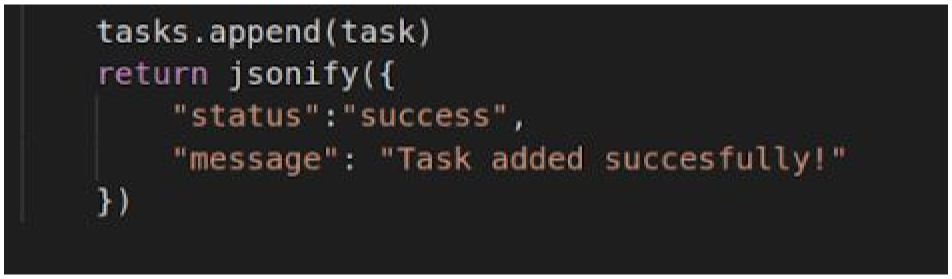
```
@app.route("/add-data", methods=["POST"])
```



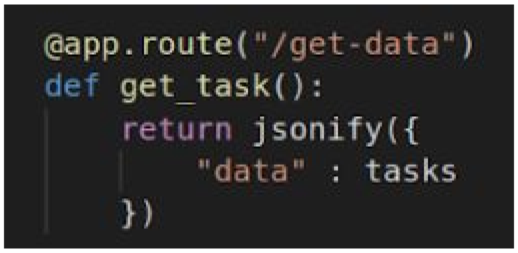
```
@app.route("/add-data", methods=["POST"])
def add_task():
    if not request.json:
        return jsonify({
            "status":"error",
            "message": "Please provide the data!"
        },400)
```



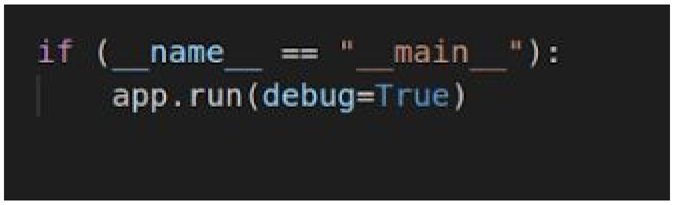
```
task = {
    'id': tasks[-1]['id'] + 1,
    'title': request.json['title'],
    'description': request.json.get('description', ""),
    'done': False
}
```



```
    tasks.append(task)
    return jsonify({
        "status":"success",
        "message": "Task added succesfully!"
    })
```

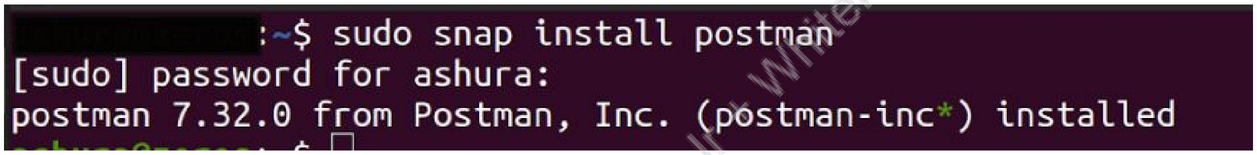


```
@app.route("/get-data")
def get_task():
    return jsonify({
        "data" : tasks
    })
```



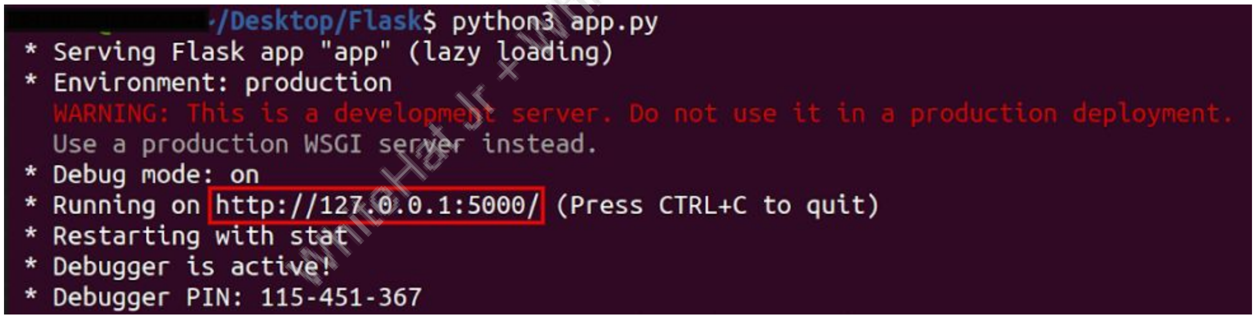
```
if (__name__ == "__main__"):
    app.run(debug=True)
```

7. We installed the postman software to test our POST method.



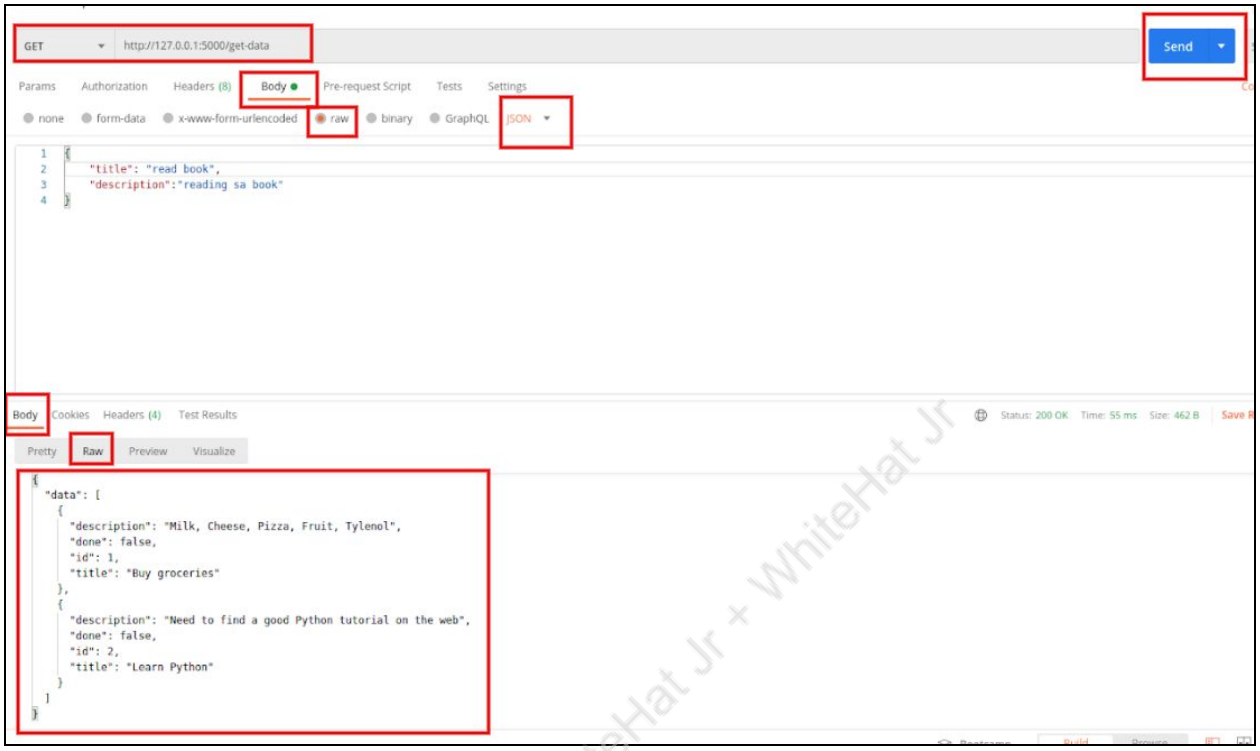
```
[redacted]:~$ sudo snap install postman
[sudo] password for ashura:
postman 7.32.0 from Postman, Inc. (postman-inc*) installed
```

8. We run the code and copied the link.



```
[redacted]~/Desktop/Flask$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 115-451-367
```

9. We tested the GET method using postman.

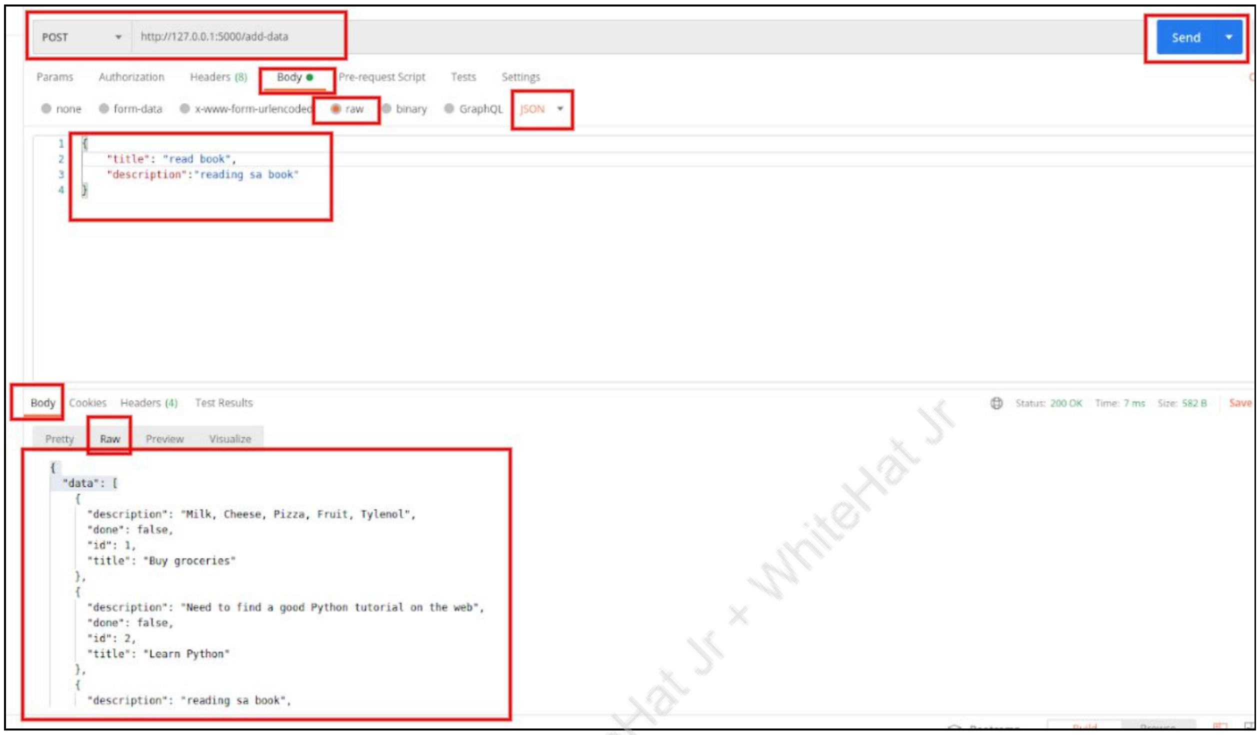


```
http://127.0.0.1:5000/get-data
```

```
{
    "title": "read book",
    "description":"reading sa book"
}
```

```
{
  "data": [
    {
      "description": "Milk, Cheese, Pizza, Fruit, Tylenol",
      "done": false,
      "id": 1,
      "title": "Buy groceries"
    },
    {
      "description": "Need to find a good Python tutorial on the web",
      "done": false,
      "id": 2,
      "title": "Learn Python"
    }
  ]
}
```

9. Then we tested the POST method.



http://127.0.0.1:5000/add-data

```
{
    "title": "read book",
    "description":"reading sa book"
}
```

Status: 200 OK Time: 7 ms Size: 582 B

```
{
  "data": [
    {
      "description": "Milk, Cheese, Pizza, Fruit, Tylenol",
      "done": false,
      "id": 1,
      "title": "Buy groceries"
    },
    {
      "description": "Need to find a good Python tutorial on the web",
      "done": false,
      "id": 2,
      "title": "Learn Python"
    },
    {
      "description": "reading sa book",
```

What's NEXT?

In the next class, we will learn to integrate Our own API with a react native app..

EXTEND YOUR KNOWLEDGE:

You can explore more about flask through the following documentation.
https://flask.palletsprojects.com/en/1.1.x/