

PROJECT NAME :News Tracker Application

TEAM ID :PNT2022TMID40290

SPRINT-3

```
# Using flask to make an api

# import necessary libraries and functions
from flask import Flask, jsonify, request

# creating a Flask app
app = Flask(__name__)

# on the terminal type: curl http://127.0.0.1:5000/

# returns hello world when we use GET.
# returns the data that we send when we use POST.
@app.route('/', methods = ['GET', 'POST'])
def home():
    if(request.method == 'GET'):

        data = "hello world"

        return jsonify({'data': data})

# A simple function to calculate the square of a number
```

```

# the number to be squared is sent in the URL when we use GET

# on the terminal type: curl http://127.0.0.1:5000 / home / 10

# this returns 100 (square of 10)

@app.route('/home/<int:num>', methods = ['GET'])

def disp(num):

    return jsonify({'data': num**2})

# driver function

if __name__ == '__main__':

    app.run(debug = True# import necessary libraries and functions

from flask import Flask, jsonify, request

# creating a Flask app

app = Flask(__name__)

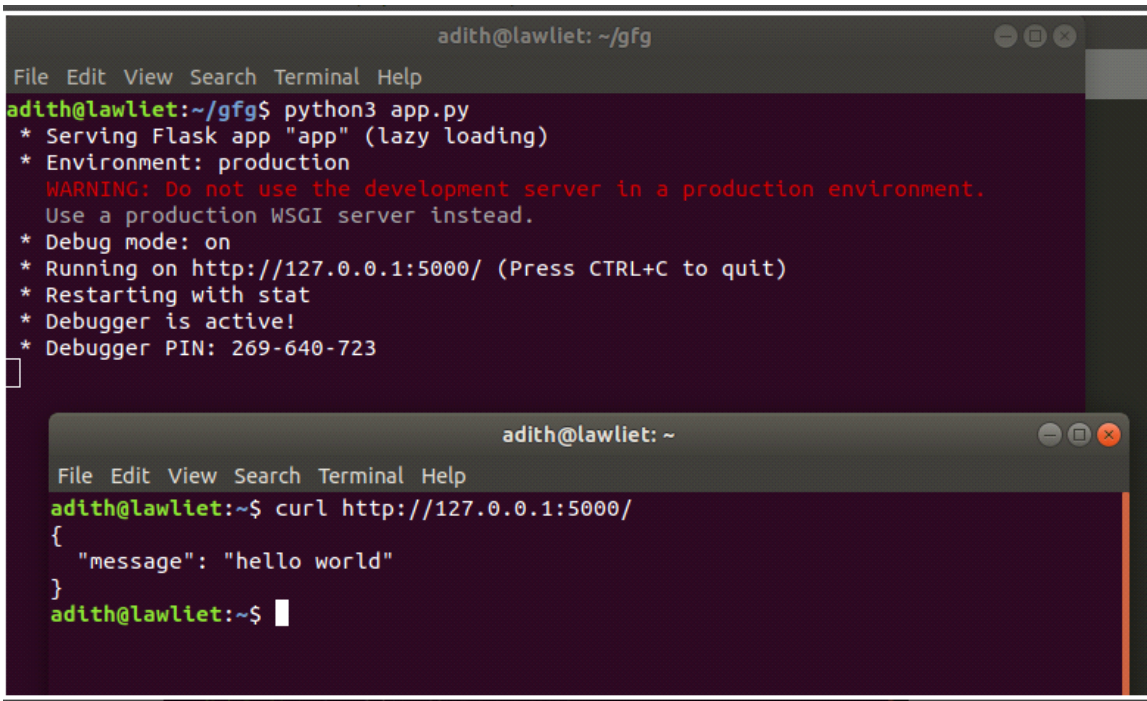
# on the terminal type: curl http://127.0.0.1:5000/

# returns hello world when we use GET.

# returns the data that we send when we use POST.

```

OUTPUT :

The image shows two terminal windows. The top window is titled 'adith@lawliet: ~/gfg' and shows the command 'python3 app.py' being executed. The output includes: '* Serving Flask app "app" (lazy loading)', '* Environment: production', a red warning 'WARNING: Do not use the development server in a production environment. 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!', and '* Debugger PIN: 269-640-723'. The bottom window is titled 'adith@lawliet: ~' and shows the command 'curl http://127.0.0.1:5000/' being executed, which returns a JSON response: '{ "message": "hello world" }'.

Method :

using flask_restful

from flask import Flask, jsonify, request

from flask_restful import Resource, Api

creating the flask app

app = Flask(__name__)

creating an API object

api = Api(app)

making a class for a particular resource

the get, post methods correspond to get and post requests

they are automatically mapped by flask_restful.

other methods include put, delete, etc.

```

class Hello(Resource):

    # corresponds to the GET request.

    # this function is called whenever there

    # is a GET request for this resource

    def get(self):

        return jsonify({'message': 'hello world'})


    # Corresponds to POST request

    def post(self):

        data = request.get_json()      # status code

        return jsonify({'data': data}), 201


# another resource to calculate the square of a number

class Square(Resource):

    def get(self, num):

        return jsonify({'square': num**2})


# adding the defined resources along with their corresponding urls

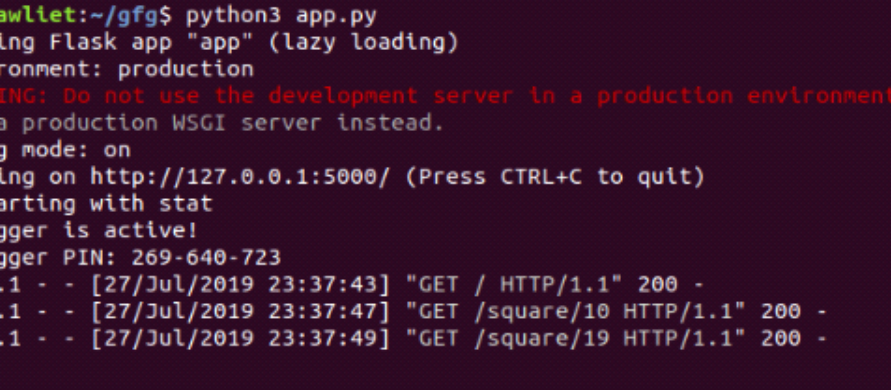
api.add_resource(Hello, '/')

```

```
# driver function
if __name__ == '__main__':

    app.run(debug = True)
```

Output:



```
adith@lawliet: ~/gfg
File Edit View Search Terminal Help
adith@lawliet:~/gfg$ python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  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: 269-640-723
127.0.0.1 - - [27/Jul/2019 23:37:43] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [27/Jul/2019 23:37:47] "GET /square/10 HTTP/1.1" 200 -
127.0.0.1 - - [27/Jul/2019 23:37:49] "GET /square/19 HTTP/1.1" 200 -

adith@lawliet: ~
File Edit View Search Terminal Help
adith@lawliet:~$ curl http://127.0.0.1:5000/
{
  "message": "hello world"
}
adith@lawliet:~$ curl http://127.0.0.1:5000/square/10
{
  "square": 100
}
adith@lawliet:~$ curl http://127.0.0.1:5000/square/19
{
  "square": 361
}
adith@lawliet:~$
```