

St. Francis Institute of Technology, Mumbai-400 103

A.Y. 2023-24

Class: SE-ITA/ITB, Semester: IV

Subject: Python Lab.

Experiment – 15: Building a simple REST API using flask.

1. **Aim:** To build a REST API using flask
2. **Prerequisite:** knowledge of the command line, Python, and web concepts .
3. **Objective:** To understand API.
4. **Requirements:** Personal Computer (PC), Windows /Linux Operating System, IDLE 3.6 for Python3.

5. Pre-Experiment Exercise: Theory:

Flask a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates.

REST API stands for Restful API that allows integrating applications or interaction with RESTful web services. It is now growing as the most common method for connecting components in a microservice architecture. APIs will enable you to get or send data to a website and perform some action to get your task done over a web service. Each website uses different types of API, like stock market trading websites integrating with Sensex or Nifty to get a current price and ups-down. Ticket booking apps use a desired single portal API to keep updated data at a familiar interface. REST stands for REpresentational State Transfer and is an architectural style used in modern web development. It defines a set of rules/constraints for a web application to send and receive data. Flask is a popular micro framework for building web applications. Since it is a micro-framework, it is very easy to use and lacks most of the advanced functionality which is found in a full-fledged framework. Therefore, building a REST API in Flask is very simple.

There are two ways of creating a REST API in Flask:

1. Using Flask without any external libraries
2. Using flask_restful library

Libraries required:

flask_restful can be installed via the pip command:

```
sudo pip3 install flask-restful
```

6. Laboratory Exercise

A. Procedure:

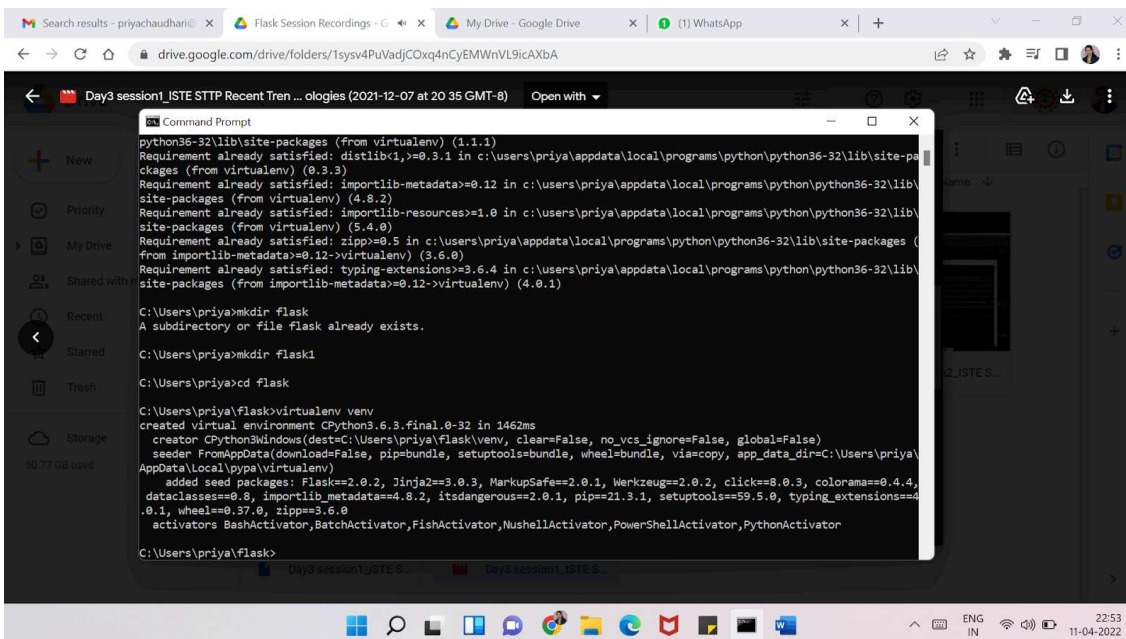
we will build a REST API in Python using the Flask framework. Flask is a popular micro framework for building web applications. Since it is a micro-framework, it is very easy to use and lacks most of the advanced functionality.

All steps of installation and execution:

a. pip install virtualenv

Create new folder for the environment

- I. mkdir flask1
- II. cd flask1
- III. virtualenv venv



```
python36-32\lib\site-packages (from virtualenv) (1.1.1)
Requirement already satisfied: distlib<1,>=0.3.1 in c:\users\priya\appdata\local\programs\python\python36-32\lib\site-pa
ckages (from virtualenv) (0.3.3)
Requirement already satisfied: importlib-metadata>=0.12 in c:\users\priya\appdata\local\programs\python\python36-32\lib\
site-packages (from virtualenv) (4.8.2)
Requirement already satisfied: importlib-resources>=1.0 in c:\users\priya\appdata\local\programs\python\python36-32\lib\
site-packages (from virtualenv) (5.4.0)
Requirement already satisfied: zipp>=0.5 in c:\users\priya\appdata\local\programs\python\python36-32\lib\site-packages (
from importlib-metadata>=0.12->virtualenv) (3.6.0)
Requirement already satisfied: typing-extensions>=3.6.4 in c:\users\priya\appdata\local\programs\python\python36-32\lib\
site-packages (from importlib-metadata>=0.12->virtualenv) (4.0.1)

C:\Users\priya>mkdir flask
A subdirectory or file flask already exists.

C:\Users\priya>mkdir flask1

C:\Users\priya>cd flask

C:\Users\priya\flask>virtualenv venv
created virtual environment CPython3.6.3.final.0-32 in 1462ms
creator CPython3Windows(dest=C:\Users\priya\flask\venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\priya\
AppData\Local\virtualenv)
added seed packages: Flask==2.0.2, Jinja2==3.0.3, MarkupSafe==2.0.1, Werkzeug==2.0.2, click==8.0.3, colorama==0.4.4,
dataclasses==0.8, importlib_metadata==4.8.2, itsdangerous==2.0.1, pip==21.3.1, setuptools==59.5.0, typing_extensions==4
.0.1, wheel==0.37.0, zipp==3.6.0
activators BashActivator,BatchActivator,FishActivator,MsShellActivator,PowerShellActivator,PythonActivator

C:\Users\priya\flask>
```

1. activate the environemnt
venv\scripts\activate

```
C:\Users\priya\flask>venv\scripts\activate

(venv) C:\Users\priya\flask>
```

1. pip install flask

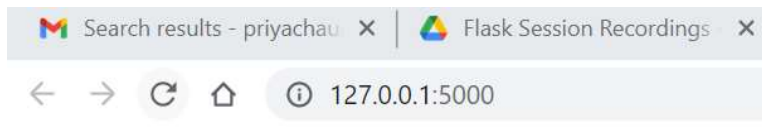
```
(venv) C:\Users\priya\flask>pip install flask
Requirement already satisfied: flask in c:\users\priya\flask\venv\lib\site-packages (2.0.2)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\priya\flask\venv\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: Werkzeug>=2.0 in c:\users\priya\flask\venv\lib\site-packages (from flask) (2.0.2)
Requirement already satisfied: click>=7.1.2 in c:\users\priya\flask\venv\lib\site-packages (from flask) (8.0.3)
Requirement already satisfied: Jinja2>=3.0 in c:\users\priya\flask\venv\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: colorama in c:\users\priya\flask\venv\lib\site-packages (from click>=7.1.2->flask) (0.4.4)
Requirement already satisfied: importlib-metadata in c:\users\priya\flask\venv\lib\site-packages (from click>=7.1.2->flask) (4.8.2)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\priya\flask\venv\lib\site-packages (from Jinja2>=3.0->flask) (2.0.1)
Requirement already satisfied: dataclasses in c:\users\priya\flask\venv\lib\site-packages (from Werkzeug>=2.0->flask) (0.8)
Requirement already satisfied: typing-extensions>=3.6.4 in c:\users\priya\flask\venv\lib\site-packages (from importlib-metadata->click>=7.1.2->flask) (4.0.1)
Requirement already satisfied: zipp>=0.5 in c:\users\priya\flask\venv\lib\site-packages (from importlib-metadata->click>=7.1.2->flask) (3.6.0)
```

- 1) Create a folder
- 2) Open python idle and import module
- 3) Create firstapp.py file.

```
firstapp.py
1  from flask import Flask
2  app = Flask(__name__)
3  @app.route('/')
4  def home():
5      return "Hello flask"
6
7  if __name__ == "__main__":
8      app.run()
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS D:\1_Jan-June 2022\Python\Programs\flask_program> python firstapp.py
* Serving Flask app 'firstapp' (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: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [11/Apr/2022 23:28:09] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Apr/2022 23:28:09] "GET /favicon.ico HTTP/1.1" 404 -
```



Hello flask

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return "Hello flask"

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

Get Started firstapp.py X

```
firstapp.py
1 from flask import Flask
2 app = Flask(__name__)
3 @app.route('/home1')
4 def home():
5     return "Hello flask"
6
7 if __name__ == "__main__":
8     app.run(debug = True)
```



Hello flask

Dynamic URLs

```
firstapp.py
1  from flask import Flask
2  app = Flask(__name__)
3  @app.route('/home1/<name>')
4  def home(name):
5      return "Hello flask "+name
6
7  if __name__ == "__main__":
8      app.run(debug = True)
```

Search results - priyachau x | Flask STTP_08_12_2021 - x |

← → ↻ 🏠 ⓘ 127.0.0.1:5000/home1/Priya

Hello flask Priya

```
firstapp.py
    from flask import Flask
    app = Flask(__name__)
    @app.route('/home1/<int:age>')
    def home(age):
        return "age: %d"%age

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

Search results - priyachau x | Flask STTP_08_12_2021 - x |

← → ↻ 🏠 ⓘ 127.0.0.1:5000/home1/34

age: 34

B. Program code with comments:

Write commands.

7. Post-Experiments Exercise

a. Extended Theory:

Understanding HTTP Request through Flask RESTApi

b. Questions/Programs:

c. Conclusion: Conclude what you understood after performing this experiment.

d. References

1. <https://towardsdatascience.com/creating-restful-apis-using-flask-and-python>
2. <https://www.analyticsvidhya.com/blog/2022/01/rest-api-with-python-and-flask/>
3. <https://www.tutorialspoint.com/flask/index.htm>

In-Lab Exercise:

1. Creating a Virtualenv

```
Windows PowerShell
PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> python -m venv venv
PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> dir

Directory: C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15

Mode                LastWriteTime         Length Name
----                -
d-----          15-04-2024         11:58         venv

PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> |
```

2. Installing Flask

```
Windows PowerShell
PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> venv/Scripts/Activate
(venv) PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
    101.7/101.7 kB 3.0 MB/s eta 0:00:00
Collecting Werkzeug>=3.0.0
  Using cached werkzeug-3.0.2-py3-none-any.whl (226 kB)
Collecting Jinja2>=3.1.2
  Using cached Jinja2-3.1.3-py3-none-any.whl (133 kB)
Collecting itsdangerous>=2.1.2
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3
  Using cached click-8.1.7-py3-none-any.whl (97 kB)
Collecting blinker>=1.6.2
  Using cached blinker-1.7.0-py3-none-any.whl (13 kB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.1.5-cp311-cp311-win_amd64.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.3 MarkupSafe-2.1.5 Werkzeug-3.0.2 blinker-1.7.0 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.1.2

[notice] A new release of pip available: 22.3 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> |
```

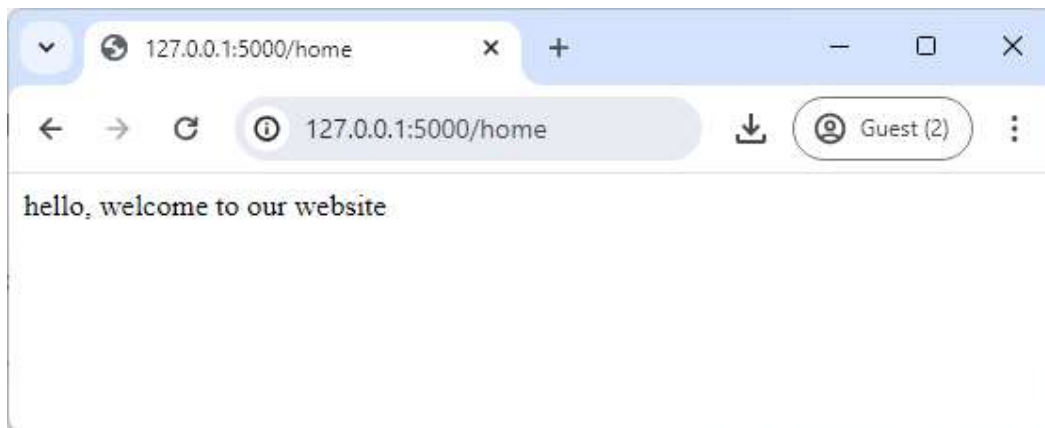

3. Creating a /home route in flask to return a basic page

```
from flask import Flask
app = Flask(__name__)

@app.route('/home')
def home():
    return "hello, welcome to our website";

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

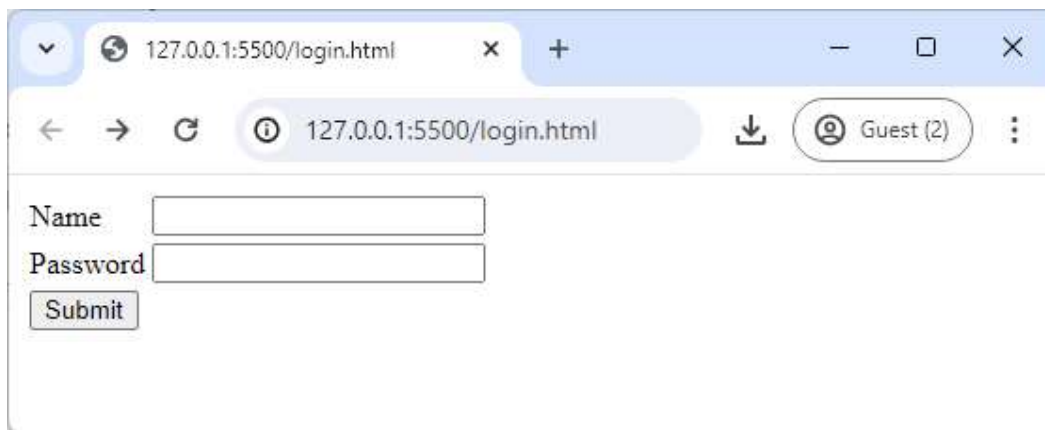
Output:



4. Creating a Login Form using HTML

```
<html>
  <body>
    <form action = "http://localhost:5000/login" method = "post">
      <table>
        <tr><td>Name</td>
        <td><input type = "text" name = "uname"></td></tr>
        <tr><td>Password</td>
        <td><input type = "password" name = "pass"></td></tr>
        <tr><td><input type = "submit"></td></tr>
      </table>
    </form>
  </body>
</html>
```

Form:



A screenshot of a web browser window showing a login form. The browser's address bar displays '127.0.0.1:5500/login.html'. The form consists of two input fields: 'Name' and 'Password', each followed by a text input box. Below these fields is a 'Submit' button. The browser's user interface includes a tab, navigation buttons (back, forward, refresh), a search bar, and a user profile icon labeled 'Guest (2)'.


5. Using a Post method to Post Form data to a Flask “\login” endpoint

```
from flask import *
app = Flask(__name__)

@app.route('/login', methods = ['Post'])
def login():
    uname=request.form['uname']
    passwd=request.form['pass']
    if uname=="Vishal" and passwd=="Mahajan":
        return "Welcome %s" %uname

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

Post Form Data [uname= Vishal and passwd = Mahajan]

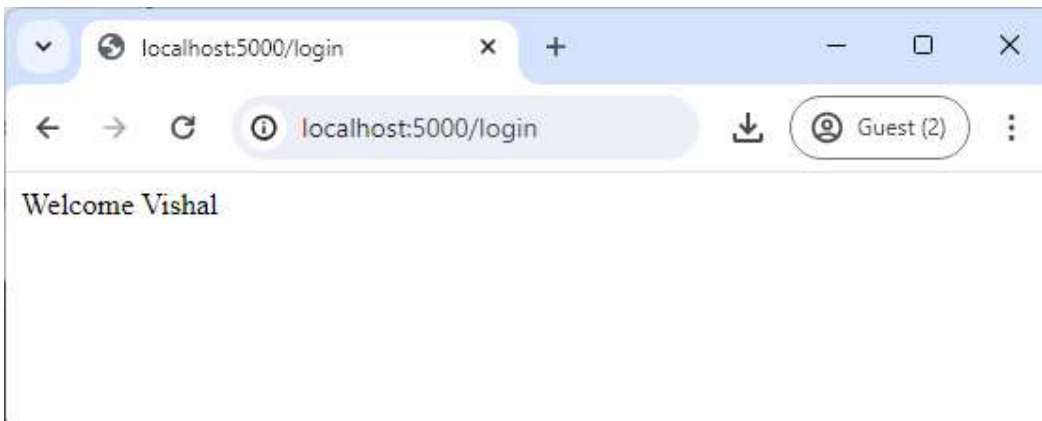


127.0.0.1:5500/login.html

Name

Password

Redirect to /login with Welcome Vishal



localhost:5000/login

Welcome Vishal

6. Installing a Flask_restx Library to for resource and API

```
Windows PowerShell
(venv) PS C:\Users\MAP\Desktop\Vishal Mahajan SEITA Python Lab\EXP15> pip install flask_re
stx
Collecting flask_restx
  Downloading flask_restx-1.3.0-py2.py3-none-any.whl (2.8 MB)
    2.8/2.8 MB 19.9 MB/s eta 0:00:00
Collecting aniso8601>=0.82
  Downloading aniso8601-9.0.1-py2.py3-none-any.whl (52 kB)
    52.8/52.8 kB ? eta 0:00:00
Collecting jsonschema
  Downloading jsonschema-4.21.1-py3-none-any.whl (85 kB)
    85.5/85.5 kB 5.0 MB/s eta 0:00:00
Requirement already satisfied: Flask!=2.0.0,>=0.8 in c:\users\map\desktop\vishal mahajan s
eita python lab\exp15\venv\lib\site-packages (from flask_restx) (3.0.3)
Requirement already satisfied: werkzeug!=2.0.0 in c:\users\map\desktop\vishal mahajan seit
a python lab\exp15\venv\lib\site-packages (from flask_restx) (3.0.2)
Collecting pytz
  Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
    505.5/505.5 kB 33.0 MB/s eta 0:00:00
Collecting importlib-resources
  Downloading importlib_resources-6.4.0-py3-none-any.whl (38 kB)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\map\desktop\vishal mahajan seita
python lab\exp15\venv\lib\site-packages (from Flask!=2.0.0,>=0.8->flask_restx) (3.1.3)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\map\desktop\vishal mahajan
```

7. Creating a API Testing Page using flask_restx (Swagger)

```
from flask import Flask, request, jsonify
from flask_restx import Resource, Api

app = Flask(__name__)

api = Api(app)

class Helloworld(Resource):

    def __init__(self):

        pass

    def get(self):

        return {

            "Hello": "World"
```

```

    }

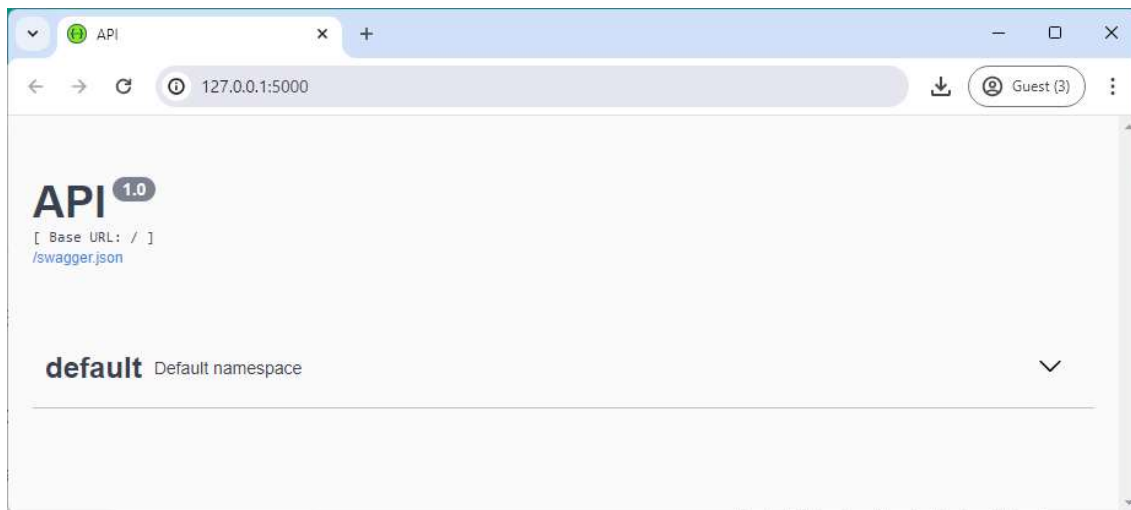
api.add_resource(Helloworld, '/')

if __name__ == '__main__':

    app.run(debug=True)

```

Output:



We can test endpoint using execute as below:

