Lab # 22

Build Your Python Flask Application

Objective:

Getting Familiar HTTP Methods and Templates.

Theory:

Flask supports multiple HTTP methods for client–server communication:

Method	Description
GET	Sends data without encryption via URL
POST	Sends form data securely (not cached)
HEAD	Same as GET but without response body
PUT	Replaces resource at a given URL
DELETE	Deletes resource from server

1. Understanding HTTP Methods in Flask

Steps to follow

Step 01: Open the terminal (Bash/CMD) and open the flask_lab folder.

- **cd flask_lab**

Step 02: Open Notepad / VS Code and write the **Program Code** and Save it.

- **notepad app.py**

```
from flask import Flask, request
app = Flask(__name__)
@app.route('/data',        methods=['GET',
'POST'])
def data():
    if request.method == 'GET':
        return "This is a GET request"
    else:
        return "This is a POST request"
if __name__ == '__main__':
    app.run(debug=True)
```

Lab # 22

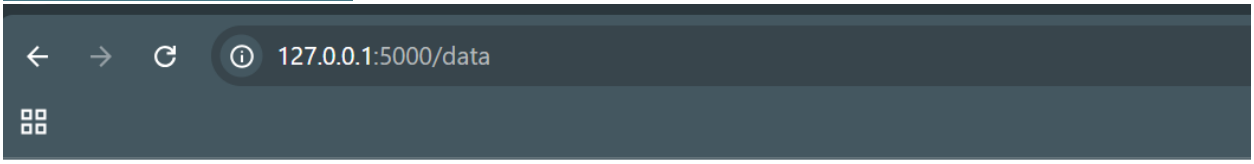
Step 03: Run the application

- **python app.py**

Note: If successful, you will see: Running on http://127.0.0.1:5000/

Step 04: Test GET Request (Browser)

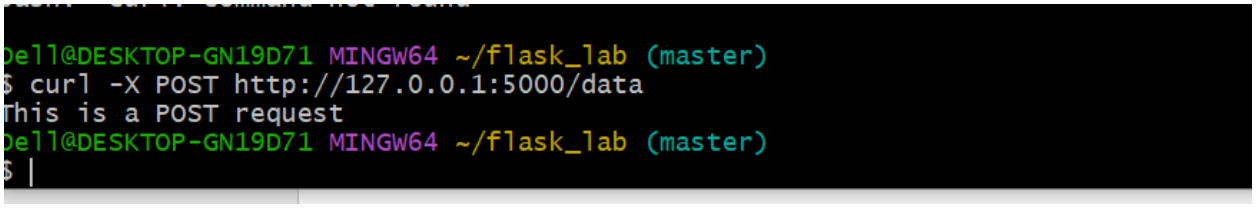
- http://127.0.0.1:5000/data



This is a GET request

Step 04: Test POST Request

1. Browsers cannot directly send POST without a form.
2. Open **another** Command Prompt / Terminal window
3. Run this command: curl -X POST http://127.0.0.1:5000/data



2. Serving HTML Templates in Flask

Flask uses a **templates** directory to store HTML files, rendered using render_template().

Directory Structure:

```
project/
|— app.py
|— templates/
|    └— index.html
```

Step 1: Created Project Folder

- **mkdir flask_template_lab**

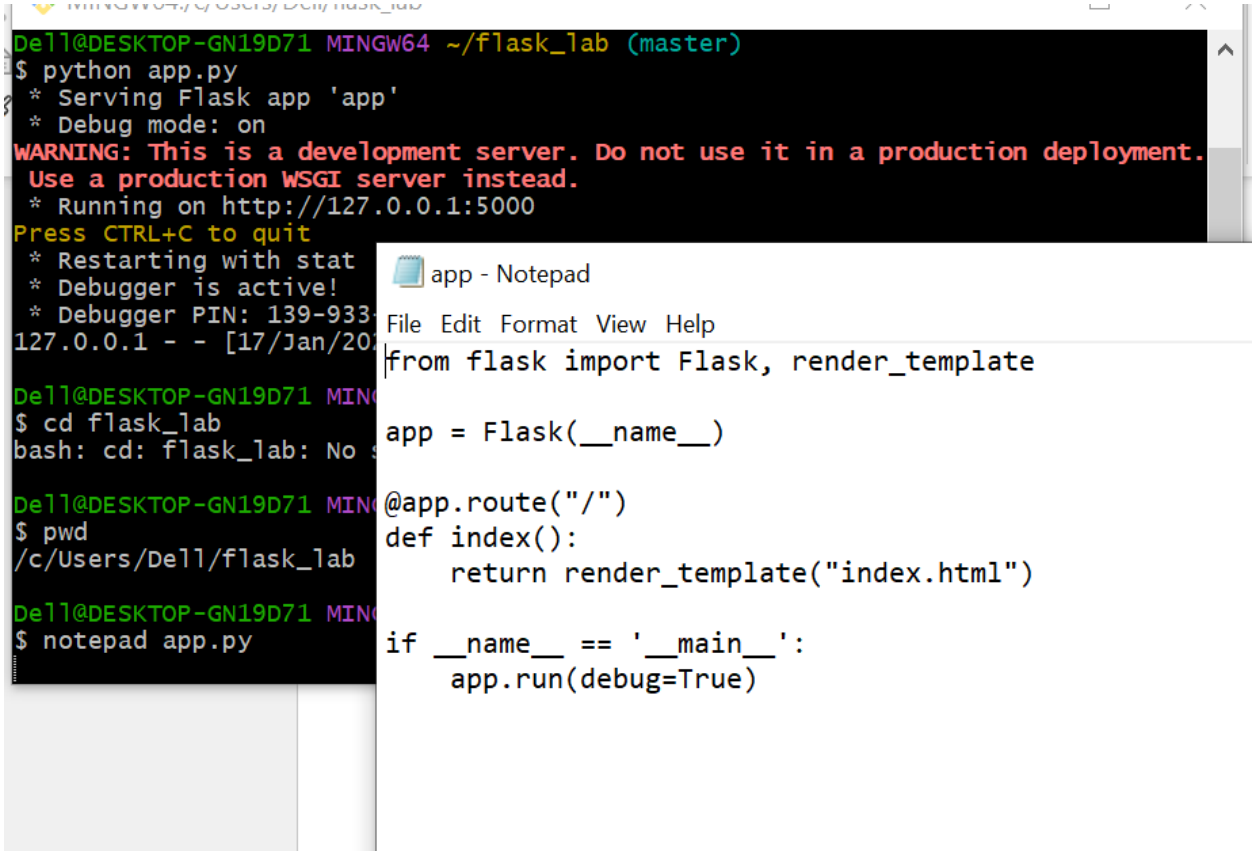
Lab # 22

Step 02: Open the Folder

- **cd flask_template_lab**

Step 03: Open **VS Code / notepad /any editor and write a Program Code, save it.**

- **notepad app.py**



Program Code

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route("/")
def index():
    return render_template("index.html")
if __name__ == '__main__':
    app.run(debug=True)
```

Step 04: Create Templates Folder inside the flask_template_lab

- **mkdir templates**

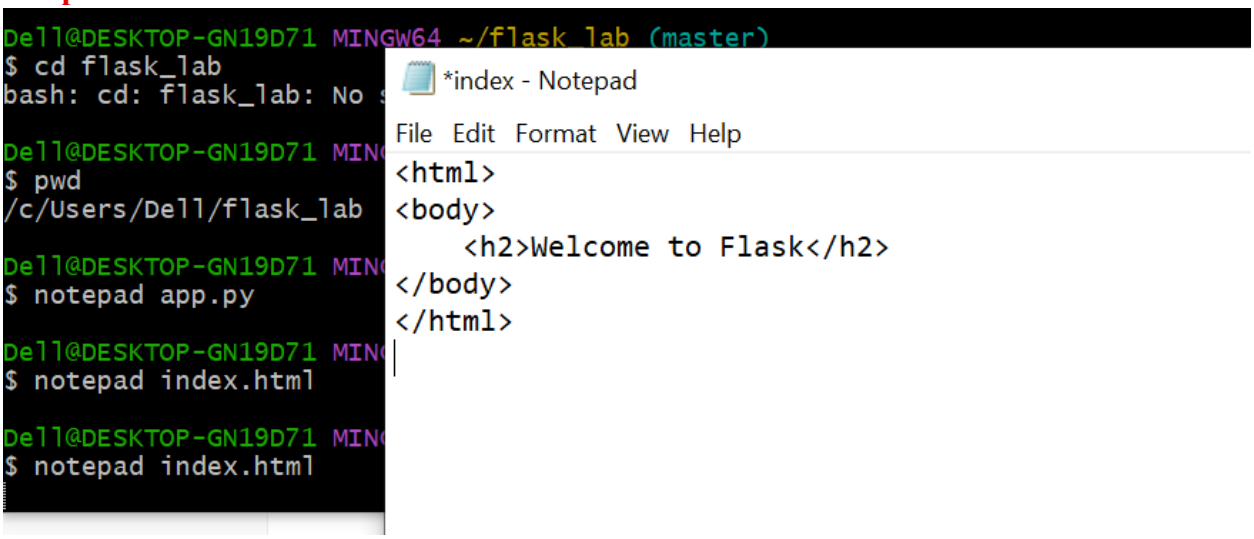
Lab # 22

Step 05: Open the templates folder

- **cd templates**

Step 06: Create a new file index.html and write a Program Code and save it.

- **notepad index.html**

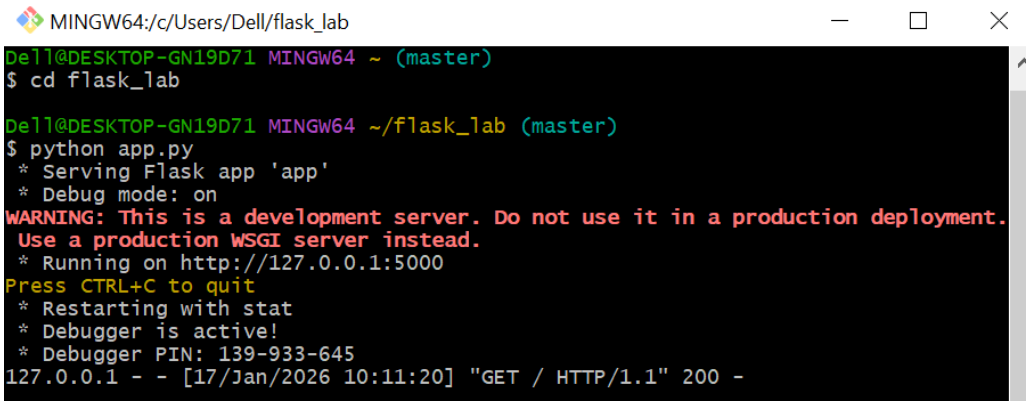


Program Code

```
<html>
<body>
   <h2>Welcome to Flask Students!</h2>
</body>
</html>
```

Step 07: Run the app

- **python app.py**



Lab # 22

Step 08: Open the Browser and visit the link:

- **http://127.0.0.1:5000/**

Output

Welcome to Flask Students!

Task:

Create your first Python Flask application that demonstrates the use of HTTP methods and HTML templates.

Your application should:

1. Use **Flask** to create a web server.
2. Create one route (/) that uses the **GET** method to display an HTML page using the **templates** folder.
3. Create one route (/data) that handles both **GET** and **POST** requests.
4. Display a different message for GET and POST requests.
5. Run the application on the default Flask server (localhost:5000).