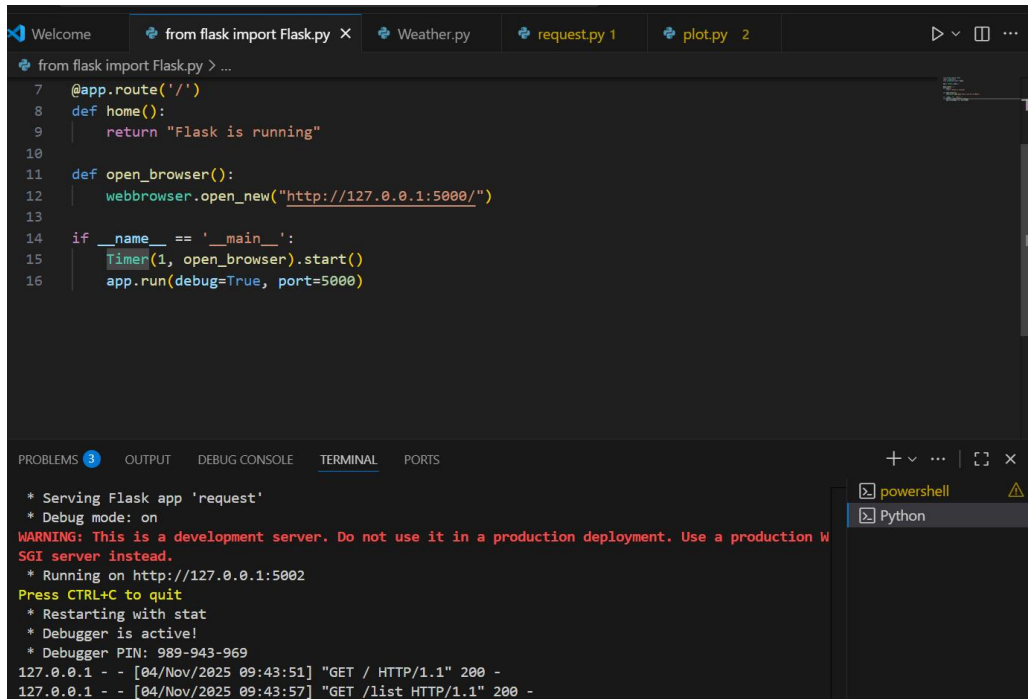


Name:S.Jewel Reddy

Reg No:22MID0161

## Flask Application:

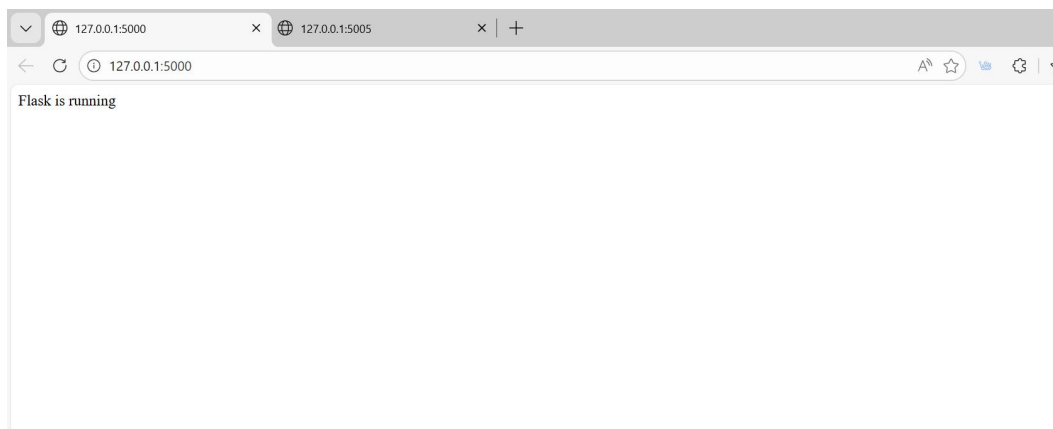


The screenshot shows a VS Code editor with a Python file named 'from flask import Flask.py'. The code defines a Flask application with a home route and a function to open a browser. The terminal output shows the application running on http://127.0.0.1:5000 and receiving two GET requests.

```
from flask import Flask > ...  
7 @app.route('/')  
8 def home():  
9     return "Flask is running"  
10  
11 def open_browser():  
12     webbrowser.open_new("http://127.0.0.1:5000/")  
13  
14 if __name__ == '__main__':  
15     Timer(1, open_browser).start()  
16     app.run(debug=True, port=5000)
```

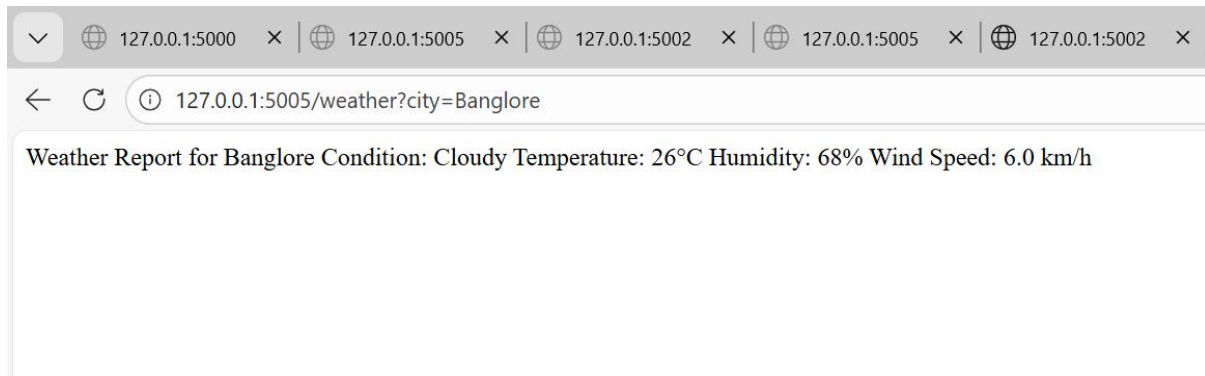
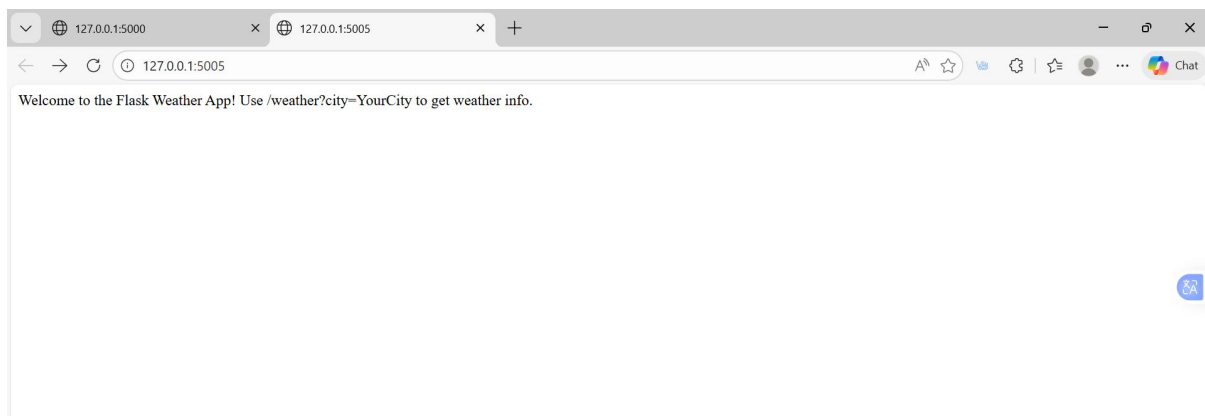
Terminal Output:

```
* Serving Flask app 'request'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production W  
SGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 989-943-969  
127.0.0.1 - - [04/Nov/2025 09:43:51] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [04/Nov/2025 09:43:57] "GET /list HTTP/1.1" 200 -
```



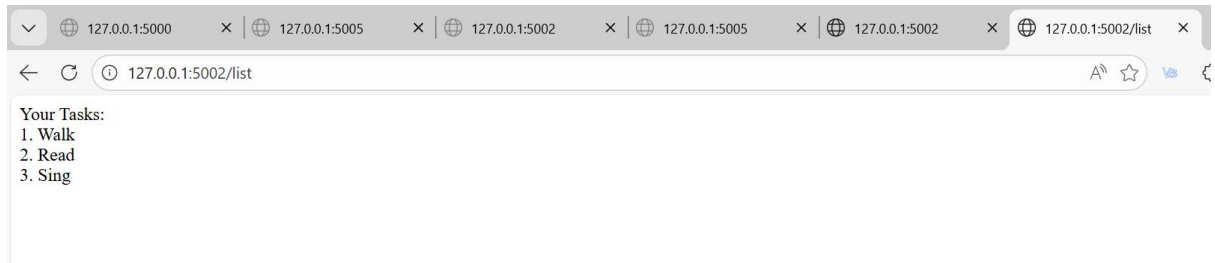
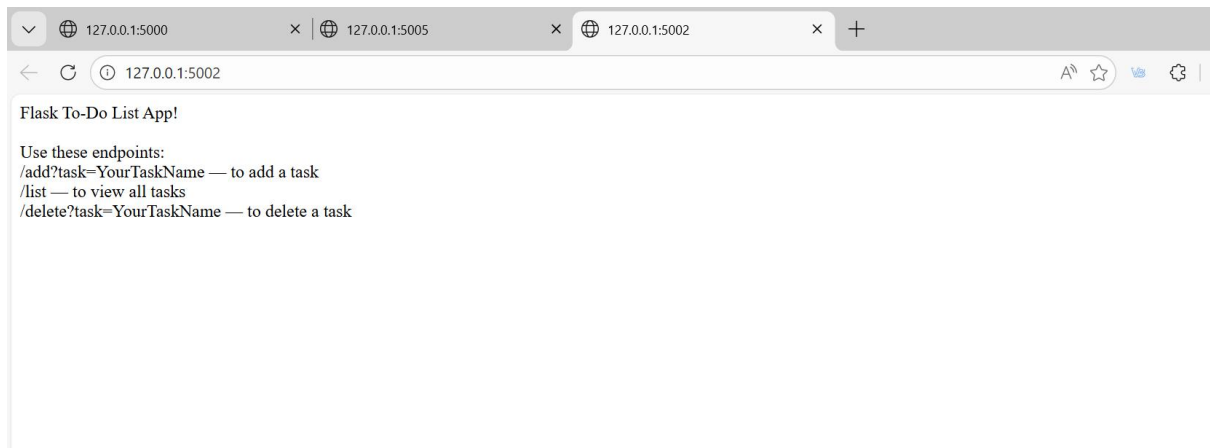
## Weather Update with Flask:

```
Welcome from flask import Flask.py 1 • Weather.py x request.py 1 plot.py 2
Weather.py > home
1 from flask import Flask, request
2 import random
3
4 app = Flask(__name__)
5
6 @app.route('/')
7 def home():
8     return "Welcome to the Flask Weather App!\n\nUse /weather?city=YourCity to get weather info."
9
10 @app.route('/weather')
11 def weather():
12     city = request.args.get('city', 'Unknown')
13     conditions = ["Sunny", "Cloudy", "Rainy", "Windy", "Stormy", "Clear Night"]
14     condition = random.choice(conditions)
15     temp = random.randint(20, 38)
16     humidity = random.randint(40, 90)
17     wind = round(random.uniform(2.0, 12.0), 1)
18
19     return (
20         f"Weather Report for {city}\n"
21         f"Condition: {condition}\n"
22         f"Temperature: {temp}°C\n"
23         f"Humidity: {humidity}%\n"
24         f"Wind Speed: {wind} km/h\n"
25     )
26
27 if __name__ == '__main__':
28     app.run(host='127.0.0.1', port=5005, debug=True)
```



Task list:

```
request.py > ...
1  from flask import Flask, request
2  app = Flask(__name__)
3  tasks = []
4
5  @app.route('/')
6  def home():
7      return (
8          "Flask To-Do List App!<br><br>"
9          "Use these endpoints:<br>"
10         "/add?task=YourTaskName - to add a task<br>"
11         "/list - to view all tasks<br>"
12         "/delete?task=YourTaskName - to delete a task"
13     )
14
15 @app.route('/add')
16 def add_task():
17     task = request.args.get('task')
18     if not task:
19         return "Please add a task using /add?task=TaskName"
20     tasks.append(task)
21     return f"Task added: {task}"
22
23 @app.route('/list')
24 def list_tasks():
25     if not tasks:
26         return "No tasks added yet!"
27     output = "<br>".join([f"{i+1}. {task}" for i, task in enumerate(tasks)])
28     return f"Your Tasks:<br>{output}"
29
30 @app.route('/delete')
31 def delete_task():
32     task = request.args.get('task')
33     if not task:
34         return "Please specify a task using /delete?task=TaskName"
35     if task in tasks:
36         tasks.remove(task)
37         return f"Task deleted: {task}"
38     else:
39         return "Task not found!"
40
41 def open_browser():
42     webbrowser.open_new("http://127.0.0.1:5002/")
43
44 if __name__ == '__main__':
```



## Plotting in Flask

```
Welcome | from flask import Flask.py 1 | Weather.py | request.py 1 | plot.py X
plot.py > ...
1  from flask import Flask, Response
2  import matplotlib.pyplot as plt
3  import io, random, webbrowser
4  from threading import Timer
5
6  app = Flask(__name__)
7
8  @app.route('/')
9  def home():
10     return (
11         "Simple Flask Data Dashboard<br><br>"
12         "Use:<br>"
13         "/plot — to view a random data chart"
14     )
15
16 @app.route('/plot')
17 def plot():
18     x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
19     y = [random.randint(10, 100) for _ in x]
20     plt.plot(x, y, marker='o')
21     plt.title("Random Data Plot")
22     plt.xlabel("X")
23     plt.ylabel("Y")
24     buf = io.BytesIO()
25     plt.savefig(buf, format='png')
26     plt.close()
27     buf.seek(0)
28     return Response(buf.getvalue(), mimetype='image/png')
29
30 def open_browser():
31     webbrowser.open_new("http://127.0.0.1:5004/")
32
33 if __name__ == '__main__':
34     Timer(1, open_browser).start()
35     app.run(debug=True, port=5004)
```

