St. Francis Institute of Technology Borivali (West), Mumbai-400103

(Autonomous Institute)
Department of Information Technology

Academic Year: 2024-25

Class: TE-ITA/B Semester: VI

Subject: Web Lab

Experiment –9: To Design a Weather App using Flask.

1. **Aim:** To design an app using Flask Framework.
2. **Objectives:** Aim of this experiment is that, the students will be able
   - To install Flask Framework
   - To understand Basics of Flask.
   - To understand Flask Application
3. **Outcomes:** After study of this experiment, the students will be able
   - To build applications.
   - To build URL
   - To understand HTTP methods.
4. **Prerequisite:** Basic understanding of HTML and Python etc
5. **Requirements:** Personal Computer, Windows operating system, VSCode, Python 2.6 or higher, browser, Internet Connection, google doc, latest version of Python.
6. **Pre-Experiment Exercise:**
   Brief Theory: Refer shared material
7. **Laboratory Exercise**
   A. **Procedure:**
      Install Python 3 on local machine
      Set up a programming environment via the command line
      Installactivate Python environment
      Install Flask using the pip package installer
      a. **Answer the following:**
         - Flask Variables and rules?
         - Flask session?
      b. **Attach screenshots:**
         - Flask SS

8. **Post-Experiments Exercise**
   A. **Extended Theory:**
      Nil
   B. **Questions:**
      - Flask applications?
   C. **Conclusion:**
      - Write what was performed in the experiment.
      - Write the significance of the topic studied in the experiment.
   D. **References:**
      1. Flask Web Development, by Miguel Grinberg

Answer the following:

- ## Flask Variables and rules?

In Flask, variables and rules are mainly used in URL routing:

➤ URL Variables (Dynamic Routing)

You can pass variables directly through the URL.

EXAMPLE:

```
@app.route('/user/<username>')
def show_user(username):
    return f'Hello {username}!'
```

Types of variables you can specify:

<string:variable> – Default

<int:variable> – Only integers

<float:variable> – Floating point numbers

<path:variable> – Accepts slashes

<uuid:variable> – UUIDs

EXAMPLE:

```
@app.route('/post/<int:post_id>')
def show_post(post_id):
    return f'Post #{post_id}'
```

- ## Flask session?

session is used to store data across requests. It behaves like a dictionary and stores info per user (client-side).

Example:

```
from flask import Flask, session
app = Flask(__name__)
app.secret_key = 'your_secret_key'  # Required for using sessions

@app.route('/login')
def login():
    session['username'] = 'tanmay'
    return 'Logged in!'

@app.route('/profile')
def profile():
    user = session.get('username')
    return f'Welcome {user}'

@app.route('/logout')
def logout():
    session.pop('username', None)
    return 'Logged out!'
```

Important Points:

Data is stored in a cookie (securely signed).

Use session['key'] = value to set.

Use session.get('key') to access.

Use session.pop('key') to remove.

OUTPUT & CODE:

app.py:

```
from flask import Flask, render_template, request
import requests

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == "POST":
        city = request.form['city']
        country = request.form['country']
        api_key = "01e9ba45f7b480c4549afe07e6d5c885"

        # Make the API request
        response = requests.get(

f'https://api.openweathermap.org/data/2.5/weather?q={city},{country}&appid={api_key}&units=imperial'
        )
        weather_data = response.json()

        # Check if the response contains the expected data
        if response.status_code != 200 or "main" not in weather_data:
            error_message = weather_data.get("message", "Error fetching weather data.")
            return render_template("result.html", error=error_message, city=city)

        # Extract the data if available
        temp = round(weather_data["main"]['temp'])
        humidity = weather_data["main"]['humidity']
        wind_speed = weather_data['wind']['speed']

        return render_template("result.html", temp=temp, humidity=humidity,
                               wind_speed=wind_speed, city=city)
    return render_template("index.html")

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

Index.html:

```
{% block content %}
<div style="background-color: #f5f5f5; height: 100vh;">
    <div class="container h-100">
        <div class="row justify-content-center">
            <div class="col-md-6 col-sm-12 col-xs-12 pt-5">
                <form method="POST" class="card border-0 p-3 shadow-sm">
                    <div class="form-group">
                        <input type="text" class="form-control" name="city" placeholder="Mumbai">
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control" name="country" placeholder="IN">
                    </div>
                    <div class="form-group">
```

```
                <button class="btn btn-info btn-block">Get Weather</button>
              </div>
            </form>
          </div>
        </div>
    </div>
</div>
{% endblock content %}
```

---

result.html

```
{% block content %}
<div style="background-color: #3deaf4; height: 100vh">
    <div class="container h-100">
        <div class="row justify-content-center">
            <div class="col-md-6 col-sm-12 col-xs-12 pt-5">
                <div class="card border-0 p-3 shadow-sm text-center">
                    <h3 class="text-muted">Current Weather For {{ city }}</h3>
                    <h5 class="lead text-muted">Temperature: {{ temp }} F</h5>
                    <h5 class="lead text-muted">Humidity: {{ humidity }}%</h5>
                    <h5 class="lead text-muted">Wind Speed: {{ wind_speed }} mph</h5>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock content %}
```