Name : Madhura Jangale

Class : D15A

Roll no. : 20

EXPERIMENT NO. 5

AIM : To create a Flask application that demonstrates template rendering by dynamically generating HTML content using the render_template() function.

PROBLEM STATEMENT :

Develop a Flask application that includes:

1. A homepage route (/) displaying a welcome message with links to additional pages.
2. A dynamic route (/user/<username>) that renders an HTML template with a personalized greeting.
3. Use Jinja2 templating features, such as variables and control structures, to enhance the templates.

Theory:

1.What does the render_template() function do in a Flask application?

The render_template() function in Flask is used to render HTML templates by combining Python variables and logic with Jinja2 syntax. It dynamically generates HTML content based on the provided template and data.

Eg:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template("index.html", title="Welcome Page")

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

Here, render_template("index.html", title="Welcome Page") loads the index.html file and passes the variable title to it.

2. What is the significance of the templates folder in a Flask project?

The templates folder is a special directory in a Flask project where all HTML template files are stored. Flask automatically looks for templates inside this folder when using render_template().

It is important because-

- Keeps HTML files separate from Python code (better organization).
- Supports Jinja2 templating for dynamic content.
- Allows template inheritance to avoid code duplication.

3. What is Jinja2, and how does it integrate with Flask?

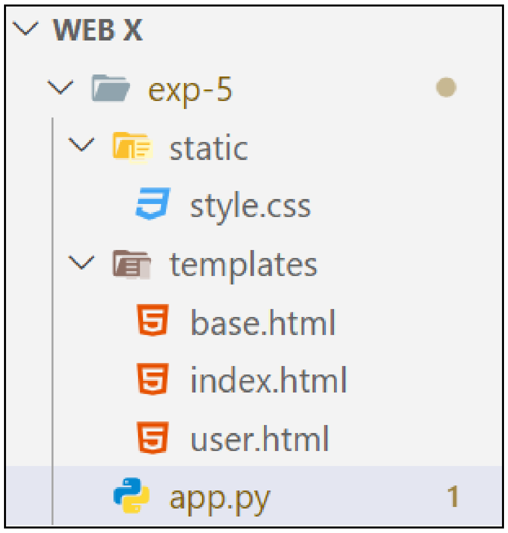
Jinja2 is a templating engine used in Flask to dynamically generate HTML pages. It allows embedding Python-like logic inside templates using special syntax.

Integration with Flask:

Flask automatically uses Jinja2 when rendering templates with render_template(). Jinja2 enables:

- Variables ({{ username }})
- Control structures ({% for item in list %})
- Template inheritance (extends and block)
- Filters ({{ name | upper }} to convert text to uppercase)

Implementation:



```
WEB X
  exp-5
    static
      style.css
    templates
      base.html
      index.html
      user.html
    app.py                1
```

app.py

```
from flask import Flask, render_template
app = Flask(__name__)
# Sample data
users = ["Madhura", "Shravani", "Rujuta"]

@app.route('/')
def home():
    return render_template("index.html", users=users)

@app.route('/user/<username>')
def user(username):
    user_details = {
        "Madhura": {"age": 20, "city": "Dombivli"},
        "Shravani": {"age": 20, "city": "Panvel"},
        "Rujuta": {"age": 20, "city": "Dombivli"},

    }
    details = user_details.get(username, None)
    return render_template("user.html", username=username, details=details)

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

Base.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <title>{% block title %}Flask App{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <div class="container">
        <div class="nav">
            <a href="/">Home</a>
        </div>
        {% block content %}{% endblock %}
    </div>
</body>
</html>
```

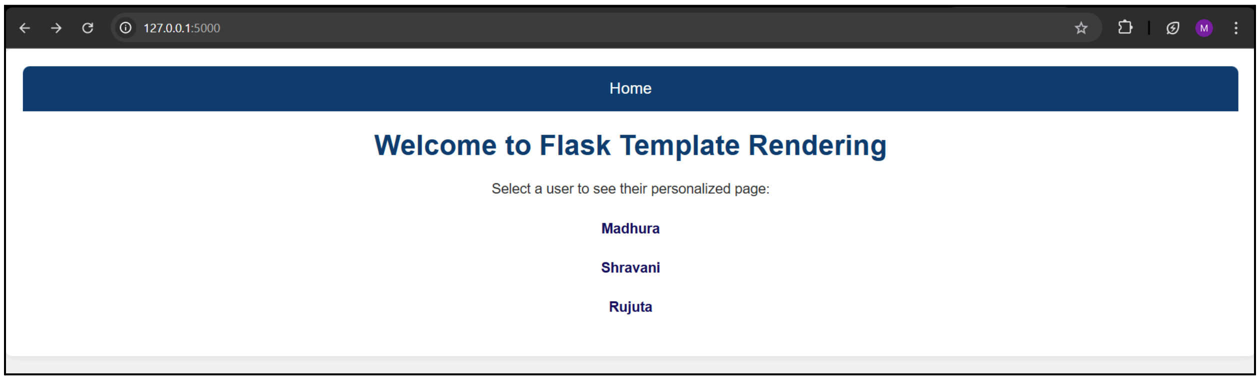
User.html

```
{% extends "base.html" %}
{% block title %}User - {{ username }}{% endblock %}
{% block content %}
    <h1>Hello, {{ username }}!</h1>

    <div class="user-details">
        {% if details %}
            <p><strong>Age:</strong> {{ details.age }}</p>
            <p><strong>City:</strong> {{ details.city }}</p>
        {% else %}
            <p>User details not found.</p>
        {% endif %}
    </div>

    <a href="/" class="button">Go back to Home</a>
{% endblock %}
```

Index.html

```
{% extends "base.html" %}
{% block title %}Home{% endblock %}
{% block content %}
    <h1>Welcome to Flask Template Rendering</h1>
    <p style="text-align: center;">Select a user to see their personalized page:</p>
    <ul>
        {% for user in users %}
            <li><a href="/user/{{ user }}">{{ user }}</a></li>
        {% endfor %}
    </ul>
{% endblock %}
```

Output:



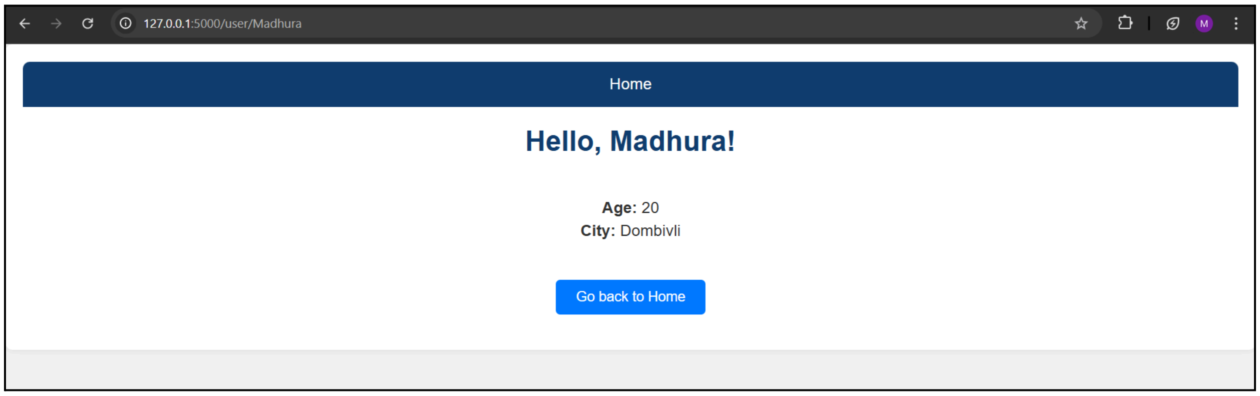
Welcome to Flask Template Rendering

Select a user to see their personalized page:

Madhura

Shravani

Rujuta



Hello, Madhura!

Age: 20
City: Dombivli