

EXPERIMENT NO. 5

Name of Student	SHRAVANI S RASAM
Class Roll No	D15A 45
D.O.P.	06\02\2025
D.O.S.	
Sign and Grade	

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 an HTML file (template) and return it as the response to a user's request. It allows you to separate the application logic from the presentation by using HTML templates with dynamic content.

Example:

```
from flask import render_template
```

```
@app.route('/')
def home():
    return render_template('index.html')
```

In this example, Flask will search for the `index.html` file in the templates folder, render it, and return it as the response.

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

The templates folder is where Flask looks for HTML templates that are rendered using the `render_template()` function. It is a special folder in Flask, and any template files (such as `.html`) should be placed inside it. Flask automatically knows to search for templates in this folder by default.

Example structure:

```
your_project/  
  app.py  
  templates/  
    Index.html  
  static/  
    style.css
```

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

Jinja2 is a modern and fast templating engine for Python, which is used in Flask to render dynamic HTML templates. It allows you to embed Python-like expressions and logic within the HTML files, such as loops, conditionals, and variable replacements. Jinja2 is integrated with Flask by default, and Flask automatically uses it to render templates.

Example of Jinja2 in a template:

```
<h1>Welcome, {{ username }}!</h1>
```

In this example, `{{ username }}` is a placeholder that Flask replaces with the actual value passed to the template when calling `render_template()`.

Example of passing variables to Jinja2 template:

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

In this case, the template will render `Welcome, Shravani!` instead of `{{ username }}`.

Github: <https://github.com/ShravaniR2412/WebX-Exp5>

OUTPUT:

Welcome to the Homepage

This is the main page of the Flask app. You can navigate to other pages using the links below:

[Visit Shravani's Page](#)

[Visit Janhavi's Page](#)

[Visit Darshan's Page](#)

Hello, Shravani!

Welcome, Shravani! We hope you're ready for a fun adventure today.

[Back to Homepage](#)

Hello, Janhavi!

Hello Janhavi! You have some new messages waiting for you.

[Back to Homepage](#)

Hello, Darshan!

Darshan, you are our star! Your achievements are impressive.

[Back to Homepage](#)