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**.
It takes an HTML file (usually stored in the templates/ folder) and **injects dynamic data** into it using **Jinja2 templating**.

- It returns a rendered HTML page to the user's browser.
- You can pass Python variables (like strings, lists, etc.) to the HTML template using it.

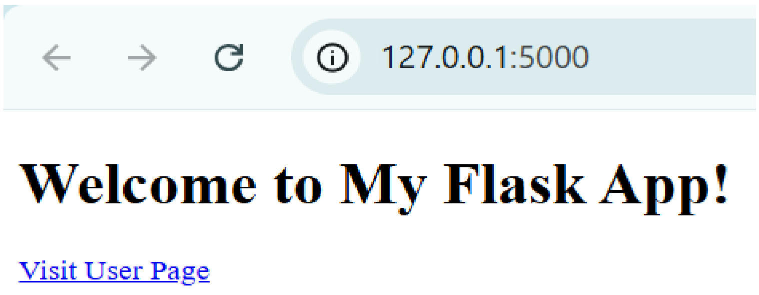
2. What is the significance of the `templates` folder in a Flask project?
Flask **automatically looks** for HTML files inside a folder called templates.
This folder is the **default location** for all the templates used with render_template().

- Keeps the project organized.
- Allows Flask to quickly locate and serve HTML content.

3. What is Jinja2, and how does it integrate with Flask?
Jinja2 is the templating engine used by Flask to generate dynamic HTML pages.
It allows you to embed Python-like expressions and logic directly into your HTML.

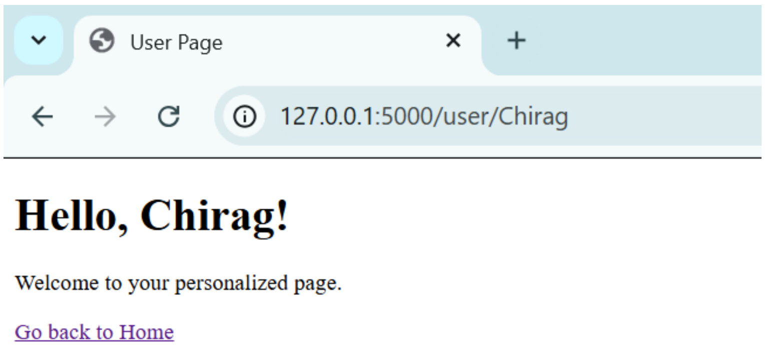
- When you call render_template(), Flask uses Jinja2 to process the HTML file.
- You can use variables, loops, conditionals, filters, etc.

Output:-



Welcome to My Flask App!

Visit User Page



Hello, Chirag!

Welcome to your personalized page.

Go back to Home