**Experiment - 5: Flask Application**

|  |  |
| --- | --- |
| **Name of Student** | **AKRUTI DABAS** |
| **Class Roll No** | **D15A / 11** |
| **D.O.P.** |  |
| **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 is used to render HTML templates stored in the templates folder. It dynamically generates web pages by passing variables from the Flask app to the template using Jinja2.

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

* The templates folder is the default location where Flask looks for HTML files.
* It maintains a clean separation between business logic (Python code) and presentation logic (HTML).
* Using the templates folder allows developers to use Jinja2 for rendering dynamic content.
* The folder can also store reusable components like base templates, headers, or footers using template inheritance.

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

Jinja2 is a templating engine used in Flask to render dynamic HTML content. It allows embedding Python expressions inside HTML files. Using Jinja2, you can:

* Display variables
* Apply logic (like loops and conditionals)
* Apply filters for formatting

Flask integrates Jinja2 by default using the render\_template() function.

# OUTPUT

* **app.py**

from flask import Flask, render\_template

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('home.html')

@app.route('/user/<username>')

def user\_profile(username):

return render\_template('user.html', username=username)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

* **base.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}Flask WebApp{% endblock %}</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

<link rel="stylesheet" href="{{ url\_for('static', filename='styles.css') }}">

</head>

<body>

<nav class="navbar navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand mx-auto" href="/">Flask WebApp</a>

</div>

</nav>

<div class="container mt-5">

{% block content %}{% endblock %}

</div>

</body>

</html>

* **home.html**

{% extends 'base.html' %}

{% block title %}Home - Flask WebApp{% endblock %}

{% block content %}

<div class="text-center">

<h1 class="display-4">Welcome to My Flask Web Application</h1>

<p class="lead">Explore the site and interact with user profiles.</p>

<div class="mt-4">

<a href="/user/Sannidhi" class="btn btn-warning btn-lg">Sannidhi's Profile</a>

<a href="/user/Guest" class="btn btn-success btn-lg">Guest Profile</a>

</div>

</div>

{% endblock %}

* **user.html**

{% extends 'base.html' %}

{% block title %}Profile - {{ username }}{% endblock %}

{% block content %}

<div class="text-center">

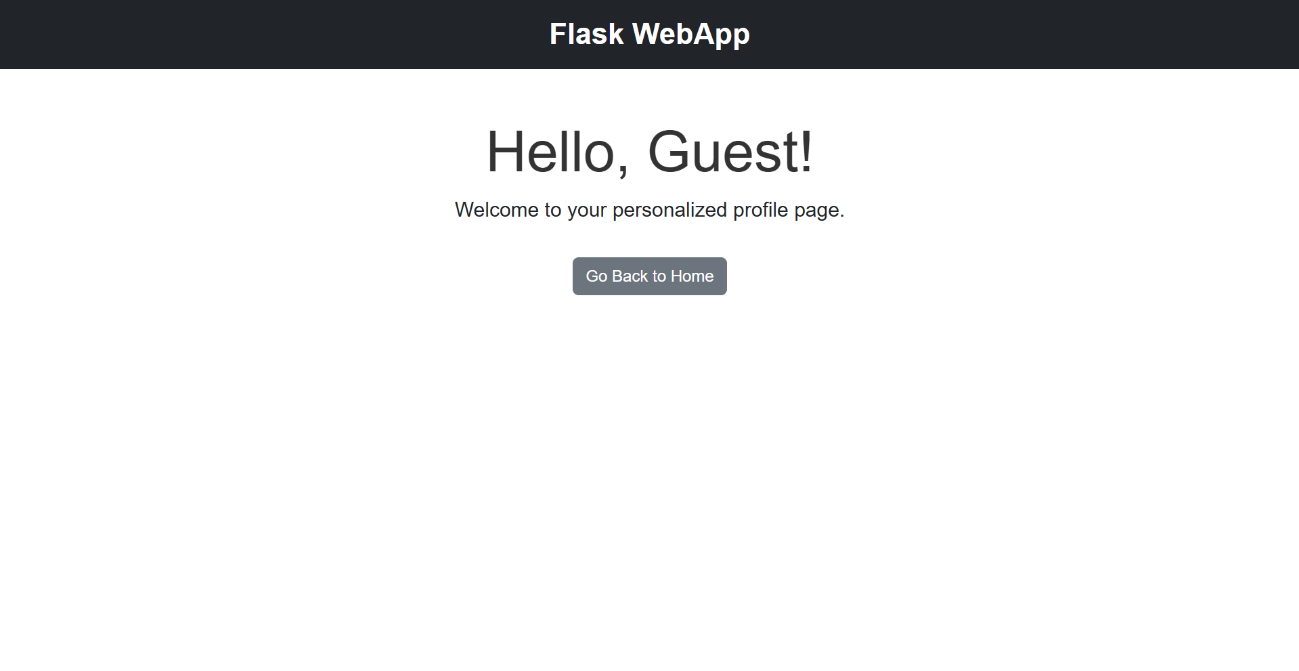
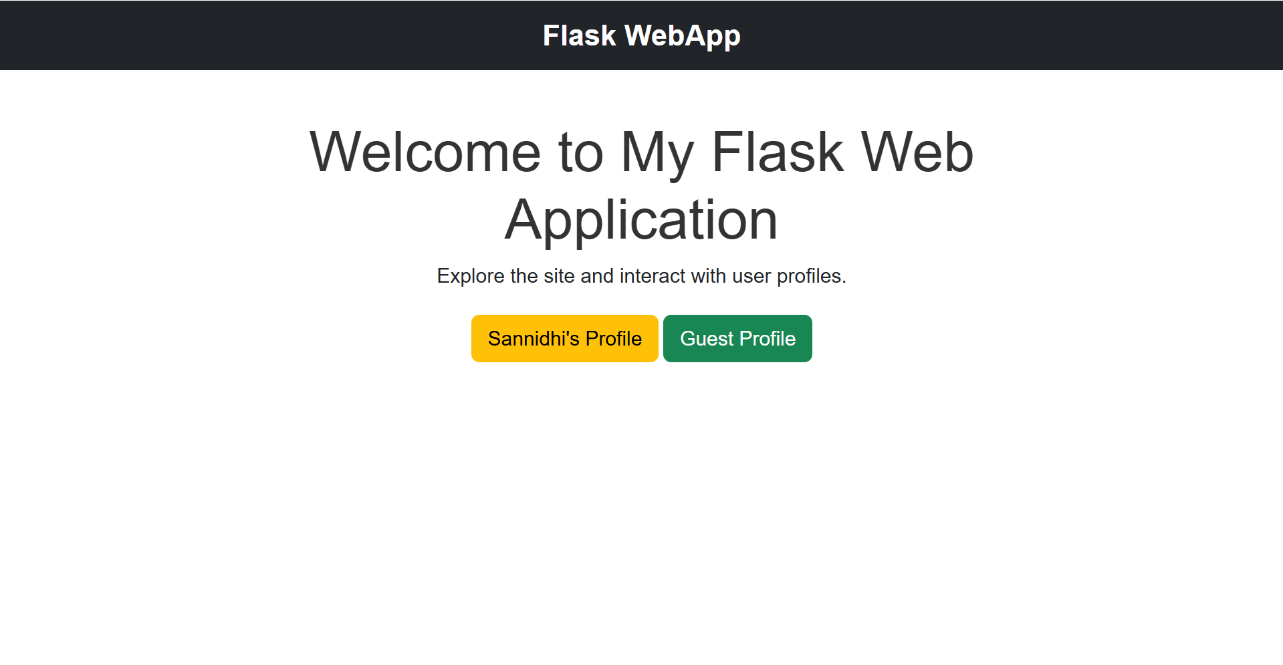
<h1 class="display-4">Hello, {{ username }}!</h1>

<p class="lead">Welcome to your personalized profile page.</p>

<a href="/" class="btn btn-secondary mt-3">Go Back to Home</a>

</div>

{% endblock %}



Akruti’s profile