

#### EXPERIMENT NO. 4

Name of Student	Snehal Anilkumar Patil
Class Roll No	D15A / 38
D.O.P.	
D.O.S.	
Sign and Grade	

#### EXPERIMENT NO. 4

**AIM :** To design a Flask application that showcases URL building and demonstrates the use of HTTP methods (GET and POST) for handling user input and processing data.

#### **PROBLEM STATEMENT :**

Create a Flask application with the following requirements:

1. A homepage (/) with links to a "Profile" page and a "Submit" page using the `url_for()` function.
2. The "Profile" page (`/profile/<username>`) dynamically displays a user's name passed in the URL.
3. A "Submit" page (`/submit`) displays a form to collect the user's name and age. The form uses the POST method to send the data, and the server displays a confirmation message with the input.

#### **Theory:**

1. What is a route in Flask, and how is it defined?
2. How can you pass parameters in a URL route?
3. What happens if two routes in a Flask application have the same URL pattern?
4. What are the commonly used HTTP methods in web applications?
5. What is a dynamic route in Flask?
6. Write an example of a dynamic route that accepts a username as a parameter.
7. What is the purpose of enabling debug mode in Flask?
8. How do you enable debug mode in a Flask application?

## Code:

```
from flask import Flask, render_template, request, url_for, redirect
```

```
app = Flask(__name__)
```

```
# Homepage Route
```

```
@app.route('/')
```

```
def home():
```

```
    return f"""
```

```
<html>
```

```
<head>
```

```
    <title>Flask App</title>
```

```
    <style>
```

```
        body {{ font-family: Arial, sans-serif; text-align: center; margin-top: 50px; }}
```

```
        a, input[type='submit'] {{ display: block; margin: 10px auto; padding: 10px 20px; text-decoration: none;
```

```
            background-color: #007BFF; color: white; border: none; border-radius: 5px;
```

```
            cursor: pointer; font-size: 16px; }}
```

```
        a:hover, input[type='submit']:hover {{ background-color: #0056b3; }}
```

```
        form {{ margin-top: 20px; }}
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <h1>Welcome to the Flask App</h1>
```

```
    <a href='{url_for('profile', username='Snehal Patil')}'>Go to Profile</a>
```

```
    <a href='{url_for('submit')}'>Go to Submit Form</a>
```

```
</body>
```

```
</html>
```

```
"""
```

```
# Dynamic Profile Page Route
```

```
@app.route('/profile/<username>')
```

```
def profile(username):
```

```
    return f"""
```

```
<html>
```

```
<head>
```

```
    <title>Profile Page</title>
```

```
    <style>
```

```
        body {{ font-family: Arial, sans-serif; text-align: center; margin-top: 50px; }}
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <h1>Profile Page</h1>
```

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

```
</body>
```

```
</html>
```

```
"""
```

```
# Submit Page with GET and POST Methods
```

@app.route('/submit', methods=['GET', 'POST'])	age = request.form.get('age')
def submit():	return f"""
if request.method == 'POST':	<html>
name = request.form.get('name')	<head>
<title>Submission Confirmation</title>	
<style>	
body {{ font-family: Arial, sans-serif; text-align: center; margin-top: 50px; }}	
</style>	
</head>	
<body>	
<h1>Thank You!</h1>	
<p>Name: {name}</p>	
<p>Age: {age}</p>	
</body>	
</html>	
"""	
return """	
<html>	
<head>	
<title>Submit Form</title>	
<style>	
body {{ font-family: Arial, sans-serif; text-align: center; margin-top: 50px; }}	
form {{ display: inline-block; text-align: left; }}	
input[type='text'], input[type='number'] {{ width: 100%; padding: 10px; margin: 10px 0; border: 1px solid #ccc; border-radius: 5px; }}	
input[type='submit'] {{ width: 100%; }}	
</style>	
</head>	
<body>	

```
<h1>Submit Form</h1>

<form method='POST'>

    <label for='name'>Name:</label>

    <input type='text' id='name' name='name' required><br>

    <label for='age'>Age:</label>

    <input type='number' id='age' name='age' required><br>

    <input type='submit' value='Submit'>

</form>

</body>

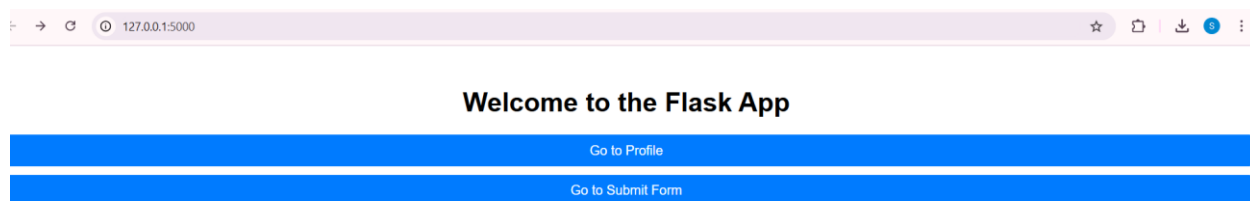
</html>

"""
```

```
if __name__ == '__main__':

    app.run(host='0.0.0.0', port=5000, debug=False)
```

## OUTPUT:



# Submit Form

Name:

Age:

## Thank You!

Name: Snehal  
Age: 20

## Profile Page

Welcome, Snehal Patil!

### 1. What is a route in Flask, and how is it defined?

A route in Flask is a URL pattern that is mapped to a specific function in the application. It defines how the application responds to a request for a particular URL. Routes are defined using the `@app.route()` decorator.

Example:

```
python
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
def home():
    return "Welcome to Flask!"
```

### 2. How can you pass parameters in a URL route?

Parameters can be passed in a Flask route by using angle brackets (`<>`). Flask extracts these parameters and passes them as arguments to the associated function.

Example:

Python

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

Visiting `/user/John` will display "Hello, John!".

### 3. What happens if two routes in a Flask application have the same URL pattern?

If two routes have the same URL pattern, Flask will raise an error because it cannot determine which function to execute for that route.

Example (Incorrect Usage):

Python

```
@app.route('/profile')
def profile1():
    return "Profile 1"
```

```
@app.route('/profile')
def profile2():
    return "Profile 2"
```

This will result in a duplicate route error.

### 4. What are the commonly used HTTP methods in web applications?

The commonly used HTTP methods in web applications include:

- GET – Retrieves data from the server.
- POST – Submits data to be processed on the server.
- PUT – Updates existing data on the server.
- DELETE – Deletes data from the server.

## 5. What is a dynamic route in Flask?

A dynamic route in Flask allows a URL to accept variable parameters, making it flexible to handle different inputs.

Example:

Python

```
@app.route('/product/<int:product_id>')
```

```
def product(product_id):
```

```
    return f"Product ID: {product_id}"
```

Visiting /product/101 will display "Product ID: 101".

## 6. Write an example of a dynamic route that accepts a username as a parameter.

python

```
@app.route('/profile/<username>')
```

```
def profile(username):
```

```
    return f"Welcome, {username}!"
```

Accessing /profile/Alex will return "Welcome, Snehal!".

---

## 7. What is the purpose of enabling debug mode in Flask?

Enabling debug mode in Flask provides:

- Automatic code reloading – No need to restart the server after code changes.
- Detailed error messages – It shows interactive error pages when an error occurs.

---

## 8. How do you enable debug mode in a Flask application?

Debug mode can be enabled by setting debug=True in the app.run() function or by setting an environment variable.

Method 1: Directly in Code

python

CopyEdit

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Method 2: Using Environment Variable

bash

CopyEdit

```
export FLASK_ENV=development
```

```
flask run
```

This enables debug mode automatically.