

## EXPERIMENT NO. 4

Name of Student	Spandan Deb
Class Roll No	13
D.O.P.	
D.O.S.	
Sign and Grade	

**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?

In Flask, a route is a URL pattern that is mapped to a specific function, which is executed when a user visits that URL. Routes define how a web application responds to different requests. They are defined using the `@app.route()` decorator.

Example:

```
from flask import Flask
app = Flask(__name__)

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

if __name__ == '__main__':
    app.run()
```

2) How can you pass parameters in a URL route?

You can pass parameters in a URL route by using angle brackets (<>) in the route definition. Flask captures the value from the URL and passes it as an argument to the function.

**Example:**

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

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 the URL mapping must be unique. The application will not start until the duplicate route is resolved.

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

Commonly used HTTP methods include:

- GET: Requests data from a server (default method in Flask routes).
- POST: Sends data to the server.
- PUT: Updates an existing resource.
- DELETE: Deletes a resource.
- PATCH: Partially updates a resource.

5) What is a dynamic route in Flask?

A dynamic route is a route that includes variables, allowing it to handle different values dynamically. These variables are captured from the URL and passed to the associated function.

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

```
@app.route('/profile/<username>')
def profile(username):
    return f"User Profile: {username}"
```

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

Debug mode allows automatic reloading of the server when code changes and provides detailed error messages when an error occurs. This is useful during development for rapid debugging.

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

Debug mode can be enabled by setting `debug=True` in `app.run()` or by setting the `FLASK_ENV` environment variable to `development`.

Example:

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

Alternatively, you can enable debug mode using the command line:

```
export FLASK_ENV=development # Linux/macOS
set FLASK_ENV=development # Windows
flask run
```

## **CODE:**

### **app.py**

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

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

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

```
def home():
```

```
    return render_template('home.html')
```

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

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

```
    return render_template('profile.html', username=username)
```

```
@app.route('/submit', methods=['GET', 'POST'])
```

```
def submit():
```

```
    if request.method == 'POST':
```

```
        name = request.form.get('name')
```

```
        age = request.form.get('age')
```

```
        return render_template('confirmation.html', name=name, age=age)
```

```
    return render_template('submit.html')
```

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

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

## Templates

### Confirmation.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Flask App - Confirmation</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 40px;
      line-height: 1.6;
    }
    .container {
      max-width: 800px;
      margin: 0 auto;
    }
    h1 {
      color: #333;
    }
    .nav {
      margin: 20px 0;
    }
    .nav a {
      margin-right: 15px;
      text-decoration: none;
      color: #0066cc;
    }
    .nav a:hover {
      text-decoration: underline;
    }
    .confirmation-card {
      background-color: #f0f8ff;
      border-radius: 8px;
      padding: 20px;
      margin-top: 20px;
      border-left: 5px solid #0066cc;
    }
  </style>
</head>
<body>
  <div class="container">
```

```

<h1>Submission Confirmation</h1>
<div class="nav">
  <a href="{{ url_for('home') }}">Home</a>
  <a href="{{ url_for('profile', username=name) }}">View Profile</a>
  <a href="{{ url_for('submit') }}">Submit Again</a>
</div>

<div class="confirmation-card">
  <h2>Thank you for your submission!</h2>
  <p>The following information has been received:</p>
  <ul>
    <li><strong>Name:</strong> {{ name }}</li>
    <li><strong>Age:</strong> {{ age }}</li>
  </ul>
</div>
</div>
</body>
</html>

```

## Home.html

```

<!DOCTYPE html>
<html>
<head>
  <title>Flask App - Home</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 40px;
      line-height: 1.6;
    }
    .container {
      max-width: 800px;
      margin: 0 auto;
    }
    h1 {
      color: #333;
    }
    .nav {
      margin: 20px 0;
    }
    .nav a {

```

```

        margin-right: 15px;
        text-decoration: none;
        color: #0066cc;
    }
    .nav a:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Welcome to the Flask App</h1>
        <div class="nav">
            <a href="{{ url_for('profile', username='default_user') }}">Profile</a>
            <a href="{{ url_for('submit') }}">Submit</a>
        </div>
        <p>This is a simple Flask application demonstrating routing and form
handling.</p>
    </div>
</body>
</html>

```

## Profile.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Flask App - Profile</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
            line-height: 1.6;
        }
        .container {
            max-width: 800px;
            margin: 0 auto;
        }
        h1 {
            color: #333;
        }
    </style>

```

```

.nav {
    margin: 20px 0;
}
.nav a {
    margin-right: 15px;
    text-decoration: none;
    color: #0066cc;
}
.nav a:hover {
    text-decoration: underline;
}
.profile-card {
    background-color: #f9f9f9;
    border-radius: 8px;
    padding: 20px;
    margin-top: 20px;
}
</style>
</head>
<body>
<div class="container">
    <h1>User Profile</h1>
    <div class="nav">
        <a href="{{ url_for('home') }}">Home</a>
        <a href="{{ url_for('submit') }}">Submit</a>
    </div>

    <div class="profile-card">
        <h2>Welcome, {{ username }}!</h2>
        <p>This is your profile page. The username was passed dynamically in the
URL.</p>
    </div>
</div>
</body>
</html>

```



## Submit.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Flask App - Submit</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 40px;
      line-height: 1.6;
    }
    .container {
      max-width: 800px;
      margin: 0 auto;
    }
    h1 {
      color: #333;
    }
    .nav {
      margin: 20px 0;
    }
    .nav a {
      margin-right: 15px;
      text-decoration: none;
      color: #0066cc;
    }
    .nav a:hover {
      text-decoration: underline;
    }
    .form-group {
      margin-bottom: 15px;
    }
    label {
      display: block;
      margin-bottom: 5px;
    }
    input {
      padding: 8px;
      width: 300px;
      border: 1px solid #ddd;
```

```

        border-radius: 4px;
    }
    button {
        padding: 10px 15px;
        background-color: #0066cc;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }
    button:hover {
        background-color: #0055aa;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Submit Information</h1>
        <div class="nav">
            <a href="{{ url_for('home') }}">Home</a>
            <a href="{{ url_for('profile', username='default_user') }}">Profile</a>
        </div>

        <form method="POST" action="{{ url_for('submit') }}">
            <div class="form-group">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required>
            </div>

            <div class="form-group">
                <label for="age">Age:</label>
                <input type="number" id="age" name="age" required>
            </div>

            <button type="submit">Submit</button>
        </form>
    </div>
</body>
</html>

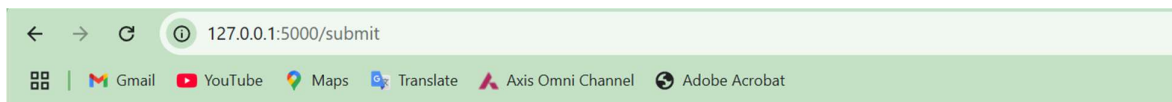
```

## OUTPUT:

# Welcome to the Flask App

[Profile](#) [Submit](#)

This is a simple Flask application demonstrating routing and form handling.

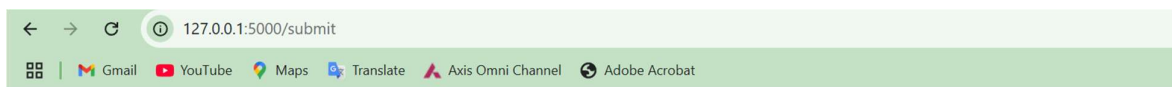


## Submit Information

[Home](#) [Profile](#)

Name:

Age:



## Submission Confirmation

[Home](#) [View Profile](#) [Submit Again](#)

**Thank you for your submission!**

The following information has been received:

- **Name:** Spandan
- **Age:** 20

## User Profile

[Home](#) [Submit](#)

**Welcome, Spandan!**

This is your profile page. The username was passed dynamically in the URL.

### CONCLUSION:

This experiment demonstrated Flask routing and HTTP methods. By implementing URL building, dynamic routes, and handling user input with GET and POST requests, gained hands-on experience in Flask web development. Enabling debug mode also improved my understanding of debugging and error handling, making development more efficient.