

## EXPERIMENT NO. 4 - Flask Application using GET and POST

Name of Student	Eesha Chavan
Class Roll No	D15A 08
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?

A route is a URL pattern linked to a function in Flask using the `@app.route()` decorator.

##### Example:

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

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

Parameters can be passed using angle brackets (`< >`) in the route. Flask will capture these values and pass them to the function as arguments. You can also specify data types like `<int:id>` or `<string:name>`.

##### Example:

```
@app.route('/user/<string:name>')
```

```
def greet_user(name):
```

```
    return f"Hello, {name}!"
```

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

If two routes share the same URL, Flask will use the last-defined route and override the previous one. This causes unexpected behavior and conflicts.

#### Example:

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

```
def hello1():
```

```
    return "Hello from function 1"
```

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

```
def hello2():
```

```
    return "Hello from function 2"
```

```
# Only "Hello from function 2" will be shown.
```

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

- HTTP methods define the type of request a client sends to a server.
- GET: Retrieve data (e.g., accessing a web page).
- POST: Send data to the server (e.g., submitting a form).
- PUT: Update existing data.
- DELETE: Remove data.
- PATCH: Partially update data.

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

A dynamic route allows variables to be embedded within the URL, making it more flexible. The data in the URL is passed to the function for further processing.

#### Example:

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

```
) def show_profile(username):
```

```
    return f"Welcome to {username}'s Profile!"
```

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

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

def welcome_user(username):

    return f"Hello, {username}! Glad to see you here."
```

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

Debug Mode is used during development for easy troubleshooting. It enables:

**Automatic Code Reloading:** The app restarts when code changes.

**Detailed Error Messages:** Displays an interactive debugger in case of an error. It should be disabled in production for security reasons.

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

You can enable debug mode using one of these methods:

```
Using app.run()

export

FLASK_ENV=development flask

run
```

**GitHub Link:**

<https://github.com/eeshachavan/WebX-Exp4-eesha>

**CODE:**

app.py:

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

app = Flask(__name__)
app.secret_key = "supersecretkey"

@app.route('/')
def home():
    return f'''
    <html>
    <head>
        <title>Flask App - Home</title>
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
        <style>
```

```
body {{
    background-color: #f8f9fa;
    text-align: center;
    color: #343a40;
    font-family: 'Segoe UI', sans-serif;
    margin: 0;
    padding: 0;
    line-height: 1.6;
}}

.container {{
    max-width: 800px;
    margin: 0 auto;
    padding: 40px 20px;
}}

.header {{
    background-color: #6c63ff;
    color: white;
    padding: 30px 0;
    margin-bottom: 40px;
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}}

.btn {{
    display: inline-block;
    margin: 10px;
    padding: 15px 30px;
    background-color: #6c63ff;
    text-decoration: none;
    color: white;
    border-radius: 8px;
    transition: all 0.3s ease;
    font-weight: 500;
    box-shadow: 0 4px 8px rgba(108, 99, 255, 0.3);
}}

.btn:hover {{
    background-color: #5a52d5;
    transform: translateY(-2px);
    box-shadow: 0 6px 12px rgba(108, 99, 255, 0.4);
}}

h1 {{
    font-size: 3em;
    margin-bottom: 20px;
}}

.footer {{
    margin-top: 60px;
    color: #6c757d;
    font-size: 0.9em;
}}

</style>
```



```
    }}
    .profile-card {{
        background-color: white;
        border-radius: 12px;
        padding: 30px;
        box-shadow: 0 8px 16px rgba(0,0,0,0.1);
        max-width: 500px;
        margin: 0 auto;
        margin-bottom: 30px;
    }}
    .profile-detail {{
        margin: 15px 0;
        font-size: 1.2em;
    }}
    .profile-label {{
        font-weight: bold;
        color: #6c63ff;
    }}
    .btn {{
        display: inline-block;
        margin: 10px;
        padding: 15px 30px;
        background-color: #6c63ff;
        text-decoration: none;
        color: white;
        border-radius: 8px;
        transition: all 0.3s ease;
        font-weight: 500;
        box-shadow: 0 4px 8px rgba(108, 99, 255, 0.3);
    }}
    .btn:hover {{
        background-color: #5a52d5;
        transform: translateY(-2px);
        box-shadow: 0 6px 12px rgba(108, 99, 255, 0.4);
    }}
    h1 {{
        font-size: 2.5em;
        margin-bottom: 20px;
    }}
    .footer {{
        margin-top: 60px;
        color: #6c757d;
        font-size: 0.9em;
    }}
</style>
</head>
<body>
    <div class="header">
```

```

        <h1>User Profile</h1>
    </div>
    <div class="container">
        <div class="profile-card">
            <div class="profile-detail">
                <span class="profile-label">Name:</span> {name}
            </div>
            <div class="profile-detail">
                <span class="profile-label">Age:</span> {age}
            </div>
        </div>
        <a class="btn" href="{url_for('home')}}">Back to Homepage</a>
        <div class="footer">
            <p>&copy; 2025 Flask Application</p>
        </div>
    </div>
</body>
</html>
'''

```

```

@app.route('/submit', methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        session['name'] = request.form.get('name', 'Unknown')
        session['age'] = request.form.get('age', 'Unknown')
        return redirect(url_for('profile'))

    return f'''
<html>
<head>
    <title>Flask App - Submit</title>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <style>
        body {{
            background-color: #f8f9fa;
            text-align: center;
            color: #343a40;
            font-family: 'Segoe UI', sans-serif;
            margin: 0;
            padding: 0;
            line-height: 1.6;
        }}
        .container {{
            max-width: 800px;
            margin: 0 auto;
            padding: 40px 20px;
        }}
    </style>
    <div class="container">
        <div class="profile-card">
            <div class="profile-detail">
                <span class="profile-label">Name:</span> {name}
            </div>
            <div class="profile-detail">
                <span class="profile-label">Age:</span> {age}
            </div>
        </div>
        <a class="btn" href="{url_for('home')}}">Back to Homepage</a>
        <div class="footer">
            <p>&copy; 2025 Flask Application</p>
        </div>
    </div>
</body>
</html>
'''

```

```
.header {{
    background-color: #6c63ff;
    color: white;
    padding: 30px 0;
    margin-bottom: 40px;
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}}

form {{
    display: inline-block;
    background-color: #ffffff;
    padding: 30px 40px;
    border-radius: 12px;
    box-shadow: 0 8px 16px rgba(0,0,0,0.1);
    text-align: left;
    min-width: 300px;
}}

.form-group {{
    margin-bottom: 20px;
}}

label {{
    display: block;
    margin-bottom: 8px;
    color: #495057;
    font-weight: 500;
}}

input {{
    padding: 12px 15px;
    border-radius: 6px;
    border: 1px solid #ced4da;
    width: 100%;
    font-size: 16px;
    box-sizing: border-box;
    transition: border-color 0.3s;
}}

input:focus {{
    outline: none;
    border-color: #6c63ff;
    box-shadow: 0 0 0 3px rgba(108, 99, 255, 0.25);
}}

input[type="submit"] {{
    background-color: #6c63ff;
    color: white;
    cursor: pointer;
    transition: all 0.3s ease;
    font-weight: 500;
    margin-top: 10px;
    box-shadow: 0 4px 8px rgba(108, 99, 255, 0.3);
}}
```





```

        </form>

        <a class="btn" href="{url_for('home')}}">Back to Homepage</a>

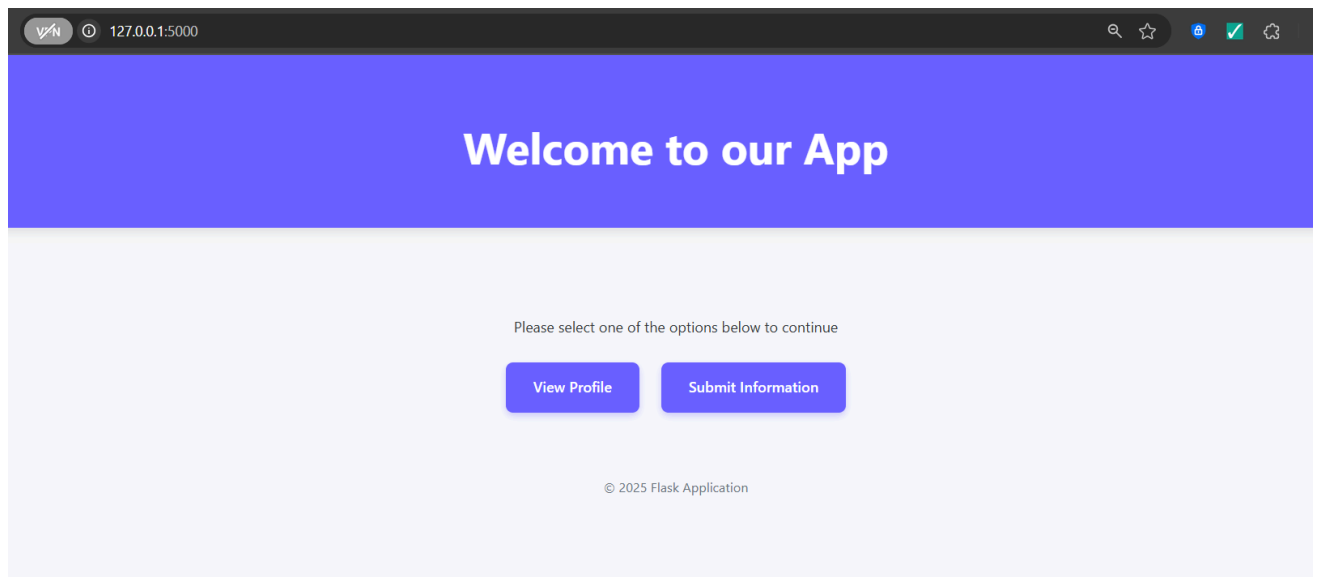
        <div class="footer">
            <p>&copy; 2025 Flask Application</p>
        </div>
    </div>
</body>
</html>
'''

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

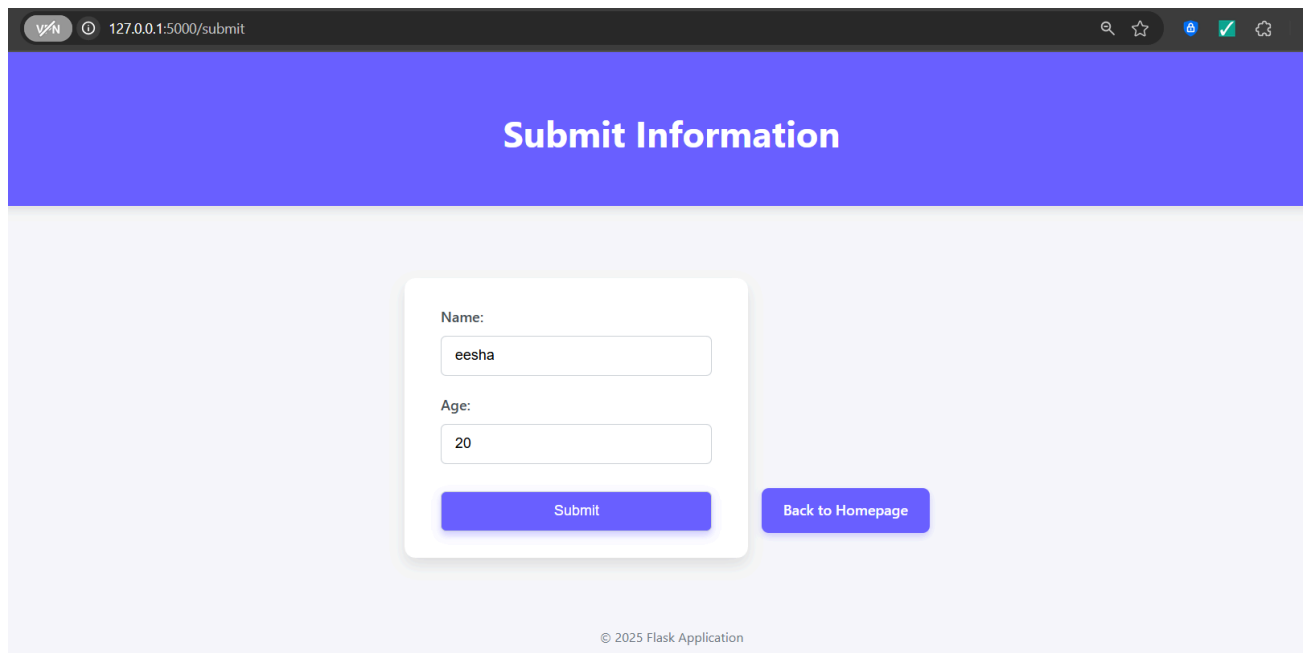
```

## OUTPUT :

**Homepage:** Displays a welcome message with navigation links to the "Profile" and "Submit" pages.

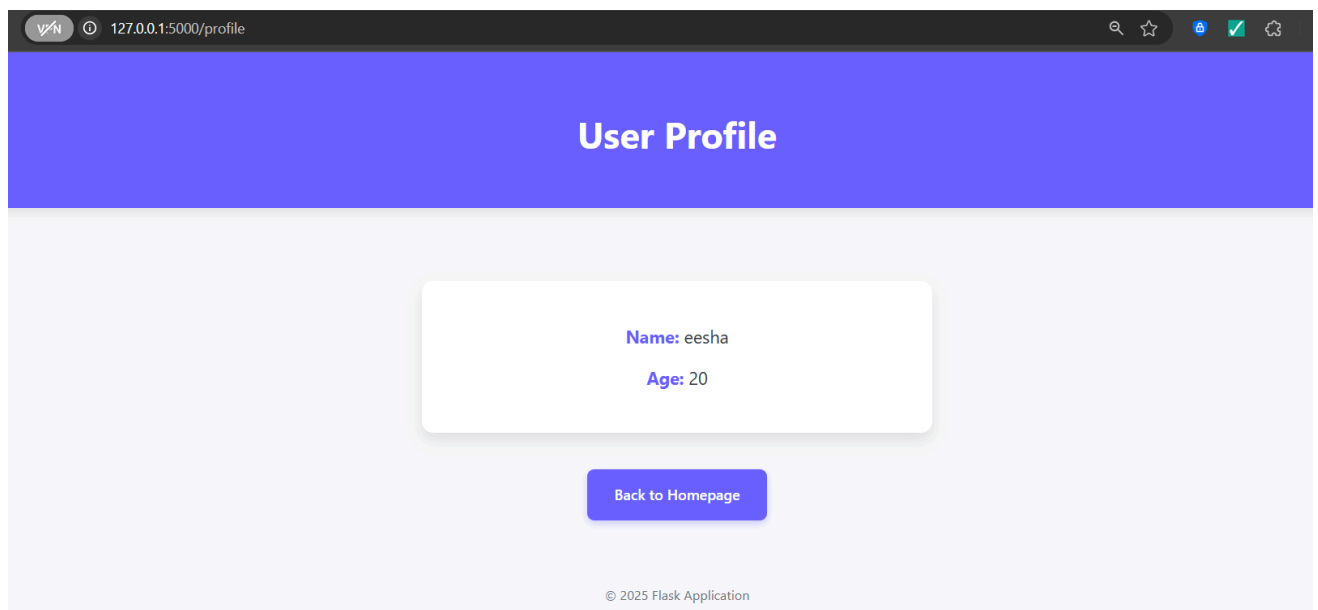


**Submit Page:** Displays a form to collect the user's name and age. Upon submitting, the data is stored using a session, and the user is redirected to the Profile page.



A screenshot of a web browser showing the 'Submit Information' page. The browser's address bar displays '127.0.0.1:5000/submit'. The page has a blue header with the title 'Submit Information'. Below the header, there is a white form box containing two input fields: 'Name:' with the value 'eesha' and 'Age:' with the value '20'. Below these fields are two buttons: a blue 'Submit' button and a blue 'Back to Homepage' button. At the bottom of the page, there is a small copyright notice: '© 2025 Flask Application'.

**Profile Page:** Displays the submitted name and age dynamically. If no data is submitted, it shows "Guest" and "Unknown" as default values.



A screenshot of a web browser showing the 'User Profile' page. The browser's address bar displays '127.0.0.1:5000/profile'. The page has a blue header with the title 'User Profile'. Below the header, there is a white box displaying the user's profile information: 'Name: eesha' and 'Age: 20'. Below this box is a blue 'Back to Homepage' button. At the bottom of the page, there is a small copyright notice: '© 2025 Flask Application'.

## Conclusion:

The experiment demonstrated the creation of a simple Flask application using GET and POST methods. The application effectively handled dynamic routing using URL parameters and processed user input via a form using the POST method. Additionally, it showcased URL building using the `url_for()` function and maintained user data using sessions.