

## EXPERIMENT NO. 5

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

## 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?
2. What is the significance of the `templates` folder in a Flask project?
3. What is Jinja2, and how does it integrate with Flask?

## Code:

### App.py

```
from flask import Flask, render_template

app = Flask(__name__)

# Homepage Route
@app.route('/')
def home():
    return render_template('index.html')

# Dynamic User Route
@app.route('/user/<username>')
def user(username):
    return render_template('user.html', name=username)

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

### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-scale=1">
  <title>Flask App - Home</title>
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.
0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="bg-light">

  <!-- Navbar -->
  <nav class="navbar navbar-dark bg-primary">
    <div class="container">
      <a class="navbar-brand" href="#">Flask
App</a>
    </div>
  </nav>
```

```
    <!-- Main Content -->
    <div class="container text-center mt-5">
      <h1 class="mb-3">Welcome to the Flask
Application!</h1>
      <p class="lead">Explore user profiles
below:</p>

      <a href="{{ url_for('user', username='Snehal
Patil') }}" class="btn btn-primary m-2">Go to
Snehal's Profile</a>
      <a href="{{ url_for('user', username='Admin')
}}" class="btn btn-primary m-2">Go to Admin's
Profile</a>
    </div>

    <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.0/
dist/js/bootstrap.bundle.min.js"></script>
  </body>
</html>
```

**User.html:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>User Profile</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="bg-light">

  <!-- Navbar -->
  <nav class="navbar navbar-dark bg-primary">
    <div class="container">
      <a class="navbar-brand" href="{{ url_for('home') }}">Flask App</a>
    </div>
  </nav>

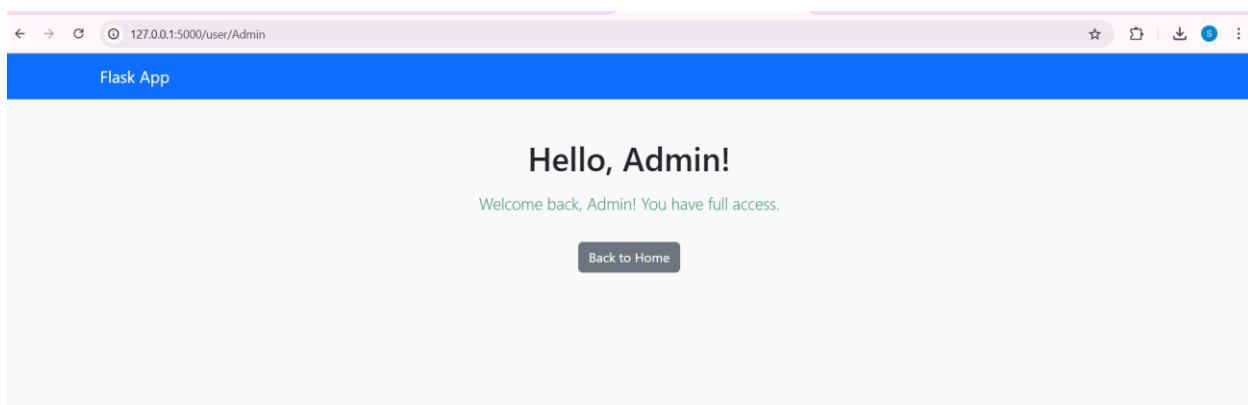
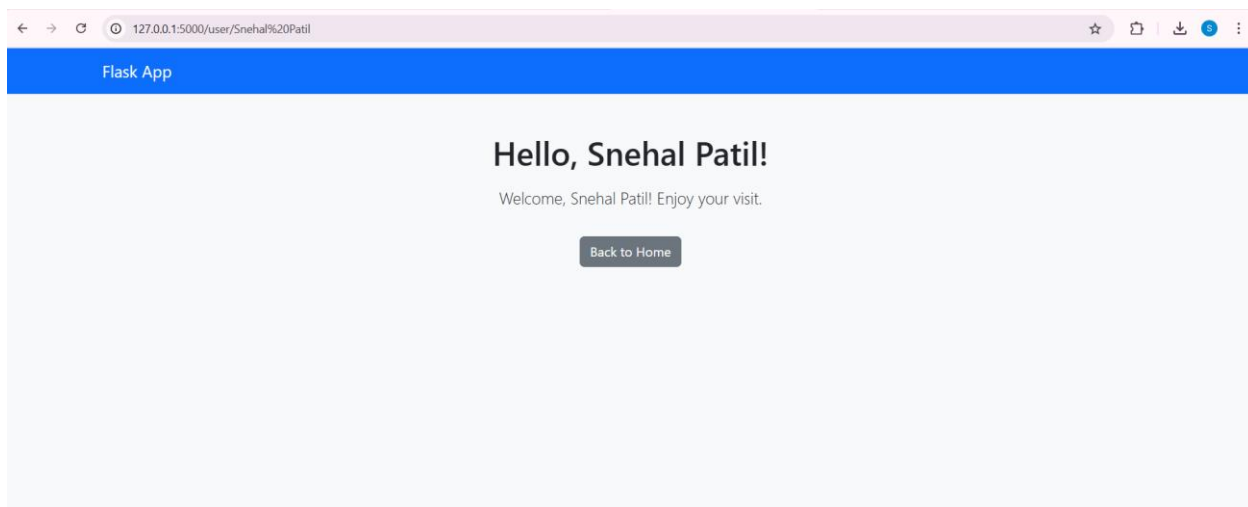
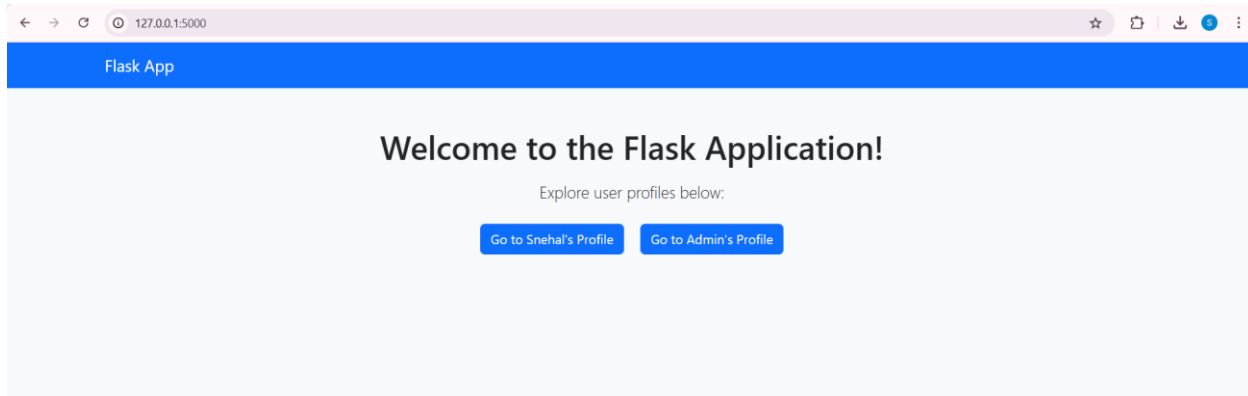
  <!-- Profile Section -->
  <div class="container text-center mt-5">
    <h1 class="mb-3">Hello, {{ name }}!</h1>

    {% if name == "Admin" %}
      <p class="lead text-success">Welcome back, Admin! You have full access.</p>
    {% else %}
      <p class="lead">Welcome, {{ name }}! Enjoy your visit.</p>
    {% endif %}

    <a href="{{ url_for('home') }}" class="btn btn-secondary mt-3">Back to Home</a>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

## IMPLEMENTATION:



## Theory Answers:

### 1. What does the `render_template()` function do in a Flask application?

- The `render_template()` function is used to **render HTML templates** from the templates folder.
- It dynamically injects data into the HTML using **Jinja2**, allowing for a more interactive user experience.
- Example:

```
from flask import Flask, render_template
```

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

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

```
def user(username):
```

```
    return render_template('user.html', name=username)
```

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

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

- This will render `user.html` and pass the `username` variable to the template.

---

### 2. What is the significance of the templates folder in a Flask project?

- Flask automatically looks for HTML templates in a folder named **templates**.
- It allows for **separation of logic and presentation**, following the **MVC (Model-View-Controller)** pattern.
- Placing HTML files inside templates makes them accessible to `render_template()`.
- Folder structure:

```
/project_folder
```

```
|— app.py (Flask application)
|— templates/
|   |— index.html
|   |— user.html
|— static/ (for CSS, JS, images)
```

---

### 3. What is Jinja2, and how does it integrate with Flask?

- **Jinja2** is a **templating engine** used in Flask to embed Python-like expressions into HTML.
- It supports:
  - **Variables:** `{{ name }}` (Inserts dynamic values)
  - **Loops:** `{% for item in list %}...{% endfor %}`
  - **Conditionals:** `{% if condition %}...{% endif %}`
- Example `user.html`:

```
html
```

```
<html>
```

```
<body>
```

```
    <h1>Hello, {{ name }}!</h1>
```

```
    {% if name == "Admin" %}
```

```
        <p>Welcome back, Admin!</p>
```

```
    {% else %}
```

```
        <p>Enjoy your visit, {{ name }}!</p>
```

```
    {% endif %}
```

```
</body>
```

```
</html>
```

- Flask automatically integrates **Jinja2** when using `render_template()`.