

## **EXPERIMENT NO. 5**

<b>Name of Student</b>	<b><u>Shreyash Ganesh Dhekane</u></b>
<b>Class Roll No</b>	<b><u>16</u></b>
<b>D.O.P.</b>	<b><u>04-03-2025</u></b>
<b>D.O.S.</b>	<b><u>18-03-2025</u></b>
<b>Sign and Grade</b>	

### **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 template

## THEORY:

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

The `render_template()` function in Flask is used to render HTML templates stored in the `templates` folder. Instead of returning plain text from a route, Flask can return a complete HTML page using this function.

#### Key Features of `render_template()`:

- Loads an HTML file and returns it as the HTTP response.
- Allows passing dynamic content (variables) to the template.
- Supports template inheritance, which helps in maintaining reusable layouts.
- Uses **Jinja2 templating** to enable dynamic content rendering.

### 2. What Is the Significance of the `templates` Folder in a Flask Project?

The `templates` folder is a **default directory** in Flask where all HTML files (templates) are stored. Flask automatically looks for templates in this folder when using `render_template()`.

#### Why Is the `templates` Folder Important?

- **Separation of Concerns:** It keeps HTML files separate from Python code, following the MVC (Model-View-Controller) architecture.
- **Organized Structure:** Helps in managing multiple pages and layouts efficiently.
- **Automatic Lookup:** Flask automatically searches for HTML templates inside this folder.
- **Supports Template Inheritance:** Allows reusing layouts using a base template.

### 3. What Is Jinja2, and How Does It Integrate with Flask?

**Jinja2** is a powerful templating engine used in Flask to dynamically generate HTML content. It allows embedding Python-like expressions inside HTML template

#### Key Features of Jinja2 in Flask:

- Supports **template variables** (`{{ variable }}`) for dynamic content.
- Allows **control structures** like loops (`{% for %}`) and conditionals (`{% if %}`).
- Supports **template inheritance** for reusable layouts.
- Provides **filters** and **macros** to modify output.

#### How Jinja2 Integrates with Flask?

- Flask uses Jinja2 internally to process templates.
- When calling `render_template()`, Flask renders the template using Jinja2.
- Developers can pass Python variables, lists, and objects to templates dynamically

**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 Page
@app.route('/user/<username>')
def user(username):
    user_interests = {
        "Shreyash": ["Reading", "Music", "Gaming"],
        "Guest": [],
        "Alice": ["Photography", "Gaming", "Reading"]
    }
    interests = user_interests.get(username, ["No specific interests"])
    return render_template('user.html', username=username, interests=interests)

# About Page
@app.route('/about')
def about():
    values = ["Innovation", "Integrity", "Collaboration", "Excellence", "Customer Focus"]
    return render_template('about.html',
                           title="About Us",
                           heading="About Us",
                           description="This is a simple Flask application demonstrating template rendering.",
                           values=values)

@app.route('/contact')
def contact():
    contact_info = {
        "email": "shreyash@gmail.com",
        "phone": "+123 456 7890",
        "address": "Kalyan , West"
    }
    return render_template('contact.html', contact_info=contact_info)

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

## templates/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flask App</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>Welcome to the Flask App</h1>
    <p>This is the homepage. Click below to navigate:</p>
    <a href="{{ url_for('user', username='Guest') }}" class="button">Visit Profile</a>
    <a href="{{ url_for('about') }}" class="button">About Us</a>
    <a href="{{ url_for('contact') }}" class="button">Contact</a>
  </div>
</body>
</html>
```

## templates/about.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{ title }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>{{ heading }}</h1>
    <p>{{ description }}</p>
```

```

<h2>Our Mission</h2>
<p>We strive to create user-friendly web applications that enhance digital experiences.
  Our goal is to simplify development with intuitive and scalable solutions.</p>

<h2>Our Values</h2>
<ul>
  {% for value in values %}
    <li>{{ value }}</li>
  {% endfor %}
</ul>

<h2>Meet the Team</h2>
<p>Our team consists of passionate developers, designers, and strategists dedicated to
delivering high-quality solutions.</p>

  <a href="{{ url_for('home') }}" class="button">Back to Home</a>
</div>
</body>
</html>

```

### templates/contact.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contact</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
  <div class="container">
    <h1>Contact Us</h1>
    <p>Email: {{ contact_info.email }}</p>
    <p>Phone: {{ contact_info.phone }}</p>
    <p>Address: {{ contact_info.address }}</p>
    <a href="{{ url_for('home') }}" class="button">Back to Home</a>
  </div>
</body>
</html>

```

### templates/user.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome {{ username }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>

```

```

<div class="container">
  <h1>Hello, {{ username }}!</h1>
  {% if username == "Guest" %}
    <p>Welcome to our platform! Consider signing up for a personalized experience.</p>
  {% else %}
    <p>Welcome back, {{ username }}! Here's your profile page where you can manage your
settings.</p>
  {% endif %}

  <h2>Your Interests</h2>
  <ul>
    {% for interest in interests %}
      <li>{{ interest }}</li>
    {% else %}
      <li>No interests found. Update your profile to add some!</li>
    {% endfor %}
  </ul>

  <a href="{{ url_for('home') }}" class="button">Back to Home</a>
</div>
</body>
</html>

```

### static/style.css

/\* General Styles \*/

```

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(to right, #c59eff, #ad31ff);
  color: #1e1e1e;
  text-align: center;
  height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  margin: 0;
}

```

/\* Container Box \*/

```

.container {
  background: rgba(255, 255, 255, 0.15);
  padding: 30px 40px;
  border-radius: 16px;
  box-shadow: 0px 8px 20px rgba(0, 0, 0, 0.15);
  backdrop-filter: blur(10px);
  -webkit-backdrop-filter: blur(10px);
  width: 90%;
  max-width: 500px;
}

```

/\* Header \*/

```

h1 {
  margin-bottom: 10px;
  font-size: 32px;
  color: #ffffff;
}

```

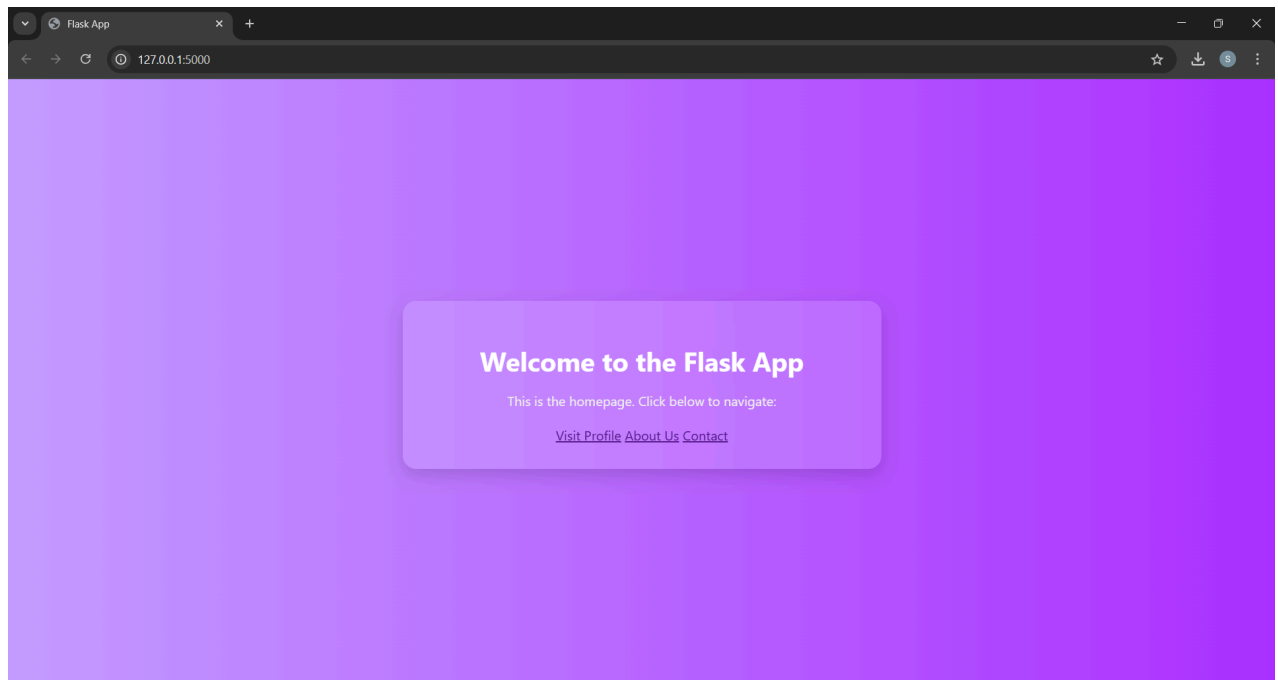
/\* Paragraph \*/

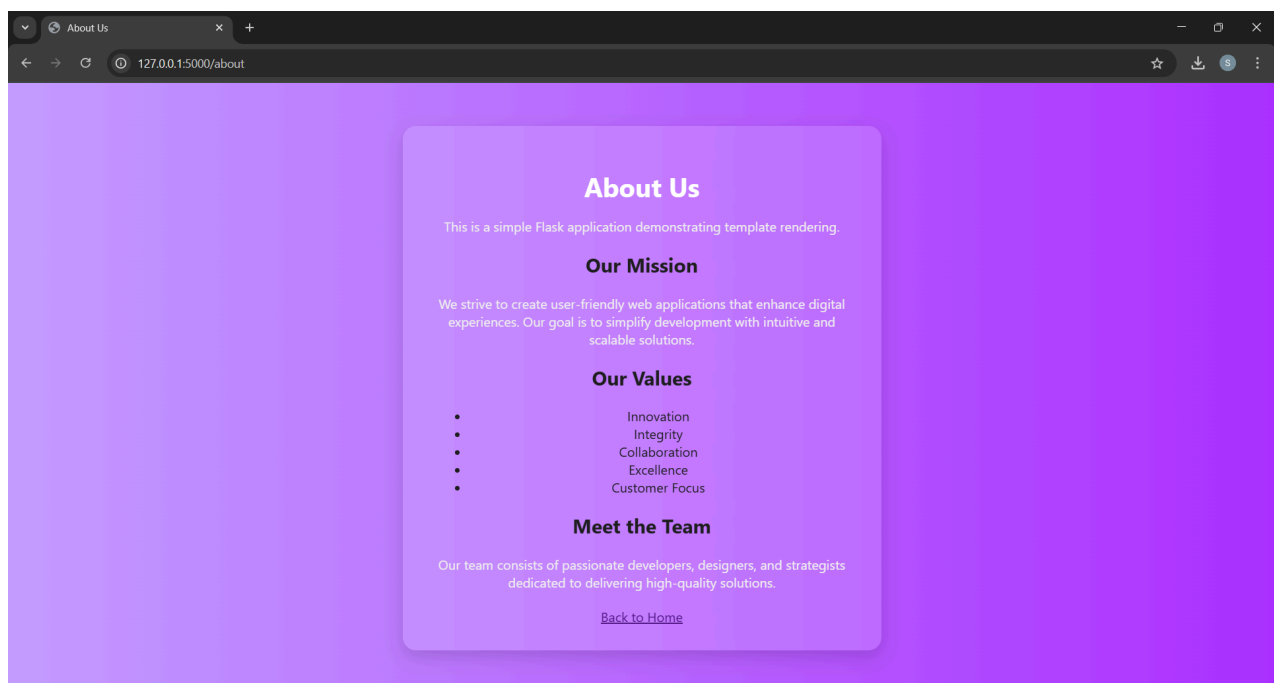
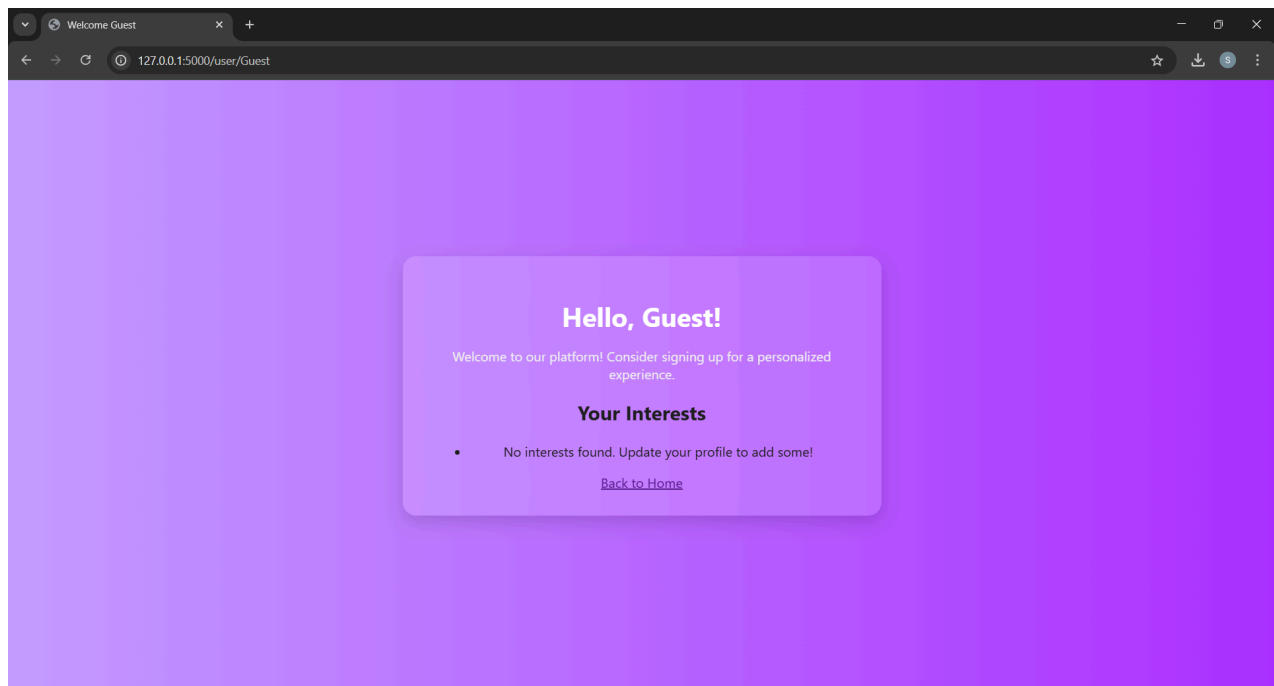
```
p {
  margin-bottom: 20px;
  font-size: 16px;
  color: #f0f0f0;
}

/* Buttons */
.btn {
  background: #ad31ff;
  border: none;
  color: white;
  padding: 12px 24px;
  text-decoration: none;
  font-size: 16px;
  border-radius: 8px;
  margin: 5px;
  display: inline-block;
  transition: background 0.3s ease, transform 0.2s, box-shadow 0.3s;
  box-shadow: 0 4px 14px rgba(173, 49, 255, 0.3);
}

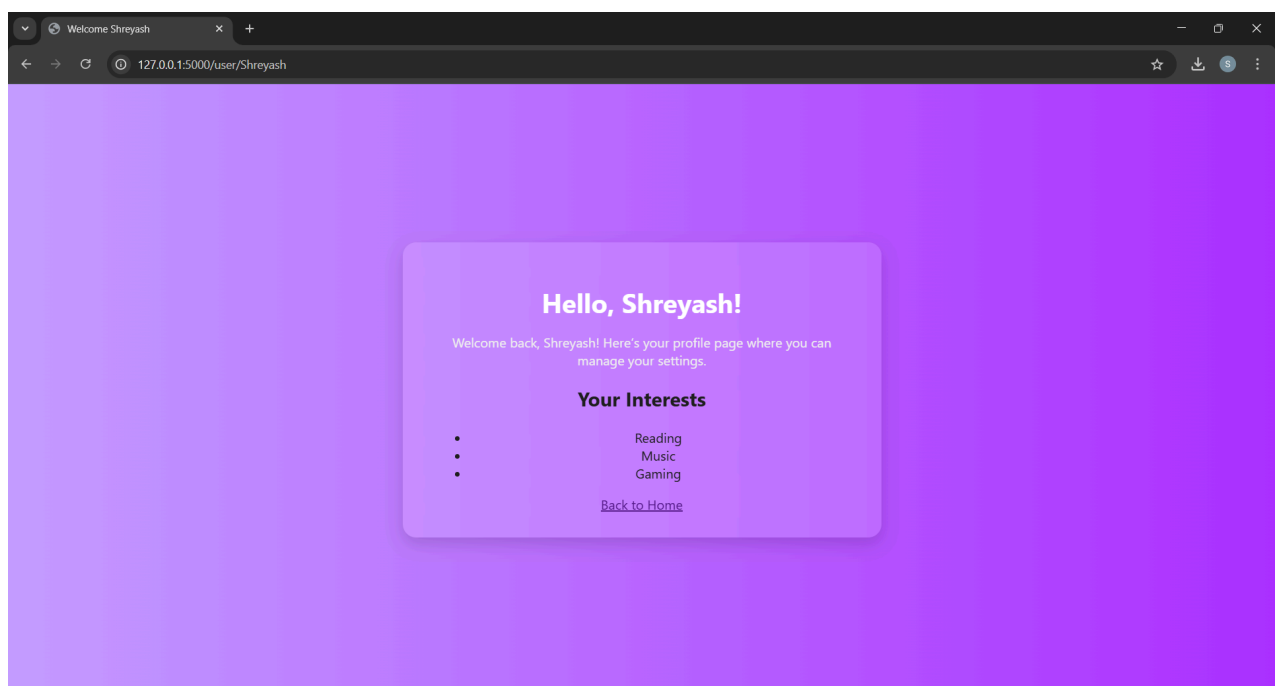
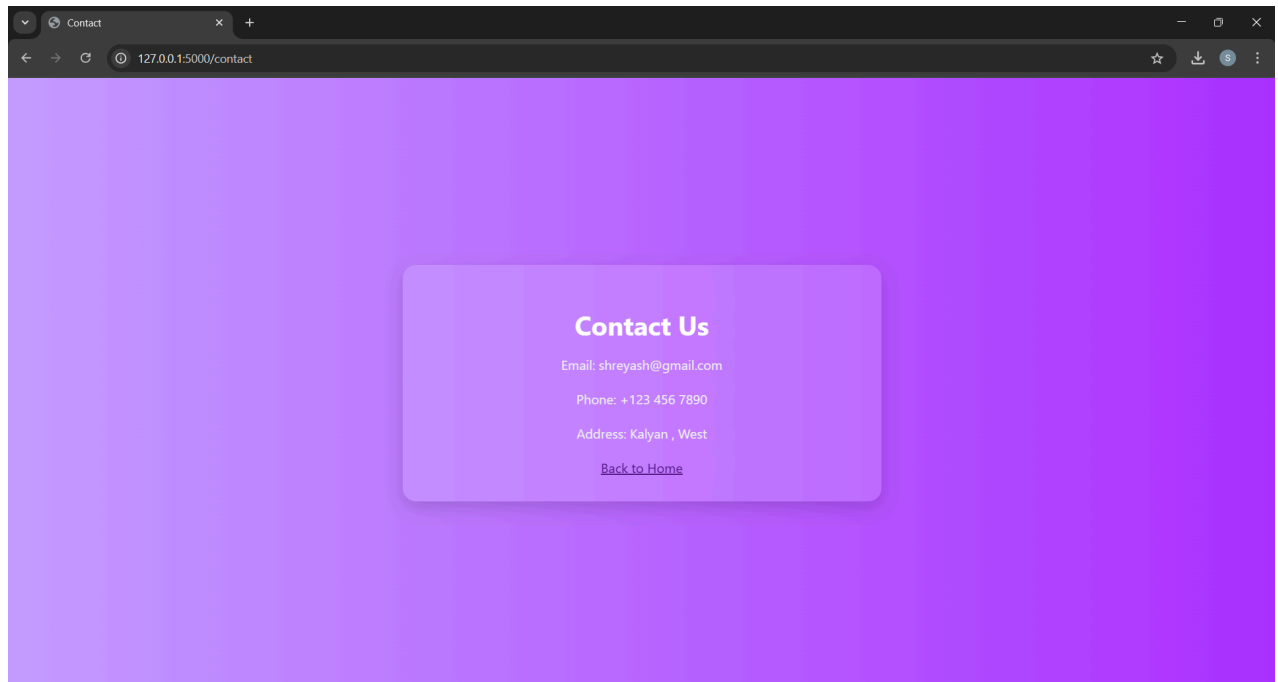
.btn:hover {
  background: #931bdf;
  transform: scale(1.07);
  box-shadow: 0 6px 18px rgba(173, 49, 255, 0.4);
}
```

## Output:-









## **Conclusion : -**

The Flask application successfully demonstrates template rendering using the `render_template()` function and Jinja2 templating features. The homepage provides navigation links, while the dynamic user route personalized content by passing data to an HTML template. By utilizing variables and control structures, the application efficiently generates dynamic web pages, showcasing Flask's ability to create flexible and interactive web applications.

