EXPERIMENT NO. 5

Name of Student	**Prajyot Shinde**
Class Roll No	**55**
D.O.P.	**04-03-25**
D.O.S.	**18-03-25**
Sign and Grade	

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?

   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 templates.

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__)

@app.route('/')
def home():
    return render_template('index.html',
                         title='Welcome',
                         featured_users=['Alex', 'Jordan', 'Taylor'])

@app.route('/user/<username>')
def user_profile(username):
    user_data = {
        'name': username,
        'join_date': '2023-05-15',
        'posts': 42,
        'likes': 128,
        'is_premium': len(username) % 2 == 0  # Just for demo
    }
    return render_template('user.html', user=user_data)

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

templates/index.html

```
<section class="hero">

    <h1>Welcome to Our Platform</h1>

    <p class="subtitle">Experience the power of Flask templating with modern UI</p>

    <a href="#featured" class="btn">Explore</a>

</section>

<section id="featured" class="featured-users">

    <h2>Featured Users</h2>

    <div class="user-grid">

        {% for user in featured_users %}

        <a href="{{ url_for('user_profile', username=user) }}" class="user-card">

            <div class="avatar">{{ user[0] }}</div>

            <h3>{{ user }}</h3>

        </a>

        {% endfor %}

    </div>

</section>
```

templates/base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Flask Template Demo{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swa
p" rel="stylesheet">
</head>
<body>
    <nav class="navbar">
        <div class="container">
            <a href="{{ url_for('home') }}" class="logo">FlaskDemo</a>
            <div class="nav-links">
                <a href="{{ url_for('home') }}">Home</a>
                <a href="#">About</a>
                <a href="#">Contact</a>
            </div>
        </div>
    </nav>

    <main class="container">
        {% block content %}{% endblock %}
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2023 Flask Template Demo. All rights reserved.</p>
        </div>
    </footer>
</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>
```

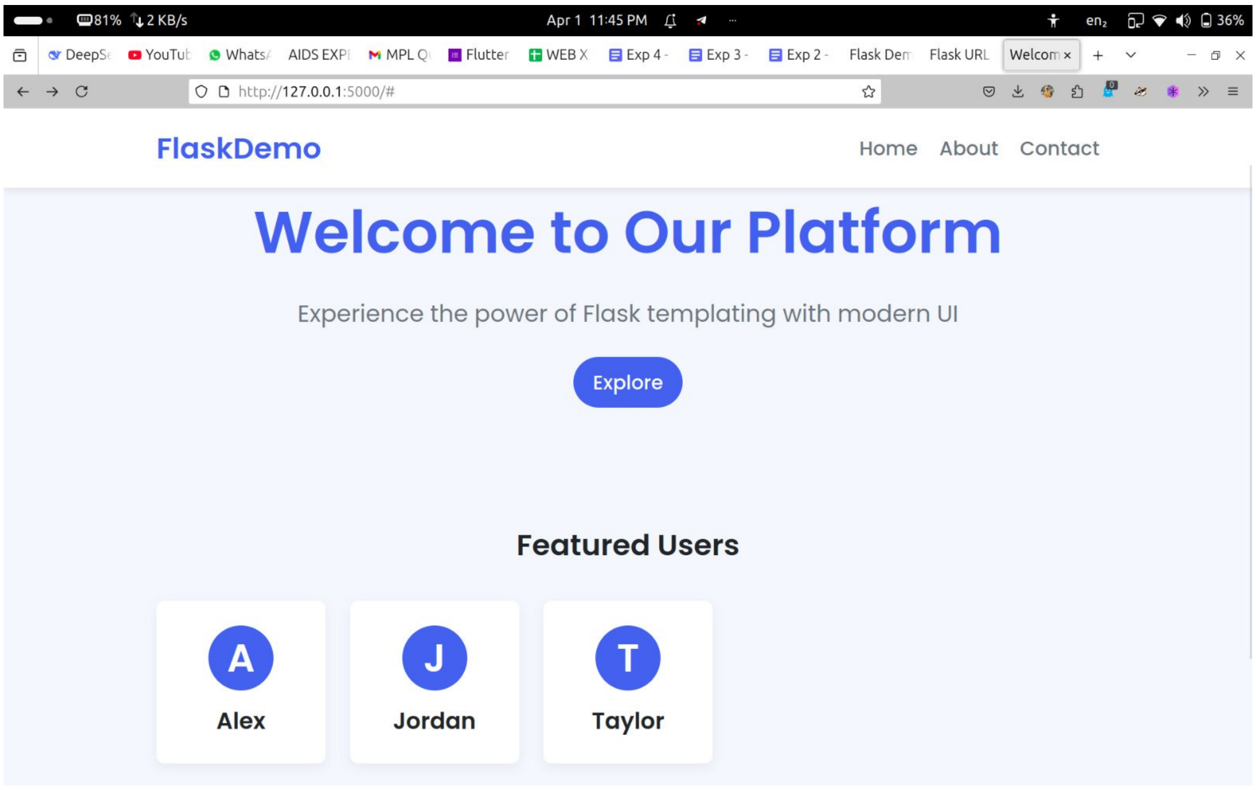
```
        {% endif %}

        <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: Arial, sans-serif;
    background: linear-gradient(to right, #c5ace0, #678ed2);
    color: rgb(0, 0, 0);
    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, 0.1);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);
    width: 90%;
    max-width: 500px;
}




/* Paragraph */
p {
    margin-bottom: 20px;
    font-size: 16px;
}

/* Buttons */
```

```
.btn {
    background: #ff9800;
    border: none;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    font-size: 16px;
    border-radius: 5px;
    margin: 5px;
    display: inline-block;
    transition: background 0.3s ease, transform 0.2s;
}

.btn:hover {
    background: #e68900;
    transform: scale(1.05);
}
```

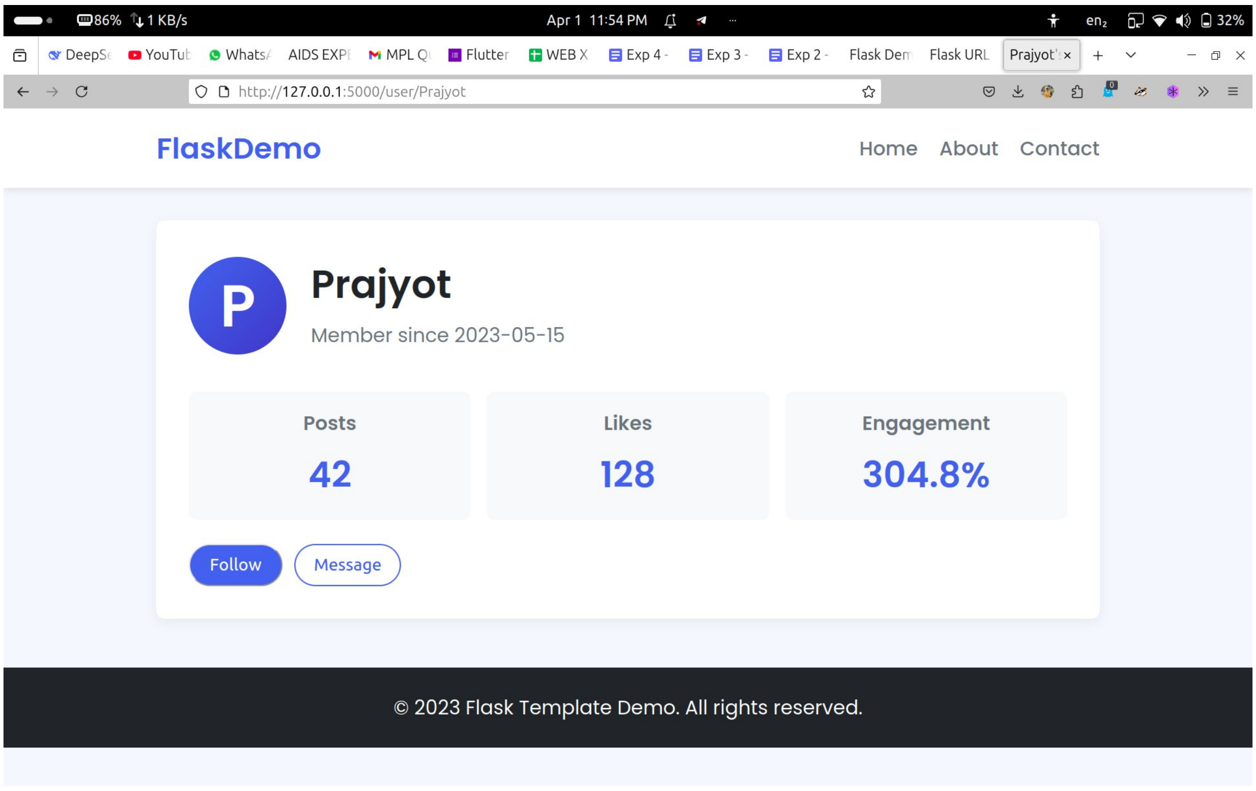
OUTPUT: -



Welcome to Our Platform

Experience the power of Flask templating with modern UI

Featured Users



http://127.0.0.1:5000/user/Prajyot

FlaskDemo
Home About Contact

Prajyot
Member since 2023-05-15

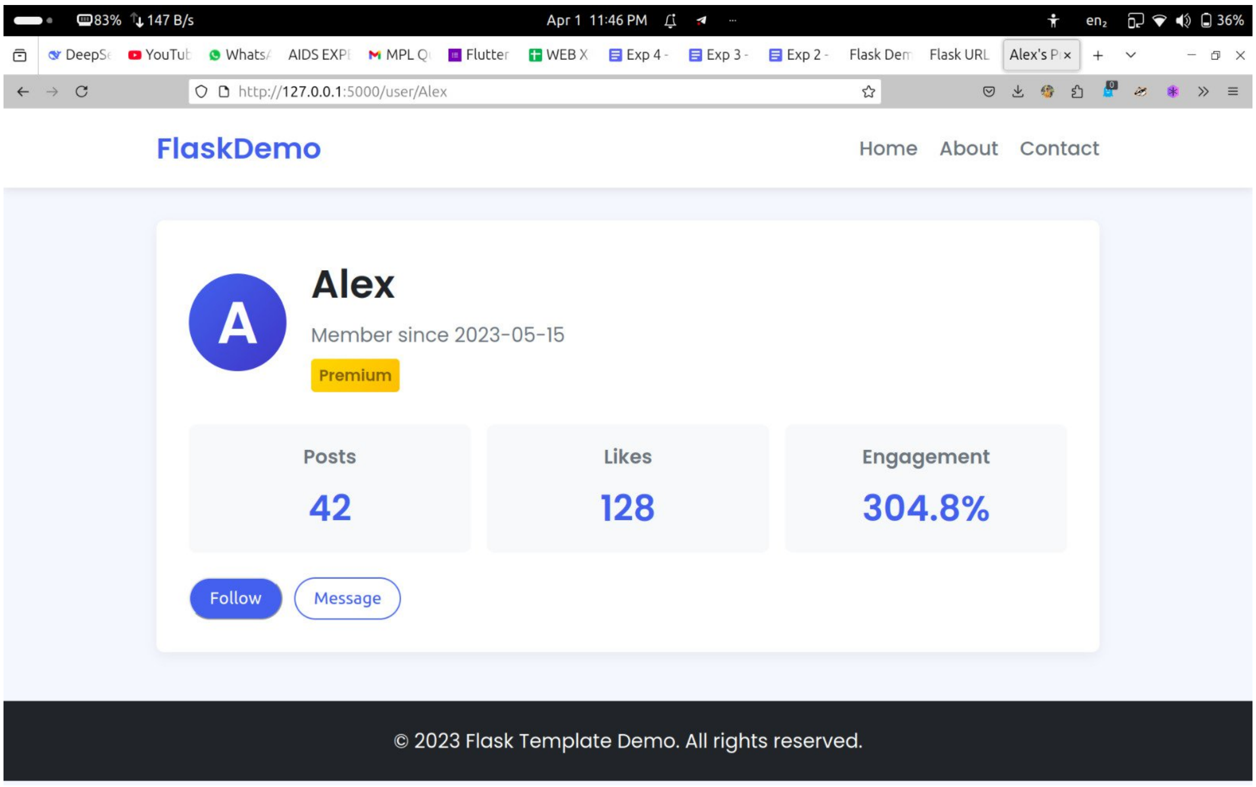
Posts
42

Likes
128

Engagement
304.8%

Follow Message

© 2023 Flask Template Demo. All rights reserved.



http://127.0.0.1:5000/user/Alex

FlaskDemo
Home About Contact

Alex
Member since 2023-05-15
Premium

Posts
42

Likes
128

Engagement
304.8%

Follow Message

© 2023 Flask Template Demo. All rights reserved.

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 personalizes 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.