Experiment - 5: Flask Application

Name of Student	**Ronak Katariya**
Class Roll No	**D15A / 23**
D.O.P.	
D.O.S.	
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 is used to render HTML templates stored in the templates folder. It dynamically generates web pages by passing variables from the Flask app to the template using Jinja2.

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

   - The templates folder is the default location where Flask looks for HTML files.
   - It maintains a clean separation between business logic (Python code) and presentation logic (HTML).
   - Using the templates folder allows developers to use Jinja2 for rendering dynamic content.

- The folder can also store reusable components like base templates, headers, or footers using template inheritance.

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

   Jinja2 is a templating engine used in Flask to render dynamic HTML content. It allows embedding Python expressions inside HTML files. Using Jinja2, you can:
   - Display variables
   - Apply logic (like loops and conditionals)
   - Apply filters for formatting

   Flask integrates Jinja2 by default using the render_template()function.

OUTPUT

- **app.py**

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


app = Flask(__name__)


@app.route('/')
def home():
    return '''<h1>Welcome to the Flask App!</h1>
              <p>Hello Ronak!</p>
              <p><a href="/contact">Go to Contact Page</a></p>
              <p><a href="/user/Ronak">Visit Ronak's Profile</a></p>'''


@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        name = request.form['name']
```

```
  email = request.form['email']
        return redirect(url_for('thank_you', name=name, email=email))
    return '''<form method="post">
                Name: <input type="text" name="name" required><br>
                Email: <input type="email" name="email" required><br>
                <input type="submit" value="Submit">
            </form>'''

@app.route('/thank_you')
def thank_you():
    name = request.args.get('name')
    email = request.args.get('email')
    return f'<h1>Thank You!</h1><p>Name: {name}</p><p>Email: {email}</p>'

@app.route('/user/<username>')
def user_profile(username):
    return render_template('user.html', username=username)

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

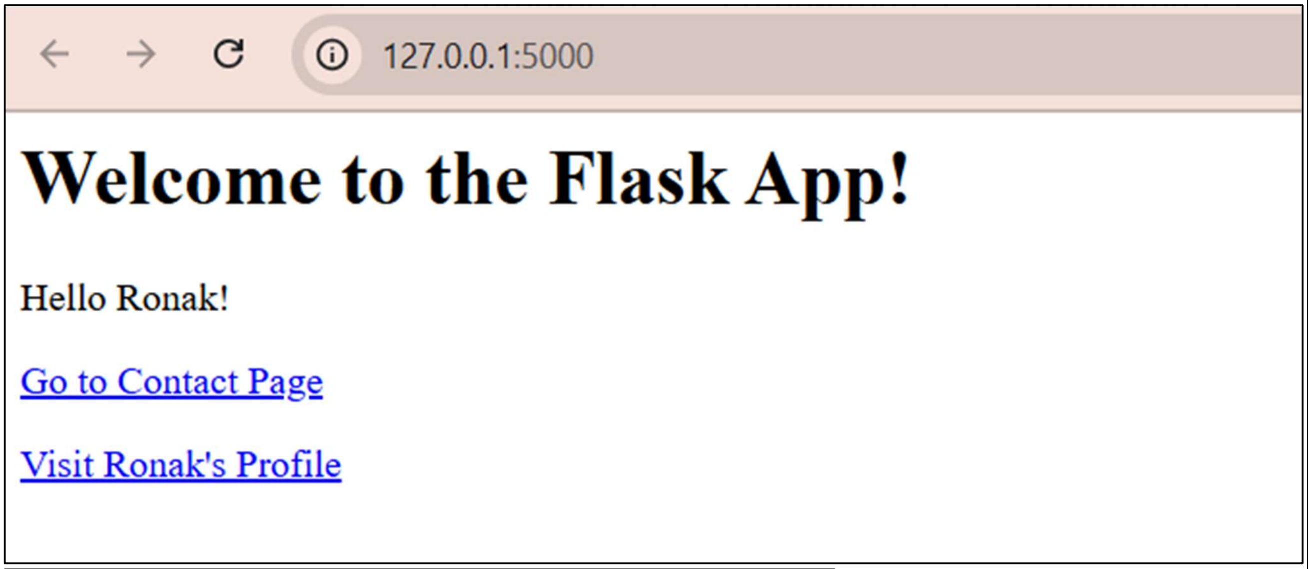
- **user.html**

```
<!-- templates/user.html -->
<!DOCTYPE html>
<html>
<head>
   <title>User Profile</title>
</head>
<body>
   <h1>Hello, {{ username }}!</h1>

   {% if username == 'Ronak' %}
      <p>Welcome back, Ronak! You're awesome!</p>


 {% else %}
      <p>Nice to meet you, {{ username }}.</p>
   {% endif %}

   <a href="/">Go back to Home</a>
</body>
</html>
```

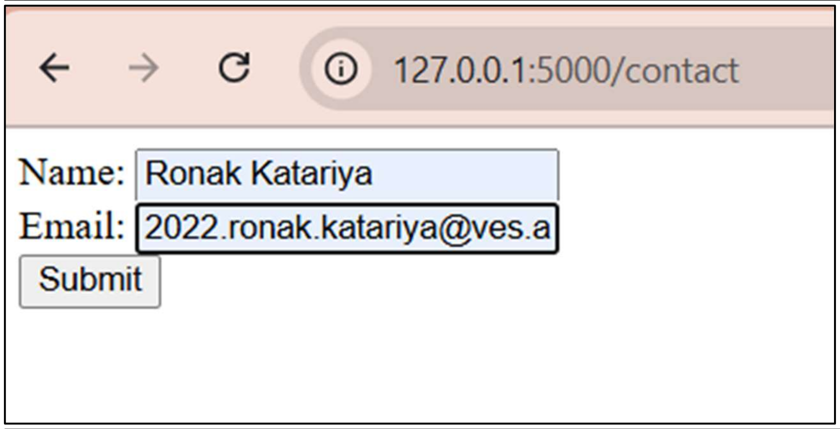

127.0.0.1:5000

Welcome to the Flask App!

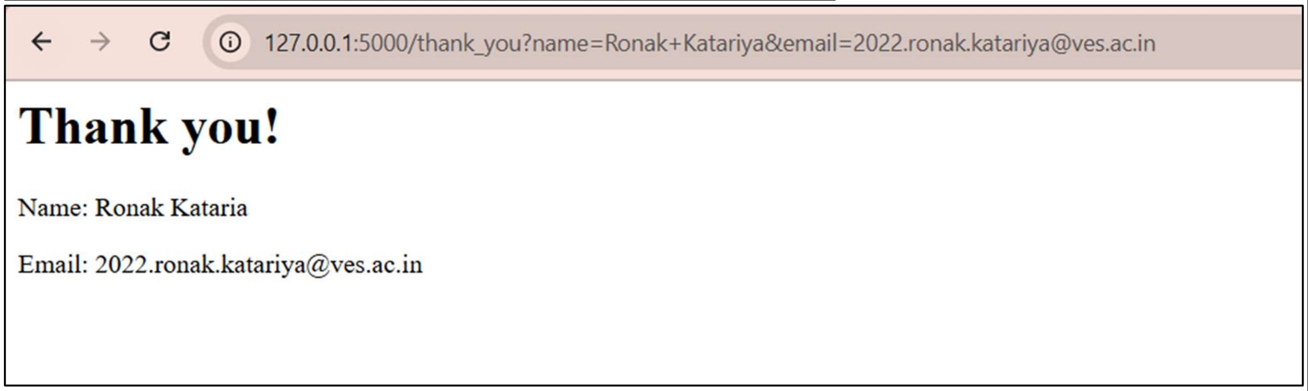
Hello Ronak!

Go to Contact Page

Visit Ronak's Profile


127.0.0.1:5000/contact

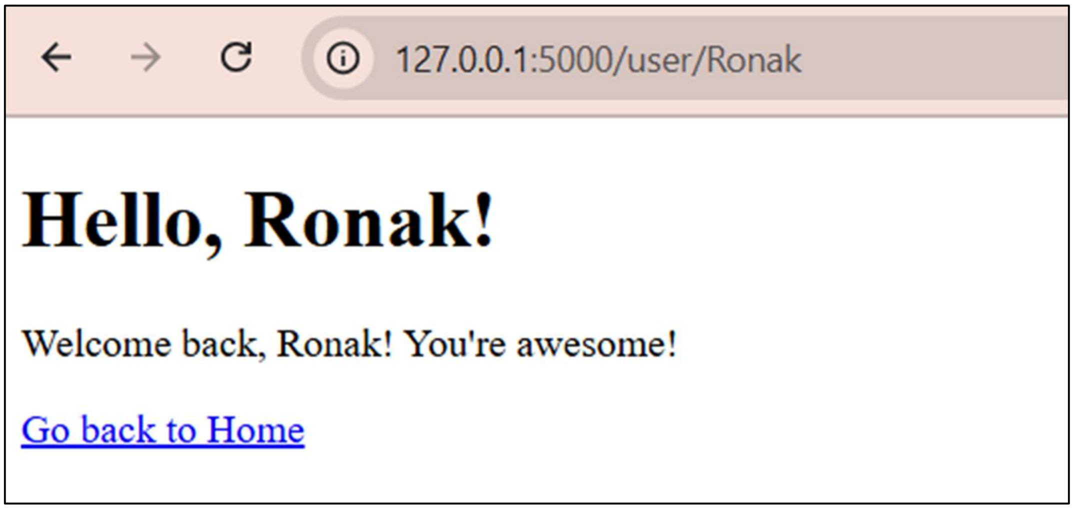
Name: Ronak Katariya
Email: 2022.ronak.katariya@ves.a
Submit


127.0.0.1:5000/thank_you?name=Ronak+Katariya&email=2022.ronak.katariya@ves.ac.in

Thank you!

Name: Ronak Kataria

Email: 2022.ronak.katariya@ves.ac.in



127.0.0.1:5000/user/Ronak

Hello, Ronak!

Welcome back, Ronak! You're awesome!

Go back to Home