EXPERIMENT NO. 5

Name of Student	**Pranav Pol**
Class Roll No	**D15A / 41**
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?
2. What is the significance of the templates folder in a Flask project?
3. What is Jinja2, and how does it integrate with Flask?

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
<html>
<body>
    <h1>Hello, {{ name }}!</h1>
    {% if name == "Admin" %}
        <p>Welcome back, Admin!</p>
    {% else %}
        <p>Enjoy your visit, {{ name }}!</p>
    {% endif %}
</body>
</html>
```

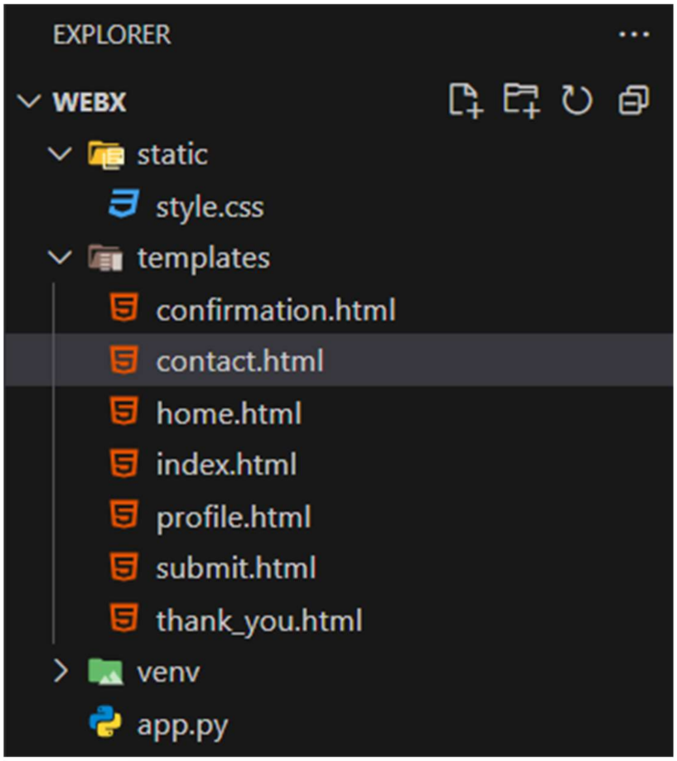
- Flask automatically integrates **Jinja2** when using render_template().

Output:

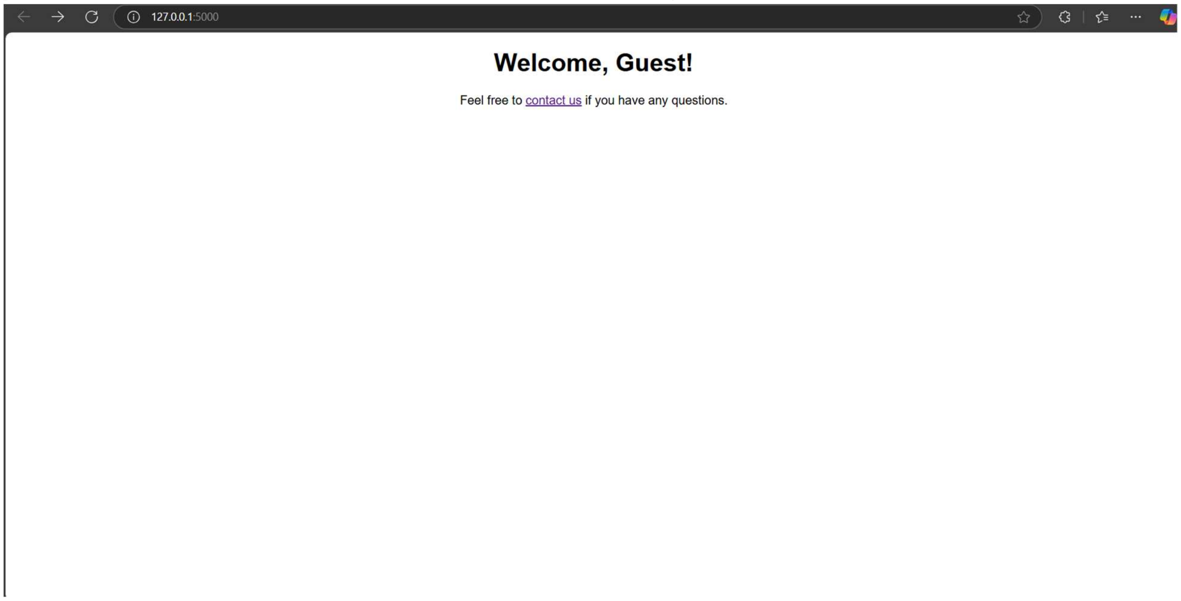
```python
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)

@app.route('/')
def home():
    name = request.args.get('name', 'Guest')
    return render_template('home.html', name=name)

@app.route('/home')
def home_redirect():
    return render_template('index.html')

@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 render_template('contact.html')

@app.route('/thank_you')
def thank_you():
    name = request.args.get('name', 'Anonymous')
    email = request.args.get('email', 'No Email Provided')
    return render_template('thank_you.html', name=name, email=email)

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

@app.route('/submit', methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        name = request.form['name']
        age = request.form['age']
        return render_template('confirmation.html', name=name, age=age)
    return render_template('submit.html')

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

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 Us</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h1>Contact Us</h1>
    <form method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

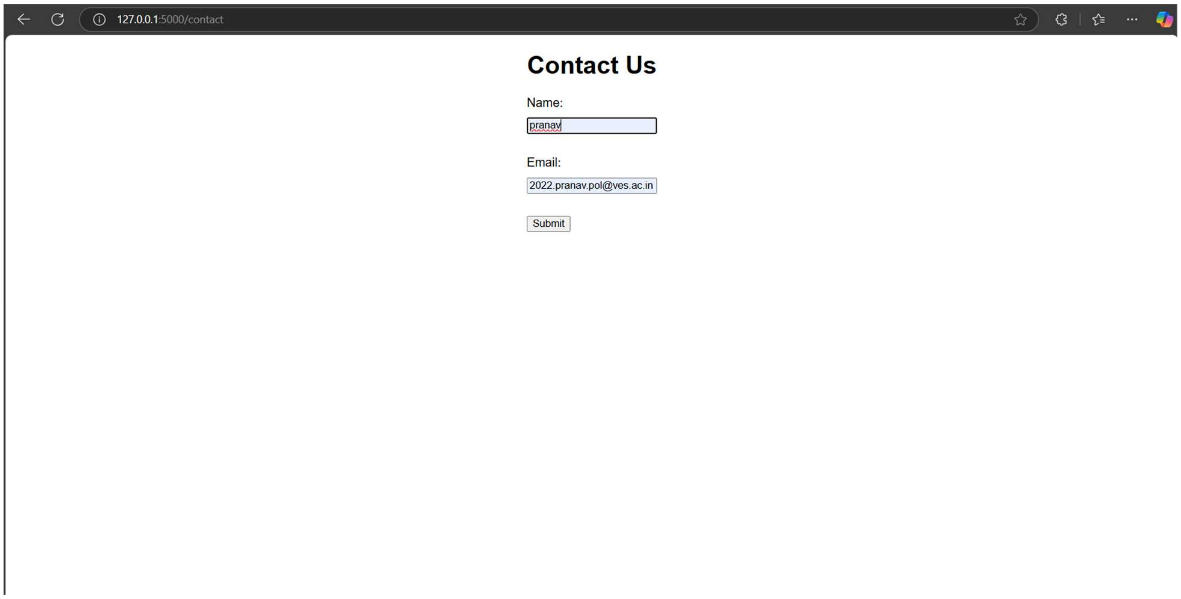


```
WEBX
  static
    style.css
  templates
    confirmation.html
    contact.html
    home.html
    index.html
    profile.html
    submit.html
    thank_you.html
  venv
  app.py
```



Welcome, Guest!

Feel free to contact us if you have any questions.



127.0.0.1:5000/contact

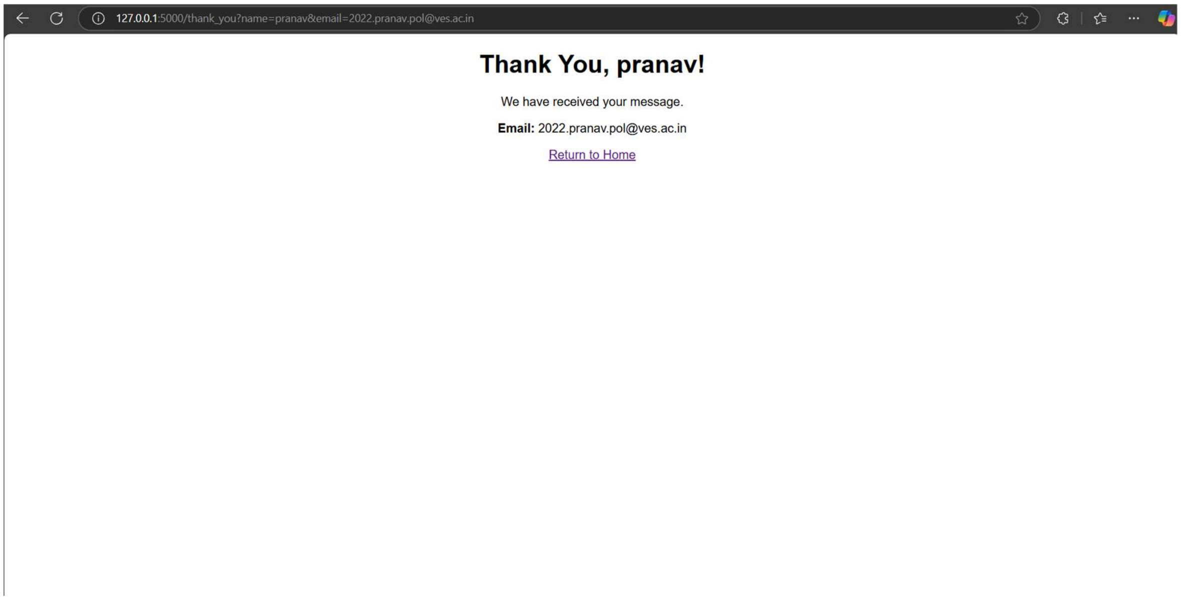
Contact Us

Name:

pranav

Email:

2022.pranav.pol@ves.ac.in



127.0.0.1:5000/thank_you?name=pranav&email=2022.pranav.pol@ves.ac.in

Thank You, pranav!

We have received your message.

Email: 2022.pranav.pol@ves.ac.in

Return to Home