EXPERIMENT NO. 3

Name of Student	**Pranav Pol**
Class Roll No	**D15A / 41**
D.O.P.	
D.O.S.	
Sign and Grade	

AIM : To develop a basic Flask application with multiple routes and demonstrate the handling of GET and POST requests.

PROBLEM STATEMENT :

Design a Flask web application with the following features:

1. A homepage (/) that provides a welcome message and a link to a contact form.
    a. Create routes for the homepage (/), contact form (/contact), and thank-you page (/thank_you).
2. A contact page (/contact) where users can fill out a form with their name and email.
3. Handle the form submission using the POST method and display the submitted data on a thank-you page (/thank_you).
    a. On the contact page, create a form to accept user details (name and email).
    b. Use the POST method to handle form submission and pass data to the thank-you page
4. Demonstrate the use of GET requests by showing a dynamic welcome message on the homepage when the user accesses it with a query parameter, e.g., /welcome?name=<user_name>.
    a. On the homepage (/), use a query parameter (name) to display a personalized welcome message.

Theory:

A. List some of the core features of Flask

Lightweight and Flexible: Flask is a micro-framework with minimal boilerplate, making it easy to use and extend.
Routing: It allows easy URL routing and handling of HTTP requests.
Template Engine (Jinja2): For dynamic HTML rendering.
Development Server: Includes a built-in server for testing and development.

RESTful Request Handling: Easily build REST APIs.
Extensible: You can integrate with many libraries and extensions for database handling, form validation, authentication, etc.

B. Why do we use Flask(__name__) in Flask?

`Flask(__name__)` creates an instance of the Flask class. The argument `__name__` tells Flask where to find resources (templates, static files, etc.). It helps the app locate the appropriate files and resources relative to the location of the current file (app.py).

C. What is Template (Template Inheritance) in Flask?

Template: A file containing HTML with placeholders for dynamic content. Flask uses Jinja2 as the template engine to render dynamic data into HTML.

Template Inheritance: Allows you to define a base template with common structure (e.g., header, footer) and extend it in child templates. This helps in code reuse and maintainability.

```
{% extends "base.html" %}

{% block content %}

  <h1>Welcome to my website</h1>

{% endblock %}
```

D. What methods of HTTP are implemented in Flask.

Flask supports all common HTTP methods:

GET: Retrieve data from the server (default for forms and links).
POST: Submit data to the server (used for form submissions).
PUT: Update data on the server.
DELETE: Remove data from the server.
PATCH: Partially update data on the server.
OPTIONS: Describe communication options for the target resource.

E. What is difference between Flask and Django framework

Flask:

- Lightweight, minimal, and flexible.
- No built-in admin panel or ORM.
- More freedom in structure and libraries used.
- Suitable for small to medium-sized applications or APIs.

Django:

- Full-stack framework, includes everything (ORM, admin panel, authentication, etc.).
- Heavier and opinionated, with a defined project structure.

- Suitable for larger applications where all components are required out of the box.

Routing

URL building

GET REQUEST

POST REQUEST

OUTPUT

```
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

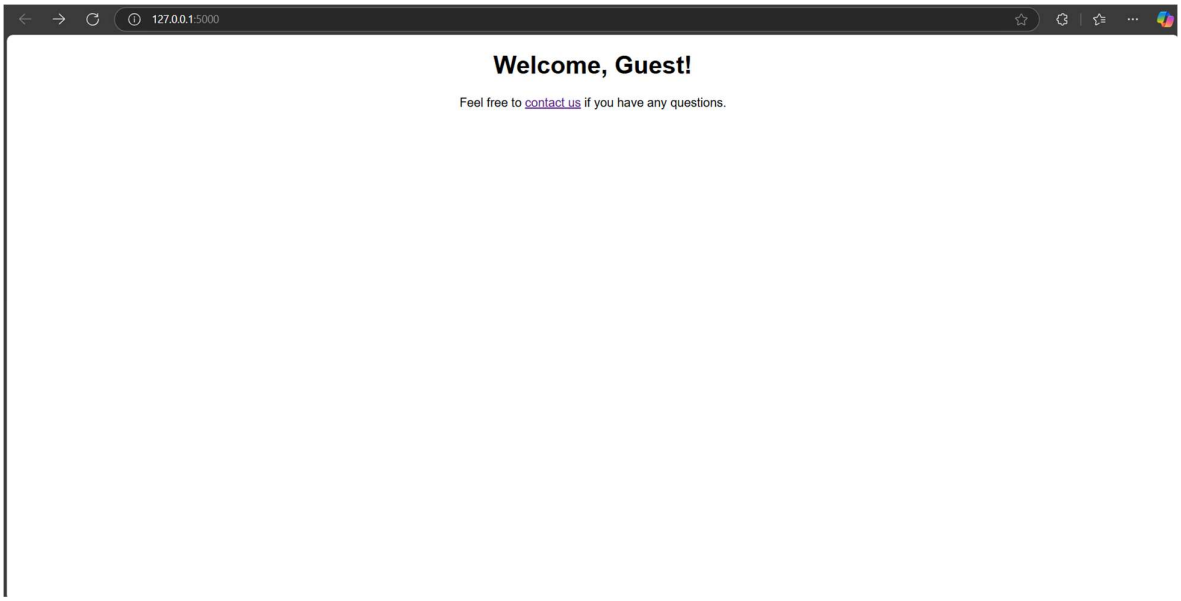
```
<!DOCTYPE html>
<html lang="en">
```

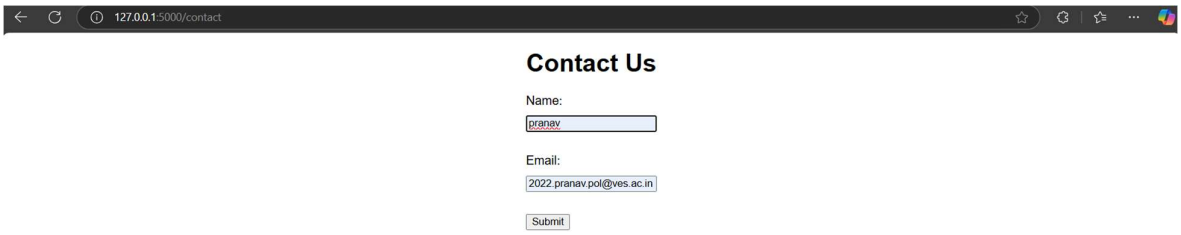
```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h1>Contact Us</h1>
    <form method="post">
        <label 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>
```

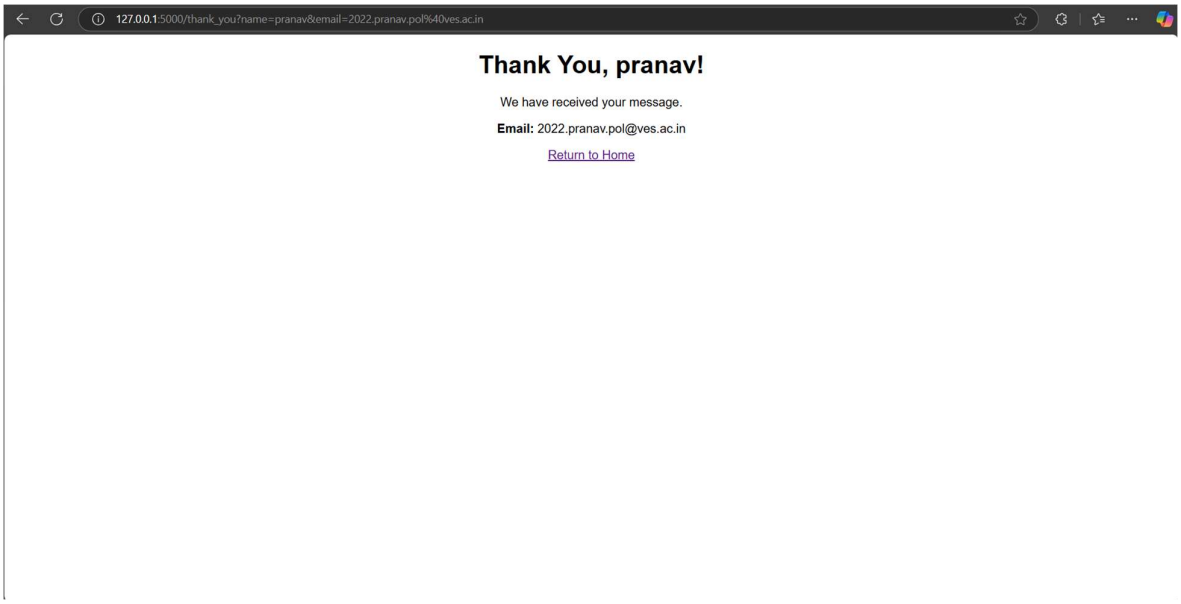
Thankyou.html

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

```
</head>
<body>
    <h1>Thank You, {{ name }}!</h1>
    <p>We have received your message.</p>
    <p><strong>Email:</strong> {{ email }}</p>
    <a href="{{ url_for('home') }}">Return to Home</a>
</body>
</html>
```




127.0.0.1:5000/contact
Contact Us
Name:
pranav
Email:
2022.pranav.pol@ves.ac.in
Submit


127.0.0.1:5000/thank_you?name=pranav&email=2022.pranav.pol%40ves.ac.in
Thank You, pranav!
We have received your message.
Email: 2022.pranav.pol@ves.ac.in
Return to Home