Experiment 3

Name of Student	Swaraj Patil
Class Roll No	**D15A_39**
D.O.P.	**06/02/2025**
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 :-

1) List some of the core features of Flask

- Lightweight and Minimalistic: Flask is a micro-framework, meaning it has no built-in dependencies, making it lightweight.
- Built-in Development Server and Debugger: Makes debugging easier during development.
- Integrated Jinja2 Templating: For dynamic HTML rendering.
- RESTful Request Handling: Flask simplifies the creation of RESTful APIs.
- Extensible: Easy to add extensions for database support, authentication, etc.
- URL Routing: Maps URLs to Python functions effortlessly.
-

2) Why do we use Flask(__name__) in Flask?

Flask(__name__) creates an instance of the Flask application.

- The __name___variable helps Flask identify the application's root directory, making it possible to locate templates, static files, etc.
- It also helps in determining if the app is running directly or is being imported as a modul

Example

```
from flask import Flask

app = Flask(__name__)
```

3) What is Template (Template Inheritance) in Flask?

Templates are used to create dynamic HTML pages in Flask. Flask uses the Jinja2 templating engine, which allows the use of variables, loops, and conditions within HTML files.

Template Inheritance:

This is a way to reuse a base template (like a master layout) and extend it in other templates. It helps maintain consistency and reduces code duplication.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>{% block title %}My Website{% endblock %}</title>
</head>
<body>
   {% block content %}{% endblock %}
</body>
</html>
```

HTTP Methods Implemented in Flask:

Flask supports the following HTTP methods:

1. **GET**: To retrieve data from the server.
2. **POST**: To submit data to the server.
3. **PUT**: To update existing resources.
4. **DELETE**: To delete resources from the server.

Example:

```
from flask import Flask, request

app = Flask(__name__)



@app.route('/submit', methods=['GET', 'POST'])

def submit_data():

    if request.method == 'POST':

        return "Data Submitted via POST"

    return "GET Request - Submit Page"
```

4) Difference Between Flask and Django Framework:

Feature	Flask	Django
Type	Micro-framework (Lightweight)	Full-stack framework
Flexibility	Highly flexible, allows customization	Follows "convention over configuration"
Built-in Admin	Not available	Comes with a powerful admin panel
Database Support	No default ORM, but can use SQLAlchemy	Built-in ORM (Object-Relational Mapping)
Learning Curve	Easier for beginners and small projects	Steeper due to its vast features
Template Engine	Jinja2	Django Template Language

- **When to use Flask:** When you need a lightweight, flexible framework for small to medium projects.
- **When to use Django:** For complex, large-scale projects requiring built-in features like authentication, admin panels, etc.

Code

```
from flask import Flask, request, render_template_string


# Initialize the Flask application
app = Flask(__name__)


# Homepage route with personalized greeting using GET query parameter
@app.route('/')
def home():
    name = request.args.get('name')  # Fetch 'name' parameter from the URL
    if name:
        return f'<h1>Welcome, {name}!</h1><p><a href="/contact">Go to the
Contact Form</a></p>'
    return '''
        <h1>Welcome to Our Homepage!</h1>
        <p>Please provide your name to get a personalized greeting: </p>
        <form action="/" method="get">
            <label for="name">Enter your name:</label><br>
            <input type="text" name="name" id="name" required><br><br>
            <input type="submit" value="Get Greeting">
        </form>
    '''


# Contact form route
@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        # Retrieve user input from the form
        name = request.form['name']
        email = request.form['email']
        # Redirect to thank-you page with POSTed data
        return f'<h1>Thank you for your submission, {name}!</h1><p>Your
email: {email}</p>'

    return '''
        <h1>Contact Us</h1>
        <form method="POST">
            <label for="name">Name:</label><br>
```

```
            <input type="text" id="name" name="name" required><br><br>
            <label for="email">Email:</label><br>
            <input type="email" id="email" name="email" required><br><br>
            <input type="submit" value="Submit">
        </form>
        <p><a href="/">Go to Home Page</a></p>
    '''


# Thank-you page route
@app.route('/thank_you')
def thank_you():
    return '''
        <h1>Thank You!</h1>
        <p>Your submission has been received. We will get in touch with
you soon!</p>
        <p><a href="/">Go back to Home</a></p>
    '''


# Run the application
if __name__ == '__main__':
    app.run(debug=True)
```

Output

Welcome to Our Homepage!

Please provide your name to get a personalized greeting:

Enter your name:
bhagyesh

Get Greeting

Welcome, bhagyesh!

Go to the Contact Form

Connect with Us

Full Name:

Email Address:

Send Message

Return Home

Appreciate Your Message, bhagyesh!

We will reach out to you at d2022.bhagyesh.patil@ves.ac.in soon.