# EXPERIMENT NO. 3

**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 simple to use
- Built-in development server and debugger
- Support for URL routing
- Templating using Jinja2
- RESTful request handling
- Support for unit testing
- Extensions for database integration, authentication, etc.

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

Flask(__name__) initializes the Flask application using the name of the module. It helps Flask determine the root path of the application, which is essential for templates, static files, and routing.

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

Template inheritance in Flask uses Jinja2 to create a base HTML template that other templates can inherit. This avoids code repetition and ensures a clean and maintainable code structure.

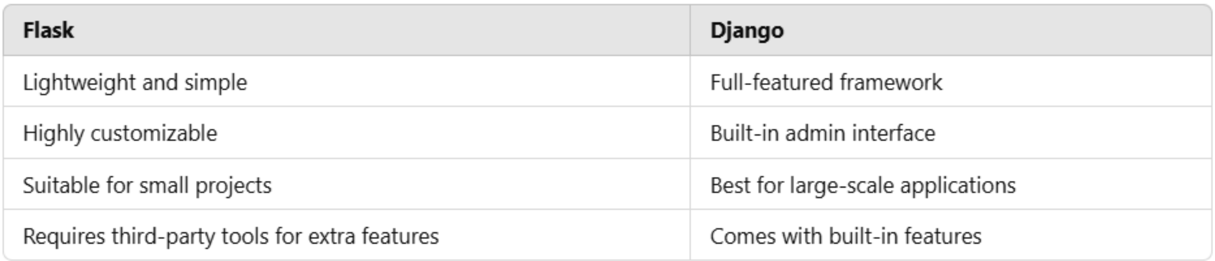
Example:

- base.html: Contains common HTML structure (header, footer).
- Other templates extend it using {% extends 'base.html' %}.

D. What methods of HTTP are implemented in Flask.

- **GET**: Requests data from the server without modifying it.
- **POST**: Sends data to the server for processing.
- **PUT**: Updates existing data.
- **DELETE**: Deletes data from the server.

E. What is difference between Flask and Django framework



| Flask | Django |
|---|---|
| Lightweight and simple | Full-featured framework |
| Highly customizable | Built-in admin interface |
| Suitable for small projects | Best for large-scale applications |
| Requires third-party tools for extra features | Comes with built-in features |

Routing

URL building

GET REQUEST

POST REQUEST

**CODE:-**

**App.py**

```
from flask import Flask, request, render_template


app = Flask(__name__)
# Homepage route
@app.route('/')
def home():
    name = request.args.get('name', 'Guest')
    return f"Welcome, {name}! <br><a href='/contact'>Go to Contact Form</a>"
# Contact form route
@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        return render_template('thank_you.html', name=name, email=email)
    return render_template('contact.html')
# Thank You route
@app.route('/thank_you')
def thank_you():
    return "Thank you for submitting your details!"


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

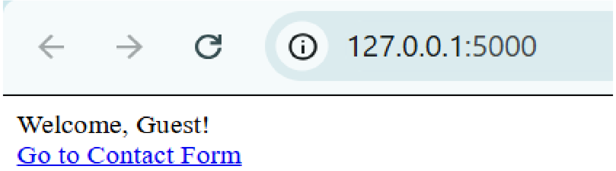
**contact.html**

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Form</title>
</head>
<body>
    <h1>Contact Us</h1>
    <form method="POST">
        Name: <input type="text" name="name" required><br>
        Email: <input type="email" name="email" required><br>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

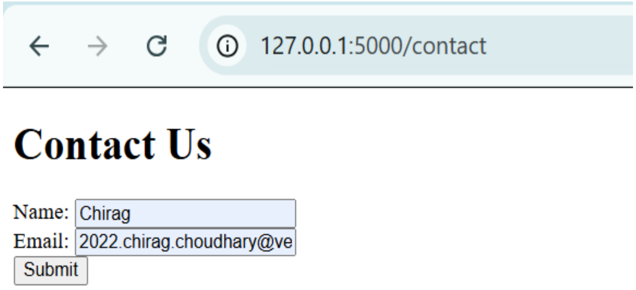
**Thank_you.html**

```
<!DOCTYPE html>

<html>

<head>

    <title>Thank You</title>

</head>

<body>

    <h1>Thank You, {{ name }}</h1>

    <p>We have received your email: {{ email }}</p>

</body>

</html>
```

**OUTPUT**

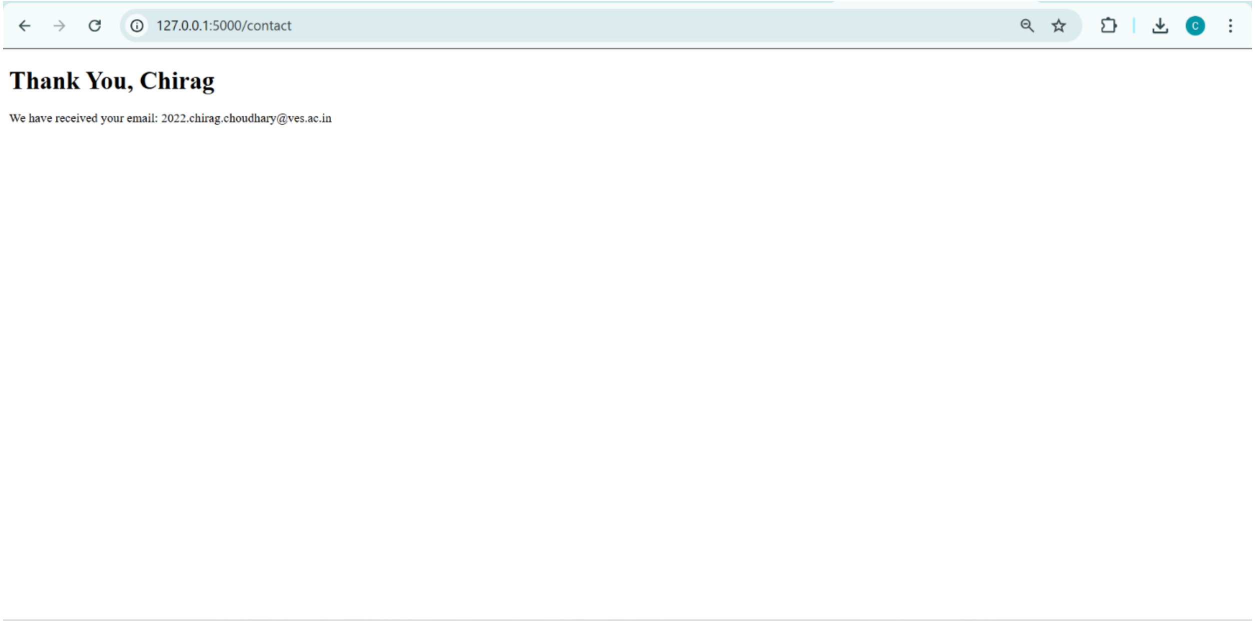


Welcome, Guest!
Go to Contact Form



# Contact Us

Name: Chirag
Email: 2022.chirag.choudhary@ve



# Thank You, Chirag

We have received your email: 2022.chirag.choudhary@ves.ac.in