**EXPERIMENT NO. 3**

|  |  |
| --- | --- |
| **Name of Student** | **Vanshika Ambwani** |
| **Class Roll No** | **D15A 2** |
| **DOP** |  |
| **DOS** |  |
| **Sign and Grade** |  |

**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.
   1. 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).
   1. On the contact page, create a form to accept user details (name and email).
   2. 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>.
   1. On the homepage (/), use a query parameter (name) to display a personalized welcome message.

**Theory:**

A.List some of the core features of Flask

Flask is a lightweight and flexible web framework for Python. Some of its core features include:

* **Minimal and Lightweight** – Flask is a microframework with no built-in ORM or authentication system.
* **Built-in Development Server & Debugger** – Helps in testing applications easily.

* **RESTful Request Handling** – Supports multiple HTTP methods (GET, POST, etc.).

* **Jinja2 Templating Engine** – Supports dynamic HTML generation using templates.

* **URL Routing** – Provides a simple way to define application URLs.

* **Session and Cookie Management** – Supports user sessions and cookies.
* **Extensibility** – Can be extended with Flask extensions for database handling, authentication, etc.
* **WSGI Support** – Built on the WSGI standard for web applications. B.Why do we use Flask(\_\_name\_\_) in Flask? When initializing a Flask application, we write:

app = Flask(\_\_name\_\_)

* \_\_name\_\_ represents the name of the current module.
* Flask uses this to determine the root path of the application.
* It helps locate resources like templates and static files. ● It also helps in debugging and logging.

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

Flask uses **Jinja2**, a templating engine, to separate logic from presentation.

* A **template** is an HTML file with placeholders for dynamic content.

* **Template Inheritance** allows reusing code by defining a base template and extending it in child templates.

Example:

**Base Template (base.html)**

<!DOCTYPE html>

<html>

<head>

<title>{% block title %}Default Title{% endblock %}</title>

</head>

<body>

<header>Flask Application</header>

<main>{% block content %}Default Content{% endblock %}</main>

</body>

</html>

**Child Template (home.html)**

{% extends 'base.html' %}

{% block title %}Home Page{% endblock %}

{% block content %}

<h1>Welcome to Flask</h1>

{% endblock %}

D.What methods of HTTP are implemented in Flask.

Flask supports various HTTP methods:

1. **GET** – Requests data from a server (default method).

1. **POST** – Sends data to the server (used in forms).

1. **PUT** – Updates an existing resource.

1. **DELETE** – Deletes a resource.

1. **PATCH** – Partially updates a resource.

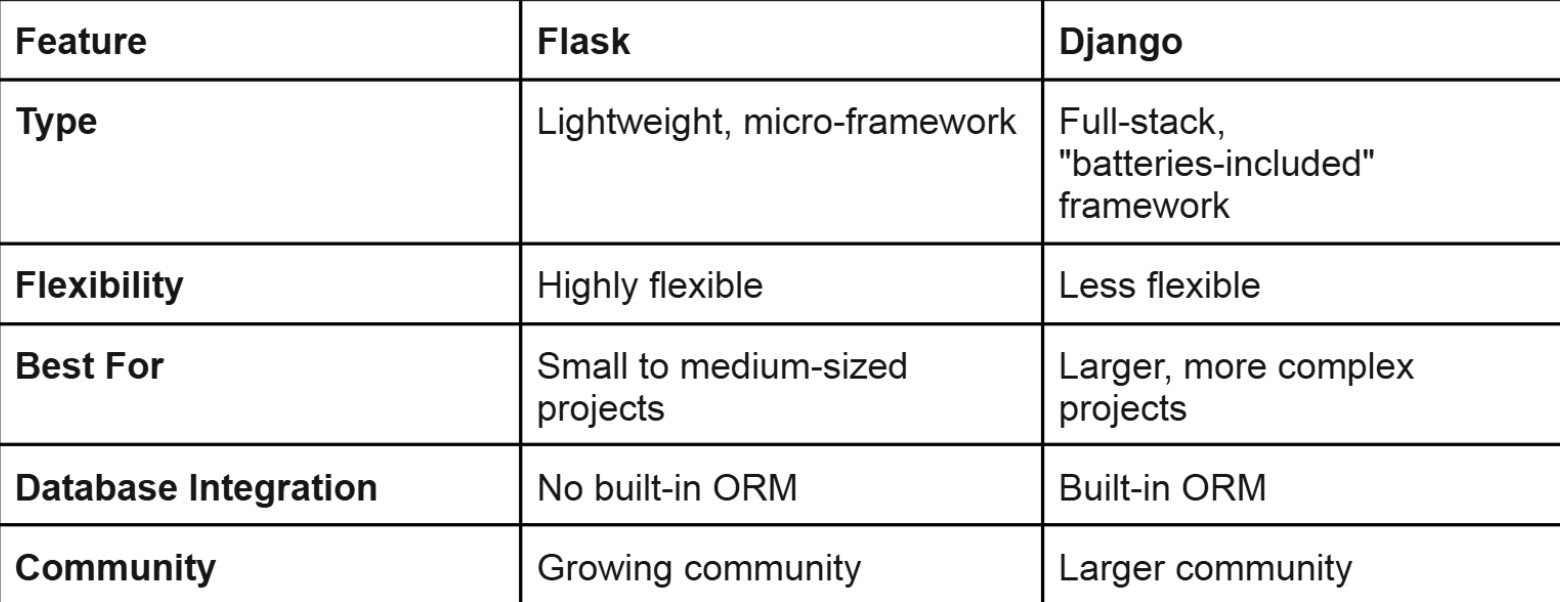
1. **HEAD** – Similar to GET but retrieves only headers, not content.

1. **OPTIONS** – Returns HTTP methods supported by the server.

Example:

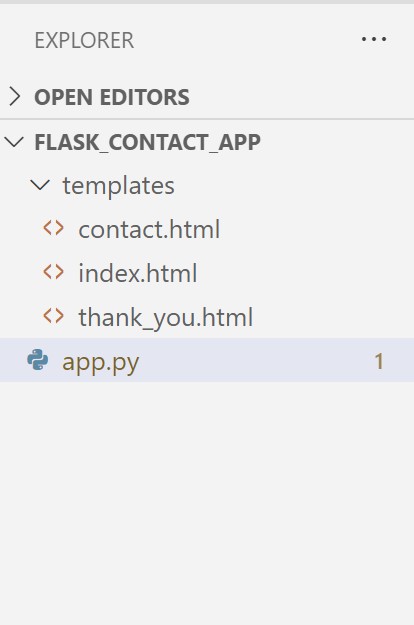
@app.route('/submit', methods=['GET', 'POST']) def submit(): if request.method == 'POST': return "Data Submitted!" return "Submit Form"

E.What is difference between Flask and Django framework



**Code:**

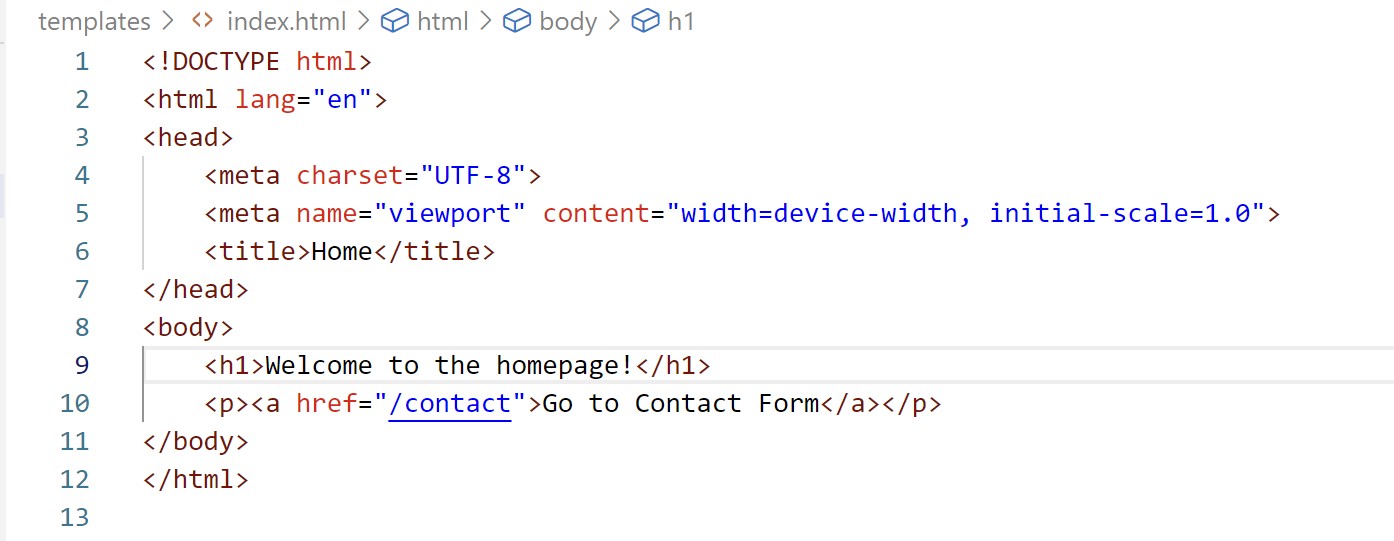
**Folder structure:**



# app.py



# index.html



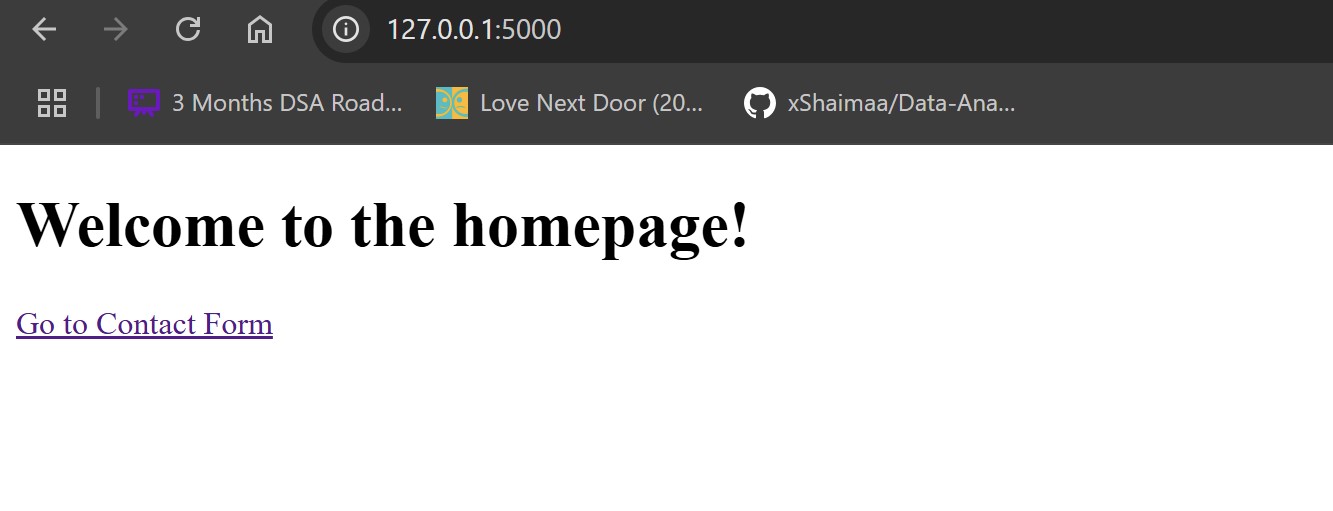
# contact.html

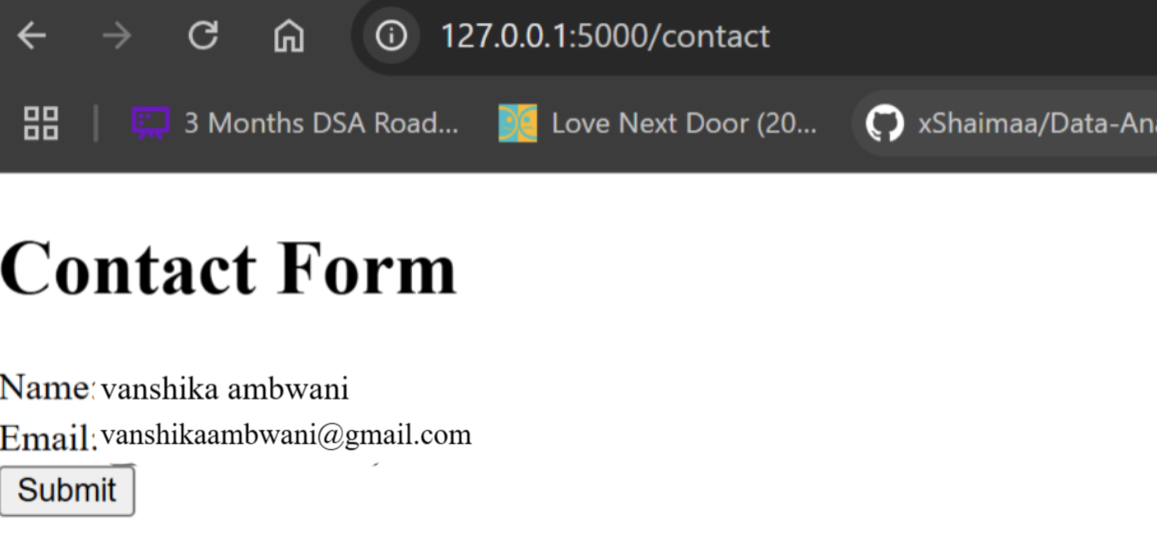


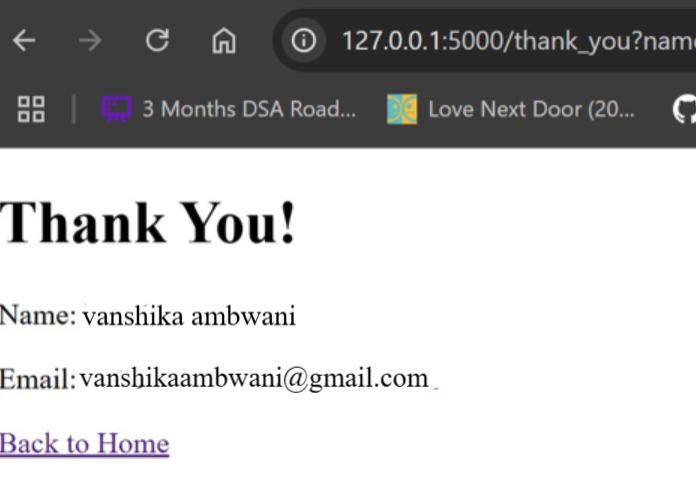
# thank\_you.html



# OUTPUT



****

****