

### **EXPERIMENT NO. 3- Flask**

<b>Name of Student</b>	<b>Eesha Chavan</b>
<b>Class Roll No</b>	<b>D15A 08</b>
<b>D.O.P.</b>	
<b>D.O.S.</b>	
<b>Sign and Grade</b>	

#### **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:**

##### **List some of the core features of Flask**

Flask is a lightweight and flexible web framework for Python, often described as "micro" because it provides the essentials to get a web application up and running without imposing unnecessary restrictions. Some core features of Flask include its minimalist design, which allows developers to add only what they need, built-in support for routing, templates, and handling HTTP requests and

responses. It uses the Jinja2 templating engine to generate dynamic HTML content and can be extended with various extensions like database support, authentication, and form handling.

### Why do we use Flask(\_name\_) in Flask?

The reason we use `Flask(_name_)` is to create an instance of the Flask class. `_name_` tells Flask if the script is being run directly or if it's being imported as a module into another script. When running directly, Flask knows to start the app, and when imported, it avoids running the app code unnecessarily.

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

In Flask, a template is an HTML file with placeholders for dynamic content, usually populated with data passed from the Flask view functions. Template inheritance allows you to create a base template (with common elements like headers or footers) and extend it in other templates. This promotes reusability and helps keep your code DRY (Don't Repeat Yourself).

### What methods of HTTP are implemented in Flask.

Flask supports several HTTP methods like GET (to retrieve data), POST (to send data), PUT (to update existing data), DELETE (to remove data), and OPTIONS (to describe communication options for the resource). These methods allow Flask to handle different types of interactions with the server.

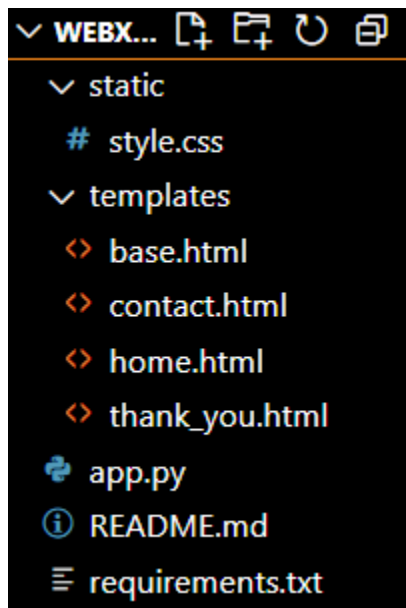
### What is difference between Flask and Django framework

Feature	Flask	Django
Type	Lightweight, micro-framework	Full-stack, "batteries-included" framework
Flexibility	Highly flexible	Less flexible
Best For	Small to medium-sized projects	Larger, more complex projects
Database Integration	No built-in ORM	Built-in ORM
Community	Growing community	Larger community

GitHub Link: [https://github.com/eeshachavan/WebX\\_exp3\\_eesha](https://github.com/eeshachavan/WebX_exp3_eesha)

CODE:

Folder Structure:



App.py:

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

app = Flask(__name__)

@app.route('/')
def home():
    # Check if name parameter exists in the URL
    user_name = request.args.get('name')
    return render_template('home.html', user_name=user_name)

@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        # Get form data
        name = request.form.get('name')
        email = request.form.get('email')

        # Redirect to thank you page with form data
        return redirect(url_for('thank_you', name=name, email=email))

    # If it's a GET request, just render the contact form
    return render_template('contact.html')
```

```

@app.route('/thank_you')
def thank_you():
    # Get data from query parameters
    name = request.args.get('name')
    email = request.args.get('email')

    return render_template('thank_you.html', name=name, email=email)

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

```

base.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Flask Web App{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
}}">
</head>
<body>
    <header>
        <nav>
            <div class="container">
                <h1>My Flask App</h1>
                <ul>
                    <li><a href="{{ url_for('home') }}">Home</a></li>
                    <li><a href="{{ url_for('contact') }}">Contact</a></li>
                </ul>
            </div>
        </nav>
    </header>

    <main class="container">
        {% block content %}{% endblock %}
    </main>

    <footer>
        <div class="container">
            <p>&copy; 2025 My Flask App</p>
        </div>
    </footer>
</body>

```

```
</html>
```

home.html:

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

{% block title %}Home - Flask Web App{% endblock %}

{% block content %}
    <section class="welcome">
        {% if user_name %}
            <h2>Welcome, {{ user_name }}!</h2>
            <p>Thanks for visiting our site with your personalized
greeting.</p>
        {% else %}
            <h2>Welcome to Our Website</h2>
            <p>This is a simple Flask web application.</p>
        {% endif %}

        <div class="cta-container">
            <p>Want to get in touch with us?</p>
            <a href="{{ url_for('contact') }}" class="cta-button">Contact
Us</a>
        </div>

    </section>
{% endblock %}
```

contact.html:

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

{% block title %}Contact Us - Flask Web App{% endblock %}

{% block content %}
    <section class="contact-form">
        <h2>Contact Us</h2>
        <p>Please fill out the form below to get in touch with us.</p>

        <form action="{{ url_for('contact') }}" method="POST">
            <div class="form-group">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required>
            </div>
        </form>
    </section>
{% endblock %}
```

```

        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>

        <button type="submit" class="submit-button">Submit</button>
    </form>
</section>
{% endblock %}

```

thank\_you.html:

```

{% extends 'base.html' %}

{% block title %}Thank You - Flask Web App{% endblock %}

{% block content %}
    <section class="thank-you">
        <h2>Thank You!</h2>

        <div class="submission-details">
            <h3>Your Submission Details:</h3>
            <p><strong>Name:</strong> {{ name }}</p>
            <p><strong>Email:</strong> {{ email }}</p>
        </div>

        <p>We appreciate you reaching out to us. We'll get back to you
soon!</p>

        <a href="{{ url_for('home') }}" class="back-button">Back to Home</a>
    </section>
{% endblock %}

```

style.css:

```

/* Base Styles */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {

```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
line-height: 1.6;
color: #333;
background-color: #f5f5f5;
}

.container {
  width: 90%;
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 20px;
}

/* Header & Navigation */
header {
  background-color: #4a6fa5;
  color: white;
  padding: 1rem 0;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

nav {
  display: flex;
}

nav .container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
}

nav h1 {
  font-size: 1.5rem;
}

nav ul {
  display: flex;
  list-style: none;
}

nav ul li {
  margin-left: 1.5rem;
}
```

```
nav a {
  color: white;
  text-decoration: none;
  font-weight: 500;
  transition: color 0.3s;
}

nav a:hover {
  color: #e0e0e0;
}

/* Main Content */
main {
  min-height: calc(100vh - 130px);
  padding: 2rem 0;
}

section {
  background-color: white;
  border-radius: 8px;
  padding: 2rem;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);
}

h2 {
  color: #4a6fa5;
  margin-bottom: 1rem;
}

h3 {
  color: #4a6fa5;
  margin: 1.5rem 0 0.5rem;
}

/* Welcome Section */
.welcome {
  text-align: center;
}

.cta-container {
  margin: 2rem 0;
}
```



```
.cta-button {
  display: inline-block;
  background-color: #4a6fa5;
  color: white;
  padding: 0.8rem 1.5rem;
  border-radius: 4px;
  text-decoration: none;
  font-weight: 500;
  margin-top: 0.5rem;
  transition: background-color 0.3s;
}

.cta-button:hover {
  background-color: #3a5980;
}

.info-box {
  background-color: #f0f5ff;
  border-left: 4px solid #4a6fa5;
  padding: 1rem;
  margin-top: 2rem;
  text-align: left;
  border-radius: 0 4px 4px 0;
}

.info-box code {
  background-color: #e0e0e0;
  padding: 0.2rem 0.4rem;
  border-radius: 3px;
  font-family: monospace;
}

/* Contact Form */
.contact-form {
  max-width: 600px;
  margin: 0 auto;
}

.form-group {
  margin-bottom: 1.5rem;
}

label {
  display: block;
```

```
margin-bottom: 0.5rem;
font-weight: 500;
}

input {
  width: 100%;
  padding: 0.8rem;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 1rem;
}

.submit-button {
  background-color: #4a6fa5;
  color: white;
  border: none;
  padding: 0.8rem 1.5rem;
  border-radius: 4px;
  cursor: pointer;
  font-size: 1rem;
  font-weight: 500;
  transition: background-color 0.3s;
}

.submit-button:hover {
  background-color: #3a5980;
}

/* Thank You Page */
.thank-you {
  text-align: center;
  max-width: 600px;
  margin: 0 auto;
}

.submission-details {
  background-color: #f0f5ff;
  border-radius: 6px;
  padding: 1.5rem;
  margin: 1.5rem 0;
  text-align: left;
}

.back-button {
```

```

        display: inline-block;
        background-color: #4a6fa5;
        color: white;
        padding: 0.8rem 1.5rem;
        border-radius: 4px;
        text-decoration: none;
        font-weight: 500;
        margin-top: 1.5rem;
        transition: background-color 0.3s;
    }

    .back-button:hover {
        background-color: #3a5980;
    }

    /* Footer */
    footer {
        background-color: #333;
        color: white;
        padding: 1rem 0;
        text-align: center;
    }

```

requirements.txt:

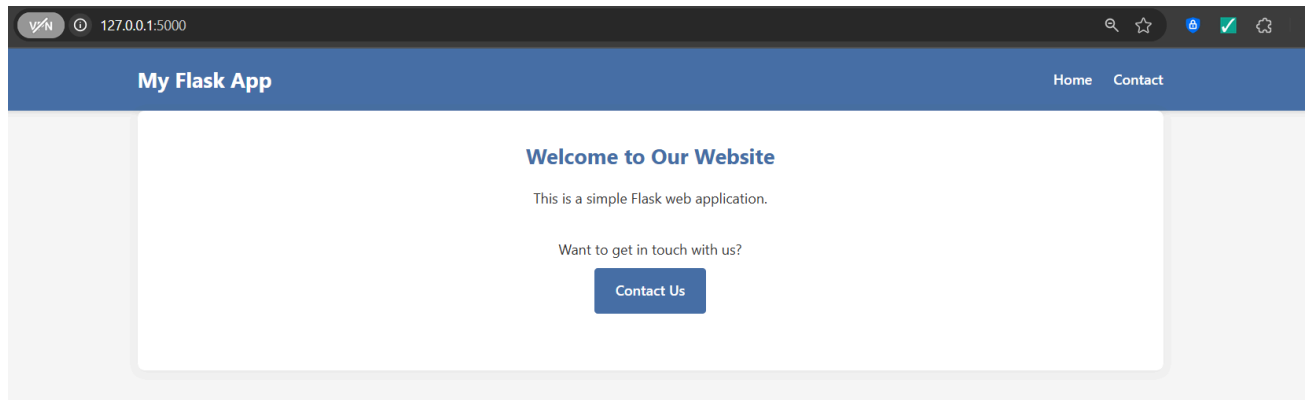
```

Flask==2.3.3
Werkzeug==2.3.7
Jinja2==3.1.2
MarkupSafe==2.1.3
itsdangerous==2.1.2
click==8.1.7

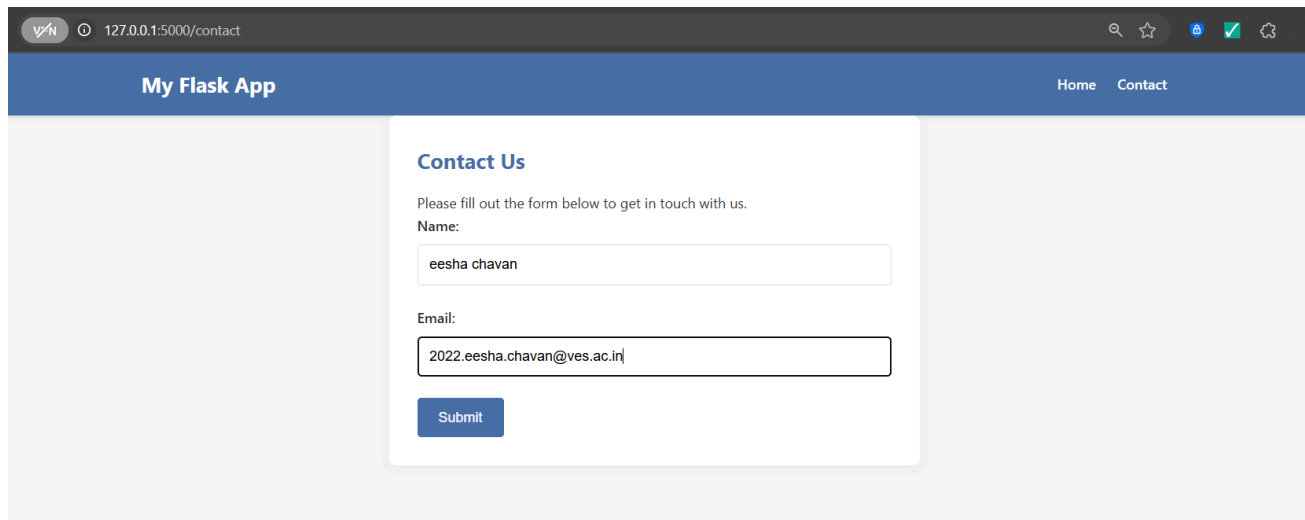
```

## OUTPUT :

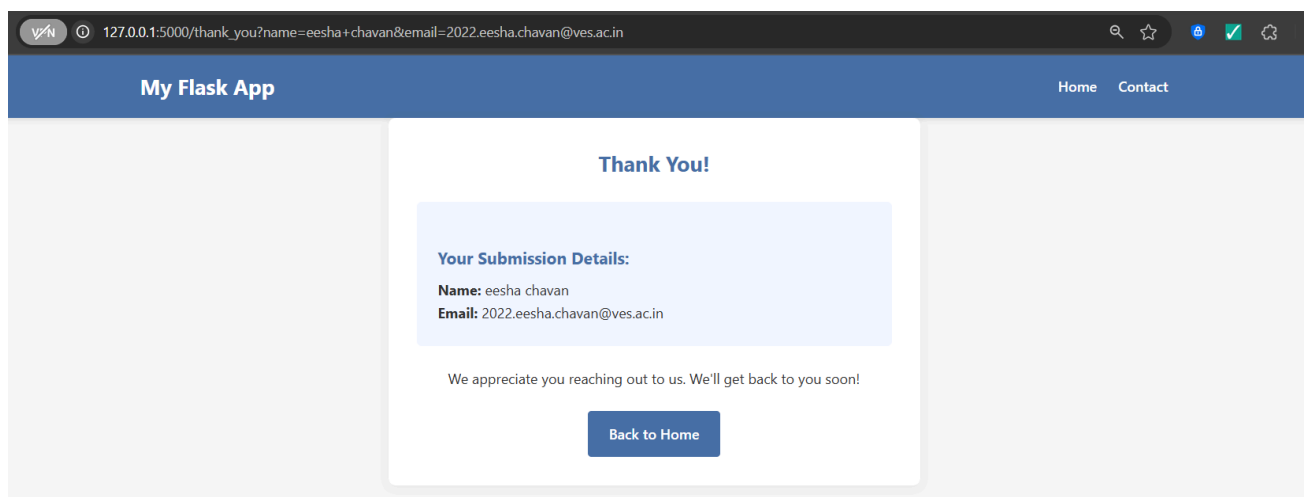
Successfully displayed the homepage with a personalized welcome message.



Displayed the contact form for user input.



Redirected to the thank-you page with user-submitted data.



## **CONCLUSION:**

The Flask application was successfully developed to demonstrate GET and POST request handling. The implementation included multiple routes, form submission, and dynamic rendering using templates. This experiment provided hands-on experience with Flask's core functionalities like routing, request handling, and template inheritance.