Flask

Flask

Flask Instatallion

1. Set Up Your Environment

- Install Python
- Create a virtual environment to keep dependencies isolated:

```
python -m venv venv_name
venv_name\Scripts\activate
```

Install Flask:

```
pip install flask
```

Create the Project Structure

Create a folder for your project. The structure should look like this:

```
flask_app/
├── app.py              # Main application
├── static/             # Static files like CSS, JS, images
├── templates/          # HTML templates
│   └── index.html      # Homepage template
└── requirements.txt    # Optional for listing dependencies
```

Write Your Flask App

```python
from flask import Flask, render_template

# Initialize Flask app
app = Flask(__name__)

# Define the homepage route
@app.route('/')
def home():
    return render_template('index.html')

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

- Key Points:
  - **@app.route('/')**: Defines the route (URL endpoint) for the homepage.
  - **render_template**: Loads the index.html file from the templates/ folder.
  - `debug=True` : Enables auto-reload and debugging (use only in development).

Create an HTML Template

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flask App</title>
</head>
<body>
    <h1>Welcome to My Flask App!</h1>
    <p>This is a simple Flask application.</p>
</body>
</html>
```

Run the app

```
python app.py
```

Add More Routes

```
@app.route('/about')
def about():
    return "<h1>About Page</h1><p>This is the about page.</p>"
```

Pass Data to HTML Templates

1. Update app.py:

```
@app.route('/user/<name>')
def user(name):
    return render_template('user.html', username=name)
```

2. Create user.html in the templates/ folder:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User Page</title>
</head>
<body>
    <h1>Hello, {{ username }}!</h1>
</body>
</html>
```

Handle Forms and User Input

- Add a route to handle forms:

```
from flask import request

@app.route('/submit', methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        name = request.form['name']
        return f"<h1>Welcome, {name}!</h1>"
    return render_template('form.html')
```

- Create form.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Form</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="name">Enter your name:</label>
        <input type="text" name="name" id="name">
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

URL : `http://127.0.0.1:5000/submit`

Use Static Files (CSS, JS, Images)

- Add a CSS file in the static/ folder:

```
static/style.css
```

Example for `style.css`:

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f9;
    color: #333;
    text-align: center;
}
```

- Link the CSS file in index.html:

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
```

Types of Requests in Flask

GET Request

- **Purpose**: Retrieve data from the server.
- **Implementation**:

```
@app.route('/get-example', methods=['GET'])
def get_example():
    return "This is a GET request"
```

- Usage: Visit http://127.0.0.1:5000/get-example in a browser.

POST Request

- **Purpose**: Send data to the server (e.g., form submissions).
- **Implementation**:

```
@app.route('/post-example', methods=['POST'])
def post_example():
    data = request.form['name']
    return f"Hello, {data}!"
```

```
<form action="/post-example" method="post">
    <input type="text" name="name" placeholder="Enter your name">
    <button type="submit">Submit</button>
</form>
```

PUT Request

- **Purpose**: Update existing resources.
- **Implementation**:

```
@app.route('/put-example', methods=['PUT'])
def put_example():
    data = request.json
    return f"Updated data: {data}"
```

DELETE Request

- **Purpose**: Delete a resource on the server.
- **Implementation**:

```
@app.route('/delete-example', methods=['DELETE'])
def delete_example():
    return "Resource deleted!"
```