**[Que-40] - Explain how to set up a Flask Application to handle form submissions using POST requests**

Setting up a Flask application to handle form submissions using POST requests involves creating a simple HTML form, defining routes to handle GET and POST requests, and processing the form data. Here's a step-by-step guide to achieve this:

### **Step 1: Install Flask**

If you haven't already installed Flask, you can install it using pip:

pip install flask

### **Step 2: Create the Flask Application**

Create a new Python file, e.g., app.py.

### **Step 3: Define Routes to Handle Form Submissions**

Here’s a basic example of setting up a Flask application to handle form submissions using POST requests:

from flask import Flask, request, render\_template\_string  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 return render\_template\_string('''  
 <form action="/submit" method="post">  
 <label for="name">Name:</label>  
 <input type="text" id="name" name="name"><br><br>  
 <input type="submit" value="Submit">  
 </form>  
 ''')  
  
@app.route('/submit', methods=['POST'])  
def submit():  
 name = request.form['name']  
 return f'Hello, {name}!'  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

### **Explanation:**

1. **Import Flask and Required Modules**:

from flask import Flask, request, render\_template\_string

* + Flask: The core Flask class.
  + request: Used to access form data.
  + render\_template\_string: Allows rendering HTML templates directly from a string.

1. **Create an Instance of the Flask Class**:

python

Copy code

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

1. **Define the Index Route**:

@app.route('/')  
def index():  
 return render\_template\_string('''  
 <form action="/submit" method="post">  
 <label for="name">Name:</label>  
 <input type="text" id="name" name="name"><br><br>  
 <input type="submit" value="Submit">  
 </form>  
 ''')

* + This route renders an HTML form that sends a POST request to the /submit route when submitted.

1. **Define the Submit Route**:

python

Copy code

@app.route('/submit', methods=['POST'])  
def submit():  
 name = request.form['name']  
 return f'Hello, {name}!'

* + This route handles POST requests sent to /submit.
  + request.form['name'] accesses the form data (the name entered in the input field).
  + The route returns a greeting message including the submitted name.

1. **Run the Application**:

if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

* + This block ensures the application runs in debug mode, providing helpful error messages and automatic reloading during development.

### **Step 4: Running the Flask Application**

1. Open a terminal or command prompt.
2. Navigate to the directory where app.py is located.
3. Run the following command:

python app.py

1. Open a web browser and go to <http://127.0.0.1:5000/>.

You should see the HTML form. When you enter a name and submit the form, the server will process the POST request and display a greeting message.

### **Using HTML Templates**

For larger applications, it’s better to separate HTML templates from your Python code. Here’s an example using an HTML template file:

1. **Create a Template Directory**: Create a directory named templates in the same directory as app.py.
2. **Create an HTML Template**: Create a file named index.html inside the templates directory with the following content:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Form Submission</title>  
</head>  
<body>  
 <form action="/submit" method="post">  
 <label for="name">Name:</label>  
 <input type="text" id="name" name="name"><br><br>  
 <input type="submit" value="Submit">  
 </form>  
</body>  
</html>

1. **Modify app.py to Use the Template**:

from flask import Flask, request, render\_template  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
@app.route('/submit', methods=['POST'])  
def submit():  
 name = request.form['name']  
 return f'Hello, {name}!'  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

Now, when you run the Flask application, it will render the form from the index.html template. This approach keeps your HTML and Python code separate, making the application easier to maintain and scale.