**Experiment – 4: Flask**

|  |  |
| --- | --- |
| **Name of Student** | **AKRUTI DABAS** |
| **Class Roll No** | **D15A / 11** |
| **D.O.P.** |  |
| **D.O.S.** |  |
| **Sign and Grade** |  |

**Aim:** To design a Flask application that showcases URL building and demonstrates the use of HTTP methods (GET and POST) for handling user input and processing data.

.

**Problem Statement:**

Create a Flask application with the following requirements:

1. A homepage (/) with links to a "Profile" page and a "Submit" page using the url\_for() function.
2. The "Profile" page (/profile/<username>) dynamically displays a user’s name passed in the URL.
3. A "Submit" page (/submit) displays a form to collect the user’s name and age. The form uses the POST method to send the data, and the server displays a confirmation message with the input.

**Theory:**

1. **What is a route in Flask, and how is it defined?**

A route in Flask is a URL pattern that is mapped to a function. It determines what happens when a user visits a specific URL in a web application. Defining a route:

In Flask, routes are created using the @app.route() decorator, which binds a URL to a function.

Example:

|  |
| --- |
| from flask import Flask    app = Flask(\_\_name\_\_)    @app.route('/') # This route maps the root URL ("/") to the home function def home(): return "Welcome to the Flask Web Application!" |

Here, visiting http://127.0.0.1:5000/ will execute the home() function and return "Welcome to the Flask Web Application!".

1. **How can you pass parameters in a URL route?**

Flask allows passing dynamic values through the URL route using angle brackets < >. These values are captured as function parameters.

Example:

|  |
| --- |
| @app.route('/profile/<username>') def profile(username): return f"Hello, {username}! Welcome to your profile." |

Output:

Hello, Alex! Welcome to your profile.

Flask automatically extracts Alex from the URL and passes it to the profile(username) function.

1. **What happens if two routes in a Flask application have the same URL pattern?**

If two routes have the same URL pattern, Flask will execute only the last defined route, overriding the previous one. This can lead to unexpected behavior.

Example:

@app.route('/about') def about\_v1(): return "About Page - Version 1"

@app.route('/about') def about\_v2(): return "About Page - Version 2"

Here, only the second function (about\_v2) will be executed, as it overrides about\_v1.

Ensure each route has a unique path, or use different HTTP methods to differentiate them.

1. **What are the commonly used HTTP methods in web applications?**

The HTTP methods define the actions that can be performed on a web server. The most commonly used ones are:

* 1. GET – Retrieves data from the server (e.g., accessing a webpage).
  2. POST – Sends data to the server (e.g., submitting a form).
  3. PUT – Updates existing data on the server.
  4. DELETE – Removes data from the server.

Example of using GET and POST in Flask:

|  |
| --- |
| @app.route('/submit', methods=['GET', 'POST']) def submit(): if request.method == 'POST': name = request.form['name'] return f"Form submitted successfully! Hello, {name}." return '<form method="post"><input name="name"><input type="submit"></form>' |

1. **What is a dynamic route in Flask?**

A dynamic route in Flask allows a URL to accept variable parts, making the route more flexible. The variable is captured and passed to the view function.

Example:

|  |
| --- |
| @app.route('/user/<username>') def greet\_user(username):  return f"Hello, {username}!" |

• If a user visits http://127.0.0.1:5000/user/Sam, Flask captures "Sam" as the username parameter.

1. **Write an example of a dynamic route that accepts a username as a parameter.**

Code:

@app.route('/profile/<username>') def profile(username): return f"Welcome to {username}'s profile!"

Visiting http://127.0.0.1:5000/profile/Jane will return output as:

Welcome to Jane's profile!

1. **What is the purpose of enabling debug mode in Flask?**

Enabling debug mode in Flask provides the following benefits:

* 1. Automatic Reloading – The server restarts whenever you change the code.
  2. Detailed Error Messages – Flask shows interactive error messages, making debugging easier.

1. **How do you enable debug mode in a Flask application?**

To enable debug mode, set debug=True when running the Flask app.

Method 1: In Code if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Method 2: Using Environment Variables export FLASK\_ENV=development

flask run

This allows Flask to reload automatically and display detailed error messages.

**Output:**

• **App.py**

from flask import Flask, render\_template, request, redirect, url\_for

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

@app.route('/') def home(): name = request.args.get('name', 'Guest') # Get name from query parameter return render\_template('home.html', name=name)

@app.route('/profile/<username>') def profile(username): return render\_template('profile.html', username=username)

@app.route('/submit', methods=['GET', 'POST']) def submit(): if request.method == 'POST': name = request.form['name'] age = request.form['age']

return render\_template('confirmation.html', name=name, age=age) return render\_template('submit.html')

@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')

@app.route('/thank\_you') def thank\_you(): return "Thank you for submitting the form!"

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

• **home.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Flask App | Home</title>

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

<style> body {

background: linear-gradient(to right,rgb(0, 0, 0),rgb(6, 88, 87)); color: white;

}

.container { max-width: 700px; margin-top: 50px;

text-align: center;

}

.btn {

margin: 10px;

}

.navbar {

background: rgba(0, 0, 0, 0.6);

}

.a{

margin-top: 500px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark">

<div class="container text-center">

<a class="navbar-brand mx-auto " href="#" style="font-size: 2rem; fontweight: bold;"></a>

</div>

</nav>

<div class="container">

<h1 class="display-4">Welcome, {{ name }}!</h1>

<p class="lead">Explore the website:</p>

<a href="{{ url\_for('profile', username='Guest') }}" class="btn btn-light btnlg">Profile</a>

<a href="{{ url\_for('submit') }}" class="btn btn-light btn-lg">Submit Details</a> <a href="{{ url\_for('contact') }}" class="btn btn-light btn-lg">Contact Us</a> </div>

</body>

</html>

• **profile.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Profile | Flask App</title>

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

<style> body {

background: linear-gradient(to right,rgb(0, 0, 0),rgb(6, 88, 87)); color: #333;

}

.container { max-width: 500px; margin-top: 100px; text-align: center; background: white; padding: 30px; border-radius: 10px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

</style>

</head>

<body>

<div class="container">

<h2>Hello, {{ username }}!</h2>

<p>This is your profile page.</p>

<a href="{{ url\_for('home') }}" class="btn btn-dark">Back to Home</a> </div>

</body>

</html>

• **submit.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Submit Details | Flask App</title>

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

<style> body {

background: linear-gradient(to right,rgb(0, 0, 0),rgb(6, 88, 87));

}

|  |  |
| --- | --- |
|  | .form-container { max-width: 400px; margin: 100px auto; padding: 30px; background: white; border-radius: 10px;  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  }  </style>  </head>  <body>  <div class="form-container">  <h3 class="text-center">Submit Your Details</h3>  <form method="POST">  <div class="mb-3">  <label class="form-label">Name:</label>  <input type="text" name="name" class="form-control" required>  </div>  <div class="mb-3">  <label class="form-label">Age:</label>  <input type="number" name="age" class="form-control" required>  </div>  <button type="submit" class="btn btn-dark w-100">Submit</button> </form>  </div>  </body>  </html> |
| • | **confirmation.html**    <!DOCTYPE html>  <html lang="en">  <head>  <title>Confirmation | Flask App</title>  <link rel="stylesheet"  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  <style> body {  background: linear-gradient(to right,rgb(0, 0, 0),rgb(6, 88, 87));  }  .container { max-width: 500px; margin-top: 100px; text-align: center; background: white; padding: 20px; |

