Experiment-10

Date: 1/8/25

Design a form to get some data at the client side from the user and try to access this data on the server by using the POST request

FLASK HTTP METHODS, HANDLE GET & POST REQUESTS

Flask has different decorators to handle http requests. *Http protocol* is the basis for data communication in the *World Wide Web*.

Different methods for retrieving data from a specified URL are defined in this protocol. The following table summarizes the different http methods:

Request	Purpose
GET	The most common method. A GET message is send, and the server returns data
POST	Used to send HTML form data to the server. The data received by the POST method is not cached by the server.
HEAD	Same as GET method, but no response body.
PUT	Replace all current representations of the target resource with uploaded content.
DELETE	Deletes all current representations of the target resource given by the URL.

Flask HTTP Methods

Form

By default, the Flask route responds to GET requests. However, you can change this preference by providing method parameters for the route () decorator.

To demonstrate the use of a POST method in a URL route, first let us create an HTML form and use the POST method to send form data to the URL.

Save the following script as **login.html**

```
<html>
<body>
<form action="/login" method="Post">
    <P> enter name : </P>
    <P><input type="text" name="nm" required
    ></P>
    <P><input type="Submit" name="Submit"></P>
</form>
</body>
</html>
```

app.py

```
from flask.inPost flask, redirect, url_for, request
                                        render_template

app=flask(__name__)
@app.route('/')
def home():
    return render_template('login.html')
@app.route('/success/<name>')
def success(name):
    return 'welcome %s' %name
@app.route('/login', method=['POST','GET'])
def login():
    if request.method=='POST':
        user=request.form['nm']
        return redirect(url_for('success', name=user))
    else:
        user=request.args.get('nm')
        return redirect(url_for('success', name=user))
```

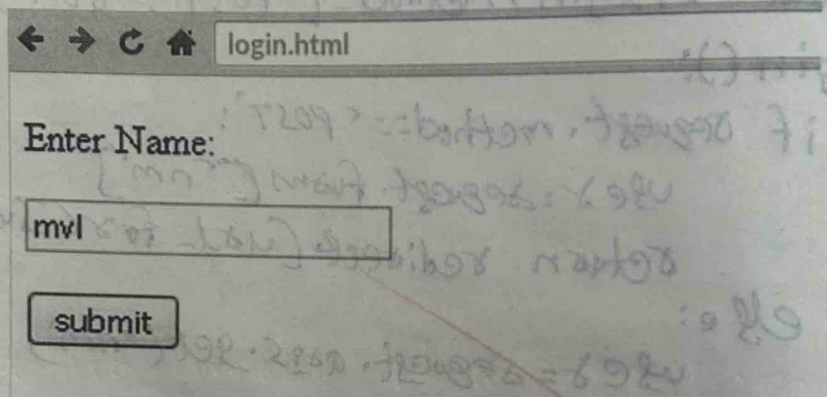
GET and POST requests

To handle both GET and POST requests, we add that in the decorator app. route () method. Whatever request you want, you change it in the decorator.

Enter the following script in the Python shell.

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

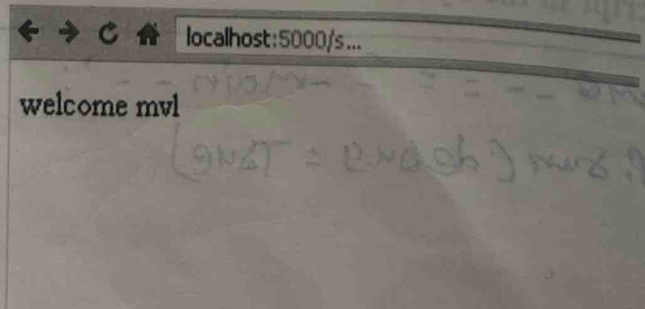
Once the development server is up and running, open login.html in the browser, enter the name in the text field, and then click Submit.



The form data will POST to the URL in the action clause of the form label.

localhost/login image to the login() function. Because the server receives data through the POST method, the value of the "nm" parameter obtained from the form data is obtained by following these steps.

It is passed as part of the variable to the '/success' URL. The browser displays a welcome message in the window.



1. What is Flask?

Ans. Flask is a web frame work. This means flask provides you with tools, libraries and technology that allow to build a webapplic

2. Why do we use Flask(__name__) in Flask?

Ans. According to the official flask documentation a=name- argument is passed in flask class to create its instance which is than used to run the application.

3. What is routing in Flask?

Ans. APP routing is the technique used to map the specific url with the associated function intended to perform some flask.

4. Who created Flask?

Ans. flask is created by Armin Ronachat of Pocoo.

5. Why do we use Flask?

Ans. flask is used for developing web Applications using Python.