

ASSIGNMENT - 2

Assignment Date	25 September 2022
Student Name	Rajarajeswari.J
Student Roll Number	820419205044
Maximum Marks	2 Marks

1) Write a flask program which should display Name, Email, phone and it should display the same details once we hit submit.

index.html

```
<!DOCTYPE html>

<html>

<head>

<title> Python</title>

</head>

<body>

<form action ="/login" method="post">

<p>Enter name:</p>

<p><input type="text" name="name"/></p>

<p>Enter phone number:</p>

<p><input type="number" name="phone"/></p>

<p>Enter email id:</p>

<p><input type="email" name="email"/></p>

<p><input type="submit" value="submit"/></p>

</form>

<p><b>{{ x }} </b></p>

<p><b>{{ y }} </b></p>

<p><b>{{ z }} </b></p>

</body>

</html>
```

app.py

```
from flask import Flask,render_template,request

app=Flask(__name__)

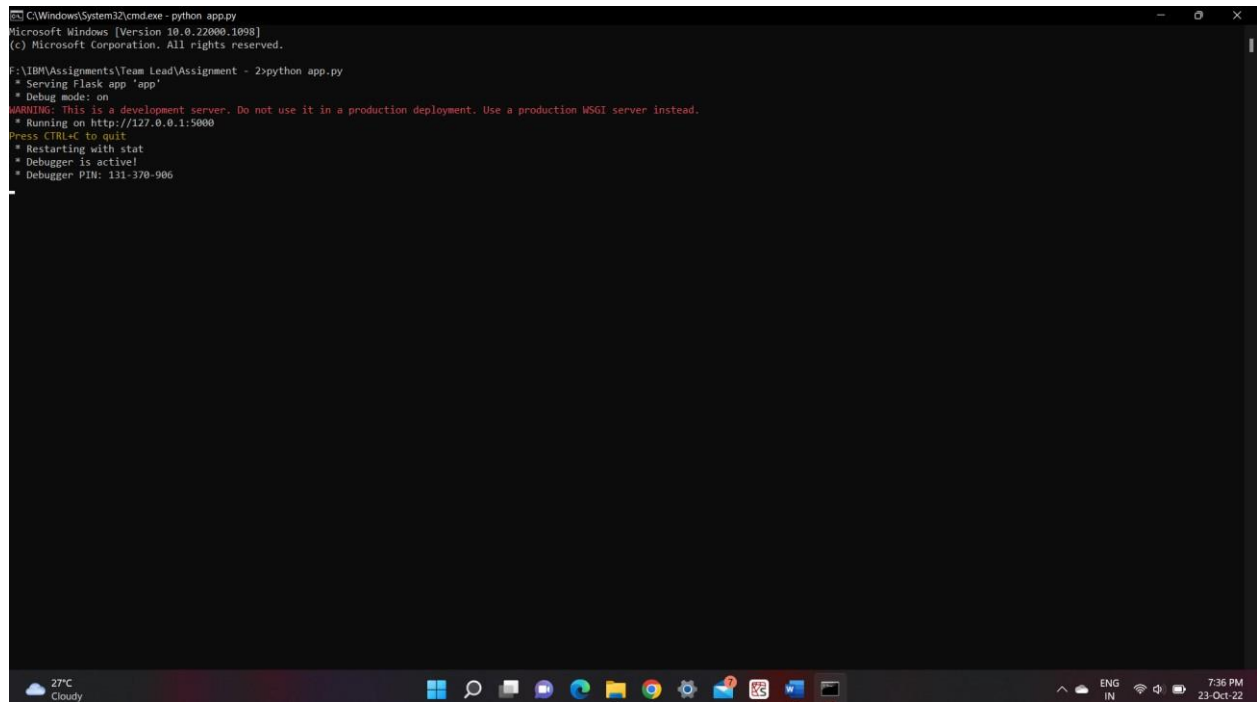
@app.route('/') def
home():

    return render_template('index.html')

@app.route('/login', methods = ["POST"])
def login():    if request.method ==
"POST":
name=request.form["name"]
phone=request.form["phone"]
email=request.form["email"]

    return render_template("index.html",x=name,y=phone,z=email) if
__name__=='__main__':
    app.run(debug=True)
```

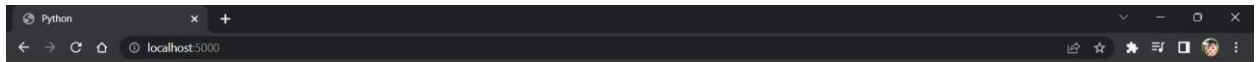
Output Screenshots:



```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

F:\IDM\Assignments\Team Lead\Assignment - 2>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 131-370-906
```

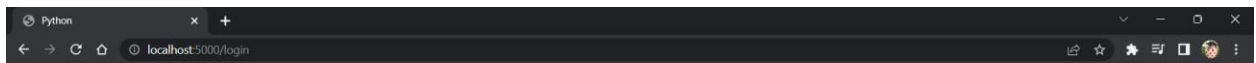
The screenshot shows a Windows command prompt window with a dark background. The title bar indicates the command being executed is 'C:\Windows\System32\cmd.exe - python app.py'. The window content shows the output of running 'python app.py' from the directory 'F:\IDM\Assignments\Team Lead\Assignment - 2'. The output includes standard Flask development server messages: 'Serving Flask app 'app'', 'Debug mode: on', a warning about using a development server, the address 'http://127.0.0.1:5000', and instructions to press CTRL+C to quit. It also shows the server restarting with stat, the debugger is active, and the debugger PIN is 131-370-906. The Windows taskbar at the bottom shows the system clock as 7:36 PM on 23-Oct-22, with a temperature of 27°C and a cloudy weather icon.



Enter name:

Enter phone number:

Enter email id:



Enter name:

Enter phone number:

Enter email id:

Nithisha
1234567890
nithi@nithi.com



2) Write a flask program which should cover cookies, sessions getsession.html

```
<html>
<head>
<title>getting the session</title>
</head>
<body>
<p>The session is set with value: <strong>{ { name } }</strong></p>
</body>
</html>
```

session.py

```
from flask import Flask, make_response ,session,
render_template app = Flask(__name__)  app.secret_key = "abc"

@app.route('/')  def
home():
    res = make_response("<h4>session variable is set, <a href='/get'>Get Variable</a></h4>")
    session['response']='session#1'    return res;

@app.route('/get')  def
getVariable():    if
'response' in session:
s = session['response'];
    return render_template('getsession.html',name = s)
```

```
@app.route('/cookies') def
cookie():

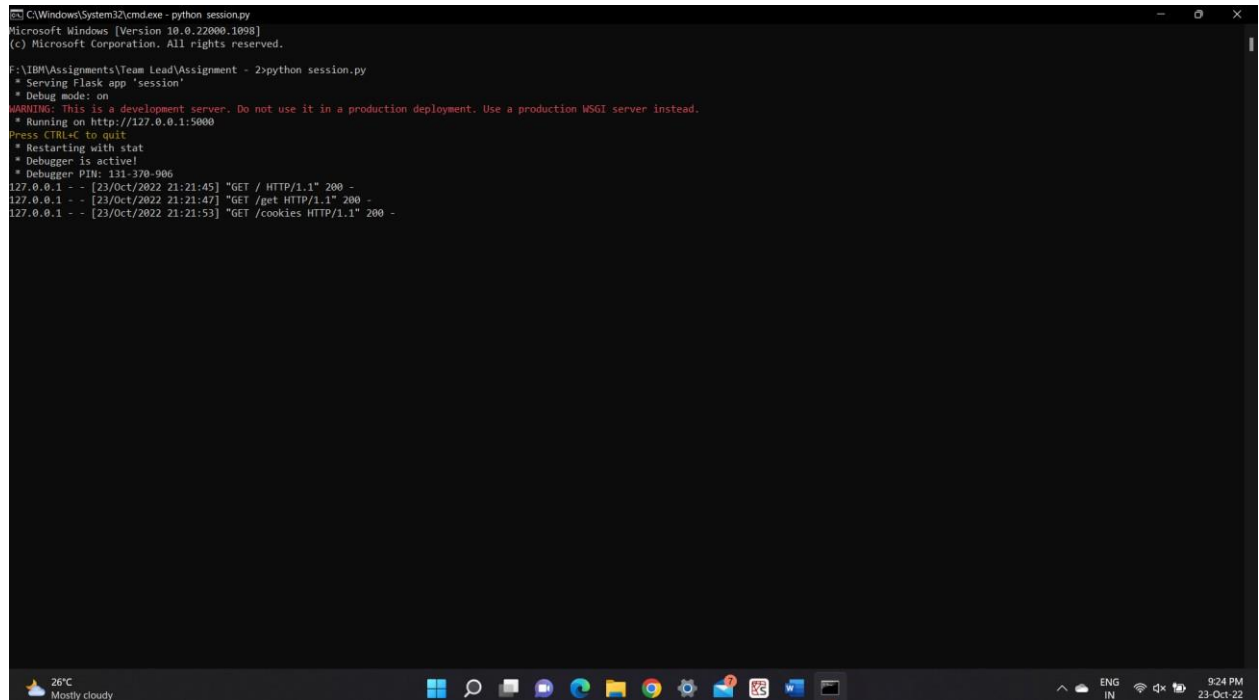
    res = make_response("<h1>cookie is set</h1>")

res.set_cookie('foo','bar')    return res


if __name__ == '__main__':

    app.run(debug = True)
```

Output Screenshots:

A screenshot of a Windows command prompt window titled "C:\Windows\System32\cmd.exe - python session.py". The window shows the output of running a Flask application. The text in the terminal is as follows:

```
Microsoft Windows [Version 10.0.22000.1998]
(c) Microsoft Corporation. All rights reserved.

F:\IBW\Assignments\Team Lead\Assignment - 2>python session.py
 * Serving Flask app 'session'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 131-370-906
127.0.0.1 - - [23/Oct/2022 21:21:45] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Oct/2022 21:21:47] "GET /get HTTP/1.1" 200 -
127.0.0.1 - - [23/Oct/2022 21:21:53] "GET /cookies HTTP/1.1" 200 -
```

The taskbar at the bottom shows the system clock as 9:24 PM on 23-Oct-22, with a temperature of 26°C and weather "Mostly cloudy".





3) Write a flask program which should display resume details and also have upload resume option by using file uploading.

upload.py

```
from flask import Flask, render_template, request
app = Flask(__name__)

@app.route('/')
def upload():
    return render_template("upload.html")

@app.route('/success', methods = ['POST'])
def success():
    if request.method == 'POST':
        f = request.files['file']
        f.save(f.filename)

        return render_template("success.html", name = f.filename)
```



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

upload.html

```
<html>  
<head>  
    <title>upload</title>  
</head>  
<body>  
    <form action = "/success" method = "post" enctype="multipart/form-data">  
        <h1> PERSONAL DETAILS </h1>  
        <p>Enter name:</p>  
        <p><input type="text" name="name"/></p>  
        <p>Enter phone number:</p>  
        <p><input type="number" name="phone"/></p>  
        <p>Enter email id:</p>  
        <p><input type="email" name="email"/></p>  
        <p>Enter CGPA:</p>  
        <p><input type="number" name="cgpa"/></p>  
        <p>Enter year of Graduation:</p>  
        <p><input type="month" name="grad"/></p>  
        <p>Enter address:</p>  
        <p><textarea name="file" rows="4" cols="50"></textarea></p>  
        <p> Upload your resume:</p>  
        <input type="file" name="file" />  
        <p><input type = "submit" value="Upload"></p>  
    </form>
```

</body>

</html>

success.html

<html>

<head>

<title>success</title>

</head>

<body>

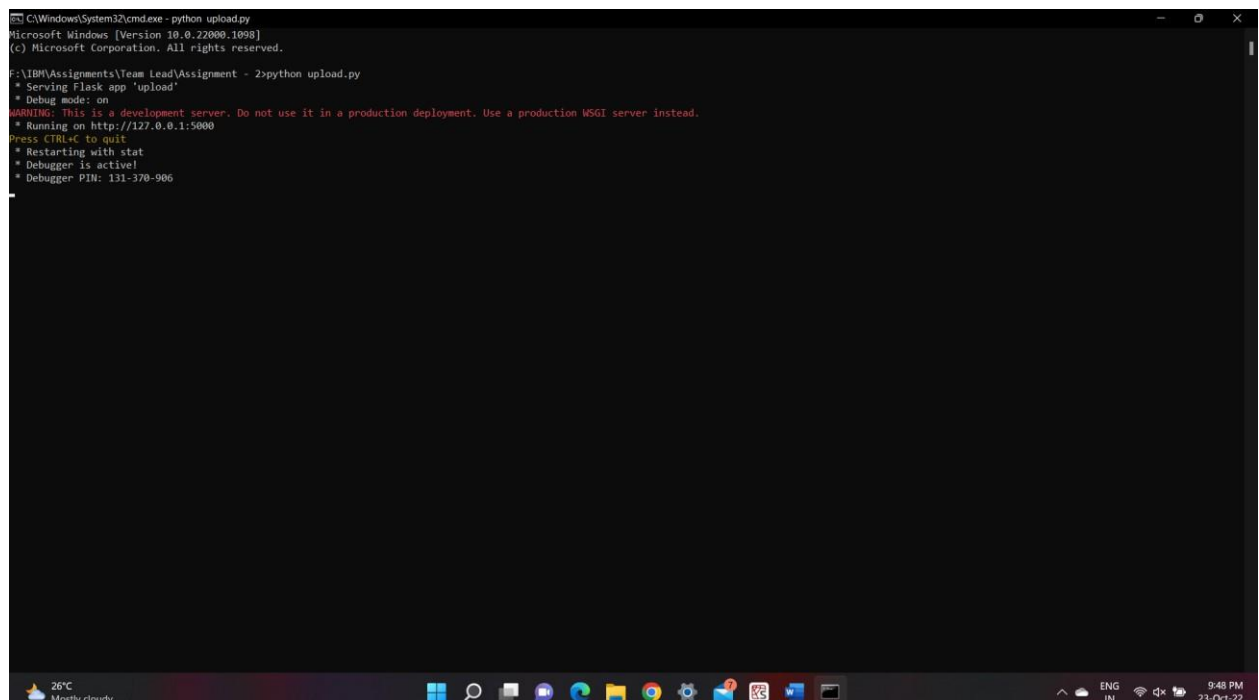
<p>File uploaded successfully</p>

<p>File Name: {{ name }}</p>

</body>

</html>

Output Screenshots:



```
C:\Windows\System32\cmd.exe - python upload.py
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

P:\IBH\Assignments\Team Lead\Assignment - 2>python upload.py
* Serving Flask app 'upload'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 131-370-906
```

upload

localhost:5000

PERSONAL DETAILS

Enter name:

Enter phone number:

Enter email id:

Enter CGPA:

Enter year of Graduation:

Enter address:

Upload your resume:

Choose File No file chosen

Upload



upload

localhost:5000

PERSONAL DETAILS

Enter name:

Enter phone number:

Enter email id:

Enter CGPA:

Enter year of Graduation:

Enter address:

Upload your resume:

Choose File Resume.pdf

Upload



