

ASSIGNMENT – 2

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"]

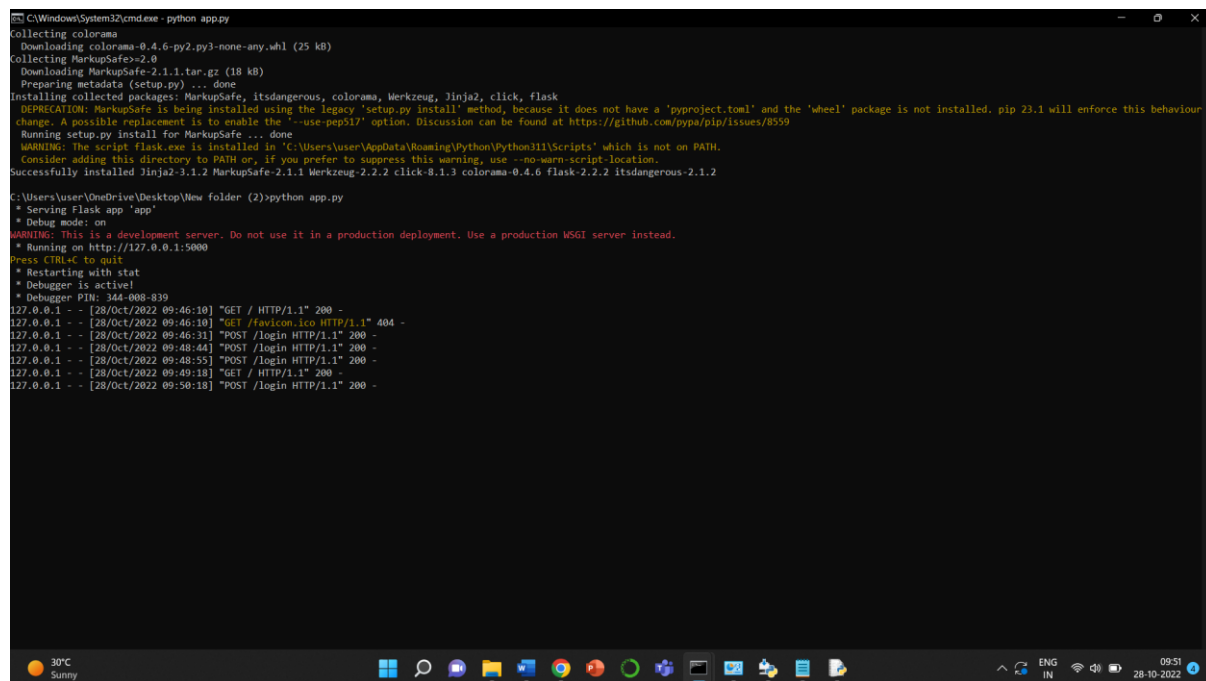
        return render_template("index.html",x=name,y=phone,z=email)

if __name__=='_main__':

    app.run(debug=True)

```

Output Screenshot:

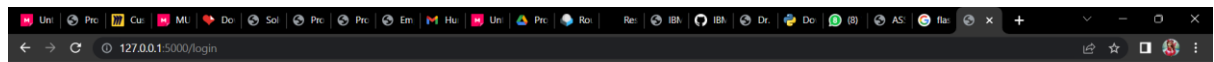


```

C:\Windows\System32\cmd.exe - python app.py
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1.tar.gz (18 kB)
  Preparing metadata (setup.py) ... done
Installing collected packages: MarkupSafe, itsdangerous, colorama, Werkzeug, Jinja2, click, flask
  WARNING: MarkupSafe is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for MarkupSafe ... done
  WARNING: The script flask.exe is installed in 'C:\Users\User\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 colorama-0.4.6 flask-2.2.2 itsdangerous-2.1.2

C:\Users\User\OneDrive\Desktop\New folder (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: 344-008-839
127.0.0.1 - - [28/Oct/2022 09:46:10] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 09:46:10] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [28/Oct/2022 09:46:11] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 09:48:44] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 09:48:55] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 09:49:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 09:50:18] "POST /login HTTP/1.1" 200 -

```



Enter name:

Enter phone number:

Enter email id:

melba

9452634178

melba1234@gmail.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>")
```

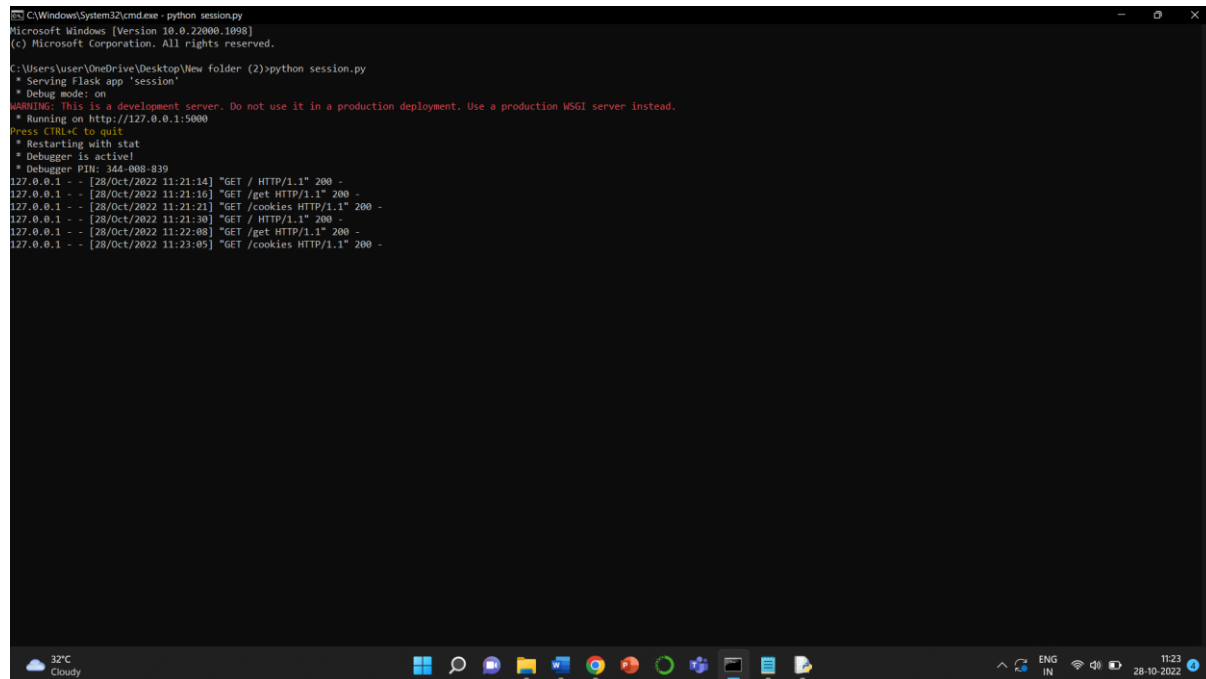
```
    res.set_cookie('foo','bar')
```

```
    return res
```

```
if __name__ == '__main__':
```

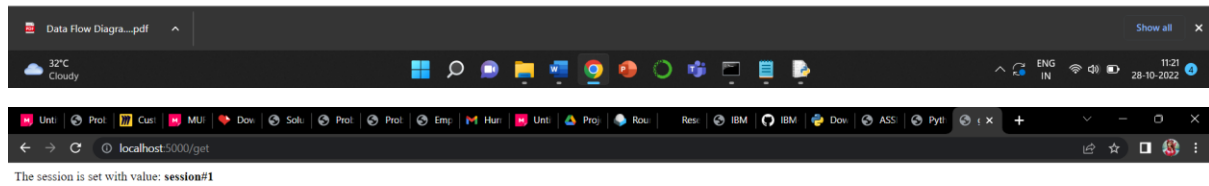
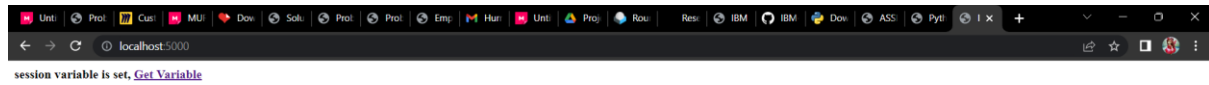
```
    app.run(debug = True)
```

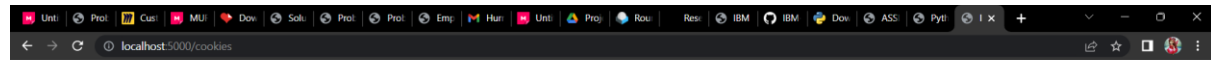
Output Screenshot:



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

C:\Users\user\OneDrive\Desktop\New folder (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: 344-008-839
127.0.0.1 - - [28/Oct/2022 11:21:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:21:16] "GET /get HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:21:21] "GET /cookies HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:21:30] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:22:08] "GET /get HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:23:05] "GET /cookies HTTP/1.1" 200 -
```





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>

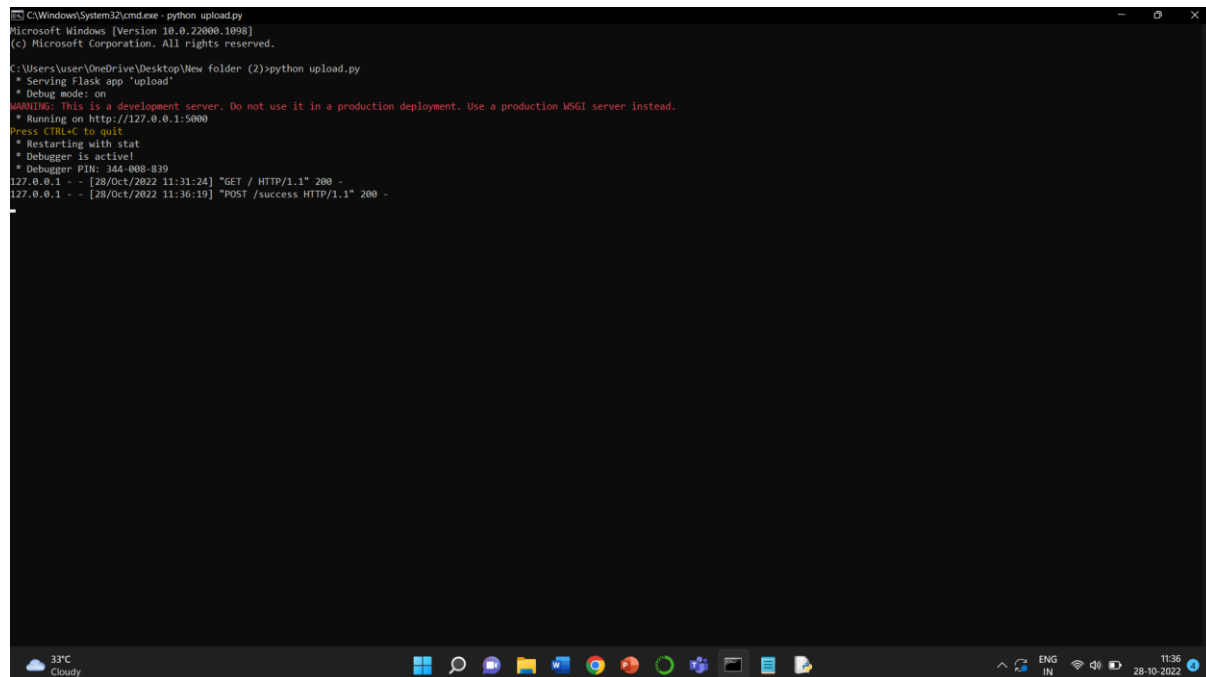
<p>File uploaded successfully</p>

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

</body>

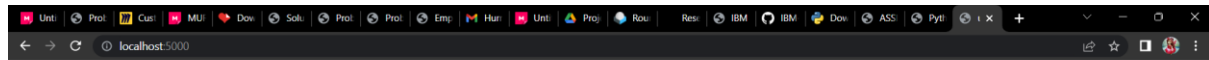
</html>

Output Screenshot:



```
C:\Windows\System32\cmd.exe - python upload.py
Microsoft Windows [Version 10.0.22000.1998]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\OneDrive\Desktop\New folder (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: 344-088-839
127.0.0.1 - - [28/Oct/2022 11:31:24] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Oct/2022 11:36:19] "POST /success HTTP/1.1" 200 -
```



PERSONAL DETAILS

Enter name:

Enter phone number:

Enter email id:

Enter CGPA:

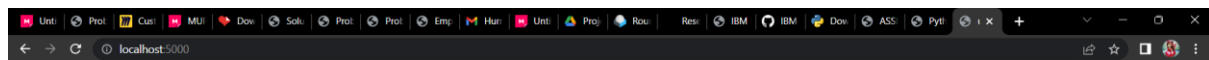
Enter year of Graduation:

Enter address:

Upload your resume:

Choose File No file chosen

Upload



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



