**Assignment -1**

**Flask Programming**

| Assignment Date | 15 october 2022 |
| --- | --- |
| Student Name | V S SHOBIKA |
| Student Roll Number | 712219106012 |
| Maximum Marks | 2 Marks |

**Question-1:**

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

**Solution:**

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

import os

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

team\_members = {“1” : “Kannaki karthika”,”2” : “srimathi”, “3” : “shobika”, “4” : “esther renisha”} @app.route(‘/data’, methods = [‘POST’,’GET’])

def api():

if request.method == ‘GET’:

return team\_members

if request.method == ‘POST’:

data = request.json

team\_members.update(data)

return “Data is inserted”

@app.route(“/data/<id>”, methods=[“PUT”])

def update(id):

data = request.form[‘member’]

team\_members[str(id)]=data

return “Data is updated”

@app.route(“/data/<id>”, methods=[“DELETE”])

def delete(id):

team\_members.pop(str(id))

return “Data Deleted”

if \_\_name\_\_ == ‘\_\_main\_\_’:

port = os.environ.get(‘FLASK\_PORT’) or 8080

port = int(port)

app.run(port=port,host=’0.0.0.0’)

**Output** {“1”:”kannaki karthika”,”2”:”srimathi”,”3”:shobika”,”4”:esther renisha”}

**Email**

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

import osapp = Flask(\_\_name\_\_)

team\_members = {“1” : kkdharshu91@gmail.com,”2” : srimathik022gmail., “3” : shobikasubashini69@gmail.com, “4” : durgamr2002@gmail.com}

@app.route(‘/data1’, methods = [‘POST’,’GET’])

def api():

if request.method == ‘GET’:

return team\_members

if request.method == ‘POST’:

data = request.json

team\_members.update(data)

return “Data is inserted”

@app.route(“/data1/<id>”, methods=[“PUT”])

def update(id):

Data = request.form[‘member’]

team\_members[str(id)]=data

return “Data is updated”

@app.route(“/data1/<id>”, methods=[“DELETE”])

def delete(id):

team\_members.pop(str(id))

return “Data Deleted”

if \_\_name\_\_ == ‘\_\_main\_\_’:

port = os.environ.get(‘FLASK\_PORT’) or 8080

port = int(port)

app.run(port=port,host=’0.0.0.0’)

**Output** {“1”:”kkdharshu91@gmail.com”,”2”:”srimathik02@gmail.com”,”3”:shobikasubasini69@gmail.com”,”4”:”durgamr2002@gmail.com”}

**Phon{{e**

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

import os

app = Flask(\_\_name\_\_) {“1” : “7695912147”,”2” : “6382510942”, “3” : “8220895328”, “4” : “8270412679”}

@app.route(‘/data2’, m

team\_members =ethods = [‘POST’,’GET’])

def api():

if request.method == ‘GET’:

return team\_members

if request.method == ‘POST’:

data = request.json

team\_members.update(data)

return “Data is inserted”

@app.route(“/data2/<id>”, methods=[“PUT”])

def update(id):

data = request.form[‘member’]

team\_members[str(id)]=data

return “Data is updated”

@app.route(“/data2/<id>”, methods=[“DELETE”]) def delete(id):

team\_members.pop(str(id))

return “Data Deleted”

if \_\_name\_\_ == ‘\_\_main\_\_’:

port = os.environ.get(‘FLASK\_PORT’) or 8080

port = int(port)

app.run(port=port,host=’0.0.0.0’)

sk import Flask, redirect,url\_for, request,render\_template,json

**Input** {“1”:”7695912147”,”2”:”712219106013”,”3”:”8220895328”,”4”:”8270412679”}

**Question-2:**

**Write a flask program which should cover cookies and session.**

**Solution:**

**Create cookie**

@app.route(‘/’)

Def index():

Return render\_template(‘index.html’)

**This HTML page contains a text input.**

<html>

<body>

<form action = “/setcookie” method = “POST”>

<p><h3>Enter userID</h3></p>

<p><input type = ‘text’ name = ‘nm’/></p>

<p><input type = ‘submit’ value = ‘Login’/></p>

</form>

</body>

</html>

**Set cookie**

@app.route(‘/setcookie’, methods = [‘POST’, ‘GET’]) Def setcookie():

If request.method == ‘POST’:

User = request.form[‘nm’]

Resp = make\_response(render\_template(‘readcookie.html’)) Resp.set\_cookie(‘userID’, user)

Return resp

**Get cookie**

@app.route(‘/getcookie’)

Def getcookie():

Name = request.cookies.get(‘userID’)

Return ‘<h1>welcome ‘ + name + ‘</h1>'

**Output**

**SESSION**

from flask import Flask, render\_template\_string, request, session, redirect, url\_for @app.route(‘/set\_email’, methods=[‘GET’, ‘POST’])

Def set\_email():

If request # Create the Flask application

App = Flask(\_\_name\_\_)

App.secret\_key = ‘BAD\_SECRET\_KEY’

.method == ‘POST’:

# Save the form data to the session object

Session[‘email’] = request.form[‘email\_address’]

Return redirect(url\_for(‘get\_email’))

Return “””

<form method=”post”>

<label for=”email”>Enter your email address:</label>

<input type=”email” id=”email” name=”email\_address” required />

<button type=”submit”>Submit</button

</form>

“””

@app.route(‘/get\_email’)

Def get\_email():

Return render\_template\_string(“””

{% if session[‘email’] %}

<h1>Welcome {{ session[‘email’] }}!</h1>

{% else %}

<h1>Welcome! Please enter your email <a href=”{{ url\_for(‘set\_email’) }}”>here.</a></h1> {% endif %}

“””)

@app.route(‘/delete\_email’)

Def delete\_email():

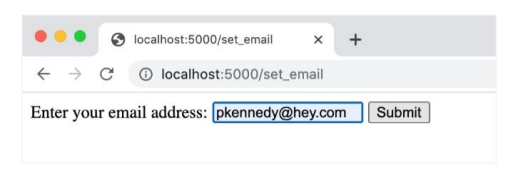
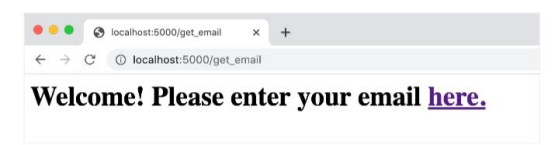
# Clear the email stored in the session objects

Session.pop(‘email’, default=None)

Return ‘<h1>Session deleted!</h1>’

If \_\_name\_\_ == ‘\_\_main\_\_’:

App.run()



**Question-3:**

Write a Flask program which should display resume details and also have upload resume option by using file uploading

**Solution**

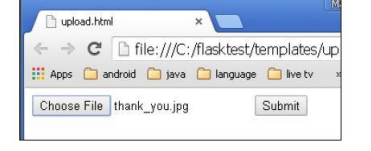
**upload.html**

| <html>  <body>  <form action = "http://localhost:5000/uploader" method = "POST"  enctype = "multipart/form-data">  <input type = "file" name = "file" />  <input type = "submit"/>  </form>  </body>  </html> |
| --- |

**upload.py**

| from flask import Flask, render\_template, request  from werkzeug import secure\_filename  app = Flask(\_\_name\_\_)  @app.route('/upload')  def upload\_file():  return render\_template('upload.html')  @app.route('/uploader', methods = ['GET', 'POST'])  def upload\_file():  if request.method == 'POST':  f = request.files['file']  f.save(secure\_filename(f.filename))  return 'file uploaded successfully'  if \_\_name\_\_ == '\_\_main\_\_':  app.run(debug = True) |
| --- |

**output**

****