

Assignment -1
Flask Programming

Assignment Date	14 September 2022
Student Name	Mr. R.Dinesh Kumar
Student Roll Number	814619104007
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": "DINESH KUMAR R", "2": "dinesh7502ravi@gmail.com", "3":  
"+911222457007"}
```

```
@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"

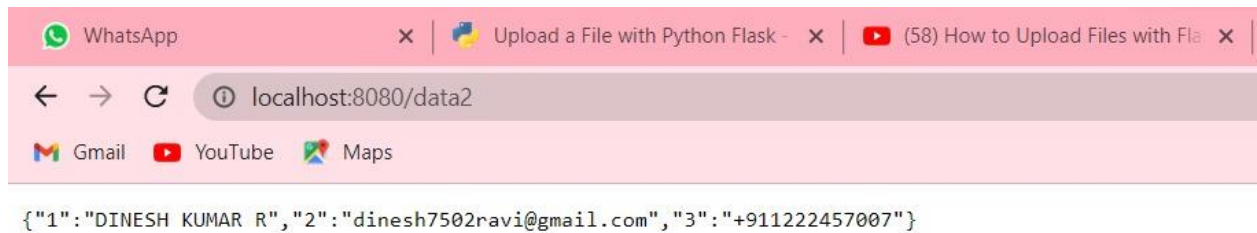
If __name__ == '__main__':

Port = os.environ.get('FLASK_PORT') or 8080

Port = int(port)

App.run(port=port,host='0.0.0.0')

```



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>

```

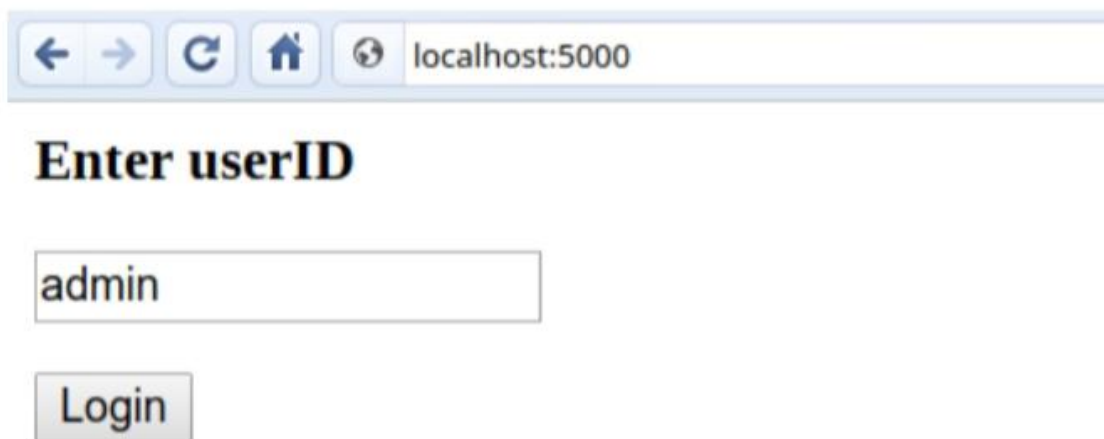
Set cookie

```
@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



← → ↻ 🏠 🌐 localhost:5000

Enter userID

SESSION

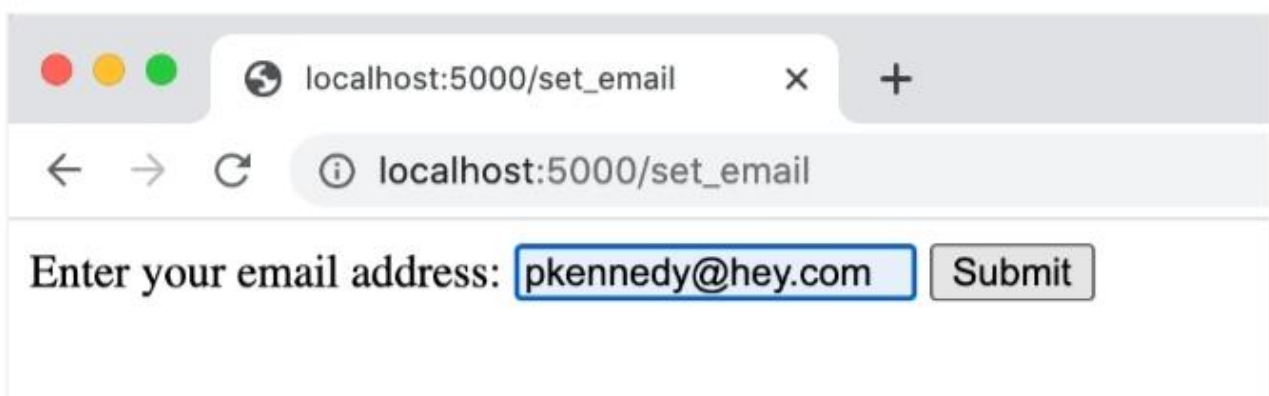
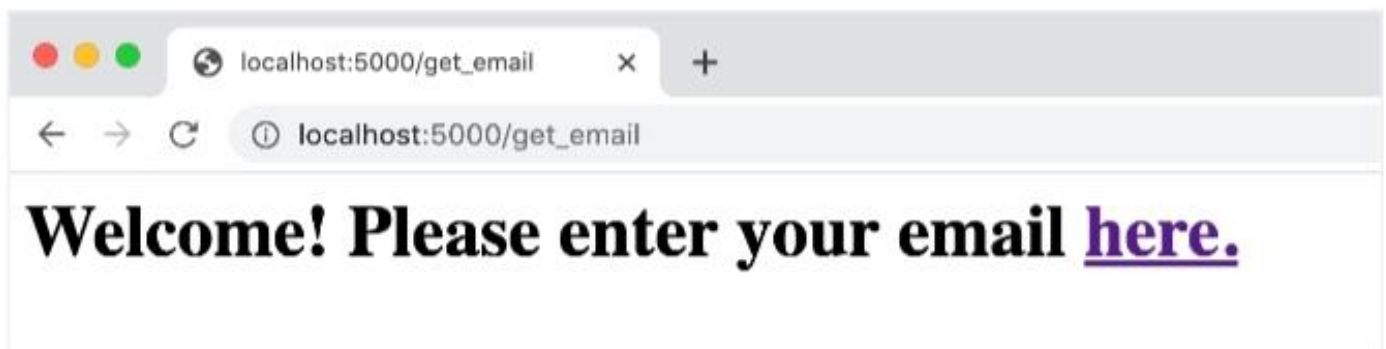
```
from flask import Flask, render_template_string, request, session, redirect, url_for
@app.route('/set_email', methods=['GET', 'POST'])
Def set_email():
    If request # Create the Flask application
    App = Flask(__name__)
    App.secret_key = 'BAD_SECRET_KEY'
    .method == 'POST':
        # Save 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

