

Assignment-8

Storing and viewing through postman

Viewnotes.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1"/>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scrip
t>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></sc
ript>
    <title>view Notes Page</title>
  </head>
  <body style="background-color: aliceblue;">
    <nav style="color: blue;background-color: aqua;" class="navbar navbar-
default">
      <div class="container-fluid">
        <div class="navbar-header">
          <a class="navbar-brand" href="">Full Stack Demo</a>
        </div>
        <ul class="nav navbar-nav">
          <li><a href="/dashboard">Home</a></li>
          <li><a href="/insertnotes">Insert notes</a></li>
          <li class="active"><a href="/viewnotes">view
notes</a></li>
          <li><a href="/logout">Logout</a></li>
        </ul>
      </div>
    </nav>
    <div class="container">
      <table class="table table-bordered">
        <tr>
          <th>S.no</th>
          <th>Notes</th>
        </tr>
        {%for i in range(1)%}
          <tr>
            <td> {{i+1}}</td>
            <td> {{data[i]}}</td>
          </tr>
        {%endfor%}
      </table>
```

```

        </div>
    </body>
</html>

```

Insertnotes.html

```

<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width,initial-scale=1"/>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scrip
t>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></sc
ript>
        <title>Dashboard Page</title>
    </head>
    <body style="background-color: aliceblue;">
        <nav style="color: blue;background-color: aqua;" class="navbar navbar-
default">
            <div class="container-fluid">
                <div class="navbar-header">
                    <a class="navbar-brand" href="">Full Stack Demo</a>
                </div>
                <ul class="nav navbar-nav">
                    <li><a href="/dashboard">Home</a></li>
                    <li class="active"><a href="/insertnotes">Insert
notes</a></li>
                    <li><a href="/viewnotes">view notes</a></li>
                    <li><a href="/logout">Logout</a></li>
                </ul>
            </div>
        </nav>
        <div class="col-sm-2"></div>

        <div class="container col-sm-6">
            <form action="/insertnotesdata" method="post">
                <div class="form-group">
                    <label>Enter the notes</label><br/>
                    <textarea rows="10" cols="100" name="notes"></textarea>
                </div>
                <button type="submit" class="btn btn-warning">Insert
notes</button>
            </form>
            <span style="color: blue">{{res2}}</span>

```

```
        </div>
    </body>
</html>
```

login.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta name="viewport" content="width=device-width,initial-
scale=1"/>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.
min.css">
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.j
s"></script>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.mi
n.js"></script>
        <title>Login Page</title>
    </head>
    <body style="background-color: aliceblue;">
        <nav style="color: blue;background-color: aqua;"
class="navbar navbar-default">
            <div class="container-fluid">
                <div class="navbar-header">
                    <a class="navbar-brand" href="">Full Stack
Demo</a>
                </div>
                <ul class="nav navbar-nav">
                    <li><a href="">Home</a></li>
                    <li><a href="/">Register</a></li>
                    <li class="active"><a
href="/login">Login</a></li>
                </ul>
            </div>
        </nav>
        <center>
            <div class="col-sm-4"></div>
            <div class="container col-md-4">
                <h1>Login Page</h1> <br/>
                <form action="/logindata" method="post">
                    <div class="form-group">
                        <label for="mobile">Mobile No:</label>
                        <input type="text" class="form-control"
name="mobile" placeholder="Enter Mobile No"/>
```

```

        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" class="form-control"
name="password" placeholder="Enter password"/>
        </div>
        <div class="form-group">
            <button class="btn btn-primary"
type="submit">Login</button>
        </div>
    </form>
    <span style="color:red">{{err1}}</span>
    <span style="color:green">{{res1}}</span>
</div>

<div class="col-sm-4"></div>
</center>
<a href="https://www.Ammu.com">Ammu</a>

</body>
</html>

```

app.py

```

from flask import Flask,render_template,request,redirect,session

from pymongo import MongoClient
client=MongoClient('127.0.0.1',27017)
db=client['Ammu']
c=db['flaskapp']
app=Flask(__name__)
app.secret_key='Ammu'
@app.route('/')
def homepage():
    return render_template('index.html')

@app.route('/login')
def loginpage():
    return render_template('login.html')
@app.route('/dashboard')
def dashboardpage():
    owner=session['username']
    return render_template('dashboard.html',owner=owner)
@app.route('/insertnotes')
def insertnotespage():
    return render_template('insertnotes.html')
@app.route('/viewnotes')
def viewnotes():
    c1=db['notes']
    data=[]

```

```

        for i in c1.find():
            if i['owner']==session['username']:
                data.append(i['notes'])
        return render_template('viewnotes.html',data=data,l=len(data))
@app.route('/formdata',methods=['post'])
def formdata():
    name=request.form['name']
    email=request.form['email']
    mobile=request.form['mobile']
    password=request.form['password']
    print(name,email,mobile,password)
    for i in c.find():
        if i['mobile']==mobile:
            return render_template('index.html',err='you have already
registered')
    k={}
    k['name']=name
    k['email']=email
    k['mobile']=mobile
    k['password']=password
    c.insert_one(k)
    return render_template('index.html',res='you have registered
successfully')
@app.route('/logindata',methods=['post'])
def logindata():
    mobile=request.form['mobile']
    password=request.form['password']
    print(mobile,password)
    for i in c.find():
        if i['mobile']==mobile and i['password']==password:
            session['username']=mobile
            return redirect('/dashboard')
            #return render_template('login.html',res1='valid
credentials')

    return render_template('login.html',err1='invalid credentials')

@app.route('/logout')
def logout():
    session['username']=None
    return redirect('/')

@app.route('/insertnotesdata',methods=['post'])
def insertnotesdata():
    notes=request.form['notes']
    owner=session['username']
    print(notes,owner)
    c1=db['notes']

```

```

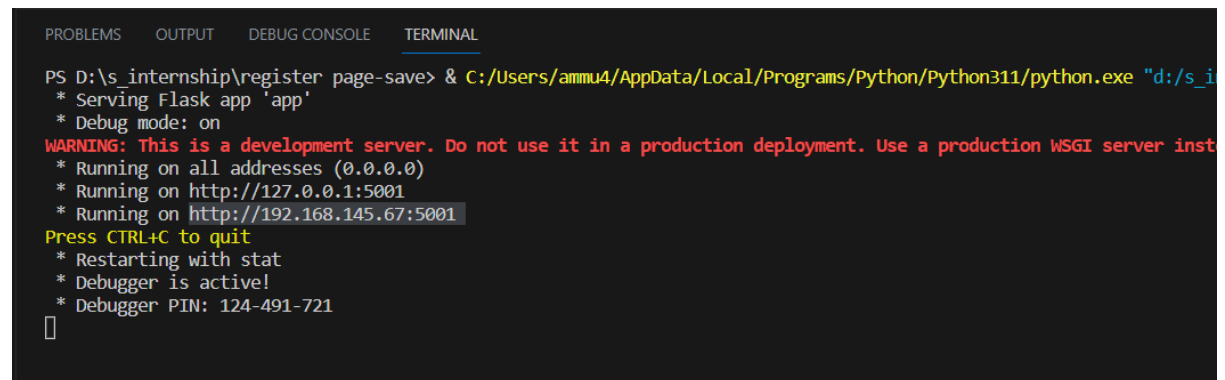
k={}
k['owner']=owner
k['notes']=notes
c1.insert_one(k)

return render_template('insertnotes.html',res2='Notes saved')

if __name__=="__main__":
    app.run(host='0.0.0.0',port=5001,debug=True)

```

Output:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\s_internship\register page-save> & C:/Users/ammu4/AppData/Local/Programs/Python/Python311/python.exe "d:/s_i
* 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 inst
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5001
* Running on http://192.168.145.67:5001
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 124-491-721

```

Now, open the postman tool app

And call the “/logindata” method and enter the mobile,password ..

If the login details are valid, it displays the “welcome to the notes manager”..i.e., its navigated to the dashboard page otherwise it displays “invalid credentials”

127.0.0.1:5001/logindata

Add to collection

POST127.0.0.1:5001/logindata

Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

Cookies

noneform-datax-www-form-urlencodedrawbinary

	Key	Value	...	Bulk Edit
<input checked="" type="checkbox"/>	mobile	9347244193		
<input checked="" type="checkbox"/>	password	abc123		
	Key	Value		

BodyCookies (1)Headers (6)Test Results

Status: 200 OKTime: 11 msSize: 1.44 KBSave Response

PrettyRawPreviewVisualizeHTML

```
19<li class="active"><a href="/dashboard/home">Home</a></li>
20<li><a href="/insertnotes">Insert notes</a></li>
21<li><a href="/viewnotes">view notes</a></li>
22<li><a href="/logout">Logout</a></li>
23</ul>
24</div>
25</nav>
26<div class="container">
27  <marquee>
28    <h2> Welcome to my notes Manager</h2>
29  </marquee>
30</div>
31</body>
32
33</html>
```

127.0.0.1:5001/logindata

Add to collection

POST127.0.0.1:5001/logindata

Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

Cookies

noneform-datax-www-form-urlencodedrawbinary

	Key	Value	...	Bulk Edit
<input checked="" type="checkbox"/>	mobile	9347244193		
<input checked="" type="checkbox"/>	password	abc12		
	Key	Value		

BodyCookies (1)Headers (5)Test Results

Status: 200 OKTime: 28 msSize: 2.34 KBSave Response

PrettyRawPreviewVisualizeHTML

```
36<input type="password" class="form-control" name="password" placeholder="Enter password"/>
37</div>
38<div class="form-group">
39  <button class="btn btn-primary" type="submit">Login</button>
40</div>
41</form>
42<span style="color:red">invalid credentials</span>
43<span style="color:green"></span>
44</div>
45
46<div class="col-sm-4"></div>
47</center>
48<a href="https://www.Ammu.com">Ammu</a>
49</body>
50
```

Now call the “/insertnotesdata” and enter the notes

Then it displays that notes saved.

The screenshot shows a REST client interface for a POST request to `127.0.0.1:5001/insertnotesdata`. The request body is set to `form-data` and contains a single key-value pair: `notes` with the value `Hello makeskilled!!!!!!!!!!!!!!!!!!!!!!!!!!!!`. The response status is `200 OK` with a time of `29 ms` and a size of `1.84 KB`. The response body is displayed in HTML format, showing a form with a label `Enter the notes`, a text area with `rows="10" cols="100" name="notes"`, and a submit button. Below the form, a message `Notes saved` is displayed in blue.

```
28 <div class="container col-sm-6">
29 <form action="/insertnotesdata" method="post">
30   <div class="form-group">
31     <label>Enter the notes</label><br/>
32     <textarea rows="10" cols="100" name="notes"></textarea>
33   </div>
34   <button type="submit" class="btn btn-warning">Insert notes</button>
35 </form>
36 <span style="color: blue">Notes saved</span>
37
38 </div>
39 </body>
40
41 </html>
```

Now call the “/viewnotes” by selecting “GET” method. It displays all the stored notes from the mongodb.

The screenshot shows a REST client interface for a GET request to `127.0.0.1:5001/viewnotes`. The response status is `200 OK` with a time of `29 ms` and a size of `2.49 KB`. The response body is displayed in HTML format, showing a table with 4 rows of notes. Each row contains an index (1-4) and the note text.

```
32
33 <tr>
34   <td> 1</td>
35   <td> hii ammu</td>
36 </tr>
37
38 <tr>
39   <td> 2</td>
40   <td> hii hlo</td>
41 </tr>
42
43 <tr>
44   <td> 3</td>
45   <td> ammu</td>
46 </tr>
47
48 <tr>
49   <td> 4</td>
50   <td> Hello world</td>
51 </tr>
```