

Assignment-6

Store notes in the MongoDB from FrontEnd

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>
```

```
    </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

```

Copy the link and paste it on the browser.

Full Stack Demo Home Register Login

Register Page

Ammu

Name:
Enter Name

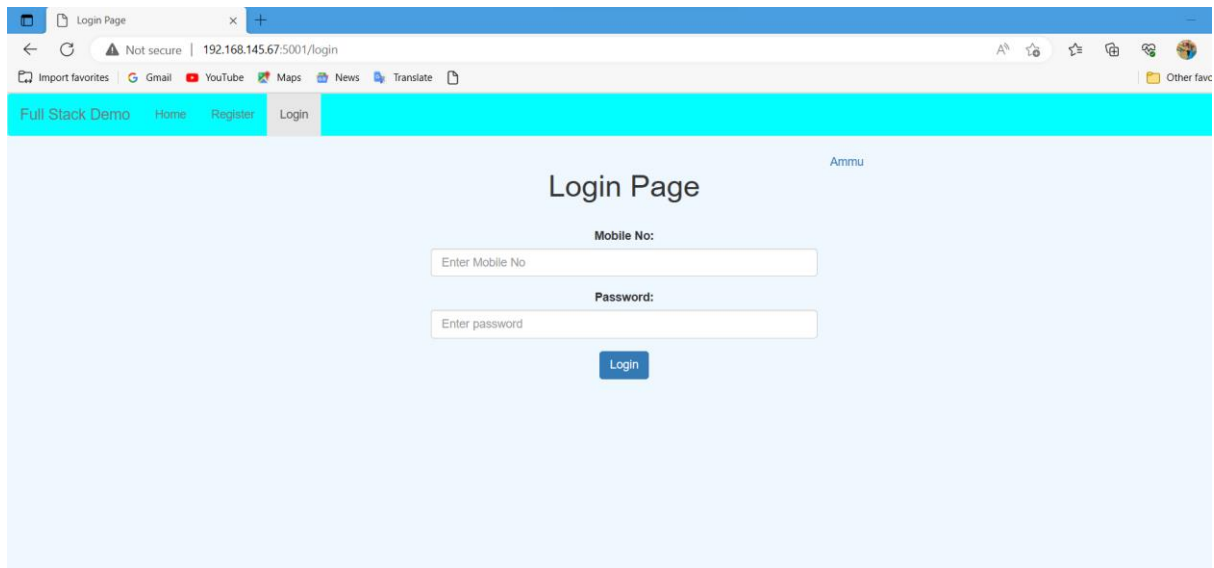
Email:
Enter Email Id

Mobile No:
Enter Mobile No

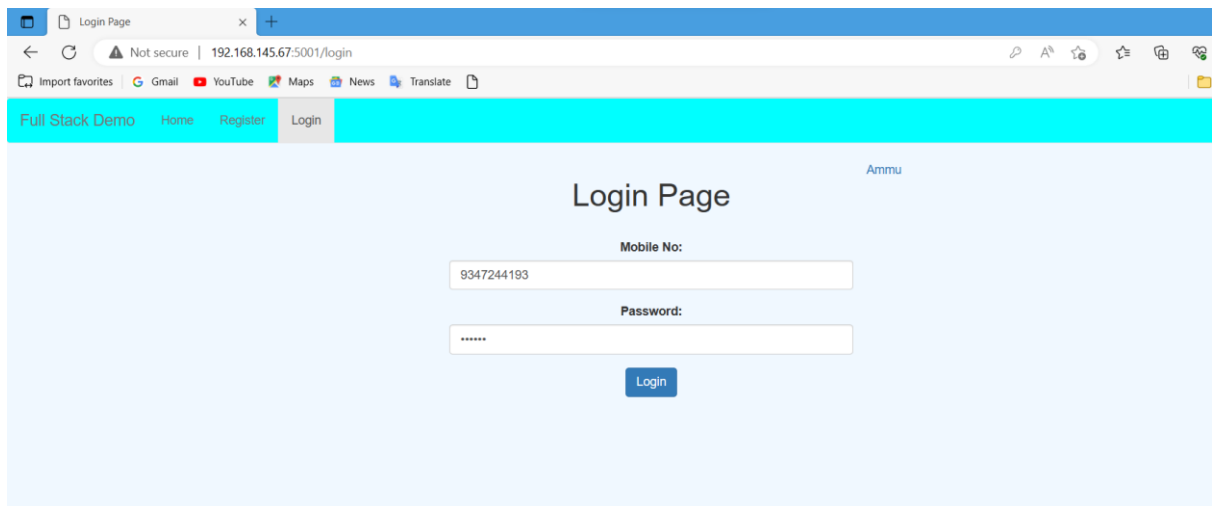
Password:
Enter password

Register

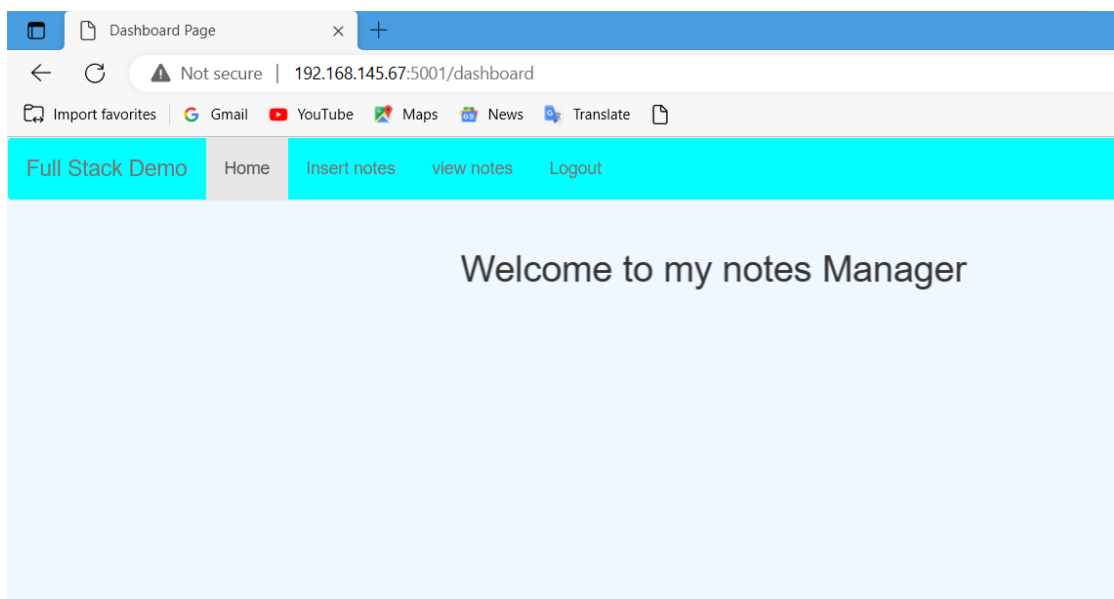
Here,by default register page was opened ,when we click on the login,it navigates to the login page



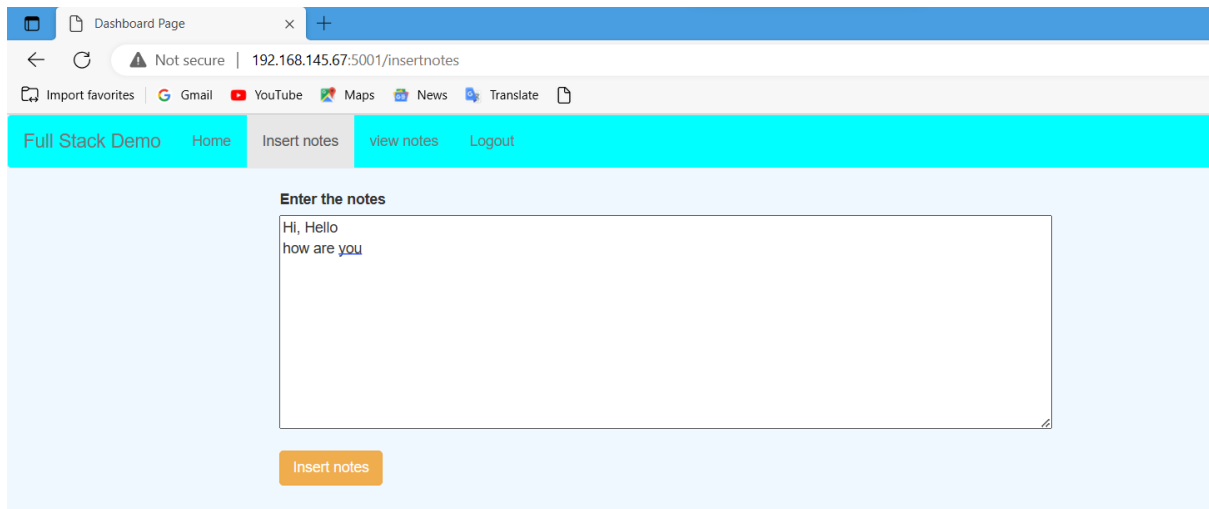
Now, enter the registered mobile.no,password and click on the login



If the login details are correct ,then it navigates to the "dashboard" page



Now,click on the insertnotes option



Dashboard Page x +

Not secure | 192.168.145.67:5001/insertnotes

Import favorites | Gmail | YouTube | Maps | News | Translate

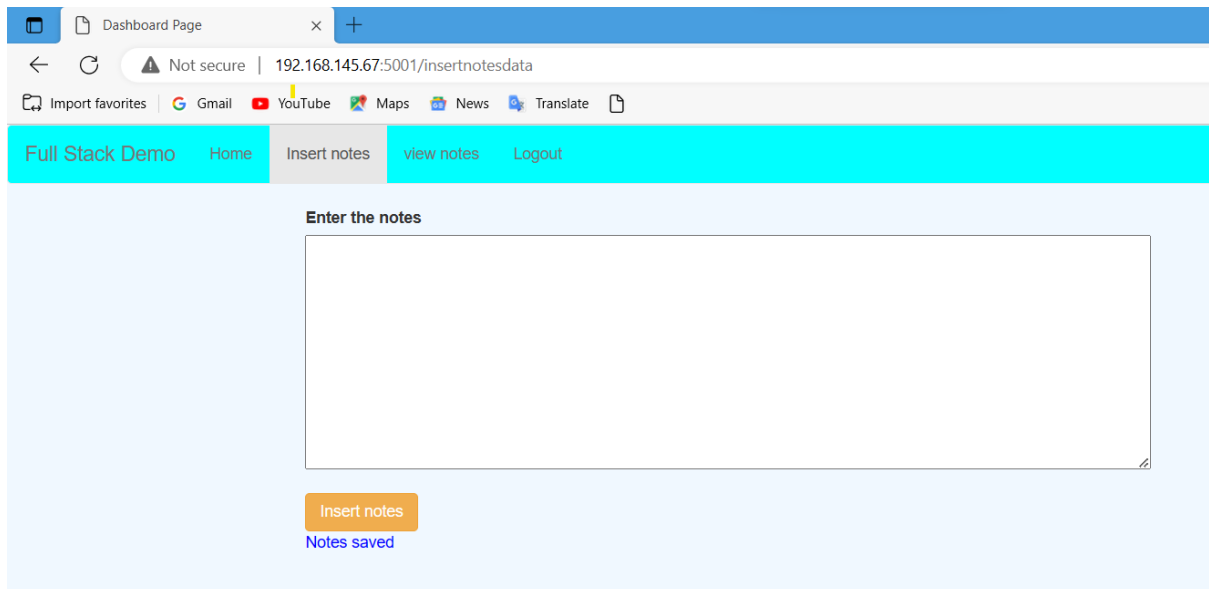
Full Stack Demo Home Insert notes view notes Logout

Enter the notes

Hi, Hello
how are you

Insert notes

Enter the notes ,and click on insertnotes button.It shows that “notes saved”.



Dashboard Page x +

Not secure | 192.168.145.67:5001/insertnotesdata

Import favorites | Gmail | YouTube | Maps | News | Translate

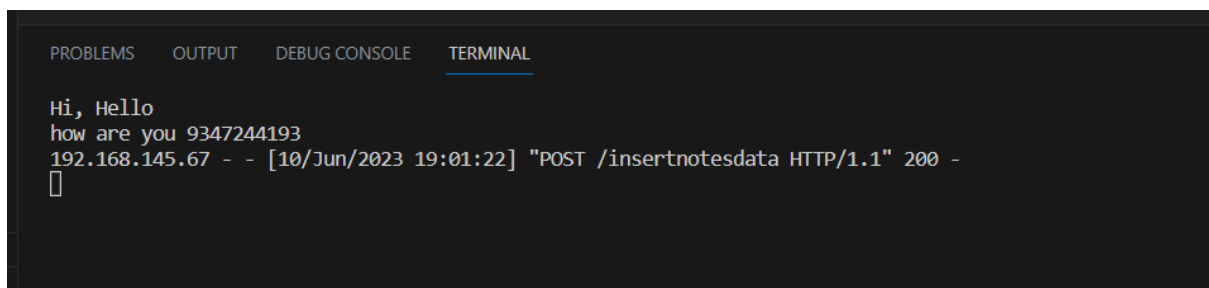
Full Stack Demo Home Insert notes view notes Logout

Enter the notes

Insert notes

Notes saved

At the server side, it shows that what data has been inserted and owner of the data



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
  
Hi, Hello  
how are you 9347244193  
192.168.145.67 - - [10/Jun/2023 19:01:22] "POST /insertnotesdata HTTP/1.1" 200 -  
[]
```

And also the data is stored in mongoddb

```
C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe
switched to db Ammu
> show collections;
flaskapp
mss
msss
notes
> db.notes.find();
{ "_id" : ObjectId("6484134302755a8af1e0378d"), "owner" : "9347244193", "notes" : "hii ammu" }
{ "_id" : ObjectId("64842d7eea67cf16c92cb99c"), "owner" : "9347244193", "notes" : "hii hlo" }
{ "_id" : ObjectId("64842e960331f7948b787fb1"), "owner" : "9347244193", "notes" : "ammu" }
{ "_id" : ObjectId("6484334adc95a02af22e3963"), "owner" : "9347244193", "notes" : "Hello world" }
{ "_id" : ObjectId("64847b29aecb3d3d8e3b78e0"), "owner" : "9347244193", "notes" : "Hi, Hello \r\nhow are you" }
>
```