

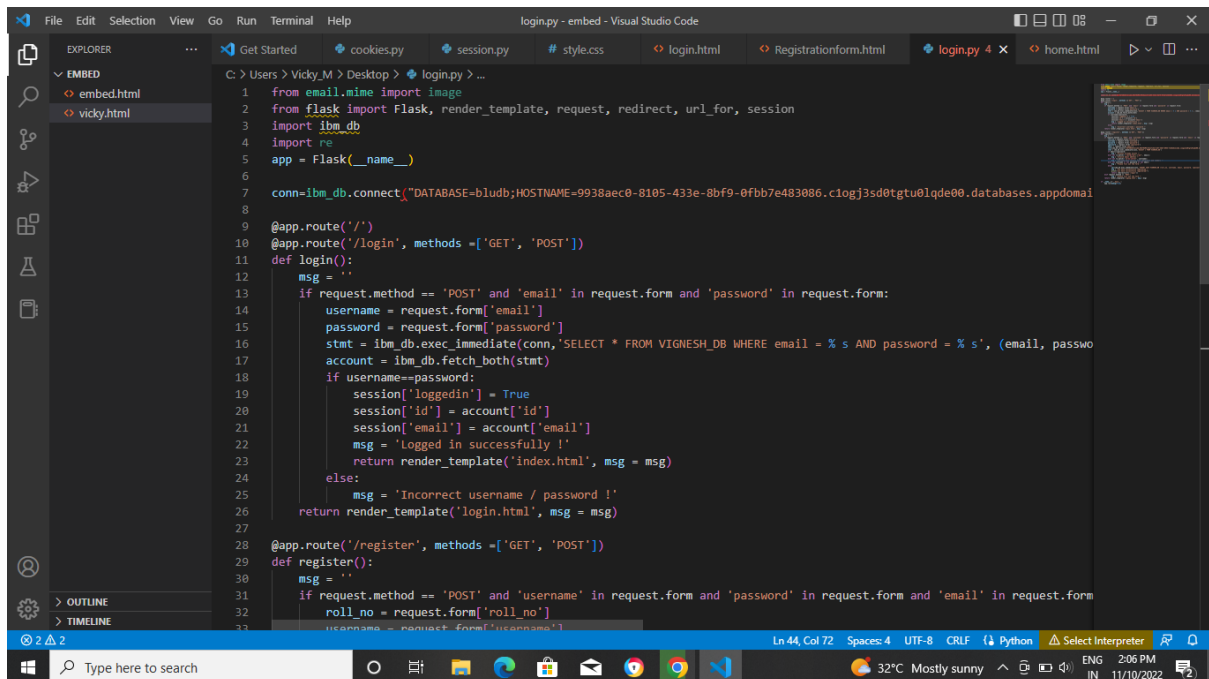
ASSIGNMENT -2

FLASK PROGRAM

Student Name	Vignesh M
Student Roll Number	731719205023
Maximum Marks	2 Marks

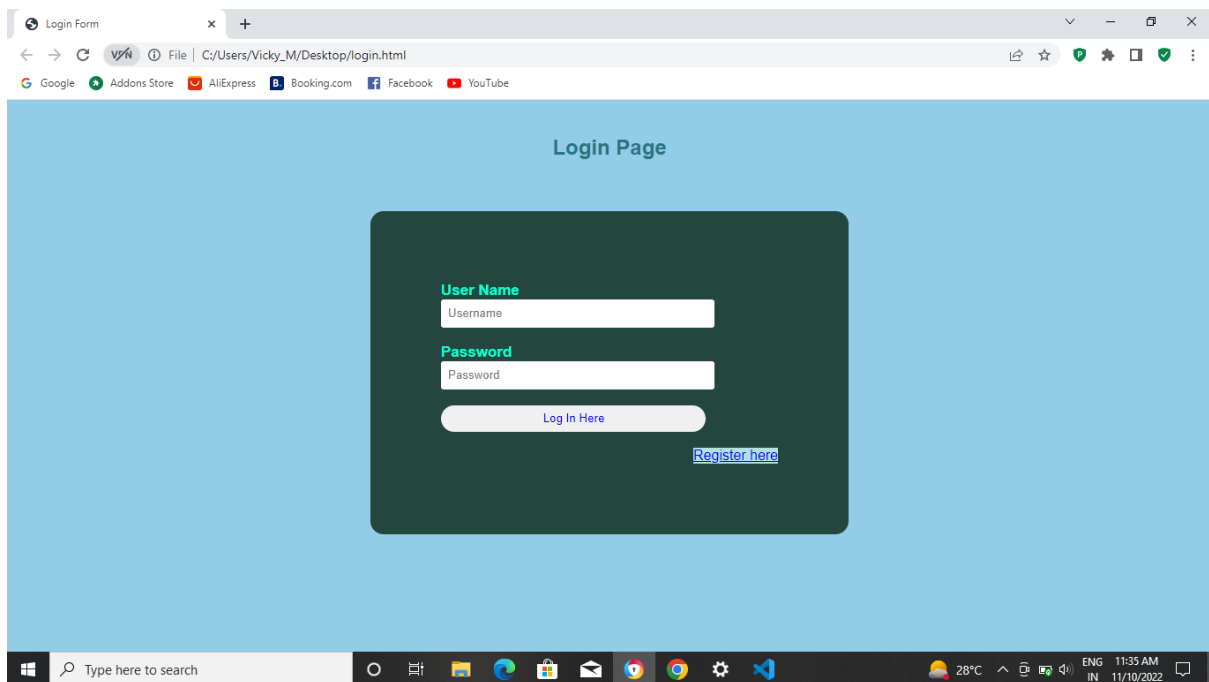
- 1) Write a Flask Program Which Should Display Name, Email, Phone and it Should Display the Same Once We Hit Submit.

CODE:



```
login.py - embed - Visual Studio Code
C:\Users\Vicky.M\Desktop> login.py > ...
1 from email.mime import image
2 from flask import Flask, render_template, request, redirect, url_for, session
3 import ibm_db
4 import re
5 app = Flask(__name__)
6
7 conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8195-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomai
8
9 @app.route('/')
10 @app.route('/login', methods =['GET', 'POST'])
11 def login():
12     msg = ''
13     if request.method == 'POST' and 'email' in request.form and 'password' in request.form:
14         username = request.form['email']
15         password = request.form['password']
16         stmt = ibm_db.exec_immediate(conn,'SELECT * FROM VIGNESH_DB WHERE email = % s AND password = % s', (email, passwo
17         account = ibm_db.fetch_both(stmt)
18         if username==password:
19             session['loggedin'] = True
20             session['id'] = account['id']
21             session['email'] = account['email']
22             msg = 'Logged in successfully !'
23             return render_template('index.html', msg = msg)
24         else:
25             msg = 'Incorrect username / password !'
26             return render_template('login.html', msg = msg)
27
28 @app.route('/register', methods =['GET', 'POST'])
29 def register():
30     msg = ''
31     if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form
32         roll_no = request.form['roll_no']
33         username = request.form['username']
```

```
File Edit Selection View Go Run Terminal Help login.py - Visual Studio Code
C:\Users\DELL\Downloads> login.py > register
24         else:
25             msg = 'Incorrect username / password !'
26             return render_template('login.html', msg = msg)
27
28 @app.route('/register', methods = ['GET', 'POST'])
29 def register():
30     msg = ''
31     if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form and 'roll_no' in request.form :
32         roll_no = request.form['roll_no']
33         username = request.form['username']
34         password = request.form['password']
35         cpassword = request.form['cpassword']
36         email = request.form['email']
37         conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c10g33sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;PROTOCOL=TCP;SECURITY=SSL;SSL_VERIFY_SERVER_CERT=0")
38         stmt = ibm_db.exec_immediate(conn,"SELECT * FROM JEEVITHA_DB")
39         if email==cpassword:
40             msg = 'Account already exists !'
41         elif not re.match(r'[^@]+\@+\.+[^@]+', email):
42             msg = 'Invalid email address !'
43         elif not re.match(r'[A-Za-z0-9]+', username):
44             msg = 'Username must contain only characters and numbers !'
45         elif not username or not password or not email:
46             msg = 'Please fill out the form !'
47         else:
48             sql=ibm_db.exec_immediate(conn,"INSERT INTO JEEVITHA_DB (roll_no, username, email, password, cpassword) VALUES (%s, %s, %s, %s, %s)")
49             msg = 'You have successfully registered !'
50             print('you have successfully registered!')
51             return redirect(url('/login'))
52     elif request.method == 'POST':
53         msg = 'Please fill out the form !'
54         return render_template('signup.html', msg = msg)
55
56 if __name__ == '__main__':
57     app.run(debug=True)
```



Browser tabs: Login Form, Registration Form, +

Address bar: C:/Users/Vicky_M/Desktop/Registrationform.html

Navigation: Google, Addons Store, AliExpress, Booking.com, Facebook, YouTube

REGISTRATION FORM

Username

Email

Password

Confirm Password

[Login here](#)



Browser tabs: Login Form, Registration Form, HOME PAGE, +

Address bar: C:/Users/Vicky_M/Desktop/home.html

Navigation: Google, Addons Store, AliExpress, Booking.com, Facebook, YouTube

WELCOME



The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, displaying a table named 'PJPW44280.VIGNESH'. The table has five columns: 'User Name', 'Roll Number', 'EMAIL', 'PASSWORD', and 'C password'. The table contains four rows of data. A 'Back' button is in the top right, and an 'Export to CSV' button is in the top right of the table area.

User Name	Roll Number	EMAIL	PASSWORD	C password
devi	3	devi@gmail.com	67890	67890
jeevitha	7	jeevi@gmail.com	9876	9876
solaimathi	19	solai@gmail.com	54321	54321
vignesh	23	vignesh@gmail.com	12345	12345

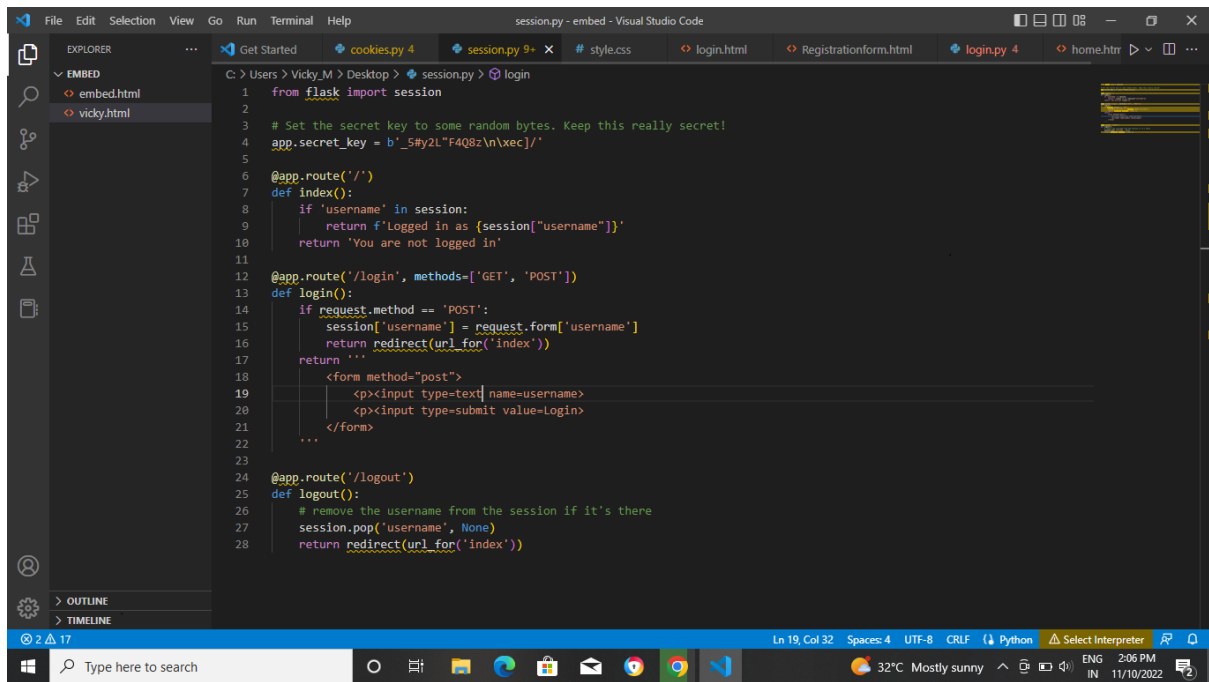
2) Write a Flask Program Which Should Cover Cookies, Sessions.

The screenshot shows the Visual Studio Code editor with a Flask application named 'cookies.py'. The code defines two routes: 'index' and 'index()'. The 'index' route uses 'request.cookies.get()' to retrieve the 'username' from the cookies. The 'index()' route uses 'make_response(render_template(...))' to render a template and 'resp.set_cookie()' to set a cookie with the username. The status bar at the bottom indicates the file is at 'Ln 3, Col 47' with 'Spaces: 4', 'UTF-8', 'CRLF', and 'Python' interpreter.

```

1 @app.route('/')
2 def index():
3     username = request.cookies.get('username')
4     def index():
5         resp = make_response(render_template(...))
6         resp.set_cookie('username', 'the username')
7         return resp

```



3) Write a Flask Program Which Should Display Resume Details Also have Upload Resume option by Using File Uploading.

```

4) <!DOCTYPE html>
5) <html>
6)     <head>
7)         <title>RESUME</title>
8)         <meta name="viewport" content="width=device-width,initial-
scale=1.0">
9)         <link rel="stylesheet" href="style5.css">
10)
11)     </head>
12)     <body>
13)
14)         <div class="wrapper">
15)             <div class="inner">
16)
17)                 <form style="justify-content: center;">
18)                     <center>
19)                         <h1>
20)                             <b>RESUME</b></h1>
21)                             FIRST NAME
22)                             <div class=""form-wrapper">
23)                                 <input type="text" placeholder="User name"
class="form-control">
24)                             </div><br>

```

```

25)                LAST NAME
26)                <div class=""form-wrapper">
27)                    <input type="text" placeholder="User name"
class="form-control">
28)                </div><br>
29)                EMAIL
30)                <div class=""form-wrapper">
31)                    <input type="text" placeholder="Email"
class="form-control">
32)                </div><br>
33)                PASSWORD
34)                <div class="form-wrapper">
35)                    <input type="password" placeholder="password"
class="form-control">
36)                </div> <br>
37)                PHONE NO
38)                <div class="form-wrapper">
39)                    <input type="Phone no" placeholder="Phone no"
class="form-control"></div><br>
40)                GENDER
41)                    <div class="form-wrapper">
42)                        <input type="radio" placeholder="male" class="form
-control">MALE
43)                        <input type="radio"
placeholder="female" class="form-control">FEMALE
44)                    </div><br><br>
45)                    <p>Click on the "Choose File" button to upload a
file:</p>
46)                <input type="file" id="myFile" name="filename"><br><br>
47)
48)                <input type="submit">
49)
50)            </center>
51)        </form>
52)    </div>
53)    </body>

```

RESUME

FIRST NAME

User name

LAST NAME

User name

EMAIL

Email

PASSWORD

password

PHONE NO

Phone no

GENDER

☐ MALE ☒ FEMALE

Click on the "Choose File" button to upload a file:

Choose file No file chosen

Submit

Activate Windows
Go to Settings to activate Windows.