Assignment -2

Assignment Date	19 September 2022
Student Name	Nanthakumaran S
Student Roll Number	737819ECR113
Maximum Marks	2 Marks

Question-1:

Create User table with user with email, username, roll number, password.

```
CREATE TABLE User (

   email varchar(255),

   username varchar(255),

   roll_no varchar(255),

   password varchar(255)

);
```

Question-2:

Perform UPDATE,DELETE Queries with user table

```
UPDATE User

SET roll_no = '737819ECR142@smartinternz.com'

WHERE email = 'rithikasivakumar8@gmail.com';



DELETE FROM User WHERE username='rithika';
```

Question-3:

Connect python code to db2.

```
db_connectionibm_db.connect("DATABASE=bludb;QUERYTIMEOUT=1;CONNECTTIMEOUT=10;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=./DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=nrl34308;PWD=7jHTeAV2cbra6tnD", "", "")
```

Question-4:

Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page

app.py

```
from flask import Flask

from flask import request, redirect, url_for, render_template

import ibm_db



app = Flask(__name__)

app.config['DEBUG'] = True

db_connection
=ibm_db.connect("DATABASE=bludb;QUERYTIMEOUT=1;CONNECTTIMEOUT=10;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURITY=SSL;SSLServerCertificate=./DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=nrl34308;PWD=7jHTeAV2cbra6tnD", "", "")



@app.route("/", methods=["GET", "POST"])

def register():

   if request.method == 'POST':
```

```
        data = request.form.to_dict()

        print(data)

          sql_query = f"INSERT INTO User (username, email, roll_no, password)
VALUES('{data['username']}',      '{data['email']}',      '{data['roll_no']}',
'{data['password']}')"

        ibm_db.exec_immediate(db_connection, sql_query)

        return redirect(url_for("login"))

    if request.method == "GET":

        return render_template("registration.html")




@app.route("/login", methods=["GET", "POST"])

def login():

    if request.method == 'POST':

        data = request.form.to_dict()

                sql_query  =  f"SELECT  password  from  User  WHERE  username  =
'{data['username']}'"

        result = ibm_db.exec_immediate(db_connection, sql_query)

        value = ibm_db.fetch_tuple(result)

        if value[0] == data["password"]:

            return redirect(url_for("welcome"))

        else:

            return "<p style=color:red;>Invalid Credentials</p>"

    if request.method == "GET":

        return render_template("login.html")
```

```
@app.route("/welcome", methods=["GET"])
def welcome():
   return "<h1 style=color:green;>Welcome!!!!!</h1>"

if __name__ == '__main__':
   app.run()
```

templates/registration.html

```
 <!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Registration Form</title>
                <link     rel="stylesheet"     href="{{     url_for('static',
filename='registration.css') }}">
</head>

<body>
   <form method="POST" action="{{url_for('register')}}">
       <div class="form">
       <input name="username" placeholder="Username" type="text" />
```

```
        <br />
        <input name="email" placeholder="contact@example.com" type="email" />
        <br />
        <input name="roll_no" placeholder="Roll number" type="text" />
        <br />
        <input name="password" placeholder="Password" type="password" />
        <br />
        <button type="submit">Register</button>
        </div>
    </form>
</body>


</html>
```

templates/login.html

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Registration Form</title>
                <link     rel="stylesheet"     href="{{     url_for('static',
filename='registration.css') }}">
```

```
</head>


<body>
   <form method="POST" action="{{url_for('login')}}">
       <div class="form">
           <input name="username" placeholder="Username" type="text" />
       <br />
       <input name="password" placeholder="Password" type="password" />
       <br />
       <button type="submit">Login</button>
       </div>
   </form>
</body>


</html>
```

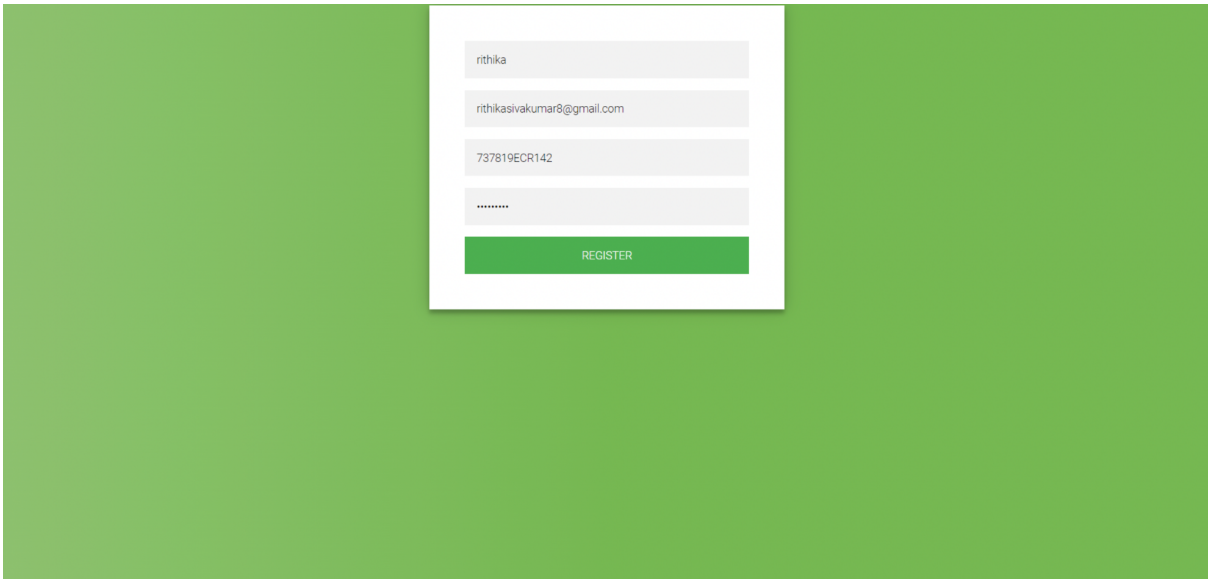
static/index.css

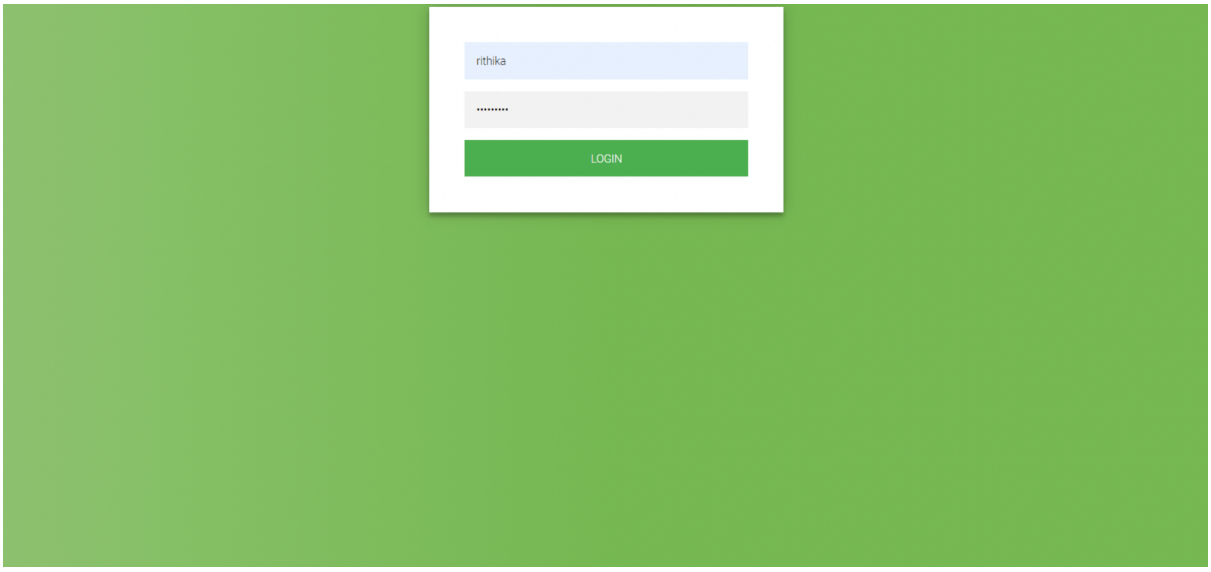
```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);


.form {
 position: relative;
 z-index: 1;
 background: #FFFFFF;
 max-width: 360px;
 margin: 0 auto 100px;
```

```
 padding: 45px;
 text-align: center;
 box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
input {
 font-family: "Roboto", sans-serif;
 outline: 0;
 background: #f2f2f2;
 width: 100%;
 border: 0;
 margin: 0 0 15px;
 padding: 15px;
 box-sizing: border-box;
 font-size: 14px;
}
button {
 font-family: "Roboto", sans-serif;
 text-transform: uppercase;
 outline: 0;
 background: #4CAF50;
 width: 100%;
 border: 0;
 padding: 15px;
 color: #FFFFFF;
 font-size: 14px;
```

```
 -webkit-transition: all 0.3 ease;

 transition: all 0.3 ease;

 cursor: pointer;

}

.form button:hover,.form button:active,.form button:focus {

 background: #43A047;

}



body {

 background: #76b852;

 background: rgb(141,194,111);

 background: linear-gradient(90deg, rgba(141,194,111,1) 0%, rgba(118,184,82,1)
50%);

 font-family: "Roboto", sans-serif;

 -webkit-font-smoothing: antialiased;

 -moz-osx-font-smoothing: grayscale;

}
```

Output


rithika
rithikasivakumar8@gmail.com
737819ECR142
REGISTER


rithika
LOGIN

Welcome!!!!!