Assignment – 2

Assignment Date	18 OCTOBER 2022
Student Name	Mr. KISHORE KUMAR
Student Roll Number	811519104058
Maximum Marks	2 Marks

Question 1:

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

2. Perform UPDATE,DELETE Queries with user table

3. Connect python code to db2.

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 with username and password. If the user is valid show the welcome page

HTML CODE

created.html

```
<h2>User Created</h2>

<h4><a href="/login">Click Here to Login</a></h4>
```

dashboard.html

```
<html>
  <head>
    <title>Welcome</title>
  </head>
  <body>
    <h2>Welcome to our Website, {{name}}</h2>
  </body>
</html>
```

index.html

```
<html>
  <head>
    <title>
      User Registration
```

```
    </title>
    <style>
        * {box-sizing: border-box}

/* Add padding to containers */
.container {
  padding: 16px;
  width: 40%;
  margin: auto;
}
h2{
   color: red;
}
/* Full-width input fields */
input[type=text], input[type=password],input[type=email],input[type=number] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}

input[type=text]:focus,
input[type=password]:focus,input[type=email]:focus,input[type=number]:focus {
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
```

```
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit/register button */
.registerbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

.registerbtn:hover {
  opacity:1;
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
}
      </style>
```

```
  </head>
  <body>
    <form id="myForm" name="myForm" action="{{ url_for("register")}}" method="post">
      <div class="container">
      {% if message=="User Already Exists" %}
        <h2>User Already Exists</h2>
        {% elif message=="Error While Creating User" %}
        <h2>Error While Creating user</h2>
        {% endif %}
        <h1>Register User</h1>
        <hr>

        <label for="email"><b>Email</b></label>
        <input type="email" placeholder="Enter Email" name="email" id="email" required>

        <label for="username"><b>Username</b></label>
        <input type="text" placeholder="Enter Username" name="username" id="username"
required>

        <label for="rollno"><b>Roll Number</b></label>
        <input type="number" placeholder="Enter Roll Number" name="rollno" id="rollno"
required>

        <label for="psw"><b>Password</b></label>
        <input type="password" placeholder="Enter Password" name="psw" id="psw" required>

        <hr>
        <button type="submit" class="registerbtn" >Register</button>
      </div>
    </form>
  </body>
```

```
</html>
```

login.html

```
<html>
   <head>
      <title>
         User Login
      </title>
      <style>
         * {box-sizing: border-box}

/* Add padding to containers */
.container {
  padding: 16px;
  width: 40%;
  margin: auto;
}
h2{
   color: red;
}
/* Full-width input fields */
input[type=text], input[type=password],input[type=email],input[type=number] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}
```

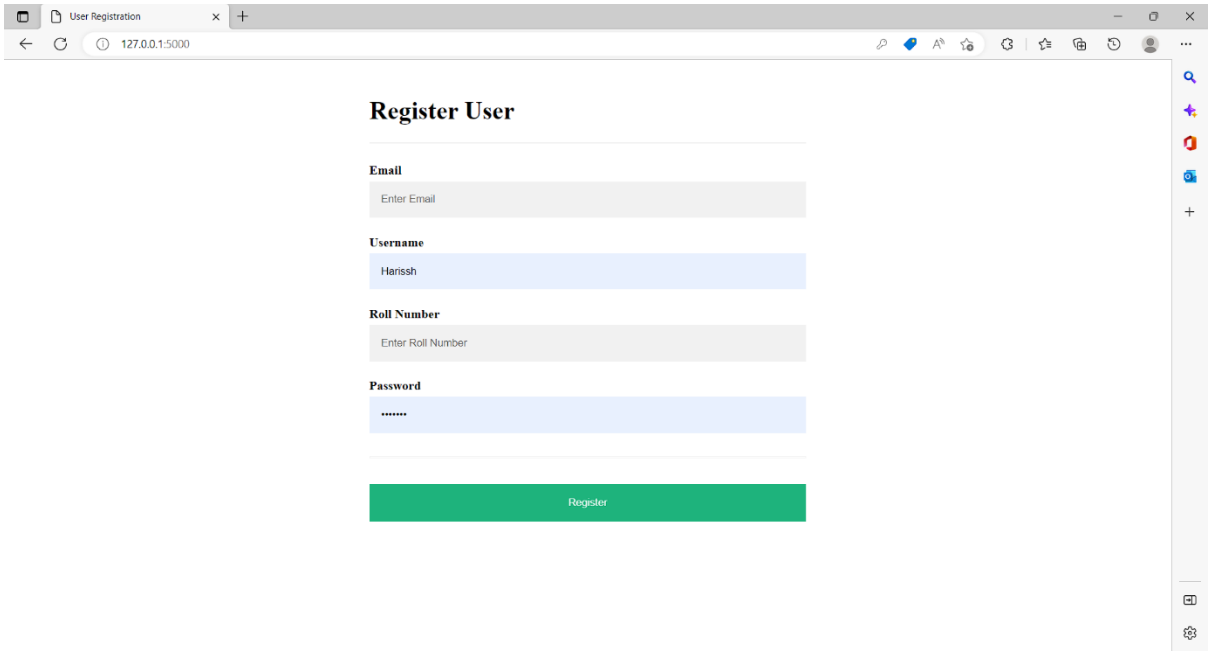
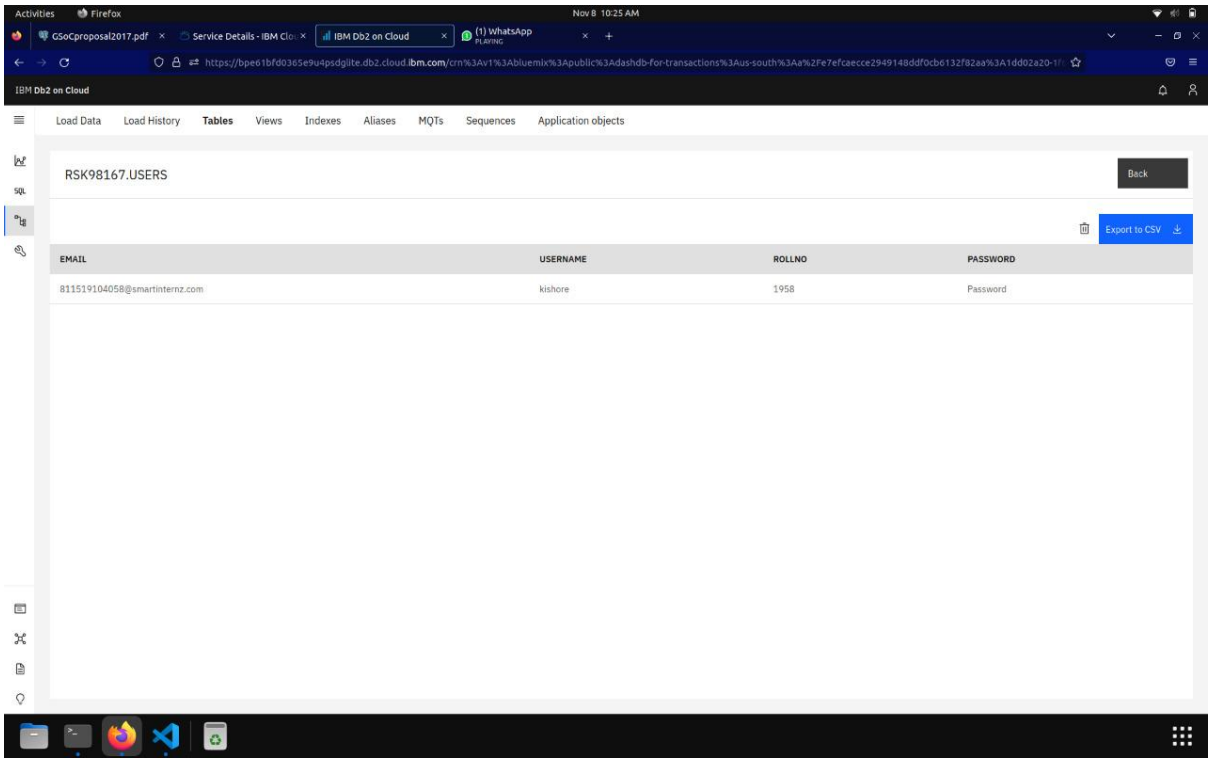
```
input[type=text]:focus,
input[type=password]:focus,input[type=email]:focus,input[type=number]:focus {
  background-color: #ddd;
  outline: none;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit/register button */
.registerbtn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

.registerbtn:hover {
  opacity:1;
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
```

```
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
}
      </style>
   </head>
   <body>
      <form id="myForm" name="myForm" action="{{ url_for("login")}}" method="post">
         <div class="container">
         {% if message=="Username Not Found" %}
            <h2>User Not Found</h2>
            {% elif message=="Wrong Password" %}
            <h2>Wrong Password</h2>
            {% endif %}
           <h1>Login User</h1>
           <hr>
           <label for="username"><b>Username</b></label>
           <input type="text" placeholder="Enter Username" name="username" id="username"
required>
           <label for="psw"><b>Password</b></label>
           <input type="password" placeholder="Enter Password" name="psw" id="psw" required>


           <hr>
           <button type="submit" class="registerbtn" >Login</button>
          </div>
        </form>
   </body>
</html>
```

app.py

```
from distutils.log import debug

from email import message

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

import ibm_db

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;PROTOCOL=TCPIP;SE
CURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=rsk98167;PWD=lC3PwBvgyonwS7N
T;", "", "")

app=Flask(__name__)


@app.route("/")

def homepage():

   return render_template("index.html")

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

def register():

   if request.method == "POST":

      sql="select * from users where email='"+request.form.get("email")+"' or
username='"+request.form.get("username")+"'";

      stmt = ibm_db.exec_immediate(conn, sql)

      result = ibm_db.fetch_both(stmt)

      if result:

         print("User Already Exists")

         return render_template("index.html",message="User Already Exists")

      email = request.form.get("email")

      username = request.form.get("username")

      rollno = request.form.get("rollno")

      passw=request.form.get("psw")

      sql="insert into users values('"+email+"','"+username+"',"+rollno+",'"+passw+"')";

      stmt = ibm_db.exec_immediate(conn, sql)

      if(ibm_db.num_rows(stmt)==1):
```

```
            return render_template("created.html")
        else:
            return render_template("index.html",message="Error While Creating User")
    return render_template("index.html");

@app.route("/login", methods =["GET", "POST"])
def login():
    if request.method == "POST":
        username = request.form.get("username")
        passw=request.form.get("psw")
        sql="select * from users where username='"+request.form.get("username")+"'";
        stmt = ibm_db.exec_immediate(conn, sql)
        result = ibm_db.fetch_both(stmt)
        if not result:
            print("User Not Exists")
            return render_template("login.html",message="Username Not Found")
        sql="select * from users where username='"+request.form.get("username")+"' and
password='"+passw+"'";
        stmt = ibm_db.exec_immediate(conn, sql)
        result = ibm_db.fetch_both(stmt)
        if not result:
            print("Wrong Password")
            return render_template("login.html",message="Wrong Password")
        # return render_template("login.html",message="Logging in")
        return redirect(url_for('dashboard',name = username))
    return render_template("login.html")
@app.route('/dashboard/<name>')
def dashboard(name):
    return render_template("dashboard.html",name=name)
if __name__=="__main__":
    if conn:
```

```
print("Database Connected Successfully")
app.run(debug=True)
```

OUTPUT