

Assignment -2

Assignment Date	23 September 2022
Student Name	Mr.P. Akilan
Student Roll Number	621319205003
Maximum Marks	2 Marks

Question-1:

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

```
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min
.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/particlesjs/2.2.3/particles.min.js">
</head>
<style>
  body {
    background: linear-gradient(to left, black, #f64c72);
    background-attachment: fixed;
    background-repeat: no-repeat;
    background-size: cover;
    margin: 0;
    padding: 0;
  }

  .form-control,
  .form-control:focus,
  .input-group-addon {
    border-color: rgb(0, 0, 0);
    border-radius: 0;
  }

  .signup {
    width: 360px;
    margin: 0 auto;
    padding: 50px 0;
  }

  .signup h2 {
    color: black;
    text-align: center;
    margin: 0 0 20px;
  }

  .signup form {
    border-radius: 10px;
```

```

    margin-bottom: 15px;
    margin-top: 150px;
    background-color: rgb(216, 47, 47) !important;
    border: 1px solid none;
    box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
    padding: 30px;
}

.signup .form-group {
    margin-bottom: 30px;
}

.signup .form-control {
    min-height: 50px;
    box-shadow: none !important;
    border-width: 0 0 1px 0;
}

.form-control {
    background-color: transparent;
    color: black;
}

.form-control:hover {
    background-color: transparent;
    color: white;
    overflow: hidden;
    text-decoration: none;
}

.signup .input-group-addon {
    max-width: 42px;
    text-align: center;
    background: none;
    border-bottom: 1px solid aliceblue;
    padding-left: 5px;
}

.input-group-addon {
    font-size: 30px;
}

.btn {
    background: linear-gradient(300deg, black, #f64c72);
    margin-left: 1px;
    border: none;
    text-align: center;
    padding: 3%;
    font-size: 16px;
    font-weight: bold;
    color: aliceblue;
}

.btn:hover {

```

```

    background: aliceblue;
    border: none;
    color: aliceblue;
    color: #fd614d;
}

.fa {
    color: aliceblue;
}

.signup .fa-envelope {
    font-size: 23px;
    color: aliceblue;
}

a {
    color: aliceblue;
    text-decoration: none;
}

a:hover {
    color: #714a4a;
    text-decoration: none;
}
</style>

<body>
<div class="signup">
<form action="{url_for(result)}" method="post">
<h2>LOGIN</h2>
<div class="form-group">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-envelope"></i></span>
<input type="email" name="email" class=" form-control" placeholder="Email" required>
</div>
</div>
<div class="form-group">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-lock"></i></span>
<input type="text" name="rollnumber" class=" form-control" placeholder="Roll Number"
required>
</div><br>
<div class="form-group">
<div class="input-group">
<span class="input-group-addon"><i class="fa fa-lock"></i></span>
<input type="password" name="password" class=" form-control" placeholder="Password"
required>
</div>
</div>
<div class="form-group">
<button type="submit" class="btn btn-primary btn-lg btn-block">Login</button>
</div>
</form>
</div>
</body>

```

</html>

Question 2:

Perform UPDATE, DELETE Queries with user table

```
import ibm_db
def display(conn):
    sql="SELECT * FROM USER"
    stmt = ibm_db.exec_immediate(conn,sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        print ("The roll no is : ", dictionary["ROLLNO"])
        print ("The Name is : ", dictionary["USERNAME"])
        print ("The Email no is : ", dictionary["EMAIL"])
        print ("The Password is : ", dictionary["PASSWORD"],"\n")
        dictionary = ibm_db.fetch_both(stmt)
def Insert(conn,rollno,name,email,password):
    sql="INSERT INTO USER
VALUES('{ }',{ }',{ }',{ })'.format(rollno,name,email,password)
    stmt=ibm_db.exec_immediate(conn,sql)
    print("The Number if affected rows: ", ibm_db.num_rows(stmt),"\n")
def Update(conn,rollno,name,email,password):
    sql="UPDATE Users SET ROLLNO =' { }', WHERE
USERNAME=' { }';".format(rollno,name)
    stmt=ibm_db.exec_immediate(conn,sql)
    print("The Number if affected rows: ", ibm_db.num_rows(stmt),"\n")
try:
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqn39u98g.databases.appdomain.cloud;POR
T=30756;SECURITY=SSL;
SSLServerCertificateDigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=n
hl80748;PWD=3yD0G9e6VuQHsOBX;"," "," ")
    print("connection Success")
    display(conn)

    Insert(conn,"621319205003","Akilan","akilanpalani2001@gmail.com","
euyfuis")
    display(conn)

except:
    print("couldn't connect")
```

Question 3 and 4:

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

```
import sys
sys.path.append('/workspace/python_db/dbconn.py')
from urllib import request
from flask import Flask,render_template,url_for
app = Flask(__name__)
@app.route("/")
def homepage():
    return render_template("index.html")
@app.route("/result",methods=['POST'])
def send_data():
    name=request.form['email']
    #dbconn.Insert_values()
    return "hello"
if __name__=="__main__":
    app.run(debug=True)
```

Output:

