

## ASSIGNMENT-2

Date	19-09-2022
Student Name	Praveena Betes
Roll no	821319104025
Maximum Marks	2 Marks

1. Create user table with email, username, roll number, password.
2. Perform update, delete, queries with user table.
3. Connect python code to db2.

```
import ibm_db

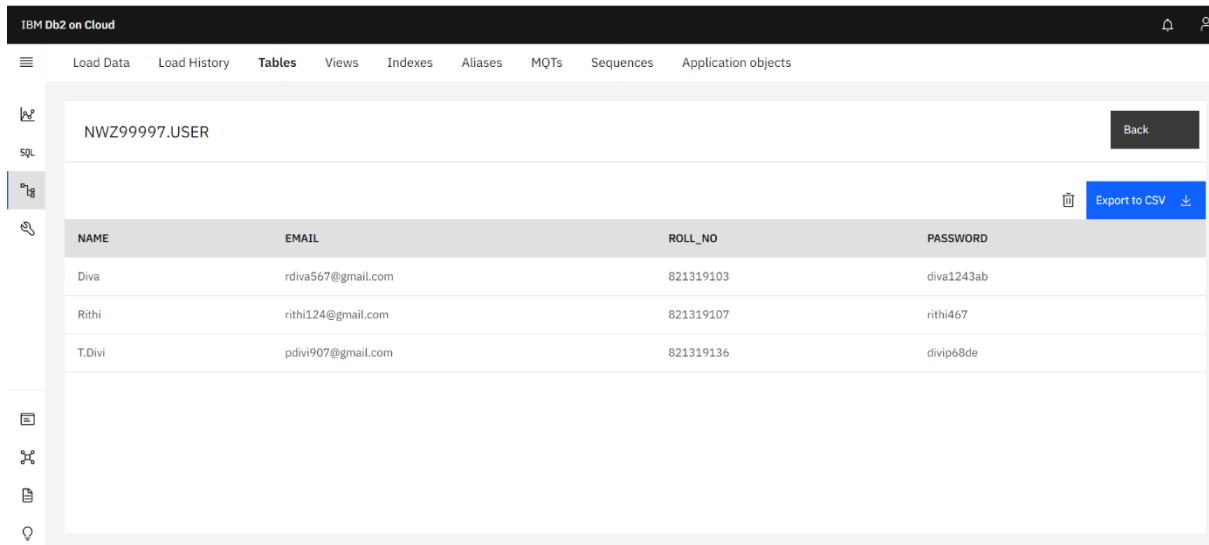
def list_all(conn):
    sql="SELECT * FROM User"
    stmt=ibm_db.exec_immediate(conn,sql)
    dictionary=ibm_db.fetch_both(stmt)
    while dictionary!=False:
        print("Name: ", dictionary["Name"])
        print("Email: ", dictionary["Email"])
        print("Roll no: ", dictionary["Roll no"])
        print("Password: ", dictionary["Password"],"\n")
        dictionary=ibm_db.fetch_both(stmt)

def insert_values(conn,name,email,roll_no,password):
    sql="INSERT INTO User
VALUES('{}','{}','{}','{}').format(name,email,roll_no,password)
    stmt=ibm_db.exec_immediate(conn,sql)
    print("Number of affected rows: ",ibm_db.num_rows(stmt),"\n")

try:
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;PROTOCO
L=TCPIP;UID=nwz99997;PWD=kZWVp8EeTNb2S7ec;", "", "")
    print("Connected to database")
    list_all(conn)
    insert_values(conn,"Rithi","rithi124@gmail.com","821319107","rithi467")
    insert_values(conn,"Diva","rdiva567@gmail.com","821319103","diva1243ab")
    insert_values(conn,'Divi','pdivi907@gmail.com','821319136','divip68de')
    stmt = ibm_db.exec_immediate(conn, "UPDATE name= 'T.divi'")

except:
    print("Error connecting to database")
```

## OUTPUT:



The screenshot shows the IBM Db2 on Cloud web interface. At the top, there's a navigation bar with 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. Below this, a sidebar on the left contains icons for various database functions. The main area displays a table named 'NWZ99997.USER'. Above the table, there's a 'Back' button and an 'Export to CSV' button. The table has four columns: 'NAME', 'EMAIL', 'ROLL\_NO', and 'PASSWORD'. It contains three rows of data.

NAME	EMAIL	ROLL_NO	PASSWORD
Diva	rdiva567@gmail.com	821319103	diva1243ab
Rithi	rithi124@gmail.com	821319107	rithi467
T.Divi	pdivi907@gmail.com	821319136	divip68de

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.

### Flask.py

```
from flask import Flask, render_template, request
from mydb import first as db
app=Flask(__name__)

@app.route("/")
def homepage():
    return render_template("signin.html")

@app.route("/signin")
def render_index1(accountok=True):
    if accountok==True:
        return render_template("signin.html",message="")
    else:
        return render_template("signin.html",message="Username/Password is wrong!!")

@app.route("/login",methods=['POST'])
def checklogin():
    username=request.form['username']
    email=request.form['email']
    pwd=request.form['password']
    if(username=="abcheie23" and email=="hey123@gmail.com" and
pwd=="abcdie23#"):
        return render_template("dashboard.html",user="abcd")
    else:
        return render_signin(False)
```

```
if __name__=="__main__":  
    app.run(debug=True)
```

signin.html

```
<!DOCTYPE html>  
<html>  
<head>  
<meta name="viewport" content="width=device-width, initial-scale=1">  
  
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/4.7.0/css/font-awesome.min.css">  
<style>  
body {font-family: Arial, Helvetica, sans-serif;}  
* {box-sizing: border-box;}  
  
.input-container {  
    display: -ms-flexbox;  
    display: flex;  
    width: 100%;  
    margin-bottom: 15px;  
}  
  
.icon {  
    padding: 10px;  
    background: dodgerblue;  
    color: white;  
    min-width: 50px;  
    text-align: center;  
}  
  
.input-field {  
    width: 100%;  
    padding: 10px;  
    outline: none;  
}  
  
.input-field:focus {  
    border: 2px solid dodgerblue;  
}  
  
/* Set a style for the submit button */  
.btn {  
    background-color: dodgerblue;  
    color: white;  
    padding: 15px 20px;  
    border: none;
```

```

    cursor: pointer;
    width: 100%;
    opacity: 0.9;
}
.btn1{
background-color: dodgerblue;
    color: white;
    padding: 15px 20px;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
}

.btn:hover {
    opacity: 1;
}
</style>
</head>
<body>

<form action="/action_page.php" style="max-width:500px;margin:auto">
    <h2><center>User Portal</h2></center>
    <div class="input-container">
        <i class="fa fa-user icon"></i>
        <input class="input-field" type="text" placeholder="Username"
name="usrnm">
    </div>

    <div class="input-container">
        <i class="fa fa-envelope icon"></i>
        <input class="input-field" type="text" placeholder="Email" name="email">
    </div>

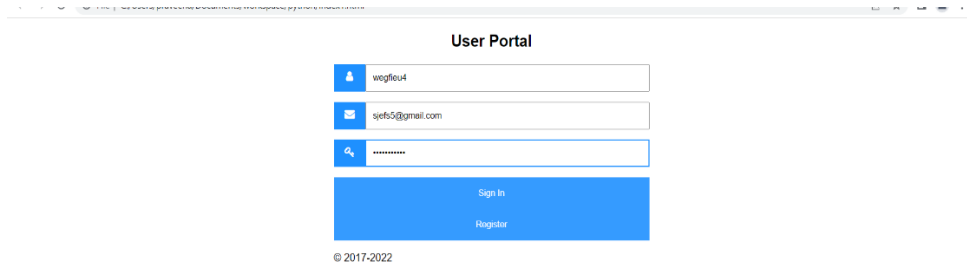
    <div class="input-container">
        <i class="fa fa-key icon"></i>
        <input class="input-field" type="password" placeholder="Password"
name="psw">
    </div>
    <button class="w-100 btn btn-lg btn-primary" type="submit">Sign In</button>
    <button class="w-100 btn btn-lg btn-primary registerButton"
type="button">Register</button>
    <p class="mt-5 mb-3 text-muted">&copy; 2017-2022</p>
</form>

</body>
</html>

```

## OUTPUT:

1.



The screenshot shows a web browser window displaying a 'User Portal' login form. The form is centered on the page and consists of three input fields stacked vertically, each with a small icon on the left: a person icon for the username field (containing 'wegfieu4'), an envelope icon for the email field (containing 'sief5@gmail.com'), and a magnifying glass icon for the password field (containing eight dots). Below these fields are two blue buttons: 'Sign In' and 'Register'. At the bottom left of the form area, there is a copyright notice: '© 2017-2022'.

2.

## WELCOME TO YOUR PAGE!!

### Get started with your page

Learn more about bootstrap

[Download examples](#)

### Design your page better

Ready to beyond the starter template? Check out these open source projects that you can quickly duplicate to a new GitHub repository.

- [Bootstrap npm starter](#)
- Bootstrap Parcel starter (coming soon!)

### Guides

Read more detailed instructions and documentation on using or contributing to Bootstrap.

- [Bootstrap quick start guide](#)
- [Bootstrap Webpack guide](#)
- [Bootstrap Parcel guide](#)
- [Bootstrap Vite guide](#)
- [Contributing to Bootstrap](#)