

# IBM Assignment -2

Project Title : Personal Expense Tracker Application

Team ID : PNT2022TMID32657

Name: Nithin T

Registration No : 813819104063

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

Table definition

USER

No statistics available.

Name	Data type	Nullable	Length	Scale	
EMAIL	VARCHAR	Y	32	0	👁
USERNAME	VARCHAR	Y	32	0	👁
ROLLNUMBER	VARCHAR	Y	32	0	👁
PASSWORD	VARCHAR	Y	32	0	👁

View data

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

BVZ19901

SQL

\*Untitled - 1

```
1 INSERT INTO USER VALUES('santy@gmail.com', 'santhosh', '221085', '098765');
2 INSERT INTO USER VALUES('sivu@gmail.com', 'sivaram', '221093', '098765');
3 INSERT INTO USER VALUES('sanjay@gmail.com', 'sanjay', '221085', '098765');
4 INSERT INTO USER VALUES('sundar@gmail.com', 'sundar', '221104', '098765');
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 31, 2022 7:52:52 PM	4	0.030 s
INSERT INTO USER VALUES('santy@gmail.com', 'santhosh', '221085', '098765')		✓	0.007 s
INSERT INTO USER VALUES('sivu@gmail.com', 'sivaram', '221093', '098765')		✓	0.007 s
INSERT INTO USER VALUES('sanjay@gmail.com', 'sanjay', '221085', '098765')		✓	0.009 s
INSERT INTO USER VALUES('sundar@gmail.com', 'sundar', '221104', '098765')		✓	0.007 s



IBM Db2 on Cloud

Data objects Saved objects

Filter objects

BVZ19901

SQL

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 31, 2022 7:55:27 PM	✓ 1	0.000 s
UPDATE USER SET password='123456' WHERE email='santy@gmail.com'		✓	0.000 s

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

BVZ19901.USER

Back

Export to CSV

EMAIL	USERNAME	ROLLNUMBER	PASSWORD
sanjay@gmail.com	sanjay	221085	098765
santy@gmail.com	santhosh	221085	123456
sivu@gmail.com	sivaram	221093	098765
sundar@gmail.com	sundar	221104	098765

B) Perform DELETE Query with user table

The top screenshot shows the IBM Db2 on Cloud web interface. The left sidebar has a search bar and a list of objects including 'BVZ19901'. The main editor area shows a SQL script: `DELETE FROM USER WHERE email='sundar@gmail.com';`. Below the editor is a 'History' table with the following data:

Script	Date	Status	Runtime
Untitled - 1	Oct 31, 2022 7:56:44 PM	✓ 1	0.007 s
DELETE FROM USER WHERE email='sundar@gmail.com'		✓	0.007 s

The bottom screenshot shows the details of the 'BVZ19901.USER' table. It includes a 'Back' button and an 'Export to CSV' button. The table data is as follows:

EMAIL	USERNAME	ROLLNUMBER	PASSWORD
sanjay@gmail.com	sanjay	221085	098765
santy@gmail.com	santhosh	221085	123456
sivu@gmail.com	sivaram	221093	098765

3) Connect python code to db2.

### Connect.py

```
import ibm_db

def get_db_connection():
```

```

try:
    conn = ibm_db.connect("DATABASE=BLUDB;\
        HOSTNAME=<hostname>;\
        PORT=30376;\
        Security=SSL;\
        SSLServerCertificate=DigiCertGlobalRootCA.crt;\
        UID=<username>;\
        PWD=<password>;", "", "")
    print("Connected to DB")
    return conn
except:
    print("error while connecting ", ibm_db.conn_errormsg())
    return 0

```

- 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

### App.py

```

from flask import Flask, render_template, request, session, redirect, url_for
import ibm_db
from connect import get_db_connection

app = Flask(__name__)
app.secret_key = 'abcde'
con = 0

@app.route('/')
@app.route("/register", methods=['GET', 'POST'])
def register():
    if(request.method == 'GET'):
        if 'auth' in session:
            if session['auth']:
                return redirect(url_for('home'))
        return render_template("register_2.html")
    elif(request.method == 'POST'):
        name = request.form['username']
        email = request.form['email']
        password = request.form['password']
        rollno = request.form['rollno']
        # TODO: check if account already exist
        qry = "INSERT INTO USER VALUES(?,?,?,?)"
        stmt = ibm_db.prepare(con, qry)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, name)

```

```

        ibm_db.bind_param(stmt,3,rollno)
        ibm_db.bind_param(stmt,4,password)

        ibm_db.execute(stmt)
        return redirect(url_for('login'))
    else:
        return "404 not found"

@app.route('/login', methods=['GET', 'POST'])
def login():
    if(request.method == 'GET'):
        if 'auth' in session:
            if session['auth']:
                return redirect(url_for('home'))
            return render_template("login.html")
        elif(request.method == 'POST'):
            email=request.form['email']
            password=request.form['password']
            sql="SELECT * FROM USER WHERE email=? AND password=?"
            stmt=ibm_db.prepare(con,sql)
            ibm_db.bind_param(stmt,1,email)
            ibm_db.bind_param(stmt,2,password)
            ibm_db.execute(stmt)
            resp=ibm_db.fetch_assoc(stmt)
            print("resp - ",resp)
            if resp:
                session['auth'] = True
                session['email'] = email
                session['name'] = resp['USERNAME']
                # session['rollno'] = rollno

                return redirect(url_for('home'))
            else:
                return redirect(url_for('login'))
        else:
            return "404 not found"

@app.route('/home')
def home():
    if 'auth' in session:
        if not session['auth']:
            return redirect(url_for('login'))
    return render_template("home.html")

@app.route('/logout')
def logout():
    session.pop('email',None)
    session.pop('name',None)

```

```

    session['auth'] = False
    return redirect(url_for('login'))

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

```

## home.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link rel="stylesheet" href="{{url_for('static', filename =
'style2.css')}}">
</head>
<body>
    <nav class="navbar">
        <h1 class="logo">MyMoney</h1>
        <ul class="nav-ul">
            <li class="nav-li"><a href="/logout">Logout</a></li>
        </ul>
    </nav>
    <div class="container">
        <h1 class="head">Welcome {{session['name']}} !!!</h1>
    </div>
</body>
</html>

```

## Register.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{url_for('static', filename =
'style2.css')}}">
    <title>Register</title>
</head>
<body>
    <nav class="navbar">
        <h1 class="logo">MyMoney</h1>

```

```

        <ul class="nav-ul">
            <li class="nav-li"><a href="/login">Login</a></li>
        </ul>
    </nav>
    <div class="container">
        <h1 class="head">Register for a New Account</h1>
        <form action="/register" method="post" class="register">
            <label for="name">UserName</label>
            <input type="text" name="username" class="inp">

            <label for="name">Rollno</label>
            <input type="text" name="rollno" class="inp">

            <label for="email">Email</label>
            <input type="email" name="email" class="inp">

            <label for="password">Password</label>
            <input type="password" name="password" class="inp">

            <input type="submit" value="Register" class="inp-btn">

        </form>
    </div>
</body>
</html>

```

## Login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{url_for('static', filename =
'style2.css')}">
    <title>Register</title>
</head>
<body>
    <nav class="navbar">
        <h1 class="logo">MyMoney</h1>
        <ul class="nav-ul">
            <li class="nav-li"><a href="/register">Register</a></li>
        </ul>
    </nav>
    <div class="container">
        <h1 class="head">Login to your Account</h1>
        <form action="/login" method="post" class="register">

```



```

        <label for="email">Email</label>
        <input type="email" name="email" class="inp">

        <label for="password">Password</label>
        <input type="password" name="password" class="inp">

        <input type="submit" value="Login" class="inp-btn">

    </form>
</div>
</body>
</html>

```

## Style2.css

```

@import
url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap');
@import url('http://fonts.cdnfonts.com/css/lemon-tuesday');
*{
    margin: 0%;
    padding: 0%;
    box-sizing: border-box;
    font-family: 'Montserrat', sans-serif;
}

.navbar{
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
    /* background-color: #6c73ee; */
    background-color: #7C4CFB;
    color: rgb(234, 237, 240);
}

.nav-ul{
    display: flex;
    flex-direction: row;
    list-style: none;
}

.nav-li{
    padding: 1rem;
    font-weight: bold;
    font-size: 20px;
}

```

```
.nav-li:hover{
  /* background-color: #545CB3; */
  background-color: #9900FF;
  cursor: pointer;
}

.logo{
  font-family: 'Lemon Tuesday', sans-serif;
  font-size: 2rem;
  padding-left: 1rem;
  padding-bottom: 0.6rem;
}

.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}

.center{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: 92vh;
}

.left{
  display: flex;
  flex-direction: column;
  align-items: flex-end;
  height: 92vh;
  overflow: hidden;
}

.right{
  display: flex;
  flex-direction: column;
  align-items: flex-end;
  height: 92vh;
  overflow: hidden;
}

.img-div-right{
  height: 100vh;
  margin-top: 13.5rem;
}

.img-div-left{
  height: 100vh;
```

```
        margin-top: 15.5rem;
    }
    .register{
        display: flex;
        flex-direction: column;
    }
    label{
        margin-top: 20px;
        font-size: 20px;
        font-weight: 600;
    }

    .inp{
        width: 20vw;
        height: 5vh;
        border: 2px solid #120E43;
        border-radius: 6px;
    }

    .inp-btn{
        margin-top: 15px;
        width: 20vw;
        height: 5vh;
        background-color: #6c73ee;
        color: #ffffff;
        font-size: 20px;
        font-weight: 600;
        border-radius: 6px;
    }
    .inp-btn:hover{
        background-color: #545CB3;
        cursor: grab;
    }

    .head{
        color: black;
    }
    .company{
        color: #ffffff;
        margin-bottom: 15vh;
    }

    .main-img-left{
        transform: scale(0.90, 0.90);
    }

    .main-img-right{
        transform: scale(0.80, 0.80);
    }
```



Course Introductio...My OSCP NotesChegg expert loginOSCPWeb Dev toolsOFF-CPlacement ResourcesFinanceCPPlacement planLCO ProCTSStrivers A2Z DSA C...

MyMoneyLogin

### Register for a New Account

UserName

sanjay

Rollno

221085

Email

sanjay@gmail.com

Password

\*\*\*\*\*

Register

Course Introductio...My OSCP NotesChegg expert loginOSCPWeb Dev toolsOFF-CPlacement ResourcesFinanceCPPlacement planLCO ProCTSStrivers A2Z DSA C...

MyMoneyRegister

### Login to your Account

Email

sanjay@gmail.com

Password

\*\*\*\*\*

Login

