|  |  |
| --- | --- |
| Assignment Date | 19 September |
| Student Roll No | 1901261 |
| Student Name | Visveswar Srinivasan S |
| Marks | 2 |

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

CREATE TABLE [dbo].[Users] (  
   [UserId] [int] NOT NULL,  
   [UserName] [varchar] (15) NOT NULL ,  
   [Pwd] [varchar] (25) NOT NULL ,  
   [userRole] [varchar] (25) NOT NULL ,  
) ON [PRIMARY]

2.. Perform UPDATE,DELETE Queries with user table

1. **CREATE** **TABLE** Employee(
2. EmpId **int**,
3. LastName **varchar**(255),
4. FirstName **varchar**(255),
5. Address **varchar**(255),
6. City **varchar**(255)
7. );

1. **INSERT** **INTO** Employee    (EmpId,LastName,FirstName,ADDRESS,City)
2. **VALUES** (1, 'XYZ', 'ABC', 'India', 'Mumbai' );
3. **INSERT** **INTO** Employee (EmpId,LastName,FirstName,ADDRESS,City)
4. VALUES (2, 'X', 'A', 'India', 'Pune' );
5. **UPDATE** Employee
6. **SET** City='Pune'

**DELETE** **FROM** Employee **WHERE** EmpId=1;

1. Connect python code to db2.

import jaydebeapi

conn\_src = jaydebeapi.connect(

'com.ibm.db2.jcc.DB2Driver',

['YourHostName:PortNo/DatabaseName','userid','password'],'C:/db2jcc4.jar'

)

cursor=conn\_src.cursor()

sql = 'Select \* from schemaname.TableName fetch first 100 rows only '

cursor.execute(sql)

print("fetchall:")

result = cursor.fetchall()

for r in result:

print(r)

1. 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

# Store this code in 'app.py' file

from flask import Flask, render\_template, request, redirect, url\_for, session

from flask\_mysqldb import MySQL

import MySQLdb.cursors

import re

app = Flask(\_\_name\_\_)

app.secret\_key = 'your secret key'

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'your password'

app.config['MYSQL\_DB'] = 'geeklogin'

mysql = MySQL(app)

@app.route('/')

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

def login():

msg = ''

if request.method == 'POST' and 'username' in request.form and 'password' in request.form:

username = request.form['username']

password = request.form['password']

cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

cursor.execute('SELECT \* FROM accounts WHERE username = % s AND password = % s', (username, password, ))

account = cursor.fetchone()

if account:

session['loggedin'] = True

session['id'] = account['id']

session['username'] = account['username']

msg = 'Logged in successfully !'

return render\_template('index.html', msg = msg)

else:

msg = 'Incorrect username / password !'

return render\_template('login.html', msg = msg)

@app.route('/logout')

def logout():

session.pop('loggedin', None)

session.pop('id', None)

session.pop('username', None)

return redirect(url\_for('login'))

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

def register():

msg = ''

if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :

username = request.form['username']

password = request.form['password']

email = request.form['email']

cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

cursor.execute('SELECT \* FROM accounts WHERE username = % s', (username, ))

account = cursor.fetchone()

if account:

msg = 'Account already exists !'

elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

msg = 'Invalid email address !'

elif not re.match(r'[A-Za-z0-9]+', username):

msg = 'Username must contain only characters and numbers !'

elif not username or not password or not email:

msg = 'Please fill out the form !'

else:

cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username, password, email, ))

mysql.connection.commit()

msg = 'You have successfully registered !'

elif request.method == 'POST':

msg = 'Please fill out the form !'

return render\_template('register.html', msg = msg)