

News Tracker Application

Assignment-2:

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

insert into User values('sherin12@gmail.com','Sherin',03,'bfhjkk');

insert into User values('jeniba@gmail.com','Jeniba',05,'ssfwef');

insert into User values('nish@gmail.com','Nishvitha',23,'dsdewk');

insert into User values('melisha@gmail.com','Melisha',16,'QXDGHC');

insert into User values('swetha@gmail.com','sweha',20,'mjhdh');

The screenshot displays the IBM Db2 on Cloud web interface. The left sidebar shows the 'Data objects' tab with a search filter 'TNK83244'. The main area shows a SQL script in a text editor with the following content:

```
1 insert into User values('sherin12@gmail.com','Sherin',03,'bfhjkk');
2 insert into User values('jeniba@gmail.com','Jeniba',05,'ssfwef');
3 insert into User values('nish@gmail.com','Nishvitha',23,'dsdewk');
4 insert into User values('melisha@gmail.com','Melisha',16,'QXDGHC');
5 UPDATE User
6 SET email = 'jebasherinsj@gmail.com', rollnumber = 111903
7 WHERE username = 'Sherin';
8 insert into User values('swetha@gmail.com','sweha',20,'mjhdh');
9 DELETE FROM User WHERE username='sweha';
```

Below the script, the 'History' tab shows a table of executed scripts:

| Script | Date | Status | Runtime |
|--|------------------------|--------|---------|
| Untitled - 1 | Nov 6, 2022 2:29:21 AM | 7 | 0.029 s |
| insert into User values('sherin12@gmail.com','Sherin',03,'bfh... | | ✓ | 0.004 s |
| insert into User values('jeniba@gmail.com','Jeniba',05,'ssfw... | | ✓ | 0.004 s |
| insert into User values('nish@gmail.com','Nishvitha',23,'dsde... | | ✓ | 0.004 s |

The bottom status bar shows the system time as 01:32 on 06-11-2022.

2. Perform UPDATE, DELETE Queries with user table

```
insert into User values('sherin12@gmail.com','Sherin',03,'bfhjkk');
```

```
insert into User values('jeniba@gmail.com','Jeniba',05,'ssfwef');
```

```
insert into User values('nish@gmail.com','Nishvitha',23,'dsdewk');
```

```
insert into User values('melisha@gmail.com','Melisha',16,'QXDGHC');
```

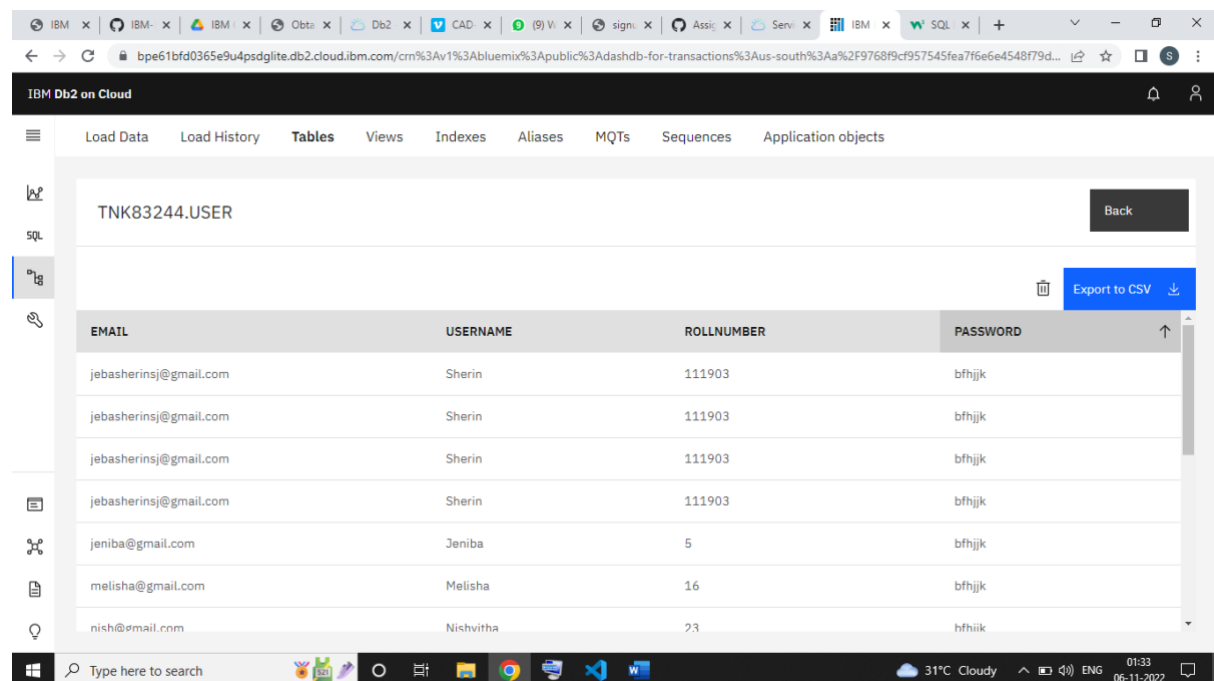
UPDATE User

```
SET email = 'jebasherinsj@gmail.com', rollnumber = 111903
```

```
WHERE username = 'Sherin';
```

```
insert into User values('swetha@gmail.com','sweha',20,'mjhdh');
```

```
DELETE FROM User WHERE username='sweha';
```



The screenshot shows the IBM Db2 on Cloud console interface. The 'Tables' tab is selected, displaying the 'TNK83244.USER' table. The table has four columns: EMAIL, USERNAME, ROLLNUMBER, and PASSWORD. The data is as follows:

| EMAIL | USERNAME | ROLLNUMBER | PASSWORD |
|------------------------|-----------|------------|----------|
| jebasherinsj@gmail.com | Sherin | 111903 | bfhjkk |
| jebasherinsj@gmail.com | Sherin | 111903 | bfhjkk |
| jebasherinsj@gmail.com | Sherin | 111903 | bfhjkk |
| jebasherinsj@gmail.com | Sherin | 111903 | bfhjkk |
| jeniba@gmail.com | Jeniba | 5 | bfhjkk |
| melisha@gmail.com | Melisha | 16 | bfhjkk |
| nish@gmail.com | Nishvitha | 23 | bfhiik |

The console also includes a search bar, navigation tabs (Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, Application objects), and a bottom status bar showing system information like temperature and time.

3. Connect python code to db2.

Code.py

```
from flask import Flask,render_template,request,redirect,url_for,session
import ibm_db
import re
app=Flask(__name__)
app.secret_key='a'
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURIT
Y=SSL;SSLSERVERCERTIFICATE=DigiCertGlobalRootCA.crt;UID=fqg93621;PWD=eezWMIvhvT
ceZxPH",'','')

@app.route("/")
def home():
    return render_template('login.html')

@app.route("/login",methods=['GET','POST'])
def login():
    global userid
    msg = ''

    if request.method=='POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM user WHERE username = ? AND password = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['username']
            userid= account['USERNAME']
            session['username']=account['USERNAME']
            msg='Logged in successfully!'
            return render_template('page.html',msg=msg)
        else:
            msg='Incorrect username/password'
            return render_template('login.html',msg=msg)

@app.route('/register',methods=['GET','POST'])
def register():
    if request.method=='POST':
        username=request.form['username']
        email=request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM user WHERE username = ?"
```

```

    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    print(account)
    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="Name must contain characters and numbers"
    else:
        insert_sql="INSERT INTO user VALUES(?,?,?)"
        prep_stmt= ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(prepare_stmt,1,username)
        ibm_db.bind_param(prepare_stmt,2,email)
        ibm_db.bind_param(prepare_stmt,3,password)
        ibm_db.execute(prepare_stmt)
        msg="You have successfully registered"
    elif request.method=="POST":
        msg="Please fill out the form"
        return render_template("registration.html",msg=msg)

@app.route('/page')
def page():
    return render_template("page.html")

@app.route('/apply',methods=['GET','POST'])
def app():
    msg=''
    if request.method=="POST":
        username=request.form['username']
        email=request.form['email']
        rollnumber=request.form['rollnumber']
        password=request.form['password']
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)

        if account():
            return render_template('page.html')

        insert_sql="INSERT INTO user VALUES(?,?,?,?)"
        prep_stmt=ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(prepare_stmt,1,username)
        ibm_db.bind_param(prepare_stmt,2,email)

```

```
ibm_db.bind_param(prepare_stmt,3,rollnumber)
ibm_db.bind_param(prepare_stmt,4,password)
ibm_db.execute(prepare_stmt)
```

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

Registration.html

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Registration Page </title>
<style>
Body {
    font-family: Calibri, Helvetica, sans-serif;
    background-image: url('C:/Users/Jeba/Desktop/img.jpg');
}
h1{
    font-family: "Cursive";
}
button {
    background-color: #4fbb80;
    width: 100%;
    color: rgb(14, 11, 7);
    padding: 15px;
    margin: 10px 0px;
    border: none;
    cursor: pointer;
}
form {
    border: 3px solid #f1f1f1;
}
input[type=text], input[type=text], input[type=email],input[type=text]{
    width: 100%;
    margin: 8px 0;
    padding: 12px 20px;
```

```

        display: inline-block;
        border: 2px solid green;
        box-sizing: border-box;
    }
    button:hover {
        opacity: 0.7;
    }
    .container {
        padding: 100px;
        background-color: rgba(252, 141, 141, 0.911);
        background-image: url('C:/Users/Jeba/Desktop/img.jpg');
    }
</style>
</head>
<body>
    <center> <h1><font color="yellow"><i> Registration Form </i></font></h1>
</center>
    <form method="post" action="/confirm">
        <div class="container">
            <label><font color="white">Username : </font></label>
            <input type="text" placeholder="Enter Username" name="username"
required>
            <label><font color="white">Email :</font> </label>
            <input type="email" placeholder="Enter Email Id" name="email"
required>
            <label><font color="white">Roll Number : </font></label>
            <input type="text" placeholder="Enter roll no." name="rollnumber"
required>
            <label><font color="white">Password : </font></label>
            <input type="text" placeholder="Enter password" name="password"
required>
            <button type="submit"><b><font
color="white">REGISTER</font></b></button>
        </div>
    </form>
</body>
</html>

```

Login.html

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Login Page </title>
<style>

```

```

Body {
  font-family: Calibri, Helvetica, sans-serif;
  background-image: url('C:/Users/Jeba/Desktop/img.jpg');
}
h1{
  font-family: "Cursive";
}
button {
  background-color: #4fbb80;
  width: 100%;
  color: rgb(14, 11, 7);
  padding: 15px;
  margin: 10px 0px;
  border: none;
  cursor: pointer;
}
form {
  border: 3px solid #f1f1f1;
}
input[type=text], input[type=password]{
  width: 100%;
  margin: 8px 0;
  padding: 12px 20px;
  display: inline-block;
  border: 2px solid green;
  box-sizing: border-box;
}
button:hover {
  opacity: 0.7;
}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  margin: 10px 5px;
}

.container {
  padding: 100px;
  background-color: rgba(252, 141, 141, 0.911);
  background-image: url('C:/Users/Jeba/Desktop/img.jpg');
}
</style>
</head>
<body>
  <center> <h1><font color="white"><i> Login Form </i></font></h1>
</center>
  <form method="post" action="page.html">

```

```

<div class="container">
  <label><font color="white">Username : </font></label>
  <input type="text" placeholder="Enter Username" name="username"
required>
  <label><font color="white">Password : </font></label>
  <input type="password" placeholder="Enter Password"
name="password" required>
  <button type="submit"><b><font color="white">LOG
IN</font></b></button>
  <input type="checkbox" checked="checked"> <b><font
color="white">Remember me</font></b>
  <button type="button" class="cancelbtn"><b>Cancel</b></button>
  <font color="white">Forgot </font><a href="#"> <font
color="white">password? </font></a>
</div>
</form>
</body>
</html>

```

Output:

The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Tables' tab is selected, and the table 'TNK83244.USER' is displayed. The table has four columns: EMAIL, USERNAME, ROLLNUMBER, and PASSWORD. There are 8 rows of data. A 'Back' button is in the top right, and an 'Export to CSV' button is in the top right of the table area. The bottom of the screen shows a Windows taskbar with various application icons and system information.

| EMAIL | USERNAME | ROLLNUMBER | PASSWORD |
|-------------------|-----------|------------|----------|
| nish@gmail.com | Nishvitha | 23 | dsdewk |
| melisha@gmail.com | Melisha | 16 | QXDGHG |
| melisha@gmail.com | Melisha | 16 | QXDGHG |
| melisha@gmail.com | Melisha | 16 | QXDGHG |
| jeniba@gmail.com | Jeniba | 5 | ssfwee |
| jeniba@gmail.com | Jeniba | 5 | ssfwee |
| jeniba@gmail.com | Jeniba | 5 | ssfwee |

IBM x IBM x IBM x IBM x Obts x Servi x IBM x CAD x (15) x Assi x SQL x Regi x + -

127.0.0.1:5000

Registration Form

Username :

Sherin

Email :

sherin345@gmail.com

Roll Number :

03

Password :

dghjk

REGISTER

DigiCertGlobalRoot...crt Show all x

Type here to search

Rain off and on ENG 08:22 06-11-2022

IBM x IBM x IBM x IBM x Obts x Servi x IBM x CAD x (13) x Assi x SQL x Regi x Logi x + -

127.0.0.1:5000

Login Form

Username :

Sherin

Password :

LOG IN

☒ Remember me Cancel Forgot password?

DigiCertGlobalRoot...crt Show all x

Type here to search

Rain off and on 08:26 Speaker/Headphone: 100%