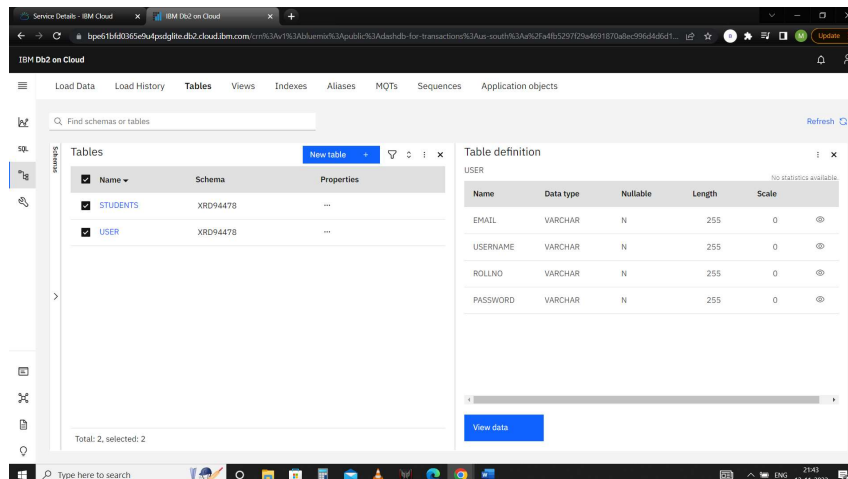
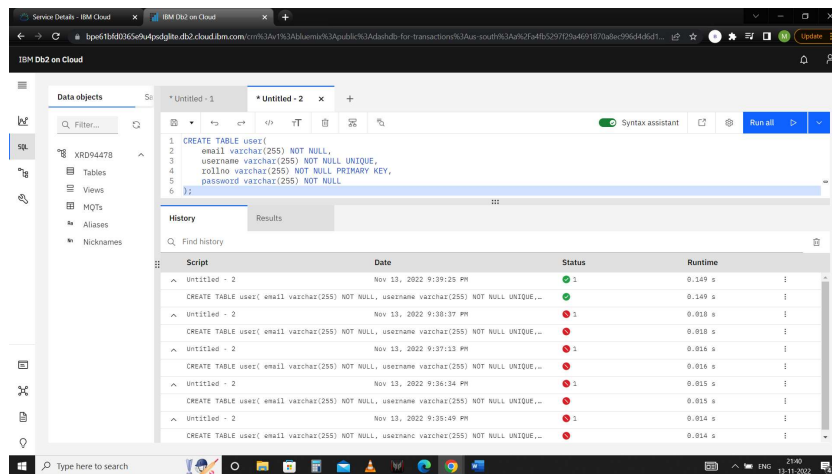


## ASSIGNMENT - 2

### FLASK APPLICATION INTEGRATION WITH DB2

Student Name	Gowtham
Student Register Number	311019106022
Maximum Marks	2 Marks

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



## 2. Perform UPDATE, and DELETE queries with the User table.

### INSERT QUERY

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tree with 'XRD94478' selected. The main editor area shows a SQL script with the following content:

```
3  username varchar(255) NOT NULL UNIQUE,  
4  rollno varchar(255) NOT NULL PRIMARY KEY,  
5  password varchar(255) NOT NULL  
6  );  
7  
8  INSERT INTO user VALUES('19ec014@kcgcollege.com', 'balaji', '311019106014', 'thunder');  
9  SELECT * FROM user;  
10  
11 INSERT INTO user VALUES('19ec020@kcgcollege.com', 'DKG', '311019106010', 'rain');  
12 SELECT * FROM user;  
13
```

The 'History' tab is active, showing a table with columns: Script, Date, Status, and Runtime. The table is empty, displaying 'No history' and the message 'Your history will appear here'.

The screenshot shows the IBM Db2 on Cloud SQL editor interface with the 'Tables' tab selected. The table 'XRD94478.USER' is displayed. The table has four columns: EMAIL, USERNAME, ROLLNO, and PASSWORD. The table contains two rows of data.

EMAIL	USERNAME	ROLLNO	PASSWORD
19ec014@kcgcollege.com	balaji	311019106014	thunder
19ec020@kcgcollege.com	DKG	311019106010	rain

The interface also includes a 'Back' button and an 'Export to CSV' button.

# UPDATE QUERY

The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tree with 'XRD94478' selected. The main editor area shows a script with the following SQL statements:

```
9 SELECT * FROM userz;  
10  
11 INSERT INTO userz VALUES('19ec020@kcgcollege.com', 'DKG', '311019106010', 'rain');  
12 SELECT * FROM userz;  
13  
14 UPDATE userz SET PASSWORD= 'DON' WHERE USERNAME='DKG';  
15 SELECT * FROM userz;  
16  
17  
18
```

The 'History' tab is active, showing a table of executed scripts:

Script	Date	Status	Runtime
Untitled - 2	Nov 13, 2022 10:07:06 PM	1	0.005 s
SELECT * FROM userz			0.005 s
Untitled - 2	Nov 13, 2022 10:06:39 PM	1	0.022 s
UPDATE userz SET PASSWORD= 'DON' WHERE USERNAME='DKG'			0.007 s
SELECT * FROM userz			0.015 s
Untitled - 2	Nov 13, 2022 10:04:58 PM	2	0.034 s
UPDATE userz SET PASSWORD= 'book' WHERE USERNAME='DKG'			0.016 s

The screenshot shows the IBM Db2 on Cloud interface displaying the details of the 'XRD94478.USER' table. The table structure is as follows:

EMAIL	USERNAME	ROLLNO	PASSWORD
19ec014@kcgcollege.com	balaji	311019106014	thunder
19ec020@kcgcollege.com	DKG	311019106010	DON

## DELETE QUERY

The screenshot shows the IBM Db2 on Cloud console. The SQL editor contains the following script:

```
12 SELECT * FROM user1,  
13  
14 UPDATE user SET PASSWORD= 'DON' WHERE USERNAME='DKG';  
15 SELECT * FROM user;  
16  
17 DELETE FROM user WHERE rollno = '311019106010';  
18 SELECT * FROM user;  
19  
20  
21  
22
```

The History tab shows the following table:

Script	Date	Status	Runtime
Untitled - 2	Nov 13, 2022 10:16:37 PM	2	0.005 s
DELETE FROM user WHERE rollno = '311019106010';			0.004 s
SELECT * FROM user;			0.001 s
Untitled - 2	Nov 13, 2022 10:16:09 PM	1	0.012 s
DELETE FROM user WHERE EMAIL = '19ec020@kcgcollege.com';			0.007 s
SELECT * FROM user;			0.005 s
Untitled - 2	Nov 13, 2022 10:13:14 PM	1	0.013 s

The screenshot shows the IBM Db2 on Cloud console displaying the details of the XRD94478.USER table. The table has the following columns: EMAIL, USERNAME, ROLLNO, and PASSWORD. The data is as follows:

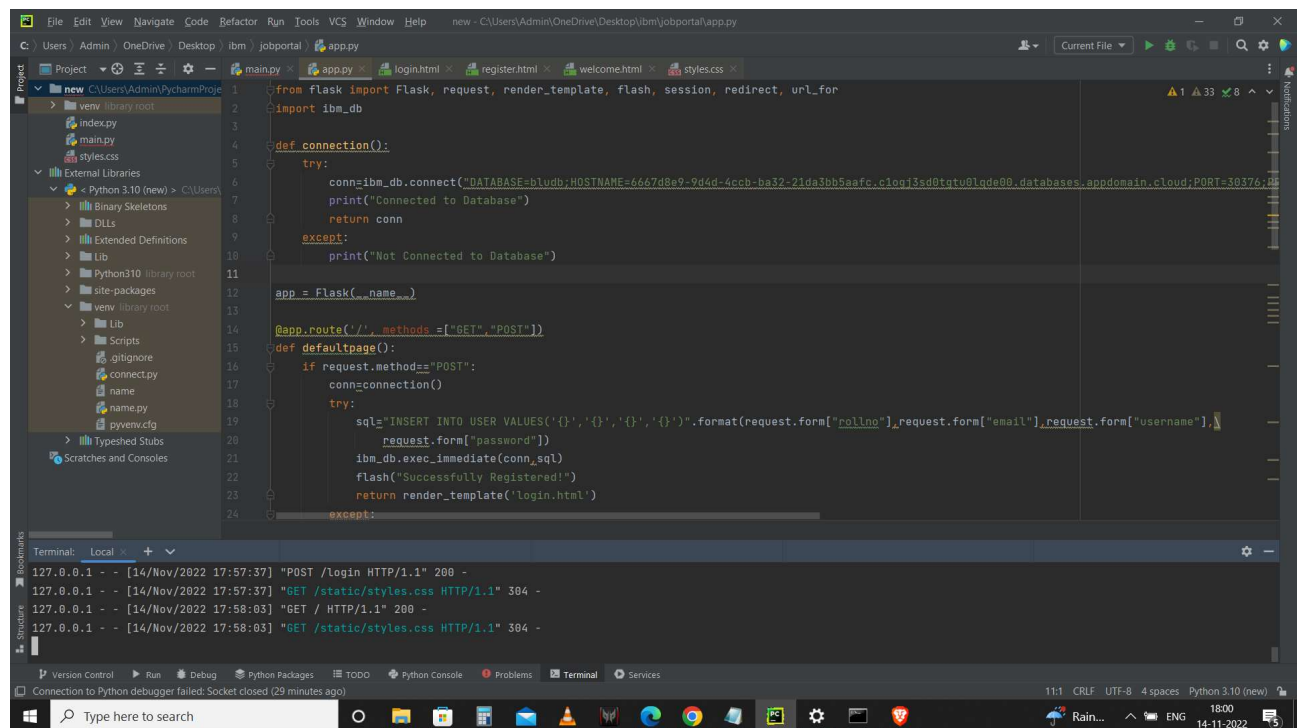
EMAIL	USERNAME	ROLLNO	PASSWORD
19ec020@kcgcollege.com	DKG	311019106010	DON

### 3. python code to DB2 &

4. Create a flask app with a registration page, login page and welcome page. By default, load the registration page once the user enters all the fields store the data in the database and navigate to the login page to authenticate the user with a username and password. If the user is valid show the welcome page.

## CODE

### app.py



```
1 from flask import Flask, request, render_template, flash, session, redirect, url_for
2 import ibm_db
3
4 def connection():
5     try:
6         conn=ibm_db.connect("DATABASE=ibmdb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1og13sd0tqtu0lqde00.databases.appdomain.cloud;PORT=30376;PL
7         print("Connected to Database")
8         return conn
9     except:
10        print("Not Connected to Database")
11
12 app = Flask(__name__)
13
14 @app.route('/', methods = ["GET", "POST"])
15 def defaultpage():
16     if request.method=="POST":
17         conn=connection()
18         try:
19             sql="INSERT INTO USER VALUES('{},{},{},{})'.format(request.form["rollno"],request.form["email"],request.form["username"],
20                 request.form["password"])
21             ibm_db.exec_immediate(conn,sql)
22             flash("Successfully Registered!")
23             return render_template('login.html')
24         except:
```

Terminal: Local x + v

```
127.0.0.1 - - [14/Nov/2022 17:57:37] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:57:37] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET /static/styles.css HTTP/1.1" 304 -
```

Version Control Run Debug Python Packages Python Console Problems Terminal Services

Connection to Python debugger failed: Socket closed (29 minutes ago)

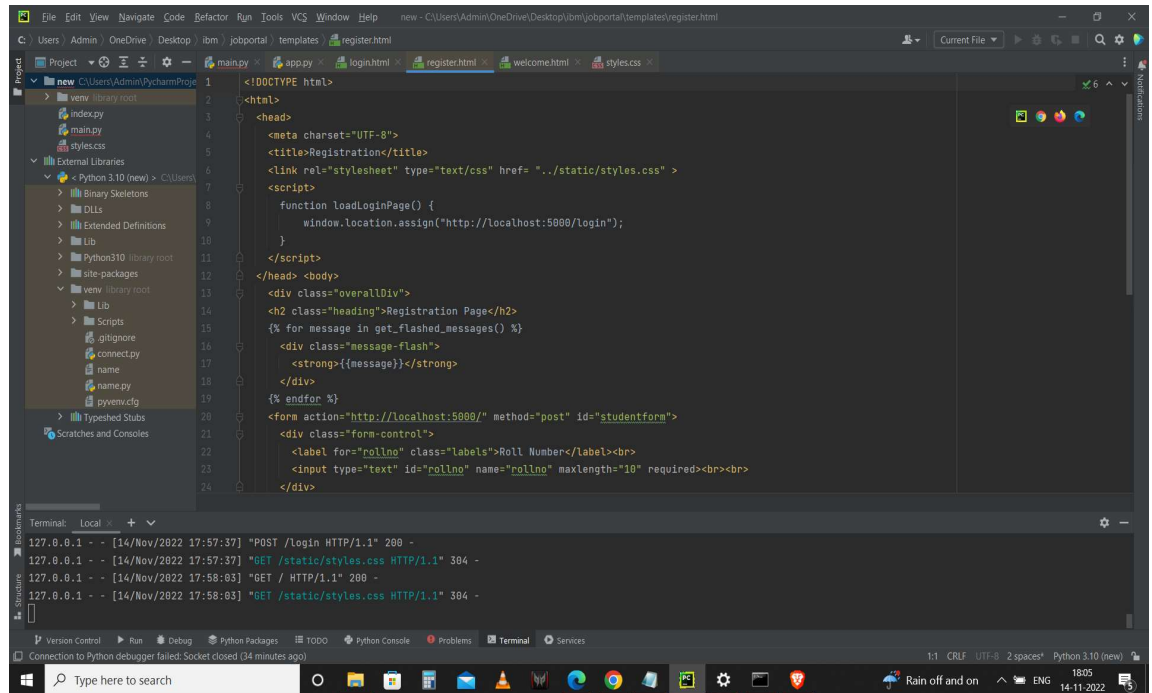
11:1 CRLF UTF-8 4 spaces Python 3.10 (new)

Type here to search

Rain... 1800 14-11-2022

# Templates

## Register.html

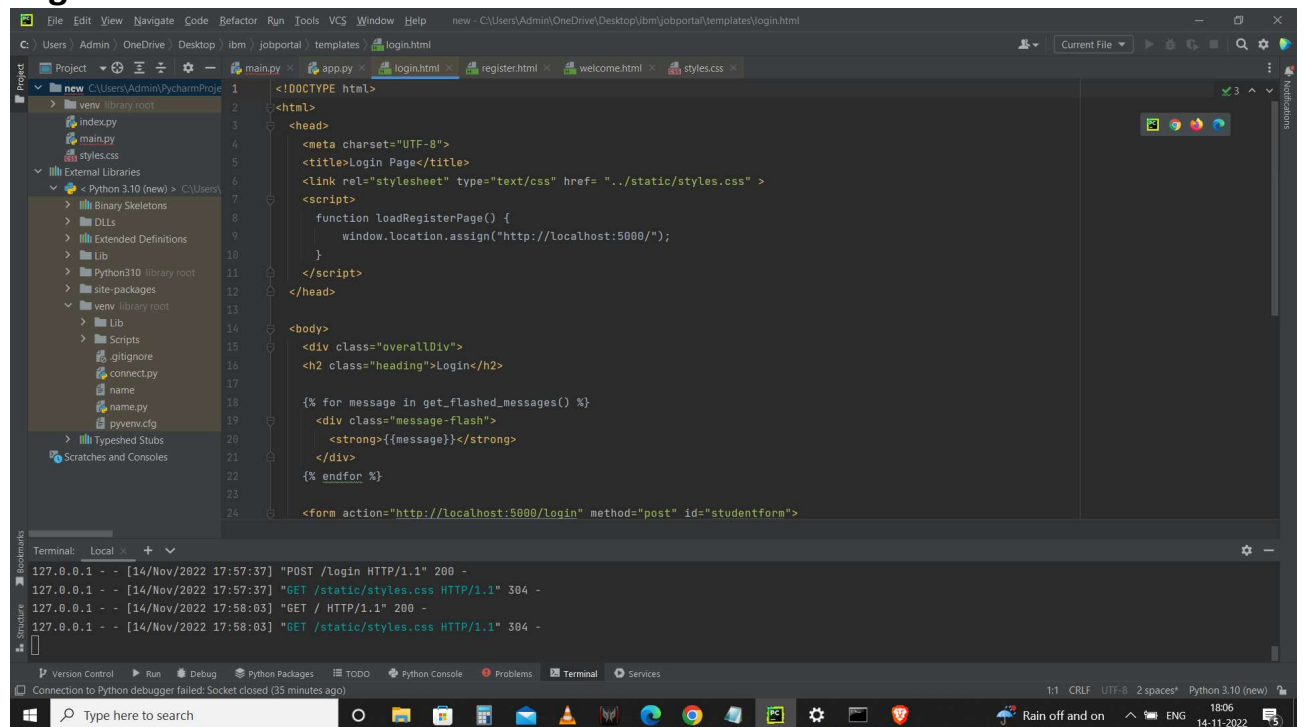


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Registration</title>
6 <link rel="stylesheet" type="text/css" href=" ../static/styles.css" >
7 <script>
8     function loadLoginPage() {
9         window.location.assign("http://localhost:5000/login");
10    }
11 </script>
12 </head> <body>
13 <div class="overallDiv">
14 <h2 class="heading">Registration Page</h2>
15 {% for message in get_flashed_messages() %}
16 <div class="message-flash">
17 <strong>{{message}}</strong>
18 </div>
19 {% endfor %}
20 <form action="http://localhost:5000/" method="post" id="studentform">
21 <div class="form-control">
22 <label for="rollno" class="labels">Roll Number</label><br>
23 <input type="text" id="rollno" name="rollno" maxlength="10" required><br><br>
24 </div>
```

Terminal: Local x +

```
127.0.0.1 - - [14/Nov/2022 17:57:37] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:57:37] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET /static/styles.css HTTP/1.1" 304 -
```

## Login.html

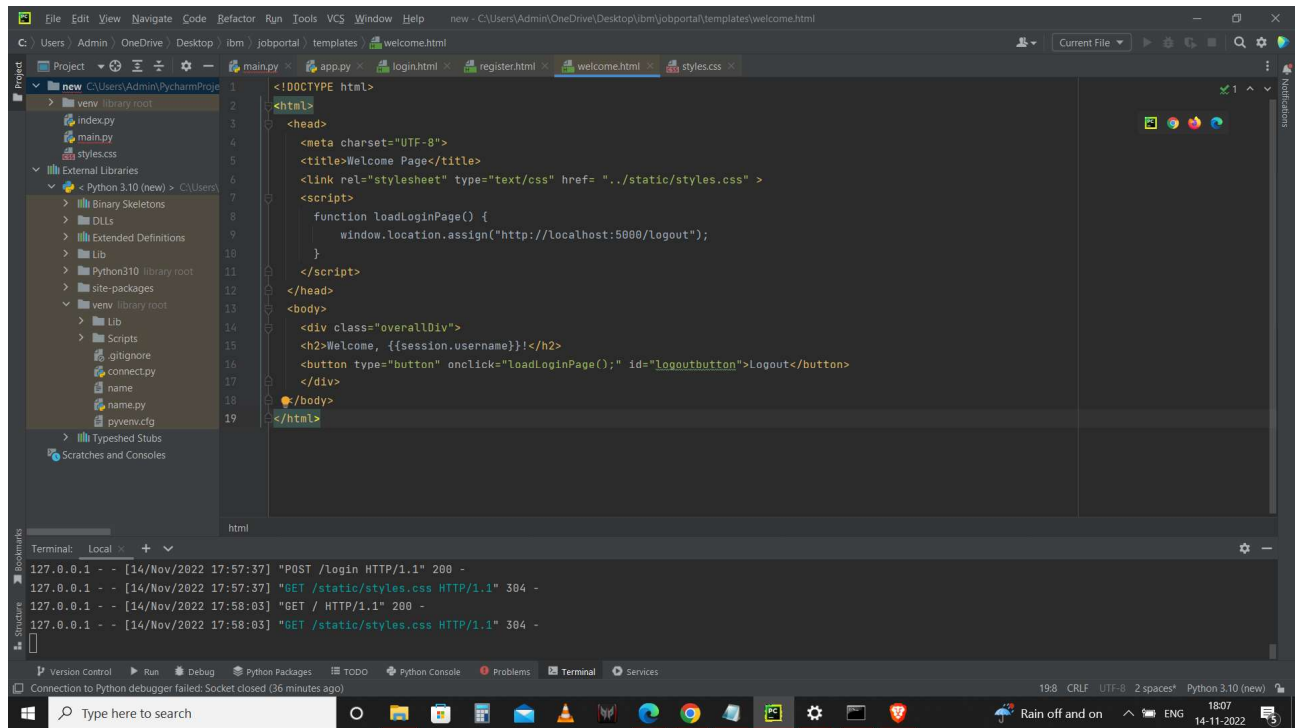


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Login Page</title>
6 <link rel="stylesheet" type="text/css" href=" ../static/styles.css" >
7 <script>
8     function loadRegisterPage() {
9         window.location.assign("http://localhost:5000/");
10    }
11 </script>
12 </head>
13 <body>
14 <div class="overallDiv">
15 <h2 class="heading">Login</h2>
16
17 {% for message in get_flashed_messages() %}
18 <div class="message-flash">
19 <strong>{{message}}</strong>
20 </div>
21 {% endfor %}
22
23 <form action="http://localhost:5000/login" method="post" id="studentform">
```

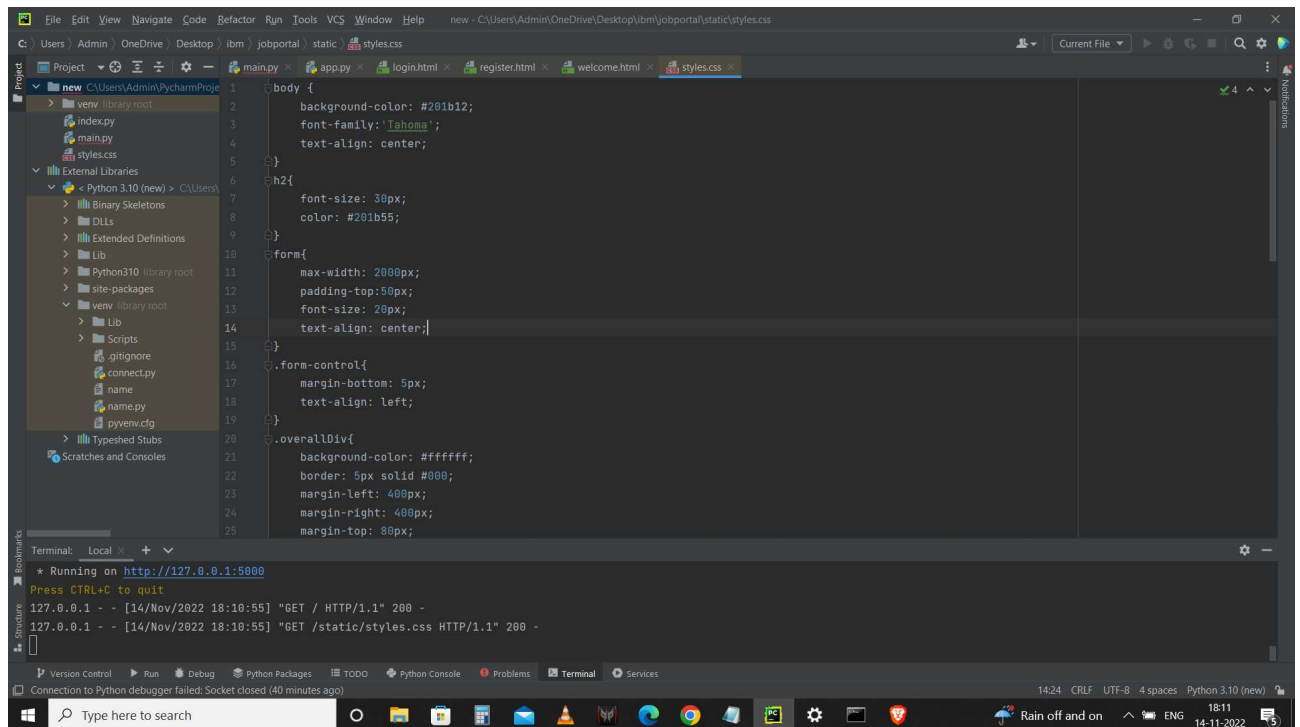
Terminal: Local x +

```
127.0.0.1 - - [14/Nov/2022 17:57:37] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:57:37] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 17:58:03] "GET /static/styles.css HTTP/1.1" 304 -
```

# Welcome.html

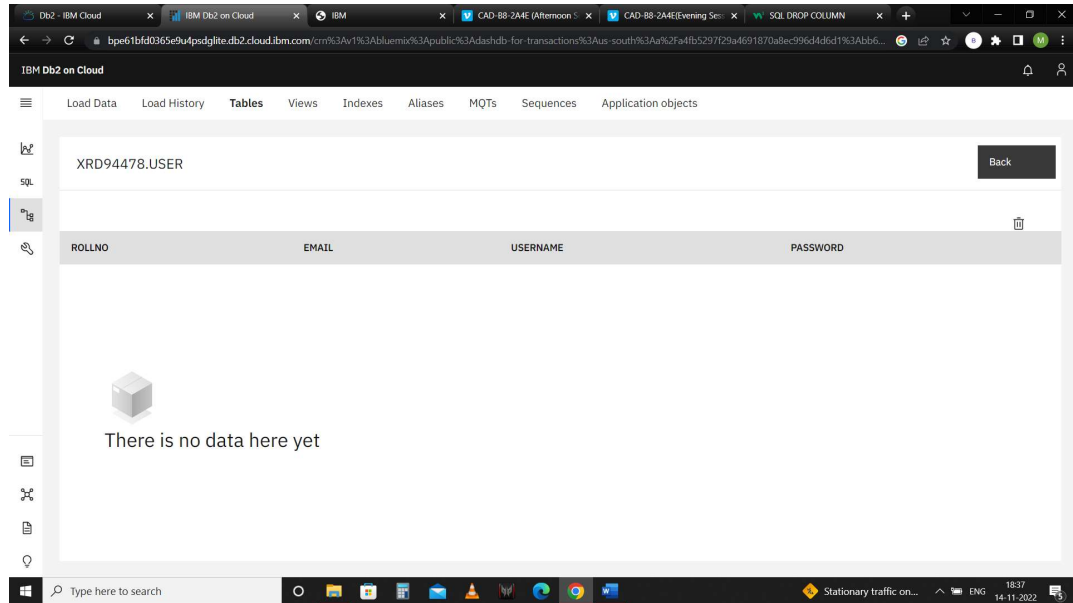


# Static.css

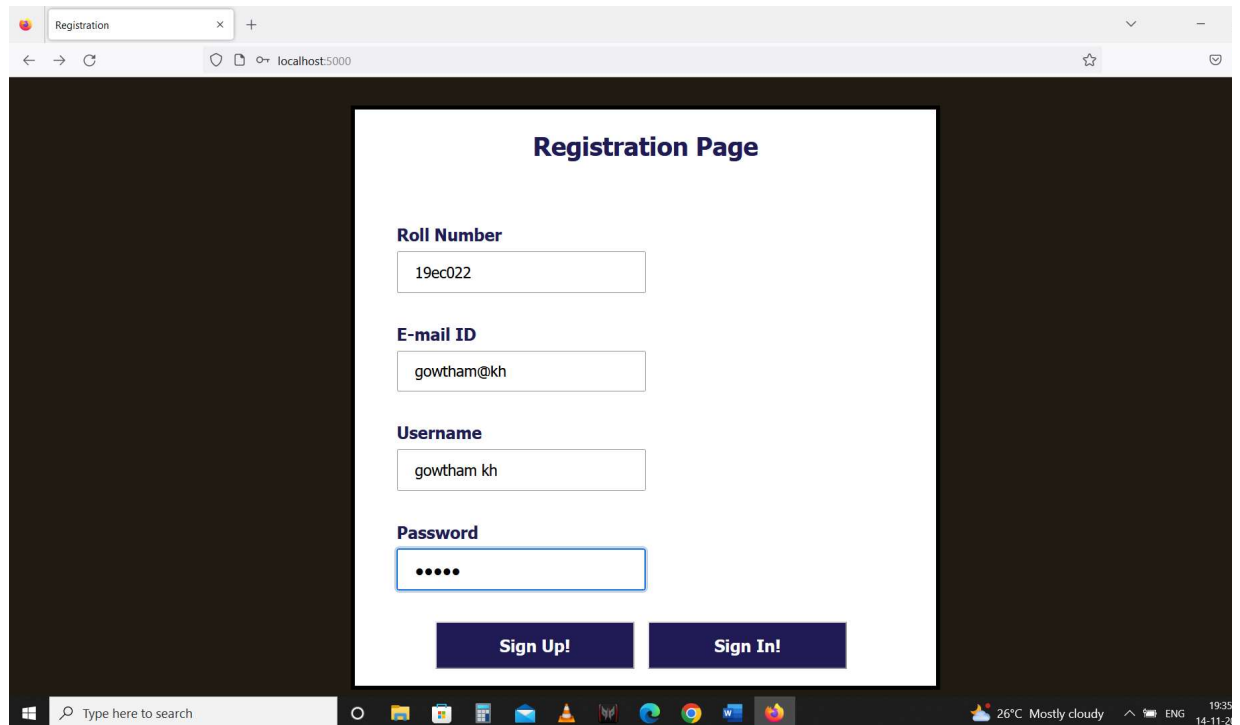


# OUTPUT

## Empty table



## Registration page





## Database updated upon registration

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, displaying a table named 'XRD94478.USER'. The table has four columns: 'EMAIL', 'USERNAME', 'ROLLNO', and 'PASSWORD'. A single row of data is visible, representing a user registration. A 'Back' button is located in the top right corner of the table view. Below the table, there is an 'Export to CSV' button. The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray displaying the date and time.

EMAIL	USERNAME	ROLLNO	PASSWORD
19ec022	gowtham@kh	gowtham kh	zxcvb

## Welcome Page Rendered on Successful Login

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/login'. The page content is a dark-themed welcome screen. In the center, there is a white rectangular box containing the text 'Welcome, gowtham@kh!' in a bold, dark font. Below this text is a dark blue button with the word 'Logout' in white. The browser's address bar also shows a 'Welcome Page' tab. The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray displaying the date and time.

## Terminal

The screenshot displays a Windows 10 desktop environment. The primary focus is the Visual Studio Code (VS Code) editor, which is open to a Python file named `app.py`. The file's content is as follows:

```

1  # Import Flask module
2  from flask import Flask, session, redirect, url_for
3
4  # Create an instance of Flask class
5  app = Flask(__name__)
6
7  # Create a route for the index page
8  @app.route('/')
9  def index():
10     return "Hello, World!"
11
12 # Create a route for the login page
13 @app.route('/login')
14 def login():
15     session.pop('logged_in', None)
16     session.pop('username', None)
17     return redirect(url_for('login'))
18
19 # Create a route for the logout page
20 @app.route('/logout')
21 def logout():
22     session.pop('logged_in', None)
23     session.pop('username', None)
24     return redirect(url_for('login'))
25
26 # Run the application
27 if __name__ == '__main__':
28     app.config['SECRET_KEY'] = 'super secret key'
29     app.config['SESSION_TYPE'] = 'filesystem'
30     app.run()
31
32 if __name__ == '__main__':

```

The VS Code interface includes a sidebar on the left showing the project structure, which includes a `new` folder containing `index.py`, `main.py`, and `styles.css`. The bottom of the VS Code window features a terminal window with the following output:

```

127.0.0.1 - - [14/Nov/2022 18:47:07] "GET /static/styles.css HTTP/1.1" 304 -
Connected to Database
127.0.0.1 - - [14/Nov/2022 18:47:19] "POST /login HTTP/1.1" 200 -
127.0.0.1 - - [14/Nov/2022 18:47:19] "GET /static/styles.css HTTP/1.1" 304 -
PS C:\Users\Admin\OneDrive\Desktop\ibm\jobportal> python app.py
* Serving Flask app 'app'
* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit

```

The Windows taskbar at the bottom shows the system clock as 6:14 PM on 14-11-2022, with the language set to ENG. The taskbar also includes icons for various applications, including the Start menu, search, and several background processes.