

ASSIGNMENT-2

AM.MARIYAAKILA

VV COLLEGE OF ENGINEERING

III-YEAR CSE

In the last Assignment you have created index.html and profile.html. So, now in this assignment you have create two html pages and style it using css.

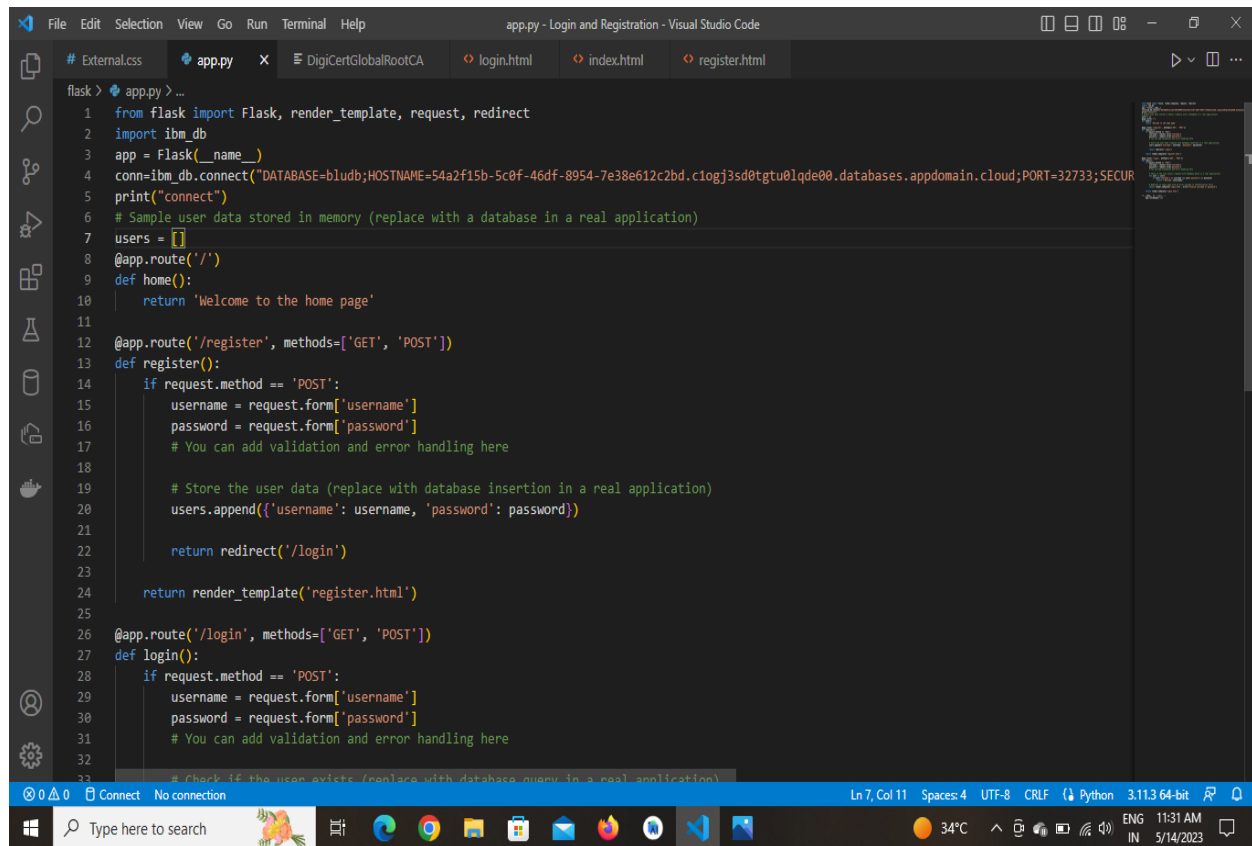
1. Login and Register html page

Task 1: You need to run the html pages in server by using Flask web framework

Task 2: Store the user registration details in IBM DB2

Task 3: while user logging in do the login validation (checking the login credentials is correct or not)

App.py:

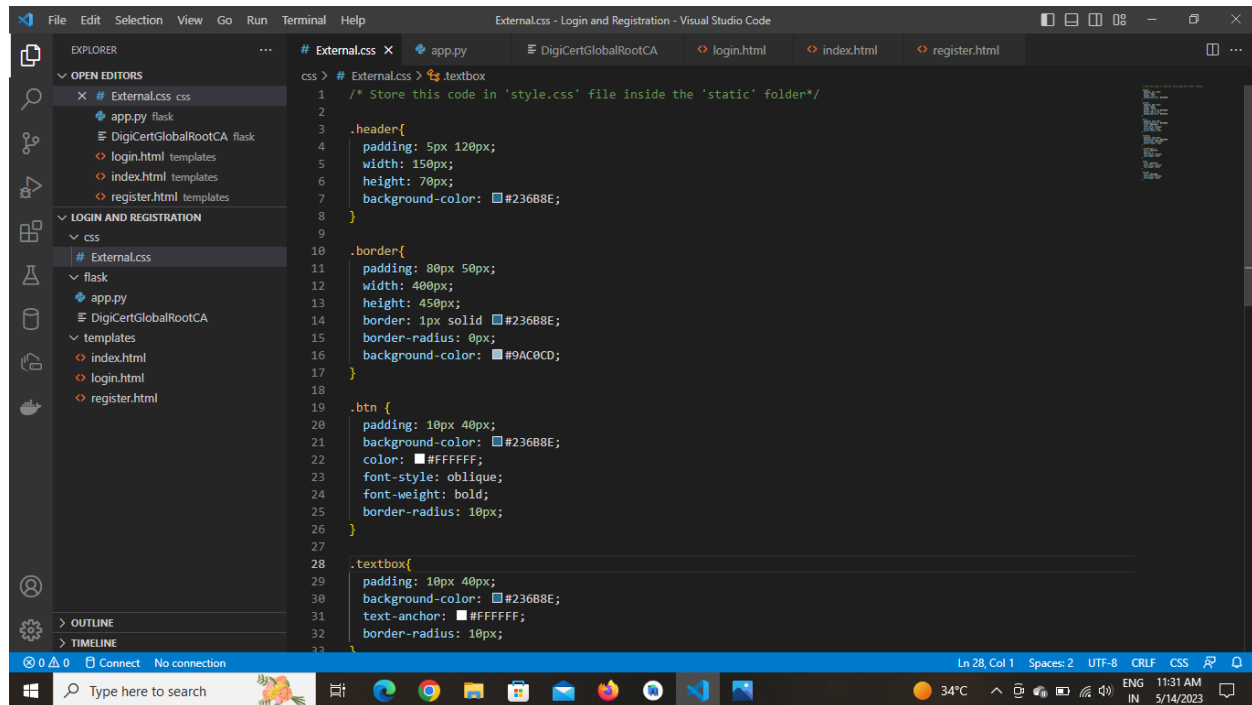


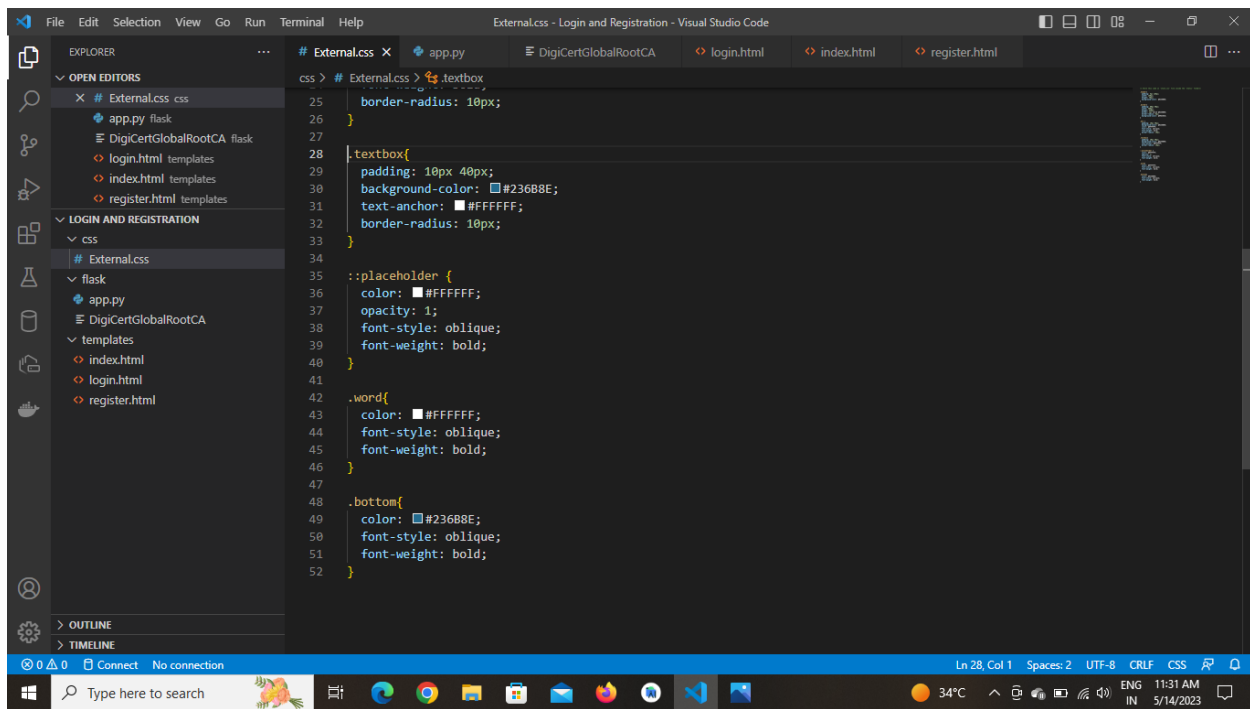
```
1 from flask import Flask, render_template, request, redirect
2 import ibm_db
3 app = Flask(__name__)
4 conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECUR
5 print("connect")
6 # Sample user data stored in memory (replace with a database in a real application)
7 users = []
8 @app.route('/')
9 def home():
10     return 'Welcome to the home page'
11
12 @app.route('/register', methods=['GET', 'POST'])
13 def register():
14     if request.method == 'POST':
15         username = request.form['username']
16         password = request.form['password']
17         # You can add validation and error handling here
18
19         # Store the user data (replace with database insertion in a real application)
20         users.append({'username': username, 'password': password})
21
22         return redirect('/login')
23
24     return render_template('register.html')
25
26 @app.route('/login', methods=['GET', 'POST'])
27 def login():
28     if request.method == 'POST':
29         username = request.form['username']
30         password = request.form['password']
31         # You can add validation and error handling here
32
33         # Check if the user exists (replace with database query in a real application)
```

The screenshot shows the Visual Studio Code editor with a Python file named `app.py` open. The code is a Flask application for login and registration. The editor has a dark theme, and the file explorer on the left shows the project structure with files like `External.css`, `app.py`, `DigiCertGlobalRootCA`, `login.html`, `index.html`, and `register.html`. The status bar at the bottom indicates the current line and column (Ln 7, Col 11), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the Python version (3.11.3 64-bit), and the system temperature (34°C). The taskbar at the bottom shows various application icons and the system clock (11:31 AM, 5/14/2023).

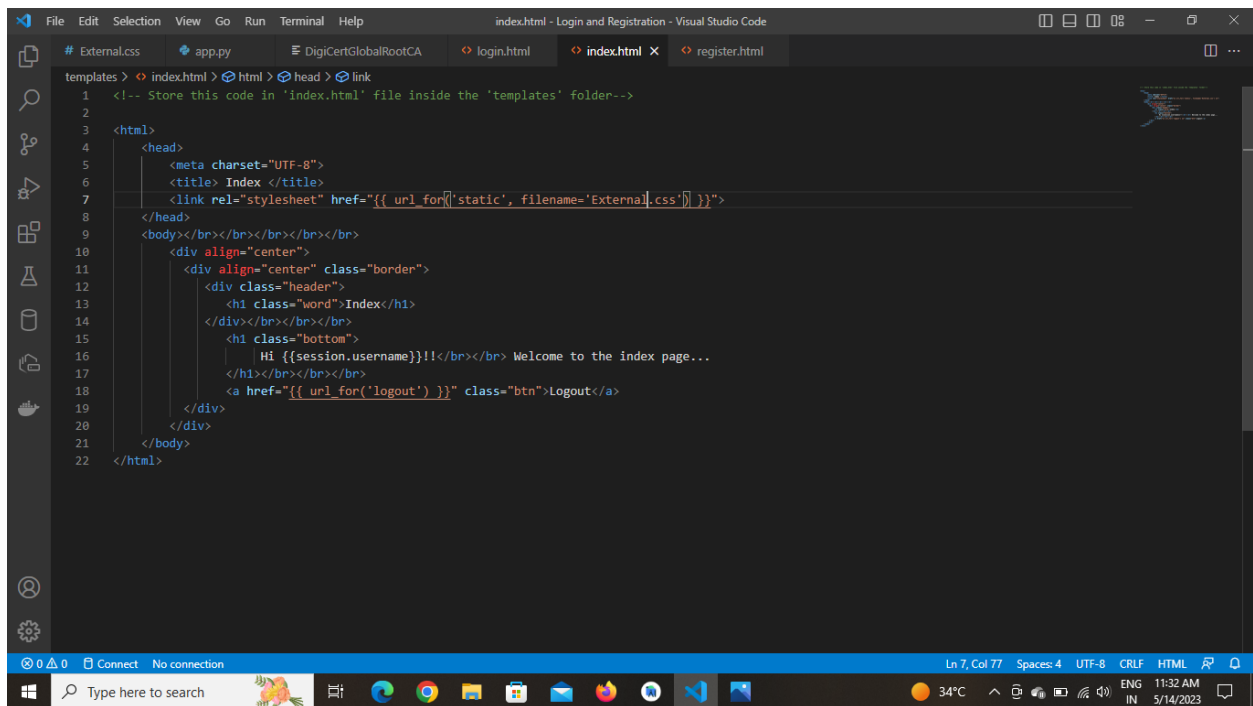
```
flask > app.py > ...
19 # store the user data (replace with database insertion in a real application)
20 users.append({'username': username, 'password': password})
21
22 return redirect('/login')
23
24 return render_template('register.html')
25
26 @app.route('/login', methods=['GET', 'POST'])
27 def login():
28     if request.method == 'POST':
29         username = request.form['username']
30         password = request.form['password']
31         # You can add validation and error handling here
32
33         # Check if the user exists (replace with database query in a real application)
34         for user in users:
35             if user['username'] == username and user['password'] == password:
36                 return f'Welcome, {username}!'
37
38         # Redirect to login page with an error message if authentication fails
39         return render_template('login.html', error='Invalid username or password')
40
41     return render_template('login.html')
42
43 if __name__ == '__main__':
44     app.run(debug=True)
45
```

External.css



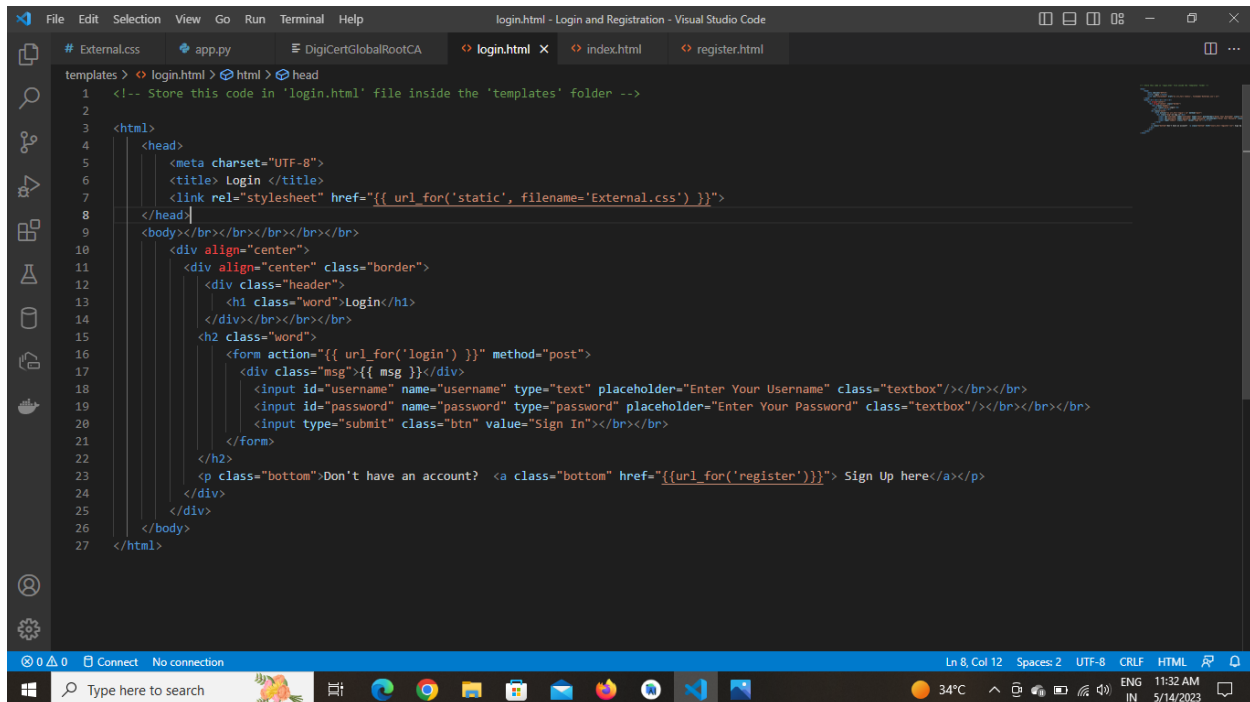


Index.html



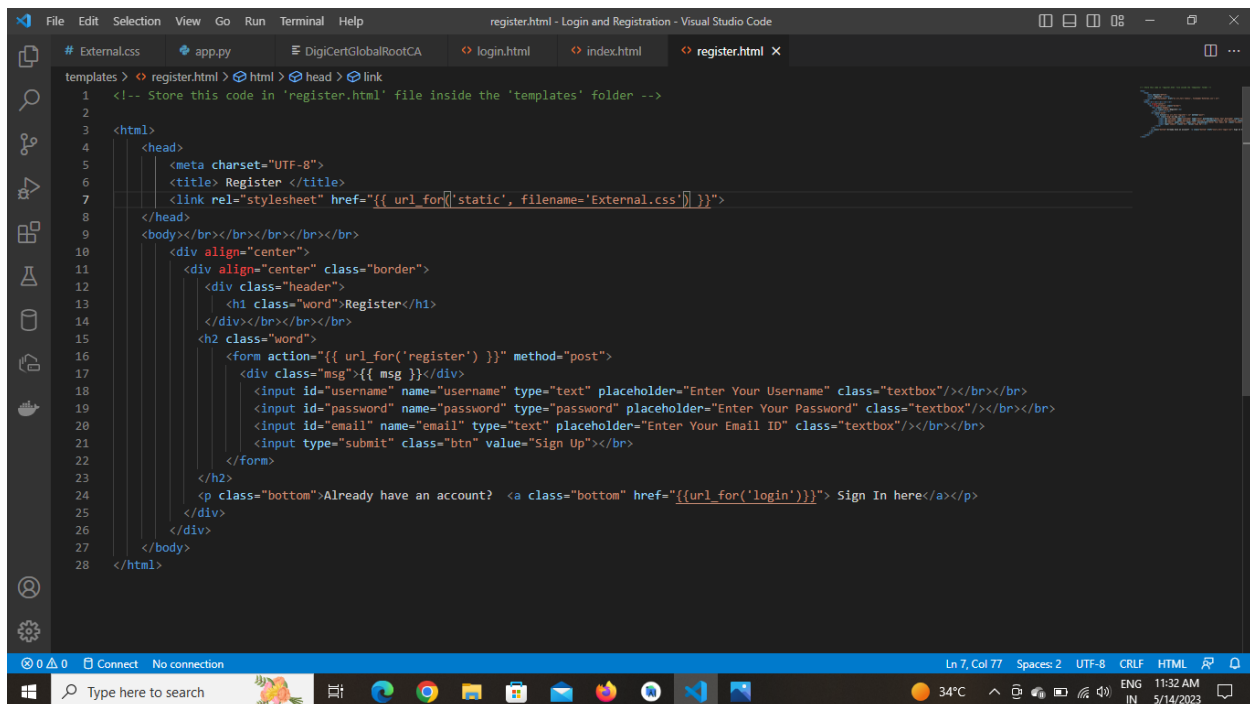
```
1  <!-- Store this code in 'index.html' file inside the 'templates' folder-->
2
3  <html>
4      <head>
5          <meta charset="UTF-8">
6          <title> Index </title>
7          <link rel="stylesheet" href="{{ url_for('static', filename='External.css')}}">
8      </head>
9      <body><br></br></br></br></br>
10         <div align="center">
11             <div align="center" class="border">
12                 <div class="header">
13                     <h1 class="word">Index</h1>
14                 </div><br></br></br>
15                 <div class="bottom">
16                     Hi {{session.username}}!!<br></br> Welcome to the index page...
17                 </div><br></br></br>
18                 <a href="{{ url_for('logout') }}" class="btn">Logout</a>
19             </div>
20         </div>
21     </body>
22 </html>
```

Login.html



```
1 <!-- Store this code in 'login.html' file inside the 'templates' folder -->
2
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title> Login </title>
7 <link rel="stylesheet" href="{{ url_for('static', filename='External.css') }}">
8 </head>
9 <body><br></br></br></br></br>
10 <div align="center">
11 <div align="center" class="border">
12 <div class="header">
13 <h1 class="word">Login</h1>
14 </div><br></br></br></br>
15 <h2 class="word">
16 <form action="{{ url_for('login') }}" method="post">
17 <div class="msg">{{ msg }}</div>
18 <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/><br></br>
19 <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/><br></br></br>
20 <input type="submit" class="btn" value="Sign In"/></br></br>
21 </form>
22 </h2>
23 <p class="bottom">Don't have an account? <a class="bottom" href="{{ url_for('register') }}"> Sign Up here</a></p>
24 </div>
25 </div>
26 </body>
27 </html>
```


Register.html



```
1 <!-- Store this code in 'register.html' file inside the 'templates' folder -->
2
3
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title> Register </title>
8     <link rel="stylesheet" href="{{ url_for('static', filename='External.css')}}">
9   </head>
10  <body><br></br></br></br></br>
11    <div align="center">
12      <div align="center" class="border">
13        <div class="header">
14          <h1 class="word">Register</h1>
15        </div><br></br></br></br>
16        <h2 class="word">
17          <form action="{{ url_for('register') }}" method="post">
18            <div class="msg">{{ msg }}</div>
19            <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/><br></br>
20            <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/><br></br>
21            <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/><br></br>
22            <input type="submit" class="btn" value="Sign Up"></br>
23          </form>
24        </h2>
25        <p class="bottom">Already have an account? <a class="bottom" href="{{url_for('login')}}"> Sign In here</a></p>
26      </div>
27    </div>
28  </body>
29 </html>
```

Login page:



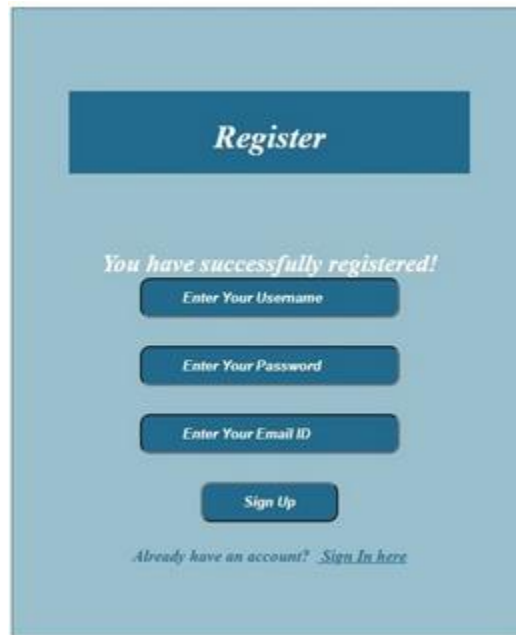
A login page form with a light blue background. At the top, there is a dark blue rectangular box containing the word "Login" in white, italicized font. Below this box, there are three input fields, each with a dark blue header and a light blue body. The first input field has the header "Enter Your Username". The second input field has the header "Enter Your Password". Below these two input fields is a dark blue button with the text "Sign In" in white. At the bottom of the form, there is a line of text: "Don't have an account? [Sign Up here](#)".

Registration page:



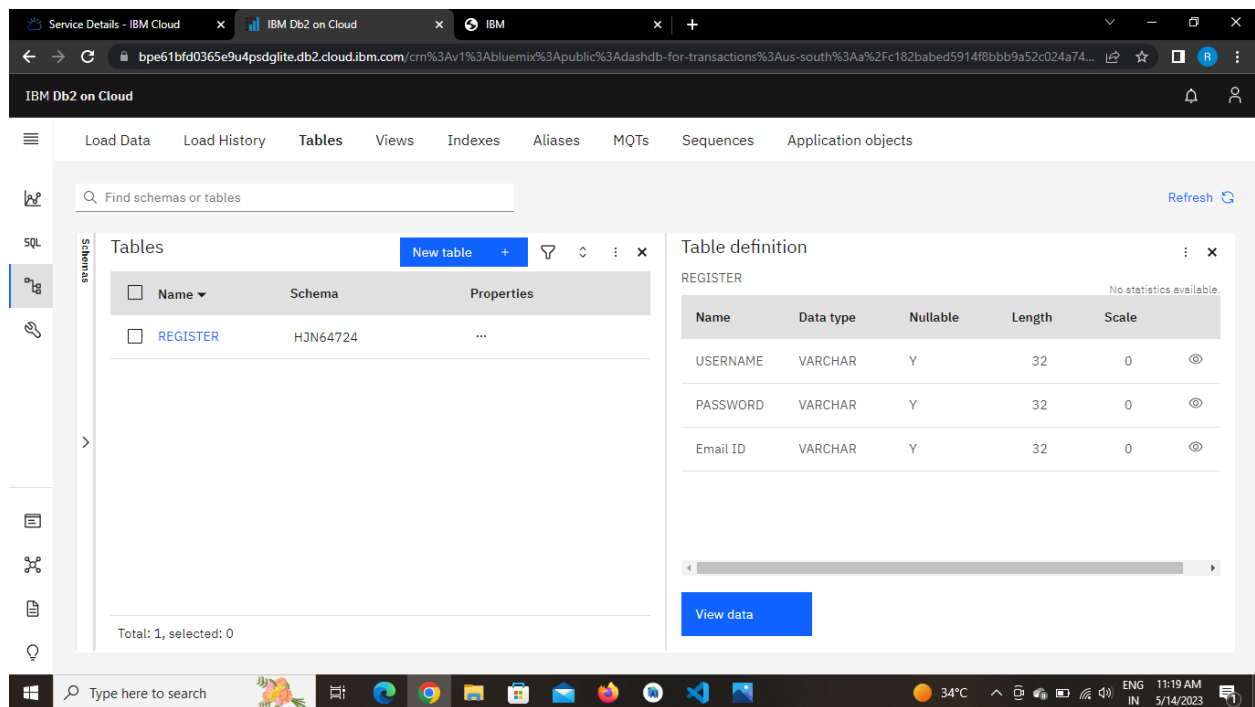
A registration page form with a light blue background. At the top, there is a dark blue rectangular box containing the word "Register" in white, italicized font. Below this box, there are three input fields, each with a dark blue header and a light blue body. The first input field has the header "Enter Your Username". The second input field has the header "Enter Your Password". The third input field has the header "Enter Your Email ID". Below these three input fields is a dark blue button with the text "Sign Up" in white. At the bottom of the form, there is a line of text: "Already have an account? [Sign In here](#)".

If registration successful:



A registration success screen with a light blue background. At the top, a dark blue button labeled "Register" is centered. Below it, the text "You have successfully registered!" is displayed. Underneath, there are three dark blue buttons stacked vertically, labeled "Enter Your Username", "Enter Your Password", and "Enter Your Email ID". At the bottom, there is a "Sign Up" button and a link that says "Already have an account? Sign In here".

Database Table:

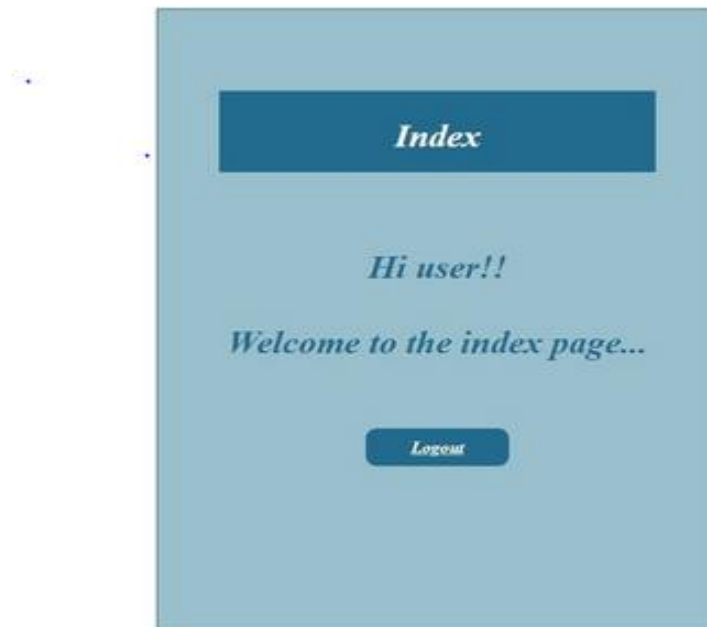


The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes tabs for "Load Data", "Load History", "Tables", "Views", "Indexes", "Aliases", "MQTs", "Sequences", and "Application objects". The "Tables" tab is active. On the left, a sidebar shows a search bar and a list of tables. The main area displays a table with columns "Name", "Schema", and "Properties". The table contains one entry: "REGISTER" under the "Name" column, "HJN64724" under the "Schema" column, and "..." under the "Properties" column. To the right of the table, a "Table definition" panel shows the schema for the "REGISTER" table. It includes a table with columns "Name", "Data type", "Nullable", "Length", and "Scale". The rows are: "USERNAME" (VARCHAR, Y, 32, 0), "PASSWORD" (VARCHAR, Y, 32, 0), and "Email ID" (VARCHAR, Y, 32, 0). Below the table definition, there is a "View data" button. The bottom of the screen shows a Windows taskbar with various application icons and a system tray with the date and time.

Name	Schema	Properties
REGISTER	HJN64724	...

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0
Email ID	VARCHAR	Y	32	0

If login successful:



Index page is displayed:

