

ASSIGNMENT-2

M.Sivabharathi

VV COLLEGE OF ENGINEERING

III-YEAR CSE

(TEAM LEADER)

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.

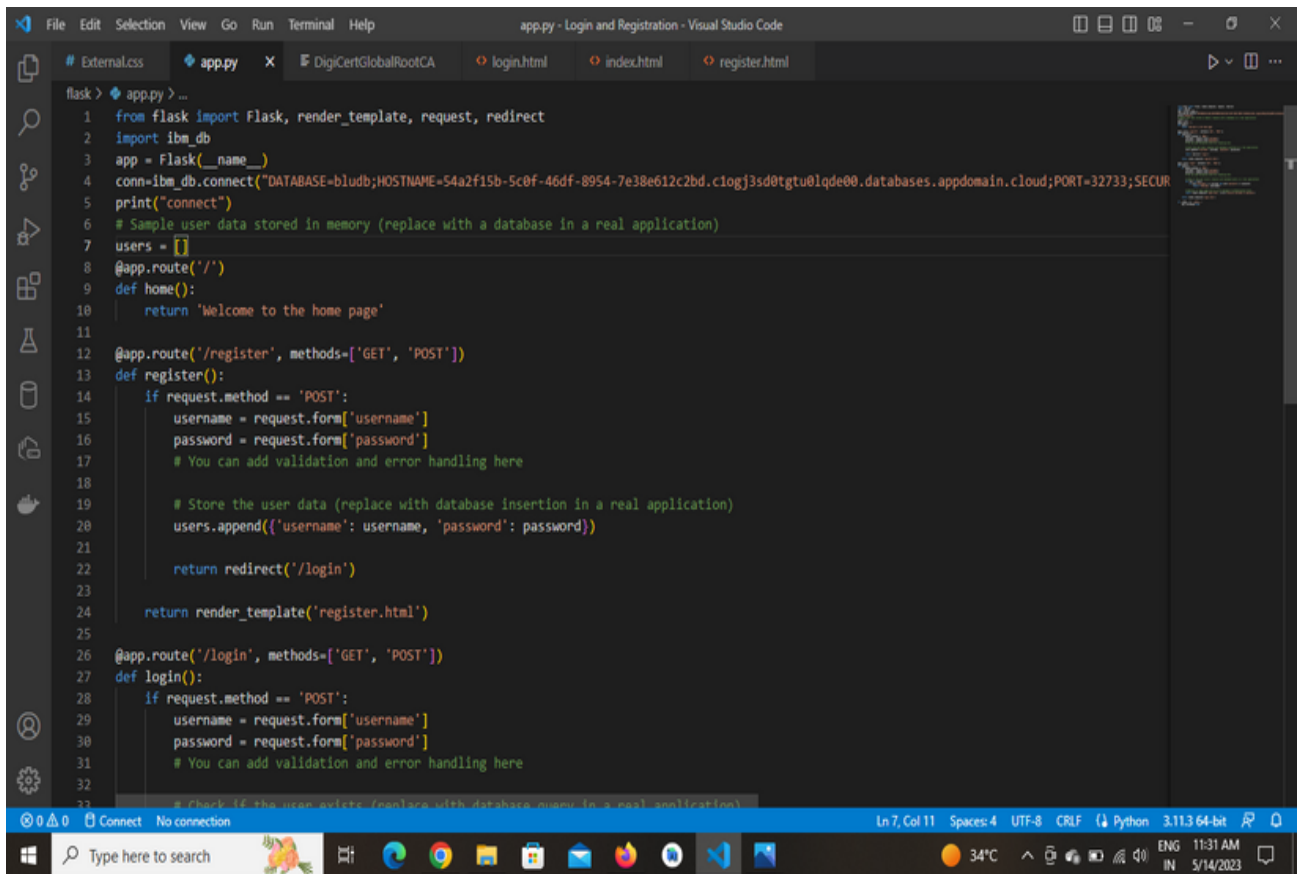
1. Login and Register html page

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)

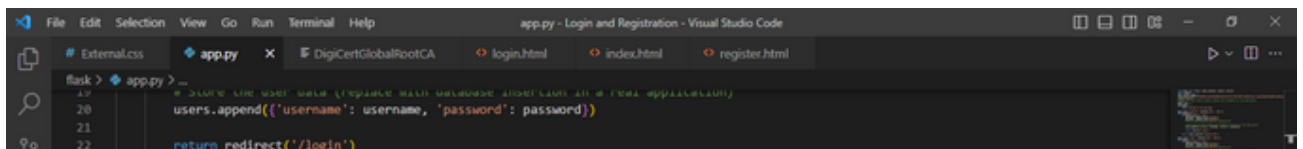
App.py:



The screenshot shows a Visual Studio Code editor window titled "app.py - Login and Registration - Visual Studio Code". The editor displays the following Python code for 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 status bar at the bottom indicates the file is at Line 7, Column 11, using UTF-8 encoding with CRLF line endings. The system tray shows a temperature of 34°C and the date 5/14/2023.



This close-up screenshot focuses on the register function in app.py, showing lines 13 through 22:

```
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     return render_template('register.html')
24
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
```

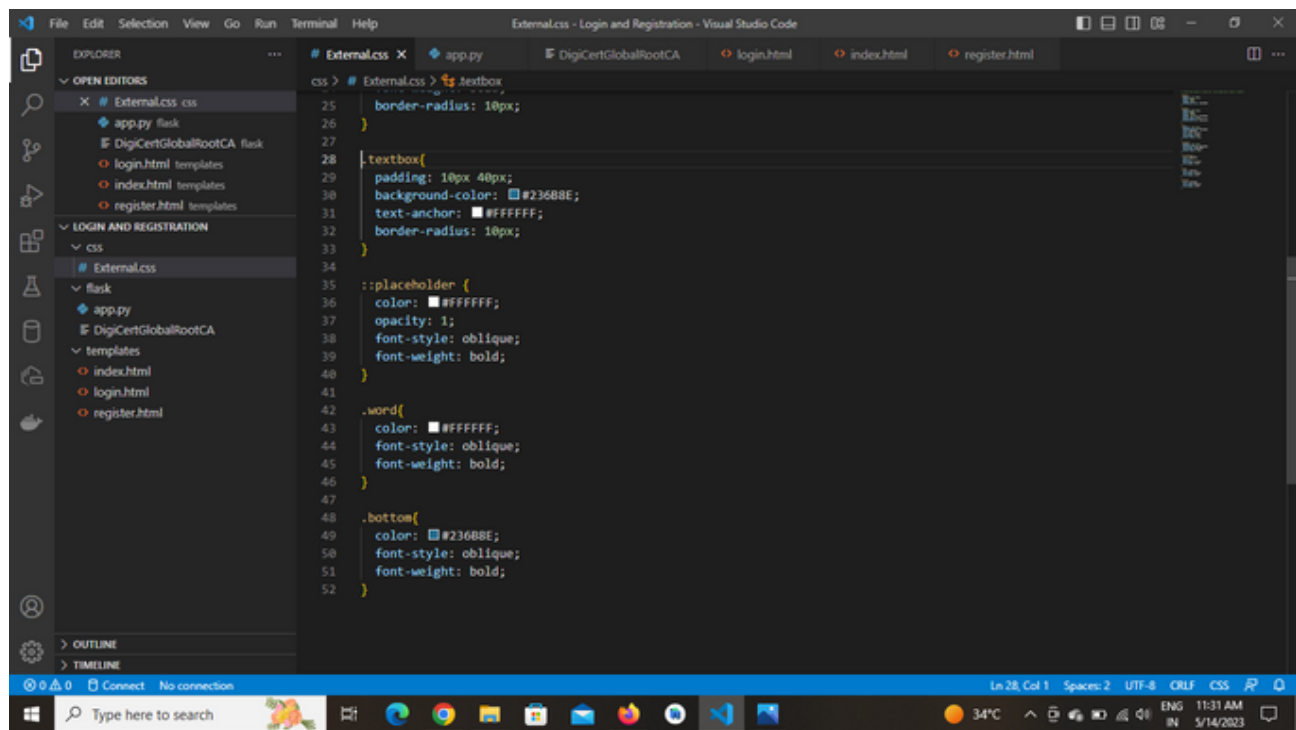
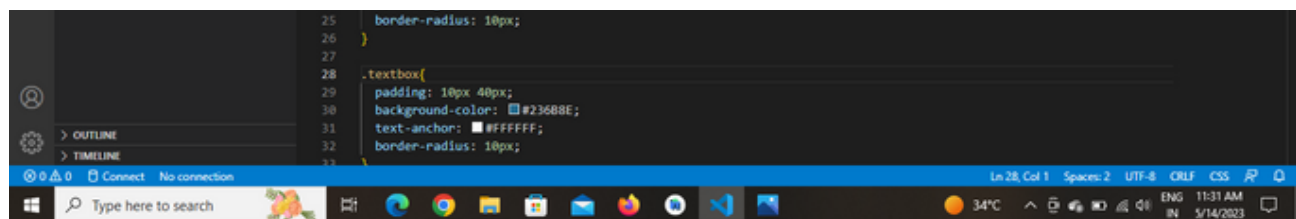
Ln 7, Col 11 Spaces: 4 UTF-8 CRLF Python 3.11.3 64-bit

Type here to search

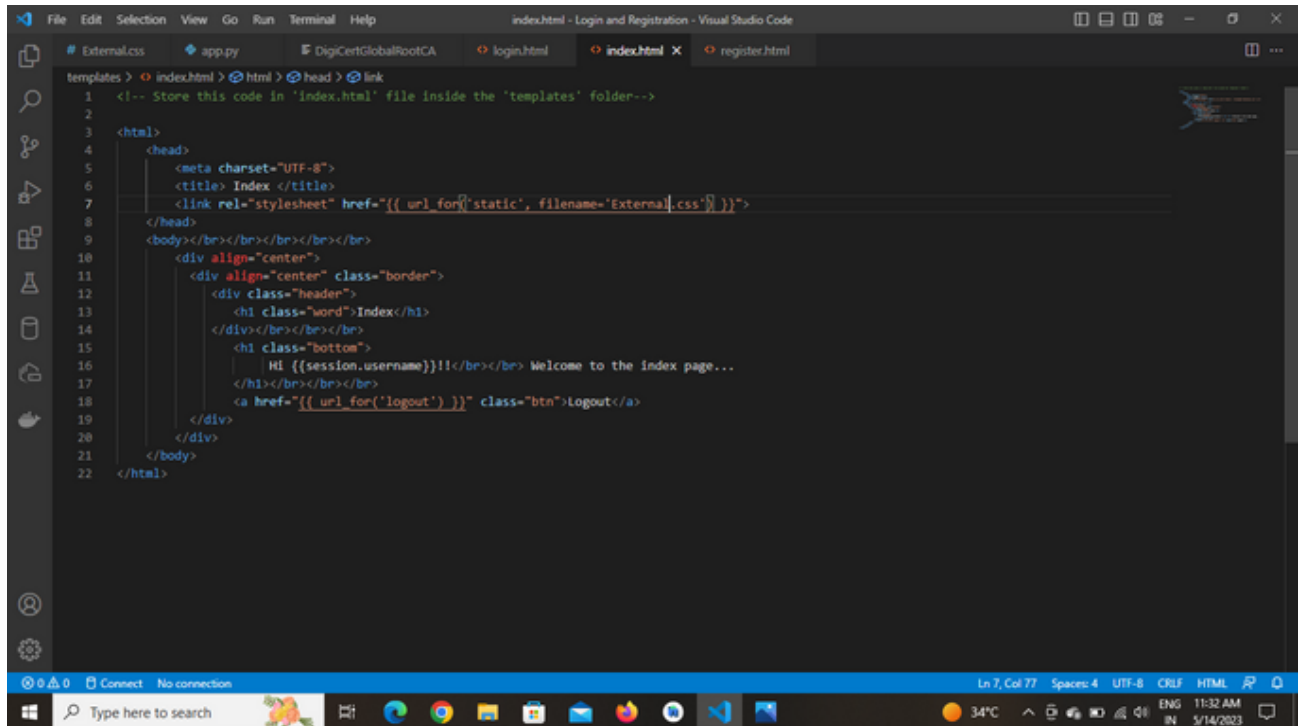
34°C ENG 11:31 AM IN 5/14/2023

External.css

```
File Edit Selection View Go Run Terminal Help External.css - Login and Registration - Visual Studio Code
EXPLORER
  OPEN EDITORS
    External.css x
    app.py flask
    DigiCertGlobalRootCA flask
    login.html templates
    index.html templates
    register.html templates
  LOGIN AND REGISTRATION
    css
    External.css
  flask
    app.py
    DigiCertGlobalRootCA
  templates
    index.html
    login.html
    register.html
css > External.css >  textbox
1  /* Store this code in 'style.css' file inside the 'static' folder*/
2
3  .header{
4      padding: 5px 120px;
5      width: 150px;
6      height: 70px;
7      background-color: #23688E;
8  }
9
10 .border{
11     padding: 80px 50px;
12     width: 400px;
13     height: 450px;
14     border: 1px solid #23688E;
15     border-radius: 0px;
16     background-color: #9AC8CD;
17 }
18
19 .btn {
20     padding: 10px 40px;
21     background-color: #23688E;
22     color: #FFFFFF;
23     font-style: oblique;
24     font-weight: bold;
```

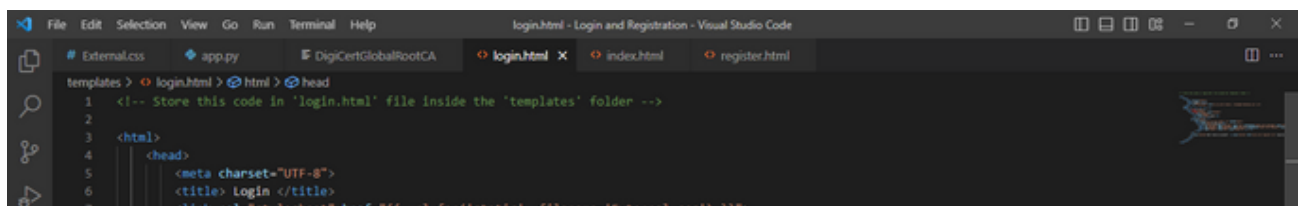


Index.html



```
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" class="border">
11       <div class="header">
12         <h1 class="word">Index</h1>
13       </div><br></br></br>
14       <div class="bottom">
15         <h1>{{ session.username }}!!</br></br> Welcome to the index page...
16         </h1><br></br></br>
17         <a href="{{ url_for('logout')}}" class="btn">Logout</a>
18       </div>
19     </div>
20   </body>
21 </html>
```

Login.html



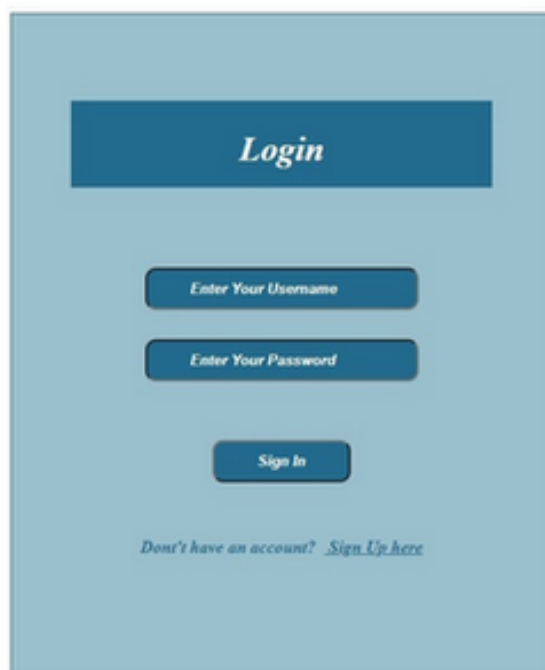
```
1 <!-- Store this code in 'login.html' file inside the 'templates' folder -->
2
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title> Login </title>
7     <link rel="stylesheet" href="{{ url_for('static', filename='External.css')}}">
```

```
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

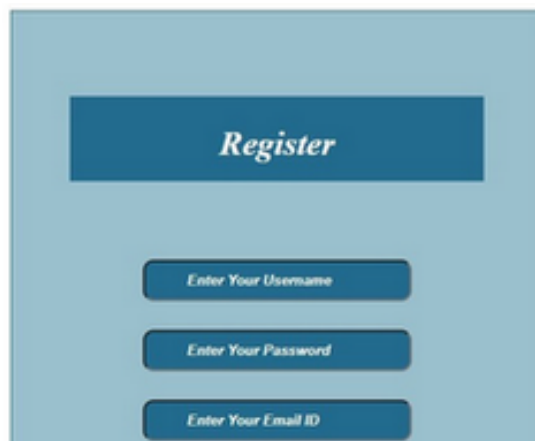
```
1  <!-- Store this code in 'register.html' file inside the 'templates' folder -->
2
3  <html>
4    <head>
5      <meta charset="UTF-8">
6      <title> Register </title>
7      <link rel="stylesheet" href="{{ url_for('static', filename='External.css')}}">
8    </head>
9    <body></br></br></br></br>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

Login page:



The login page features a light blue background. At the top, a dark blue rectangular box contains the word "Login" in a white, italicized serif font. Below this, there are three dark blue rounded rectangular input fields. The first field is labeled "Enter Your Username" in a small, italicized white font. The second field is labeled "Enter Your Password" in the same font. Below these fields is a dark blue rounded rectangular button labeled "Sign In" in a small, italicized white font. At the bottom of the form, the text "Don't have an account? [Sign Up here](#)" is displayed in a small, italicized dark blue font, with the link underlined.

Registration page:



The registration page features a light blue background. At the top, a dark blue rectangular box contains the word "Register" in a white, italicized serif font. Below this, there are three dark blue rounded rectangular input fields. The first field is labeled "Enter Your Username" in a small, italicized white font. The second field is labeled "Enter Your Password" in the same font. The third field is labeled "Enter Your Email ID" in the same font.

[Sign Up](#)

Already have an account? [Sign In here](#)

If registration successful:

Register

You have successfully registered!

[Enter Your Username](#)

[Enter Your Password](#)

[Enter Your Email ID](#)

[Sign Up](#)

Already have an account? [Sign In here](#)

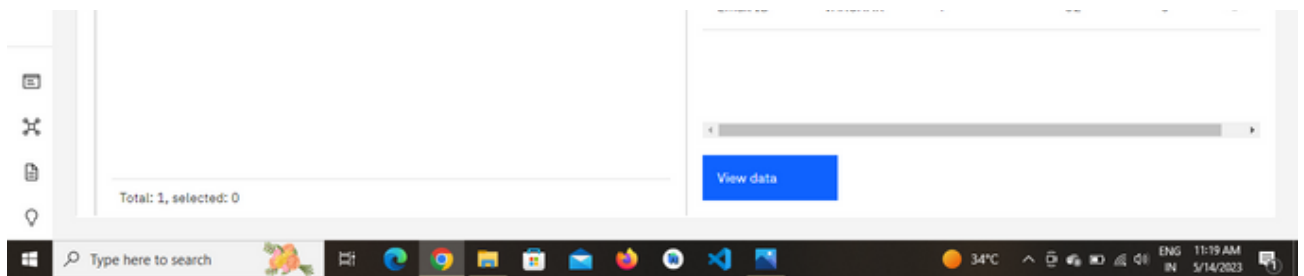
Database Table:

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. On the left, a search bar 'Find schemas or tables' is present. Below it, a table lists the available tables:

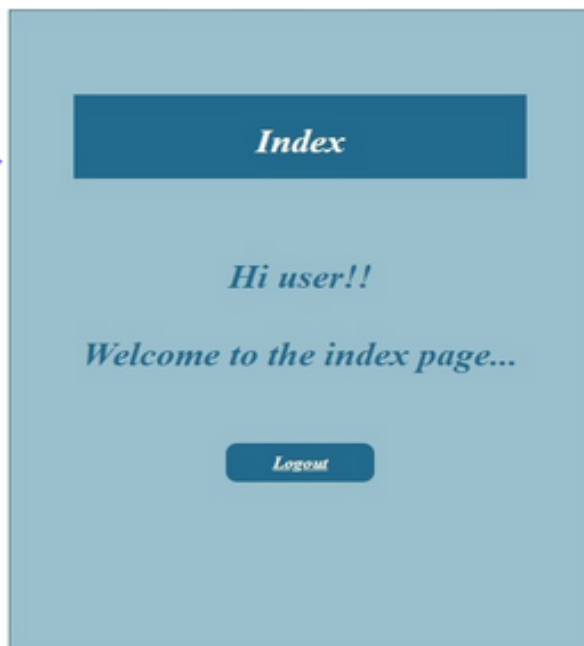
Name	Schema	Properties
REGISTER	HJN64724	...

On the right, the 'Table definition' panel for the 'REGISTER' table is displayed. It shows the table's structure with columns: USERNAME, PASSWORD, and Email ID. Each column is of type VARCHAR with a length of 32 and is nullable (Y). The scale is 0 for all columns.

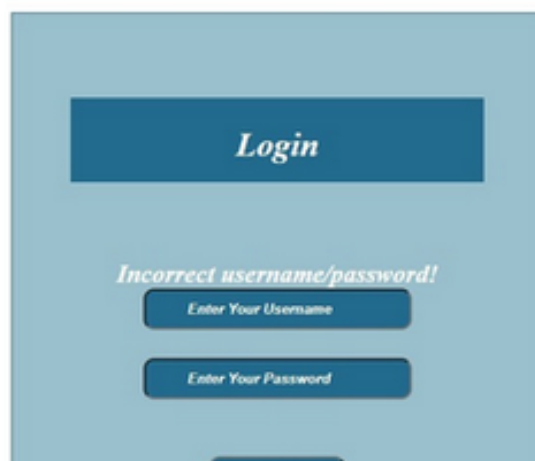
Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0
Email ID	VARCHAR	Y	32	0



If login successful:



Index page is displayed:



[Sign In](#)

Don't have an account? [Sign Up here](#)

