

ASSIGNMENT-2

S. DIVYA THARSHINI

VV COLLEGE OF ENGINEERING

III-YEAR CSE

CLOUD APP DEVELOPMENT

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.

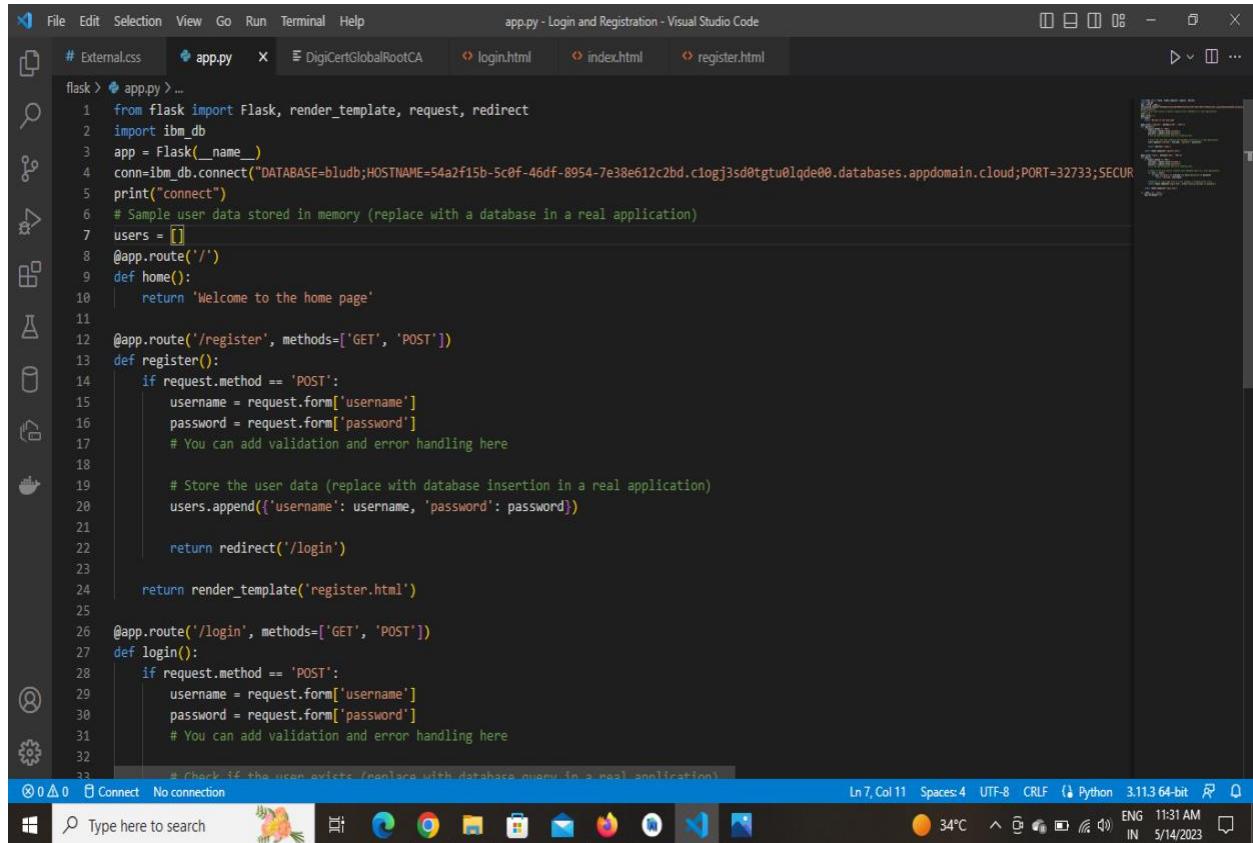
1. Login and Register html page

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:



```
flask > 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 a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: External.css, app.py, DigiCertGlobalRootCA, login.html, index.html, and register.html.
- Code Editor:** Displays the content of `app.py`. The code implements a simple login and registration system using Flask. It defines a list of users and handles POST requests for login. If successful, it returns a welcome message; otherwise, it renders the login template with an error message.
- Terminal:** Shows the command `flask run` being run, resulting in the output: `* Serving static files from: C:\Users\HP\PycharmProjects\LoginAndRegistration\ (Press CTRL+C to quit)`.
- Status Bar:** Shows the status bar with information like Ln 7, Col 11, Spaces: 4, UTF-8, CRLF, Python 3.11.3 64-bit, 34°C, ENG, 11:31 AM, and 5/14/2023.

External.css

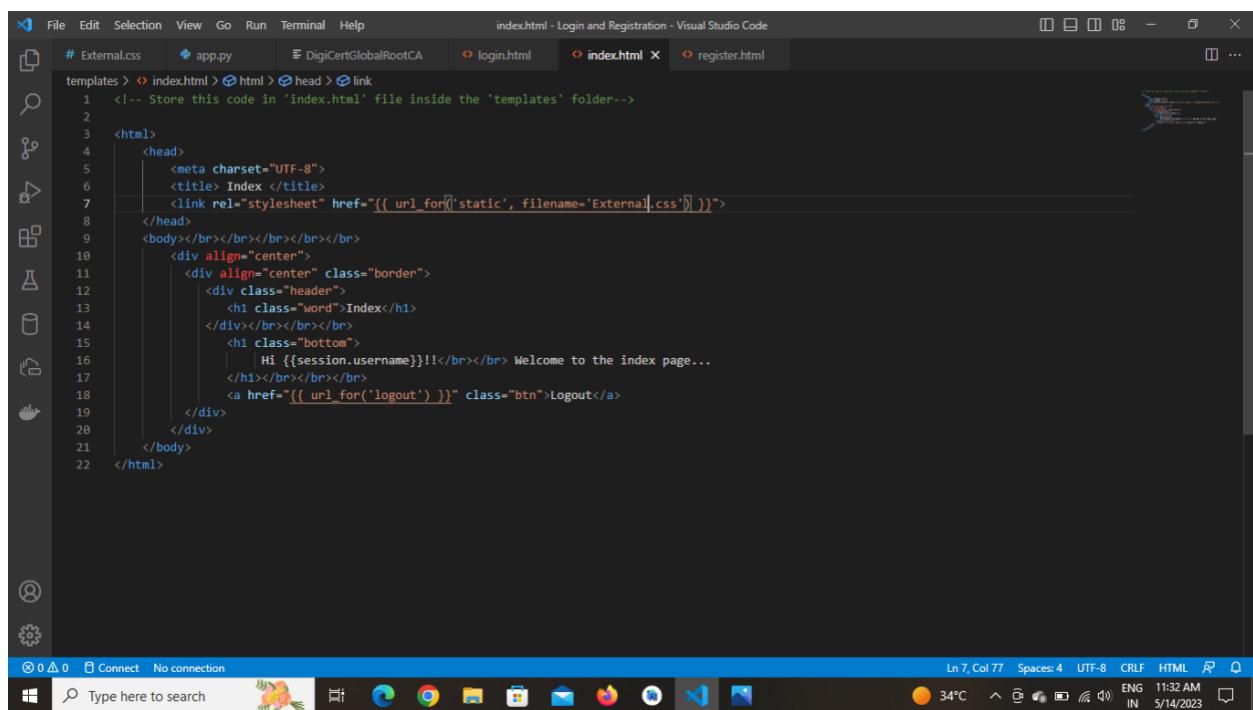
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS".
 - css > # External.css.css
 - app.py
 - DigiCertGlobalRootCA.flask
 - login.html.templates
 - index.html.templates
 - register.html.templates- Editor:** The "External.css" file is open, displaying CSS code for a login and registration application.
- Code:**

```
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: #236B8E;
8  }
9
10 .border{
11     padding: 80px 50px;
12     width: 400px;
13     height: 450px;
14     border: 1px solid #236B8E;
15     border-radius: 0px;
16     background-color: #9AC0CD;
17 }
18
19 .btn {
20     padding: 10px 40px;
21     background-color: #236B8E;
22     color: #FFFFFF;
23     font-style: oblique;
24     font-weight: bold;
25     border-radius: 10px;
26 }
27
28 .textbox{
29     padding: 10px 40px;
30     background-color: #236B8E;
31     text-anchor: #FFFFFF;
32     border-radius: 10px;
33 }
```
- Bottom Status Bar:** Shows "Ln 28 Col 1 Spaces:2 UTF-8 CRLF CSS" and system information like "34°C", "ENG 11:31 AM", and "IN 5/14/2023".

```
css > # External.css > .textbox
25 border-radius: 10px;
26 }
27
28 .textbox{
29 padding: 10px 40px;
30 background-color: #23688E;
31 text-anchor: #FFFFFF;
32 border-radius: 10px;
33 }
34
35 ::placeholder {
36 color: #FFFFFF;
37 opacity: 1;
38 font-style: oblique;
39 font-weight: bold;
40 }
41
42 .word{
43 color: #FFFFFF;
44 font-style: oblique;
45 font-weight: bold;
46 }
47
48 .bottom{
49 color: #23688E;
50 font-style: oblique;
51 font-weight: bold;
52 }
```

Index.html

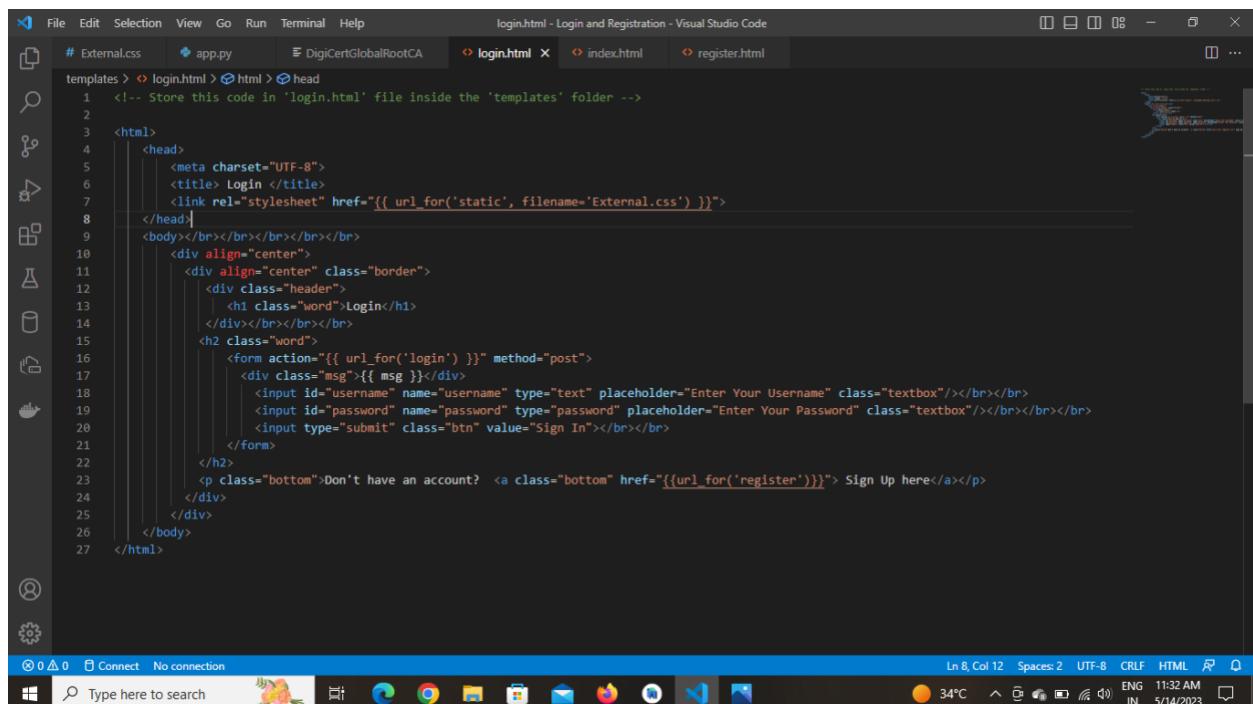


The screenshot shows the Visual Studio Code interface with the file "index.html" open. The code editor displays the following HTML template:

```
# External.css app.py DigiCertGlobalRootCA login.html index.html register.html
templates > index.html > html > head > link
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>
10     <div align="center">
11         <div align="center" class="border">
12             <div class="header">
13                 <h1 class="word">Index</h1>
14             </div><br><br>
15             <h1 class="bottom">
16                 |   Hi {{session.username}}!<br>
17                 |   </h1><br><br>
18                 |   <a href="{{ url_for('logout') }}" class="btn">Logout</a>
19             </div>
20         </div>
21     </body>
22 </html>
```

The code includes a comment at the top indicating it should be stored in the 'index.html' file inside the 'templates' folder. It features a centered layout with a border around the main content area. The content itself is a simple greeting message followed by a logout link.

Login.html



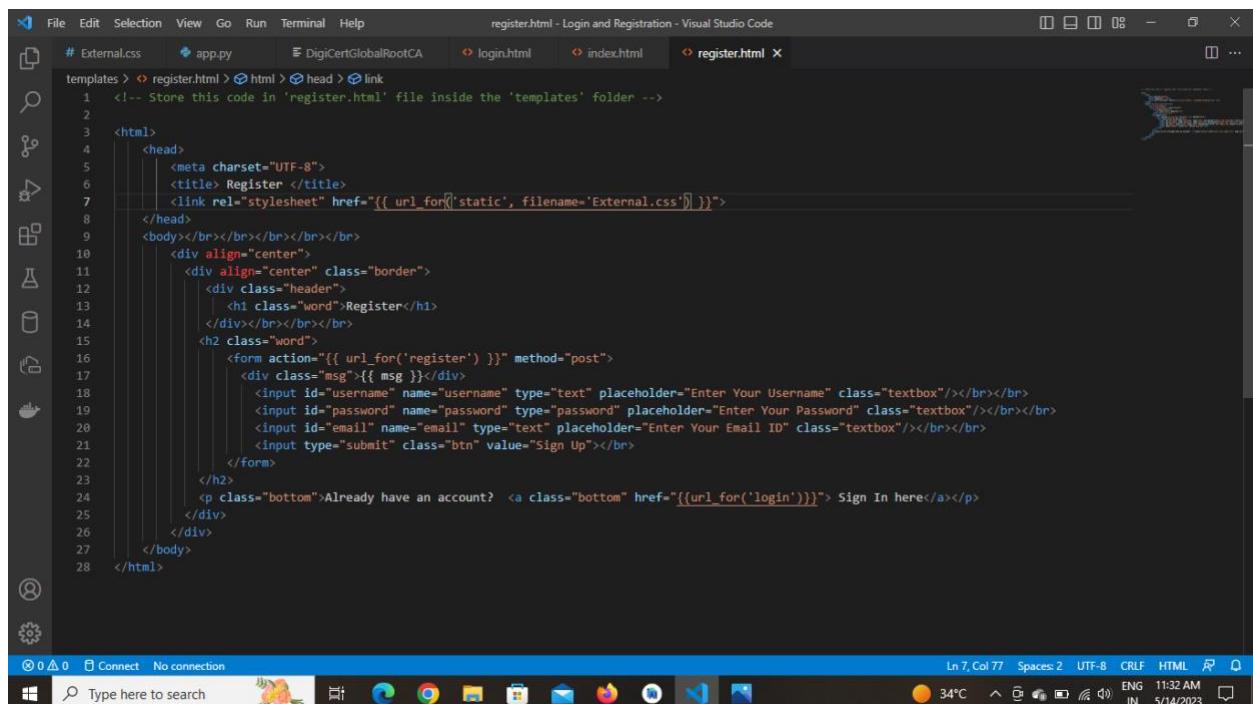
The screenshot shows the Visual Studio Code interface with the 'login.html' file open. The code is a template for a login page:

```
# External.css
# app.py
# DigiCertGlobalRootCA
# login.html
# index.html
# register.html

templates > login.html > html > head
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>
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>
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>
```

The code includes a header section with meta tags, title, and a CSS link. The body contains a centered form with a border. It features a header with a large 'Login' h1, a sub-section with a 'Sign In' button, and a bottom section for new users.

Register.html



The screenshot shows the Visual Studio Code interface with the file 'register.html' open. The code is a template for a registration form:

```
# External.css app.py DigiCertGlobalRootCA login.html index.html register.html - Login and Registration - Visual Studio Code

templates > register.html > HTML > head > link
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>
10     <div align="center">
11         <div align="center" class="border">
12             <div class="header">
13                 <h1 class="word">Register</h1>
14             </div><br><br>
15             <h2 class="word">
16                 <form action="{{ url_for('register') }}" method="post">
17                     <div class="msg">{{ msg }}>
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>
```

The code includes HTML tags for the document structure, a header section with a title and a CSS link, a main content area with a border, and a form for user input. It also includes a message div and a bottom navigation link.

Login page:



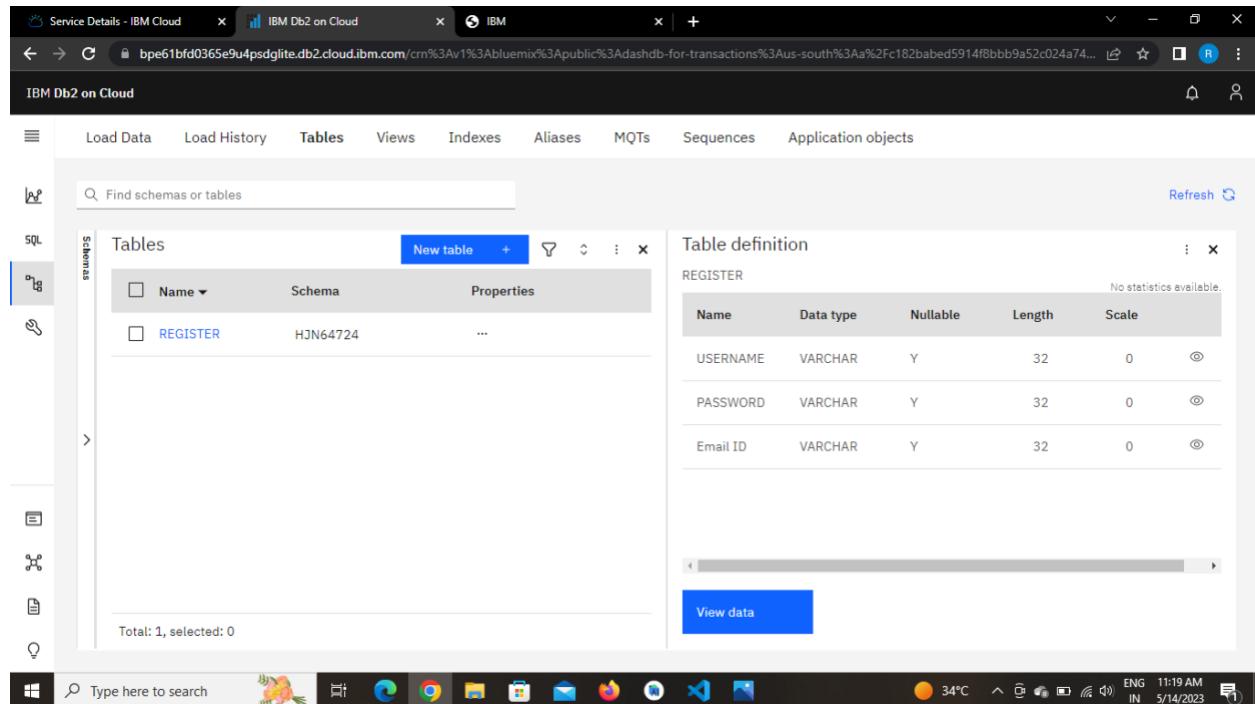
Registration page:



If registration successful:



Database Table:



Screenshot of the IBM Db2 on Cloud interface. The top navigation bar shows "Service Details - IBM Cloud", "IBM Db2 on Cloud", and "IBM". The main title is "IBM Db2 on Cloud". The left sidebar has icons for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The "Tables" icon is selected. The main content area shows a table named "REGISTER" in the "Tables" section. To the right, a "Table definition" panel shows the structure of the "REGISTER" table:

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

At the bottom of the table definition panel is a "View data" button.

If login successful:



Indexpage is displayed:



