

Assignment -2
IBM CLOUD DB2

Assignment Date	29 September 2022
Student Name	ARUNA.P
Student Roll Number	613019104007
Maximum Marks	2 Marks

Questions:

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

The screenshot displays the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is active, showing a search bar and a table list. The table list contains one entry: 'STUDENTS' in the 'KGB92696' schema. The 'Table definition' panel on the right shows the structure of the 'STUDENTS' table, which has four columns: 'USERNAME' (CHAR, 5, nullable), 'ROLL NUMBER' (INTEGER, nullable), 'EMAIL ID' (VARCHAR, 32, nullable), and 'PASSWORD' (VARCHAR, 32, nullable). The table is approximately 1 row in size (32.0 KB) and was last updated on 2022-10-16 at 13:31:57. A 'View data' button is visible at the bottom of the table definition panel. The bottom status bar shows the system temperature as 27°C Cloudy and the time as 10:26 AM on 17-10-2022.

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	5	0
ROLL NUMBER	INTEGER	Y		0
EMAIL ID	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

2. Perform UPDATE,DELETE Queries with user table.

UPDATE TABLE:-

The screenshot shows the IBM Db2 on Cloud console. On the left, the 'Data objects' tab is selected, showing a database named 'KGB92696'. The 'Data' tab is also visible. In the center, a SQL script is being edited in a text area. The script contains the following SQL statement:

```
INSERT INTO students values('aruna','06','aruna@gmail.com','1248');
```

Below the script editor, a 'History' table shows the execution history of the script. The table has columns: Script, Date, Status, and Runtime.

Script	Date	Status	Runtime
Untitled - 1	Oct 17, 2022 1:22:07 PM	✓ 1	0.006 s
INSERT INTO students values('aruna','06','aruna@gmail.com','124...		✓	0.006 s

The screenshot shows the IBM Db2 on Cloud console. The 'Tables' tab is selected, showing a table named 'KGB92696.STUDENTS'. The table has four columns: USERNAME, ROLL NUMBER, EMAIL ID, and PASSWORD. The table contains three rows of data.

USERNAME	ROLL NUMBER	EMAIL ID	PASSWORD
aruna	6	aruna@gmail.com	1248
deva	15	deva@gmail.com	6520
deva	15	deva@gmail.com	6520

DELETE TABLE:-

The screenshot shows the IBM Db2 on Cloud SQL editor interface. On the left, the 'Data objects' pane shows a database named 'KGB92696'. The main editor area shows a script titled '*Untitled - 1' with the following SQL statement:

```
1 DELETE FROM students WHERE USERNAME= 'deva';
2
```

Below the script, the 'History' pane displays the execution history of the script:

Script	Date	Status	Runtime
Untitled - 1	Oct 17, 2022 1:32:43 PM	1	0.014 s
DELETE FROM students WHERE USERNAME= 'deva'			0.014 s
Untitled - 1	Oct 17, 2022 1:29:52 PM	1	0.005 s
DELETE FROM students WHERE USERNAME= 'deva'			0.005 s
Untitled - 1	Oct 17, 2022 1:25:14 PM	2	0.010 s

The status column shows a yellow triangle with the number 1 for the first two entries and a green checkmark with the number 2 for the last entry, indicating successful execution.

The screenshot shows the IBM Db2 on Cloud SQL editor interface, displaying the contents of the 'KGB92696.STUDENTS' table. The table has four columns: USERNAME, ROLL NUMBER, EMAIL ID, and PASSWORD. The data is as follows:

USERNAME	ROLL NUMBER	EMAIL ID	PASSWORD
aruna	6	aruna@gmail.com	1248
aruna	6	aruna@gmail.com	1248
aruna	6	aruna@gmail.com	1248

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

3.CONNECT PYTHON CODE TO DB2.

NOTE:- Question 4 contains Question 3 answer

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

Answers

App.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=Db2New;HOSTNAME=19af6446-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=30699;USERNAM
E=kgb92696;PASSWORD=3IlWyGYU3bMIvwnX;SECURITY=SSL;SSLSERVERCERTIFICATE=DigiCer
tGlobalRootCA.crt;", "", "")

@app.route("/", methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users 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 only characters and numbers !'
```

```

        else:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg=msg)

@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 users WHERE username=? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        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 !'

            msg = 'Logged in successfully !'
            return render_template('dashboard.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg=msg)

if __name__ == '__main__':
    app.run(host='0.0.0.0')
    # app.run(debug=True)

```

Register.html

```
<!DOCTYPE html>
```

```

<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Smart Fashion register</title>

  <!-- bootstrap css cdn -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
"
    integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css">
    <!-- css stylesheet -->
    <link rel="stylesheet" href="style.css">
    <!-- font styles cdn -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">
</head>

<body>
  <!-- bootstrap navbar -->
  <!-- navbar ends -->
  <!-- Login form -->
  <div class="login text-center mt-5">

    <form action="/" method="post">
      <h2 class="form-text-h2"> Register Form </h2>
      <div class="msg">{{ msg }}</div>
      <input type="text" class="form-control" name="username"
placeholder="Enter Your Username" id="username"
        required>
      <input type="email" class="form-control" name="email"
placeholder="Enter Your Email ID" id="email" required>
      <input type="password" class="form-control" name="password"
placeholder="Enter Your Password" id="password"
        required>
      <button type="submit" id="button" class="btn btn-primary">
Register </button>

```

```

        <div class="note mt-3 text-center">
            <!--Register form -->
            <p> already have an account ? please login <a
href="/login">login! </a> </p>

        </div>
    </form>

</div>

</body>

</html>

```

Login.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Smart Fashion Login</title>
    <!-- favicon -->
    <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico"
type="image/x-icon"> -->
    <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon">
-->
    <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-
32x32.png">
    <!-- bootstrap css cdn -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
"
        integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css">
    <!-- css stylesheet -->
    <link rel="stylesheet" href="style2.css">
    <!-- font styles cdn -->
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap"
rel="stylesheet">

```

```

    <link
href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">
</head>

<body>

    <div class="login text-center mt-5">

        <form action="/login" method="post">
            <h2 class="form-text-h2"> Login Form </h2>
            <div class="msg">{{ msg }}</div>
            <div class="form-group mb-3">
                <label class="label" for="username">Username</label>
                <input type="text" class="form-control" placeholder="Username"
name="username" required>
            </div>
            <div class="form-group mb-3">
                <label class="label" for="password">Password</label>
                <input type="password" class="form-control"
placeholder="Password" name="password" required>
            </div>

            <button type="submit" id="button" class="btn btn-primary"> Login
</button>

            <div class="note mt-3 text-center">
                <!--Register form -->
                <p> Don't have an account yet? Click here to <a
href="/">register! </a> </p>
            </div>
        </form>
    </div>

</body>

</html>

```

Style.css

```

* {
    margin: 0;
    padding: 0;
}
body {
    color: #444444;
    background: linear-gradient(-45deg, #EE7752, #E73C7E, #23A6D5, #23D5AB);
    background-size: 400% 400%;
    position: relative;
    display: flex;
    flex-direction: column;

```



```
    align-items: center;
    justify-content: center;
    animation: anime 10s ease-in-out infinite;
}
@keyframes anime{
    0%{
        background-position: 0 50%;
    }
    50%{
        background-position: 100% 50%;
    }
    100%{
        background-position: 0 50%;
    }
}
```

```
form {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    background-color: white;
    width: 40%;
    height: 450px;
    border-radius: 50px;
    margin: 100px;
    gap: 25px;
}
```

```
h2 {
    font-size: large;
    font-weight: bolder;
}
```

```
input {
    width: 80% !important;
}
```

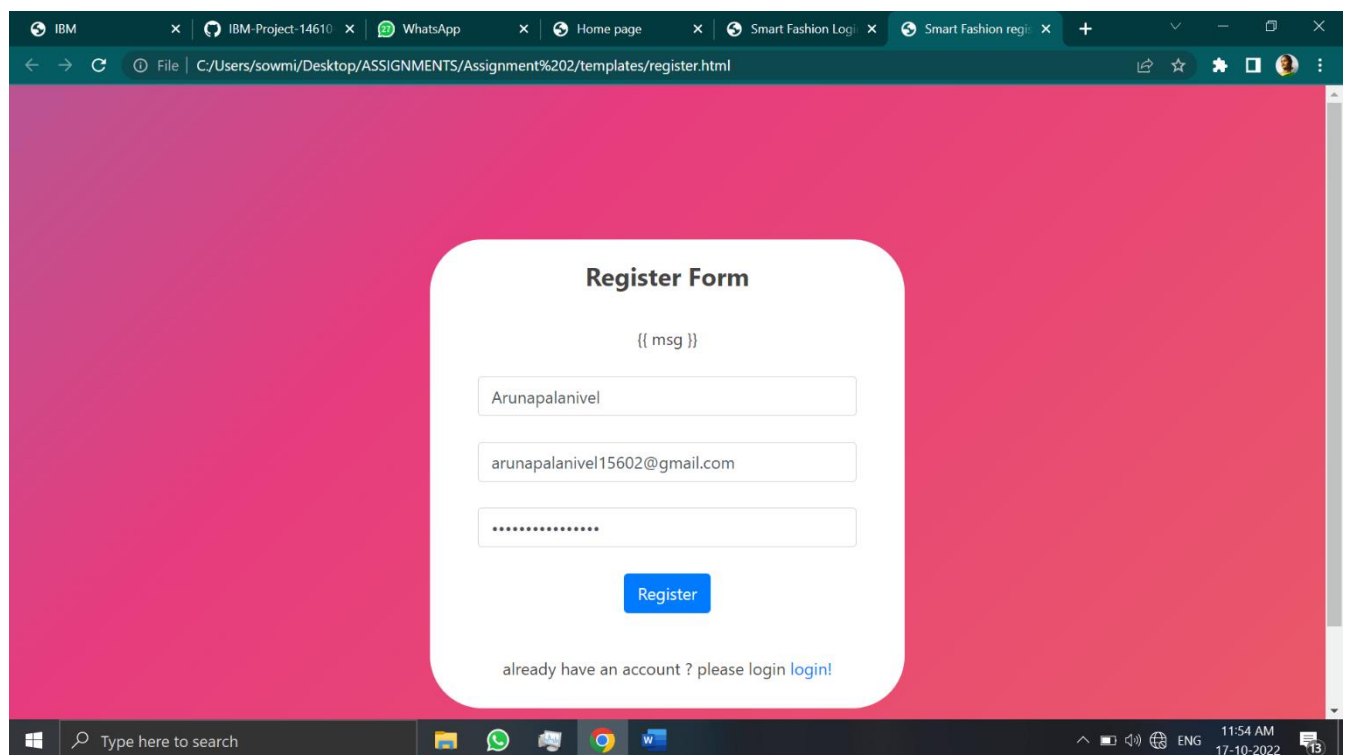
```
div {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    width: 90%;
}
```

```
.form-text-vk {
    font-size: small;
```

```
padding-bottom: 10px;
}

.form-text-h2 {
  font-size: x-large;
  padding-top: 10px;
}
```

OUTPUT:-
REGISTRATION PAGE:-



The screenshot shows a web browser window with multiple tabs. The active tab is titled "Smart Fashion regi...". The address bar shows the file path "C:/Users/sowmi/Desktop/ASSIGNMENTS/Assignment%202/templates/register.html". The main content area has a solid pink background. In the center, there is a white rounded rectangle containing a registration form. The form is titled "Register Form" and includes a placeholder message "{{ msg }}" above three input fields. The first input field contains the text "Arunapalanivel". The second input field contains the email address "arunapalanivel15602@gmail.com". The third input field is a password field with masked characters "*****". Below the input fields is a blue "Register" button. At the bottom of the form, there is a link that says "already have an account ? please login [login!](#)". The Windows taskbar is visible at the bottom of the screen, showing the search bar, task view button, and several application icons (File Explorer, WhatsApp, Edge, Word). The system clock in the bottom right corner indicates the time is 11:54 AM on 17-10-2022.

Register Form

{{ msg }}

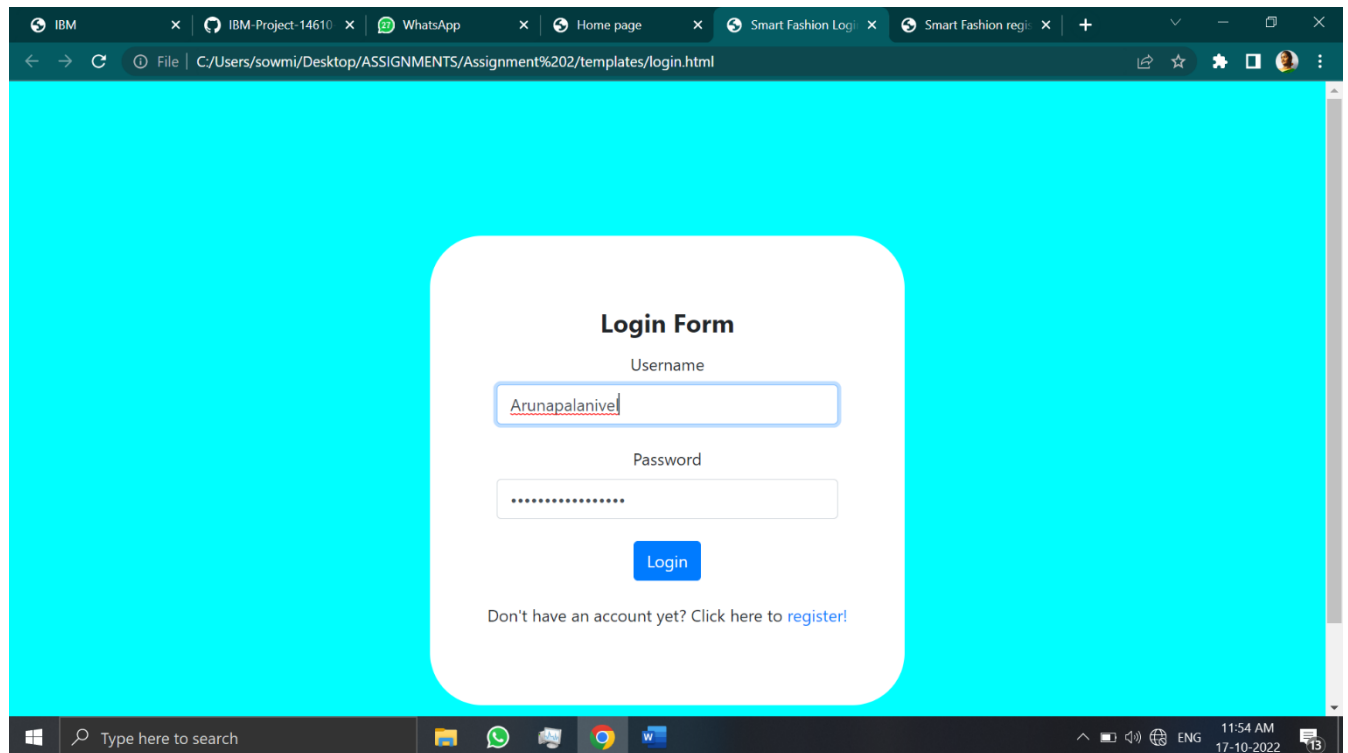
Arunapalanivel

arunapalanivel15602@gmail.com

Register

already have an account ? please login [login!](#)

LOGIN PAGE:-



IBM Project-14610 WhatsApp Home page Smart Fashion Log Smart Fashion regi

File C:/Users/sowmi/Desktop/ASSIGNMENTS/Assignment%202/templates/login.html

Login Form

Username

Password

Login

Don't have an account yet? Click here to [register!](#)

Type here to search 11:54 AM 17-10-2022

WELCOME PAGE:-

