

ASSIGNMENT-2

IBM CLOUD DBS

Assignment Date	29 September 2022
Student Name	K.Kowsika
Student Roll Number	613019106031
Maximum Marks	2 Marks

Questions:

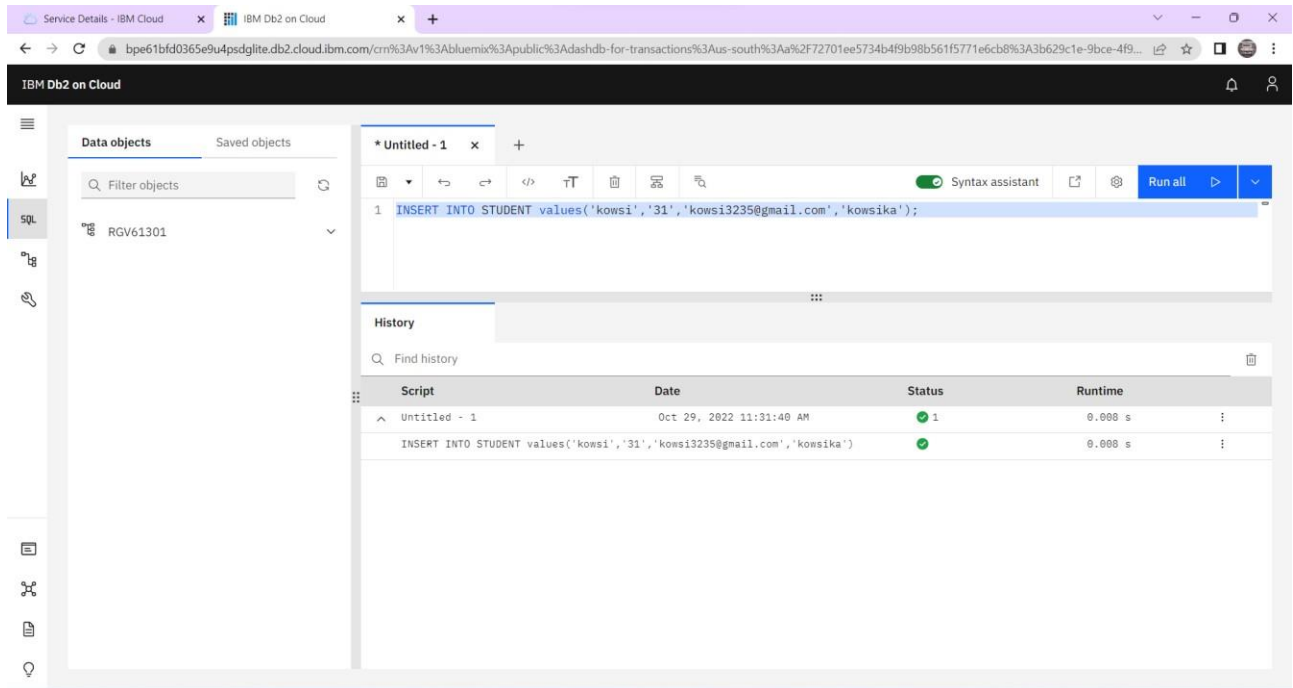
1. Create User table with user with email, username, roll number, password.

The screenshot displays the IBM Db2 on Cloud web interface. The 'Tables' tab is active, showing a list of tables. The 'STUDENT' table is selected, and its definition is shown on the right. The table has four columns: USERNAME (CHAR, 5, nullable), ROLL NUMBER (INTEGER, nullable), EMAIL ID (VARCHAR, 32, nullable), and PASSWORD (VARCHAR, 32, nullable). A 'View data' button is located at the bottom right of the table definition panel.

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:-

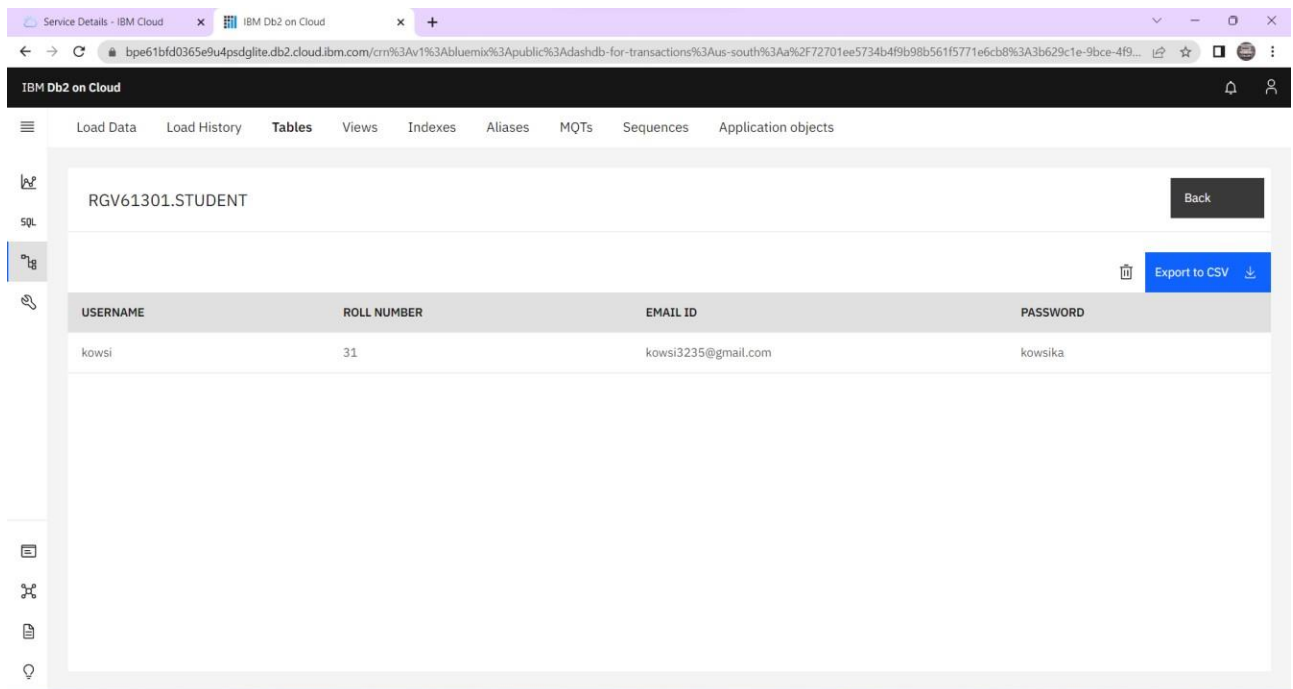


The screenshot shows the IBM Db2 on Cloud SQL editor interface. The left sidebar displays the 'Data objects' tab with a search filter and a list of objects, including 'RGV61301'. The main editor area shows a SQL script in a file named '*Untitled - 1'. The script contains the following SQL statement:

```
1 INSERT INTO STUDENT values('kowsi','31','kowsi3235@gmail.com','kowsika');
```

Below the script editor, the 'History' tab is active, showing a table of execution history:

Script	Date	Status	Runtime
Untitled - 1	Oct 29, 2022 11:31:40 AM	✓ 1	0.008 s
INSERT INTO STUDENT values('kowsi','31','kowsi3235@gmail.com','kowsika')		✓	0.008 s



The screenshot shows the IBM Db2 on Cloud SQL editor interface, displaying the table structure and data for 'RGV61301.STUDENT'. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The table structure is shown in the main editor area, with columns: USERNAME, ROLL NUMBER, EMAIL ID, and PASSWORD. The data is displayed in a table below the structure:

USERNAME	ROLL NUMBER	EMAIL ID	PASSWORD
kowsi	31	kowsi3235@gmail.com	kowsika

Buttons for 'Back' and 'Export to CSV' are visible in the top right corner of the table view.

DELETE TABLE:-

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F72701ee5734b4f9b98b561f5771e6cb8%3A3b629c1e-9bce-4f9...

IBM Db2 on Cloud

Data objects Saved objects

Filter objects

RGV61301

*Untitled - 1 x +

```
1 INSERT INTO STUDENT values('kowsi','31','kowsi3235@gmail.com','kowsika');
2 INSERT INTO STUDENT values('kow','36','kow3235@gmail.com','kow@123');
3 DELETE FROM STUDENT WHERE USERNAME = 'Kow';
```

History

Find history

Script	Date	Status	Runtime
Untitled - 1	Oct 29, 2022 11:35:21 AM	2 1	0.014 s
INSERT INTO STUDENT values('kowsi','31','kowsi3235@gmail.com','kowsika')			0.004 s
INSERT INTO STUDENT values('kow','36','kow3235@gmail.com','kow@123')			0.004 s
DELETE FROM STUDENT WHERE USERNAME = 'Kow'			0.006 s
Untitled - 1	Oct 29, 2022 11:31:40 AM	1	0.008 s
INSERT INTO STUDENT values('kowsi','31','kowsi3235@gmail.com','kowsika')			0.008 s

Service Details - IBM Cloud x IBM Db2 on Cloud x +

bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F72701ee5734b4f9b98b561f5771e6cb8%3A3b629c1e-9bce-4f9...

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

RGV61301.STUDENT

Back

Export to CSV

USERNAME	ROLL NUMBER	EMAIL ID	PASSWORD
kow	36	kow3235@gmail.com	kow@123
kow	36	kow3235@gmail.com	kow@123
kow	36	kow3235@gmail.com	kow@123
kow	36	kow3235@gmail.com	kow@123

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 =

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(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.execute(prepare_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>Register Form</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>Login Form</title>

    <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-
32x32.png">

    <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">

    <link rel="stylesheet" href="styleless.css">

    <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:-

Assignment 1.pdf x mail.google.com x docs.google.com x Skill and Job Recommender x Skill and Job Recommender x Skill and Job Recommender x + -

File | C:/Users/rajas/Downloads/Register%20(4).html

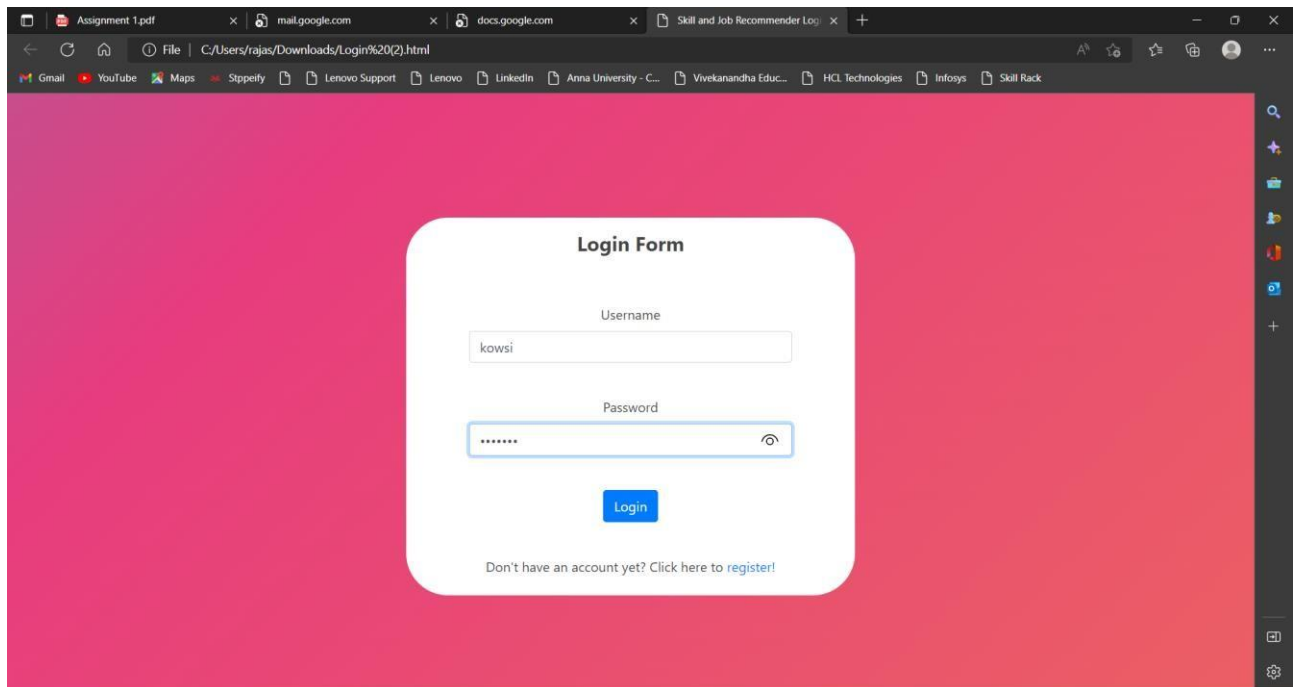
Gmail YouTube Maps Stopeify Lenovo Support Lenovo LinkedIn Anna University - C... Vivekanandha Educ... HCL Technologies Infosys Skill Rack

Register Form

Register

Already have an account ? Please login login!

LOGIN PAGE:-



Assignment 1.pdf x mail.google.com x docs.google.com x Skill and Job Recommender Log x +

File | C:/Users/rajas/Downloads/Login%20(2).html

Gmail YouTube Maps Stopeify Lenovo Support Lenovo LinkedIn Anna University - C... Vivekanandha Educ... HCL Technologies Infosys Skill Rack

Login Form

Username

kowsi

Password

Login

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

WELCOME PAGE:-

