

Create DB2 and connect python

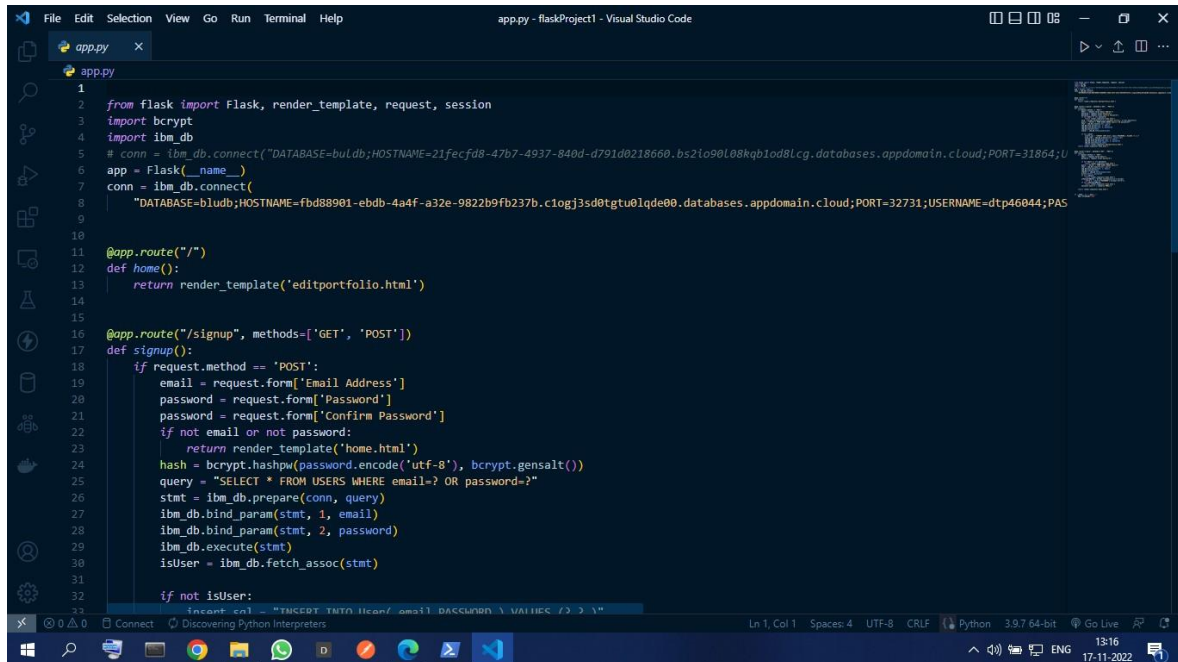
Date	17 November 2022
Team ID	PNT2022TMID46354
Project Name	Smart Fashion Recommender Application

CREATING IBM DB2

IBM Cloud console interface showing the 'Db2-hy' resource page. The page includes a 'Getting started' section with instructions on finding credentials and a 'Need help?' section with a link to submit a support case. The left sidebar shows 'Manage' options like 'Getting started', 'Service credentials', and 'Connections'.

IBM Db2 on Cloud console interface showing the 'Tables' tab. The 'Schemas' section lists a schema named 'YGH02694' of type 'User'. The 'Tables' section lists two tables: 'LOGINDB' and 'REGISTER', both belonging to the 'YGH02694' schema. The bottom status bar shows 'Total: 1, selected: 1' for schemas and 'Total: 2, selected: 0' for tables.

CONNECT WITH PYTHON CODE



The image shows a Visual Studio Code editor window titled 'app.py - flaskProject1 - Visual Studio Code'. The editor displays a Python file named 'app.py' with the following code:

```
1 from flask import Flask, render_template, request, session
2 import bcrypt
3 import ibm_db
4 # conn = ibm_db.connect("DATABASE=buLdb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8Lcg.databases.appdomain.cloud;PORT=31864;U
5 app = Flask(__name__)
6 conn = ibm_db.connect(
7     "DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32731;USERNAME=dt46044;PAS
8
9
10
11 @app.route("/")
12 def home():
13     return render_template('editportfolio.html')
14
15
16 @app.route("/signup", methods=['GET', 'POST'])
17 def signup():
18     if request.method == 'POST':
19         email = request.form['Email Address']
20         password = request.form['Password']
21         password = request.form['Confirm Password']
22         if not email or not password:
23             return render_template('home.html')
24         hash = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
25         query = "SELECT * FROM USERS WHERE email=? OR password=?"
26         stmt = ibm_db.prepare(conn, query)
27         ibm_db.bind_param(stmt, 1, email)
28         ibm_db.bind_param(stmt, 2, password)
29         ibm_db.execute(stmt)
30         isUser = ibm_db.fetch_assoc(stmt)
31
32         if not isUser:
```

The code is a Flask application that connects to an IBM DB. It includes a home route and a signup route. The signup route checks if the email or password is empty and hashes the password before checking if the user exists in the database.

The status bar at the bottom shows the file encoding as 'UTF-8', the line and column as 'Ln 1, Col 1', and the Python interpreter as 'Python 3.9.7 64-bit'.