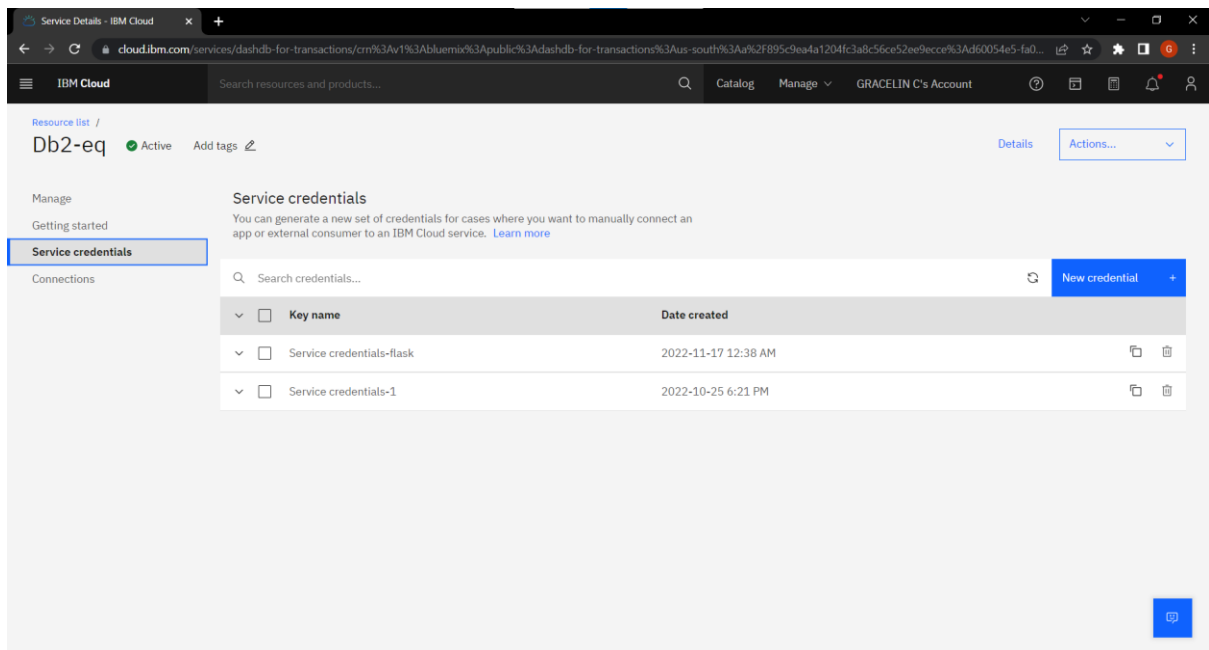






CREATE IBM DB2 AND CONNECT WITH PYTHON

TEAM ID	PNT2022TMID22598
PROJECT	SMART FASHION RECOMMENDER APPLICATION

Creating IBM DB2

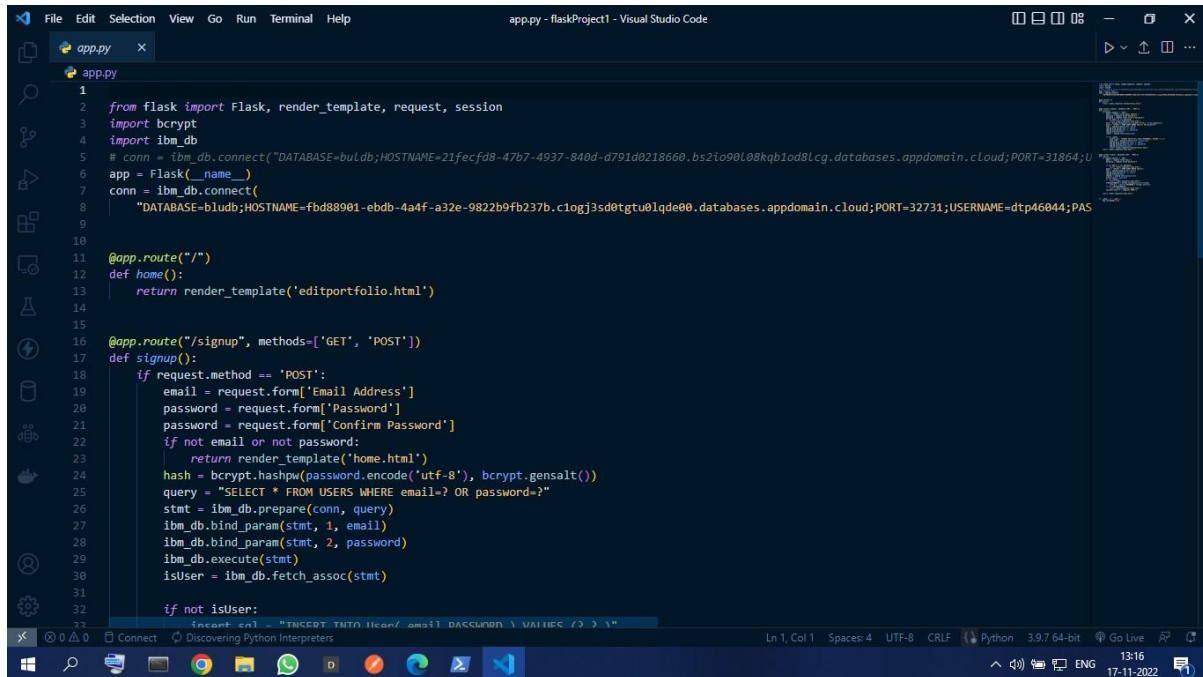


The screenshot shows the IBM Cloud console interface. The top navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Manage, and the user's account (GRACELIN C's Account). The main content area is titled 'Service Details - IBM Cloud' and shows the 'Db2-eq' resource, which is 'Active'. The left sidebar contains a 'Resource list' and a 'Manage' section with links to 'Getting started' and 'Service credentials' (which is highlighted). The 'Service credentials' section displays a table of existing credentials:

Key name	Date created	
Service credentials-flask	2022-11-17 12:38 AM	 
Service credentials-1	2022-10-25 6:21 PM	 

Below the table is a 'New credential' button. The bottom right corner of the console shows a blue chat icon.

CONNECTING WITH PYTHON



The screenshot 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
2 from flask import Flask, render_template, request, session
3 import bcrypt
4 import ibm_db
5 # conn = ibm_db.connect("DATABASE=buldb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;U
6 app = Flask(__name__)
7 conn = ibm_db.connect(
8     "DATABASE=buldb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0gtu0lqde00.databases.appdomain.cloud;PORT=32731;USERNAME=dtP46044;PAS
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 instance. It defines a home route and a signup route. The signup route handles POST requests, checks for email and password, hashes the password, and checks if the user already exists in the database. The database connection string is provided in the comments.

The status bar at the bottom shows the file encoding as "Ln 1, Col 1", the character set as "UTF-8", and the line ending as "CRLF". The Python interpreter is set to "Python 3.9.7 64-bit". The system clock shows "13:16" and the date "17-11-2022".