

## CREATE DB2 AND CONNECT PYTHON

Date	17 November 2022
Team ID	PNT2022TMID22341
Project Name	Smart Fashion Recommender Application

### DATABASE IN IBMDB2:

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. On the left, under 'Schemas', a table lists the schema 'LTV80221' of type 'User' with 7 tables. On the right, under 'Tables', a list shows tables 'ADMIN', 'CART', 'HOODIE', 'PRODUCTS', 'SHIRT', and 'SHOE' all belonging to the 'LTV80221' schema. A 'New table' button is visible at the top right of the tables list.

The screenshot shows the IBM Db2 on Cloud console interface with the 'LTV80221.USERS' table selected. The table has columns 'USERNAME', 'EMAIL', and 'PASSWORD'. The data is as follows:

USERNAME	EMAIL	PASSWORD
Hello	Hello1@gmail.com	helo123
a	a@gmail.com	a123
ab	ab@gmail.com	ab123
ak	sherazjabeen18@gmail.com	ak1
av2	abc@gmail.com	av2
b	b@gmail.com	b123

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

## PYTHON:

```
from flask import *
import ibm_db

conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2io90l08qkb1od8lcg.databases.appdomain.cloud; PORT=31498;
SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=ltv80221;
PWD=8fS5XCKIWaGcUCUW",'','')

app = Flask(__name__)

@app.route('/register')
def register():
    return render_template('register.html')

@app.route('/addrec',methods = ['POST', 'GET'])
def addrec():
    msg = ' '
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

        sql = "SELECT * FROM users WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not username or not password or not email:
            msg = 'Please fill the Missing Details !'

        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 = 'Account created successfully '
```

```

        from_email = Email("praba3043@gmail.com")
        to_email = To(email)
        subject = "Thank You For Registration"
        content = Content("text/plain", "Welcome to our S-Mart App")
        mail = Mail(from_email, to_email, subject, content)
        response = sg.client.mail.send.post(request_body=mail.get())
        print(response.status_code)
        print(response.body)
        print(response.headers)

        return render_template('login.html',msg=msg)
    return render_template('register.html',msg=msg)

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/checkrec',methods = ['POST', 'GET'])
def checkrec():
    msg = ' '
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE Email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_both(stmt)
        accounts=account
        if (account):
            if(password == accounts['PASSWORD']):
                msg = 'Logged in successfully !'
                return render_template('index.html',msg=msg)
            else :
                msg='Wrong Credentials'
    return render_template('login.html',msg=msg)

```