IMPLEMENTING WEB APPLICATION

Team ID	PNT2022TMID34001
Project Name	Inventory Management System for Retailers

Create IBM DB2 and connect with python
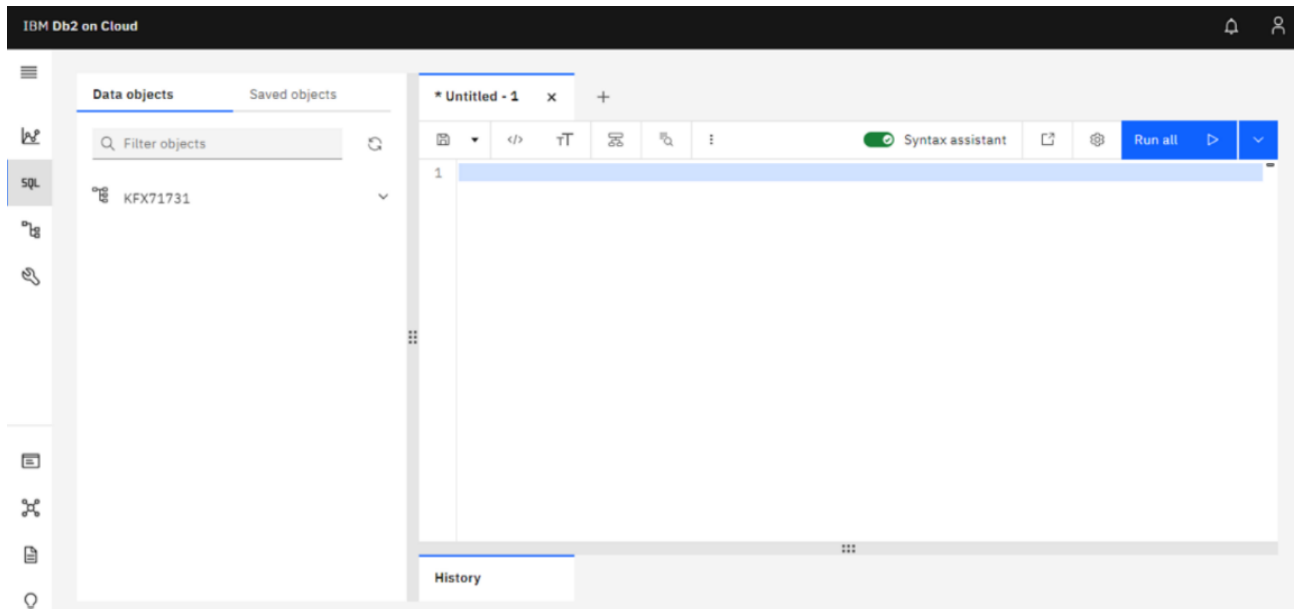
Sql_call.py

```python
import base64

import os

from flask import *

import ibm_db

import sendgrid

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail

from flask_session import Session

con = ibm_db.connect("DATABASE=bludb; HOSTNAME=_____;
PORT=_____; SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=_____;
PWD=_____","","")

class fetch_query_data:

    def fetch_data(sql):

        list = []
```

```python
        stmt = ibm_db.exec_immediate(con, sql)

        dictionary = ibm_db.fetch_both(stmt)

        while dictionary != False:

            list.append(dictionary)

            dictionary = ibm_db.fetch_both(stmt)

        if list:

            return list

        else:

            print("Data not fetched............. ")


class create_table:

    def item_table(email):

        user_id=check_query_data.check_signle_column('user','email',email)

        sql=f"CREATE TABLE product_details_{user_id[0]}(product_id BIGINT NOT NULL GENERATED
ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),supplier VARCHAR(50),date timestamp
DEFAULT CURRENT_TIMESTAMP,product_name VARCHAR(50) NOT NULL UNIQUE,purchase_price
BIGINT,selling_price BIGINT,stock BIGINT,total_selling_price BIGINT,status VARCHAR(20),low_stock
BIGINT,PRIMARY KEY (product_id));"

        ibm_db.exec_immediate(con, sql)


class insert_data_database:

    def insert_user_table(name,email,password,company_name):

        inser_sql = f"INSERT INTO user (name,email,password,company_name ) VALUES (?,?,?,?);"

        stmt = ibm_db.prepare(con, inser_sql)

        ibm_db.bind_param(stmt, 1, name)

        ibm_db.bind_param(stmt, 2, email)

        ibm_db.bind_param(stmt, 3, password)

        ibm_db.bind_param(stmt, 4, company_name)

        ibm_db.execute(stmt)
```

← → C 🔒 bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fbfa9a5654f9c430b80e8bef9cde5a7a3%3Ae97a5-5d26...

IBM Db2 on Cloud

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

QXH83236.USER

Back

🗑   Export to CSV ⬇

USER_ID	NAME	EMAIL	PASSWORD	COMPANY_NAME	PROFIT
23	srijith	srijithr2@gmail.com		srijith limited	37500
24	vinoth	vk103688@gmail.com	123456789	vk	0

← → C 🔒 bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fbfa9a5654f9c430b80e8bef9cde5a7a3%3Ae97a5-5d26...

IBM Db2 on Cloud

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

QXH83236.SALES_DETAILS_23

Back

🗑   Export to CSV ⬇

SALES_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	PRODUCT_ID	PHONE_NUMBER	DATE	PRODUCT_NAME	QUANTITY	BILLING_AMOUNT
1	vinoth	vinoth@gmail.com	3	7894563285	2022-11-18 15:41:36.108051	moto g 40 fusion	35	700000
2	pavan	pavan@gmail.com	3	784523596	2022-11-18 15:43:03.006777	moto g 40 fusion	5	100000
3	gowtham	gowtham@gmail.com	2	7894562586	2022-11-18 15:53:01.293793	boat tws 114	125	187500

← → C 🔒 bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fbfa9a5654f9c430b80e8bef9cde5a7a3%3Ae97a5-5d26...

IBM Db2 on Cloud

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

QXH83236.PRODUCT_DETAILS_23

Back

🗑   Export to CSV ⬇

PRODUCT_ID	SUPPLIER	DATE	PRODUCT_NAME	PURCHASE_PRICE	SELLING_PRICE	STOCK	TOTAL_SELLING_PRICE	STATUS	LOW_STOCK
1	lenovo	2022-11-18 15:39:13.236837	lenovo ideapad 320	25000	30000	65	1950000	Instock	10
2	Boat	2022-11-18 15:40:01.035559	boat tws 114	1200	1500	25	37500	Low stock	30
3	Motorola	2022-11-18 15:41:00.226766	moto g 40 fusion	15000	20000	0	0	out of stock	10