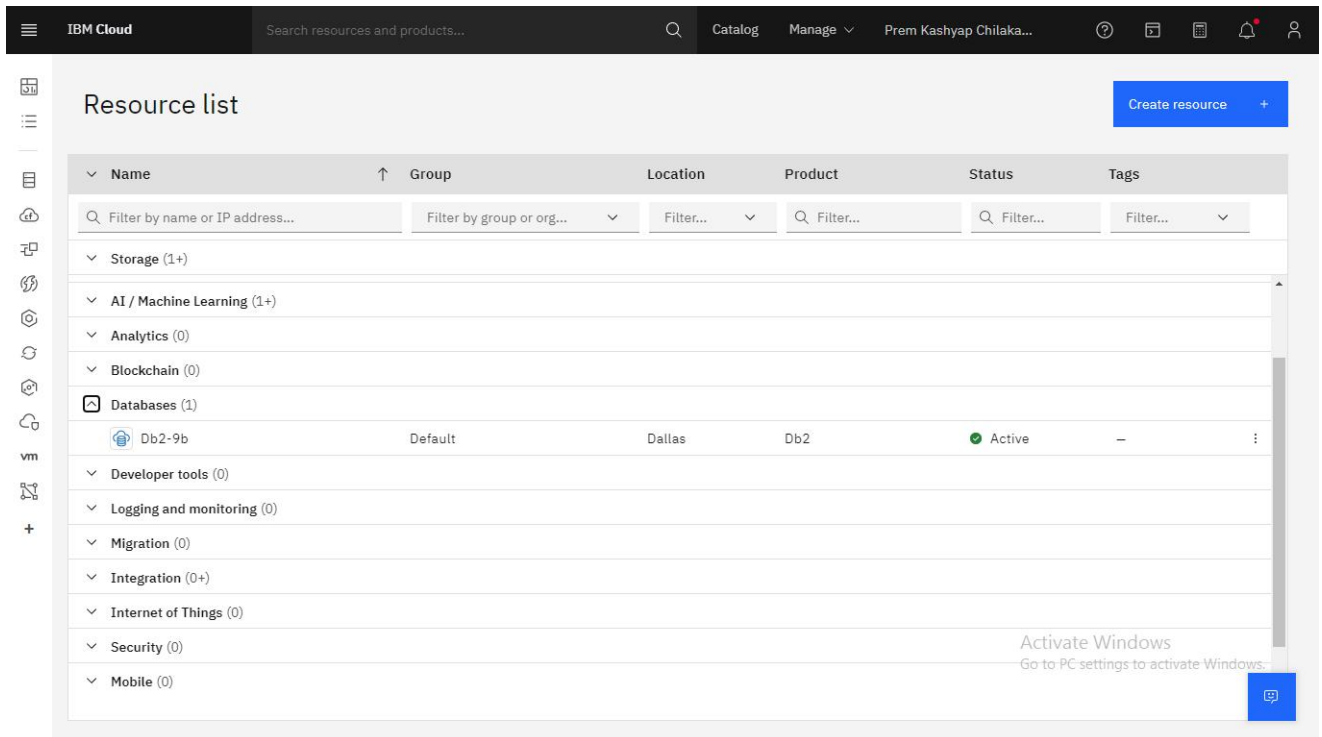


## IMPLEMENTING WEB APPLICATION

### Create UI to interact with application

Date	15 November 2022
Team ID	PNT2022TMID47207
Project Name	Personal Expense Tracker Application

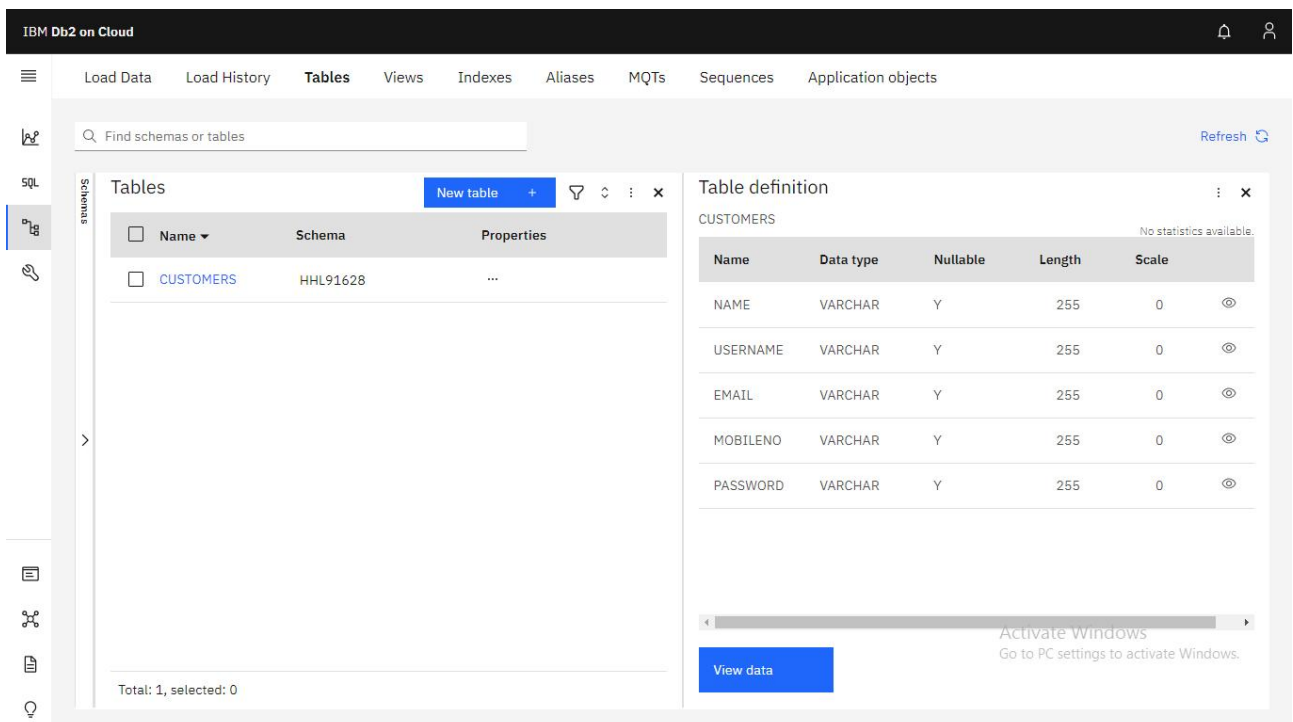
### DATABASE :



The screenshot shows the IBM Cloud console interface. At the top, there's a navigation bar with the IBM Cloud logo, a search bar, and various icons. The main section is titled "Resource list" and features a table with columns: Name, Group, Location, Product, Status, and Tags. The table is filtered to show "Databases (1)" under the "Databases" group. The single entry is "Db2-9b" with a status of "Active". A sidebar on the left contains icons for different resource categories like Storage, AI / Machine Learning, Analytics, Blockchain, Databases, Developer tools, Logging and monitoring, Migration, Integration, Internet of Things, Security, and Mobile. A "Create resource" button is visible in the top right corner.

Name	Group	Location	Product	Status	Tags
Db2-9b	Default	Dallas	Db2	Active	-

### TABLE AND ITS DETAILS :



The screenshot displays the IBM Db2 on Cloud console. The top navigation bar includes "Load Data", "Load History", "Tables", "Views", "Indexes", "Aliases", "MQTs", "Sequences", and "Application objects". The "Tables" section is active, showing a list of tables under the "CUSTOMERS" schema. The "Table definition" panel on the right provides details for the "CUSTOMERS" table, including columns: NAME, USERNAME, EMAIL, MOBILENO, and PASSWORD, all of type VARCHAR with a length of 255 and a nullable status of Y. A "View data" button is located at the bottom of the table definition panel. The sidebar on the left shows the database structure with "CUSTOMERS" selected.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
USERNAME	VARCHAR	Y	255	0
EMAIL	VARCHAR	Y	255	0
MOBILENO	VARCHAR	Y	255	0
PASSWORD	VARCHAR	Y	255	0

## TABLE DATA:

IBM Db2 on Cloud					
	Load Data	Load History	Tables	Views	Indexes
				Aliases	MQTs
					Sequences
					Application objects
HHL91628.CUSTOMERS					Back
					Export to CSV
NAME	USERNAME	EMAIL	MOBILENO	PASSWORD	
Jhon	Jhon	Jhon@gmail.com	9857648672	Jhon@32424	
Johnff	Johnff	hello	095026	cgtj	
Johnff	Johnff	hello@sgaasdf	095026	sdafsf	
Johnff	Johnff	hello@sga	095026	sdfa	
Johnfffasd	Johnfffasd	hello@sgaasdfsdf	095026	sdfg	
Prem Kashyap	Prem	iamprem@gmail.com	8789806965	Prem@3456	
Prem Kashyap	Prem	nmani3008@gmail.com	0985764867	1234	
Wick	Wick	wick2@gmail.com	0985764867	wick@123	
ghfsd	ghfsd	dfgs	dsgfd	dsgf	

## CODE TO CONNECT DATABASE :

app.py:

```
import ibm_db
from flask import Flask, flash, redirect, render_template, request, url_for

app = Flask(__name__)
app.debug=True

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hhl91628;PWD=vM4lAZxo4LsBPoJ","","")

@app.route("/")
def home():
    return render_template("home.html")

@app.route("/signup", methods=('GET', 'POST'))
def signup():
    if request.method == 'POST':
        name = request.form['fname']
        username =name.split(" ")[0]
        email = request.form['femail']
        mobile = request.form['mobile']
        password = request.form['password']

        sql = "SELECT * FROM customers WHERE username =? and email=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

```

    if account:
        return render_template('registered.html')
    else:
        insert_sql = "INSERT INTO customers VALUES (?, ?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, name)
        ibm_db.bind_param(prepare_stmt, 2, username)
        ibm_db.bind_param(prepare_stmt, 3, email)
        ibm_db.bind_param(prepare_stmt, 4, mobile)
        ibm_db.bind_param(prepare_stmt, 5, password)
        ibm_db.execute(prepare_stmt)
        return render_template('success.html')

return render_template('signup.html')

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

@app.route("/signin", methods=('GET', 'POST'))
def signin():
    if request.method == 'POST':
        femail = request.form['femail']
        fpassword = request.form['password']

        sql = "SELECT * FROM customers WHERE email =? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, femail)
        ibm_db.bind_param(stmt, 2, fpassword)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('loginsuccess.html')
        else:
            return render_template('loginfailure.html')

return render_template('signin.html')

if __name__ == '__main__':
    app.run()

```