SPRINT-2

1. Create Database in IBM DB2 for the Application

Schemas

Name	Type	Tables
ZPJ04244	User	6

Tables

New table +

Name	Schema	Properties
ACCOUNT	ZPJ04244	···
CATEGORY	ZPJ04244	···
ORDER	ZPJ04244	···
ORDERDETAILS	ZPJ04244	···
PRODUCT	ZPJ04244	···
USERS	ZPJ04244	···

2. Insert Data into Respective Table.

ZPJ04244.CATEGORY

Back

Export to CSV

CATEGORYID	CATEGORYNAME
1	Shirts
2	Pants
3	TShirts
4	Watches
5	Shoes

Tables

New table +

Name	Schema	Properties
ACCOUNT	ZPJ04244	...
CATEGORY	ZPJ04244	...
ORDER	ZPJ04244	...
ORDERDETAILS	ZPJ04244	...
PRODUCT	ZPJ04244	...
USERS	ZPJ04244	...

Total: 6, selected: 0

Table definition

PRODUCT

Approximate 9 rows (32.0 KB)
Updated on 2022-11-15 05:58:39

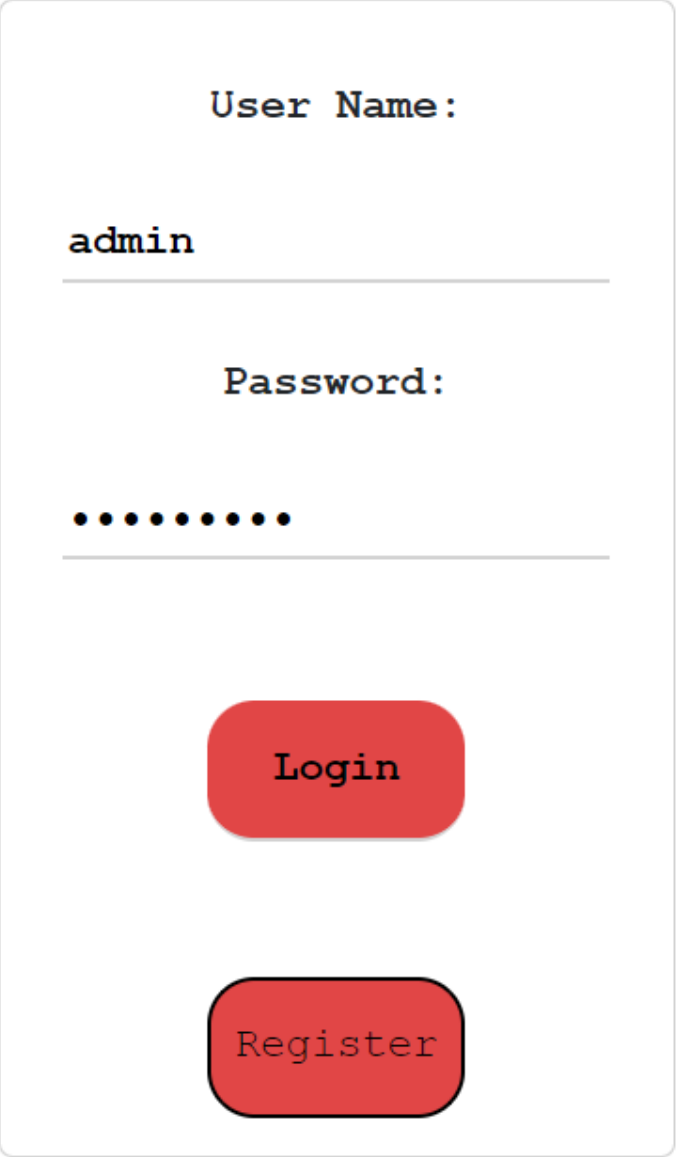
Name	Data type	Nullable	Length	Scale
PRODUCTID	INTEGER	N		0
PRODUCTNAME	VARCHAR	N	255	0
CATEGORYID	INTEGER	Y		0
UNIT	INTEGER	Y		0
PRICE	INTEGER	Y		0

View data

Export to CSV

PRODUCTID	PRODUCTNAME	CATEGORYID	UNIT	PRICE
1	Peter England Shirt	1	1	1500
2	VanHausen Shirt	1	1	1000
3	Otto Shirt	1	1	700
4	Peter England Pant	2	1	1600
5	Raymond Pant	2	1	1800
6	Basics Pant	2	1	800
7	Tommy Hilfiger TShirts	3	1	2200
8	Lewis TShirts	3	1	2500
9	Puma TShirts	3	1	2000

3. Create an Login Page for Admin Side .



Admin.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
/6.2.0/css/all.min.css">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/c
ss/bootstrap.min.css" rel="stylesheet"
integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV
8i59U5AR6csBvApHHNl/vI1Bx" crossorigin="anonymous">

    <title>Admin</title>
    <style>
       #response
        {
            margin-top: 50px;
            text-align:center;
            margin-left: 550px;
            font-family: 'Courier New', Courier,
monospace;
            font-weight: bolder;
            font-size: large;
            margin-top: 10px;
            margin-bottom: 10px;
      }
      #response1
        {
            margin-top: 50px;
            text-align:center;
            margin-left: 450px;
            font-family: 'Courier New', Courier,
monospace;
            font-weight: bolder;
            font-size: large;
            margin-top: 10px;
            margin-bottom: 10px;
      }
```

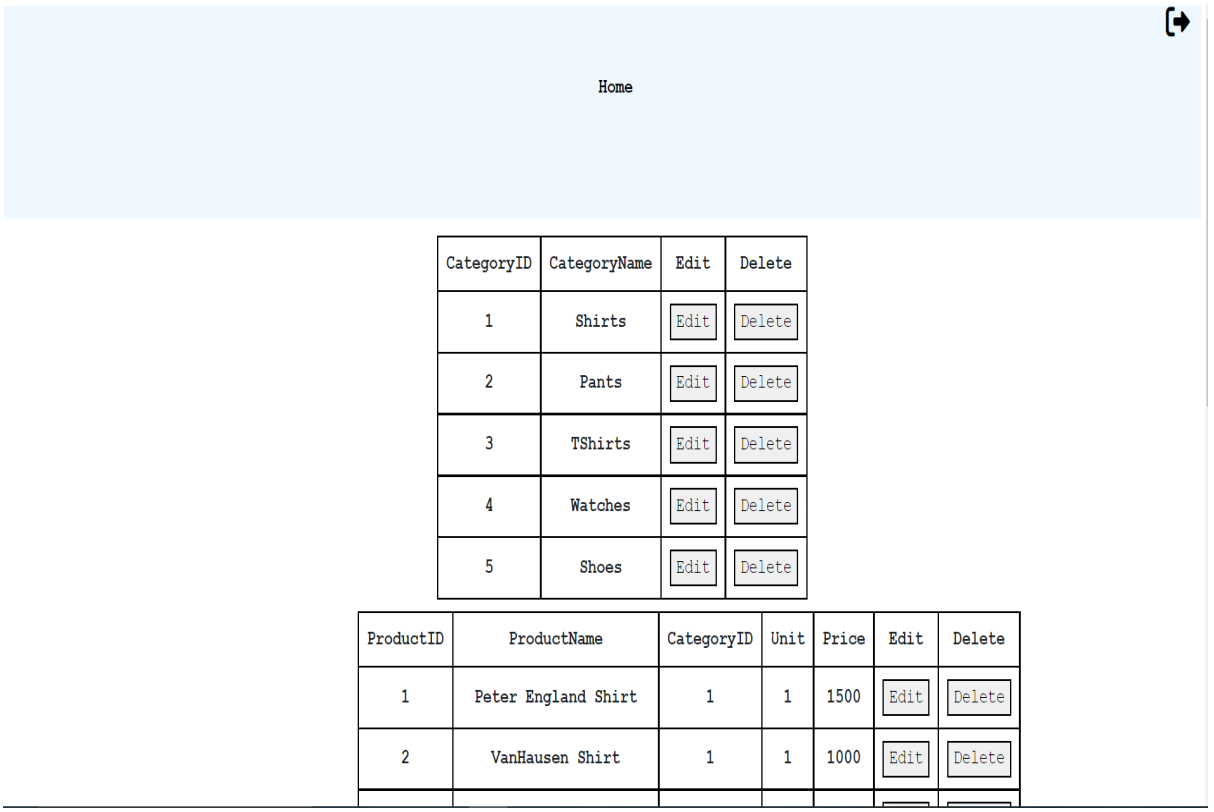
```
    #table1,th,td,tr
    {
       table-layout: fixed;
       border:2px solid black;
       padding: 10px;
    }
    a{
      text-decoration: none;
      color: black;
     }
  li{
      display: inline;
      margin:20px;

  }
  ul{
      margin-top: 60px;
  }
  #navbar
  {
      text-align: center;
      border:5px solid white;
      background-color: aliceblue;
      width: 100%;
      height:200px;
      font-family: 'Courier New', Courier, monospace;
      font-size: large;
      font-weight: bold;

  }
  #log
      {
          border: none;
          background-color: transparent;
          width: 50px;
          align: right;
          float: right;
```

```
            margin-right: 5px;
        }
    </style>
</head>
<body>
   <nav>
      <div id="navbar">
         <div id="log"><a class="nav-link"
href="{{url_for('logout')}}"><i class="fa-solid
fa-right-from-bracket"
style="font-size:30px;color:black"></i></a></div>
          <ul>
              <li><a
href="url_for('hello')">Home</a></li>
          </ul>
      </div>
  </nav>
  <div id="response">
   <table border = 1>
      <thead>
         <td>CategoryID</td>
         <td>CategoryName</td>
         <td>Edit</td>
         <td>Delete</td>
      </thead>

      {% for row in category %}
         <tr>
            <td>{{row["CATEGORYID"]}}</td>
            <td>{{row["CATEGORYNAME"]}}</td>
            <td><button><a
href="/edit/{{'CATEGORY'}}/{{row['CATEGORYID']}}">Edit</a
></button></td>
            <td><button><a
href="/delete/{{'CATEGORY'}}/{{row['CATEGORYID']}}">Delet
e</a></button></td>
         </tr>
```

```
      {% endfor %}
   </table>
  </div>
  <div id="response1">
   <table border = 1>
      <thead>
         <td>ProductID</td>
         <td>ProductName</td>
         <td>CategoryID</td>
         <td>Unit</td>
         <td>Price</td>
         <td>Edit</td>
         <td>Delete</td>
      </thead>

      {% for row in prods %}
         <tr>
            <td>{{row["PRODUCTID"]}}</td>
            <td>{{row["PRODUCTNAME"]}}</td>
            <td>{{row["CATEGORYID"]}}</td>
            <td>{{row["UNIT"]}}</td>
            <td>{{row["PRICE"]}}</td>
            <td><button><a
href="/edit/{{'PRODUCT'}}/{{row['PRODUCTID']}}">Edit</a><
/button></td>
            <td><button><a
href="/delete/{{'PRODUCT'}}/{{row['PRODUCTID']}}">Delete<
/a></button></td>
         </tr>
      {% endfor %}
   </table>
  </div>

     <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/d
ist/umd/popper.min.js"
integrity="sha384-Xe+8cL9oJa6tN/veChSP7q+mnSPaj5Bcu9mPX5F
```

```
5xIGE0DVittaqT5lorf0EI7Vk"
crossorigin="anonymous"></script>
     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js
/bootstrap.min.js"
integrity="sha384-ODmDIVzN+pFdexxHEHFBQH3/9/vQ9uori45z4Jj
nFsRydbmQbmL5t1tQ0culUzyK"
crossorigin="anonymous"></script>

</body>
</html>
```

Screenshot of Admin.html



3	Otto Shirt	1	1	700	Edit	Delete
4	Peter England Pant	2	1	1600	Edit	Delete
5	Raymond Pant	2	1	1800	Edit	Delete
6	Basics Pant	2	1	800	Edit	Delete
7	Tommy Hilfiger TShirts	3	1	2200	Edit	Delete
8	Lewis TShirts	3	1	2500	Edit	Delete
9	Puma TShirts	3	1	2000	Edit	Delete
10	Timex Watch	4	1	2000	Edit	Delete
11	Titan Watch	4	1	3500	Edit	Delete
12	Casio GShock Watch	4	1	5000	Edit	Delete
13	Sparx Shoes	5	1	2000	Edit	Delete
14	Nike Shoes	5	1	4500	Edit	Delete
15	Puma Shoes	5	1	4000	Edit	Delete

App.py:

```
import sqlite3 as sql

from flask import (Flask, jsonify, redirect,
render_template, request, session,
                   url_for,flash)
from flask_sqlalchemy import SQLAlchemy
from markupsafe import escape
import ibm_db

app= Flask(__name__)
app.config['DEBUG']=True
app.secret_key="mgyftuiu"
connection=ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc18
29-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.datab
ases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCer
tificate=DigiCertGlobalRootCA.crt;UID=zpj04244;PWD=xB9fQa
RJsweFh5pv","","")
```

```
@app.route('/home')
def hello():
     if 'loggedin' in session and 'name' in session:
        return render_template('home.html',msg="TRUE")
     return redirect(url_for('login'))

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

@app.route("/blog")
def blog():
    return "<h1>Hello World from blog page</h1>"

@app.route("/blog/<int:id>")
def blogId(id):
    return "<h1>Hello World from blog page"
+str(id)+"</h1>"

@app.route('/about')
def about():
    if 'loggedin' in session and 'name' in session:
        return render_template('about.html')
    return redirect(url_for('login'))


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

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

@app.route("/profiles/<username>")
def user(username):
    return "<h2>Hello "+username+"</h2>"
```

```
@app.route('/adduser',methods= ['POST','GET'])
def adduser():
    message=""
    if request.method=='POST':
        try:
            name= request.form['username']
            mail= request.form['mail']
            mobile= request.form['mobile']
            password= request.form['password']

            # with sql.connect("database.db") as con:
            #     cur= con.cursor()
            #     cur.execute('insert into user
values(?,?,?,?)',(name,mail,mobile,password))
            #     con.commit()
            #     message="User Registered"
            #     return redirect(url_for('login'))
            sql = "SELECT * FROM ACCOUNT WHERE USERNAME
=?"
            stmt = ibm_db.prepare(connection, sql)
            ibm_db.bind_param(stmt,1,name)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)

            if account:
                return render_template('list.html',
msg="Account Already Exists")
            else:
                command="INSERT INTO ACCOUNT
(USERNAME,EMAIL,PHONENO,PASSWORD) VALUES (?,?,?,?)"
                prep_stmt = ibm_db.prepare(connection,
command)
                ibm_db.bind_param(prep_stmt, 1, name)
                ibm_db.bind_param(prep_stmt, 2, mail)
                ibm_db.bind_param(prep_stmt, 3, mobile)
                ibm_db.bind_param(prep_stmt, 4, password)
```

```
                ibm_db.execute(prep_stmt)
                return render_template('login.html',
msg="TRUE")

        except:
            con.rollback()
            message="error"
            return
render_template("signup.html",error="TRUE")
    # return jsonify({message})
    # return render_template('signup.html')

@app.route('/admin')
def admin():
    data=[]
    products=[]
    if 'loggedin' in session and 'name' in session:
        sql="SELECT * FROM CATEGORY"
        stmt = ibm_db.prepare(connection, sql)
        ibm_db.execute(stmt)
        resultSet = ibm_db.fetch_both(stmt)
        while resultSet !=False:
            data.append(resultSet)
            resultSet=ibm_db.fetch_both(stmt)
        command="SELECT * FROM PRODUCT"
        stmt1 = ibm_db.prepare(connection, command)
        ibm_db.execute(stmt1)
        results = ibm_db.fetch_both(stmt1)
        while results !=False:
            products.append(results)
            results=ibm_db.fetch_both(stmt1)
        return
render_template('admin.html',category=data,prods=products
)
    return render_template('login.html')

@app.route("/check", methods=['POST','GET'])
```

```
def check():
    msg=""
    if request.method=='POST':
        username = request.form['username']
        password = request.form['password']

        account=""
        # con=sql.connect("database.db")
        # con.row_factory=sql.Row

        # cur =con.cursor()
        # cur.execute("SELECT * FROM user")

        # # rows= cur.fetchall()
        # cur.execute('SELECT * FROM user where name=?
and password=?', (username,password))
        # account = cur.fetchone()
        try:
            if username.lower()=='admin' and
password.lower()=='admin@123':
                session['loggedin'] = True
                session['name'] = username.lower()
                return redirect(url_for('admin'))

            else:
                command= "SELECT * FROM ACCOUNT WHERE
USERNAME =? AND PASSWORD =?"
                stmt=ibm_db.prepare(connection,command)
                ibm_db.bind_param(stmt, 1, username)
                ibm_db.bind_param(stmt, 2, password)
                ibm_db.execute(stmt)
                result = ibm_db.fetch_assoc(stmt)
                while result !=False:
                    account=result['USERNAME']
                    result = ibm_db.fetch_assoc(stmt)
                if account:
                    session['loggedin'] = True
```

```
                    session['name'] = account
                    return redirect(url_for('hello'))
                else:
                    msg = "Incorrect username/password!"
                    return render_template('login.html',
msg=msg)
        except:
            return render_template('login.html',
msg="Account not found")


        #return render_template("list.html",rows=rows)

@app.route('/logout')
def logout():
   session.pop('loggedin', None)
   session.pop('name', None)
   return redirect(url_for('login'))


@app.route('/delete/<int:ID>')
def delete(name):
  sql = f"SELECT * FROM ? WHERE = ?"
  print(sql)
  stmt = ibm_db.exec_immediate(conn, sql)
  student = ibm_db.fetch_row(stmt)
  print ("The Name is : ",  student)
  if student:
    sql = f"DELETE FROM Students WHERE
name='{escape(ID)}'"
    print(sql)
    stmt = ibm_db.exec_immediate(conn, sql)

    students = []
    sql = "SELECT * FROM Students"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
```

```
    while dictionary != False:
      students.append(dictionary)
      dictionary = ibm_db.fetch_both(stmt)
    if students:
      return render_template("list.html", students =
students, msg="Delete successfully")
    return "success..."
@app.route('/edit/<int:ID>')
def edit(ID):
    return "<h2>Hello "+str(ID)+"</h2>"

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