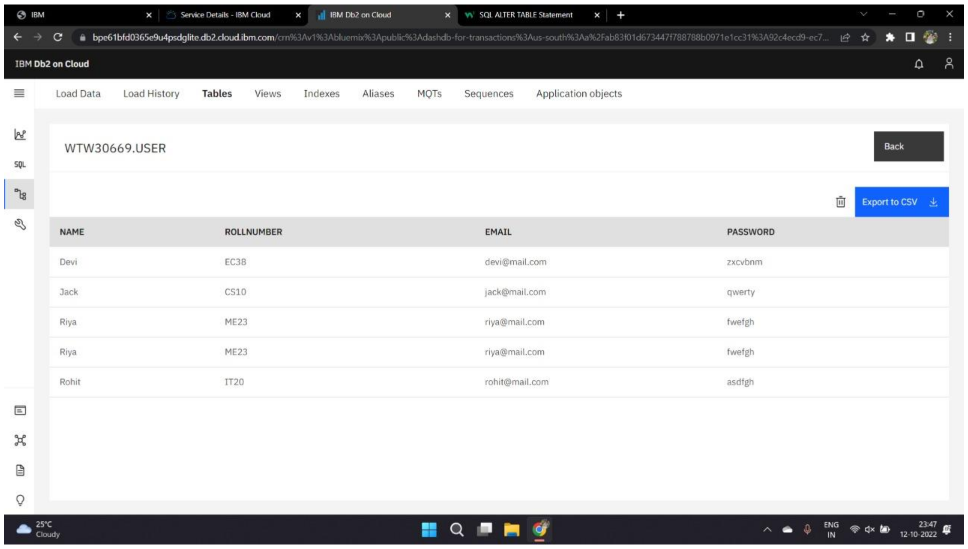
Assignment 2

Shanmathi M

1. Create User table with user with email, username, roll number password.

Inserted table

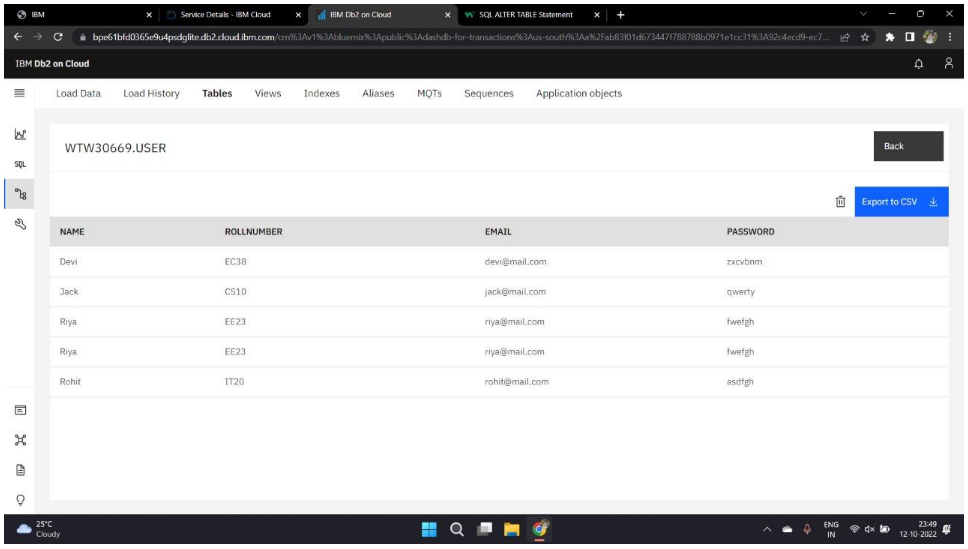


WTW30669.USER

NAME	ROLLNUMBER	EMAIL	PASSWORD
Devi	EC38	devi@mail.com	zxcvbnm
Jack	CS10	jack@mail.com	qwerty
Riya	ME23	riya@mail.com	fwefgh
Riya	ME23	riya@mail.com	fwefgh
Rohit	IT20	rohit@mail.com	asdfgh

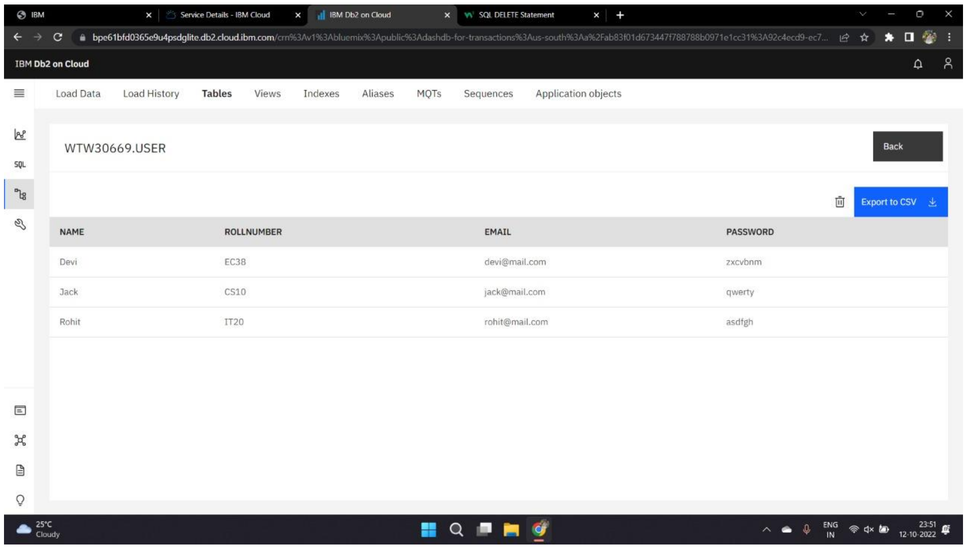
2. Perform UPDATE DELETE Queries with user table

After update and delete



WTW30669.USER

NAME	ROLLNUMBER	EMAIL	PASSWORD
Devi	EC38	devi@mail.com	zxcvbnm
Jack	CS10	jack@mail.com	qwerty
Riya	EE23	riya@mail.com	fwefgh
Riya	EE23	riya@mail.com	fwefgh
Rohit	IT20	rohit@mail.com	asdfgh



WTW30669.USER

NAME	ROLLNUMBER	EMAIL	PASSWORD
Devi	EC38	devi@mail.com	zxcvbnm
Jack	CS10	jack@mail.com	qwerty
Rohit	IT20	rohit@mail.com	asdfgh

SQL:

```
insert into user values('Devi', 'EC38','devi@mail.com', 'zxcvbnm');

insert into user values( 'Jack','CS10','jack@mail.com', 'qwerty');

insert into user values( 'Riya','ME23','riya@mail.com', 'fwefgh');

insert into user values( 'Riya','ME23','riya@mail.com', 'fwefgh');

insert into user values( 'Rohit','IT20','rohit@mail.com', 'asdfgh');


update user set rollnumber = 'EE23' where name = 'Riya'


delete from users where name = 'Riya';
```

3. Connect python code to db2.

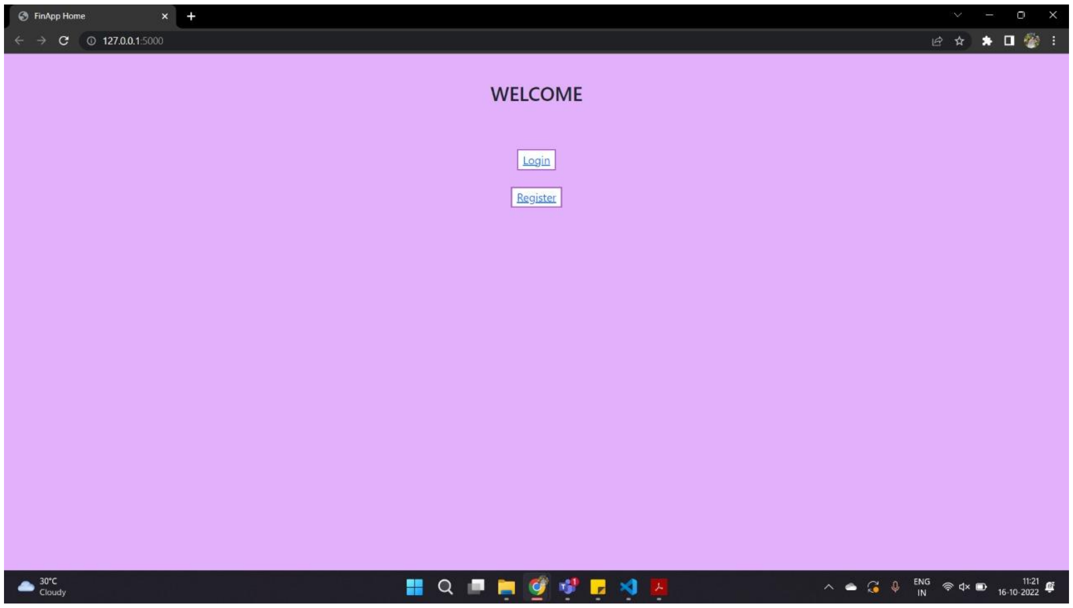
```
conn = ibm_db.connect(r"DATABASE=bludb;" r"HOSTNAME=b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
                      r"PORT=32716;"
                      r"SECURITY=SSL;"
                      r"SSLServerCertificate=C:\Users\shanm\Downloads\DigiCertGlobalRootCA.crt;"
                      r"UID=wtw30669;"
                      r"PWD=Q6BpxIorRdGrfsdD;", "", "")
```

4. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password, if the user is valid show the welcome page

App.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
app = Flask(__name__, template_folder='templates')
app.secret_key = 'Zenik'
conn = ibm_db.connect(r"DATABASE=bludb;" r"HOSTNAME=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;"
 r"PORT=32716;"
r"SECURITY=SSL;"
r"SSLServerCertificate=C:\Users\shanm\Downloads\DigiCertGlobalRootCA.crt;"
 r"UID=wtw30669;"
 r"PWD=Q6BpxIorRdGrfsdD;", "", "")

@app.route('/')
@app.route('/home')
def home():
 return render_template('home.html', title='Home', msg=" ")

@app.route('/dashboard')
def dashboard():
 sql = "SELECT * FROM USER WHERE name =?"
 stmt = ibm_db.prepare(conn, sql)
 ibm_db.bind_param(stmt, 1, session['name'])
 ibm_db.execute(stmt)
 account = ibm_db.fetch_assoc(stmt)
 return render_template('dashboard.html', title='Dashboard', account=account)

@app.route('/logout')
def logout():
 session.pop('Loggedin', None)
 session.pop('id', None)
 session.pop('name', None)
 return redirect('/')

@app.route('/login', methods=['GET', 'POST'])
def login():
 global userid
 msg = " "
 if request.method == "POST":
  name = request.form['name']
  password = request.form['password']
  sql = "SELECT * FROM USER WHERE name =? AND password =?"
  stmt = ibm_db.prepare(conn, sql)
  ibm_db.bind_param(stmt, 1, name)
  ibm_db.bind_param(stmt, 2, password)
  ibm_db.execute(stmt)
  account = ibm_db.fetch_assoc(stmt)
  print(account)
  if account:
   session['Loggedin'] = True
   session['id'] = account['NAME']
   userid = account['NAME']
   session['name'] = account['NAME']
   return redirect('/dashboard')
 else:
  msg = "Incorrect login credentials"
 return render_template('login.html', title='Login', msg=msg)

@app.route('/register', methods=['GET', 'POST'])
def register():
 msg = " "
 if request.method == "POST":
  name = request.form['name']
  rollnumber = request.form['rollnumber']
  email = request.form['email']
  password = request.form['password']
  password1 = request.form['password1']
  print(name)
  print(rollnumber)
  print(email)
  print(password)
  sql = "SELECT * FROM USER WHERE name =? or email=? "
  stmt = ibm_db.prepare(conn, sql)
  ibm_db.bind_param(stmt, 1, name)
  ibm_db.bind_param(stmt, 2, email)
  ibm_db.execute(stmt)
  account = ibm_db.fetch_assoc(stmt)
  print(account)
  if account:
   msg = "Account already exists"
  elif password1 != password:
   msg = "re-entered password doesnt match"
  elif not re.match(r'[A-Za-z0-9]+', name):
   msg = "Username should be only alphabets and numbers"
  else:
   sql = "INSERT INTO USER VALUES (?,?,?,?)"
   stmt = ibm_db.prepare(conn, sql)
   ibm_db.bind_param(stmt, 1, name)
   ibm_db.bind_param(stmt, 3, email)
   ibm_db.bind_param(stmt, 2, rollnumber)
   ibm_db.bind_param(stmt, 4, password)
   ibm_db.execute(stmt)
   return redirect('/login')
 return render_template('register.html', msg=msg, title="Register")
if __name__ == '__main__':


 app.run()
```

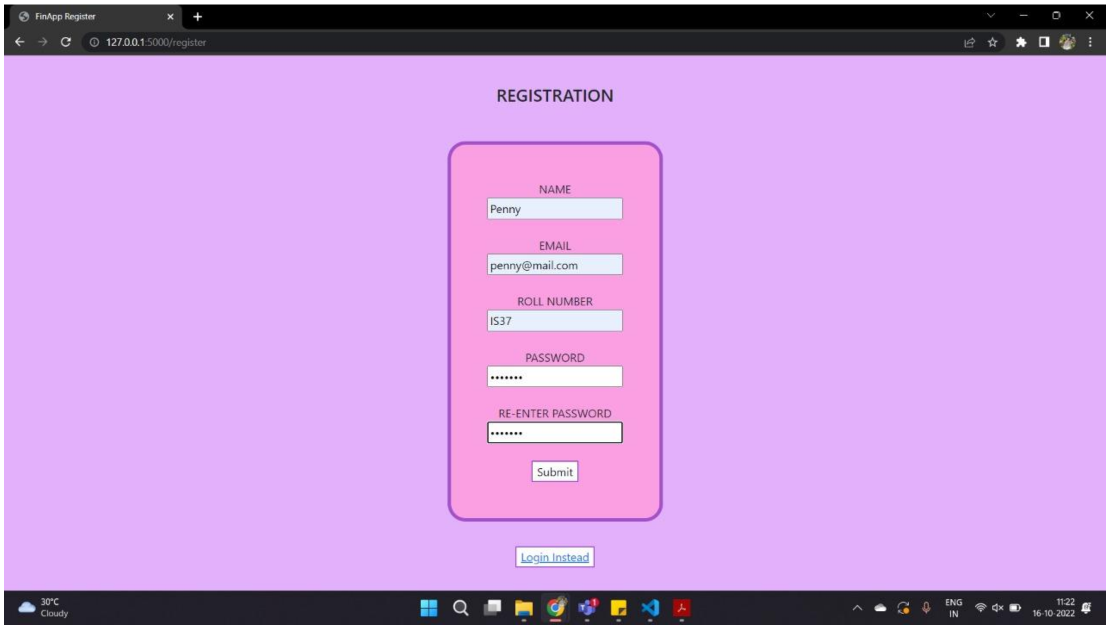
Respective html pages were created. (layout,home,login,registration,dashboard)



WELCOME

Login

Register



REGISTRATION

NAME
Penny

EMAIL
penny@mail.com

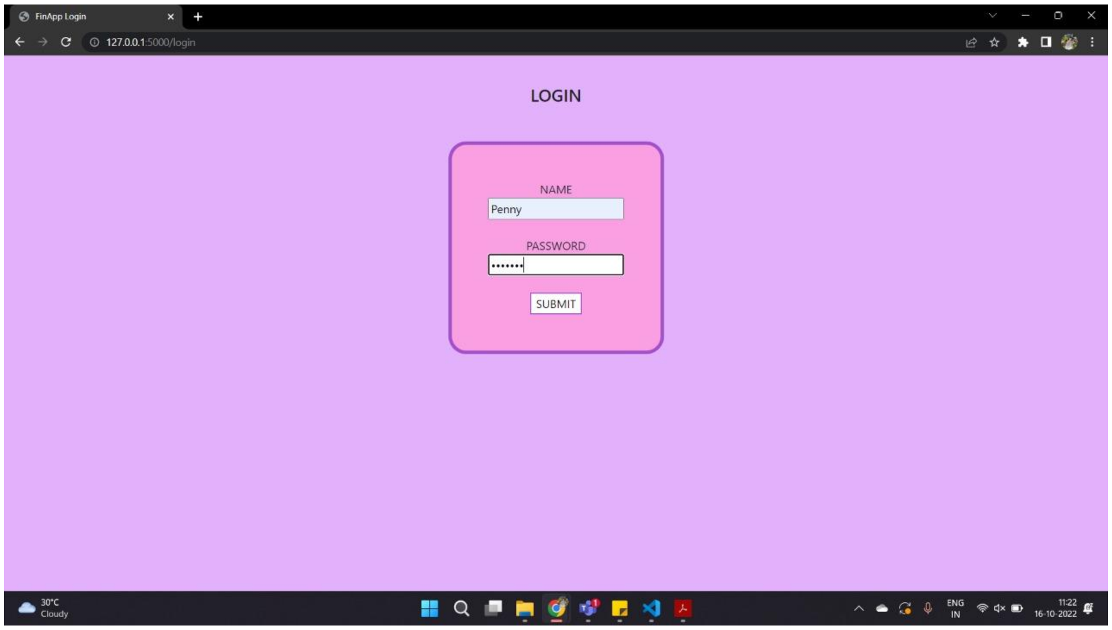
ROLL NUMBER
IS37

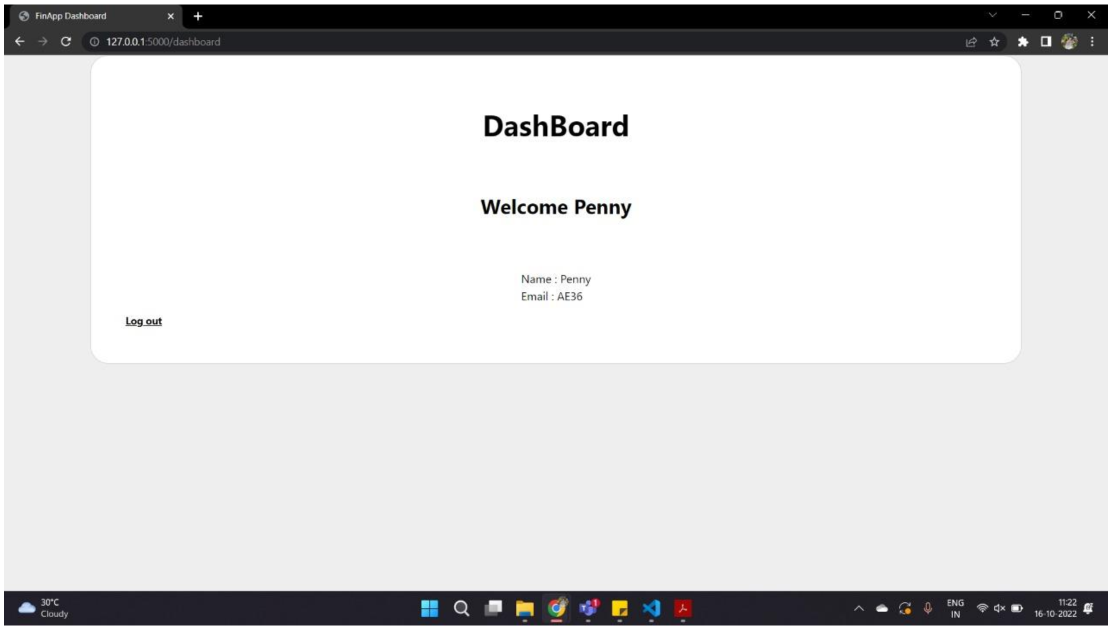
PASSWORD
•••••••

RE-ENTER PASSWORD
•••••••

Submit

Login Instead





Name : Penny
Email : AE36