Name: Himmaja Reddy BS

M14832991

**Cloud Project 2- AWS**

**Link: http://ec2-18-119-114-54.us-east-2.compute.amazonaws.com/login**

**Html Code for Login**

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/html" xmlns="http://www.w3.org/1999/html">

<head>

<meta charset="UTF-8">

<title>Login</title>

</head>

<body style="background-color:khaki";>

<div align="center">

<div><h1>**<i>**Login Page**</i>**</h1></div>

<div>

<form action="{{url\_for('login')}}" method="POST">

<div>{{ message }}</div> <br><br>

<label>**<b>**User Name :**</b>** </label><input id="username" name="username" type="text" placeholder="User Name" autocomplete="off"/> <br><br>

<label>**<b>**Password :**</b>** </label><input id="password" name="password" type="password" placeholder="Password" autocomplete="off"/> <br><br>

<button id="submit" name="submit">Sign in</button>

</form>

</div>

<p>**<b>**If you don't have an account? Please click**</b>** <a href="{{url\_for('registration')}}">**<b>**here**</b>**</a></p>

</div>

</body>

</html>

**Html code for Registration**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Register</title>

</head>

<body style="background-color:khaki";>

<div align="center">

<div><h1>**<i>**Registration Page**</i>**</h1></div>

<div>

<form action="{{url\_for('registration')}}" method="POST" enctype="multipart/form-data">

<div>{{ message }}</div><br><br>

<label>**<b>**User Name :**</b>** </label><input id="username" name="username" type="text" placeholder="User Name" autocomplete="off"/> <br><br>

<label>**<b>**Password :**</b>** </label><input id="password" name="password" type="password" placeholder="Password" autocomplete="off"/> <br><br>

<label>**<b>**First Name :**</b>** </label><input id="firstname" name="firstname" type="text" placeholder="First Name" autocomplete="off"/> <br><br>

<label>**<b>**Last Name :**</b>** </label><input id="lastname" name="lastname" type="text" placeholder="Last Name" autocomplete="off"/> <br><br>

<label>**<b>**Email : **</b>**</label><input id="email" name="email" type="text" placeholder="Email" autocomplete="off"/> <br><br>

<label>**<b>**Upload File :**</b>** </label><input type="file" id="textfile" name="textfile"><br><br>

<button id="submit" name="submit">Register</button>

</form>

</div>

<p>**<b>**If you already have an account? Please click**</b>** <a href="{{url\_for('login')}}">**<b>**here to login**</b>**</a></p>

</div>

</body>

</html>

import sqlite3

from flask import Flask, request, g, render\_template, send\_file

DATABASE = '/var/www/html/flaskapp/example.db'

app = Flask(\_\_name\_\_)

app.config.from\_object(\_\_name\_\_)

def connect\_to\_database():

return sqlite3.connect(app.config['DATABASE'])

def get\_db():

db = getattr(g, 'db', None)

if db is None:

db = g.db = connect\_to\_database()

return db

@app.teardown\_appcontext

def close\_connection(exception):

db = getattr(g, 'db', None)

if db is not None:

db.close()

def execute\_query(query, args=()):

cur = get\_db().execute(query, args)

rows = cur.fetchall()

cur.close()

return rows

def commit():

get\_db().commit()

@app.route("/")

def hello():

execute\_query("DROP TABLE IF EXISTS users")

execute\_query("CREATE TABLE users (Username text,Password text,firstname text, lastname text, email text, count integer)")

return render\_template('index.html')

@app.route('/login', methods =['POST', 'GET'])

def login():

message = ''

if request.method == 'POST' and str(request.form['username']) !="" and str(request.form['password']) != "":

username = str(request.form['username'])

password = str(request.form['password'])

result = execute\_query("""SELECT firstname,lastname,email,count FROM users WHERE Username = (?) AND Password = (?)""", (username, password ))

if result:

for row in result:

return responsePage(row[0], row[1], row[2], row[3])

else:

message = 'Invalid Credentials !'

elif request.method == 'POST':

message = 'Please enter Credentials'

return render\_template('index.html', message = message)

@app.route('/registration', methods =['GET', 'POST'])

def registration():

message = ''

if request.method == 'POST' and str(request.form['username']) !="" and str(request.form['password']) !="" and str(request.form['firstname']) !="" and str(request.form['lastname']) !="" and str(request.form['email']) !="":

username = str(request.form['username'])

password = str(request.form['password'])

firstname = str(request.form['firstname'])

lastname = str(request.form['lastname'])

email = str(request.form['email'])

uploaded\_file = request.files['textfile']

if not uploaded\_file:

filename = null

word\_count = null

else :

filename = uploaded\_file.filename

word\_count = getNumberOfWords(uploaded\_file)

result = execute\_query("""SELECT \* FROM users WHERE Username = (?)""", (username, ))

if result:

message = 'User has already registered!'

else:

result1 = execute\_query("""INSERT INTO users (username, password, firstname, lastname, email, count) values (?, ?, ?, ?, ?, ? )""", (username, password, firstname, lastname, email, word\_count, ))

commit()

result2 = execute\_query("""SELECT firstname,lastname,email,count FROM users WHERE Username = (?) AND Password = (?)""", (username, password ))

if result2:

for row in result2:

return responsePage(row[0], row[1], row[2], row[3])

elif request.method == 'POST':

message = 'Some of the fields are missing!'

return render\_template('registration.html', message = message)

@app.route("/download")

def download():

path = "Limerick.txt"

return send\_file(path, as\_attachment=True)

def getNumberOfWords(file):

data = file.read()

words = data.split()

return str(len(words))

def responsePage(firstname, lastname, email, count):

return """ First Name : """ + str(firstname) + """ <br> Last Name : """ + str(lastname) + """ <br> Email : """ + str(email) + """ <br> Word Count : """ + str(count) + """ <br><br> <a href="/download" >Download</a> """

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

execute\_query("CREATE TABLE users (Username text,Password text,firstname text, lastname text, email text, count integer)")