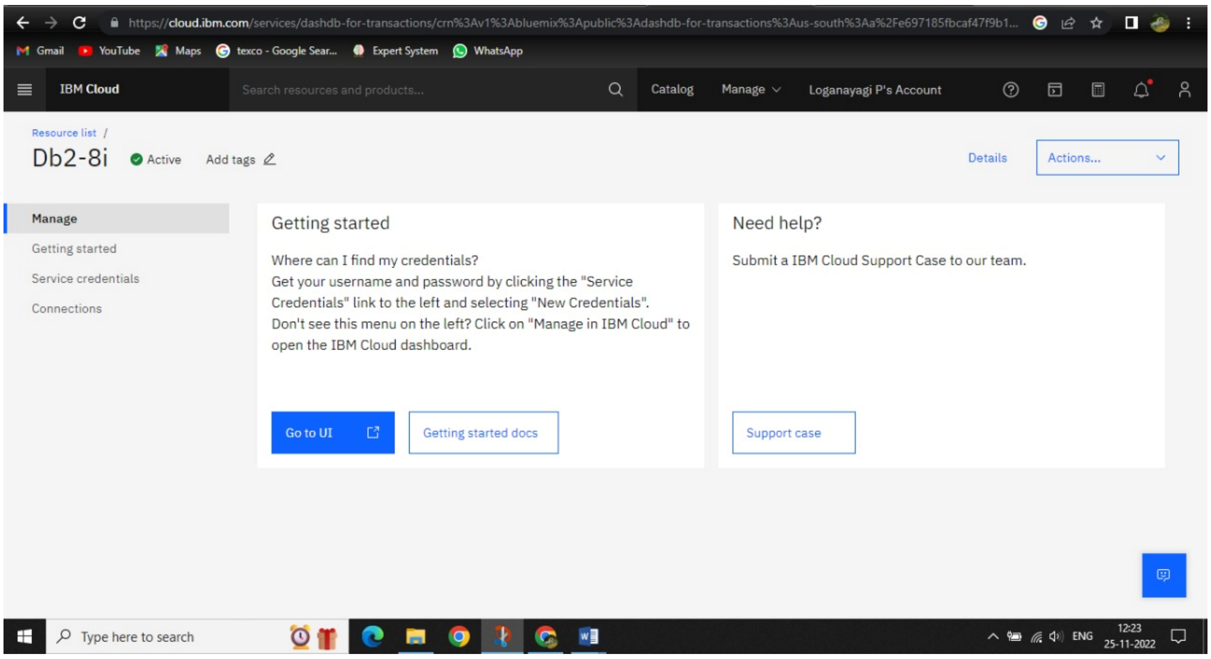
Create IBM DB2 and Connect with Python

Team ID	PNT2022TMID11592
Project Name	Project - Skill and Job Recommender



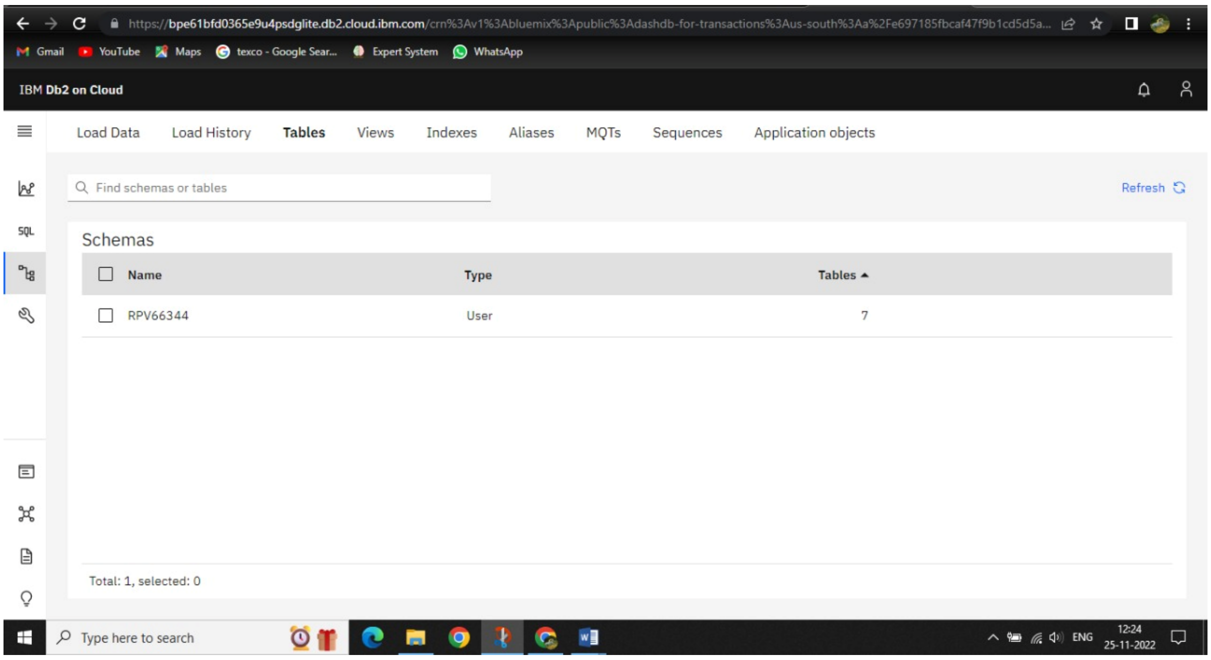
Db2-8i Active

Getting started

Where can I find my credentials?
Get your username and password by clicking the "Service Credentials" link to the left and selecting "New Credentials". Don't see this menu on the left? Click on "Manage in IBM Cloud" to open the IBM Cloud dashboard.

Need help?

Submit a IBM Cloud Support Case to our team.

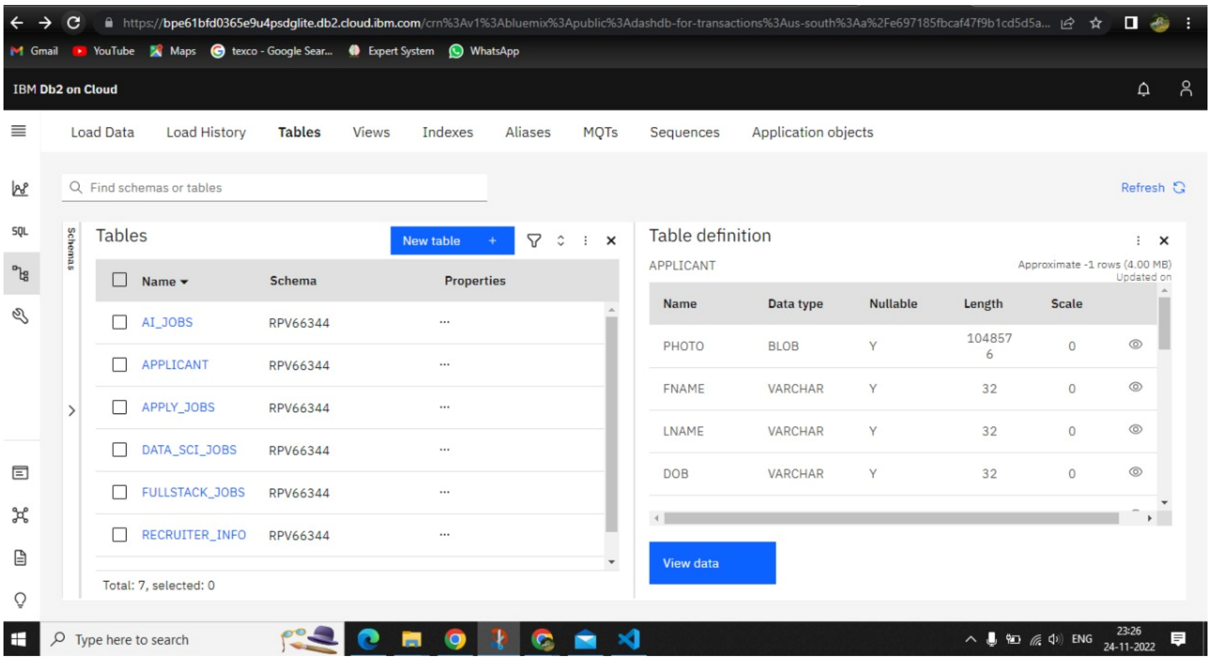


IBM Db2 on Cloud

Schemas

Name	Type	Tables
RPV66344	User	7

Total: 1, selected: 0


IBM Db2 on Cloud

Tables

Name	Schema	Properties
AI_JOBS	RPV66344	···
APPLICANT	RPV66344	···
APPLY_JOBS	RPV66344	···
DATA_SCI_JOBS	RPV66344	···
FULLSTACK_JOBS	RPV66344	···
RECRUITER_INFO	RPV66344	···

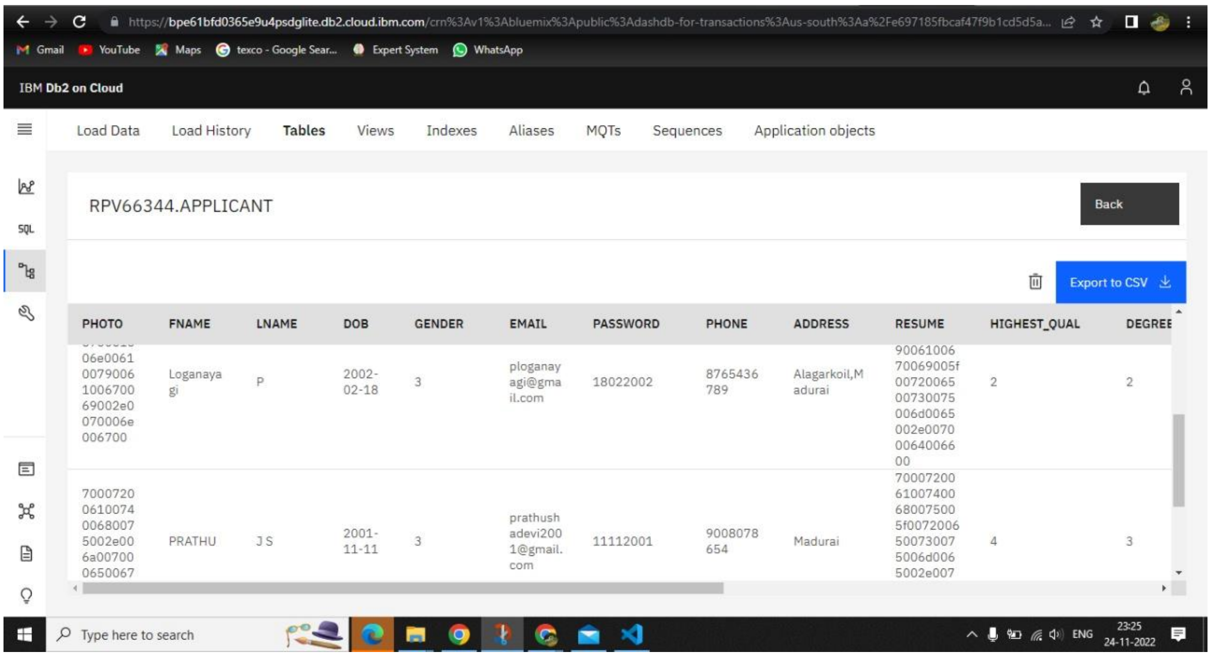
Total: 7, selected: 0

Table definition

APPLICANT

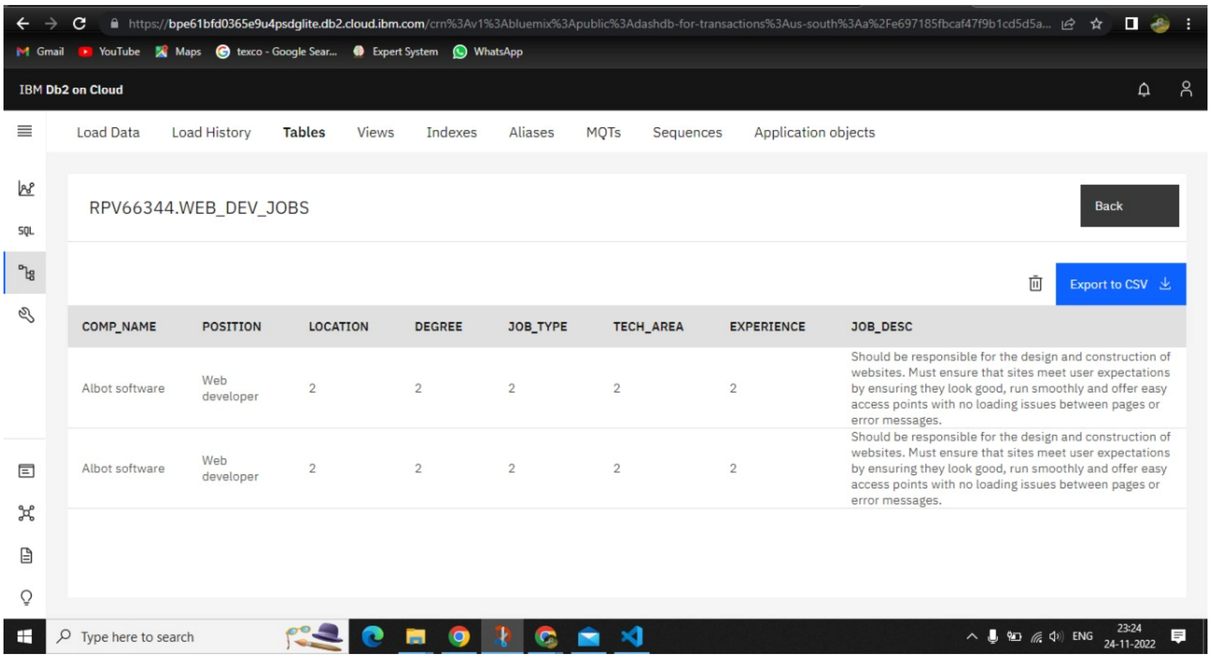
Approximate -1 rows (4.00 MB)

Name	Data type	Nullable	Length	Scale
PHOTO	BLOB	Y	1048576	0
FNAME	VARCHAR	Y	32	0
LNAME	VARCHAR	Y	32	0
DOB	VARCHAR	Y	32	0


IBM Db2 on Cloud

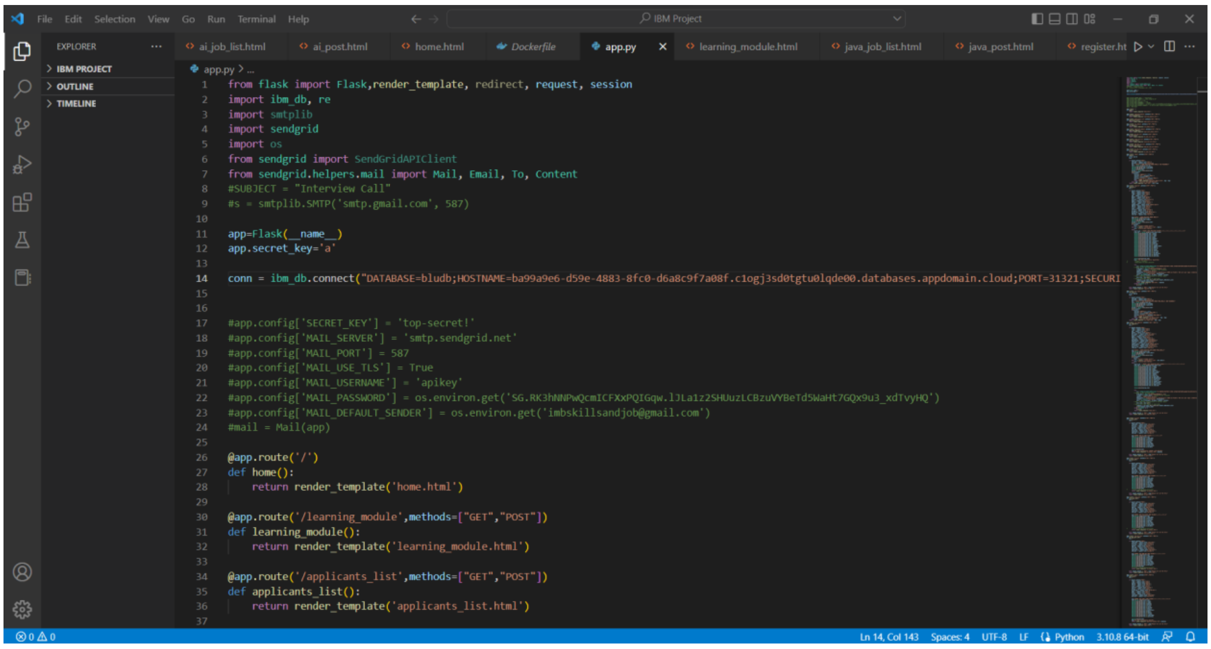
RPV66344.APPLICANT

PHOTO	FNAME	LNAME	DOB	GENDER	EMAIL	PASSWORD	PHONE	ADDRESS	RESUME	HIGHEST_QUAL	DEGREE
06e0061 0079006 1006700 69002e0 070006e 006700	Loganayagi	P	2002-02-18	3	ploganayagi@gmail.com	18022002	8765436789	Alagarkoil,Madurai	90061006 70069005f 00720065 00730075 006d0065 002e0070 00640066 00	2	2
7000720 0610074 0068007 5002e00 6a00700 0650067	PRATHU	J S	2001-11-11	3	prathushadevi2001@gmail.com	11112001	9008078654	Madurai	70007200 61007400 68007500 5f0072006 50073007 5006d006 5002e007	4	3

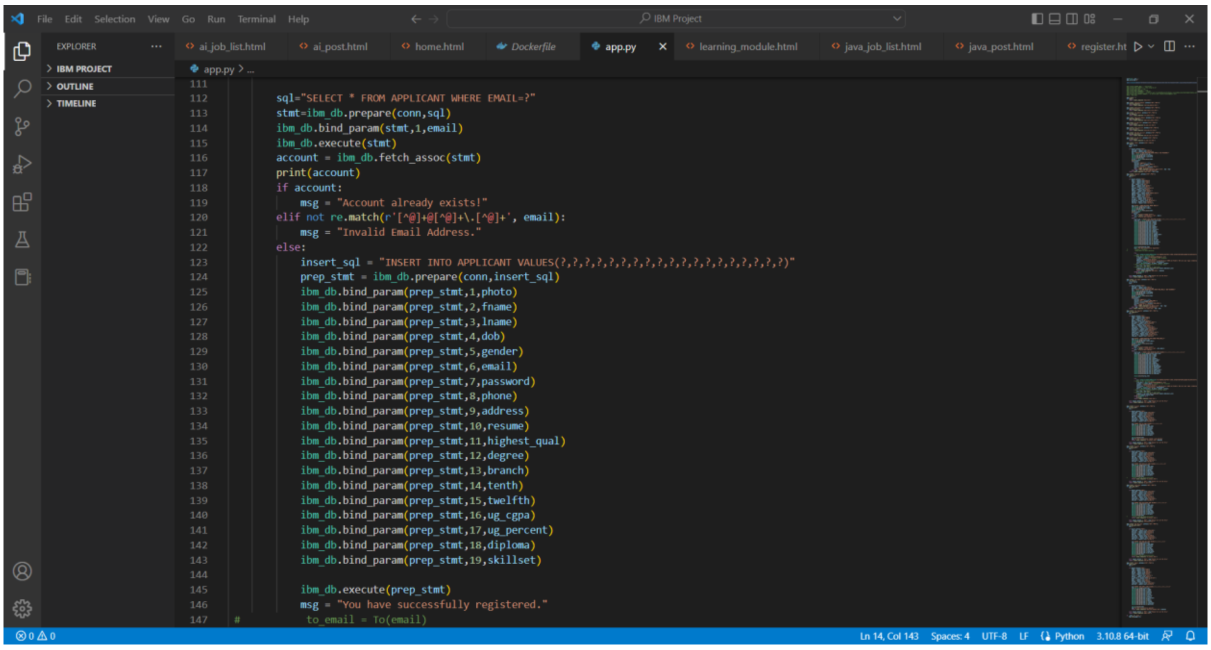


RPV66344.WEB_DEV_JOBS

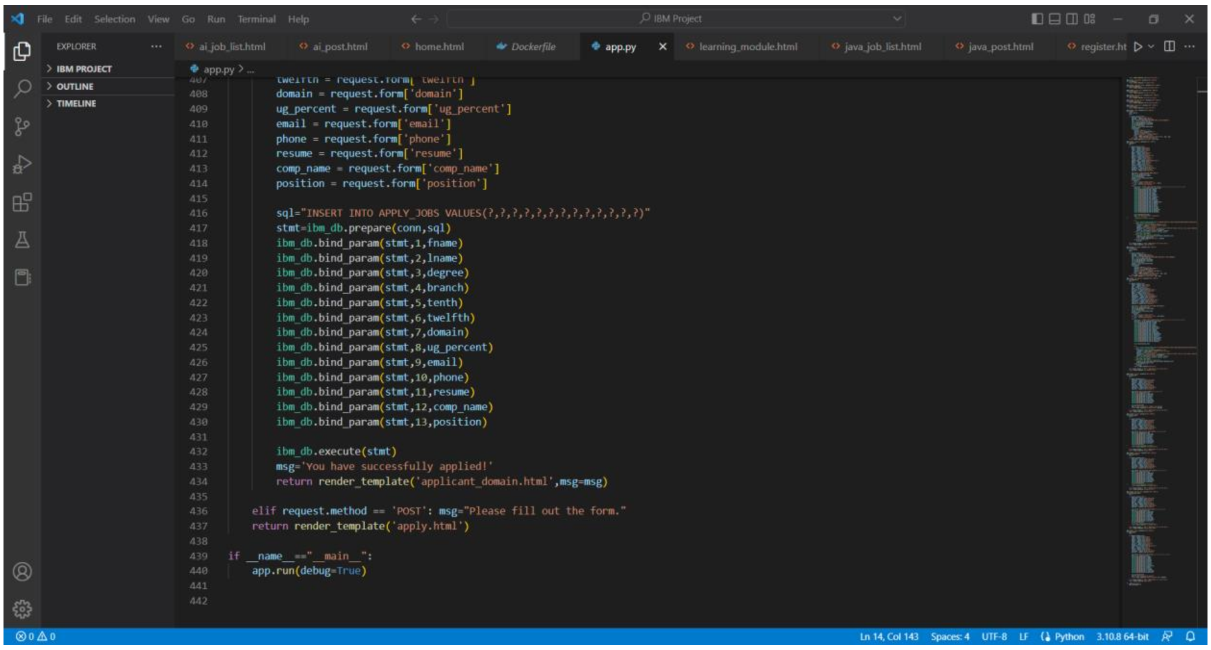
COMP_NAME	POSITION	LOCATION	DEGREE	JOB_TYPE	TECH_AREA	EXPERIENCE	JOB_DESC
Albot software	Web developer	2	2	2	2	2	Should be responsible for the design and construction of websites. Must ensure that sites meet user expectations by ensuring they look good, run smoothly and offer easy access points with no loading issues between pages or error messages.
Albot software	Web developer	2	2	2	2	2	Should be responsible for the design and construction of websites. Must ensure that sites meet user expectations by ensuring they look good, run smoothly and offer easy access points with no loading issues between pages or error messages.



```python
from flask import Flask,render_template, redirect, request, session
import ibm_db, re
import smtplib
import sendgrid
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail, Email, To, Content
#SUBJECT = "Interview Call"
#s = smtplib.SMTP('smtp.gmail.com', 587)

app=Flask(__name__)
app.secret_key='a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;SECURI


#app.config['SECRET_KEY'] = 'top-secret!'
#app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
#app.config['MAIL_PORT'] = 587
#app.config['MAIL_USE_TLS'] = True
#app.config['MAIL_USERNAME'] = 'apikey'
#app.config['MAIL_PASSWORD'] = os.environ.get('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')
#app.config['MAIL_DEFAULT_SENDER'] = os.environ.get('imbskillsandjob@gmail.com')
#mail = Mail(app)

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

@app.route('/learning_module',methods=["GET","POST"])
def learning_module():
    return render_template('learning_module.html')

@app.route('/applicants_list',methods=["GET","POST"])
def applicants_list():
    return render_template('applicants_list.html')

```



```python

        sql="SELECT * FROM APPLICANT WHERE EMAIL=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = "Account already exists!"
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = "Invalid Email Address."
        else:
            insert_sql = "INSERT INTO APPLICANT VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prep_stmt,1,photo)
            ibm_db.bind_param(prep_stmt,2,fname)
            ibm_db.bind_param(prep_stmt,3,lname)
            ibm_db.bind_param(prep_stmt,4,dob)
            ibm_db.bind_param(prep_stmt,5,gender)
            ibm_db.bind_param(prep_stmt,6,email)
            ibm_db.bind_param(prep_stmt,7,password)
            ibm_db.bind_param(prep_stmt,8,phone)
            ibm_db.bind_param(prep_stmt,9,address)
            ibm_db.bind_param(prep_stmt,10,resume)
            ibm_db.bind_param(prep_stmt,11,highest_qual)
            ibm_db.bind_param(prep_stmt,12,degree)
            ibm_db.bind_param(prep_stmt,13,branch)
            ibm_db.bind_param(prep_stmt,14,tenth)
            ibm_db.bind_param(prep_stmt,15,twelfth)
            ibm_db.bind_param(prep_stmt,16,ug_cgpa)
            ibm_db.bind_param(prep_stmt,17,ug_percent)
            ibm_db.bind_param(prep_stmt,18,diploma)
            ibm_db.bind_param(prep_stmt,19,skillset)

            ibm_db.execute(prep_stmt)
            msg = "You have successfully registered."
#            to_email = To(email)
```



```python
        twelfth = request.form['twelfth']
        domain = request.form['domain']
        ug_percent = request.form['ug_percent']
        email = request.form['email']
        phone = request.form['phone']
        resume = request.form['resume']
        comp_name = request.form['comp_name']
        position = request.form['position']

        sql="INSERT INTO APPLY_JOBS VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,fname)
        ibm_db.bind_param(stmt,2,lname)
        ibm_db.bind_param(stmt,3,degree)
        ibm_db.bind_param(stmt,4,branch)
        ibm_db.bind_param(stmt,5,tenth)
        ibm_db.bind_param(stmt,6,twelfth)
        ibm_db.bind_param(stmt,7,domain)
        ibm_db.bind_param(stmt,8,ug_percent)
        ibm_db.bind_param(stmt,9,email)
        ibm_db.bind_param(stmt,10,phone)
        ibm_db.bind_param(stmt,11,resume)
        ibm_db.bind_param(stmt,12,comp_name)
        ibm_db.bind_param(stmt,13,position)

        ibm_db.execute(stmt)
        msg='You have successfully applied!'
        return render_template('applicant_domain.html',msg=msg)

    elif request.method == 'POST': msg="Please fill out the form."
    return render_template('apply.html')

if __name__=="__main__":
    app.run(debug=True)


```

V