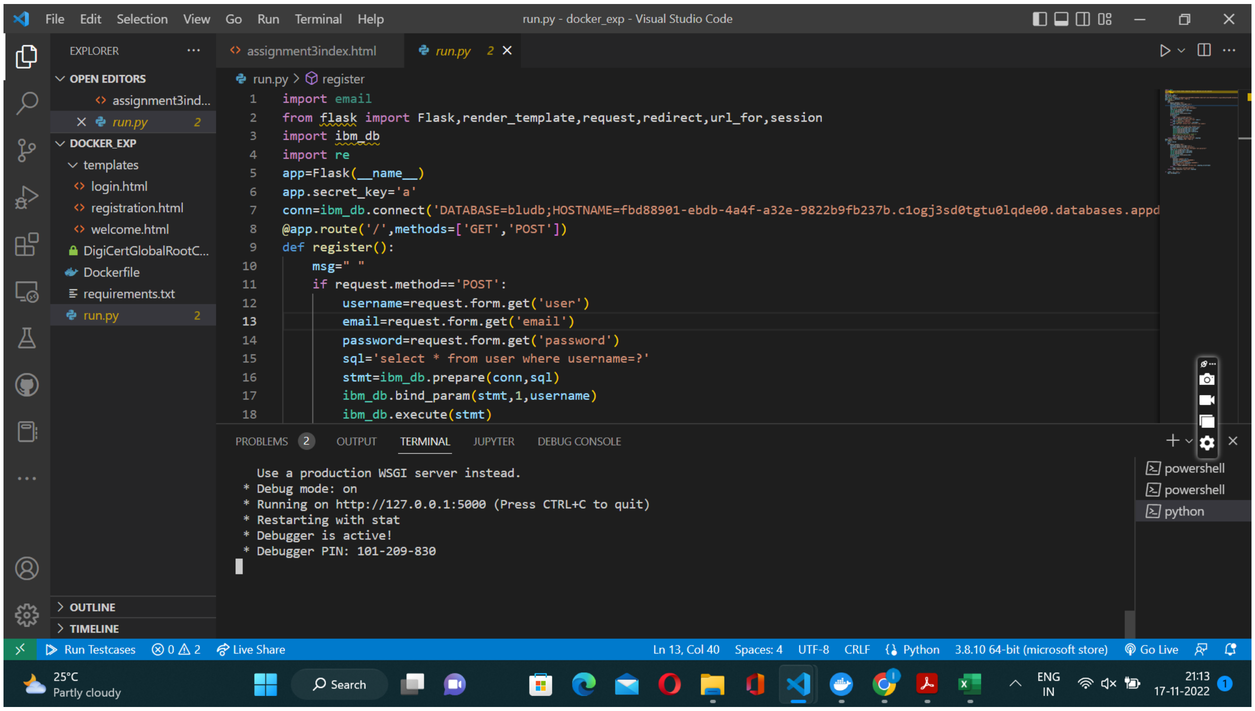
SETTING UP WEB APPLICATION

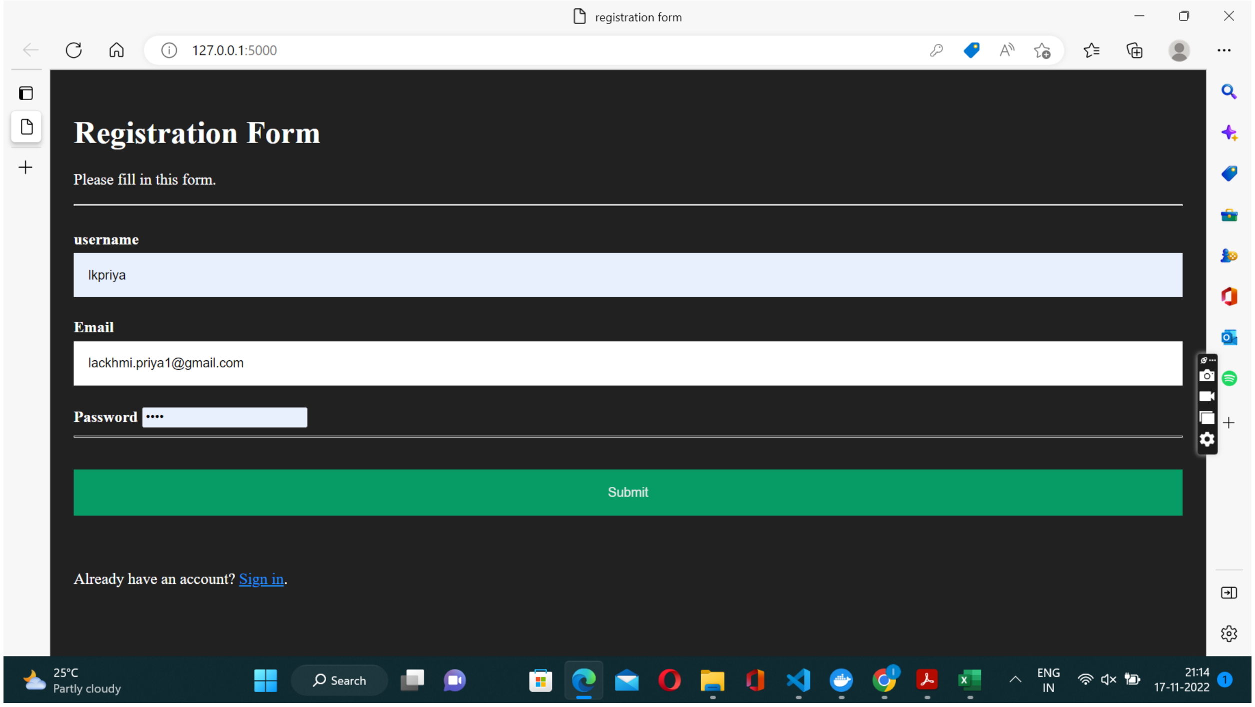
Team ID	PNT2022TMID18119
Project Name	Inventory management system for retailer

CREATE A UI TO INTERACT WITH APPLICATION



```python
import email
from flask import Flask,render_template,request,redirect,url_for,session
import ibm_db
import re
app=Flask(__name__)
app.secret_key='a'
conn=ibm_db.connect('DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appd
@app.route('/',methods=['GET','POST'])
def register():
    msg=" "
    if request.method=='POST':
        username=request.form.get('user')
        email=request.form.get('email')
        password=request.form.get('password')
        sql='select * from user where username=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
```

```
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 101-209-830
```



Registration Form

Please fill in this form.

username

lkpriya

Email

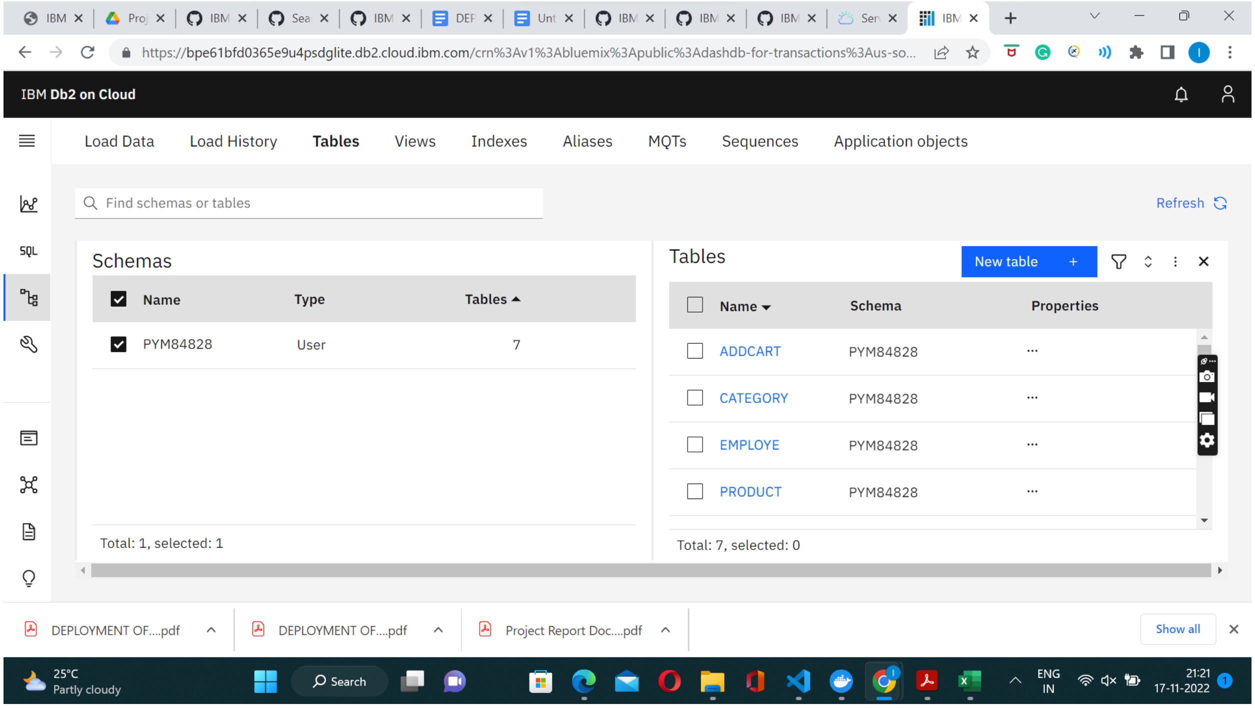
lackhmi.priya1@gmail.com

Password ••••

Submit

Already have an account? Sign in.

Create ibm db2 and connect it with python



IBM Db2 on Cloud

Load Data | Load History | Tables | Views | Indexes | Aliases | MQTs | Sequences | Application objects

Find schemas or tables

Refresh

Schemas

☑	Name	Type	Tables ▲
☑	PYM84828	User	7

Total: 1, selected: 1

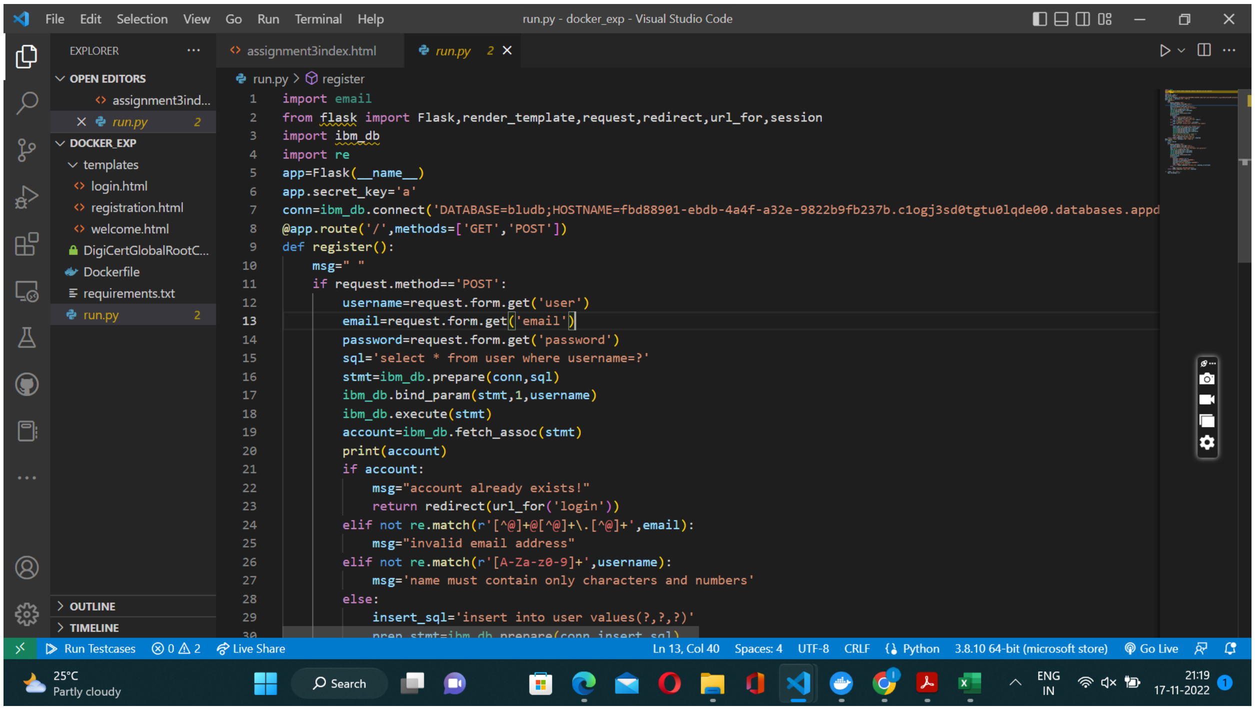
Tables

New table +

□	Name ▾	Schema	Properties
□	ADDCART	PYM84828	···
□	CATEGORY	PYM84828	···
□	EMPLOYE	PYM84828	···
□	PRODUCT	PYM84828	···

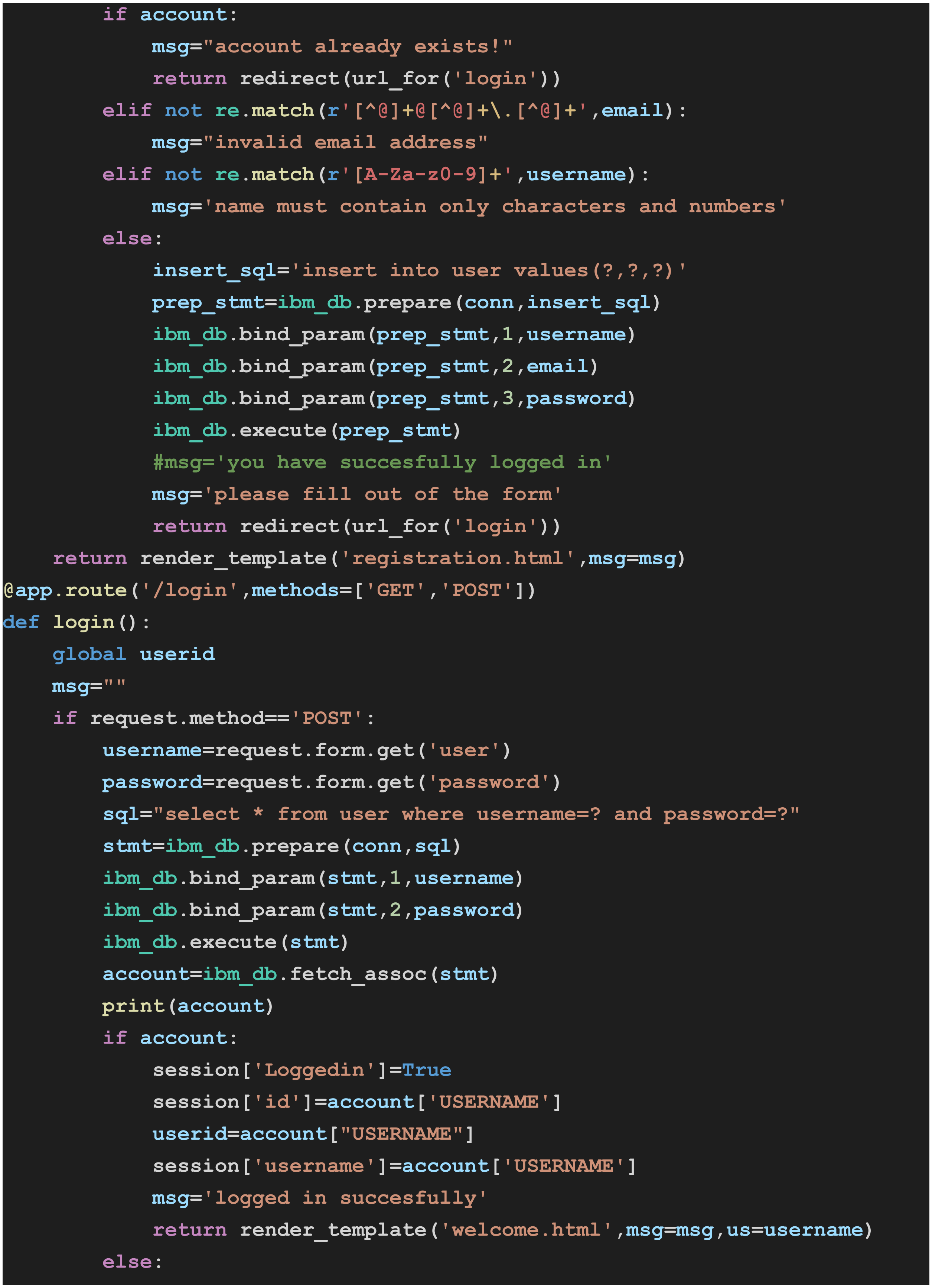
Total: 7, selected: 0

https://bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-so...
DEPLOYMENT OF....pdf
DEPLOYMENT OF....pdf
Project Report Doc....pdf
Show all
25°C
Partly cloudy
Search
ENG
IN
21:21
17-11-2022



```python
import email
from flask import Flask,render_template,request,redirect,url_for,session
import ibm_db
import re
app=Flask(__name__)
app.secret_key='a'
conn=ibm_db.connect('DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appd
@app.route('/',methods=['GET','POST'])
def register():
    msg=" "
    if request.method=='POST':
        username=request.form.get('user')
        email=request.form.get('email')
        password=request.form.get('password')
        sql='select * from user where username=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg="account already exists!"
            return redirect(url_for('login'))
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg="invalid email address"
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg='name must contain only characters and numbers'
        else:
            insert_sql='insert into user values(?,?,?)'
            prep_stmt=ibm_db.prepare(conn,insert_sql)
```

```python
import email
from flask import Flask,render_template,request,redirect,url_for,session
import ibm_db
import re
app=Flask(__name__)
app.secret_key='a'
conn=ibm_db.connect('DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32731;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA (1).crt;UID=mcb20916;PWD=iAPZMikUQf6agwXY','','')
@app.route('/',methods=['GET','POST'])
def register():
    msg=" "
    if request.method=='POST':
        username=request.form.get('user')
        email=request.form.get('email')
        password=request.form.get('password')
        sql='select * from user where username=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
```



```
        if account:
            msg="account already exists!"
            return redirect(url_for('login'))
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg="invalid email address"
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg='name must contain only characters and numbers'
        else:
            insert_sql='insert into user values(?,?,?)'
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prep_stmt,1,username)
            ibm_db.bind_param(prep_stmt,2,email)
            ibm_db.bind_param(prep_stmt,3,password)
            ibm_db.execute(prep_stmt)
            #msg='you have succesfully logged in'
            msg='please fill out of the form'
            return redirect(url_for('login'))
    return render_template('registration.html',msg=msg)
@app.route('/login',methods=['GET','POST'])
def login():
    global userid
    msg=""
    if request.method=='POST':
        username=request.form.get('user')
        password=request.form.get('password')
        sql="select * from user where username=? and password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        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['USERNAME']
            userid=account["USERNAME"]
            session['username']=account['USERNAME']
            msg='logged in succesfully'
            return render_template('welcome.html',msg=msg,us=username)
        else:
            msg='incorrect username password'
    return render_template('login.html',msg=msg)

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