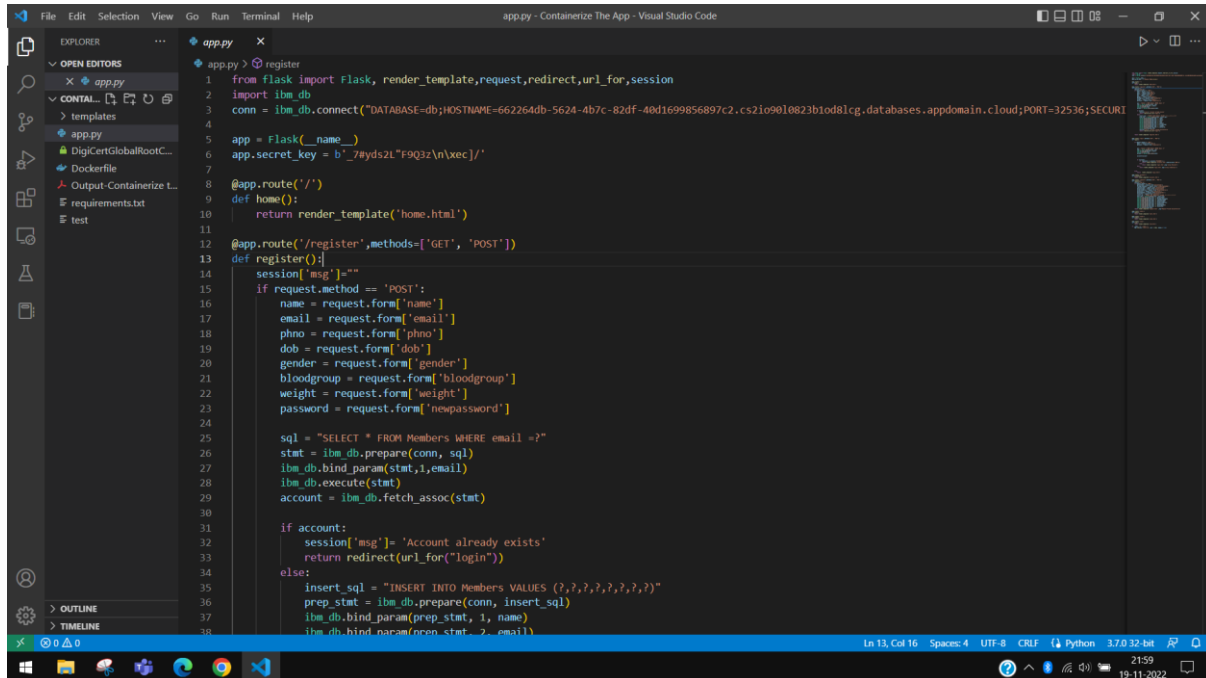


DEPLOYMENT OF APP IN IBM CLOUD

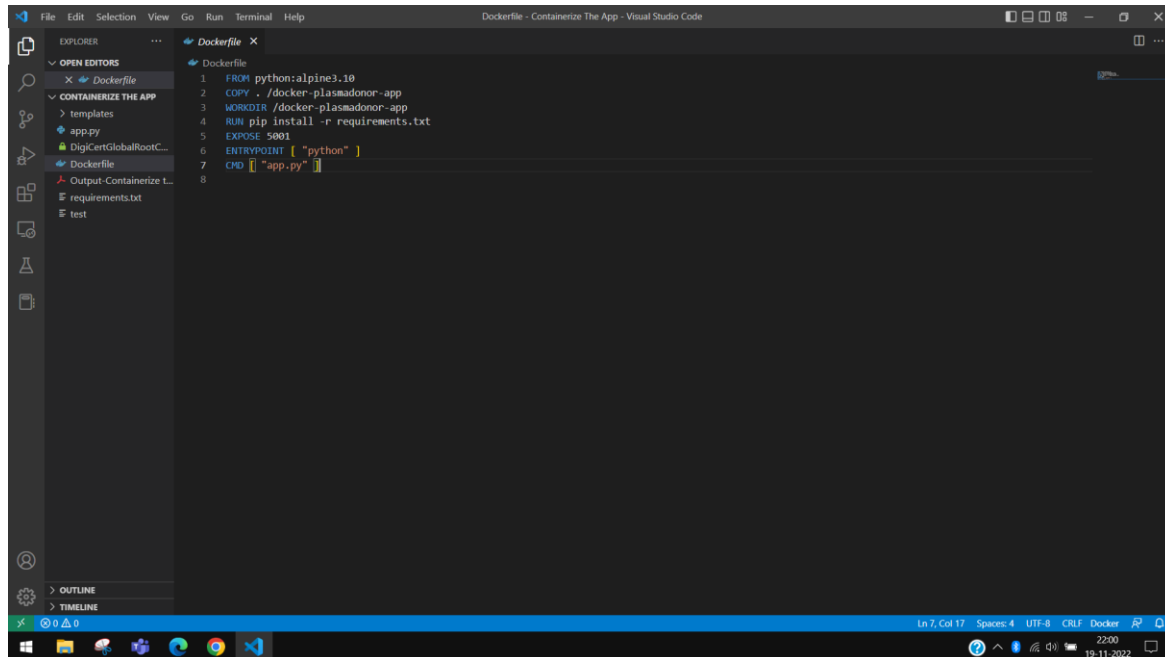
Containerize the App

Task: Create a Dockerfile for Plasma donation app and deploy it in docker desktop

- **CREATE A FLASK APP FOR PLASMA DONATION AND ADD DOCKER FILE AND NECESSARY REQUIREMENTS**



```
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 conn = ibm_db.connect("DATABASE=db;HOSTNAME=662264db-5624-4b7c-82df-40d1699856897c2.cs2io90l0823b1od8l1cg.databases.appdomain.cloud;PORT=32536;SECURI
4
5 app = Flask(__name__)
6 app.secret_key = b'_7ayds2l"F9Q3z\n\xec)/'
7
8 @app.route('/')
9 def home():
10     return render_template("home.html")
11
12 @app.route('/register', methods=['GET', 'POST'])
13 def register():
14     session['msg']=""
15     if request.method == 'POST':
16         name = request.form['name']
17         email = request.form['email']
18         phno = request.form['phno']
19         dob = request.form['dob']
20         gender = request.form['gender']
21         bloodgroup = request.form['bloodgroup']
22         weight = request.form['weight']
23         password = request.form['newpassword']
24
25         sql = "SELECT * FROM Members WHERE email =?"
26         stmt = ibm_db.prepare(conn, sql)
27         ibm_db.bind_param(stmt, 1, email)
28         ibm_db.execute(stmt)
29         account = ibm_db.fetch_assoc(stmt)
30
31         if account:
32             session['msg'] = 'Account already exists'
33             return redirect(url_for("login"))
34         else:
35             insert_sql = "INSERT INTO Members VALUES (?, ?, ?, ?, ?, ?, ?)"
36             prep_stmt = ibm_db.prepare(conn, insert_sql)
37             ibm_db.bind_param(prepare_stmt, 1, name)
38             ibm_db.bind_param(prepare_stmt, 2, email)
```



```
1 FROM python:alpine3.10
2 COPY . /docker-plasmadonor-app
3 WORKDIR /docker-plasmadonor-app
4 RUN pip install -r requirements.txt
5 EXPOSE 5001
6 ENTRYPOINT [ "python" ]
7 CMD [ "app.py" ]
```

- **BUILD A DOCKER IMAGE IN DOCKER DESKTOP FOR JOB PORTAL APP USING CMD**

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO\Desktop\PLASMA APP\PRE-DEVELOPMENT\DEPLOYMENT OF APP IN IBM CLOUD\Containerize the app>docker build -t plasma-app
"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

C:\Users\LENOVO\Desktop\PLASMA APP\PRE-DEVELOPMENT\DEPLOYMENT OF APP IN IBM CLOUD\Containerize the app>
C:\Users\LENOVO\Desktop\PLASMA APP\PRE-DEVELOPMENT\DEPLOYMENT OF APP IN IBM CLOUD\Containerize the app>docker build -t plasma-app .
[+] Building 31.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 1.4s
=> => transferring dockerfile: 221B 0.4s
=> [internal] load dockerignore 1.1s
=> => transferring context: 3B 0.2s
=> [internal] load metadata for docker.io/library/python:alpine3.10 3.7s
=> [auth] library/python:pull token for registry-1.docker.io 0.8s
=> [internal] load build context 4.1s
=> => transferring context: 27.82kB 3.8s
=> CACHED [1/4] FROM docker.io/library/python:alpine3.10@sha256:152b1952d4b42e360f2e7d3037d79b645328c0cc6fba9c63dec6ff407b96a 0.0s
=> [2/4] COPY . /docker-plasmadonor-app 0.7s
=> [3/4] WORKDIR /docker-plasmadonor-app 0.5s
=> [4/4] RUN pip install -r requirements.txt 20.0s
=> exporting image 1.3s
=> => exporting layers 1.1s
=> => writing image sha256:b94bd55b48eaf5cae9d1a8add50fc138d895486aa6f6429f3b6fef0f38167f4 0.0s
=> => naming to docker.io/library/plasma-app 0.0s

C:\Users\LENOVO\Desktop\PLASMA APP\PRE-DEVELOPMENT\DEPLOYMENT OF APP IN IBM CLOUD\Containerize the app>
```

TASK COMPLETED SUCCESSFULLY!