Make a folder in documents: myProject

Open Docker Desktop, ensure it's running and dismiss any Agreements, skip tutorial

from C:\util, copy the following files into myProject: docker-compose.yml, requirements.txt, Dockerfile

Windows Start> cmd:

C:

cd C:\Users\%username%\Documents\myProject

in cmd paste the following (including the period):

docker-compose run web django-admin startproject composeexample .

once that's completed, a project file called composeexample will have a file called settings.py

open it, add the following under the comments:

import os

then copy the following and paste over the Databases section:

```
DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': os.environ.get('POSTGRES_NAME'),

        'USER': os.environ.get('POSTGRES_USER'),

        'PASSWORD': os.environ.get('POSTGRES_PASSWORD'),

        'HOST': 'db',
```

```
        'PORT': 5432,

    }

}
```

Save settings.py

OR

Copy composeexample to docker project file.

run the command:

docker compose up

To create a new virtualenv, open Docker, in Containers tab, select the drop down (name will match folder where docker-compose runs from), in Web-1 click terminal icon:

Run:

. /opt/conda/etc/profile.d/conda.sh (consider the space between . and /)

conda activate my_env

This will give you a virtual environment where you can run (for example):

conda install django

Once you know which dependencies will work for your project you can add them to environment.yml.

To apply the dependency changes to an existing environment (my_env) you can run the line below in the terminal in which you activated my_env:

conda env update --file environment.yml --prune

Please be advised that by running *docker compose up* the most recent environment.yml will be used.

To stop docker, press 'ctrl c' keys in cmd window.

To save project, copy to OneDrive, and export .sql from pgadmin, ensure data file is deleted.

To restore, copy it back to Documents folder and import .sql and run the following 3 commands:

C:

cd C:\Users\%username%\Documents\myProject

docker compose up

To export db:

docker exec pg_container pg_dump -U root -d postgres >backup.sql

To import db:

cat backup.sql | docker exec -i pg_container psql -U root -d postgres

To Run psql commands, open Docker, in Containers tab, select the drop down (name will match folder where docker-compose runs from), in pg_container click terminal icon:

psql –U root postgres (full command: psql -h localhost -p 5432 -U root postgres –W)

and enter password

To Run PG Admin, use URL:

localhost:5050

username: admin@admin.com

password: root

connect to DB with the settings:

General: name: any

Connection: hostname/address: pg_container

user: root

password: root

You will have to input these settings after re-importing the db from the command line.