

Aung Phone Kyaw

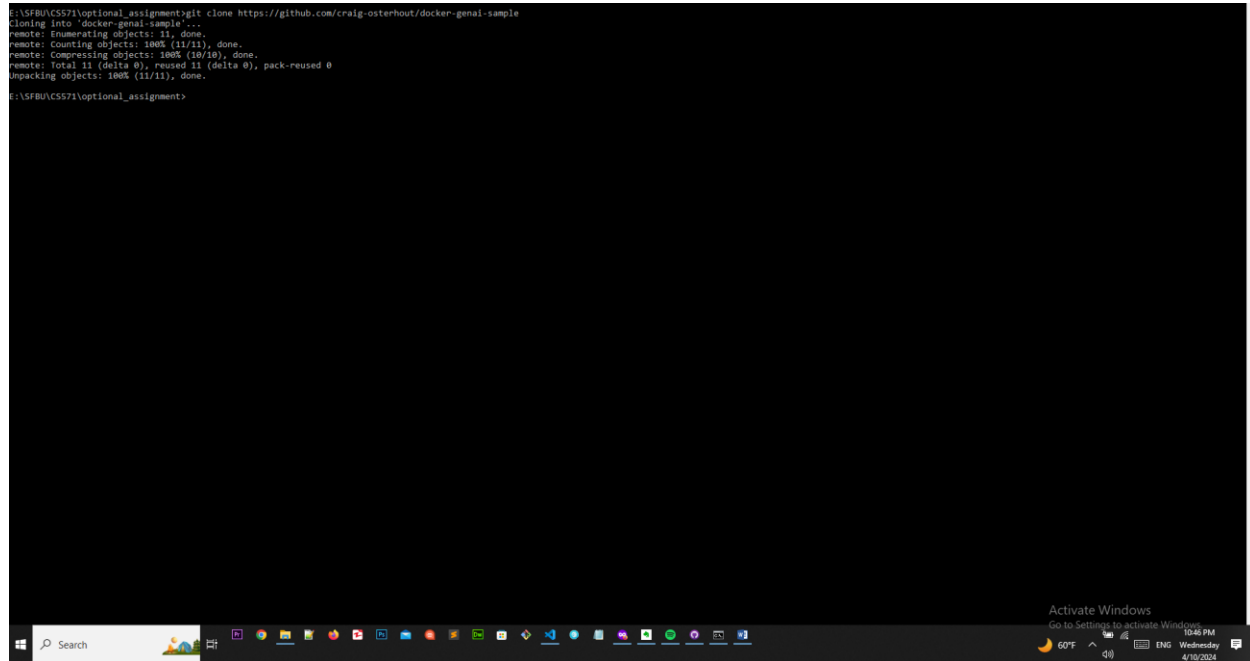
19930

CS571

Week 11

Question 5

```
E:\SFBU\CS571\optional_assignment>git clone https://github.com/craig-osterhout/docker-genai-sample
Cloning into 'docker-genai-sample'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
E:\SFBU\CS571\optional_assignment>
```



Clone gen ai project

git clone <https://github.com/craig-osterhout/docker-genai-sample>

```
E:\SFBU\CS571\optional_assignment\docker init
Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for your project:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md

Let's get started!

? What application platform does your project use? Python
? What version of Python do you want to use? (3.11.9) 3.11.4
? What version of Python do you want to use? 3.11.4
? What port do you want your app to listen on? (8000)
? What port do you want your app to listen on? 8000
? What is the command to run your app (e.g., gunicorn 'myapp.example:app' --bind=0.0.0.0:8000)? streamlit run app.py --server.address=0.0.0.0 --server.port=8000
? What is the command to run your app (e.g., gunicorn 'myapp.example:app' --bind=0.0.0.0:8000)? streamlit run app.py --server.address=0.0.0.0 --server.port=8000

CREATED: .dockerignore
CREATED: Dockerfile
CREATED: compose.yaml
CREATED: README.Docker.md

Your Docker files are ready!

Take a moment to review them and tailor them to your application.

WARNING: No requirements.txt file found. Be sure to create one that contains the dependencies for your application before running it.

When you're ready, start your application by running: docker compose up --build

Your application will be available at http://localhost:8000

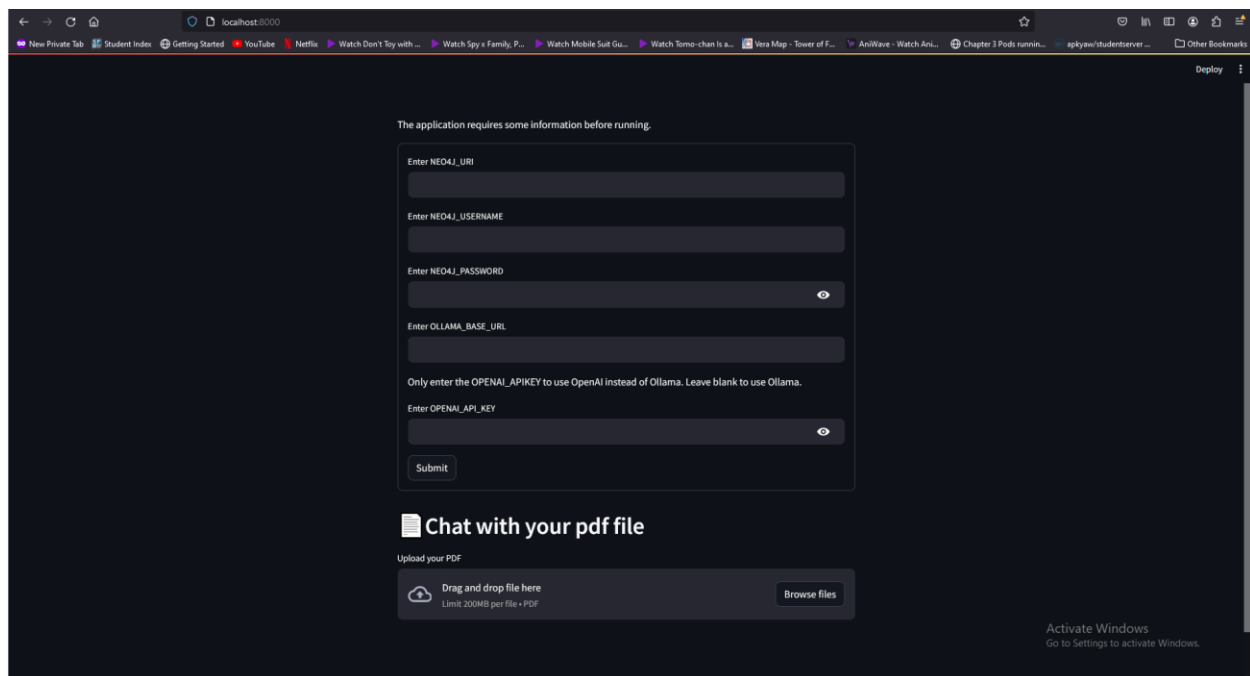
Consult README.Docker.md for more information about using the generated files.

E:\SFBU\CS571\optional_assignment
```

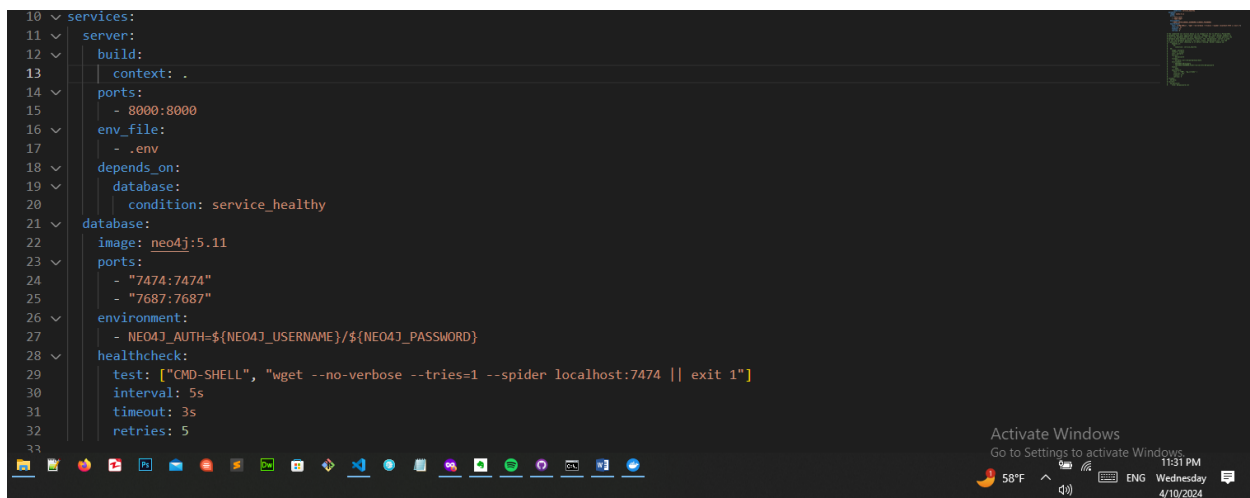
User docker init command to create docker file and other configuration file
Please choose the correct options as shown

```
E:\SFBU\CS571\optional_assignment\docker-genai-sample\docker compose up --build
[*] Building 618.25 (14/14) FINISHED
-> [server internal] load build definition from Dockerfile
-> [server internal] load .dockerignore
-> [server internal] transfer context: 60B
-> [server] resolve image config for docker.io/docker/dockerfile:1
-> [server auth] docker/dockerfile:pull token for registry-1.docker.io
-> CACHED [server] docker-image /docker.io/docker/dockerfile:1@sha256:dbd5a059e8a077f7ea0233b21b36aa11b04c13c
-> [server internal] load metadata for docker.io/library/python:3.11.4-slim
-> [server auth] library/python:pull token for registry-1.docker.io
-> [server base 3/5] FROM docker.io/library/python:3.11.4-slim@sha256:17462d681d9ecf20aaedc60959cf83b0ba3dc247
-> [server internal] load build context
-> [server base 2/5] WORKDIR /app
-> CACHED [server base 2/5] RUN adduser --disabled-password --gecos "" --home "/nonexistent" --s
-> [server base 4/5] RUN --mount=type=cache,target=/root/.cache/pip --mount=type=bind,source=requirements.
-> [server base 5/5] COPY
-> [server] exporting to image
-> [server] exporting layers
-> [server] writing image sha256:79b1dc5288de408063c6a3c47dbcb70e6cd7ac77610c728d40857d1170a
-> [server] naming to docker.io/library/docker-genai-sample-server
[*] Network docker-genai-sample_default Created
[*] Container docker-genai-sample-server-1 Created
Attaching to server-1
server-1 Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.
server-1
server-1 You can now view your Streamlit app in your browser.
server-1
server-1 URL: http://0.0.0.0:8000
server-1
server-1 There was a problem when trying to write in your cache folder (/nonexistent/.cache/huggingface/hub). You should set the environment variable TRANSFORMERS_CACHE to a writable directory.
```

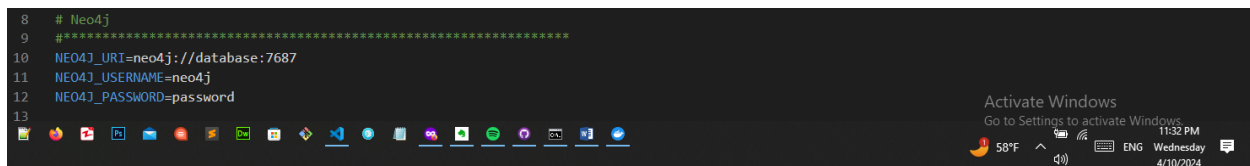
Build using docker compose up --build



Access it using localhost:8000



Add configuration step for neo4j in docker compose file



Add .env file add user name and password fields for neo4j

```
E:\SFBU\CS571\optional_assignment\docker-genai-sample>docker compose up --build
2024/04/10 23:27:39 http://server: error reading preface from client //./pipe/docker_engine: file has already been closed
[*] Building 50:its (12/12) FINISHED
-> [server internal] load .dockerignore
-> [server internal] load build definition from Dockerfile
-> [server internal] load build context
-> [server] resolve image config for docker.io/docker/dockerfile:1
-> CACHED [server] docker image /docker.io/docker/dockerfile:189ba256:d8d5e059eba077f7ead233b21b3baad3bb4c53c645f1017a4dd10bb83bea50
-> [server internal] load metadata for docker.io/library/python:3.11.4-slim
-> [server base 1/5] FROM docker.io/library/python:3.11.4-slim@sha256:170c2d081d9ecf20aadc0009ebcf8300ba3dc247013e2f43e10a4a4d4901
-> [server internal] load build context
-> [server] exporting to image
-> [server] pushing layers
-> [server] pushing image sha256:79218c528bd408636cae3ca7dcbba70ec8d7ac77010c728d40b57d1170ba
-> [server] naming to docker.io/library/docker-genai-sample-server
[*] Running 1/0
Container docker-genai-sample-database-1 Running
Attaching to database-1, server-1
```

Build using docker compose up --build

```
21 ollama-pull:
22   condition: service_completed_successfully
23 ollama-pull:
24   image: docker/genai:ollama-pull
25   env_file:
26     - .env
27 database:
28   image: neo4j:5.11
29   ports:
30     - "7474:7474"
31     - "7687:7687"
32   environment:
33     - NEO4J_AUTH=${NEO4J_USERNAME}/${NEO4J_PASSWORD}
34   healthcheck:
35     test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider localhost:7474 || exit 1"]
36     interval: 5s
37     timeout: 3s
38     retries: 5
39 ollama:
40   image: ollama/ollama:latest
41   ports:
42     - "11434:11434"
43   volumes:
44     - ollama_volume:/root/.ollama
45   deploy:
46     resources:
47       reservations:
48         devices:
49           - driver: nvidia
50             count: all
51             capabilities: [gpu]
52 volumes:
53   ollama_volume:
54
55 # The commented out section below is an example of how to define a PostgreSQL
```

Add configuration for ollama pull and ollam in docker compose file

https://github.com/AungPhoneKyaw/gen_ai_containerize_app