# Project: GenAI - Containerized video transcription and chat app

## Introduction
- This project involves creating a chatbot that can answer questions from a video. It showcases a blend of technologies including Docker, OpenAI, Whisper, Embeddings, Chat completions, Pinecone, and Retrieval-Augmented Generation.

## Prerequisites
- OpenAI API Key [https://platform.openai.com/api-keys]
- Pinecone API Key [https://app.pinecone.io/organizations/-NuXXZWkLDvM83Ow2Jo/projects/80dc8048-05b6-40a0-8972-eb9b6e5e7e3b/indexes]
- Latest version of Docker Desktop
- Git client

## Application Purpose
- The chatbot is designed to transcribe video content and answer questions related to the video, providing relevant timestamps from the source.

## Step 1: GenAI - Containerize your app.
1. First, install the latest version of Docker Desktop for windows.
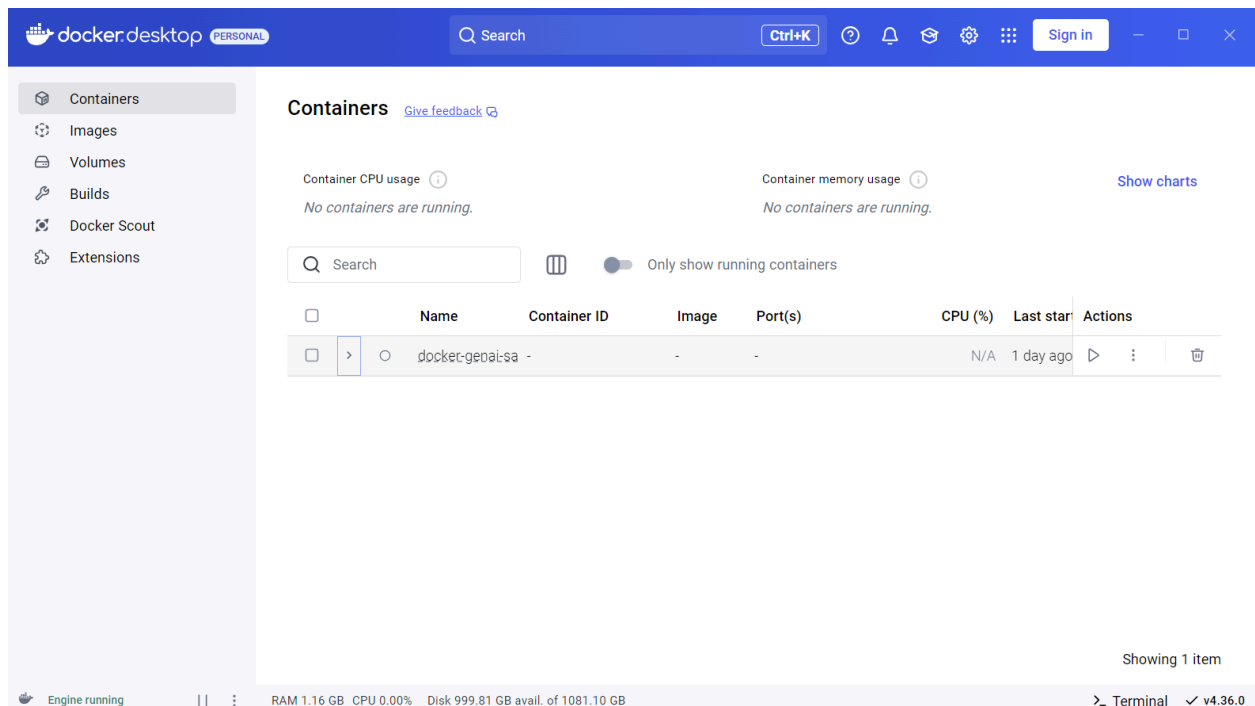
2. Create a working directory and navigate inside it(optional).

```
C:\Users\patel>mkdir video_transcription_chat_app

C:\Users\patel>cd video_transcription_chat_app

C:\Users\patel\video_transcription_chat_app>
```

3. Clone the repository from GitHub to your machine by using a command-line based git client:
   - git clone https://github.com/Davidnet/docker-genai.git

```
C:\Users\patel\video_transcription_chat_app>git clone https://github.com/Davidnet/docker-genai.git
Cloning into 'docker-genai'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 66 (delta 24), reused 60 (delta 20), pack-reused 0 (from 0)
Receiving objects: 100% (66/66), 114.38 KiB | 1.14 MiB/s, done.
Resolving deltas: 100% (24/24), done.

C:\Users\patel\video_transcription_chat_app>
```

4. Navigate to the directory of the cloned repository.
   - cd docker-genai

```
C:\Users\patel\video_transcription_chat_app>cd docker-genai

C:\Users\patel\video_transcription_chat_app\docker-genai>
```

   - You should now have the following files inside the directory.

```
C:\Users\patel\video_transcription_chat_app\docker-genai>dir
 Volume in drive C is OS
 Volume Serial Number is 8AA4-9ED7

 Directory of C:\Users\patel\video_transcription_chat_app\docker-genai

11/29/2024  01:06 PM    <DIR>          .
11/29/2024  01:06 PM    <DIR>          ..
11/29/2024  01:06 PM               494 .env.example
11/29/2024  01:06 PM             3,668 .gitignore
11/29/2024  01:06 PM    <DIR>          docker-bot
11/29/2024  01:06 PM               298 docker-compose.yml
11/29/2024  01:06 PM             1,091 LICENSE
11/29/2024  01:06 PM               103 README.md
11/29/2024  01:06 PM    <DIR>          yt-whisper
               5 File(s)          5,654 bytes
               4 Dir(s)  837,838,516,224 bytes free

C:\Users\patel\video_transcription_chat_app\docker-genai>
```

5. Next, in the docker-genai directory, create a text file called .env and specify your API keys.

```
C:\Users\patel\video_transcription_chat_app\docker-genai>notepad .env.txt

C:\Users\patel\video_transcription_chat_app\docker-genai>
```

```
#-------------------------------------------------
# OpenAI
# #-----------------------------------------------
# OPENAI_TOKEN=sk-proj-ErSHKC4CgdHCJusHVpsQkOHE7kK0t457
4BVFUX2fT3BlbkFJ3jaigLncHYaQQPUDhqFP7CkXzVaAjCXs1ISpi86_
A_# Replace your-api-key with your personal API key
#
# #-----------------------------------------------
# # Pinecone
# #-----------------------------------------------
# PINECONE_TOKEN=pcsk_5B9dwX_5WYn8C1REwg1sV8GrLtLZ7P6HdN
qi9 # Replace your-api-key with your personal API key
```

```
C:\Users\patel\video_transcription_chat_app\docker-genai>dir
 Volume in drive C is OS
 Volume Serial Number is 8AA4-9ED7

 Directory of C:\Users\patel\video_transcription_chat_app\docker-genai

11/29/2024  01:33 PM    <DIR>          .
11/29/2024  01:06 PM    <DIR>          ..
11/29/2024  01:33 PM             1,602 ..env.un~
11/29/2024  01:33 PM               724 .env
11/29/2024  01:06 PM               494 .env.example
11/29/2024  01:06 PM             3,668 .gitignore
11/29/2024  01:06 PM    <DIR>          docker-bot
11/29/2024  01:06 PM               298 docker-compose.yml
11/29/2024  01:06 PM             1,091 LICENSE
11/29/2024  01:06 PM               103 README.md
11/29/2024  01:06 PM    <DIR>          yt-whisper
               7 File(s)          7,980 bytes
               4 Dir(s)  837,600,632,832 bytes free

C:\Users\patel\video_transcription_chat_app\docker-genai>
```

6. Build and Run the application. For Docker to do this, run the following command in a terminal inside the docker-genai directory:
- docker compose up - -build

```
C:\Users\patel\video_transcription_chat_app\docker-genai>docker compose up --build
[+] Building 6.6s (9/16)                                          docker:desktop-linux
 => [yt-whisper internal] load build definition from Dockerfile                    0.1s
 => => transferring dockerfile: 1.88kB                                             0.0s
 => [bot internal] load build definition from Dockerfile                           0.1s
 => => transferring dockerfile: 1.88kB                                             0.0s
 => [bot] resolve image config for docker-image://docker.io/docker/dockerfile:1    1.7s
 => CACHED [yt-whisper] docker-image://docker.io/docker/dockerfile:1@sha256:865e5d 0.0s
 => => resolve docker.io/docker/dockerfile:1@sha256:865e5dd094beca432e8c0a1d5e1c46 0.0s
 => [yt-whisper internal] load metadata for docker.io/library/python:3.11-slim     1.4s
 => [bot internal] load .dockerignore                                              0.1s
 => => transferring context: 2B                                                    0.0s
 => [yt-whisper internal] load .dockerignore                                       0.1s
 => => transferring context: 2B                                                    0.0s
 => [yt-whisper base 1/5] FROM docker.io/library/python:3.11-slim@sha256:e8381c802 3.1s
 => => resolve docker.io/library/python:3.11-slim@sha256:e8381c802593deb0c4d25bd3f 0.1s
 => => sha256:2d429b9e73a6cf90a5bb85105c8118b30a1b2deedeae3ea9587 5.24MB / 29.13MB 2.8s
 => => sha256:173289c0cbe5b5760030dda93a84319ef683a489a0b33b176284679a 250B / 250B 0.4s
 => => sha256:71ba669986f7c60a5e178baa52bc67b3821d038c49d6bf03741 1.05MB / 16.20MB 2.8s
 => => sha256:14dbff54af923889a0e26a829553caa713f43c3b921620fd2d5d 1.05MB / 3.51MB 2.8s
 => [bot internal] load build context                                              0.2s
 => => transferring context: 132.54kB                                              0.0s
```

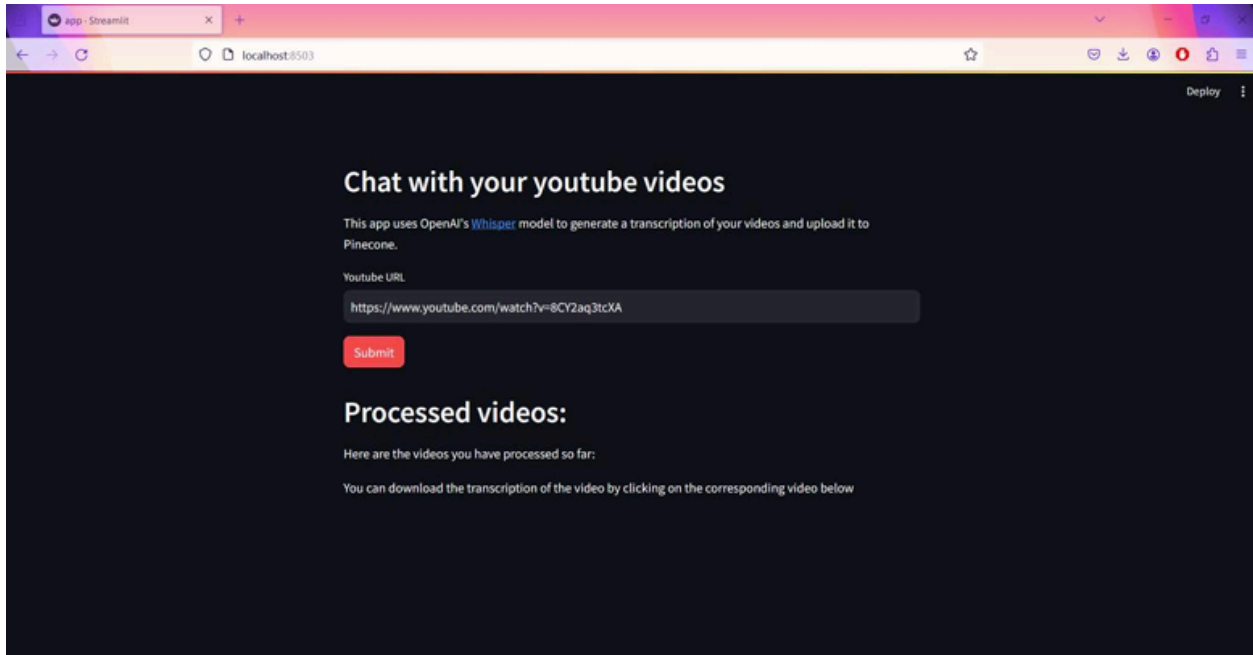# Project: GenAI - Containerized video transcription and chat app
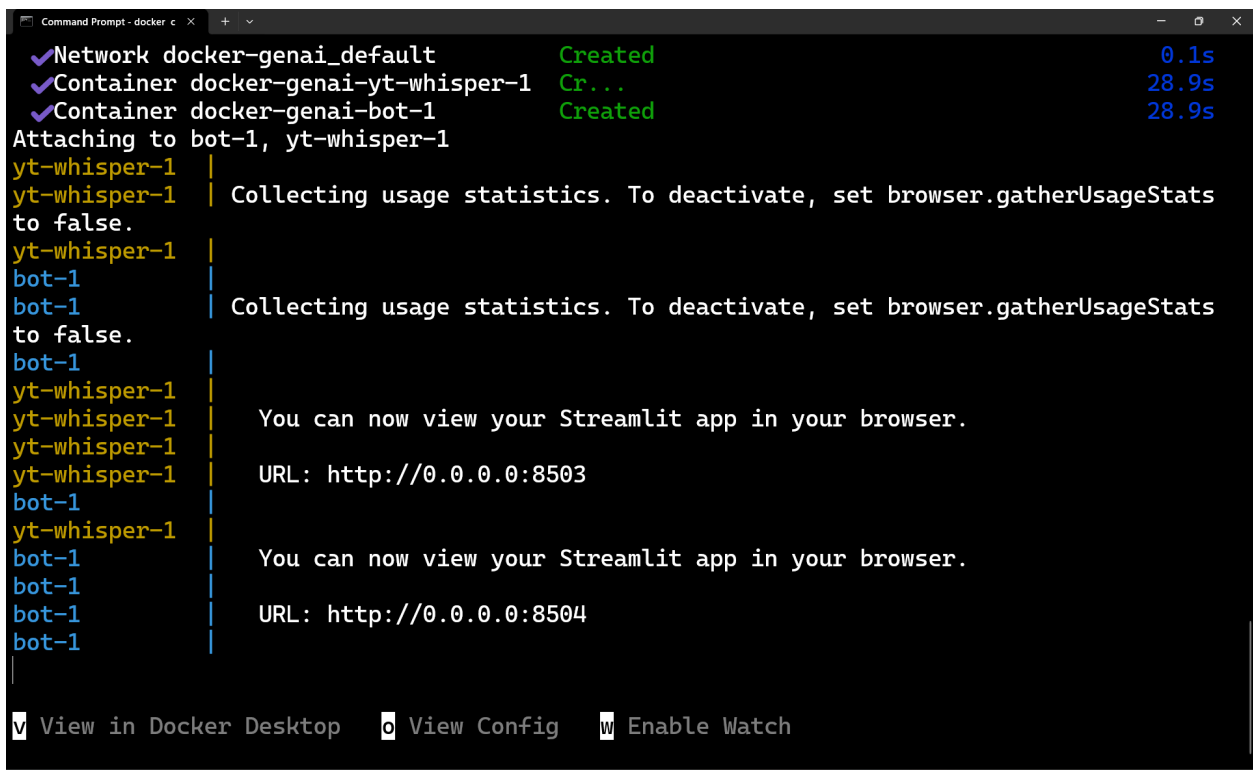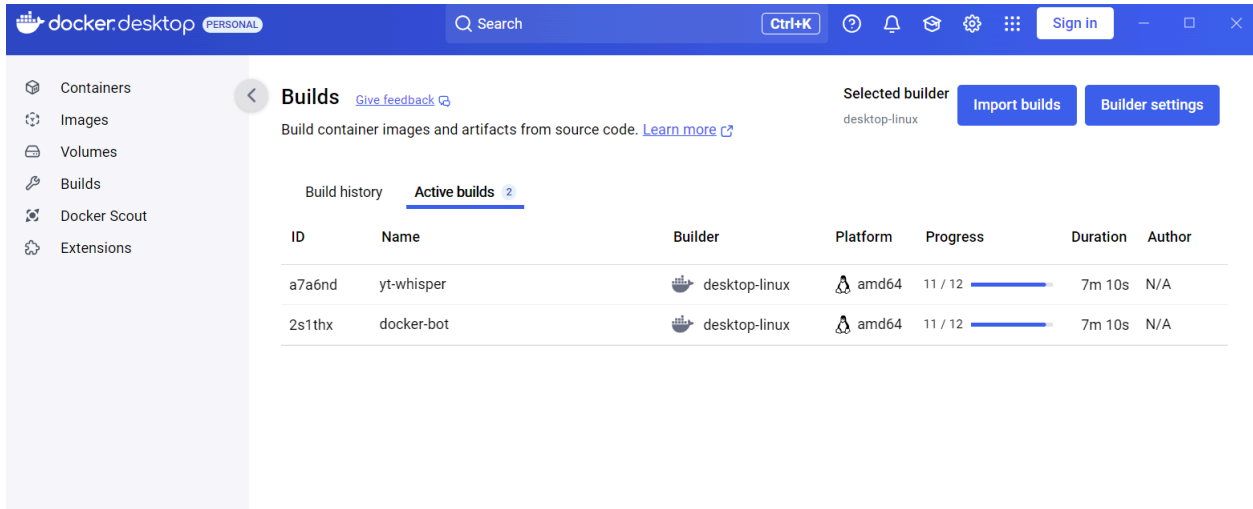


- In the above logs, we can see the services are exposed on ports **8503** and **8504**. The two services are complementary to each other.
- The yt-whisper service is running on port 8503. This service feeds the Pinecone database with videos that you want to archive.

## Step 2: Video transcription and chat
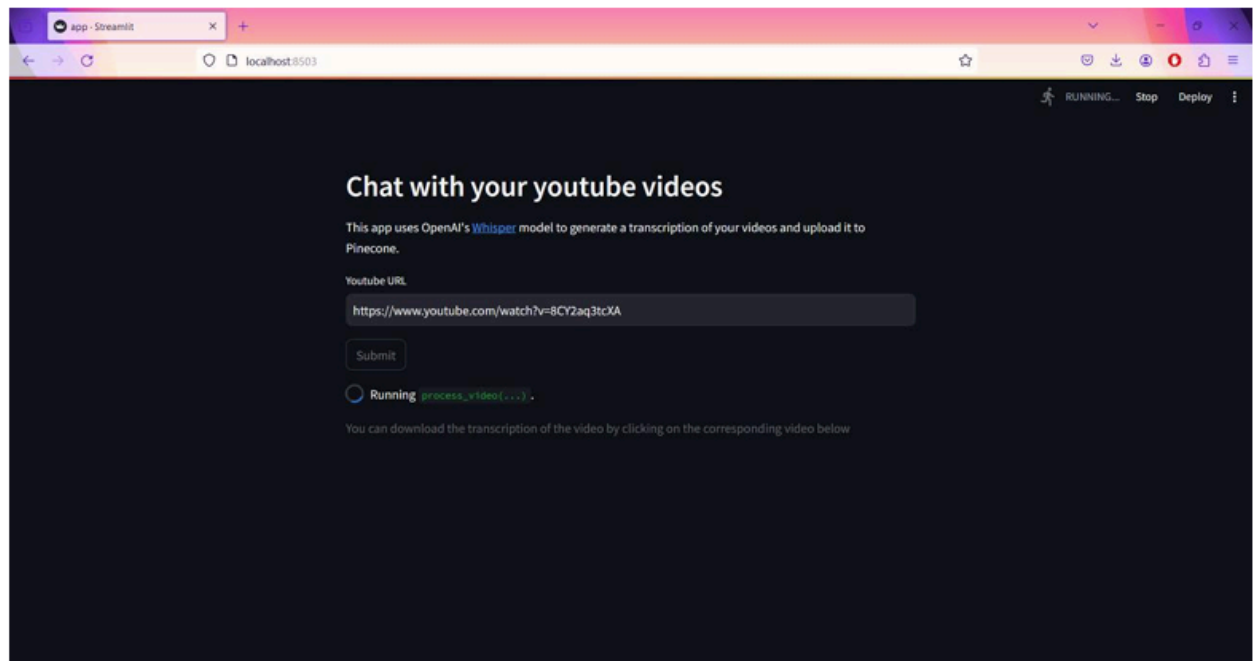Using the yt-whisper Service
7. Open a browser and access the yt-whisper service at:
   - http://localhost:8503

8. Once the application appears, in the Youtube URL field specify a Youtube video URL and select Submit. Let's try these links.
   ● https://www.youtube.com/watch?v=8CY2aq3tcXA
   ● www.youtube.com/watch?v=yaQZFhrW0fU
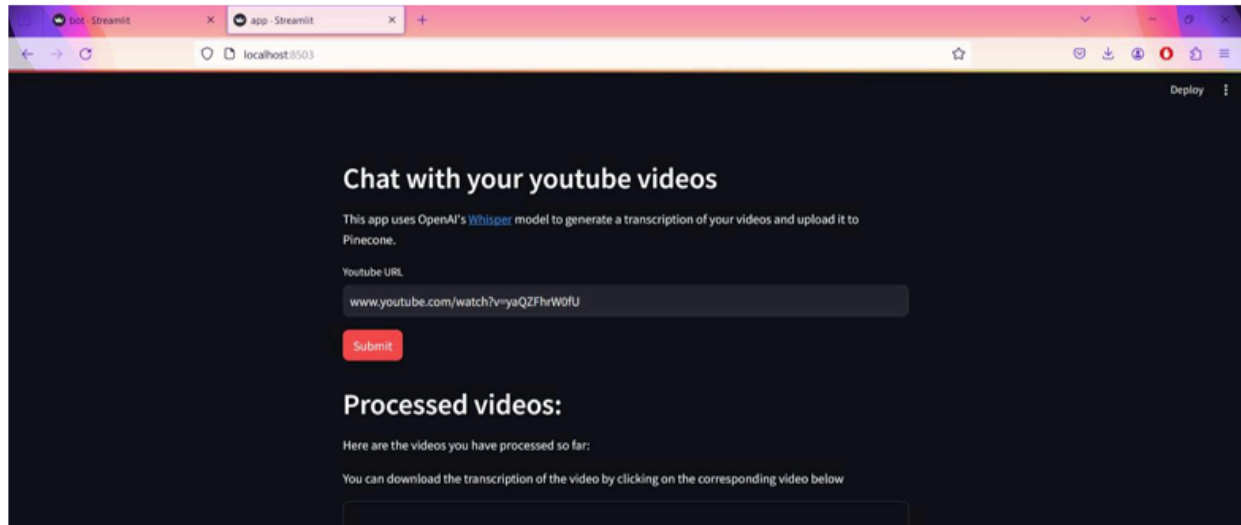
9. The yt-whisper service downloads the audio of the video, uses Whisper to transcribe it into a WebVTT (*.vtt) format then uses the text-embedding-3-small model to create embeddings, and finally uploads those embeddings into the Pinecone database.

10. After processing the video, a video list will appear in the web app showing the videos that have been indexed in Pinecone. You can click the button to download the transcript.
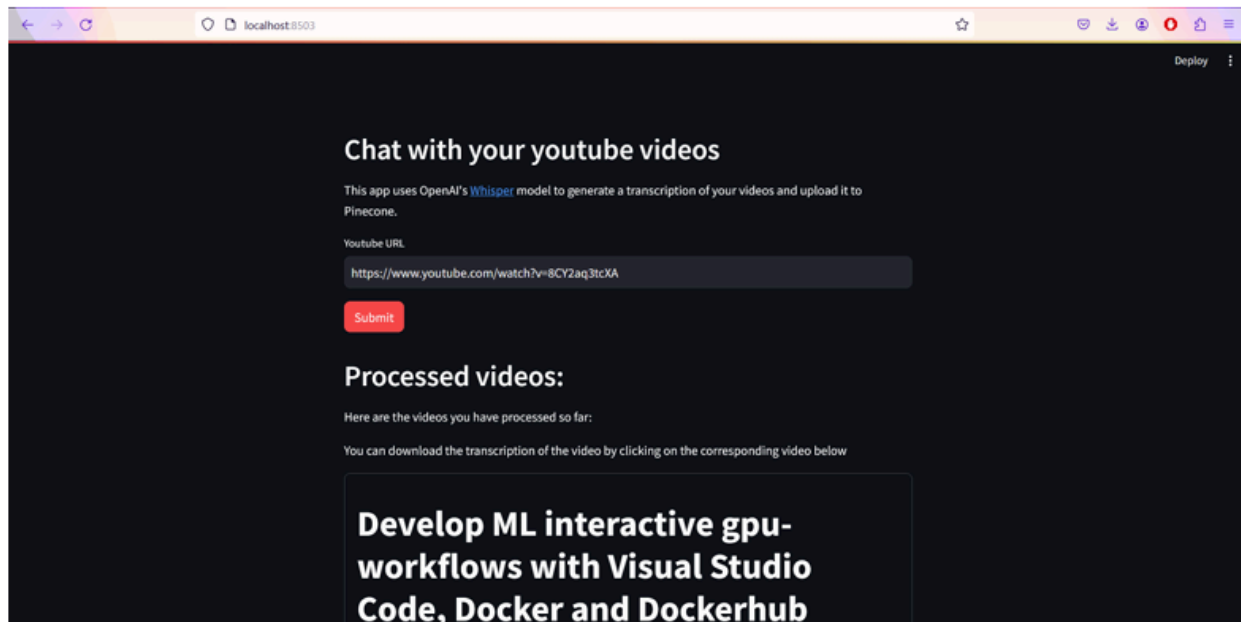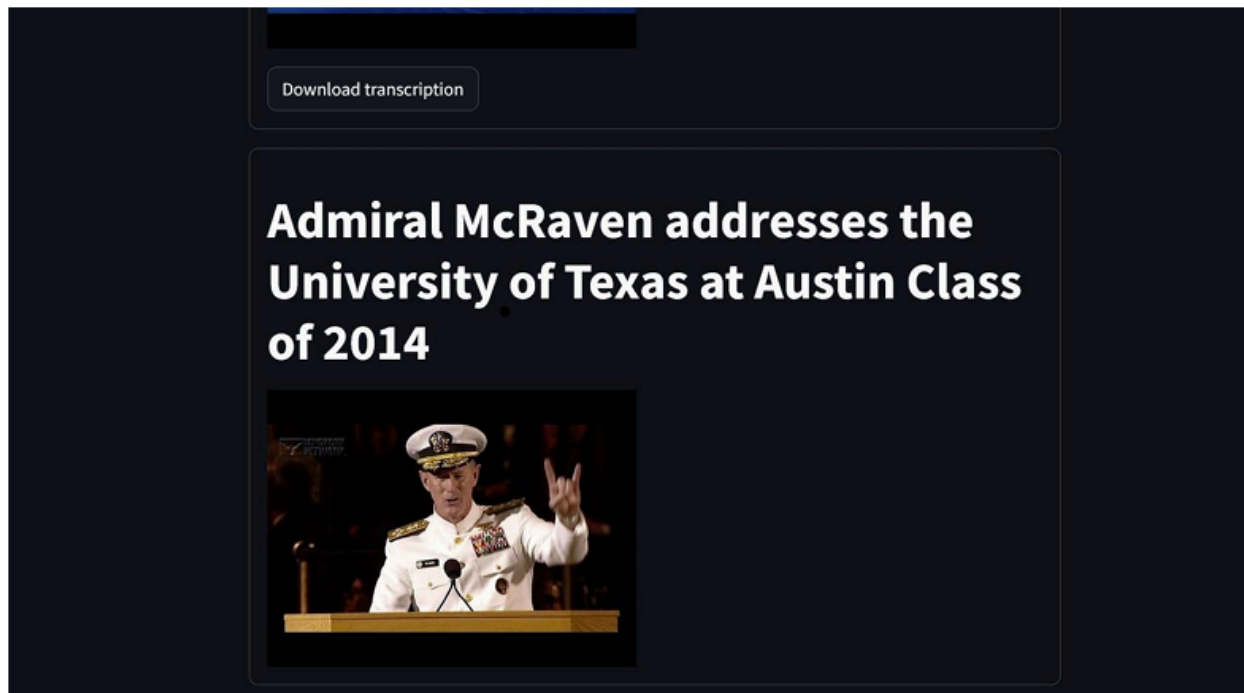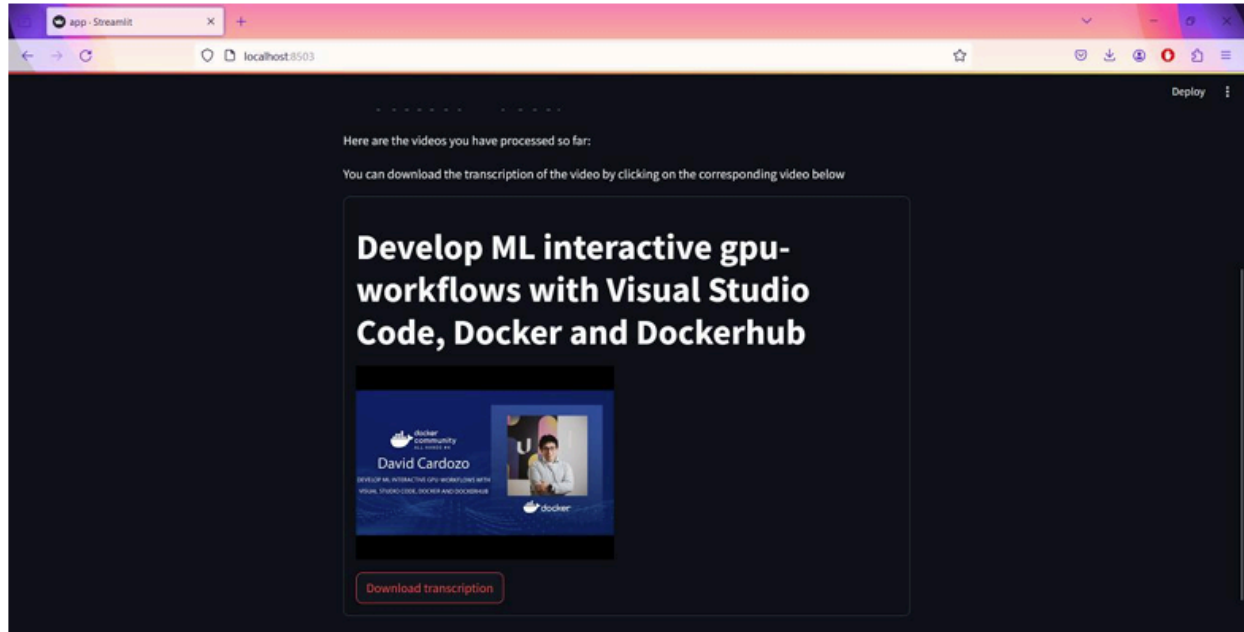
**Using the dockerbot Service**

11.    Now we can access the dockerbot service on port 8504 and ask questions about the videos that were processed by the yt- whisper service. Open a browser and access the service at

-    http://localhost:8504
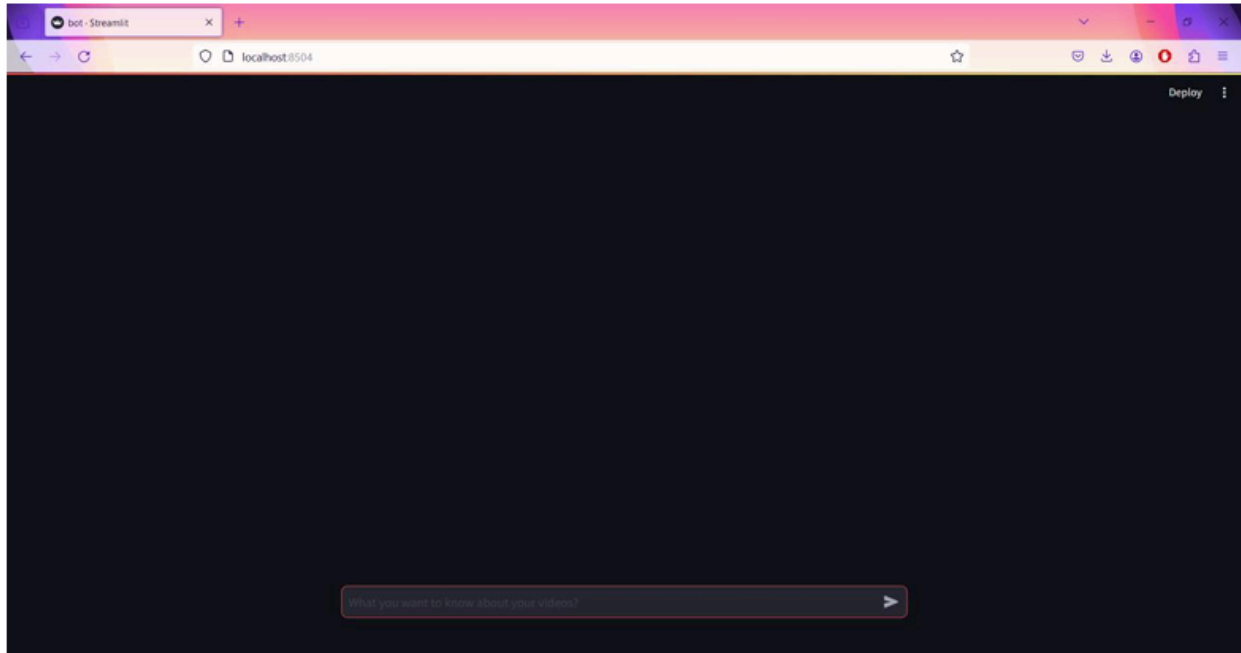
12. What do you want to know about your videos? text box, ask the Dockerbot a question "What is a sugar cookie?". The answer to that question should be displayed as it exists in one of the videos processed in the previous example,
https://www.youtube.com/watch?v=yaQZFhrW0fU
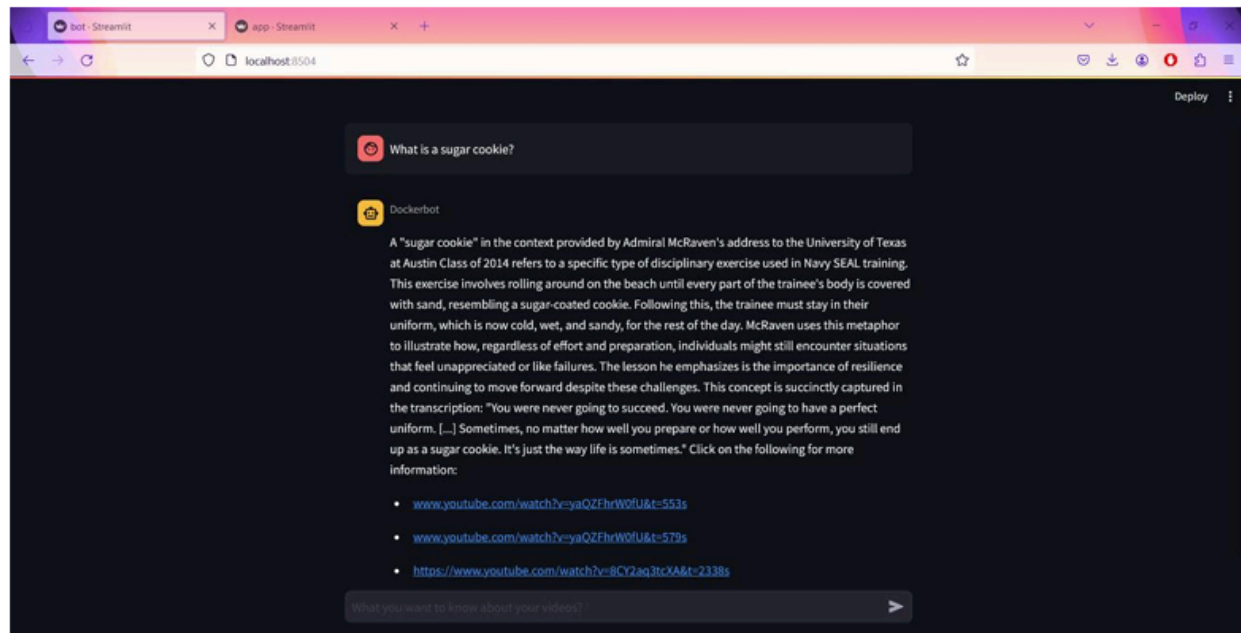
13. Docker Compose is a tool for defining and running multi-container applications. Here with a single command, it enables to easily run this application and simplifies the control of your entire application stack, making it easy to manage all things in a single, comprehensible YAML configuration file.



14. To stop the application, we press ctrl+C in the terminal.