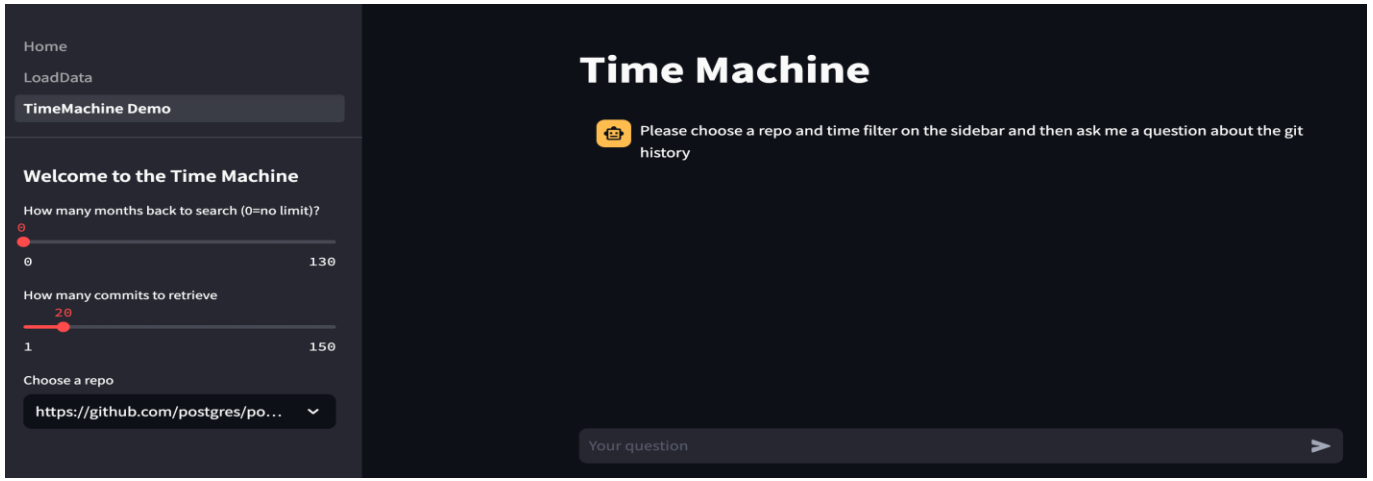
Tech Review

Project Name: TLDHubeR

A GPT-Powered Search and Summary App for the Huberman Lab Podcast

Background

1. **Intro:** Our application concept is an LLM-powered podcast which we are tuning to search and summarize the Huberman Lab Podcast.
2. **Problem:** Podcasts are long and numerous. What if you had a tool to help people find content within a podcast, but needed a way for people to access your tool?
3. **Constraint:** Coding front-end from scratch would be infeasible.



User Story and Use Case

1. User Story:
   - Alice is a longtime fan of the of the Huberman Lab podcast. She wants to search the podcast for when a certain topic was discussed, but she can't quite remember what she heard or the episode it was from.
2. Use Case:
   - Alice wants a simple to use webapp that she can use to find podcast titles, discussion timestamps, and maybe summaries by entering search terms into a search box.
     - Suppose we have the tool, but need to make it into a webapp? What should we do?

Python Package Choices

1. Streamlit
   - Created by Streamlit Inc. (acquired by Snowflake in early 2022)
   - Open-source app framework for Data Science teams designed to turn data scripts into web apps with minimal effort, without requiring front-end experience

2. Dash
   - Developed by Plotly
   - Python framework for building web analytic applications. It is written on top of Flask, Plotly.js, and React.js, and it is ideal for building data visualization apps with highly customizable user interfaces in pure Python.

Package Comparison

STREAMLIT

- Excellent for rapid prototyping of a user interface due to prebuilt components, but it is not as customizable
- Lower learning curve and straightforward setup
- Streamlit allows injection of HTML into the app, but it's considered more of a "hack" than a feature
- Does not have a callback system, which results in user interaction triggering rerun of the entire script (can be computationally expensive)

DASH

- Requires a bit more familiarity with web development concepts, but offers more flexibility
- Allows for the direct use of HTML components through the dash_html_components library, and CSS can be linked/included inline for customization
- Dash's callback system allows developers to define how the UI should update in response to interactions, which allows more precise control

Our Choice : Streamlit

We opted for Streamlit due to its versatility and simplicity in building interactive web applications with Python.

Streamlined Development: Streamlit's intuitive framework allows for rapid development without the need for extensive coding.

All-in-One Solution: Streamlit offers integrated support for back-end functionalities, including API integration and logic implementation, minimizing development complexity.

Seamless Integration: Streamlit's seamless integration with Python libraries, such as OpenAI API and Llama Index logic, facilitates cohesive implementation within a single platform.

Enhanced Efficiency: Leveraging Streamlit eliminates the need for building separate components for front-end and back-end, streamlining development efforts and reducing overhead.

Drawbacks/Remaining Concerns

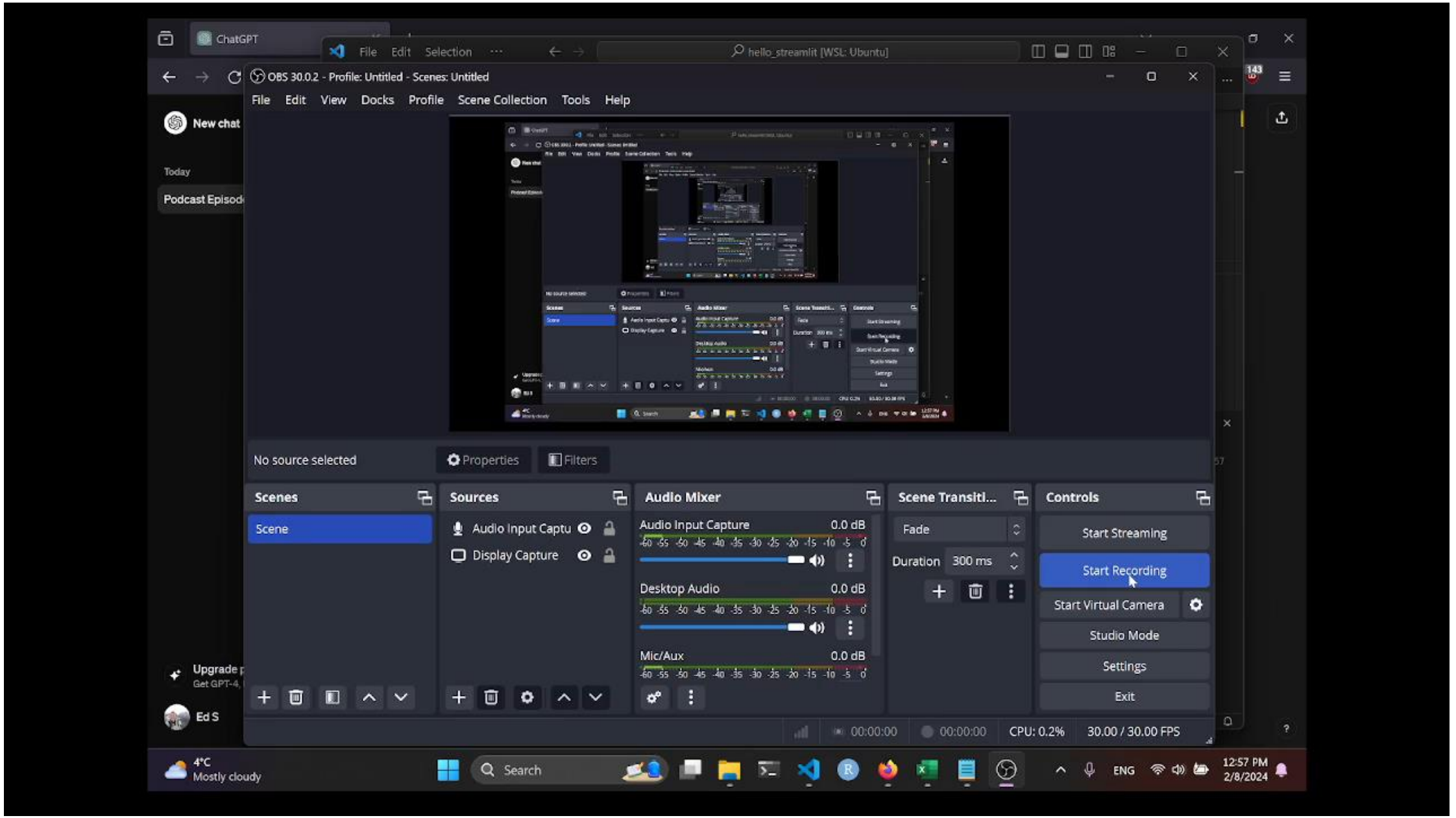
Potential Drawbacks:

- **Limited Customizability**: Streamlit offers less flexibility than Dash.
- **Performance Concerns**: For extremely large datasets or highly interactive applications, Streamlit's performance might lag behind Dash (Dash can leverage direct Flask and React.js optimizations).
- **Community and Support**: Dash, being older, may have a broader range of community-contributed components and examples. May need to utilize open-source Streamlit projects to see examples.

Point to Watch:

- **Scalability**: Monitor application performance as user base and dataset size grows with more transcripts and video information.

Tech Demo: Hello Huberman (Next Slide)


ChatGPT
File Edit Selection
hello_streamlit [WSL: Ubuntu]
OBS 30.0.2 - Profile: Untitled - Scenes: Untitled
File Edit View Docks Profile Scene Collection Tools Help
New chat
Today
Podcast Episod
No source selected
Properties
Filters
Scenes
Scene
Sources
Audio Input Captu
Display Capture
Audio Mixer
Audio Input Capture 0.0 dB
Desktop Audio 0.0 dB
Mic/Aux 0.0 dB
Scene Transiti...
Fade
Duration 300 ms
Controls
Start Streaming
Start Recording
Start Virtual Camera
Studio Mode
Settings
Exit
00:00:00
00:00:00
CPU: 0.2%
30.00 / 30.00 FPS
Upgrade
Get GPT-4,
Ed S
4°C
Mostly cloudy
Search
ENG
12:57 PM
2/8/2024