Why I selected these approaches/tools/libraries

- **FFmpeg** – I chose FFmpeg because it's the most reliable and widely supported tool for exporting .mp4 files from .m3u8 streams. It's cross-platform, works well with both live and static streams, and preserves quality without unnecessary re-encoding. Since my project needs to run across different operating systems, I opted to require the end-user to download and install FFmpeg locally, avoiding compatibility issues that could arise if I bundled binaries for a single platform.
- **Puppeteer** – Mako hides its .m3u8 sources inside a custom playlist rather than exposing them directly in the HTML. Puppeteer allows me to spin up a headless browser, imitate a real user opening the page, and intercept the network requests to locate the master.m3u8 URL. This approach avoids brittle HTML parsing and ensures I'm getting the exact stream URLs the site delivers to real browsers.

Limitations / Edge Cases

- **FFmpeg** – Users must install it manually, which can be a hurdle for non-technical users. Network interruptions can corrupt output files unless handled with retry logic. Some .m3u8 streams may use encryption (e.g., AES-128), requiring additional handling.
- **Puppeteer** – Launching a browser adds overhead and can be slow compared to direct network calls. If Mako changes their playlist structure or adds obfuscation, my scraping logic will need updates. Puppeteer also requires more system resources than simple HTTP requests.

How AI contributed to my solution

The AI helped refine my approach by:

- Suggesting practical alternatives (e.g., regex parsing for <script> tags) and explaining the trade-offs of different extraction methods.
- Advising against unnecessary dependencies like Cheerio when my data could be extracted directly from the raw HTML.
- I used **GitHub Copilot** to speed up development by autocompleting repetitive code patterns and suggesting efficient syntax.
- I used **ChatGPT** to troubleshoot unexplained edge cases, refine my Puppeteer workflow, and identify potential bugs earlier in the process. AI also helped validate my reasoning for selecting the best-fit tools and provided clarity on how to handle cross-platform compatibility.