DEMO OF AI POWERED MEDICAL IMAGE SEGMENTATION

BY AHMED TAEHA

PROJECT OVERVIEW

- When I started this project, my goal was to create a web-based demo that could perform AI-powered medical image segmentation. I used Total-Segmentator as a reference point and aimed at developing a simplified version of it.

SYSTEM ARCHITECTURE

- I designed the system architecture to be both efficient and scalable. For the frontend, I chose to use HTML, CSS, and JavaScript with Vue.js. I really like Vue.js for its reactivity and component-based structure - it made creating a dynamic user interface much easier.
- On the backend, I went with Node.js. Its event-driven nature made it perfect for handling file uploads and managing the long-running processes involved in image segmentation.
- The actual processing happens in a Docker container running a Python script. This was a bit tricky to set up, but it allowed me to encapsulate all the dependencies needed for the segmentation process.
- The data flow I implemented goes like this: The user uploads an image through the web interface, which is then received by the Node.js server. The server passes this to the Docker container, where my Python script processes the image. The results then come back through Node.js and are displayed on the frontend. It was challenging to get all these pieces working together smoothly but I made it work.

FRONTEND DEVELOPMENT

- For the frontend, I focused on creating a user-friendly interface. I used HTML5 for the structure, CSS3 for styling, and Vue.js to manage the dynamic content and user interactions.
- I implemented several key features to enhance the user experience. First, I created a drag-and-drop file upload area to make submitting CT scans as easy as possible. I also added a progress bar to keep users informed during the segmentation process - waiting for results can be frustrating if you don't know what's happening!
- For displaying results, I created a simple list showing all the generated files, each representing a different segmented anatomical structure. I also included a section for process logs. Initially, this was for my own debugging, but I realized it could be helpful for users to see what's happening behind the scenes.

NOW TIME FOR THE LIVE DEMO!!!