

Here is a summary of all the mistakes encountered yesterday and the fixes we implemented. You can use this to structure your notes.

1. Frontend Logic Crashes

- **The Mistake:** The app would crash (White Screen of Death) when opening the Watch page or searching.
- **The Error:** `Uncaught TypeError: results.map is not a function.`
- **The Cause:** The `results` variable was becoming `null` or `undefined` (either initially or from a bad API response), and the code tried to loop through it immediately.
- **The Fix:**
- **In `Watch.jsx`:** Added a "safety check" variable: `const safeResults = results || []`; We then mapped over `safeResults` instead of `results`.
- **In `App.jsx`:** Initialized the state with an empty array: `useState([])` instead of `null`.

2. Data Mismatch (The "No Results" Bug)

- **The Mistake:** The backend was finding videos (returning 200 OK), but the frontend kept saying "No Results Found."
- **The Cause:** The Backend sent data as an object: `{"results": [...]}`. The Frontend was saving that *entire object* into the list. The UI couldn't read a list from an object.
- **The Fix:**
- Updated `handleSearch` in `App.jsx` to "unwrap" the data:
- *Old:* `setResults(data)`
- *New:* `setResults(data.results || [])`

3. UI Layout & Clickability

- **The Mistake:** The Search Bar in the Navbar was unclickable, and the Watch page looked "cut off" at the top.
- **The Cause:**
- 1. **Z-Index/Padding:** The `Watch` page content was starting at the very top of the screen (`pixel 0`), sliding *underneath* the fixed Navbar.
- 2. **Fake Input:** The "AI Search" box in the sidebar was just a text title (`<h3>`), not an actual input field.
- **The Fix:**
- Added `pt-24` (Padding Top) to the `Watch` container to push content down.
- Replaced the sidebar title with a real `<input>` and connected it to the `searchQuery` state.

4. Backend Performance (The 30-Second Delay)

- **The Mistake:** The `/search` endpoint was taking 30+ seconds, and `/list` was also slow.
- **The Cause:**
- The `/list` function (which just lists files) was using the heavy `VideoSearcher` class.
- This forced the server to load the massive AI model into the GPU just to list filenames, causing a "Cold Start" delay every time you refreshed the home page.

- **The Fix:**
- Created a lightweight `FileHelper` class (CPU only) in `main.py` specifically for listing files.
- Updated the `/list` endpoint to use this helper, keeping the heavy GPU Searcher free and ready for actual searches.

5. Component Communication (Props)

- **The Mistake:** The Home page was empty even after uploading videos.
- **The Cause: Prop Mismatch.** `App.jsx` was passing the list as `existingVideos={videoList}`, but `Home.jsx` wasn't correctly receiving or naming that prop.
- **The Fix:** Updated `Home.jsx` to explicitly accept `{ existingVideos }` and added a check to display a "Loading..." or "No videos found" message if the list was empty.