My current focus is building a working React front end for a Token Research Explorer that my college blockchain group can use to organize token notes and discussion points. At this stage, the app lets a user browse a curated list of tokens, search by name ticker chain or category, and open a detail view for a selected token, which is what I aimed to accomplish this week.

I created TokenList to render the searchable list of tokens and handle selection events, TokenCard to display a single token summary row with an Open button, and TokenDetail to display deeper information for the selected token including metrics thesis risks and a research checklist. State currently lives in App, including the token data, the selected token id, and the search query, and App passes the relevant data down to child components through props. User actions flow upward through callback props, such as clicking Open to set the selected token, typing in the search box to update the query, and toggling checklist items to update token state.

The project uses conditional rendering for page style view switching by showing the list view when no token is selected and the detail view when a token is selected. The view change is triggered by clicking Open on a token card, and the Back button clears the selection to return to the list. The most solid part of the project is the component separation and predictable data flow through props and callbacks. The most uncertain part is deciding how to expand the scope next, such as adding sorting and filters, adding a notes editor, or replacing mock token data with real data sources.

I used AI to help brainstorm component structure and clarify React state and prop patterns, but I decided myself which features to include this week to keep the project manageable and aligned with the course requirements.