

Audit History of Prompts for React Investment App

1. App Creation

Original Prompt:

Create a React application which displays a Fictional Banking Investment portfolio page with React financial charts incorporating CRUD operations.

Refined Prompt:

Update the component to display holdings and asset classes in AG Grid using mock data. Incorporate Add and Delete icons.

Ideal Prompt:

As a front-end engineer, build a responsive React app for a fictional banking investment dashboard. Use AG Grid to display asset classes and holdings with CRUD operations and mock data for visual charts.

2. Grid Implementation

Original Prompt:

Use AG Grid with updating the mocks and App.jsx

Refined Prompt:

Use AG Grid with light gray background, contrasting labels, equal width columns, scrollable overflow.

Ideal Prompt:

Implement AG Grid to display investment data (asset classes and holdings) with full CRUD, column resizing, and responsive layout. Use a light gray background and visually distinct label styles.

3. Mock Data Setup

Original Prompt:

Create mock based on the AG Grid column definitions

Refined Prompt:

Use realistic public portfolio values for 10+ columns like market cap, risk, region, tax.

Ideal Prompt:

Generate comprehensive mock data for investment asset classes and holdings using real-world terminology (e.g., market cap, allocation, region). Structure it for AG Grid compatibility.

4. API Integration

Original Prompt:

Create a service file to perform Fetch (GET), Save (POST), and Delete

Refined Prompt:

Use Axios. Replace mock data in App.tsx with service fetch logic.

Ideal Prompt:

Implement `apiService.ts` using Axios with methods for GET (fetch data), POST (save new records), and DELETE (remove by ID). Refactor App.tsx to initialize state via API.

5. Save/Delete UI

Original Prompt:

Add a tick icon beside delete to trigger a save action

Refined Prompt:

Use FiCheck and FiTrash2 icons in the last column of AG Grid for save and delete.

Ideal Prompt:

Add action buttons to each AG Grid row: a Save save (POST) and Delete delete (DELETE). Trigger API calls through `apiService.ts` on click.

6. Dark Mode & Styling

Original Prompt:

Create a background layout of light grey with contrasting colors

Refined Prompt:

Add global styling in App.css, support dark mode with toggle and localStorage persistence.

Ideal Prompt:

Style layout with a light gray theme and contrasting section labels. Add global dark mode support with toggle switch and localStorage preference.

7. Prompt Tracking & PDF Export

Original Prompt:

Update the prompt repo as new prompts come in

Refined Prompt:

Create a prompt library grouped by initial, refined, and role-based prompts.

Ideal Prompt:

Maintain a prompt history log tracking all initial prompts, refinements, and role-based rewrites. Export the collection as a structured PDF file.