

Token List Technical Assessment Exercise

Our team has put together an assessment to better understand how you approach project work similar to what we do. Please read through the entire exercise and let us know if you have any questions or feedback. Instructions on submitting are at the bottom.

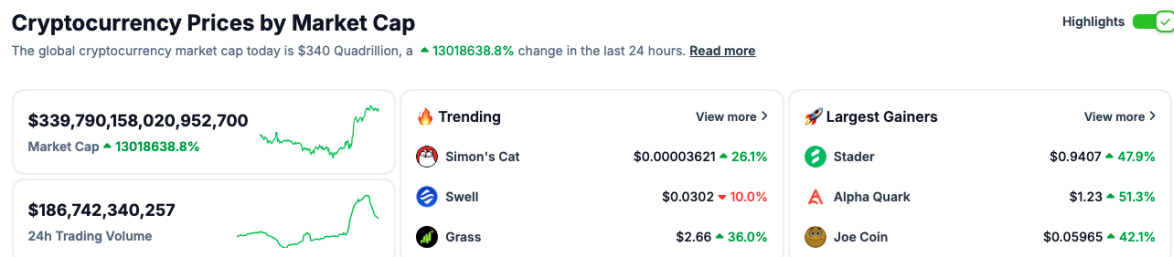
Overview

The objective of this project is to develop a fungible token-based frontend web application in our required stack. The page will display a table of tokens with their respective market data using a default set of columns, in a default column order, called Trending view. Users should be able to drag column order and use the “Customize view modal” to add/remove columns, and enter a “label name” to save them to local storage. Saved views will be accessible in tabs on top of the table, alongside the Trending (default) label.

Final UI should closely resemble <https://www.coingecko.com/>; please find the complete list of requirements below.

Technical Requirements

1. Frameworks and libraries: Next.js (with page router), Tailwind CSS, shadcn/ui, Coingecko API, React Query, TanStack table, Jotai.
2. Refer to the Coingecko homepage, you need to include following sections:
 - a. Highlights/summary section



- b. Tokens tables (with additional features)

- i. There would be a set of tabs to help users navigate through different saved views.
3. Following columns should be included in the table:
 - a. Coin name
 - b. Price
 - c. 1h, 24h, 7d price change percentage
 - d. 24h Volume
 - e. Market Cap
 - f. Last 7 days price chart (using <https://nivo.rocks/line/>)
4. Technical requirements for table:
 - a. Table should be implemented using TanStack Tables
 - b. Columns should be sortable (ascending/descending)
 - c. Column headers should be draggable, allowing users to rearrange columns. Use <https://dndkit.com/> to implement this functionality.
 - d. Customize button (dissimilar functionality than CoinGecko, detailed below)
5. Customize button/modal requirements:
 - a. Opens a shadcn/ui modal listing all the columns mentioned in #3, with a checkmark in front of them.
 - b. Users can uncheck the select box to remove the column from the table or vice-versa to add it.
 - c. Users can enter a name for their customized table view with the selected set of columns and their order, to save to local storage.
 - d. Schema for customized table view would simply be an array in order of what column has to appear first in the table.
 - e. All saved customized views will show in tabs above the table, alongside the default Trending/main tab.

Submission

This assignment should be accomplished in 48 hours of sharing. The primary deliverable is a GitHub repository containing the source code, with a commit history reflecting the development process.