Cryptonite

FRONT-END ASSIGNMENT REVIEW ROUND - GROWW

Problem Statement:

Create a Web Application for a Cryptocurrency Tracker Website.

Overview

This document outlines the creation of a web application for a cryptocurrency tracker website using Next.js.

Objectives

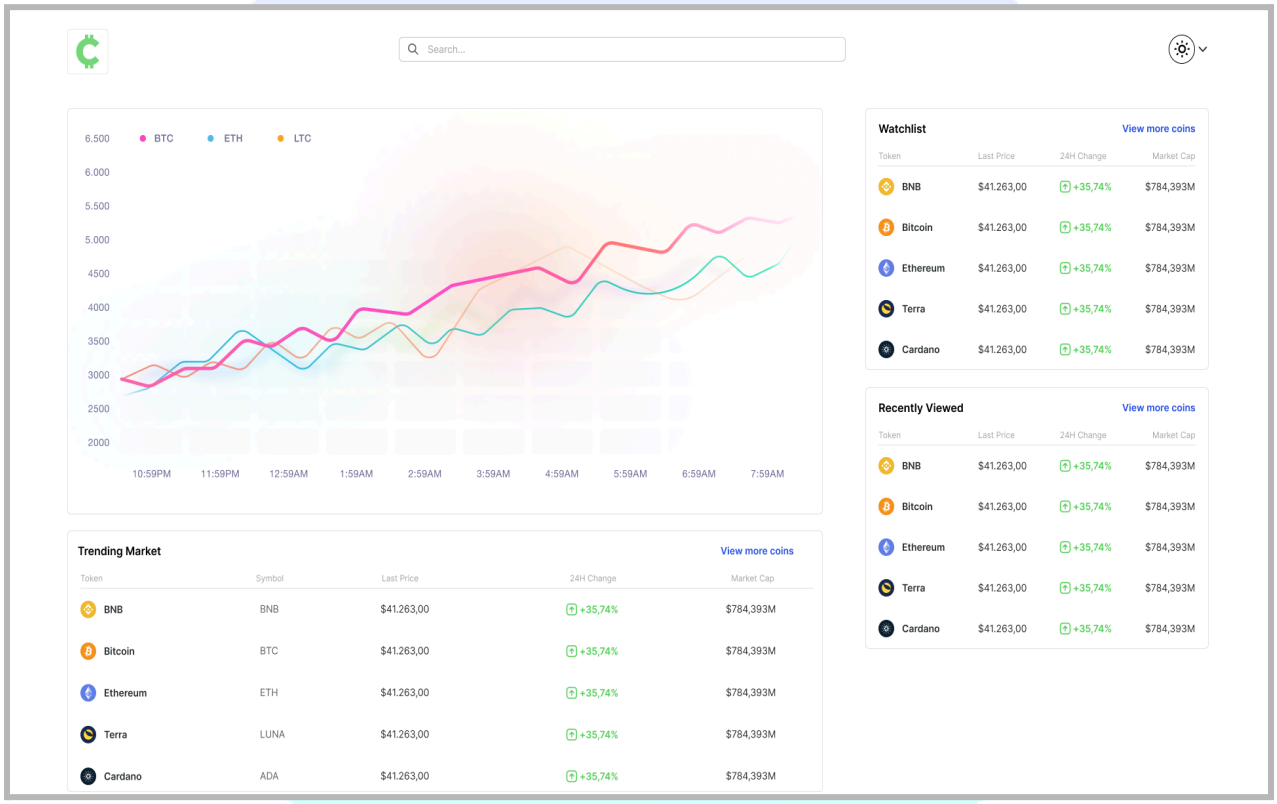
1. Develop a user-friendly web application to track various cryptocurrencies.
2. Provide real-time updates on cryptocurrency prices.
3. Offer detailed information on individual cryptocurrencies, including historical data and market trends.

Technologies

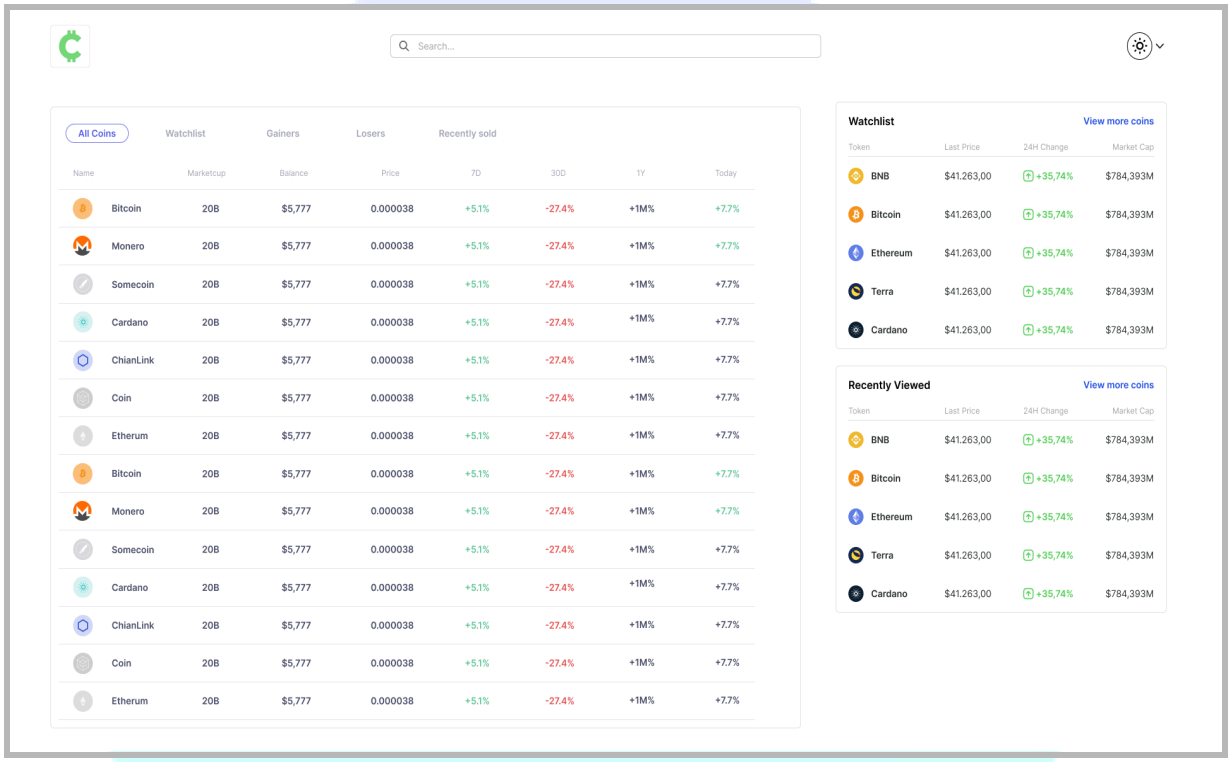
- **Frontend Framework:** Next.js
- **Styling:** CSS, Tailwind CSS, Styled-Components
- **State Management:** Redux/Redux Toolkit, Zustand, Jotai etc.
- **Deployment:** Vercel, Netlify, or AWS

Features

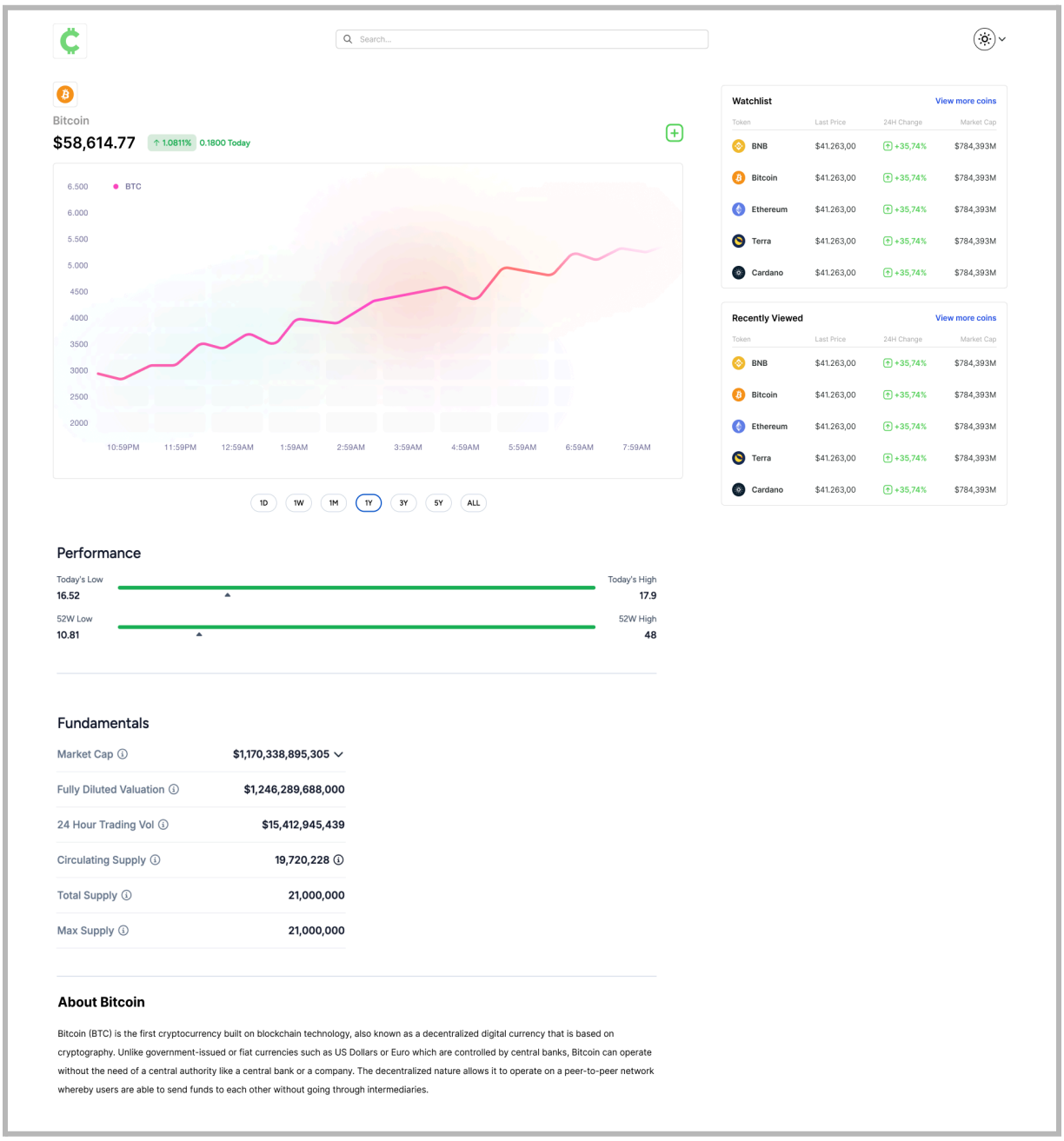
1. **Homepage:**
    - **Global Market Cap Chart:** Display a Line / candle graph showing the global market cap data for cryptocurrencies.
    - **Public Companies Holdings:** Display information about public companies holding Bitcoin and Ethereum.



2. **Explore Page:**
    - **Paginated Coin List/Grid:** Display a paginated list or grid of cryptocurrencies. Each page contains a specified number (20 items) of items with navigation to load more.
    - **Navigation:** Clicking on a card routes the user to the product page of the selected cryptocurrency.



3. **Product Page:**
    - Display basic information about the selected cryptocurrency.
    - Show a candle / line graph of the cryptocurrency's price over time.



4. **Common Header:**
    - Display the application name.
    - Include a search bar that shows suggested cryptocurrencies as the user types.
    - **Draggable Watchlist:** Implement a drag-and-drop functionality that allows users to easily add coins to their watchlist by dragging and dropping them.

API Integration

- Use a reliable cryptocurrency API (e.g. CoinGecko API) to fetch real-time data and historical data.
- Use additional APIs to fetch data on public companies holding Bitcoin and Ethereum (if required).

Here are the points to keep in check:

- You'll need an API key to access the endpoints. Generate an API key for the app and check the limits on requests per minute and requests per day.
- You can use any state management library (Redux/Redux Toolkit, Zustand, Jotai, etc.).
- Handle loading, error, and empty states for all cases where necessary.
- Follow a standard, well-defined folder structure.
- Ensure the UI is responsive and supports multiple screen sizes.
- Deploy the working code on any website hosting platform (GitHub, Vercel, Netlify, Firebase, etc.).
- Ensure you do not use any utility library such as underscore.js, lodash, day.js, etc.
- You can use a third-party library for line graphs.
- Cache API responses (with expiration) for all the pages.
- Try not to create the exact same UI which is provided. Explore and use your creativity to come up with a cool yet usable UI.

Brownie Points:

Note:

Focus on the basic functionality first. Once you have working basic functionality you can start exploring the below points:

1. **Dynamic Theme Switching**: Implement a dynamic theme-switching feature that allows users to toggle between Light and Dark modes seamlessly.

2. **Server-Side Rendering with Next.js**: Utilize Next.js's server-side rendering capabilities.

3. **Search Bar with Suggestions**: Integrate a feature in the search bar that displays recently searched items as suggested items.

4. **Mock Live Price Ticker**: Simulate a live price ticker using a Pub-Sub mechanism to mock a socket connection, providing real-time updates on product page only (live update after 1 min.)

5. **Drag-and-Drop to Watchlist**: Implement a drag-and-drop functionality that allows users to easily add coins to their watchlist by dragging and dropping them.

FAQ:

API and Data Handling

Q: What happens if the API request limit is reached?
A: If the API request limit is reached, the website may temporarily stop fetching new data until the limit resets. You may see a message indicating that the API limit has been reached and suggesting you try again later.

Q: How do I create a new API key?
A: To create a new API key, follow these steps (link):

1. Visit the API provider's website (e.g., CoinGecko).
2. Sign in to your account or create a new account if you don't have one.
3. Navigate to the API section or dashboard.
4. Look for an option to generate a new API key.
5. Follow the prompts to create and copy the new API key.
6. Update your web application's configuration with the new API key.

Q: What are the common API errors and rate limits?

A: Common API errors include exceeding the rate limit, invalid API keys, and network issues. Each API provider has specific rate limits, which are typically documented on their website. For more information on common errors and rate limits, please follow the link provided by the API provider. (link)

Q: Who can I contact for support or feedback?

A: For support or feedback, you can contact the team through the provided contact information in the email.

All the best!