Hyperion Code Challenge

Task Overview

The goal of this task is to learn MCP (Model-Consumer-Provider) using TypeScript and Vercel's AI SDK. The task is to implement an MCP server that can be added to a Claude desktop . This server should be able to:

- Accept requests to check the prices of specific crypto assets.
- Determine whether the price of a given crypto asset is in a *trending* or *ranging* regime on the 1-hour chart, using both price data and technical analysis (TA) indicators.

Requirements

Tech Stack (links are bellow):

- TypeScript
- Vercel's AI SDK
- ModelContextProtocol package
- Use either Cursor, Claude with Intlij or basically any agentic enabled editor (AI assistance recommended for the development process).

Data Sources:

- You can fetch the price and technical analysis (TA) data using the free versions of the CoinMarketCap API and TAAPI.
- You will need to parse the price and technical analysis data to decide if the asset is trending or ranging.

Technical Indicators:

Use TA indicators to help determine whether the asset is trending or ranging. Some possible indicators include:

- **Moving Averages** (SMA, EMA)
- **Relative Strength Index (RSI)**
- **MACD (Moving Average Convergence Divergence)**
- **Bollinger Bands**
- **Average True Range (ATR)**
- **Stochastic Oscillator**

These indicators can help identify price momentum, overbought/oversold conditions, and volatility, all of which can be used to classify the asset's regime.

Trending vs. Ranging:

- *Trending*: The price of the asset is moving significantly in one direction (either upward or downward) on the 1-hour chart.
- *Ranging*: The price is relatively stable, moving within a defined range with minor fluctuations on the 1-hour chart.

Note: If AI is used to determine whether the asset is in either the trending or ranging regime, that's a *huge plus*! You can leverage AI to analyze the chart data and automatically classify the asset.

Hosting & Docker:

- The project should be hosted (or at least able to start with a Docker Compose command). You should provide clear instructions for how the project can be started.

Code Quality:

- Ensure the code is readable and well-structured.
- Set up linting for the project to ensure code quality.

Deliverables:

- A working server that can be queried for crypto asset prices and its trend/range status.
- Documentation explaining how to run the project locally or in a hosted environment (with Docker Compose support).
- Linting configured for the project.
- Links or resources for learning about MCP and relevant technologies:
  - MCP Concept Overview
  - Vercel AI SDK Documentation
  - ModelContextProtocol SDK Documentation
  - CoinMarketCap API Documentation
  - TAAPI Documentation