Task: Stock Market Data Visualization with Vue.js

Project Overview:

You are tasked with creating a web application using Vue.js that allows users to fetch and visualize mock stock market data generated by a web server. You should also ensure the code is well-structured and maintainable.

Requirements:

1. Server setup:

- Create a simple web application (using exressJS/fastify server) that generates mock stock market data and exposes an HTTP endpoint that returns this mock data (/api/search?symbol=$symbol&period=$period).
- The API should take the two query params and return structured data as JSON.
- The data should include attributes such as date/time, close price, high price, low price, and volume.

2. User interface:

- Create a minimalistic and clean UI in Vue.js, including the following elements:
- Input field for selecting a stock instrument.
- Dropdown for selecting either daily or hourly interval data.
- A "Fetch Data" button to trigger the data retrieval and chart rendering.

3. Data fetching:

- Use Vue.js methods to make an HTTP request to the Fastify server's endpoint for fetching mock stock market data.
- Based on the selected interval (daily or hourly), fetch the corresponding data from the Fastify server.
- Handle API errors gracefully and display appropriate error messages to the user.

4. Data visualization:

- Use the Lightweight Charts (https://in.tradingview.com/lightweight-charts/) within your Vue.js components to create a candlestick chart or line chart.
- Plot the mock data points based on date/time and price values of the symbols.
- Ensure that the chart is responsive, interactive and allows users to zoom in and out.

5. Styling:

- Style the user interface to make it visually appealing and user-friendly.
- Ensure that the design is responsive and looks good on both desktop and mobile devices.

6. Submission:

- Commit the code to a repository on Github and host the application somewhere (eg: Heroku, AWS, DigitalOcean etc.) and share the links.

*You will be evaluated based on the completeness of the task, code quality, best practices, UI/UX design and error handling.