Assignment: Build an Order Book Visualization and Order Entry UI

Overview

The objective is to thoroughly understand a trader's latest feature request, design a solution, and then implement it into a React-based UI application. Please submit the finalized solution in a zip file, using the provided template as the basis. The task is expected to be completed within a maximum time frame of 3 hours, excluding any bonus features.

Trader's Requirement

As a cryptocurrency trader, I require a user interface to access orderbook data for the Bitcoin (BTC) and Ethereum (ETH) spot trading pairs. The interface should enable me to select an asset and view its order book, presenting bid and ask prices in an easily understandable format**.** Additionally, I need an order entry form where I can specify the details of a limit order for either buying or selling the chosen asset, including the desired quantity. Basic validation must be in place to ensure that negative or zero values are not accepted in the quantity field. Any successful or erroneous actions should be communicated back on the UI.

Provided

- ➔ A create-react-app template is provided for the solution implementation. The submission requires implementation of the UI code (located in the src folder) and your UX designs (you can put them in the ux folder).
- ➔ A mock server is also included (located in the server folder). Please use the provided endpoints for your solutions; no modifications to the server code are necessary. There is already a proxy to the endpoints. These should also serve as a guide to the required data (for POSTs).

Deliverables

- ➔ Share UX designs (paper, wireframe, Figma/InVision) to effectively communicate concepts before UI development. There is a ux folder you can use for this.
- ➔ Offer any notes or explanation in the README (Candidate Notes section) that you think is relevant, such as scope of solution, tests, designs, rationale behind your choice of libraries or tools, etc.
- ➔ Please upload your solution to a personal GitHub repository, deploy it on Vercel, and share both the GitHub link and the live application URL.

Evaluation

- ➔ Construct the UI utilizing React along with any design library of your preference.
- ➔ Utilize React props and hooks correctly & effectively for optimal performance, ensuring clear component responsibilities and state management.
- ➔ Employ the mock server API to acquire orderbook data and initiate orders.
- ➔ Ensuring alignment with the trader's specified requirements and functionalities.
- ➔ Assessing the clarity and ease of user interface design and user experience.
- ➔ Evaluating the quality, structure, organization, and presence of tests.
- ➔ Submission of UX designs to effectively convey concepts before UI development.
- ➔ Clear explanation of library/tool choices. There are no restrictions on the number of libraries used; however, simplicity is encouraged over over-engineering.

Bonus Features

- ★ **Display Trades:** Allow traders to view their trades. Note: no API implementation.
- ★ **Enhanced Order Flexibility:** Provide an order type option for traders to create *market* orders in addition to *limit* orders.
- ★ **Efficient Order Placement:** Implement a feature to swiftly place orders directly from the orderbook. Upon clicking on a limit price, the system should fill the order entry form with the correct side and the clicked limit price, and immediately send the order. The quantity should be pre-filled by the trader in the order entry form.

Definitions

- ➔ Orderbook: A record of all buy and sell orders for a particular asset on an exchange. It typically includes information such as price, quantity, and order type (buy or sell). Please read this short guide on how a typical orderbook UI works.
- ➔ Bid and Ask Prices: The highest price a buyer is willing to pay (bid) and the lowest price a seller is willing to accept (ask) for a particular asset.
- ➔ Order Entry Form: A form or interface where users can input details of an order, such as the asset to buy or sell, the quantity, and the price. It also shows the notional (quantity x price). Users can typically enter 2 of these (quantity, price, notional) and the 3rd input can be autocalculated.
- ➔ Limit Order: An order to buy or sell an asset at a specified price or better. For example, a buy limit order specifies the maximum price the trader is willing to pay, while a sell limit order specifies the minimum price the trader is willing to accept. The user will need to enter the quantity or notional and price.
- ➔ Market Order: An order to buy or sell an instrument at the best available price in the market, executed immediately. The user doesn’t need to enter the price.

Don't hesitate to contact us if you have any inquiries. Thank you!