



# Full-stack practical

👤 Created by	<span>T</span> Tim Meggs
🕒 Created time	@April 15, 2024 12:54 PM
☰ fund	Hiring

## Streaming Order Book Web App

### Objective

Your task is to build a basic web app that displays live streaming order book data for a crypto coin (truncated; limit to 5 depth levels) . This will involve subscribing to a WebSocket endpoint provided by us, handling live data streams, and updating the web app's UI in real time based on user interactions.

For a high-level explanation of what an order book is, this is a good starting place: [Investopedia Order Book](#)

### Requirements

#### Data Source

- The below is a WebSocket endpoint to which your app should subscribe to receive live truncated order book data (please keep this information private):
  - `wss://mock.lo.tech:8443/ws/orderbook`
- The live data stream include information for 5 different crypto coins from two different exchanges delivered in a standardised format. The schema for payload on the WebSocket is as follows:

```
{  
  "timestamp": "float",
```

```
"exchange": "string",
"coin": "string",
"bids": [
  ["float", "float"],
  ["float", "float"],
  ["float", "float"],
  ["float", "float"],
  ["float", "float"]
],
"asks": [
  ["float", "float"],
  ["float", "float"],
  ["float", "float"],
  ["float", "float"],
  ["float", "float"]
]
}
```

## User Interface requirements

- **Coin and Exchange Selection:** Users must be able to select which token's order book to display from those available. This selection should be facilitated through a user-friendly interface allowing for the selection of crypto coins and their respective exchanges.
- **Live Data Display:** The selected order books must update in real-time with each new piece of relevant data received from the WebSocket. The UI should clearly reflect changes to the order book in a manner that is easy to understand.

## Bonus (Optional)

- **Multiple Order Books:** Allow users to view order books for more than one token simultaneously, if time permits. Adding and removing order books should be handled gracefully in the UI.
- **UI Enhancements:** Add features like highlighting the highest bid and lowest ask prices.

## Submission Guidelines

- Submit your project as a GitHub repository link.
- Your README.md should include:
  - Setup and running instructions.
  - A brief discussion of your technology choices and any assumptions you made.

## Evaluation Criteria

- **Functionality:** Does the app meet the specified requirements with smooth real-time updates?
- **Code Quality:** Is the code well-structured, clean, and commented?
- **UI/UX Design:** How intuitive and user-friendly is the application interface?
- **Technical Choices:** Are the chosen technologies and implementation strategies effectively justified?