

Frontend Assessment - 2

Real-Time Stock Market Updates

In this assessment, your objective is to create a real-time stock market application using HTML/SCSS frontend and integrate it with a Node.js backend server using sockets for live updates.

Requirements

→ Viewport Layout:

- ◆ The viewport should consist of two tabs with width ratios of 3:1
 - Tab1 (Main) = 75% and Tab2 (Side) = 25%
 - Main will have a bar-graph container
 - Height of bar-graph container: 500px
 - Max number of bars : (Main-width/20px)
 - ◆ i.e. Each bar width: 20px
- ◆ **Graded-stretch goal :**
Implement a visual grid on the bar-graph container.
 - Each cell of the grid will have width : 100px and height : 125px
 - 5 bars can be accommodated in a cell-column
 - The X-axis must have hard-coded scale values 0, 100, 200 ...

→ Color Representation:

- ◆ If a negative change is received, display the bar in red.
- ◆ For a positive change, display the bar in green.

→ Buy and Sell Buttons:

- ◆ They should take the value from the textbox and perform the desired functionality.
- ◆ On every button operation, an API call must be made to the server.
 - i) Store the history in-memory at the nodejs server
 - ii) Populate the history at side container

→ Dashboard

- ◆ On page load,
 - i) API call to fetch the base data.

- Name of stock and Initial price. Which will be populated on the screen.
 - ii) Socket begins sending real-time data to populate bar graph
 - iii) API call to fetch history and populate side-container. The history format to be followed is shown in the mock UI.
- ◆ After every 5 seconds, the server must send a socket message indicating the increase/decrease in price amount.
 - The price amount will range from 0 to 500 only. To fit the height of the bar-graph container.
 - On every socket message, a bar (20px) must be populated on the X-Axis with red or green color.
 - The height of the bar denotes the current price received from the socket message.
 - To plot, assume 1px = Re.1 and set graph Y value accordingly.
 - Graph X value denotes the time interval when a message was received.
 - **Note: The dotted lines should be behind the bars**

Assets:

1. Font-family:
<https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap>
2. Zomato logo:
<https://mir-s3-cdn-cf.behance.net/projects/404/af1307136664867.Y3JvcCw1NzUzLDQ1MDAsNDIsMA.jpg>