SSE

This project creates a real-time stock price chart using Server-Sent Events (SSE) with HTML, JavaScript, Express.js, and Chart.js. The HTML file sets up a canvas to display the live stock price chart. JavaScript utilizes Chart.js to render the chart and Event Source to receive updates from the server. Express.js handles the server-side logic, generating and broadcasting random stock price data to clients every 500 milliseconds. Clients receive updates via SSE, enabling real-time visualization of stock prices on the chart.

App.js

const express = require('express')

const cors = require('cors')

const PORT = process.env.PORT || 3000

const clients = []

const app = express()

app.use(cors())

app.use(express.json())

app.get('/', (req, res) => {

    res.send('Hello I am alive!')

})

app.get('/price', (req, res) => {

    res.set({

        "Content-Type": "text/event-stream"

    })

    clients.push(res)

    res.on('close', () => {

        clients.splice(clients.indexOf(res), 1)

    })

})

let previousPrice = 500

function generateStockPrice(previousPrice) {

    const changePercent = (Math.random() - 5) / 50;

    const newPrice = previousPrice \* (1 + changePercent);

    previousPrice = newPrice;

*return* Math.round(newPrice \* 100) / 100;

}

setInterval(() => {

    const data = {

        time: new Date().getTime(),

        price: generateStockPrice(previousPrice)

    }

    clients.forEach(client => {

        client.write(`data: ${JSON.stringify(data)}\n\n`)

    })

}, 500)

app.listen(PORT, () => {

    console.log(`Server is running on http://localhost:${PORT}`)

})

Index.js

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Server Sent events</title>

    <style>

        body {

            margin: 0;

            padding: 0;

            font-family: Arial, sans-serif;

        }

*.container* {

            position: relative;

            width: 100%;

            height: 100vh;

            display: flex;

            justify-content: center;

            align-items: center;

        }

*.content* {

            position: relative;

            padding: 20px;

            border-radius: 20px;

            background-color: rgba(244, 244, 246, 1);

        }

    </style>

</head>

<body>

    <div *class*="container">

        <div *class*="content-container">

            <h2>Live Stock Price Chart</h2>

            <div *class*="content">

                <canvas *id*="line-chart" *width*="599" *height*="336"></canvas>

            </div>

        </div>

    </div>

    <script *src*="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <script>

        const line\_chart = new Chart(document.getElementById("line-chart"), {

            type: 'line',

            data: {

                labels: [],

                datasets: [

                    {

                        data: [],

                        label: "Stock Price",

                        borderColor: "#8e5ea2",

                        fill: true,

                        backgroundColor: "rgba(142, 94, 162, 0.2)"

                    }

                ]

            },

            options: {

                title: {

                    display: true,

                    text: 'Stock Price'

                },

                scales: {

                    y: {

                        title: {

                            display: true,

                            text: 'Price'

                        },

                        suggestedMin: 420,

                    },

                    x: {

                        title: {

                            display: true,

                            text: 'Time'

                        }

                    }

                },

                hover: {

                    mode: 'nearest',

                    intersect: true

                },

                tooltips: {

                    mode: 'index',

                    intersect: false

                }

            }

        });

        const eventSource = new EventSource("http://localhost:3000/price");

        eventSource.onmessage = (event) => {

            const data = JSON.parse(event.data);

            const time = new Date(data.time).toLocaleTimeString();

            const price = data.price;

            line\_chart.data.labels.push(time);

            line\_chart.data.datasets[0].data.push(price);

            line\_chart.update();

        };

    </script>

</body>

</html>

{

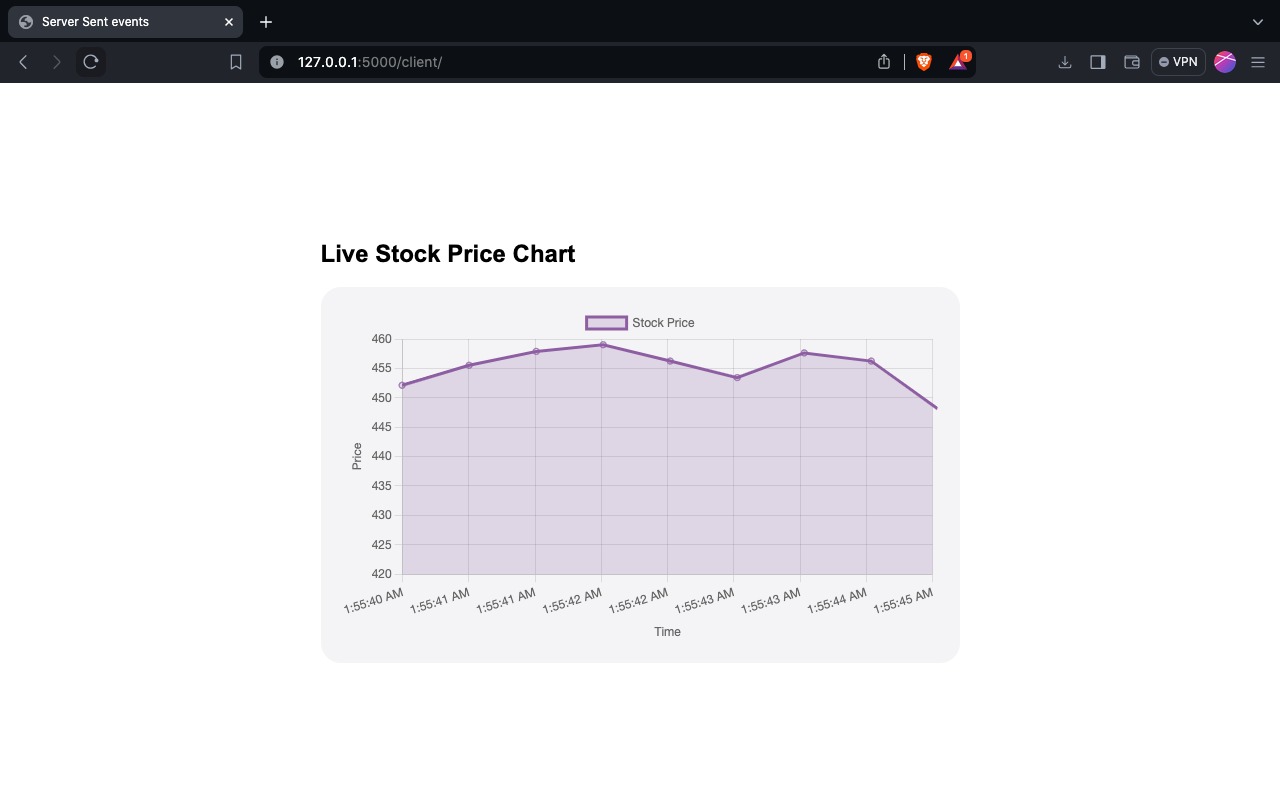
  "dependencies": {

    "cors": "^2.8.5",

    "express": "^4.18.3"

  }

}



A graph with purple lines

Description automatically generated