# HTML code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Binance Market Data WebSocket</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Binance Market Data - Candlestick Chart</h1>

<div class="controls">

<label for="cryptocurrency">Select Cryptocurrency:</label>

<select id="cryptocurrency">

<option value="ethusdt">ETH/USDT</option>

<option value="bnbusdt">BNB/USDT</option>

<option value="dotusdt">DOT/USDT</option>

</select>

<label for="interval">Select Interval:</label>

<select id="interval">

<option value="1m">1 Minute</option>

<option value="3m">3 Minutes</option> <option value="5m">5 Minutes</option>

</select>

</div>

<canvas id="candlestickChart"></canvas>

</div>

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

<script src="app.js"></script>

</body>

</html>

# CSS code

Body

{ font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0; padding: 0;

}

.container

{ max-width: 900px; margin: 50px auto; padding: 20px; background: #fff; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 { text-align: center;

}

.controls

{

display: flex;

justify-content: space-around; margin-bottom: 20px;

}

canvas { width: 100%; height: 400px;

}

# Javascript code

let socket; let chart;

let chartData = {};

let selectedCoin ='ethusdt';

let selectedInterval = '1m';

if (localStorage.getItem('chartData')) { chartData = JSON.parse(localStorage.getItem('chartData'));

}

const ctx = document.getElementById('candlestickChart').getContext('2d');

initChart(data) {

if (chart) { chart.destroy();

}

chart = new Chart(ctx, { type: 'candlestick', data: { datasets: [{ label: selectedCoin.toUpperCase(), data: data

}]

},

options: { responsive: true

}

});

}

connectWebSocket() { const symbol = document.getElementById('cryptocurrency').value; const interval = document.getElementById('interval').value;

const url = `wss://stream.binance.com:9443/ws/${symbol}@kline\_${interval}`;

if (socket) { socket.close();

}

socket = new WebSocket(url);

socket.onmessage = function (event) { const message = JSON.parse(event.data); const candlestick = message.k;

if (candlestick.x) { // candlestick is closed const candleData = { t: candlestick.t, // timestamp o: candlestick.o, // open h: candlestick.h, // high l: candlestick.l, // low c: candlestick.c // close

};

if (!chartData[selectedCoin])

{ chartData[selectedCoin] = [];

}

chartData[selectedCoin].push(candleData); localStorage.setItem('chartData', JSON.stringify(chartData));

initChart(chartData[selectedCoin]);

}

};

socket.onerror = function (error) { console.log('WebSocket error:', error);

};

socket.onclose = function () { console.log('WebSocket connection closed');

};

}

document.getElementById('cryptocurrency').addEventListener('change', (e) => { selectedCoin = e.target.value; if (chartData[selectedCoin]) { initChart(chartData[selectedCoin]);

}

connectWebSocket();

});

document.getElementById('interval').addEventListener('change', (e) => { selectedInterval = e.target.value; connectWebSocket();

});

if (chartData[selectedCoin]) { initChart(chartData[selectedCoin]);

}

connectWebSocket();