For Stock prediction codes as follows

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Stock Market Prediction</title>

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

</head>

<body>

<header>

<h1>Stock Market Prediction</h1>

</header>

<main>

<section id="stock-data">

<h2>Stock Data</h2>

<table id="stock-table">

<thead>

<tr>

<th>Symbol</th>

<th>Name</th>

<th>Price</th>

<th>Change</th>

</tr>

</thead>

<tbody id="stock-table-body">

</tbody>

</table>

</section>

<section id="prediction">

<h2>Prediction</h2>

<p id="prediction-text"></p>

</section>

<section id="search">

<h2>Search</h2>

<input id="search-input" type="text" placeholder="Enter stock symbol">

<button id="search-button">Search</button>

</section>

</main>

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

</body>

</html>

**CSS**

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #333;

color: #fff;

padding: 1em;

text-align: center;

}

main {

display: flex;

flex-direction: column;

align-items: center;

padding: 2em;

}

section {

background-color: #f7f7f7;

padding: 1em;

margin-bottom: 2em;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

table {

width: 100%;

border-collapse: collapse;

}

th, td {

border: 1px solid #ddd;

padding: 0.5em;

text-align: left;

}

th {

background-color: #333;

color: #fff;

}

#prediction-text {

font-size: 1.5em;

font-weight: bold;

color: #333;

}

#search-input {

padding: 0.5em;

border: 1px solid #ddd;

border-radius: 0.5em;

}

#search-button {

background-color: #333;

color: #fff;

padding: 0.5em 1em;

border: none;

border-radius: 0.5em;

cursor: pointer;

}

#search-button:hover {

background-color: #444;

}

**JS**

const allStocks = [

{ symbol: "AAPL", name: "Apple Inc.", price: 150.54, change: 0.5 },

{ symbol: "GOOGL", name: "Alphabet Inc.", price: 2800.66, change: -1.2 },

{ symbol: "AMZN", name: "Amazon.com Inc.", price: 3401.46, change: 2.1 },

// ...

];

const watchlistStocks = [];

const stockTableBody = document.getElementById("stock-table-body");

const watchlist = document.getElementById("watchlist");

const stockInfo = document.getElementById("stock-info");

const searchInput = document.getElementById("search-input");

const searchButton = document.getElementById("search-button");

function displayStockInfo(stock) {

const changeClass = stock.change >= 0 ? 'positive' : 'negative';

stockInfo.innerHTML = `

<h3>${stock.name} (${stock.symbol})</h3>

<p>Price: $${stock.price.toFixed(2)}</p>

<p class="${changeClass}">Change: ${stock.change}%</p>

`;

}

function updateWatchlist() {

watchlist.innerHTML = '';

watchlistStocks.forEach((stock) => {

const li = document.createElement('li');

li.textContent = `${stock.name} (${stock.symbol}) `;

const removeButton = document.createElement('button');

removeButton.textContent = 'Remove';

removeButton.classList.add('remove');

removeButton.addEventListener('click', () => {

watchlistStocks = watchlistStocks.filter(item => item.symbol !== stock.symbol);

updateWatchlist();

displayStockTable(allStocks);

});

li.appendChild(removeButton);

watchlist.appendChild(li);

});

}

function displayStockTable(stocks) {

stockTableBody.innerHTML = '';

stocks.forEach((stock) => {

const changeClass = stock.change >= 0 ? 'positive' : 'negative';

const tr = document.createElement('tr');

tr.innerHTML = `

<td>${stock.name}</td>

<td>$${stock.price.toFixed(1)}</td>

<td class="${changeClass}">${stock.change}%</td>

<td class="actions"></td>

`;

const actionsTd = tr.querySelector('.actions');

if (!watchlistStocks.some(item => item.symbol === stock.symbol)) {

const addButton = document.createElement('button');

addButton.textContent = 'Add';

addButton.classList.add('add');

addButton.addEventListener('click', () => {

watchlistStocks.push(stock);

updateWatchlist();

displayStockTable(allStocks);

});

actionsTd.appendChild(addButton);

} else {

const removeButton = document.createElement('button');

removeButton.textContent = 'Remove';

removeButton.classList.add('remove');

removeButton.addEventListener('click', () => {

watchlistStocks = watchlistStocks.filter(item => item.symbol !== stock.symbol);

updateWatchlist();

displayStockTable(allStocks);

});

actionsTd.appendChild(removeButton);

}

tr.addEventListener('click', () => {

displayStockInfo(stock);

});

stockTableBody.appendChild(tr);

});

}

searchInput.addEventListener('input', (event) => {

const searchTerm = event.target.value.toLowerCase();

const filteredStocks = allStocks.filter(stock => stock.name.toLowerCase().includes(searchTerm) || stock.symbol.toLowerCase().includes(searchTerm));

displayStockTable(filteredStocks);

});

searchButton.addEventListener('click', () => {

const searchTerm = searchInput.value.toLowerCase();

const filteredStocks = allStocks.filter(stock => stock.name.toLowerCase().includes(searchTerm) || stock.symbol.toLowerCase().includes(searchTerm));

displayStockTable(filteredStocks);

});

displayStockTable(allStocks);

updateWatchlist();