

**The Superior University Lahore**

**Faculty of Computer Science & Information**

**Technology**

**Name: Alishba Haroon**

**Roll No: BSAI-116-4C**

**Date: 21 Mar 2025**

**Subject: PAI LAB**

**Explanation:**

This is the Stock Price Checker web application using HTML, CSS, and JavaScript. It allows users to enter a stock symbol, fetch its current price, and display the trend (uptrend/downtrend) based on a simulated previous price.

**How It Works**

1. **HTML Structure:**

* A user interface with two sections:
  + - Stock Input Form (#stock-form): Where users enter a stock symbol.
    - Stock Result Section (#stock-result): Displays stock price and trend after fetching data.
  + Buttons are used to check stock prices and go back to the input form.

1. **CSS Styling:**
   * Sets a background image.
   * Centers the content using Flexbox.
   * Styles buttons and input fields to have a modern look.
   * Green is used for uptrends and red for downtrends.
2. **JavaScript Functionality:**
   * fetchStock(): Gets the stock symbol input and sends an AJAX request (POST) to the backend (/get\_price) to retrieve the stock price.
     + If there’s an error (e.g., invalid symbol), it displays an error message.

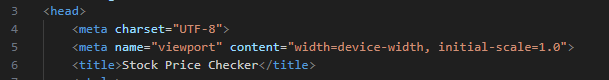
* Otherwise, it calculates a fake previous price (by randomly adjusting the price) and determines the trend (📈 or 📉).
  + goBack(): Allows users to go back to the stock input form.

**HTML**



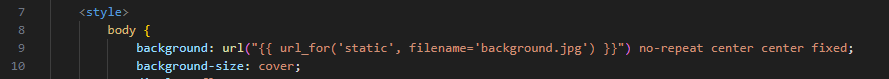
* + <!DOCTYPE html>: Declares the document as an HTML5 page.
  + <html lang="en">: Specifies that the document is in English.

1. **Head Section (<head>)**

****

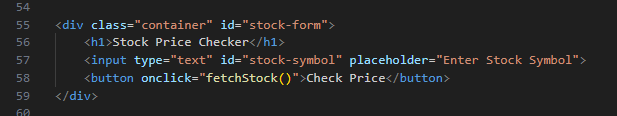
* + <meta charset="UTF-8">: Ensures support for all character sets (e.g., special symbols).
  + <meta name="viewport" content="width=device-width, initial-scale=1.0">: Makes the webpage responsive on mobile devices.
  + <title>Stock Price Checker</title>: Sets the title of the page (visible in the browser tab).

1. **Background Image**

****

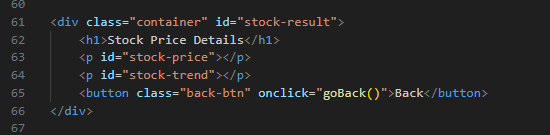
* url\_for('static', filename='background.jpg'): This Flask function loads a background image from the /static/ folder.
* background-size: cover;: Ensures the background covers the entire page.
* no-repeat center fixed;: Keeps the background fixed and centered.

1. **Body Section (<body>) & Stock Input Form (#stock-form)**

****

* ***<div class="container" id="stock-form">***
* Container that holds the stock input form.
* ***<h1>Stock Price Checker</h1>***
* Heading for the webpage.
* ***<input type="text" id="stock-symbol" placeholder="Enter Stock Symbol">***
* A text input field where the user enters the stock symbol.
* ***<button onclick="fetchStock()">Check Price</button>***
* A button that triggers the fetchStock() JavaScript function when clicked

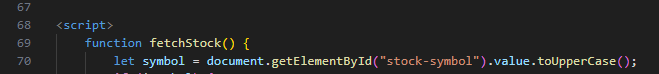
1. **Stock Result Section (#stock-result)**

****

* ***<div class="container" id="stock-result">***
* Another container that holds the stock result.
* ***<p id="stock-price"></p>***
* Displays the stock price (filled dynamically by JavaScript).
* ***<p id="stock-trend"></p>***
* Displays whether the stock is in an uptrend or downtrend.
* ***<button class="back-btn" onclick="goBack()">Back</button>***
* Back button that triggers goBack() to return to the input form.

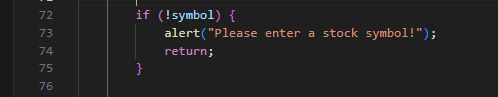
**JavaScript**

1. **fetchStock() Function (Fetching Stock Data)**

****

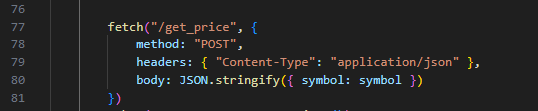
* Gets the stock symbol entered by the user from the input field (id="stock-symbol").
* Converts it to uppercase (toUpperCase()) to ensure uniform formatting.

1. **Validation Check**

****

* If the user has not entered anything, an alert is displayed: "Please enter a stock symbol!"
* The function stops execution (return;) if no input is provided.

1. **Making a POST Request to /get\_price**

****

* Sends a POST request to the Flask backend route /get\_price.
* The request includes JSON data containing the stock symbol entered by the user.
* Headers specify Content-Type: "application/json" to ensure the data is sent in JSON format.

1. **Handling the Response**

****

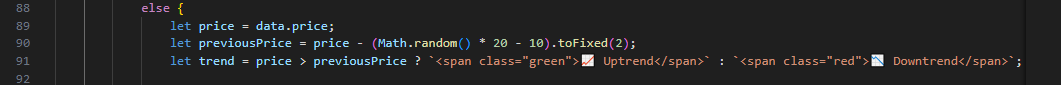
* Converts the response into JSON format.
* The data object contains either stock price information or an error message.

1. **Error Handling**

****

* If the backend returns an error (e.g., invalid stock symbol), the error message is displayed in red.

1. **Displaying Stock Price and Trend**

****

* Gets the stock price from data.price.
* Generates a fake previous price (randomly fluctuating within ±10 units) to simulate a trend.



* Compares the current price with the previous price:
* If current price is higher, it's an uptrend (📈).
* Otherwise, it's a downtrend (📉).

1. **Updating the UI**

****

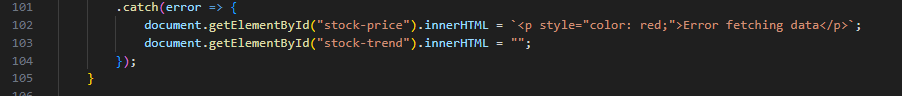
* Displays the stock price in gold color.
* Displays the trend (Uptrend or Downtrend) in green/red.

1. **Hiding Input Form & Showing Results**

****

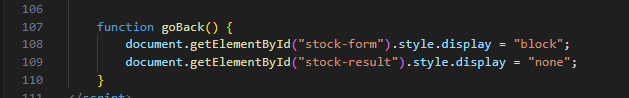
* Hides the stock input form (stock-form).
* Shows the stock result section (stock-result).

1. **catch(error) (Handling Fetch Errors)**

****

* If there is an error in fetching data (e.g., server down, network issue), it shows:
* "Error fetching data" message in red.
* Clears any trend data.

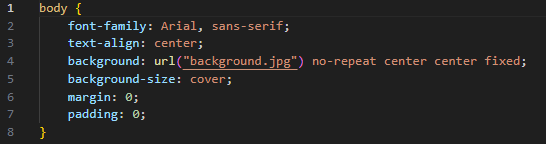
1. **goBack() Function (Returning to Input Form)**

****

* **Restores the stock input form (stock-form).**
* **Hides the stock result section (stock-result).**

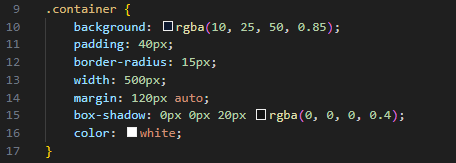
**CSS Explanation**

1. **Background Styling (body)**

****

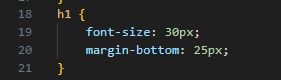
* Background Image: background.jpg is used as the page background.
* Centered Content: display: flex; justify-content: center; align-items: center;
* Full Page Height: height: 100vh; (100% of the viewport height).
* Removes Margin: margin: 0;

1. **Container Styling**

****

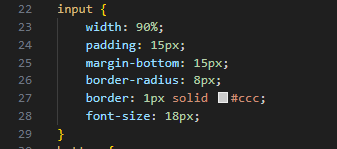
* Semi-transparent background (rgba(0, 0, 0, 0.7)) for a dark overlay.
* Padding (30px) gives space around the content.
* Rounded corners (border-radius: 12px) make it look modern.
* Text centered (text-align: center).
* Fixed width (width: 400px) makes it a neat box.
* Text color white (color: white;).

1. **Headings (h1)**

****

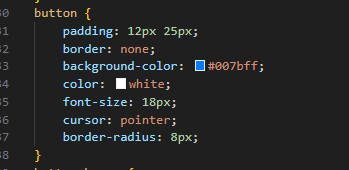
* Font size is 24px.

1. **Input & Button Styling**

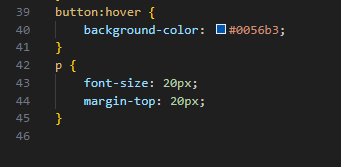
****

* width: 80%: Makes input and button responsive.
* Padding (10px): Gives better touch area.
* Rounded corners (border-radius: 5px).
* No borders (border: none;)

1. **Button Styling (Check Price & Back)**

****

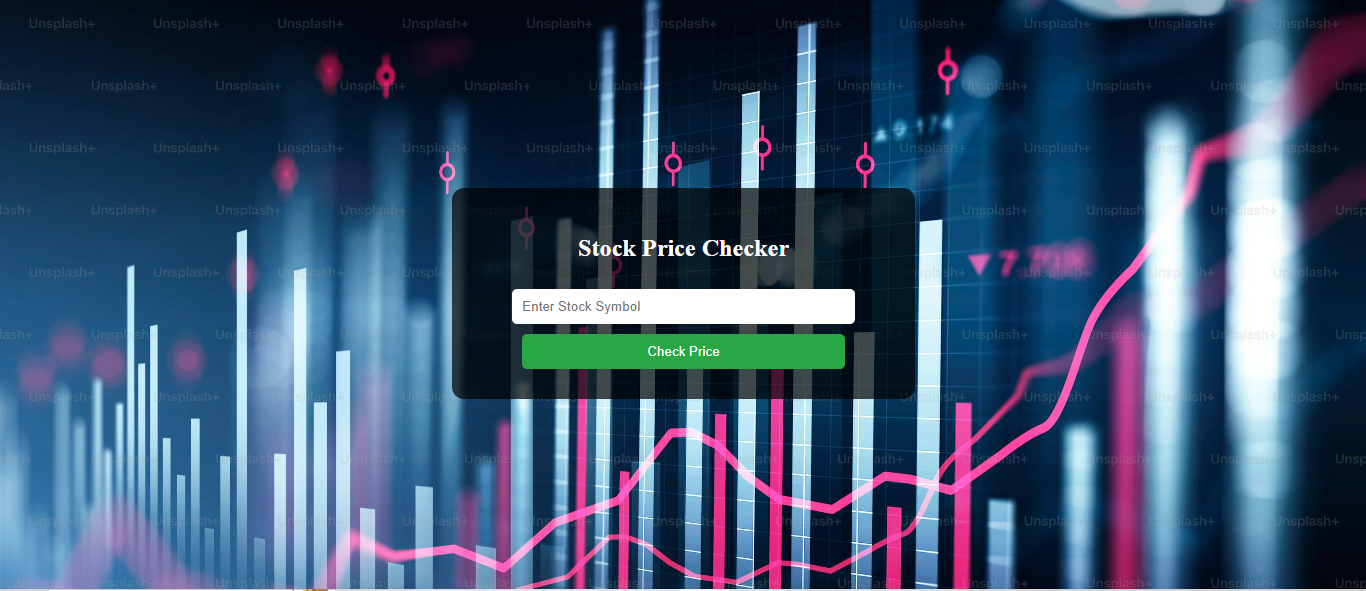
* Green background (#28a745) for the "Check Price" button.
* White text (color: white;).
* Pointer cursor (cursor: pointer;) when hovered.



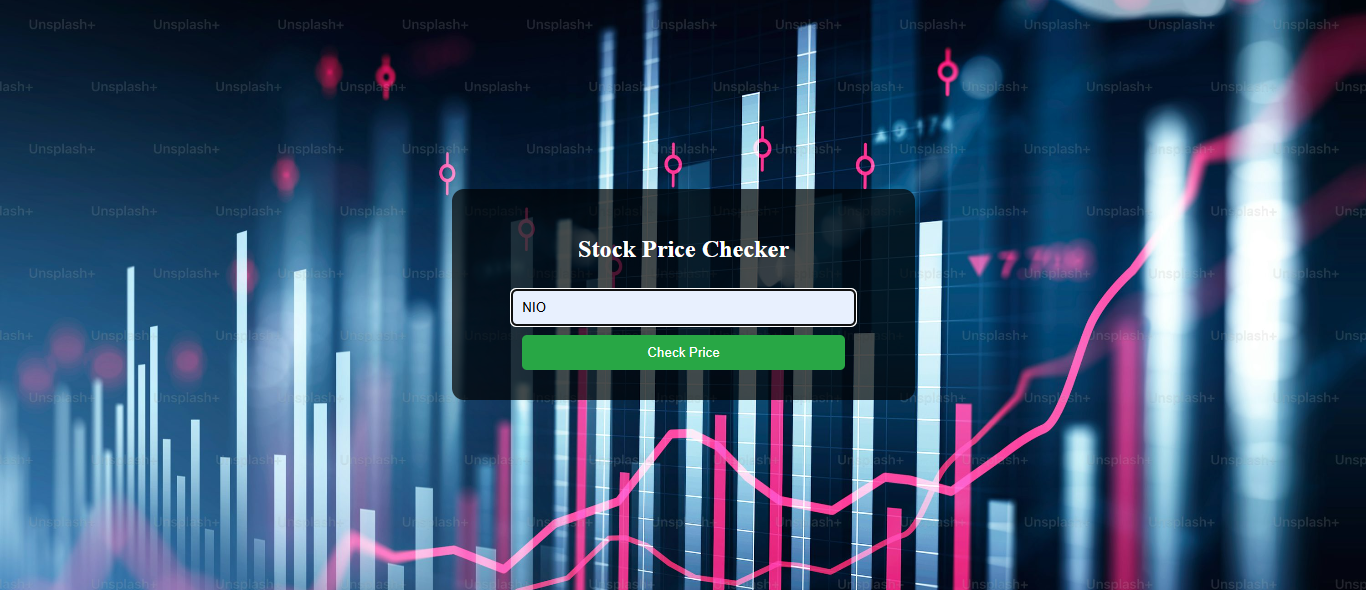
* Blue background (#007bff) for the Back button.
* Adds margin-top (margin-top: 10px;) to separate it from results.

**OUTPUT:**

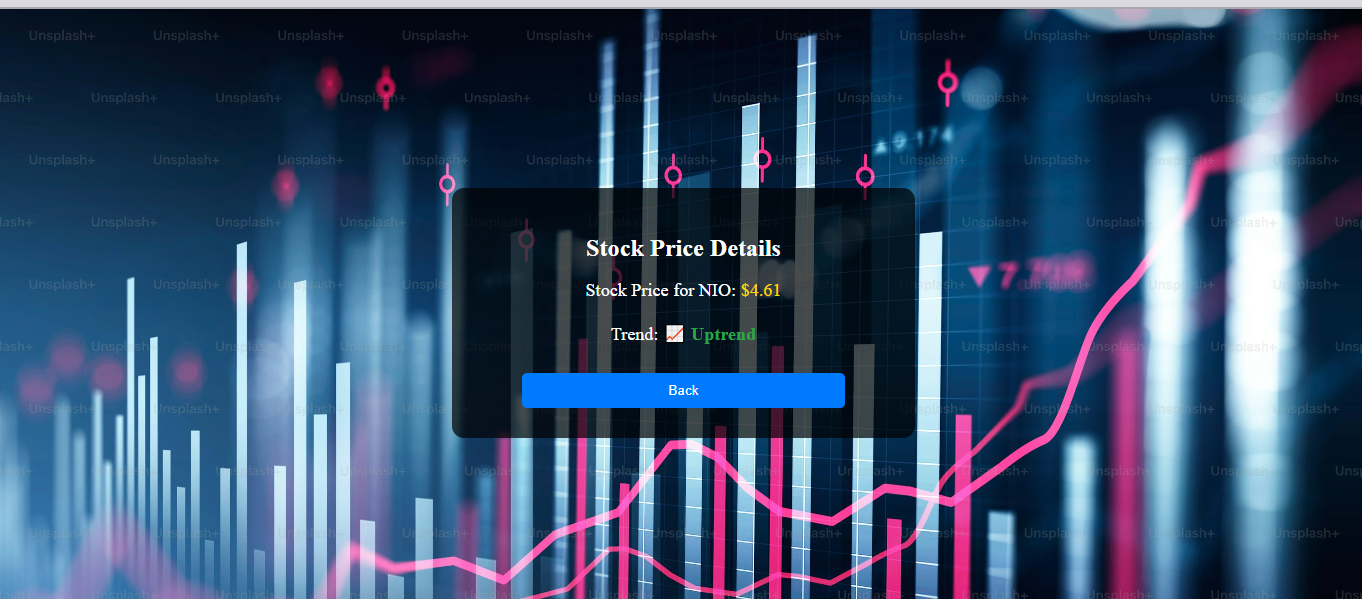
* **Main Page**

****

* **Enter the Nio Stock Symbol**

****

* **The application provides stock price details and indicates whether the stock is experiencing an uptrend or a downtrend.**

****