**Front-End UI/UX Mini Project**

**TITLE PAGE**

* **Project Title:** Stock Market Website
* **Submitted By:**
  + **Team Members:** Anshika Sharon Sonwani, Freida B. Rodrigues, Evan Devanand P A
  + **Roll Numbers: 2460334, 2460367, 2460365**
  + **College Email ID:**
  + **[anshika.sharon@btech.christuniversity.in](mailto:anshika.sharon@btech.christuniversity.in  )**

[**freida.b@btech.christuniversity.in**](mailto:freida.b@btech.christuniversity.in)

[**evan.devanand@btech.christuniversity.in**](mailto:evan.devanand@btech.christuniversity.in)

* **Course:** *UI/UX Design Fundamentals*
* **Instructor Name:** Mr. Narendra
* **Institution:** *Christ University*
* **Date of Submission:** 26/09/2025

**Abstract**

This project centers on designing and developing a responsive stock market website that enables users to view, track, and analyze real-time stock data through a clean and intuitive interface. Leveraging technologies such as HTML, CSS, JavaScript, Bootstrap, and jQuery, the application emphasizes usability, responsiveness, and accessibility across devices. Users can explore dynamic stock listings featuring company names, ticker symbols, and current prices, with robust search and filter capabilities by sector, market cap, or performance. Each stock entry links to a detailed view showcasing historical price charts, company profiles, and key financial metrics. Real-time data integration ensures that stock prices and analytics update dynamically, offering users a seamless and engaging experience. The final product is a visually appealing, front-end application that demonstrates proficiency in API integration, DOM manipulation, and responsive design principles tailored to financial data visualization.

**Objectives**

* Design a clean, modern, and visually appealing interface for financial information and analytics.
* Display stock listings with company names, ticker symbols, and real-time price updates.
* Understand how to interact with external APIs to fetch and display live stock market data.
* Create detailed stock pages showing price history, interactive charts, company profiles, and key financial metrics.
* Ensure full responsiveness using Bootstrap for seamless viewing across desktops, tablets, and mobile devices.
* Use JavaScript and jQuery for dynamic content rendering, API integration, and user interaction.
* Enhance user experience with intuitive navigation, clear data visualization, and accessibility best practices.

**Scope of the Project**

* Develop a responsive front-end interface using HTML, CSS, JavaScript, Bootstrap, and jQuery.
* Integrate external APIs to fetch and display live stock market data dynamically.
* Enable users to open **detailed modal views** with charts and key metrics.
* Ensure seamless user experience across desktop, tablet, and mobile devices through responsive design.

**Tools & Technologies Used**

* **HTML5** – Structuring content and ensuring semantic clarity across the interface
* **CSS3** – Styling, responsive layout, and visual enhancements
* **JavaScript** – Dynamic content rendering and API integration
* **jQuery** – DOM manipulation and event handling
* **Bootstrap** – Responsive grid system and UI components
* **Stock Market Data API–** Fetching live market data and financial metrics
* **VS Code** – Code editing

**HTML, CSS, and JS Structure Overview**

**HTML Structure**

* **Main Page (index.html)**
  + **Slide-out navigation menu** from right side with close button and navigation links
  + **Floating animated shapes** as background decorative elements
  + **Features section** with three feature cards highlighting platform capabilities
  + **Watchlist section** for displaying user's tracked stocks
  + **Bootstrap modal** for detailed stock information with charts and metrics
* **Dashboard Page (dashboard.html)**
  + Header with brand, search input, and filters
  + Stock cards showing **company, ticker, price, and sector**
  + Detailed view with **charts and metrics** dynamically updated
  + Linked to style.css and script.js.
* **Semantic Elements Used**
  + <header>: Contains branding and navigation controls
  + <section>: Separates major content areas (hero, market, features, watchlist).
  + <nav>: Side menu with smooth slide transitions.
  + <div>: Card containers and layout wrappers.
  + <canvas>: Chart rendering areas for stock price visualization

**CSS Structure (style.css)**

* **CSS Custom Properties**
  + Resets for body and html, setting full height and font-family.
  + Base styles for consistent colors and fonts.
* **Layout**
  + .index-page and .dashboard-page for background designs.
  + .content-wrapper to wrap the main dashboard content.
  + Bootstrap container and grid classes for responsive layout.
* **Header**
  + Flexbox used to align logo, brand, search input, and unit selector.
  + Responsive adjustments to hide brand on smaller screens.
* **Stock Cards**
  + Rounded input field with shadow effect.
  + Button styling with hover effect.
* **Header** 
  + Flexbox for alignment of logo, brand, and search bar
* **Responsive Design**
  + Media queries for tablet (max-width: 991.98px) and mobile (max-width: 767.98px & 575.98px) to adjust padding, font sizes, and layout.

**JavaScript Structure (script.js)**

* **API Integration**
  + Fetch stock data (price, volume, P/E ratio, etc.)
  + Display search results dynamically
* **DOM Manipulation**
  + Elements cached in els object for efficient access.
  + Render stock listings and watchlist dynamicallyForecast cards (renderForecast)
  + Populate **modal details and charts** when a stock is clicked
* **Chart.js Visualization**
  + Line chart for **30-day price history**
* **Event Handling**
  + Search input: Enter key triggers stock data fetch.
  + Filter buttons: Show stocks by sector
  + Forecast button: Open detailed modal.
* **Error Handling**
  + Show message if **invalid stock symbol** entered.
  + Fallback UI for missing data.

**CSS Styling Strategy**

* Bootstrap grid for responsive layout
* Flexbox for content alignment and distribution
* Media queries for adaptive design on mobile, tablet, and desktop
* Clear typography and contrasting colors for readability
* Hover animations for stock cards to improve engagement

**Key Features**

* **Stock Listings:** Company names, tickers, and prices
* **Search & Filter:** Find stocks by ticker, sector, or performance
* **Detailed Stock Modal:** Price history chart + financial metrics
* **Responsive Design:** Works across all devices
* **User-Friendly UI:** Smooth navigation and modern design

**Challenges Faced & Solutions**

* **Challenge:** Handling asynchronous API calls.  
  **Solution:** Used fetch with error handling and JSON parsing
* **Challenge:** Displaying stock details dynamically

**Solution:** Manipulated the DOM using jQuery to render forecast cards programmatically.

* **Challenge:** Ensuring full responsiveness across multiple devices.  
  **Solution:** Utilized Bootstrap grid, Flexbox, and media queries for adaptive layouts.
* **Challenge:** Handling invalid or misspelled location input.  
  **Solution:** Added validation and user-friendly error messages when API returns errors.

**Outcome**

The project produced a **responsive stock market dashboard** that allows users to:

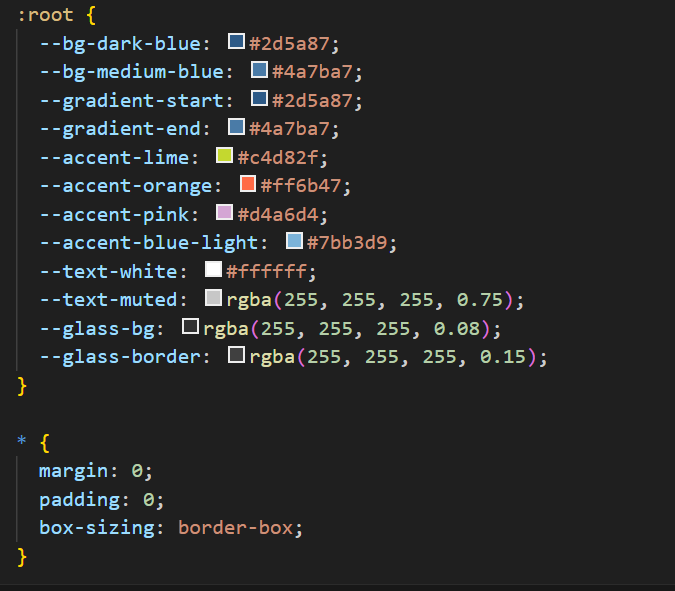
* Track **real-time stock performance**
* View **historical charts and metrics**
* Add stocks to a **watchlist**
* Enjoy a **smooth, modern UI** with animations and responsive design

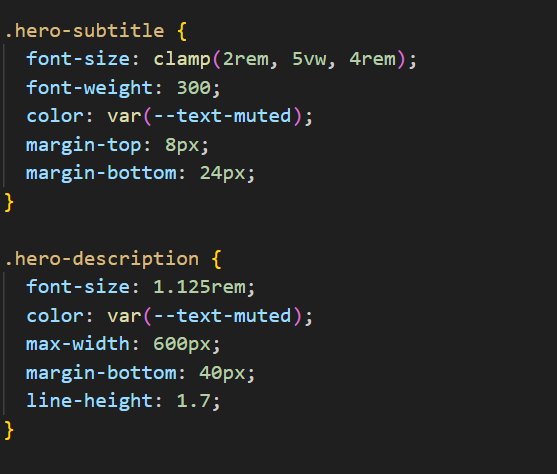
It successfully demonstrated **API integration, Bootstrap responsiveness, and financial data visualization with Chart.js**.

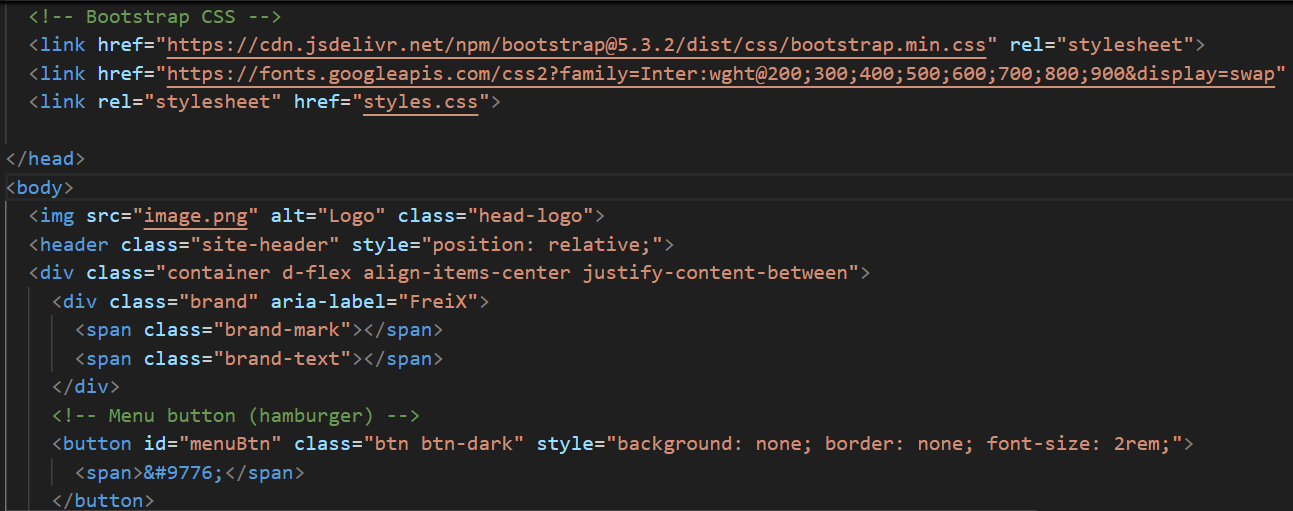
**Future Enhancements**

* **Real-time streaming with WebSockets** for live prices
* **Dark/Light mode toggle**
* **Advanced analytics**: technical indicators, alerts, portfolio tracking
* **Mobile app version** for iOS/Android
* **Performance optimization** with caching

**Sample Code**

****

****

****

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**OUTPUT SCREENSHOT**

A screen shot of a phone

AI-generated content may be incorrect.

