

In partial fulfillment for the award of the degree

**ARITIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**DBMS MINI PROJECT REPORT ON**

**STOCK MANAGEMENT**

**Submitted by**

**R SUPRAJA 231501164**

**BONAFIDE**

**CERTIFICATE**

Certified that this project report **“STOCK MANAGEMENT”** is the Bonafide work of **“R SUPRAJA (231501164)**

who carried out the project work under my

supervision.

Submitted for the Practical Examination held on

**-----------------------------------------**

**SIGNATURE SIGNATURE**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ABSTRACT**

**Problem Statement**

The Stock Market Management System (SMMS) is designed to streamline the operations of stock trading, investment management, and data tracking in a stock market environment. The system aims to provide a robust platform for investors, traders, and market analysts to manage their portfolios, track stock prices, analyze trends, and execute transactions.

**Introduction**

The project focuses on the system that allows users to register, authenticate, and create personalized portfolios that reflect real-time market conditions. Through its interface, users can monitor stock prices, view transaction histories, and analyze trends. Additionally, the system provides features for generating performance reports and managing user accounts, with an admin panel to oversee activities and ensure data integrity. By maintaining detailed records of all stock transactions and portfolio changes, the system offers a comprehensive solution for financial management in the stock market. With a focus on security, efficiency, and scalability, this project aims to provide a reliable tool for both individual investors and market administrators.

**Novelty**

The novelty of the Stock Market Management System (SMMS) lies in its comprehensive approach to managing both user portfolios and stock transactions within a single integrated platform. Unlike basic stock tracking systems, this project combines real-time market data with personalized portfolio management, enabling users to perform transactions, track performance, and receive historical analytics within the same environment.

**Key Features**

**Real-Time Stock Price Updates:** The system integrates live market data, allowing users to track stock prices in real time and make informed buy/sell decisions instantly, enhancing the user experience.

**Transaction History & Analytics:** The ability to generate detailed reports on past transactions, stock performance.

**Admin Panel for System Oversight:** This project includes an admin panel for managing users, overseeing transactions, and updating stock information, ensuring smooth system operations.

**Dynamic Portfolio Management:** The system supports dynamic portfolio management where users can buy, sell, and track their investments seamlessly.

**Security and Data Integrity:** The project emphasizes secure user authentication and data protection to safeguard sensitive financial information, which is crucial for any online stock trading platform.

**Impact**

The Stock Market Management System (SMMS) helps users manage investments more effectively by providing real-time stock tracking, secure transactions, and detailed portfolio insights. It empowers investors to make informed decisions, improves financial literacy, and enhances market transparency, making stock trading more accessible and efficient.

**Potential Future Applications**

Future applications include integrating AI for predictive analytics and personalized recommendations, developing mobile app versions for on-the-go access, supporting automated trading based on user-defined strategies, utilizing blockchain for enhanced transaction security, and expanding to global markets for broader investment opportunities. These advancements would improve user experience, decision-making, and security.

**Objective**

The primary objectives of this project are:

* **Real-Time Stock Tracking:** To provide users with real-time updates on stock prices and market trends for informed decision-making.
* **Portfolio Management:** To enable users to create, manage, and track their stock portfolios, including buying, selling, and monitoring their investments.
* **Secure Transactions:** To ensure secure and transparent transactions for buying and selling stocks while safeguarding user data and financial information.
* **Transaction History and Reporting:** To maintain a comprehensive record of user transactions and generate reports on portfolio performance, profits, and losses.
* **Admin Management:** To provide administrators with tools for managing user accounts, monitoring market activities, and updating stock prices and system data.
* **User-Friendly Interface:** To create an intuitive, easy-to-use platform for both novice and experienced traders to interact with the system efficiently.
* **Scalability and Performance:** To design the system to handle increasing amounts of data and users without compromising performance.

**Data Security and Privacy:** To implement robust security features, including secure logins, data encryption, and access controls to protect user informatio

**LANGUAGES USED**

**FRONT END / USER INTERFACE**

**HTML (HyperText Markup Language)**

HTML is the foundational language for creating web content structure. It organizes elements like text, images, and multimedia within a webpage, laying the groundwork for an interactive user experience. By defining sections, forms, and multimedia placement, HTML forms the basis upon which all web content is built.

Purpose in the Project: In our project, HTML is essential for structuring the interface components and interaction points:

• User Interaction Forms: HTML is used to create structured forms for data submission, ensuring that user inputs are collected accurately.

• Application Data Display: HTML enables the organized presentation of application data, facilitating intuitive navigation and information retrieval.

• Content Layout: HTML organizes content elements, providing a clear structure for user-friendly interaction.

By establishing a well-organized layout, HTML contributes to creating an intuitive and accessible user experience.

**CSS (Cascading Style Sheets)**

CSS is used to style HTML elements, giving them a polished, visually appealing presentation. It enables developers to define color schemes, fonts, layout spacing, and responsive designs, ensuring the interface is aesthetically pleasing and accessible on multiple devices.

Purpose in the Project: CSS is essential in our project to create a cohesive, professional design:

Visual Consistency: CSS maintains a consistent look across the application by styling components like buttons, forms, and tables.

Enhanced Usability: CSS defines a clear visual hierarchy, making navigation intuitive and aiding users in locating important sections quickly.

Responsive Design: CSS ensures adaptability across mobile, tablet, and desktop views, delivering a seamless experience regardless of the device.

4User Feedback: CSS animations and transitions provide interactive feedback, enriching the user experience with dynamic visual responses.

**JavaScript (Programming Language)**

JavaScript is a versatile language that enables interactive web functionalities. It powers user-driven behaviors, handles client-side data processing, and enables responsive interactions. With frameworks like Node.js, JavaScript also powers back-end functionality, enabling full-stack applications.

Purpose in the Project: JavaScript plays a crucial role in providing dynamic functionality and responsiveness:

• **Real-Time Interactivity:** JavaScript allows interactive behaviors such as updating data displays without reloading the page.

• **Form Validation:** JavaScript validates user input in real-time, enhancing the data accuracy before submission.

• **Client-Server Communication:** JavaScript is used to fetch data asynchronously, facilitating smooth interactions with the back-end database.

• **Session Management:** JavaScript manages application states, maintaining data as users navigate between various sections of the application.

**Database**

**MySQL (Relational Database Management System)**

MySQL is a popular open-source relational database management system (RDBMS) that organizes data into structured tables and supports efficient data management. Known for its reliability, MySQL provides the data integrity and query flexibility necessary for applications that handle user data.

**Purpose of MySQL in the Project**: In our project, MySQL is the core database system, enabling data storage and retrieval for user interactions:

• **Structured Data Management**: MySQL organizes application data into tables, storing it in a structured format for easy retrieval and manipulation.

• **Efficient Data Retrieval**: SQL queries enable rapid access to data, such as retrieving user profiles, session histories, and application logs.

• **Data Integrity and Security**: MySQL’s transaction capabilities ensure data consistency, especially useful in handling concurrent data access.

• **Scalability**: As a robust RDBMS, MySQL allows our application to scale by managing larger volumes of data and supporting additional users over time.

**SQLite** is a lightweight, serverless, self-contained relational database management system (RDBMS). It is an embedded database engine that provides a simple, fast, and efficient way to store and manage data within applications. Unlike traditional database management systems, SQLite does not require a separate server process or system to manage the database. It stores all data in a single file, making it portable and easy to integrate into software applications.

**Purpose of SQLite in the Stock Market Management System (SMMS)**

In the **Stock Market Management System (SMMS)**, SQLite serves as the database engine to manage and store all the system's data, such as:

* **User Information:** Storing user credentials, profiles, and authentication details.
* **Portfolio Data:** Keeping track of users' stock holdings, portfolio values, and transaction histories.
* **Stock Data:** Storing real-time and historical stock prices, stock symbols, and other related information.
* **Transaction Records:** Logging all buy/sell transactions, including the date, quantity, stock symbol, and price.

**Softwares and Modules**

**Node.js and Express.js Frameworks**

**Node.js** Node.js is a powerful, open-source, cross-platform runtime environment that enables developers to run JavaScript code outside the browser. It is built on Chrome's V8 JavaScript engine and is primarily used for building scalable, fast, and lightweight server-side applications. Node.js uses an event-driven, non-blocking I/O model, making it ideal for applications that handle multiple connections simultaneously, such as real-time chat apps, APIs, and data-intensive web applications.

**Key Features of Node.js:**

• **Asynchronous and Event-Driven**: Non-blocking execution enhances performance.

• **High Performance**: Powered by the V8 engine for fast code execution.

• **Single Programming Language**: Enables full-stack development using JavaScript.

• **Rich Ecosystem**: npm (Node Package Manager) provides thousands of libraries and modules.

**Express.js** Express.js is a minimal and flexible web application framework for Node.js. It simplifies the process of building web applications and APIs by providing a set of robust features and middleware for handling HTTP requests, routing, and more. Express serves as a foundation for building server-side applications and is highly extensible, allowing developers to integrate additional modules as needed.

**Key Features of Express.js:**

• **Routing**: Efficient and simple routing mechanisms to handle various HTTP requests.

• **Middleware**: Support for middleware functions to process requests and responses.

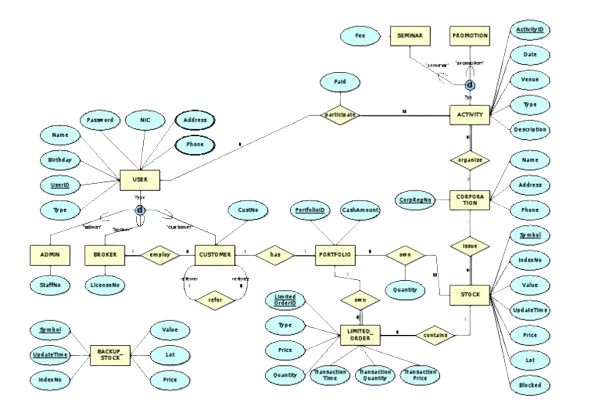
• **Template Engines**: Integration with template engines like EJS, Pug, or Handlebars for dynamic HTML generation.

• **Extensibility**: Easily integrates with databases, authentication libraries, and other tools.

• **REST API Development**: Ideal for building robust and scalable RESTful APIs.

**Node.js + Express.js Together** Express.js is commonly used with Node.js to streamline backend development. While Node.js handles the underlying runtime and server environment, Express provides a framework for structuring the application logic. Together, they form a powerful combination for creating modern, high-performance web and API services.

**ER DIAGRAM:**



**PROGRAM CODE**

**Front End**

**Home page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Stock Prediction Website - Home</title>

<style>

/\* Basic Reset \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Background and Font Styling \*/

body {

font-family: Arial, sans-serif;

color: #f5f5f5;

background-color: #111;

overflow-x: hidden;

}

/\* Sticky Header \*/

header {

background-color: rgba(0, 0, 0, 0.8);

padding: 15px;

text-align: center;

position: fixed;

width: 100%;

top: 0;

z-index: 1000;

}

nav a {

margin: 0 20px;

color: #fff;

font-size: 1.1em;

text-decoration: none;

transition: color 0.3s;

}

nav a:hover {

color: #ffd700;

}

/\* Hero Section \*/

.hero {

height: 100vh;

background-image: url('img.jpeg');

background-position: center;

background-size: cover;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

text-align: center;

color: #f5f5f5;

animation: fadeIn 2s ease-in-out;

}

.hero h1 {

font-size: 3em;

margin-bottom: 20px;

text-shadow: 2px 2px 5px #000;

}

.hero p {

font-size: 1.2em;

max-width: 800px;

line-height: 1.5;

}

.hero .cta-button {

margin-top: 30px;

padding: 15px 30px;

font-size: 1.2em;

color: #111;

background-color: #ffd700;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.cta-button:hover {

background-color: #ffae00;

}

/\* Sections Styling \*/

section {

padding: 60px 20px;

text-align: center;

}

.features, .benefits, .testimonials {

background-color: rgba(0, 0, 0, 0.8);

color: white;

padding: 50px;

border-radius: 10px;

margin: 40px auto;

width: 90%;

max-width: 800px;

box-shadow: 0 8px 16px rgba(0, 0, 0, 0.7);

}

h2 {

font-size: 2.2em;

margin-bottom: 15px;

}

p {

font-size: 1.1em;

margin-bottom: 15px;

}

/\* Testimonials \*/

.testimonials p {

font-style: italic;

opacity: 0.9;

}

/\* Scroll Animation \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(20px); }

to { opacity: 1; transform: translateY(0); }

}

</style>

</head>

<body>

<!-- Header -->

<header>

<nav>

<a href>Home</a>

<a href="perdiction.html">Prediction Details</a>

<a href="about.html">About</a>

<a href="contact.html">Contact</a>

</nav>

</header>

<!-- Hero Section -->

<section class="hero">

<h1>Welcome to the Future of Stock Prediction</h1>

<p>Experience the power of AI-driven stock forecasts with market-leading accuracy and timely insights. Navigate the markets with confidence and precision.</p>

<button class="cta-button" onclick="window.location.href='perdiction.html'">Start Predicting Now</button>

</section>

<!-- Features Section -->

<section class="features">

<h2>Features</h2>

<p>Our platform provides real-time predictions for the most popular stocks. With advanced analytics, we bring you the insights you need to make smarter investment decisions.</p>

<ul>

<li><strong>Real-Time Predictions:</strong> Updated every hour with the latest market data.</li>

<li><strong>Sentiment Analysis:</strong> Gain insights into market sentiment for your favorite stocks.</li>

<li><strong>Historical Data:</strong> Access past performance to evaluate trends and patterns.</li>

</ul>

</section>

<!-- Benefits Section -->

<section class="benefits">

<h2>Benefits</h2>

<p>Our predictions go beyond numbers, offering context and expert-driven insights. Here’s what you’ll gain:</p>

<ul>

<li><strong>Accurate Forecasts:</strong> Make decisions with the highest accuracy predictions available.</li>

<li><strong>Market Sentiment Analysis:</strong> Understand the impact of current events on stocks.</li>

<li><strong>Risk Management:</strong> Leverage data-driven tools to manage your investments wisely.</li>

</ul>

</section>

<!-- Testimonials Section -->

<section class="testimonials">

<h2>What Our Users Say</h2>

<p>"The predictions are incredibly accurate. I feel confident making trades based on their insights!" - <strong>Jordan T.</strong></p>

<p>"A must-have tool for any serious investor. The sentiment analysis alone is worth it." - <strong>Alex K.</strong></p>

<p>"I’ve saved so much time and money thanks to this website’s predictions." - <strong>Sarah L.</strong></p>

</section>

<!-- Footer -->

<footer>

<p>&copy; 2024 Stock Prediction Website</p>

</footer>

</body>

</html>

**Registration Page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>STOCK MARKET AND PREDICTION - Register</title>

<style>

/\* Set background image with a dull effect \*/

body {

background-image: url('1.jfif');

background-position: center;

background-attachment: fixed;

background-repeat: no-repeat;

background-size: cover;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

body::before {

content: "";

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: rgba(0, 0, 0, 0.5);

z-index: -1;

}

.register-container {

background-color: rgba(255, 255, 255, 0.8);

padding: 40px;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

text-align: center;

z-index: 1;

}

h2 {

margin-bottom: 20px;

color: #333;

}

.form-group {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

color: #444;

}

input[type="text"],

input[type="email"],

input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

margin-top: 5px;

}

button {

width: 100%;

padding: 10px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<div class="register-container">

<h2>REGISTER PAGE</h2>

<form id="registerForm">

<div class="form-group">

<label for="email">EMAIL:</label>

<input type="email" id="email" name="email" required>

</div>

<div class="form-group">

<label for="username">USER NAME:</label>

<input type="text" id="username" name="username" required>

</div>

<div class="form-group">

<label for="password">PASSWORD:</label>

<input type="password" id="password" name="password" required>

</div>

<button type="submit">REGISTER</button>

</form>

<p>Already have an account? <a href="signin.html">Login here</a></p>

</div>

<script>

// Handle registration form submission

document.getElementById('registerForm').addEventListener('submit', async (e) => {

e.preventDefault(); // Prevent page reload on form submission

const email = document.getElementById('email').value;

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

// Create the data to send

const registerData = {

email: email,

username: username,

password: password

};

try {

const response = await fetch('/registration', { // Ensure this matches the backend route

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(registerData)

});

const data = await response.json();

if (response.ok) {

// Handle success (e.g., redirect to homepage or login page)

alert('Registration successful!');

window.location.href = '/signin'; // Redirect to login page after successful registration

} else {

// Handle error from the server (e.g., duplicate email, etc.)

alert('Registration failed: ' + data.error);

}

} catch (error) {

console.error('Error:', error);

alert('An error occurred while registering.');

}

});

</script>

</body>

</html>

**SignIn Page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>STOCK MARKET AND PREDICTION - Login</title>

<style>

/\* Set background image with a dull effect \*/

body {

background-image: url('2.jfif');

background-position: center;

background-attachment: fixed;

background-repeat: no-repeat;

background-size: cover;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

body::before {

content: "";

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background-color: rgba(0, 0, 0, 0.5);

z-index: -1;

}

.login-container {

background-color: rgba(255, 255, 255, 0.8);

padding: 40px;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

text-align: center;

z-index: 1;

}

h2 {

margin-bottom: 20px;

color: #333;

}

.form-group {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

color: #444;

}

input[type="text"],

input[type="password"] {

width: 100%;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

margin-top: 5px;

}

button {

width: 100%;

padding: 10px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<div class="login-container">

<h2>LOGIN PAGE</h2>

<form id="loginForm">

<div class="form-group">

<label for="username">USER NAME:</label>

<input type="text" id="username" name="username" required>

</div>

<div class="form-group">

<label for="password">PASSWORD:</label>

<input type="password" id="password" name="password" required>

</div>

<button type="submit">LOGIN</button>

</form>

<p>Don't have an account? <a href="registration.html">Register here</a></p>

</div>

<script>

// Handle login form submission

document.getElementById('loginForm').addEventListener('submit', async (e) => {

e.preventDefault(); // Prevent page reload on form submission

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

// Create the data to send

const loginData = {

username: username,

password: password

};

try {

const response = await fetch('/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(loginData)

});

const data = await response.json();

if (response.ok) {

// Handle success (e.g., redirect to home page or show success message)

alert('Login successful!');

window.location.href = '/home'; // Redirect to home page

} else {

// Handle error

alert('Login failed: ' + data.message);

}

} catch (error) {

console.error('Error:', error);

alert('An error occurred while logging in.');

}

});

</script>

</body>

</html>

**Prediction Page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Prediction Results - Stock Prediction Website</title>

<style>

body {

background-color: #111;

background-position: center;

background-attachment: fixed;

background-repeat: no-repeat;

background-size: cover;

color: white;

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

}

header {

background-color: rgba(0, 0, 0, 0.7);

padding: 10px;

text-align: center;

position: sticky;

top: 0;

z-index: 1000;

}

nav a {

margin: 0 15px;

color: white;

text-decoration: none;

}

section {

background: rgba(0, 0, 0, 0.8);

padding: 20px;

border-radius: 10px;

margin: 20px auto;

width: 80%;

max-width: 600px;

}

h2 {

margin: 0 0 15px;

}

footer {

margin-top: 20px;

text-align: center;

}

.button {

display: inline-block;

padding: 10px 20px;

color: #111;

background-color: #ffd700;

border: none;

border-radius: 5px;

text-decoration: none;

margin: 10px 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.button:hover {

background-color: #ffae00;

}

select {

padding: 8px;

border-radius: 5px;

background-color: #333;

color: #fff;

border: none;

margin-bottom: 15px;

width: 100%;

}

</style>

</head>

<body>

<header>

<nav>

<a href="homepage.html">Home</a>

<a href="prediction.html">Prediction Details</a>

<a href="About.html">Results</a>

<a href="contact.html">Contact</a>

</nav>

</header>

<main>

<section id="result-section">

<h2>Prediction Results</h2>

<!-- Dropdown for selecting stock symbol -->

<label for="stock-symbol-dropdown">Select Stock Symbol:</label>

<select id="stock-symbol-dropdown">

<option value="AAPL">Apple (AAPL)</option>

<option value="GOOGL">Google (GOOGL)</option>

<option value="MSFT">Microsoft (MSFT)</option>

<option value="AMZN">Amazon (AMZN)</option>

<option value="NSGN">NeuroSign (NSGN)</option> <!-- Added NeuroSign -->

</select>

<!-- Dropdown for selecting time frame -->

<label for="time-frame-dropdown">Select Time Frame:</label>

<select id="time-frame-dropdown">

<option value="Next Day">Next Day</option>

<option value="Next Week">Next Week</option>

<option value="Next Month">Next Month</option>

</select>

<!-- Display the prediction output -->

<div id="result-output">

<p><strong>Stock Symbol:</strong> <span id="stock-symbol">AAPL</span></p>

<p><strong>Predicted Change:</strong> <span id="predicted-change">+5.2%</span></p>

<p><strong>Time Frame:</strong> <span id="time-frame">Next Week</span></p>

<p><strong>Insights:</strong> <span id="insights">Positive sentiment in market.</span></p>

</div>

<!-- Buttons for additional details -->

<div id="details-buttons">

<button class="button" onclick="showSentimentAnalysis()">Sentiment Analysis</button>

<button class="button" onclick="showHistoricalData()">Historical Data</button>

<button class="button" onclick="showRiskAnalysis()">Risk Analysis</button>

</div>

</section>

**About Page**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>About Us - Stock Prediction Website</title>

<style>

body {

background-color: #1e1e1e; /\* Dark background \*/

color: white;

font-family: 'Arial', sans-serif;

margin: 0;

padding: 20px;

}

header {

background-color: rgba(0, 0, 0, 0.8);

padding: 15px;

text-align: center;

position: sticky;

top: 0;

z-index: 1000;

border-bottom: 2px solid #444;

}

nav a {

margin: 0 20px;

color: #ffd700;

text-decoration: none;

font-weight: bold;

text-transform: uppercase;

}

nav a:hover {

color: #ffae00;

}

section {

background: rgba(0, 0, 0, 0.85);

padding: 30px;

border-radius: 10px;

margin: 30px auto;

width: 80%;

max-width: 800px;

border: 1px solid #333;

}

h2 {

margin: 0 0 20px;

font-size: 24px;

font-weight: bold;

text-transform: uppercase;

}

p {

font-size: 16px;

line-height: 1.6;

margin: 10px 0;

}

footer {

margin-top: 40px;

text-align: center;

color: #aaa;

}

footer p {

font-size: 14px;

}

.team-member {

margin: 15px 0;

padding: 10px;

background: rgba(0, 0, 0, 0.7);

border-radius: 8px;

}

.team-member h3 {

margin: 0;

font-size: 18px;

color: #ffd700;

}

.team-member p {

margin-top: 5px;

color: #ccc;

}

</style>

</head>

<body>

<header>

<nav>

<a href="homepage.html">Home</a>

<a href="prediction.html">Prediction Details</a>

<a href="about.html">About</a>

<a href="contact.html">Contact</a>

</nav>

</header>

<main>

<section>

<h2>About Us</h2>

<p>Welcome to the Stock Prediction Website! Our mission is to provide accurate and actionable stock predictions based on the latest data and advanced algorithms. Whether you're a seasoned investor or a beginner, our platform helps you make informed decisions by offering stock price predictions, sentiment analysis, historical performance, and risk assessments.</p>

<h2>Our Features</h2>

<p>Our platform offers the following key features:</p>

<ul>

<li><strong>Stock Predictions:</strong> Get accurate predictions on stock price changes for various companies like Apple, Google, Microsoft, Amazon, and even NeuroSign (NSGN).</li>

<li><strong>Sentiment Analysis:</strong> Understand market sentiment to gauge the investor outlook for your chosen stock.</li>

<li><strong>Historical Data:</strong> Review past performance of stocks to identify trends and make data-driven decisions.</li>

<li><strong>Risk Analysis:</strong> Evaluate the risk factor of each stock, helping you make safer investment choices.</li>

</ul>

<h2>Our Team</h2>

<div class="team-member">

<h3>SUPRAJA R</h3>

<p>Founder & CEO</p>

<p>John is a seasoned stock market expert with over 10 years of experience. He is passionate about using AI to predict stock trends and empower investors.</p>

</div>

<div class="team-member">

<h3>SRI BALAJI P</h3>

<p>CTO</p>

<p>Jane brings a strong background in machine learning and data analysis to the team. She is responsible for developing and fine-tuning the prediction algorithms used on our platform.</p>

</div>

<div class="team-member">

<h3>SUBASHINI R</h3>

<p>Software Engineer</p>

<p>Sam is the technical backbone of our platform, ensuring smooth user experience, fast data processing, and seamless integration of new features.</p>

</div>

<h2>Our Technology</h2>

<p>We use a combination of cutting-edge technologies to power our platform, including:</p>

<ul>

<li><strong>Machine Learning:</strong> We leverage advanced machine learning algorithms to predict stock price movements and analyze market sentiment.</li>

<li><strong>Big Data:</strong> Our platform analyzes vast amounts of historical data to provide accurate and reliable predictions.</li>

<li><strong>Cloud Computing:</strong> Our platform runs on cloud infrastructure to ensure high availability and fast performance.</li>

</ul>

<p>Our team is constantly working to improve the platform, incorporate new stocks, and enhance our prediction models. We strive to provide the most reliable and up-to-date information for our users.</p>

</section>

</main>

<footer>

<p>&copy; 2024 Stock Prediction Website. All Rights Reserved.</p>

</footer>

</body>

</html>

<!-- Additional Information Sections -->

<section id="sentiment-analysis" style="display: none;">

<h2>Sentiment Analysis</h2>

<p>The market sentiment for <span id="selected-stock-symbol">AAPL</span> shows a <strong>positive outlook</strong> due to recent market trends and news sentiment.</p>

</section>

<section id="historical-data" style="display: none;">

<h2>Historical Data</h2>

<p>Past performance of <span id="selected-stock-symbol">AAPL</span> indicates a consistent growth trend with a high of +10% last month.</p>

</section>

<section id="risk-analysis" style="display: none;">

<h2>Risk Analysis</h2>

<p>The current risk factor for <span id="selected-stock-symbol">AAPL</span> is <strong>moderate</strong> based on market volatility and recent price fluctuations.</p>

</section>

</main>

<footer>

<p>&copy; 2024 Stock Prediction Website</p>

</footer>

<script>

// Sample data structure to hold insights for each stock, timeframe, and predicted change

const predictedChangeData = {

"AAPL": {

"Next Day": "+0.5%",

"Next Week": "+5.2%",

"Next Month": "+10%"

},

"GOOGL": {

"Next Day": "-0.3%",

"Next Week": "+3.0%",

"Next Month": "+7.5%"

},

"MSFT": {

"Next Day": "+0.2%",

"Next Week": "+4.5%",

"Next Month": "+8.0%"

},

"AMZN": {

"Next Day": "-1.0%",

"Next Week": "+2.0%",

"Next Month": "+6.0%"

},

"NSGN": { // Added NeuroSign data

"Next Day": "+1.2%",

"Next Week": "+6.5%",

"Next Month": "+12%"

}

};

// Sentiment, historical data, and risk analysis for each company

const sentimentData = {

"AAPL": "Positive sentiment in market due to strong earnings and product innovation.",

"GOOGL": "Mixed sentiment as investors await updates on AI advancements.",

"MSFT": "Neutral sentiment with steady demand for cloud services.",

"AMZN": "Positive sentiment driven by strong Q4 sales expectations.",

"NSGN": "Bullish sentiment due to groundbreaking advancements in AI for healthcare."

};

const historicalData = {

"AAPL": "Apple has seen a consistent increase of 5% per month for the last 6 months.",

"GOOGL": "Google’s stock saw a 15% increase over the last quarter following AI developments.",

"MSFT": "Microsoft reported a 10% growth in cloud services in the last year.",

"AMZN": "Amazon's stock grew by 8% during the last holiday season.",

"NSGN": "NeuroSign saw a 20% growth in stock value last month following AI breakthrough in healthcare."

};

const riskAnalysis = {

"AAPL": "Low risk: Apple has strong market fundamentals and brand loyalty.",

"GOOGL": "Moderate risk: Regulatory challenges could affect growth.",

"MSFT": "Low risk: Microsoft’s cloud business continues to grow steadily.",

"AMZN": "High risk: Seasonal volatility can lead to fluctuating earnings.",

"NSGN": "Moderate risk: While AI in healthcare is promising, competition and regulation may pose challenges."

};

// Function to update the displayed prediction details and insights based on selections

function updatePredictionDetails() {

const stockSymbol = document.getElementById('stock-symbol-dropdown').value;

const timeFrame = document.getElementById('time-frame-dropdown').value;

// Update the displayed stock symbol and time frame

document.getElementById("stock-symbol").textContent = stockSymbol;

document.getElementById("time-frame").textContent = timeFrame;

document.getElementById("selected-stock-symbol").textContent = stockSymbol;

// Get the predicted change for the selected stock and time frame

const predictedChange = predictedChangeData[stockSymbol][timeFrame];

document.getElementById("predicted-change").textContent = predictedChange;

// Update insights based on the selected stock symbol and time frame

const insight = insightsData[stockSymbol][timeFrame];

document.getElementById("insights").textContent = insight;

}

// Update prediction details when dropdowns change

document.getElementById('stock-symbol-dropdown').addEventListener('change', updatePredictionDetails);

document.getElementById('time-frame-dropdown').addEventListener('change', updatePredictionDetails);

// Functions to show and hide additional sections

function showSentimentAnalysis() {

hideAllDetails();

const stockSymbol = document.getElementById('stock-symbol-dropdown').value;

const sentiment = sentimentData[stockSymbol];

document.getElementById('sentiment-analysis').style.display = 'block';

document.getElementById('sentiment-analysis').innerHTML = `

<h2>Sentiment Analysis</h2>

<p>The market sentiment for <span id="selected-stock-symbol">${stockSymbol}</span> shows a <strong>${sentiment}</strong></p>

`;

}

function showHistoricalData() {

hideAllDetails();

const stockSymbol = document.getElementById('stock-symbol-dropdown').value;

const history = historicalData[stockSymbol];

document.getElementById('historical-data').style.display = 'block';

document.getElementById('historical-data').innerHTML = `

<h2>Historical Data</h2>

<p>Past performance of <span id="selected-stock-symbol">${stockSymbol}</span> indicates: <strong>${history}</strong></p>

`;

}

function showRiskAnalysis() {

hideAllDetails();

const stockSymbol = document.getElementById('stock-symbol-dropdown').value;

const risk = riskAnalysis[stockSymbol];

document.getElementById('risk-analysis').style.display = 'block';

document.getElementById('risk-analysis').innerHTML = `

<h2>Risk Analysis</h2>

<p>The current risk factor for <span id="selected-stock-symbol">${stockSymbol}</span> is: <strong>${risk}</strong></p>

`;

}

function hideAllDetails() {

document.getElementById('sentiment-analysis').style.display = 'none';

document.getElementById('historical-data').style.display = 'none';

document.getElementById('risk-analysis').style.display = 'none';

}

// Initialize prediction details on page load

updatePredictionDetails();

</script>

</body>

</html>

**Back End**

const express = require('express');

const sqlite3 = require('sqlite3');

const bcrypt = require('bcrypt');

const bodyParser = require('body-parser');

const path = require('path');

// Initialize Express app

const app = express();

const port = 170;

// Middleware to parse JSON bodies

app.use(bodyParser.json());

// Serve static files from the public directory

app.use(express.static(path.join(\_\_dirname, 'public'))); // Ensure 'public' is the correct folder for static files

// Initialize SQLite database (Ensure stockuser.db is in the same directory)

const db = new sqlite3.Database('stockuser.db', (err) => {

if (err) {

console.error('Error opening database:', err.message);

} else {

console.log('Connected to SQLite database');

}

});

// Create the stockuser table if it does not exist

db.run(`

CREATE TABLE IF NOT EXISTS stockuser (

id INTEGER PRIMARY KEY AUTOINCREMENT,

email TEXT NOT NULL UNIQUE,

username TEXT NOT NULL,

password TEXT NOT NULL

);

`);

// Route to serve registration page

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

res.sendFile(path.join(\_\_dirname, 'public', 'registration.html')); // Adjust path to where your registration.html is located

});

// Route to serve login page (sign-in)

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

res.sendFile(path.join(\_\_dirname, 'public', 'signin.html')); // Adjust path to where your signin.html is located

});

// Route for user registration (sign up)

app.post('/registration', async (req, res) => {

const { email, username, password } = req.body;

// Validate input fields

if (!email || !username || !password) {

return res.status(400).json({ error: 'All fields are required' });

}

try {

// Hash the password before storing it

const hashedPassword = await bcrypt.hash(password, 10);

// Insert the new user into the database

const query = 'INSERT INTO stockuser (email, username, password) VALUES (?, ?, ?)';

db.run(query, [email, username, hashedPassword], function (err) {

if (err) {

console.error('Error inserting user:', err.message);

return res.status(500).json({ error: 'Database error: ' + err.message });

}

// Send success response

res.status(201).json({

message: 'User registered successfully',

userId: this.lastID

});

});

} catch (error) {

console.error('Error during registration:', error.message);

res.status(500).json({ error: 'Server error' });

}

});

// Route for user sign-in (log in)

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

const { email, password } = req.body;

// Fetch the user by email

const query = 'SELECT \* FROM stockuser WHERE email = ?';

db.get(query, [email], async (err, user) => {

if (err) {

console.error('Error fetching user:', err.message);

return res.status(500).json({ error: 'Database error: ' + err.message });

}

// Check if user exists

if (!user) {

return res.status(401).json({ error: 'Invalid email or password' });

}

// Verify the password

const passwordMatch = await bcrypt.compare(password, user.password);

if (!passwordMatch) {

return res.status(401).json({ error: 'Invalid email or password' });

}

// Successful login

res.status(200).json({ message: 'Logged in successfully!' });

});

});

// Start the server

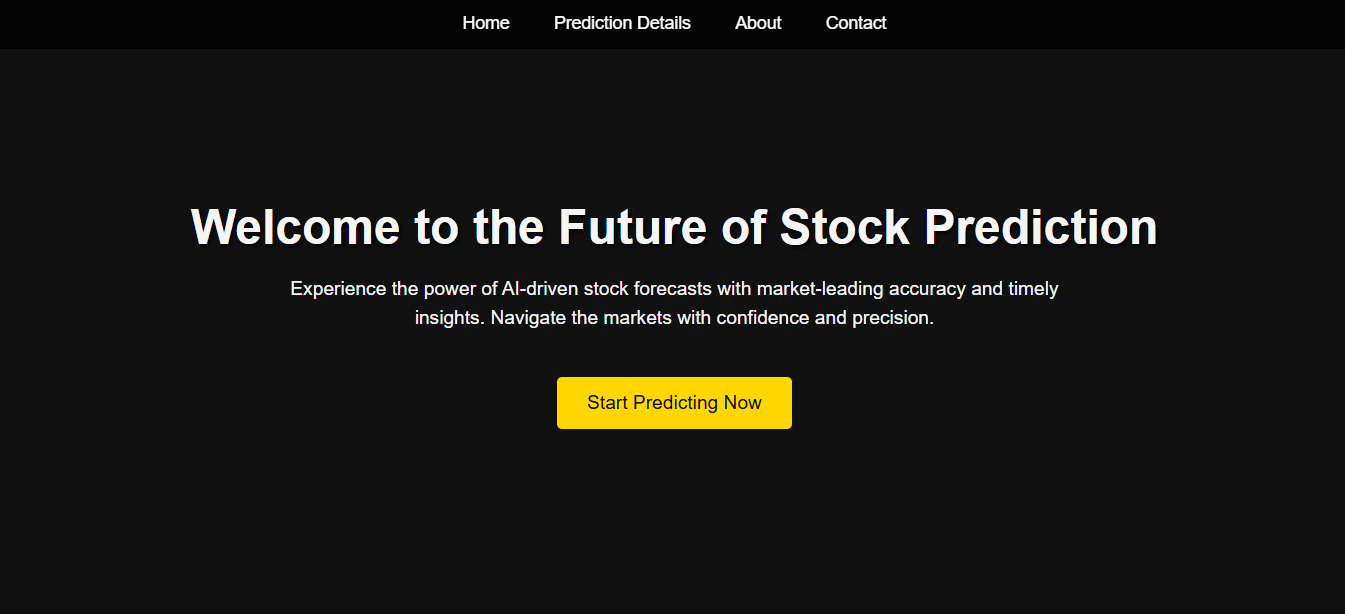
app.listen(port, () => {

console.log(`Server running at http://localhost:${port}`);

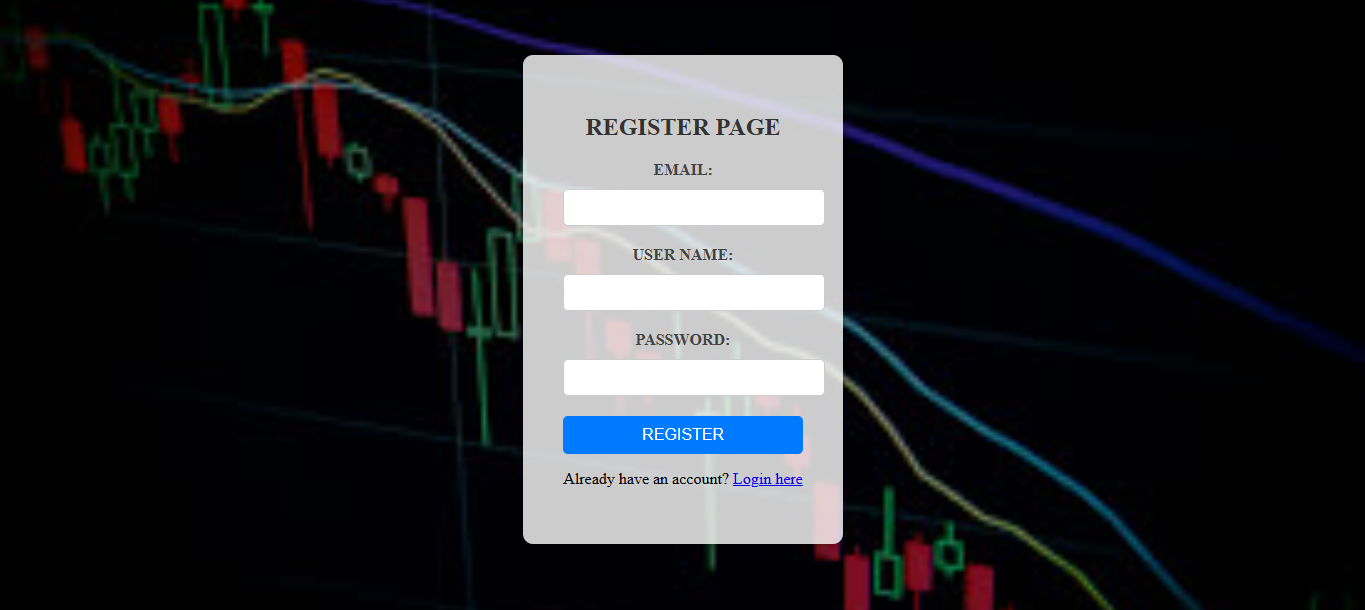
});

**OUTPUT**

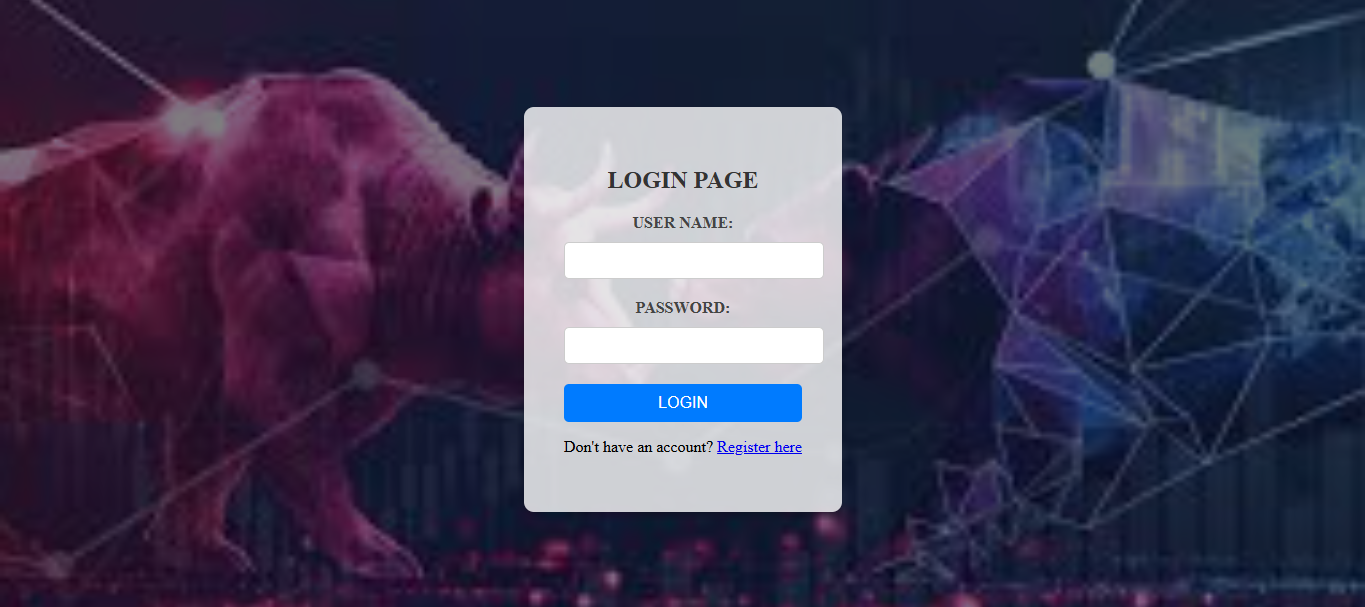
**Home Page**



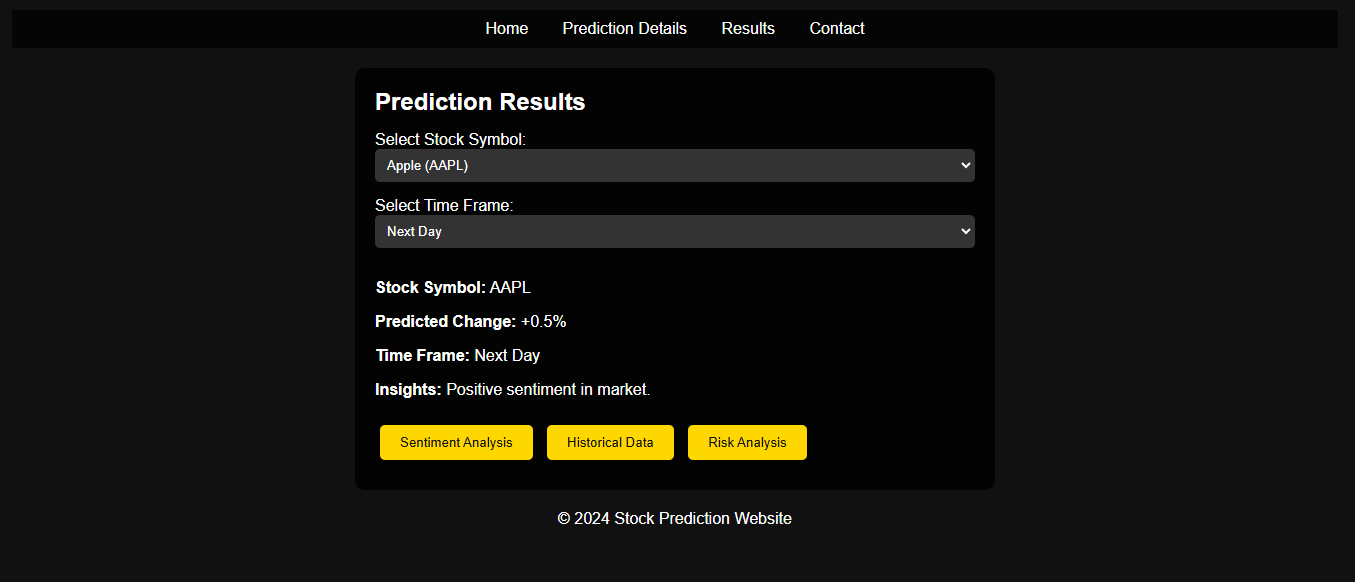
**Registration Page**



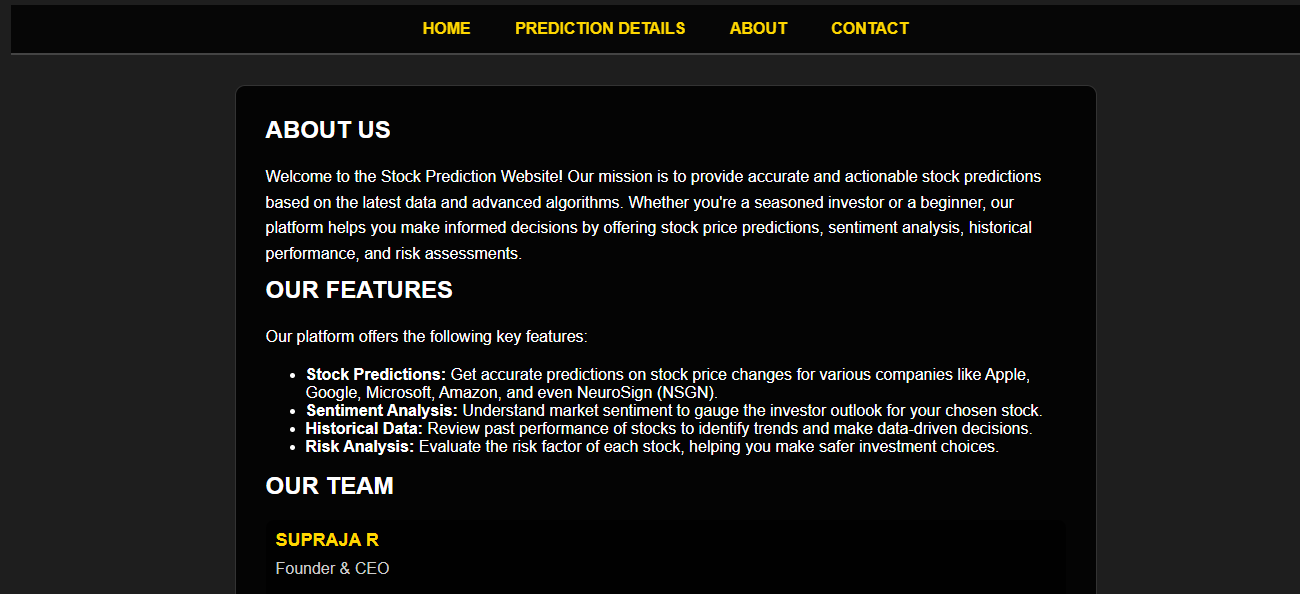
**SignIn Page**

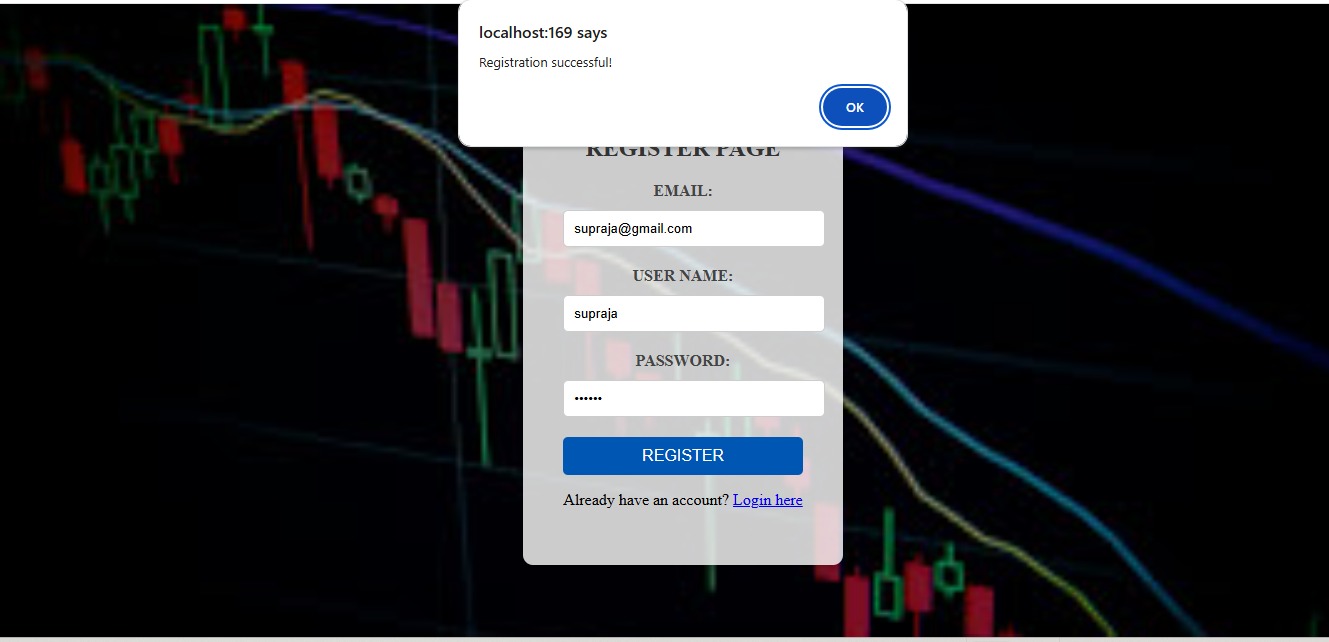


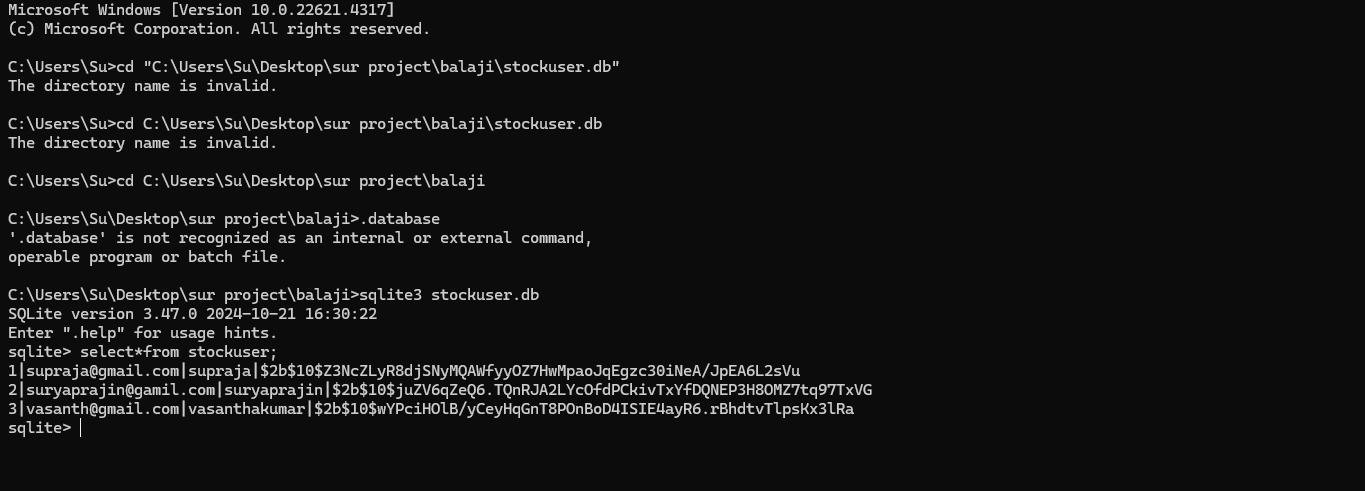
**Prediction Page**



**About Us page:**



****

****

****

**Challenges Observed:**

* Limited Scalability: While SQLite is suitable for small to medium-scale systems, it may not scale effectively for large numbers of users or real-time trading scenarios with high-frequency transactions.
* No External Data Integration: In a real-world scenario, stock data would need to be fetched from a live market feed, which is not part of this system's current design.

**Future Improvements:**

* Enhanced Scalability: Moving to a more robust database system like MySQL or PostgreSQL for handling larger data loads and more complex queries.
* Live Market Data Integration: Incorporating real-time stock price feeds from stock exchanges (like Nasdaq or NYSE) for dynamic market data updates.
* Mobile Application: Developing a mobile version to allow users to access and manage their portfolios on the go.
* Advanced Analytics: Implementing AI-driven predictive analytics to assist users in making smarter investment decisions.

**Bibliography:**

1. "Database Management Systems."  
   Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, McGraw-Hill, 2003.This book provides foundational knowledge of database design and management, including concepts relevant to relational databases likeSQLite.
2. "SQLiteDocumentation."  
   SQLite Consortium, https://www.sqlite.org/doclist.html.Official SQLite documentation detailing its usage, features, and technical specifications for implementation in various systems.
3. "Stock Market Management System Using Database."  
   P. P. Patil and N. A. Shinde, International Journal of Computer Science and Information Technologies, Volume 4, 2013.

This paper discusses the implementation of stock market systems using databases and offers a relevant reference for understanding data management in trading platforms.

1. "Developing Web Applications Using SQLite Database."  
   Bruce A. Powel, 1st Edition, Apress, 2019.

This book guides developers on integrating SQLite into web applications

similar to its role in the Stock Market Management System.