**Project Scope – Stock Prediction Using Machine Learning Web Application**

**Project Objectives**

* Develop a Web Application for Stock Price Prediction:
  + Create a user-friendly web application that allows users to input historical stock market data and receive predictions regarding whether a specific stock will go up or down on the next trading day.
* Train and Implement a Machine Learning Model:
  + Build and train a robust machine learning model that leverages historical stock market data for accurate predictions. This model should be capable of handling various stocks and historical data sets.
* Real-Time Data Integration (Optional):
  + Consider integrating real-time stock market data feeds (if available) to provide up-to-the-minute predictions and enhance user experience. This could be an advanced feature if time permits.
* User Authentication (Optional):
  + Implement user authentication and user account management features to enhance security and enable personalized experiences for registered users.
* Interactive User Interface:
  + Design an intuitive and visually appealing user interface that provides users with a seamless experience. Include features like data visualization, charting, and easy data input.
* Performance Optimization:
  + Ensure the web application's performance is optimized for efficient predictions and responsiveness. This may involve optimizing code, caching, and server scaling strategies.
* Security Measures:
  + Implement security measures to protect user data and ensure the integrity of your application. This includes secure data transmission, input validation, and user authorization.
* Documentation and User Instructions:
  + Create comprehensive documentation for your web application, including clear instructions for users on how to use the application effectively.
* Deployment on Hosting Platform:
  + Deploy the web application to a reliable hosting platform (e.g., Heroku, AWS, or Azure) to make it accessible to users on the internet.
* Monitoring and Maintenance Plan:
  + Develop a plan for monitoring the application's performance and resolving issues promptly. Consider setting up logging and alerting mechanisms.

**Stakeholder Identification:**

* Project Owner:
  + Description: The primary stakeholder is the project owner, Jonathon Mitchell, who is undertaking this personal project to develop and showcase skills on a CV.
  + Interest: The project owner's main interest is self-improvement, skill development, and enhancing their employability.
* Potential Employers or Hiring Managers (Indirect Stakeholders):
  + Description: Potential employers or hiring managers are indirect stakeholders who may review the project on the project owner's CV.
  + Interest: Their interest lies in evaluating the project owner's skills, knowledge, and practical experience demonstrated by the project.
* Fellow Developers or Peers:
  + Description: Fellow developers and peers in the tech community may interact with and provide feedback on the project.
  + Interest: They have an interest in knowledge sharing and learning from the project.
* End Users (if publicly deployed):
  + Description: If the web application is made publicly accessible, end users are stakeholders who may use the application for stock predictions.
  + Interest: Their interest centres around the usability and effectiveness of the application for stock market predictions.

**Project Deliverables:**

* Functioning Web Application:
  + A fully operational web application accessible via a web browser, where users can input historical stock data and receive predictions on whether a specific stock will go up or down on the next trading day.
* Trained Machine Learning Model:
  + A machine learning model trained on historical stock market data that can make accurate stock price predictions. This includes the model file and any necessary pre-processing or feature engineering scripts.
* User Interface (UI) Design:
  + The design components and assets for the user interface, including layout, graphics, and user interaction elements, to create an intuitive and visually appealing user experience.
* Documentation:
  + Comprehensive documentation, including user guides, installation instructions, and an explanation of how the machine learning model works. This documentation should be clear and user-friendly.
* Source Code:
  + The complete source code for both the frontend (React) and backend (Node.js with Express.js), including any relevant scripts, configurations, and dependencies.
* Deployment Package:
  + A package containing all necessary files and instructions for deploying the web application on a hosting platform, such as Heroku, AWS, or another provider.
* Security Measures Implementation:
  + Documentation outlining the security measures implemented in the application to protect user data and ensure data integrity.
* Performance Optimization Documentation:
  + Documentation highlighting the performance optimization strategies employed in the application, such as code optimizations and caching mechanisms.
* Version Control Repository:
  + A version control repository (e.g., on GitHub) where the project's source code and related files are hosted for version tracking, collaboration, and community engagement.
* Monitoring and Maintenance Plan:
  + A plan outlining how the application will be monitored, maintained, and updated to ensure its reliability and sustainability.

**Functional Requirements:**

* Users will be able to input a company’s name, and if it is in the data that the model has been trained on, the model will output whether the stock will go up or down the next day with a certain level of accuracy

**Non-Functional Requirements:**

* Performance Requirements:
  + Response Time: The web application should respond to user interactions within 2 seconds.
  + Scalability: The application should be designed to gracefully handle increased user traffic without experiencing significant performance degradation.
  + Load Testing: Load testing should be conducted to validate the application's responsiveness under various user loads.
* Reliability Requirements:
  + Availability: The application should maintain a high level of availability, with a target uptime of 99% to ensure consistent user access.
  + Fault Tolerance: The application should implement measures to gracefully handle unexpected failures, minimising downtime and service disruption.
* Usability Requirements:
  + User Interface Consistency: The application must maintain a consistent and user-friendly interface throughout all its pages and features.
  + Accessibility: The application should be accessible to users with disabilities, complying with web accessibility standards such as WCAG.
* Compatibility Requirements:
  + Cross-Browser Compatibility: The application must function correctly on popular web browsers like Chrome, Firefox, and Safari.
  + Mobile Responsiveness: The application should be responsive and user-friendly on various devices, including smartphones and tablets.
* Documentation Requirements:
  + User Documentation: The application should provide user-friendly documentation to assist users in effectively utilising its features.
  + Code Documentation: Maintain clear and comprehensive code documentation to facilitate future development and maintenance efforts.