**TEAM MEMBERS NAME, SRN:**

|  |  |
| --- | --- |
| **PES2UG24AM185** | **Yajat Soni** |
| **PES2UG24CS586** | **Vikhyat Agarwal** |
| **PES2UG24CS598** | **Vismaya Harish** |
| **PES2UG24CS603** | **Vyshali Aytha** |

**TITLE: Stock Market Analytics with C Programming**

**(Stock Correlation Analysis**

**Compare stock trends and show correlation coefficients to identify relationships between stocks. )**

**TEAM NO: 2**

**Software Requirement Specification (SRS) Document**

**Stock Market Analytics with C Programming**

**1. Brief Description of Project**

Stock Market Analytics with C Programming is a simple, command-line interface stock market analytics system written in C. It allows users to manage a portfolio of stocks by entering details such as stock name, current price, and recent historical percentage changes. The program provides investment advice (buy, sell, or hold) based on recent performance, compares trends between different stocks using correlation analysis, and calculates key metrics like average daily return and volatility.

**2. Purpose / Goal**

**Purpose**: To help users make more informed investment decisions by providing automated analysis and recommendations for their stock portfolio. By evaluating recent price movements and historical trends, the system aims to guide users on whether to buy, sell, or hold a particular stock, and to highlight relationships between different stocks in their portfolio.

**Goal**: To offer a practical tool for beginners or anyone interested in basic stock analysis, using straightforward logic and statistical methods that are easy to understand and implement

**3. Usefulness / Benefit**

By automating analytical tasks-such as calculating average returns, measuring volatility, and detecting correlations-it saves users time and reduces the risk of emotional or uninformed decision-making. The clear recommendations and statistics provided by the system can help users build a more balanced and effective portfolio, understand the risks and potential of their investments, and ultimately make better financial choices with confidence.

**4. Hardware / Software Involved**

**Hardware Requirements:**

* Standard PC/Laptop
* Minimum 2GB RAM
* 500MB disk space

**Software Requirements:**

* Programming Language: C
* Compiler: GCC / Turbo C / Code: Blocks
* Operating System: Windows/Linux/MacOS

**5. Detailed Feature List**

**Features:**

**1. Investment Advice**

 Analyses a stock’s recent and historical performance to recommend whether to buy, sell, or hold. Implements a simple momentum-based investment strategy by evaluating the latest percentage change and the average of recent historical changes. It provides a recommendation and reasoning.

**2. Trend Comparison**

Compares two stocks from the portfolio by name. Calculates and displays the Pearson correlation coefficient between their recent historical percentage changes. This helps the user understand if the stocks tend to move together, inversely, or independently, supporting diversification and risk assessment.

**3. Correlation Analysis**

Supports the comparison feature by quantifying how similarly or differently two stocks move, aiding in portfolio analysis and diversification decisions.

**4. Estimation of Return Volatility**

Displays key risk and return metrics for a selected stock. Calculates and presents the average daily return (mean of the last 5 percentage changes) and volatility (standard deviation). This gives insight into the stock’s typical performance and risk level.

**5. Determining the Long-Term scope of the stock**

Calculates the behaviour of the stock while looking at its long-term future and how well it performs in future markets.

**6. Test / Demonstration Plan**

* **Unit Testing:** Test individual features such as investment suggestion, trend comparison, correlation coefficient, state of volatility and future scope.
* **Integration Testing:** Ensure seamless interaction between the user and the algorithm.
* **System Testing:** Validate the system’s overall functionality.
* **User Acceptance Testing:** Test with sample users to check usability and performance.

**7. Expected Interaction Interface and Sample Use Cases**

**Interaction Interface:**

* Console-based UI with menu-driven navigation
* Simple text-based input for user interaction
* Structured outputs with clear descriptions and reasoning

**Sample Use Cases:**

1. **User starts program**
   * User logs in → Selects option from menu → enter choice → wait for system response
2. **Program processes request**
   * Program takes in request → Takes details of a sample stock → Registers stock details
3. **Program displays menu again**
   * Menu displayed again → User enters choice of calculation → Program receives choice
4. **Program displays answer**
   * Program reads details of stock → Performs necessary calculation→ Presents requested answer with details and reasoning

**Individual member contribution:**

**Student 1 Name: Yajat**

**What have you done for the project?**

I built the feature responsible to perform ‘correlation analysis’ on the given stock. This feature:

* Implements the Pearson correlation coefficient by using the formula
* Identifies strong positive/negative correlations
* Helps understand stock relationships

**Student 2 Name: Vyshali**

**What have you done for the project?**

**I implemented the compare\_stocks() function, which compares two stocks based on their recent returns. It registers important details of the second stock that is to be compared to the first stock like its name and past returns. I initialised and assigned values to the variables of the correlation formula.**

**Student 3 Name: Vikhyat**

**What have you done for the project?**

**In this project, I created 2 simple C function to help analyze stocks. The main focus was on checking how much the stock price changes (called volatility) and whether the stock is good for long-term investment. I used past price data to calculate the stock's volatility and gave it a risk level (low, medium, or high). I also used financial data like revenue, profit, and debt to check the company's overall performance. Based on this, the program gives basic advice on whether the stock is suitable to hold for the long term.**

**Student 4 Name: Vismaya**

**What have you done for the project?**

**I implemented the feature that analyzes percentage price changes over several days to generate trading recommendations. Specifically, I wrote the calc\_avg() and make\_rec() functions. I developed a function to calculate the average of historical percentage changes(calc\_avg) and a function that evaluates the latest price change against the historical average and provides a recommendation to BUY, SELL, or HOLD(make\_rec) . I also handled user input and ensured data validation. My contribution focused on the analytical core of the project, allowing users to make basic investment decisions based on recent market behavior.**