QUANTFOLIO DOCUMENTATION

GitHub
Figma Mockup
UML Diagram

Problem Statement:

Managing investments across multiple platforms is a challenge that every investor faces. Whether you're trading stocks, holding cryptocurrencies, or investing in mutual funds, keeping track of profits, losses, and overall portfolio performance can feel overwhelming. Investors often find themselves switching between different trading platforms, manually calculating returns, and ensuring their gains in one asset class offset losses in another.

Quantfolio is designed to simplify portfolio management by consolidating investment tracking into a single, easy-to-use tool. Instead of dealing with complex spreadsheets and fragmented tracking, Quantfolio helps investors by:

- Recording transactions and organizing holdings in one place.
- Calculating Profit & Loss (PnL) and percentage returns to track performance.
- Fetching stock price updates to keep the portfolio up to date.
- Setting target prices and sending alerts when stocks reach predefined sell points.

For investors, timing is everything—but constantly checking market fluctuations manually is neither practical nor efficient. With Quantfolio, you no longer must watch stock charts all day. The system helps you stay informed without unnecessary effort, ensuring you never miss the right moment to sell.

Invest smarter, track efficiently, and make better financial decisions—all in one place with Quantfolio.

Business Requirements:

- ◊ Users should be able to log in their transactions using key details like ticker symbol, quantity, purchase/selling price and target selling price.
- ◊ System should allow users to choose whether they want to invest at the market value or record a buy/sell price of their own.
- ◊ Trader should be able to record the platform where each transaction is being made. This will allow user to assess overall performance across multiple brokers.
- ◊ System should provide a comprehensive summary of the user's portfolio, displaying all the stocks they own with key details such as number of shares held, purchase price, target selling price, and current market price.
- ◊ User's portfolio should also calculate and display the total invested amount, the current portfolio value and the overall percentage return.
- ◊ Users should be able to update their stock transactions manually to reflect market changes.
- ◊ The main feature of the user's portfolio should be the Profit/Loss calculated and its percentage return.
- ◊ User should be able to set a target selling price for each stock, and the system should automatically alert them whenever the stock reaches or exceeds this target price. It can be a simple console message or notification, ensuring the user does not miss the right moment to sell their stocks based on their predefined investment strategy.

Nouns and Verbs:

Nouns:

1. User
2. System
3. Trader
4. Transaction
5. Details
6. Ticker Symbol
7. Quantity
8. Purchase Price
9. Selling Price
10. Target Selling Price
11. Market Value
12. Buy/Sell Price
13. Platform
14. Performance
15. Brokers
16. Summary
17. Portfolio
18. Stock
19. Number of Shares
20. Current Market Price
21. Total Invested Amount
22. Current Portfolio Value
23. Overall Percentage Return
24. Stock Transactions
25. Market Changes
26. Profit/Loss (PnL)
27. Percentage Return
28. Target Price
29. Console Message
30. Notification
31. Investment Strategy

Verbs:

1. Log
2. Allow
3. Choose
4. Invest
5. Record
6. Made
7. Assess
8. Provide
9. Display

10. Own
11. Calculate
12. Update
13. Reflect
14. Set
15. Alert
16. Reaches
17. Exceeds
18. Ensure
19. Miss
20. Sell

Target Audience:

1. Retail Investors and everyday traders:
    a. Individuals who actively trade stocks, ETF, or cryptocurrencies across multiple platforms.
    b. People who manually track their Profit/Loss (PnL) and wish to automate portfolio monitoring.
    c. Investors who set target prices for selling but often miss opportunities due to market fluctuations.

2. Beginner and Intermediate Investors:
    a. New investors who want a simple, easy to use tool to track their stock holdings.
    b. Individuals learning about portfolio performance and how to set investment strategies.
    c. Users who find spreadsheets overwhelming and prefer an automated portfolio tracker.

3. Finance and Business Students:
    a. Students who need a hands-on way to learn about returns, PnL, and market behaviour.
    b. Students practicing hypothetical trading or analyzing past stock performance.

4. Busy Professionals and Long term investors:
    a. People investing for the long term who want a quick portfolio overview without logging into multiple platforms.
    b. Investors with diversified portfolios who want to check if gains in one asset class offset losses in another.
    c. Working professionals who do not have time to manually calculate every PnL and prefer automated tracking.

5. Multi-Platform Traders:
    a. Users who invest across multiple platforms.
    b. People who switch platforms often and need a centralized view of their total investments.

Rules:

1. Mandatory Transaction details:
    a. Users must provide the following details while logging a transaction:
        i. Ticker Symbol (ie. AAPL, TSLA)
        ii. Quantity (number of shares or units)
        iii. Purchase price (either market price or custom price)
        iv. Target selling price (optional but recommended)
    b. Transactions without the mandatory details should not be recorded.
2. Platform Specification:
    a. Users must specify the platform (eg. Zerodha, Robinhood, Binance) when recording a transaction.
3. Price updates and Market Data Handling:
    a. User must manually update stock prices if not using an API.
    b. If an API is used, prices should be fetched periodically or upon user request.
    c. If the stock does not have an updated price, the system should use the last available price.
4. PnL and Returns Calculations:
    a. Profit and Loss must be calculated using (Current Price – PurchasePrice) * quantity
    b. Percentage return should be calculated as: PnL*100/(PurchasePrice * Quantity). If current price is missing, last available price should be used.
5. Target Price Alerts:
    a. User must set a target price if they want sell alerts.
    b. If a stock's current price reaches or exceeds the target price, the system must notify the user.
    c. Notifications should be simple (console message or log entry).
6. Portfolio Summary Accuracy:
    a. The system must calculate and display:
        i. Total invested amount
        ii. Total portfolio value
        iii. Overall Portfolio return
    b. If no transactions exist, the summary should not display incorrect values (eg. Division by zero errors).

Challenge Questions:

1. What happens if a user tries to log a transaction without entering a required detail (e.g., missing ticker symbol, quantity, or price)?

2. How should the system handle fractional shares or cryptocurrency investments, where quantity can be decimal (e.g., 0.5 shares or 0.002 BTC)?

3. What should happen if a user enters a negative value for quantity, purchase price, or target price? Should we allow short-selling transactions?

4. How frequently should stock prices be updated, and should users be able to specify their own refresh intervals?

5. If using an API for stock price updates, how should the system handle API rate limits and failures? Should it fall back to the last known price?

6. What should happen if a user enters the same stock multiple times? Should we merge transactions or treat them separately?

7. If a user records transactions from multiple platforms, how should platform-based filtering work when viewing the portfolio summary?

8. How should target price alerts be handled? Should the system only alert the first time it's hit, or should it continuously alert until the stock is sold?

9. Should the system allow users to update past transactions (e.g., change purchase price or quantity) or should all transactions be immutable once recorded?

10. What should happen if a user tries to delete a stock from their portfolio? Should they be able to remove individual transactions, or just hide them from view?

Summary of Classes, Attributes and Associations:

Classes and their Attributes:

1. **User:**
    a. Attributes: name, email, portfolio
    b. Methods: logTransaction(), viewPortfolio(), updateStockPrice()

2. **Portfolio:**

   a. Attributes: stocks (list of Stock Objects), totalInvestedAmount, currentPortfolioValue, overallPercentageReturn
   b. Methods: calculateTotalInvested(), calculatePortfolioValue(), calculateOverallReturn()

3. **Stocks:**
   a. Attributes: tickersymbol, quantity, purchaseprice, sellingprice, targetprice, currentprice, platform, profitloss, percentagereturn
   b. Methods: updateprice(), calculatePnL(), checktargetprice()

4. **Transaction:**
   a. Attributes: stock, quantity, buySellPrice, date, platform
   b. Methods: recordTransaction(), modifyTransaction(), deleteTransaction()

5. **MarketData (Optional for API integration)**
   a. Attributes: tickerSymbol, currentPrice, lastUpdated
   b. Methods: fetchPrice(), updatePrice()

6. **AlertSystem**
   a. Attributes: stock, targetPrice, alertTriggered
   b. Methods: checkAlert(), sendNotification()

Associations:

1. **User to Portfolio (One to One):** Each User has a Portfolio object containing their Stock holdings.
2. **Portfolio to Stock (One to Many):** A Portfolio consists of multiple Stock objects.
3. **Stock to MarketData(One to One):** Each Stock has a corresponding MarketData object for price updates.
4. **Stock to Alert System(One to One):** Each Stock has an AlertSystem to notify users when the target price is reached.

Ranking Dimensions for User Personas:

1. **Trading Frequency**
   Measures how often the user actively buys and sells stocks or assets.
2. **Investment Experience**
   Represents the user's familiarity with investment concepts, market behavior, and trading strategies.
3. **Diversification**
   Measures the breadth of assets and platforms a user engages with, including multiple brokers and asset classes.
4. **Time Availability**
   Represents how much time a user must monitor and manage their investments.

5. **Tech-Savviness**
   Assesses how comfortable the user is with digital tools, investment apps, and automated tracking.

User Personas and their User Stories:

1. John – The Active Retail Investor

Background: John is a 43-year-old DBMS Professor who actively invests in stocks and cryptocurrencies. He manages his portfolio across multiple platforms like Robinhood, Binance and Zerodha. John frequently buys and sells stocks based on market trends and target prices, but he struggles with tracking all his investments in one place.

User Stories:

a. As an investor, I want to record my stock transactions, so that I can track my investments over time.
b. As an investor, I want to enter my purchase price or the market price, so that I can log accurate transaction details.
c. As an investor, I want to set a target selling price for each stock, so that I can get alerts when it's time to sell.
d. As an investor, I want to see a summary of my total portfolio value and percentage return, so that I can quickly evaluate my investment performance.
e. As an investor, I want to track which platform I used for each transaction, so that I can assess my performance across different brokers.
f. As an investor, I want to update stock prices manually, so that I can adjust my portfolio in case real-time data isn't available.

Ranking Dimensions:

- Trading Frequency: 5 – Trades frequently across multiple platforms.
- Investment Experience: 4 – Experienced investor but not a professional trader.
- Diversification: 5 – Uses multiple platforms and asset classes (stocks and crypto).
- Time Availability: 2 – Busy professor, has limited time for manual tracking.
- Tech-Savviness: 4 – Comfortable with trading apps but prefers a streamlined tracking tool.

2. Abhishek – The finance Student and Beginner Investor

Background: Abhishek is a 24-year-old finance student who is learning about investing strategies and market behavior. He has recently started investing in stocks through Fidelity and E-trade and wants to analyze his portfolio performance over time. Abhishek prefers to manually input stock prices to simulate different market conditions for learning purposes.

User Stories:

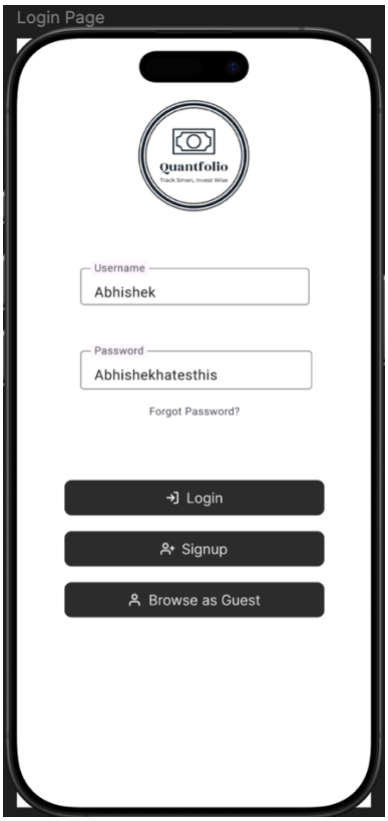
a. As a finance student, I want to add stocks to my portfolio with different purchase prices, so that I can simulate investment scenarios.
b. As a finance student, I want to manually update stock prices, so that I can analyze different market conditions.
c. As a finance student, I want to see how my portfolio value changes over time, so that I can understand stock performance.
d. As a finance student, I want to set a target selling price for my stocks, so that I can test different investment strategies.
e. As a finance student, I want to receive alerts when a stock reaches my target price, so that I can learn about timing trades effectively.
f. As a finance student, I want to view my Profit/Loss and percentage return for each stock, so that I can better understand investment performance.

Ranking Dimensions:

- Trading Frequency: 2 – Not actively trading, mostly experimenting.
- Investment Experience: 2 – Beginner investor still learning strategies.
- Diversification: 3 – Uses multiple platforms but with a limited portfolio.
- Time Availability: 4 – Has time to manually update and experiment.
- Tech-Savviness: 5 – Highly comfortable with digital tools and manual tracking.

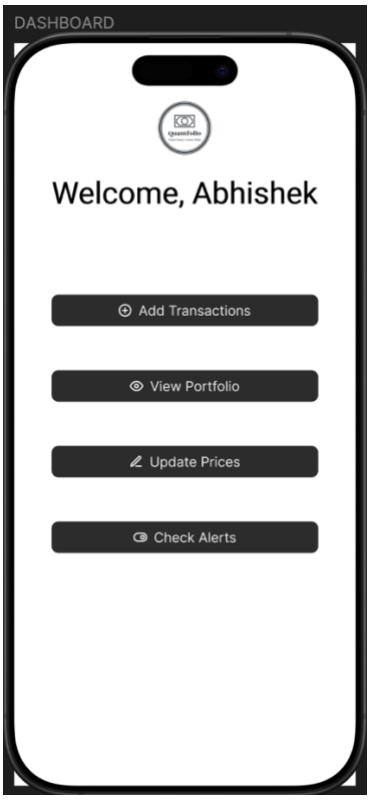
Interface Mockups:

1. Login:



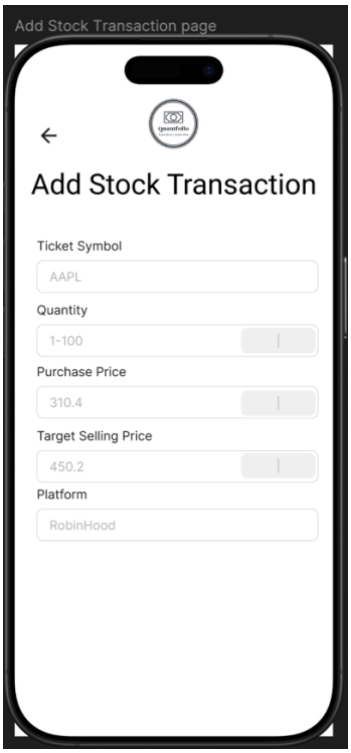
The page includes a username and password input field, a forgot password link, and three main actions: login for existing users, signup for new users, and browse as a guest. The Quantfolio logo is placed at the top for branding.

2. Home Page (Dashboard / Main Menu)



a. User Stories Covered:
    i. As an investor, I want to see a summary of my total portfolio value and percentage return, so that I can quickly evaluate my investment performance.
    ii. As a finance student, I want to view my Profit/Loss and percentage return for each stock, so that I can better understand investment performance.
b. Page Features:
    i. Welcome message
    ii. Options to Add Transaction, View Portfolio, Update Prices, Check Alerts
    iii. Display Total Portfolio Value, PnL, % Return

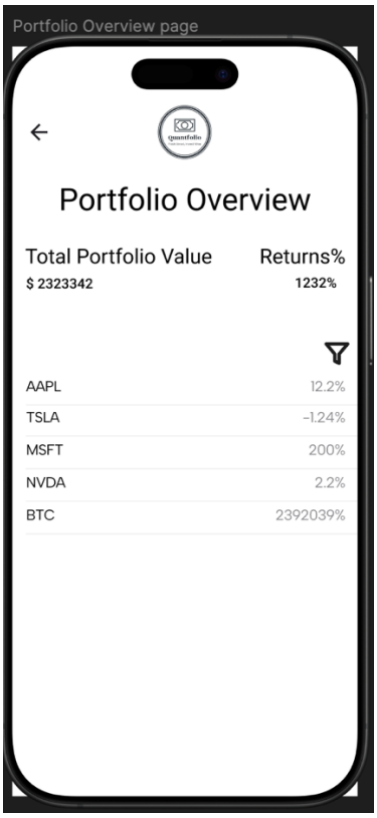
3. Add Stock Transaction Page:



a. User Stories Covered:
    i. As an investor, I want to record my stock transactions, so that I can track my investments over time.
    ii. As a finance student, I want to add stocks to my portfolio with different purchase prices, so that I can simulate investment scenarios.
    iii. As an investor, I want to enter my purchase price or the market price, so that I can log accurate transaction details.

b. Page Features:
    i. Fields: Ticker Symbol, Quantity, Purchase Price, Target Selling Price, Platform
    ii. Option to record at market price or manually enter a price
    iii. Submit button to save transaction

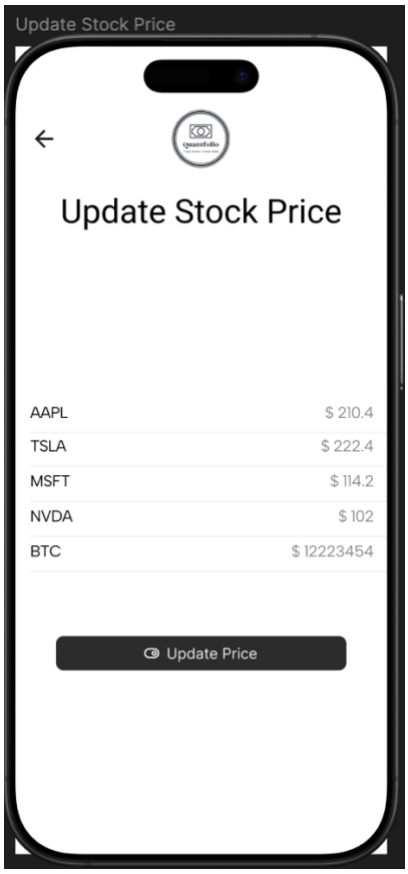
4. Portfolio Overview Page:



a. User Stories Covered:
    i. As an investor, I want to track which platform I used for each transaction, so that I can assess my performance across different brokers.
    ii. As a finance student, I want to see how my portfolio value changes over time, so that I can understand stock performance.

b. Page Features:
    i. Table displaying: Stock Symbol, Platform, Quantity, Purchase Price, Current Price, PnL, % Return
    ii. Sorting/filtering options (e.g., by platform or % return)

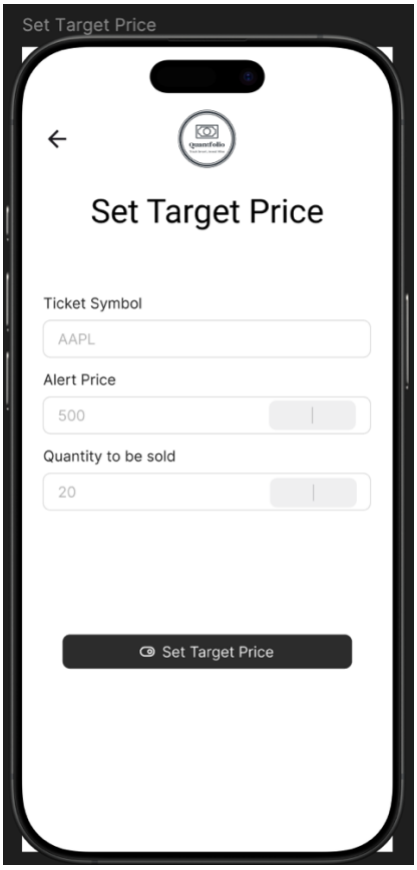
5. Update Stock Price Page:



a. User Stories Covered:
    i. As an investor, I want to update stock prices manually, so that I can adjust my portfolio in case real-time data isn't available.
    ii. As a finance student, I want to manually update stock prices, so that I can analyze different market conditions.

b. Page Features:
    i. List of all stocks in portfolio with editable current price field
    ii. "Update Price" button to save changes

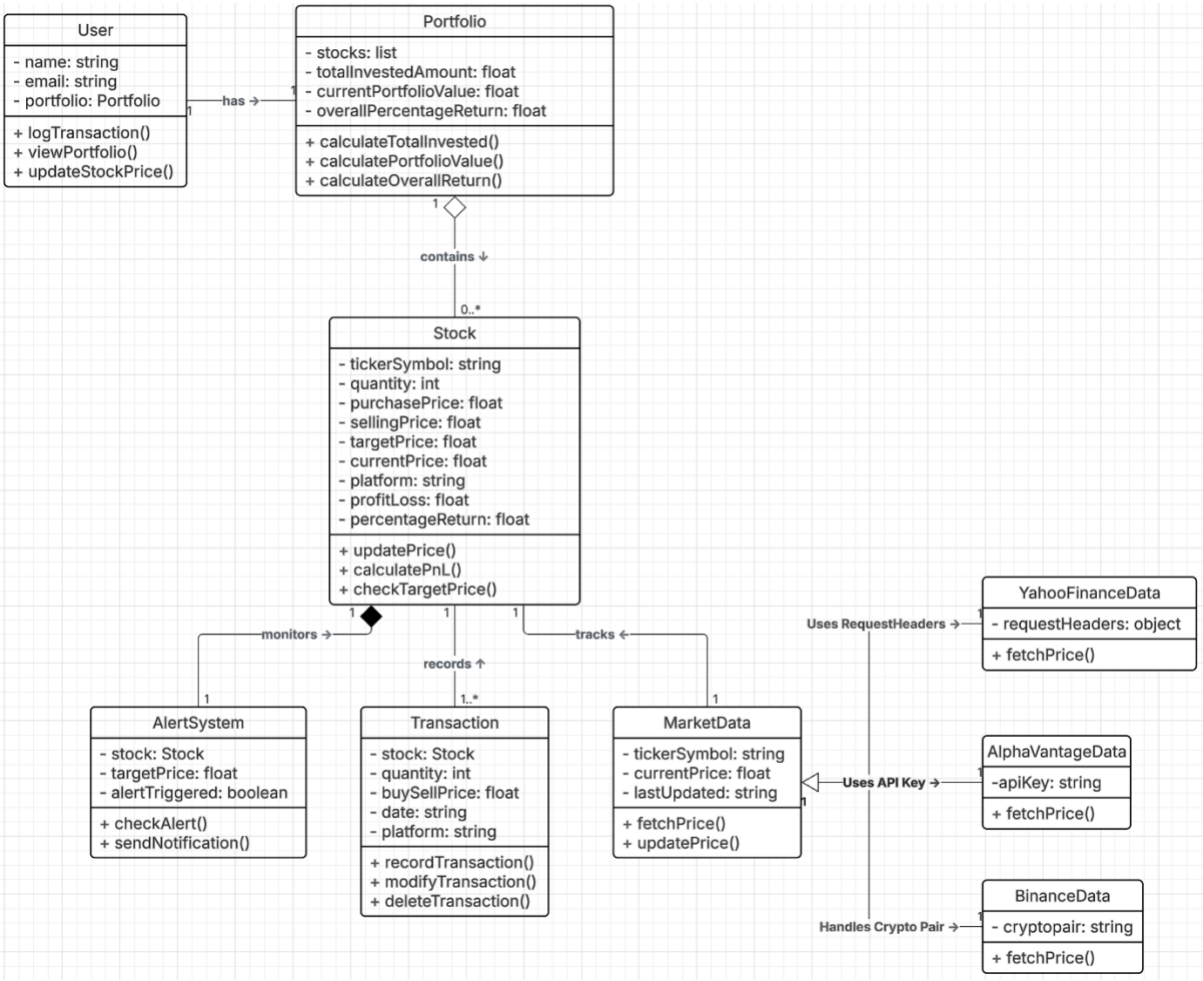
6. Set & Check Target Price Alerts Page



a. User Stories Covered:
    i. As an investor, I want to set a target selling price for each stock, so that I can get alerts when it's time to sell.
    ii. As a finance student, I want to set a target selling price for my stocks, so that I can test different investment strategies.
    iii. As an investor, I want to receive alerts when a stock reaches my target price, so that I can learn about timing trades effectively.

b. Page Features:
    i. Input field to set or edit target price for each stock
    ii. Button to enable alerts
    iii. Alert status display: Triggered / Not Reached

UML Class Diagram:



Resources Used:

1. GenAI Acknowledgment

A) **Model Used**: ChatGPT (GPT-4 Turbo, February 2025 version)

How GenAI Was Used: GenAI assistance was only used for structuring and formatting the README.md file in Markdown.

Prompts Used:

- "Format this README file in Markdown based on my provided project details."
- "Ensure proper sectioning and bullet points without adding new content."
- "Keep it minimal, without extra formatting, and ensure code blocks are correctly displayed."

Attachments Provided: Quantfolio Documentation.pdf AND UML_CLASS_DIAGRAM.PNG

Verification Process:
1. Manual Review: Each section was reviewed against the project specifications to ensure correctness.
2. Cross-checking with Existing Documents: The content was compared with the original system documentation to verify that no additional AI-generated details were introduced.
3. Testing Markdown Output: The final README file was previewed in GitHub to confirm proper rendering.

B) **Model Used:** ChatGPT (GPT-4 Turbo, February 2025 version)

How GenAI Was Used: GenAI assistance was used to understand JavaScript Documentation (JSDoc) standards and generate JSDoc comments for the provided class implementations.

Prompts Used:

- "Explain how to document JavaScript classes and methods using JSDoc."
- "Generate JSDoc comments for the following JavaScript code while ensuring clarity and best practices."
- "Ensure all function parameters, return values, and class properties are well-documented."

Attachments Provided:

- Quantfolio_Classes.js (Complete JavaScript implementation of classes and methods)
- Quantfolio_Documentation.pdf (Reference documentation)

Verification Process:
1. Manual Review: The generated JSDoc comments were reviewed against official JSDoc guidelines and best practices.
2. Code Integration Testing: The documentation was tested using npx jsdoc -d docs src/ to verify that all classes and methods were properly recognized.
3. Cross-checking with Existing Documentation: The generated JSDoc was compared with the original system documentation to ensure correctness.
4. Peer Review: A manual check was conducted to confirm that the AI-generated comments accurately described the function of each class and method.

C) **Model Used:** ChatGPT (GPT-4 Turbo, February 2025 version)

How GenAI Was Used: GenAI assistance was used to structure and format the AI Usage Acknowledgement sections in a clear and professional manner.

Prompts Used:

- “Help me write an AI usage acknowledgment for using ChatGPT to generate JavaScript documentation (JSDoc).”
- “Create an AI usage acknowledgment for ChatGPT’s assistance in structuring the README file.”
- “Ensure it follows a formal format while maintaining transparency about how AI was used.”

Verification Process:
1. Manual Review: The acknowledgments were reviewed to ensure they accurately describe how AI was used in the project.
2. Cross-checking with Project Work: Each acknowledgment was verified against actual AI-assisted work to ensure transparency.
3. Formatting Check: The final text was proofread for clarity and consistency within the README file.

2. Resources Used in Project:

Stock Market Data APIs: Documentation from Alpha Vantage, Yahoo Finance, and Binance for understanding stock data retrieval.