Team 2 GitHub Documentation

1. Abstract
   - A brief summary of the implementation to the project and objectives.

     This feature implements a **Peer Comparison** functionality in a finance application, enabling users to compare multiple stock symbols based on financial metrics. The key objectives include allowing users to input stock symbols, view a side-by-side comparison of selected metrics in a table and chart, and export the results as CSV or PDF files. This addition enhances user experience by providing intuitive visualizations and export options for deeper analysis.

2. Features Implemented
   - A list of features developed, including descriptions and significance.

     Stock Symbol Search: Users can input stock symbols (e.g., AAPL, TSLA) to retrieve their financial data.
     Comparison Table: A tabular display of financial metrics like P/E ratio, market capitalization, and more for the selected stock symbols.
     Dynamic Charting: Users can select metrics to visualize as bar charts for a better understanding of financial trends. Currently this functionality is not working. We will be creating git issues for functionality.
     Export Options:
     CSV Export: Downloads the comparison table as a CSV file.
     PDF Export: Captures the visual chart or comparison table as a PDF file.

3. Code Highlights
   - Key pieces of code or logic that is worth mentioning

     The implementation is straightforward and focuses on clarity and functionality. The code is well-structured and easy to read, making it self-explanatory. Key features like stock symbol search, metrics comparison table, and export options are seamlessly integrated. So not adding any code pieces here.

4. Challenges and issues
   - Possible challenges and issues arised or may arise in the future.

     API Key Management: Handling sensitive API keys securely during development and deployment.
     Incomplete or Inaccurate Data: Stock data may be missing or outdated, leading to inaccuracies.
     UI Scalability: Handling a large number of stock symbols and rendering charts with many metrics could slow down the application.
     Export Limitations: Exporting large or complex charts as PDFs might fail due to canvas rendering issues.

5. Solutions and Fixes
   - Strategies to fix possible challenges

     API Key Security: Use environment variables to store API keys and avoid hardcoding them in the codebase.
     Fallback for Missing Data: Implement default values or warnings for missing data to maintain functionality.
     Pagination and Lazy Loading: Introduce pagination or lazy loading for stock symbols to enhance scalability and performance.
     Enhanced Export Handling: Use more robust PDF generation libraries like jspdf for better scalability and performance.

6. Future Work and suggestions
   - Suggestions for further development or improvement of the project or the feature

     Additional Metrics: Expand the range of metrics available for comparison, such as ROI, ROE, and sector analysis.
     Interactive Charts: Allow users to interact with chart points (e.g., click to see detailed stock information).
     Real-Time Updates: Implement WebSocket or polling mechanisms for real-time stock data updates.
     Thematic Customization: Provide options for light/dark themes and personalized table/chart designs.

7. Repository and Documentation Links
   - URLs to any relevant documentation.

     GitHub Repository:
     [https://github.com/jeffreywallphd/AutoProphet/tree/FALL2024-BA5200-Team2]
     Alpha Vantage API Documentation: [https://www.alphavantage.co/documentation/]
     HTML2Canvas Documentation: [https://html2canvas.hertzen.com/]