STOCKY Summary/Reference File

List of Deliverables

In our Github (link: https://github.com/alex-keninger/Stocky) (and our zipped file), you will find 5 files. These files include:

- Project.py
  - The main commit file with all the code for running Stocky
- Brainstorming_Notes.txt
  - This was a reference used throughout the semester to jot down some goals and ideas for Stocky along the way
- Requirements.txt
  - This requirements file is used to list and load all of the libraries needed to run Stocky in the project.py file
- Sandbox.py
  - This was an file used to experiment with Streamlit objects, python libraries, and showcase some of the available widgets for Streamlit for reference
- Summary.pdf
  - (This file)

Project.py (Stocky) Explanation

Stocky can be accessed by running the project.py file or through its hosted online address via Streamlit Cloud (link: https://share.streamlit.io/akeninger7/githubproject/main/project.py). Included in Stocky are:

- (Home) A title and brief introduction
- (Home) A text input box which stock tickers can be entered and then searched by clicking the checkbox below. If a valid ticker is entered (an error message with pop up otherwise), this will pull the stock data from Yahoo Finance using the Yahoo Finance API and load the data.
- (Home) A dropdown select box allows you to select a time period to look back for your data. Options include 1 week, 1 month, 3 months, 1 year, and all Time. The selected option will populate the raw data table below.
- (Home) A collapsible dataframe which displays detailed stock information (ex: open price, high price, low price, etc.). This is followed by a title which displays which stock is currently being analyzed.
- (Home) A graph labeled 'Time Series Data' visualizes stock price over time. The x-axis displays years (will start when stock first became public and end with current date. The y-axis denotes the price of the stock in US dollars. A legend on the right labels the plotted blue and red lines. This graph provides multiple Plotly functions as well such as downloading as a png, zooming, panning, and scaling.

- (Home) A slider below the graph allows the user to manipulate the time scale shown on the x-axis of the graph, providing a variable scope to analyze the time period.
- (Sidebar) A 'Navigation' multiselect that lets the user travel between the three pages (Home, Watchlist, Analysis)
- (Sidebar) A dropdown select box that shows the 'Current Watchlist'. If a stock is selected, by clicking the button below the Home Page will populate with the selected watchlist stock data.
- (Watchlist) A text input box where stock can better entered and added to the watchlist by checking the checkbox below.
- (Watchlist) A dropdown select box labeled 'Current Watchlist' which shows a list of the stocks the user added to their watchlist. Below are three more watchlist functions: the first button clears the watchlist, the next button downloads the watchlist as a .txt file to your local computer, and the third allows the user to upload a file (a previously downloaded watchlist) which will load into the app and populate the watchlist on screen.
- (Analysis) While this page is unfinished, text displays the algorithm code planned to be used in the future to build a recommendation algorithm for the app.

Final Status Summary

In summary, Stocky is roughly 90% finished. While we were unable to finish the recommendation and prediction functionality by the end of the semester, we were able to achieve most of our goals.

Completed goals include:

- Building a functional web app
- Allowing the user to input a stock ticker and pull data from Yahoo Finance
- Loading data from the Yahoo Finance API to display in a raw data and plot format
- Using Plotly to provide a versatile and visually appealing graphed chart of the stock data over time
- Building a slider to further manipulate the graph in terms of time period
- Building a selectbox that allows the user to manipulate the time range of data to view
- Creating a watchlist the user can add stocks to for reference
- Building the watchlist functionality to seamlessly showcase selected stocks and building it into a sidebar widget for convenience
- Allowing the user to upload/download their watchlist
- Hosting the web app as a free-to-use service on a widely accessible domain
- Making the app visually appealing and efficiently designed
- Designing an algorithm to recommend good stock investments

Incomplete goals include:

- Integrating our recommendation algorithm into the web app to generate recommendations for the app user
- Adding onto the graph with a forecasted/future prediction of stock direction (with Facebook Prophet)