**Goals for the Stock Finder Project**

Build internal/independent private network storage server or private cloud-stored website solution/application, providing investment tools for in-house decision making and offering investment solutions exclusively to the existing and potential investors (clients), based on predetermined algorithmic data calculations.

The website will provide the sentiment analysis for investors to discover, react and respond to market opinion, based Equity Rating and Action combined. The sentiment is monitored and analyzed by the algorithms.

Product solution designation is a Hedge-Fund/Private Equity Firm/Wealth Management Company.

**Project Performance Report:**

Constructed a web-ready HMTL four page dashboard called the Stock Finder – Financial Planning Assistance tool.

**Coding Programs applied:**

The logic is written in Word and Excel.

JavaScript applied for building data tables, filtering buttons, plotted time chart.

HTML implemented code to display web page design/GUI output.

Python libraries created the linear plot charts.

Visual Studio Code platform compiled and stored the code/scripts.

**Web-Ready Dashboard consists of four (4) HTML pages:**

**Page#1-Stock Finder:** displaying the data table retrieved from the engineered CSV file. Table columns: Symbol, Name, Sector, Price, Price/Earn, 52w High, 52w Low, 52w Avg. were pulled from the S&P500 datahub.io dataset found online. Additional table rows: Price/Annual Average Percentage, Position on Annual Average Scale, Rating and Action were all created using custom algorithms.

**Tested:**

Plot the price per earnings (P/E) vs Rating and Action. Further, P/E/ Rating vs. P/E/Action ratio.

The preliminary numerical data test found formed numeric relationships among three (3) pairs of Rating/Action groups.

Investment advice is derived from the relational combination of Rating and Action columns.

**Page#1 created columns explanation:**

Column = P/AA % (Percentage of Price per share / Annual Average Price)

Column = Position w/3-Modes of P/AA% (in ascending order, top most valued):

1. Above 52Week Low

2. Below 52Week High

3. Below 52Week Low

Column = Rating (a part of investment advice, in ascending order, top most valued)

1.Best

2.Good

3.Better

4.Caution

5.Bad

Column = Action (a part of investment advice, in ascending order, top most valued)

1.Buy

2.Hold

3.Sell

4.Look

**Page#2-Comparison:** displaying the linear plot cluster charts created in Python Jupiter Notebook platform applying a Scikit library and using a sklearn function. The savefig() function is used to save the created cluster plot as png files. Saved png files call is performed in the html code. First three cluster plots are displaying the assumed relationship between “well-known” financial metrics.

Last or the 4th Cluster Plot named Price/Earnings vs. Rating/Action is a custom design, “unknown” to the financial world, The cluster plot is separated into 4 cluster groups, drawn from 505 samples:

Represents patterned linear correlation between random groups in Price/Earnings/Rating (PE/R) vs. Price/Earnings/Action (PE/A) ratios.

Standard-English language/human comprehended decision-making data is displayed in a table on page#1.

**Page#3-Chart:** displaying the plotted time/price per share line graph/chart. Chart is plotted using a D3 JavaScript library with an on click event listener function. Plotly library is used to create and populate the line chart. Time series/Stock price data is retrieved from open-source Quandl website. Stock ticker name is entered into a Stock search box and Plot Stock button is clicked, data is retrieved from Quandl, populated and output is displayed on webpage.

**Page#4 –Testimonial:** displaying text data introducing the viewer to the Stock Finder Project.

**This financial model is intended for show purposes only, a final released version of financial platform will contain many new methods and proper formulations to aid an investor in making a positive monetary gain or money rewarding long-term investment.**