Design and Implementation of a Web-Based Stock Scoring Engine

Leul Ejigu

January 28, 2026

References

[1] Xiaodan Dong, Weidong Huang, and Jitong Wang. Financial big data visualization: A machine learning perspective. In *Proceedings of the 17th International Symposium on Visual Information Communication and Interaction (VINCI '24)*, New York, NY, USA, 2024. ACM. Accessed: 2026-01-27.

> Visualizing data for over 500 companies is a major challenge for my web app. Dong, Huang, and Wang propose a machine learning approach to solving this Big Data visualization problem. They break down the process into three stages: before, during, and after the model runs. This is useful for me because I need to show a simple Score out of 130, but also allow users to look into the details if they want . I will use their "post-model" visualization strategies to decide how to display the summary statistics and trends for the top 10 stocks in my ranked list. This ensures that the dashboard remains readable even when processing the full S&P 500 dataset.

[2] Michael Ettredge, Vernon J. Richardson, and Susan Scholz. A web site design model for financial information. *Communications of the ACM*, 44(11):51–55, 2001. Accessed: 2026-01-27.

> Ettredge, Richardson, and Scholz investigate how financial websites should be structured to help investors actually make decisions. They distinguish between hard data such as stock prices and soft information like investor relations updates. This distinction is exactly what I will be dealing with in my project. My 13-metric system combines hard numbers like Revenue Growth with subjective scores like Moat or Leadership, so their design model gives me a way to organize these differently on the screen. Although the paper is older, the core design principles for financial data haven't changed much. I plan to use their ideas to design my Ranked Table so users can see the important metrics without getting overwhelmed by the subjective ones.

[3] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010. Accessed: 2026-01-27.

This paper introduces "pandas," the fundamental Python library I will be needing to build my application. McKinney explains the design of the DataFrame, which is the exact data structure I will need to store and rank the 500 stocks. I cited this because my project is heavily dependent on this tool for calculating the 13 metrics efficiently.

[4] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems*, 27(2):12:1–12:19, 2009. Accessed: 2026-01-27.

Schumaker and Chen present a system for automating the analysis of financial news to predict stock prices. While my project will focus largely on numerical ratios, I also have to score qualitative metrics like Leadership and Culture. This paper is essential because it demonstrates how to treat unstructured text as data. The authors use a machine learning approach to tag and classify financial terms, which is the same logic I will need to implement if I want my web app to automatically score the News/Catalyst portion of my dashboard.

[5] S&P Dow Jones Indices. S&p u.s. indices methodology. Technical report, McGraw Hill Financial, February 2016. Accessed: 2026-01-27.

This official methodology document from S&P Dow Jones Indices outlines the eligibility criteria and maintenance rules for the S&P 500 index. It defines the universe of stocks my application will rank. Understanding the divisor-based weighting and float-adjustment ensures that my app's data filtering logic remains consistent with industry standards.