Vanguard ETF Digital Assistant

Introduction:

This website application is designed to be a one-stop shop for Vanguard existing customers and potential future customers to track and manage their ETF investment.

ETFs issued by different companies and institutions can be very hard to keep track of. Currently, most of the free applications such as yahoo finance or google finance have stocks, ETF and other mutual funds mixed together so that new opportunities in ETFs can be easily missed. This app aims to provide a platform for people who are interested in monitoring the ETFs along with their holdings issued by Vanguard in the most efficient way. In order to achieve this, this digital assistant offers the following features:

1. Track ETFs on customized watchlists.
2. Discover the top movers on a particular date.
3. Analyze the top 10 holdings of an ETF.
4. Get ETF recommendations based off performance, risk, and sector makeup of Watchlist
5. Get in-depth time-series performance metrics on ETFs and top holdings

Architecture:

List of technologies:

- Python Libraries: wrds, yfinance, and Pandas used heavily in data ingestion, preprocessing and cleaning stage
- MySQL: Relational Database Management System used to host cleaned data
- AWS RDS: cloud instance used to store MySQL database
- DataGrip: integrated development environment used to interact with MySQL database
- Node.js: JavaScript runtime environment used to execute event-driven queries and scripts
- React: JavaScript library used for developing interactive user interface that connects with backend functionality

Data:

CRSP Financial Data

Link: https://wrds-www.wharton.upenn.edu/pages/get-data/center-research-security-prices-crsp/

Description: The Center for Research in Security Prices (CRSP) dataset was found through the Wharton Research Data Services (WRDS) website. It contains a robust set of security price, return, and volume data for the NYSE, AMEX and NASDAQ stock markets. This includes annual, quarterly, monthly, and intraday trading data dating all the way back to 1926.

Size: This dataset is also extremely rich, well-exceeding the project requirement of over 10,000 rows. Additionally it encompasses over 60 attributes including returns, dividends, industry indicators, and many more. On top of the stock ticker, there is also a Permanent Security Number (permno) to provide a ubiquitous id for each stock holding even if stock tickers have changed over the years.

Summary Statistics: Contains 62 attributes, including dates of transactions, share codes for type of security, trading status, volume, return, bid/ask, and closing price. For context, 2023's data for all ETFs tracked by CRSP has over 2.3 million rows. Mean opening price is around $95 USD while mean number traded is around 7507. For a popular Vanguard ETF "VOX", mean shares outstanding in 2023 was 29481.128, and mean open price was $102.71.

How it Was Used: CRSP was used to get data through 2023 for daily ETF and stock performance (i.e. price, volume) for stocks identified as in the top 10 holdings of at least one of the Vanguard ETF.

ETF Global Constituents

Link: https://wrds-www.wharton.upenn.edu/pages/get-data/etf-global/constituents/

Description: ETF Global tracks over 1900 ETFs and includes robust data about constituents of index ETFs, and was also found through WRDS. It includes holdings data, weights, number of shares, asset class, and other relevant information that is necessary to fully understand the ETFs offered by vendors such as Vanguard.

Size: Like CRSP, the data set is very rich, with 251 million rows for 2023 ETF data. Since this is very large, may consider preprocessing the dataset to include only the relevant data for Vanguard ETFs. Like CRSP, there are identifiers that are able to account for changes in tickers or M&As.

Summary Statistics: Contains 17 attributes, including date, asset class, constituent tickers, country of exchange, currency traded, weight, and market value. Country of exchange could provide meaningful data that could pair well with the World Bank datasets, while the constituent data is necessary for CRSP data to be meaningful for this project. For the Vanguard ETF "VOX"'s constituents, the upper quartile for weight in December 2023 was .55%, while the mean shares held was 683427.3. The max weight of a holding in Dec 2023 was 21.8%.

How it Was Used: ETFG was used to get the top 10 holdings of each Vanguard ETF and their corresponding weights.

Yahoo Finance Market Data

Link: https://pypi.org/project/yfinance/

Description: This library provides a convenient way to access Yahoo Finance's market data and statistics, including historical stock prices, company financials, and market indicators. It allows for easy extraction of data for a wide range of stocks, currencies, commodities, and more, directly into Python for analysis and visualization. The tool is aimed at investors, financial analysts, and researchers looking to perform detailed market analysis or develop financial applications.

Size: The dataset accessible through yfinance is extensive, covering a vast array of securities and financial instruments worldwide. It includes daily, weekly, and monthly historical price data, as well as financials and performance metrics for thousands of stocks and other assets, potentially amounting to millions of data points. This makes it highly suitable for comprehensive market analysis, academic research, and the development of sophisticated financial models and applications.

Summary Statistics: As an example, the yfinance.Ticker(*insert ticker*) function in this library provides easy access to 53 attributes including stock ticker, market day/year lows/highs, historical averages, volume, 3 and 5-year average returns, and many more. Running yfinance.Ticker(**insert ticker**).history("1mo") has 9 attributes and gives you the stock price history for each business day of the past month in a Pandas Dataframe.

How It Was Used: In this project, used Yahoo finance to get 2024 daily performance data on ETFs and stocks. This is because CRSP only had data through 2023. Additionally, used it to get the sector and industry for stocks that were identified as a top 10 holding in at least one of the Vanguard ETFs.

Database:

Data Ingestion Procedure:

For the ETFGlobal dataset and CRSP dataset, used the WRDS Python library to access the WRDS API from Python and load the data into a Pandas DataFrame using the WRDS.Connection().raw_sql() function to execute SQL queries on the various tables needed to populate our MySQL DB on AWS. To get the ETF constituents and their weight % within the ETF, queried the etfg.constituents table and retrieved the stock holdings for each Vanguard ETF. To get the ETF daily performance and the daily performance of the individual constituents within each ETF, queried the crsp.dsf table.

To get the industry/sector info for each stock and 2024 performance data used the yfinance library in Python to get the mappings on Yahoo Finance. The sector/industry data was merged, using Pandas, with the CRSP data on the stock ticker column.

Once the data was ready to be exported, used the pandas.DataFrame.to_csv() function on each of the pandas DataFrames that constructed in the preprocessing/cleaning section. This generated a CSV file export for each dataframe. Then used DataGrip to import these CSV files into our MySQL database on AWS into different tables.

Entity Resolution Diagram:

Web App description:

1. **User Authentication:**
   Secure login and registration process ensures user privacy and protects watchlist data.
2. **My Watchlist Page:**

- Watchlist Management:
  - Function: Monitor the performance of the ETFs a user adds to the watchlist
  - Features: Users can effortlessly add and remove ETFs from their personalized watchlist

3. **Top Movers Page:**

- List of Gainers/Losers
  - Function: Discover the most significant gainers and losers among Vanguard ETFs on the given day.
  - Features:
    - Users can search the ETF by name or symbol or sector;
    - Users can add the ETF from the list to their watchlist
    - Users can remove the ETF from the list if in their watchlist
    - Users can sort the ETF by given fields
    - Users can change the number of ETFs to show per page

4. **Recommendations Page:**

- Recommendations Management:

    - Function: Provide a dynamically updated portfolio of recommended ETFs based on performance, risk, and sector makeup of ETFs in the user's watchlist
    - Features:
        - Users can select between 3 tabs: performance, risk, and sector
        - On each tab is a curated list of ETFs with relevant performance metrics
        - ETFs are recommended based on proximity of average daily returns, volatility, and a weighted sector makeup of ETFs in the user's Watchlist for the performance, risk, and sector sections respectively

5. **Performance Metrics Page:**

- <u>Time-Series Performance Analytics:</u>
    - Function: Key performance indicators like price change, percentage change, and daily volume are readily available for ETFs and their top 10 stock holdings
    - Features:
        - Users can deep dive on the ETF to show the time series of the ETF price trend, key performance indicators and top 10 stock holdings.
        - Users can also click on stock symbols that direct to the stock page that contains a time series chart, key performance indicators and other information about this

API Spec

Path: /login

Method: POST

Description: Logs in a user by verifying provided username and password and storing credentials in the session

Request parameters (uses body parameters)

- email (string) email address of user
- password (string) password of user

Response parameters

- message (string) for outcome of route
- user (object)

    - id (string): the CID assigned to session user

    - name (string): the name assigned to session user

Path: /logout

Method: POST

Description: Logs out the current user

Request parameters: None Response
parameters

- message (string) for outcome of route

Path: /clients

Method: POST

Description: Creates and stores a new client (account) Request
parameters (uses body parameters)

- name (string): name of new user
- password (string): password of new user
- email (string): email of new user

Response parameters

- message (string) for outcome of route
- CID (int): CID assigned to new user

Path: /clients

Method: DELETE

Description: Deletes the client (account) from the system Request
parameters (uses body parameters)

- email (string): email of user
- password (string): user's password

Response parameters

- message (string) for outcome of route

Path: /watchlist

Method: GET

Description: Gets the current user's watchlist

Request parameters: None Response
parameters

- message (string) for outcome of route message (string) for outcome of route
  - data (array), the SQL results corresponding to the watchlist

Path: /watchlist/:CID

Method: POST

Description: Adds an ETF to the current user's watchlist

Request parameters (uses body parameters)

- PERMNO (int): the PERMNO of the ETF to add Response parameters
- message (string) for outcome of route message (string) for outcome of route
  Path: /watchlist/remove

Method: DELETE

Description: Removes an ETF from the current user's watchlist

Request parameters (uses body parameters)

- PERMNO (int): the PERMNO of the ETF to remove Response parameters
- message (string) for outcome of route message (string) for outcome of route

Path: /etfs

Method: GET

Description: Gets all ETFs (tickers)

Request parameters: None Response
parameters

- message (string) for outcome of route message (string) for outcome of route
  - results (array) containing all tickers of ETFs

Path: /etfs/:PERMNO

Method: GET

Description: Gets ETF information about given ETF

Request parameters

- PERMNO (int): ETF PERMNO to get info about Response parameters
- message (string) for outcome of route message (string) for outcome of route
  - result (object) containing single query result

Path: /etf-prices/:PERMNO

Method: GET

Description: Gets ETF price information about a given ETF

Request parameters

- PERMNO (int): PERMNO of ETF to get price info about Response parameters
- message (string) for outcome of route message (string) for outcome of route
  - results (array) containing query results including all dates in DB for ETF

Path: /etf-holdings/:ETF_PERMNO

Method: GET

Description: Gets the top 10 holdings (stocks) of the given ETF Request parameters

- ETF_PERMNO (int): ETF PERMNO to get holding information about

Response parameters

- message (string) for outcome of route message (string) for outcome of route
  - results (array) containing the top (up to) 10 holdings from the query

Path: /etf-industry/:ETF_PERMNO

Method: GET

Description: Gets the breakdown of industry by percentage for given ETF

Request parameters

- ETF_PERMNO (int): ETF PERMNO to get industry info about Response parameters
- message (string) for outcome of route message (string) for outcome of route
  - results (array) for outcome of the query

Path: /stocks/:PERMNO

Method: GET

Description: Gets general information about the given stock

Request parameters

- PERMNO (int): Stock PERMNO to retrieve info about Response parameters
- message (string) for outcome of route message (string) for outcome of route
  - results (array) containing outcome of the SQL query

Path: /stock-prices/:PERMNO

Method: GET

Description: Gets price information about the given stock Request parameters

- PERMNO (int): Stock PERMNO to get price info about Response parameters
- message (string) for outcome of route message (string) for outcome of route
  - results (array) including all price information corresponding to the stock

Path: /stock-composites/:Stock_PERMNO

Method: GET

Description: Gets top 10 holders (ETFs) of the given stock. That is, those ETFs for which the stock comprises the greatest percentage. Request parameters

- Stock_PERMNO (int): Stock PERMNO to get composite information about

Response parameters

- message (string) for outcome of route message (string) for outcome of route
  - results (array) representing the top 10 holders of the stock

Path: /search

Method: GET

Description: Search ETFs and Stocks when given a ticker, name, or PERMNO

Query parameters:

- query (string), the query that can match to a Stock name, ticker, or PERMNO or ETF name, ticker, or PERMNO Response parameters:
- message (string) for outcome of route
- results (array) all results corresponding to the search query

Path: /etf-volatility

Method: GET

Description: Get volatility of all ETFs sorted in descending order

Query parameters: None Response parameters:

- message(string) for outcome of route
- results (array) all results corresponding to volatility query

Path: /etf-returns

Method: GET

Description: Get annualized returns for top 10 performing ETFs in last 10 years

Query parameters: None Response
parameters:

- message(string) for outcome of route
- results (array) all results corresponding to annualized returns query

Path: /etf-volume

Method: GET

Description: Get average daily volume for top 10 ETFs

Query parameters: None Response
parameters:

- message(string) for outcome of route
- results (array) all results corresponding to volume query