**📁 Main Files and Their Roles**

**1.**[day.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** The central orchestrator of your trading dashboard and analytics.
* **Responsibilities:**
  + Imports and initializes all major modules and APIs (Schwab, E\*TRADE, Finnhub, QuiverQuant, etc.).
  + Loads and processes historical and real-time data for ETFs and stocks.
  + Calculates technical indicators (ADX, PMO, CCI, ATR, etc.).
  + Aggregates and ranks ETFs/stocks using both technical and alternative data (news, whale/institutional/government trades).
  + Runs a Dash web dashboard for visualization and trade logging.
  + Schedules and manages background jobs for data refresh, news/whale cache updates, and AI recommendation refreshes.
  + Handles configuration/settings via [settings.json](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") and GUI ([day\_settings\_gui.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)).

**2.**[day\_settings\_gui.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** GUI for editing app settings.
* **Responsibilities:**
  + Provides a Tkinter-based interface for users to adjust settings (e.g., dashboard interval, volatility lookback bars).
  + Reads and writes to [settings.json](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ").

**3.**[settings.json](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ")

* **Role:** Stores user-configurable settings.
* **Responsibilities:**
  + Holds values like dashboard\_interval (minutes per bar) and volatility\_lookback\_bars (bars for volatility/indicator calculations).

**4.**[historical\_data.csv](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** Stores historical OHLCV data for all tracked tickers.
* **Responsibilities:**
  + Used for technical analysis, dashboard plotting, and candidate selection.

**5.**[Top\_ETFS\_for\_DayTrade.xlsx](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** List of ETF symbols to track and analyze.
* **Responsibilities:**
  + Provides the universe of symbols for the day app to process.

**6.**[trade\_log.xlsx](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**(and/or**[sso\_sds\_trade\_log.csv](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**)**

* **Role:** Stores trade logs for executed trades.
* **Responsibilities:**
  + Used for tracking, reporting, and displaying trade history and P/L.

**7.**[ai\_module.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** Provides AI-based trade recommendations.
* **Responsibilities:**
  + Analyzes tickers and returns top trade ideas with probabilities, entries, targets, stops, and recommendations.

**8. Schwab\_auth.py,**[schwab\_data.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** Schwab API authentication and data fetching.
* **Responsibilities:**
  + Handles OAuth, token refresh, and data retrieval from Schwab.

**9.**[etrade\_auth.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** E\*TRADE API authentication and data fetching.
* **Responsibilities:**
  + Handles OAuth and data retrieval from E\*TRADE.

**10.**[rank\_top5\_etfs.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** Ranks ETFs based on composite scores.
* **Responsibilities:**
  + Combines technical, news, and whale data to rank and select top candidates.

**11.**[edgar\_whale.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** Fetches and processes SEC 13F filings for whale/institutional data.
* **Responsibilities:**
  + Supplements whale/institutional data for tickers.

**12.**[screen\_dividends.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**(optional/related)**

* **Role:** Screens for dividend continuity using ML.
* **Responsibilities:**
  + Trains and applies a logistic regression model for dividend-paying stocks.

**13.**[SSO\_SDS\_Trade\_strategy.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)

* **Role:** GUI and logic for SSO/SDS trading strategy and trade log.
* **Responsibilities:**
  + Provides a Tkinter GUI for viewing and logging SSO/SDS trades, including P/L calculations.

**🧩 How the Day App Works**

**Startup Sequence**

1. **Configuration:** Loads settings from [settings.json](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") (intervals, lookback, etc.).
2. **Authentication:** Ensures valid Schwab and E\*TRADE tokens via their respective auth modules.
3. **Data Loading:** Loads historical OHLCV data from CSV or fetches from Schwab if missing.
4. **Symbol Universe:** Loads ETF symbols from [Top\_ETFS\_for\_DayTrade.xlsx](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html).
5. **AI Recommendations:** Calls [ai\_module.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) to get trade recommendations for all tickers.
6. **Technical Calculations:** Computes indicators (ADX, PMO, CCI, ATR, etc.) for all tickers.
7. **Alternative Data:** Loads news and whale/institutional/government data from caches/APIs.
8. **Ranking:** Ranks all tickers using a composite of technical and alternative data.
9. **Dashboard Launch:** Starts a Dash web app for visualization, trade logging, and settings access.
10. **Scheduler:** Starts background threads for periodic data refresh, news/whale cache updates, and AI refresh.

**During Operation**

* **Real-Time Data:** Continuously updates historical data with real-time bars (via Schwab streaming or polling).
* **Alerts:** Triggers audio/visual alerts for technical, news, or whale events.
* **User Interaction:** Users can view the dashboard, log trades, and adjust settings via the GUI.
* **Trade Log:** Trades are logged and P/L is calculated and displayed in both the dashboard and the SSO/SDS trade log GUI.
* **Background Jobs:** Scheduler ensures all data and caches are kept up to date.

**📝 Summary of What the Day App Does**

* **Aggregates and analyzes real-time and historical data for a universe of ETFs/stocks.**
* **Calculates technical indicators and combines them with news and whale/institutional/government data.**
* **Uses AI to recommend top trade candidates.**
* **Ranks and displays the best opportunities in a web dashboard.**
* **Provides real-time alerts for actionable events.**
* **Allows users to log trades and track P/L.**
* **Offers a GUI for adjusting key settings and reviewing trade logs.**
* **Automates data refresh, news/whale cache updates, and AI recommendation updates via a scheduler.**

**🗂️ File/App Map (Quick Reference)**

| **File/App** | **Purpose/Role** |
| --- | --- |
| [day.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Main orchestrator, dashboard, analytics, scheduler |
| [day\_settings\_gui.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Settings GUI (Tkinter) |
| [settings.json](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | User settings (intervals, lookback, etc.) |
| [historical\_data.csv](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Historical OHLCV data |
| Top\_ETFS\_for\_DayTrade.xlsx | List of tracked ETF symbols |
| [trade\_log.xlsx](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) / .csv | Trade log and P/L tracking |
| [ai\_module.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | AI trade recommendations |
| [Schwab\_auth.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Schwab API authentication |
| [schwab\_data.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Schwab data fetching |
| [etrade\_auth.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | E\*TRADE API authentication |
| [rank\_top5\_etfs.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | ETF ranking logic |
| [edgar\_whale.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Whale/institutional data from SEC filings |
| [screen\_dividends.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | Dividend continuity screening (ML) |
| [SSO\_SDS\_Trade\_strategy.py](vscode-file://vscode-app/c:/Users/mjmat/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) | SSO/SDS trade log GUI and logic |