

From Zero to Your First Signal

A Step-by-Step Guide to Launching Your Personal Stock Trading Dashboard

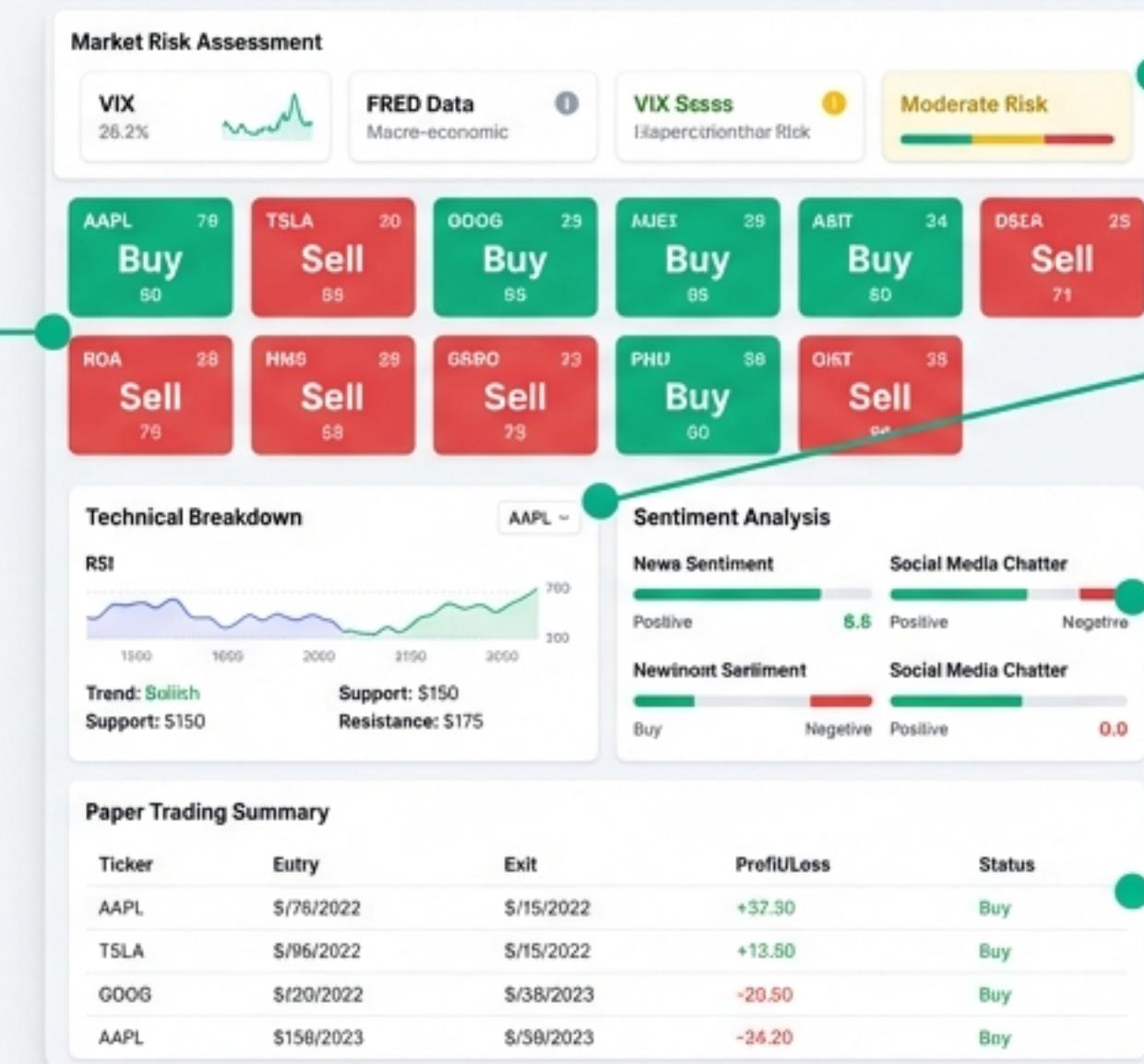


100% FREE Data Sources.
No Recurring Costs. **Forever.**

This is Where We're Going: Your Personal Trading Dashboard

Signal Cards

At-a-glance trading signals, color-coded by conviction.



Market Risk Assessment

A macro-economic weather report for the market (VIX, FRED data).

Technical Breakdown

Key indicators like RSI, trend, and support/resistance levels.

Sentiment Analysis

Scores based on news and social media chatter.

Paper Trading Summary

Track the performance of mock trades to validate signals risk-free.

This entire dashboard is generated on your computer using free, publicly available data.
We will walk you through every step to get it running.

Our Journey in Three Simple Parts



Part 1: Assemble Your Toolkit

We'll install the two essential (and free) pieces of software required to run the project: Git and Python.



Part 2: Bring the Code Home

Using a few simple commands, we'll download the complete Stock Trader project from its home on GitHub to your computer.



Part 3: Launch & Configure

We'll start the application using its user-friendly Graphical Interface (GUI), add your free API keys, and generate your very first dashboard.

Step 1: Assembling Your Toolkit



WHAT

Git is a system that helps manage and track changes in code projects.

WHY

We need it to download a perfect, up-to-date copy of the Stock Trader project from its home on GitHub.

[Download Git for Windows](#)

WHAT

Python is the popular programming language the Stock Trader application is written in.

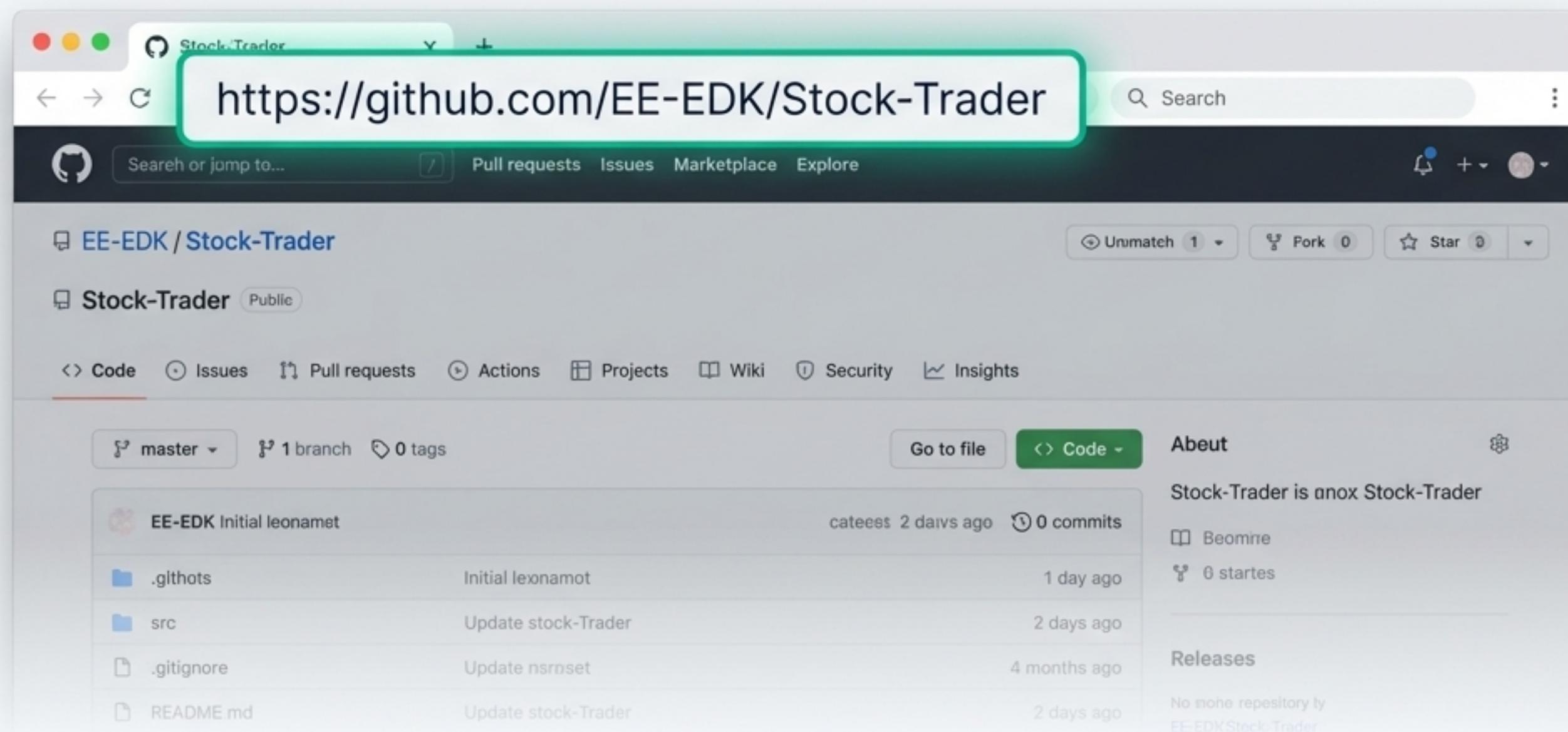
WHY

It's the engine that will run the code to collect data and generate your dashboard.

[Download Python](#)

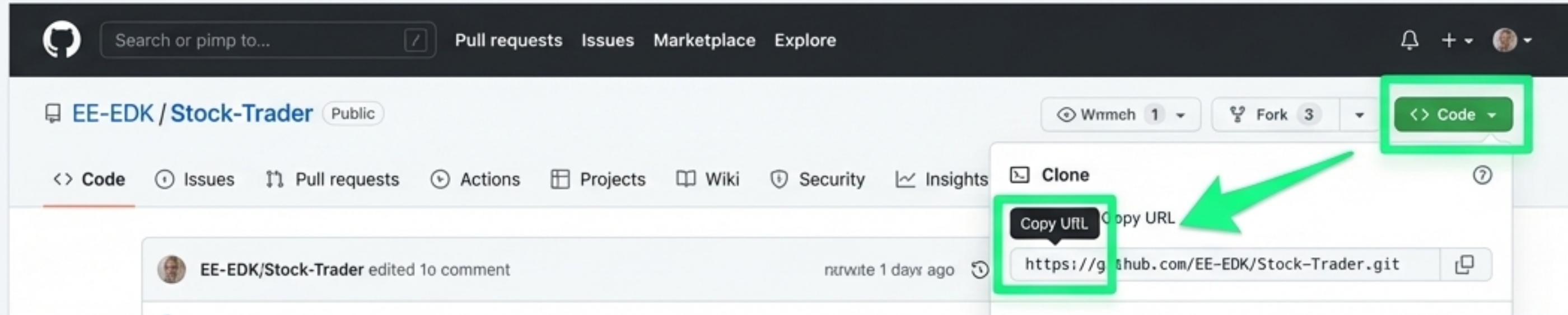
**During installation, be sure to check the box that says 'Add Python to PATH'.*

Step 2: Locating the Project's Home Base on GitHub



GitHub is a platform where developers host and share their code. Our first step is to navigate to the Stock Trader project page. This is the single source of truth for all the application's code.

Step 3: Bringing the Code Home with a Single Command



```
C:\Users\YourName> git clone https://github.com/EE-EDK/Stock-Trader
Cloning into 'Stock-Trader'...
remote: Enumerating objects: 1234, done.
Receiving objects: 100% (1234/1234), 1.23 MiB | 4.56 MiB/s, done.
Resolving deltas: 100% (567/567), done.
```

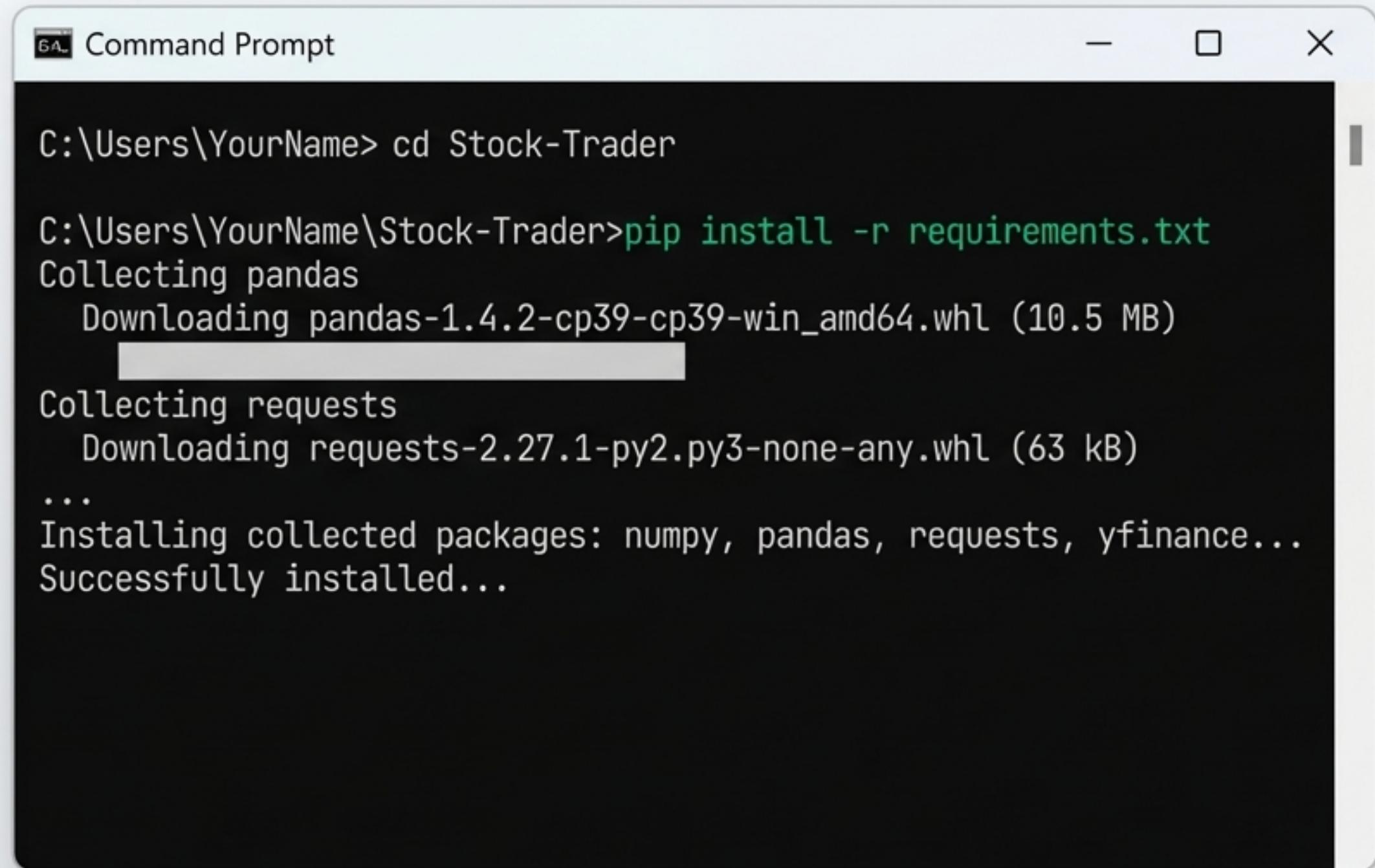
WHAT

The `git clone` command copies a project from a remote location (like GitHub).

WHY

This command downloads the entire Stock Trader application and all its files into a new folder called `Stock-Trader` on your computer.

Step 4: Installing the Project's Building Blocks



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following command and its execution:

```
C:\Users\YourName> cd Stock-Trader
C:\Users\YourName\Stock-Trader>pip install -r requirements.txt
Collecting pandas
  Downloading pandas-1.4.2-cp39-cp39-win_amd64.whl (10.5 MB)
[██████████] 100% |██████████| 10.5 MB 1.0MB/s
Collecting requests
  Downloading requests-2.27.1-py2.py3-none-any.whl (63 kB)
...
Installing collected packages: numpy, pandas, requests, yfinance...
Successfully installed...
```

WHAT

`pip` is Python's Package Installer. It reads the `requirements.txt` file, which is a shopping list of all the other code libraries the project needs to function.

WHY

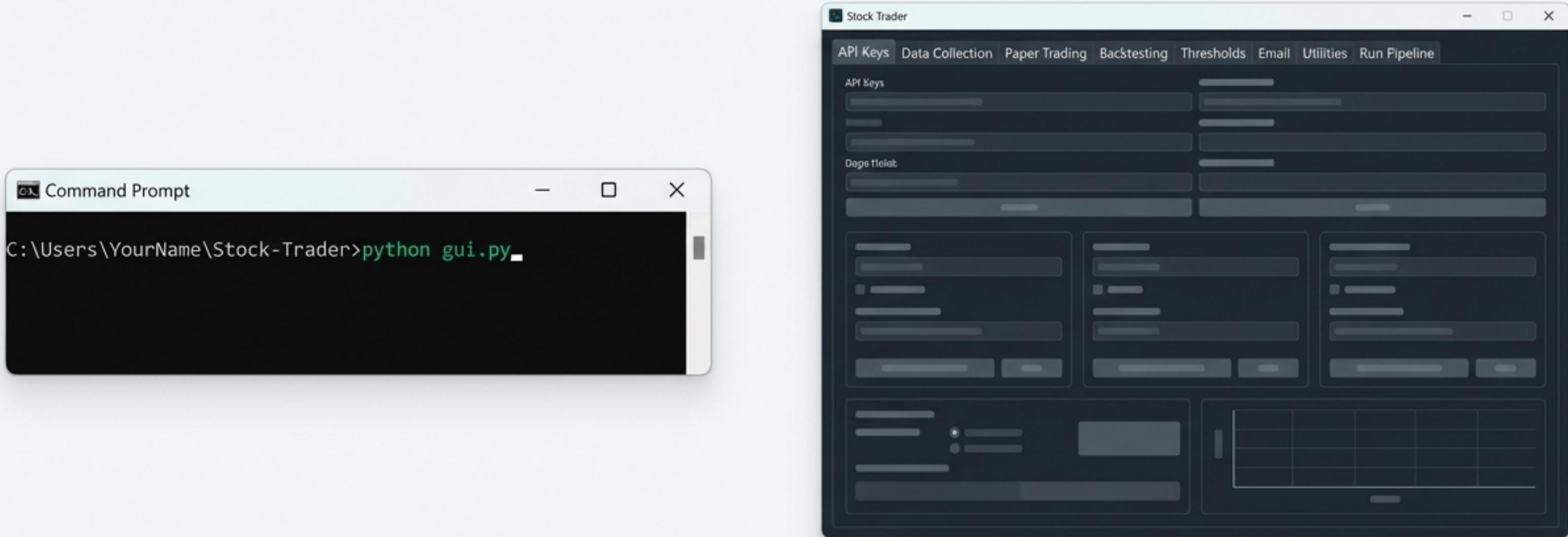
This single command automatically finds, downloads, and installs all the necessary dependencies, such as `pandas` for data analysis and `requests` for accessing APIs.

The Hard Part is Over. Meet Your Control Center: The GUI.

While this tool can be run entirely from the command line, it includes a full-featured Graphical User Interface (GUI) designed for easy configuration. For beginners, this is the recommended path.

Feature	GUI	Command Line
Configuration	Visual forms	Edit YAML file
API Key Setup	Copy/paste in tabs	Edit text file
Pipeline Execution	Click button	Run `python main.py`
Best For	Beginners, visual learners 	Advanced users, automation

Step 5: Launching the Application



With all the setup complete, a single command launches the entire control center. Welcome to the Stock Trader GUI.

Powering Up the Data Engine: Adding Your Free API Keys

The application pulls data from various sources using APIs. Think of an API key as a unique password that gives our program permission to request data. We'll start with the two most important ones, which are free and take minutes to get.

1. Finnhub (Required)

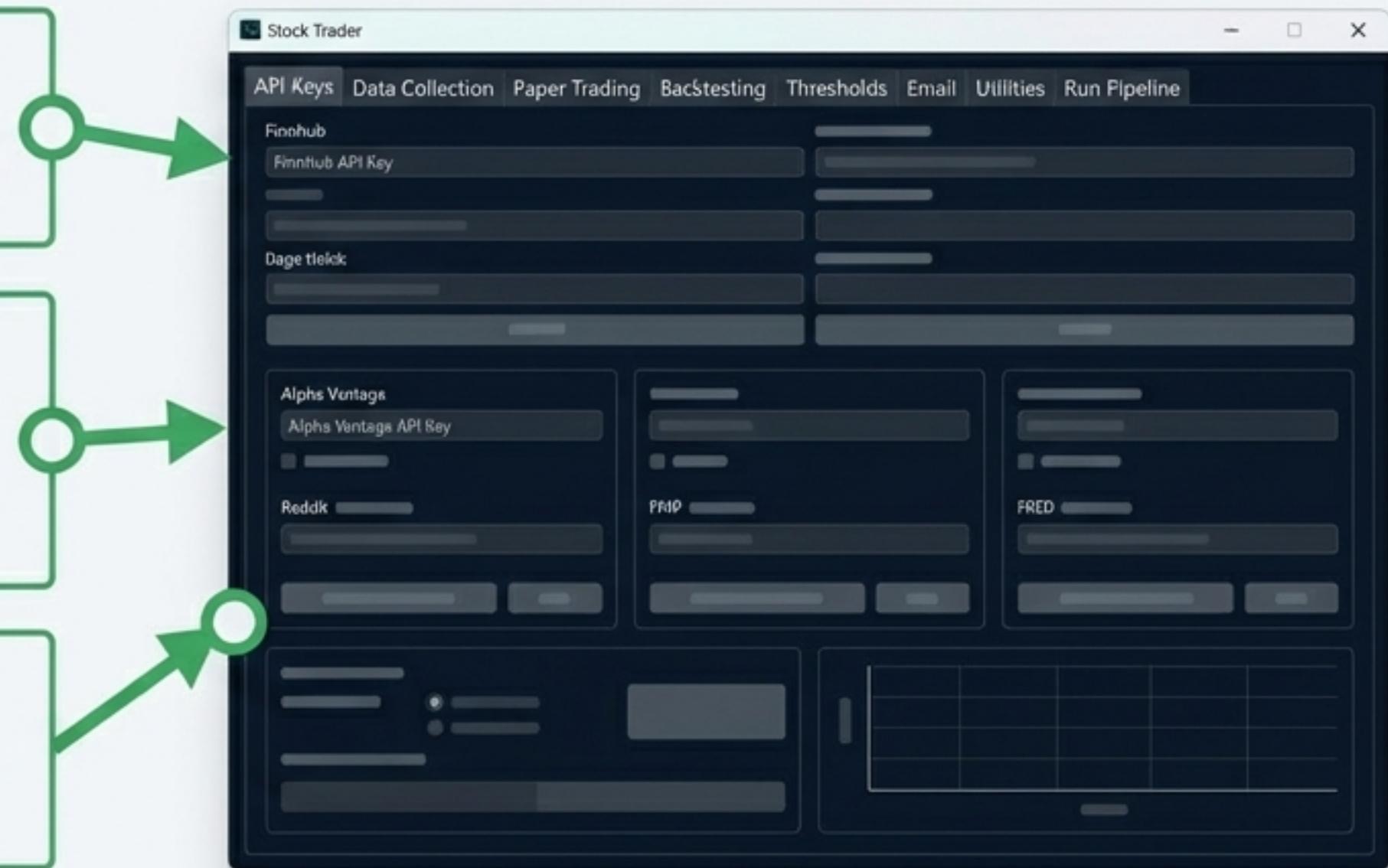
Get stock prices. Visit <https://finnhub.io/register> to get your free key.

2. Alpha Vantage (Recommended)

Get news sentiment. Visit <https://www.alphavantage.co/support/#api-key> for your free key.

3. (Optional)

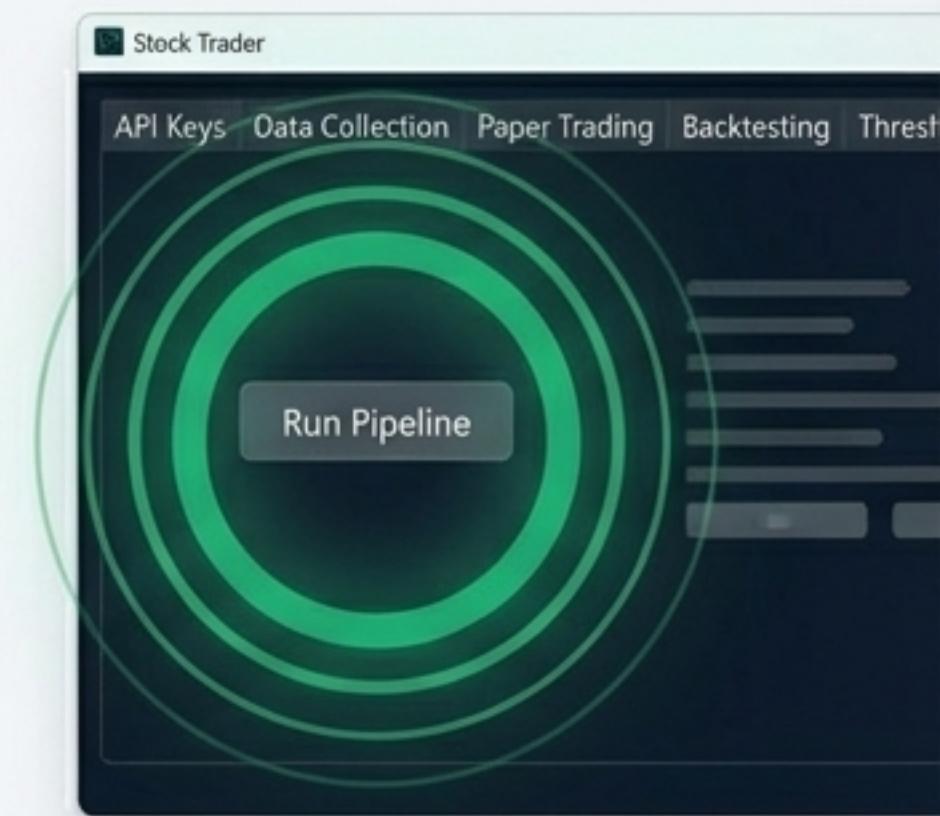
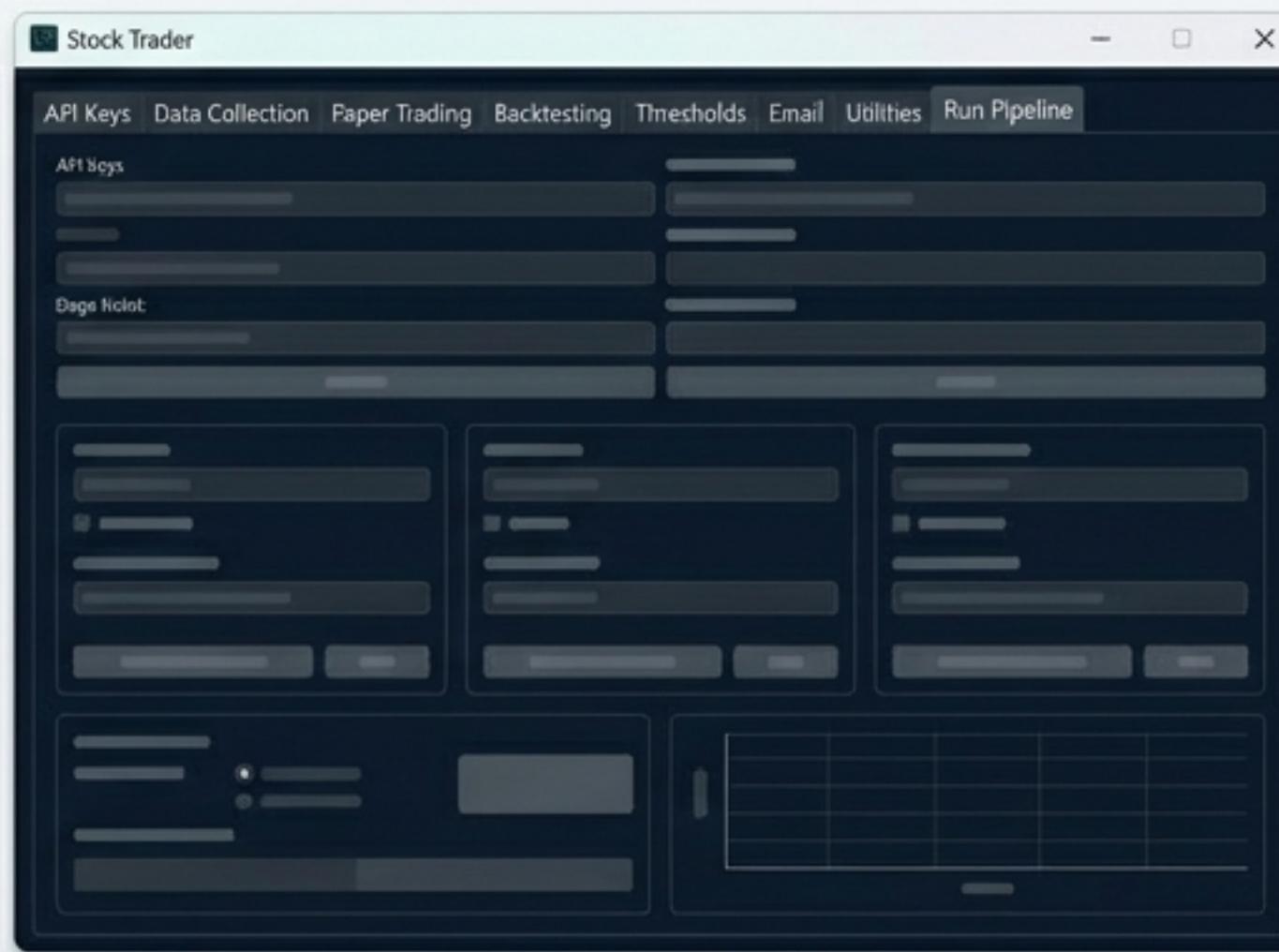
You can add more data sources like Reddit later. The system works great with just these two.



Step 6: Running the Pipeline to Fetch Your Data

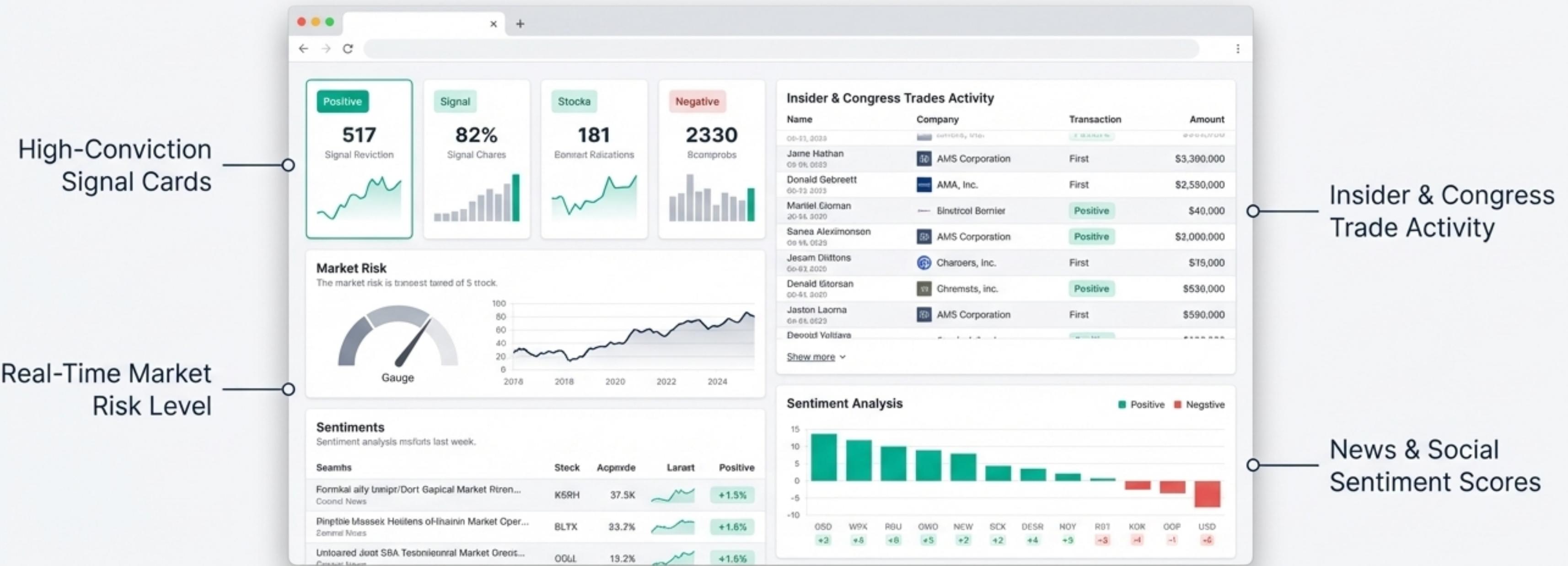


First, click ‘Save Configuration’ to save your new API keys.



Now, simply click ‘Run Pipeline’ to begin fetching data. For your first run, the default settings are perfect.

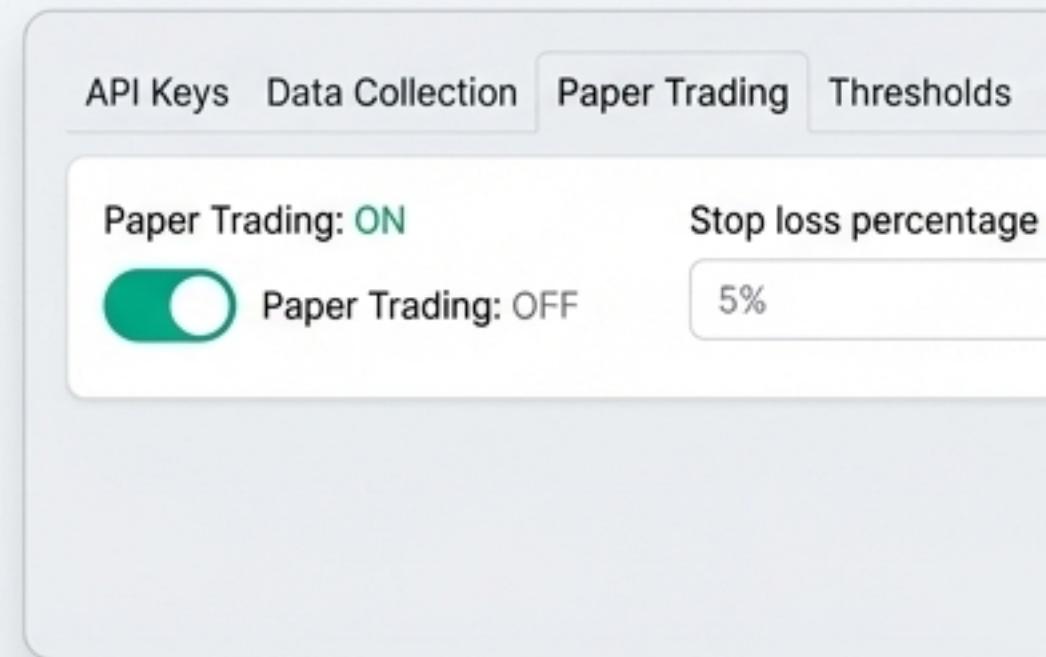
Success! Your First Dashboard is Live.



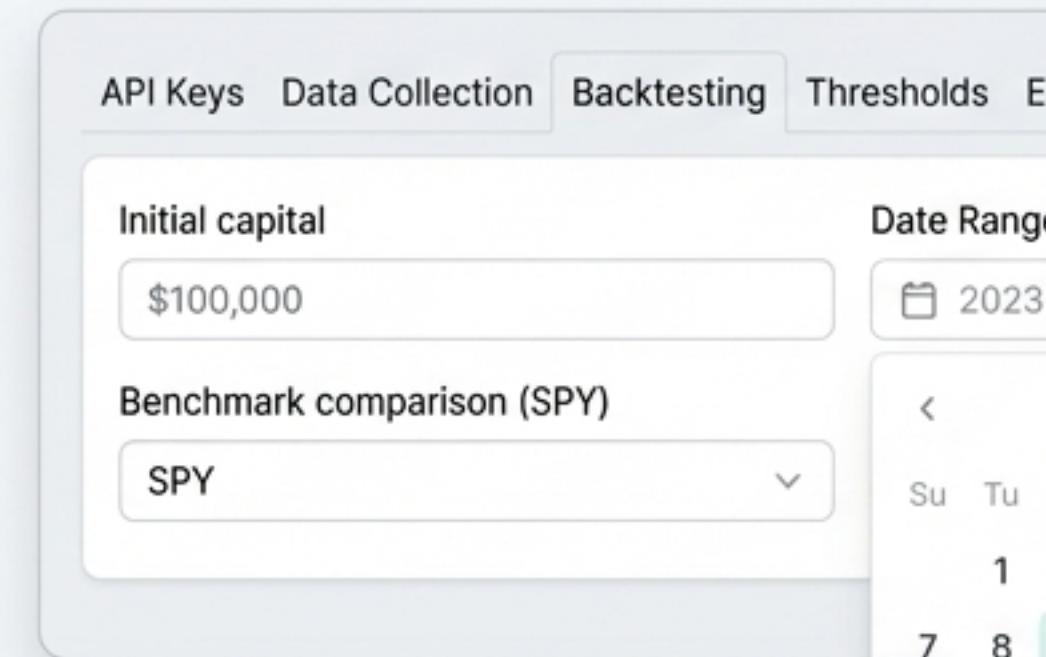
After the pipeline finishes, a file named `stock_dashboard.html` is created in the project folder. Open it in any web browser to see your results. This dashboard will update every time you run the pipeline.

Just the Beginning: A Universe of Analysis Awaits

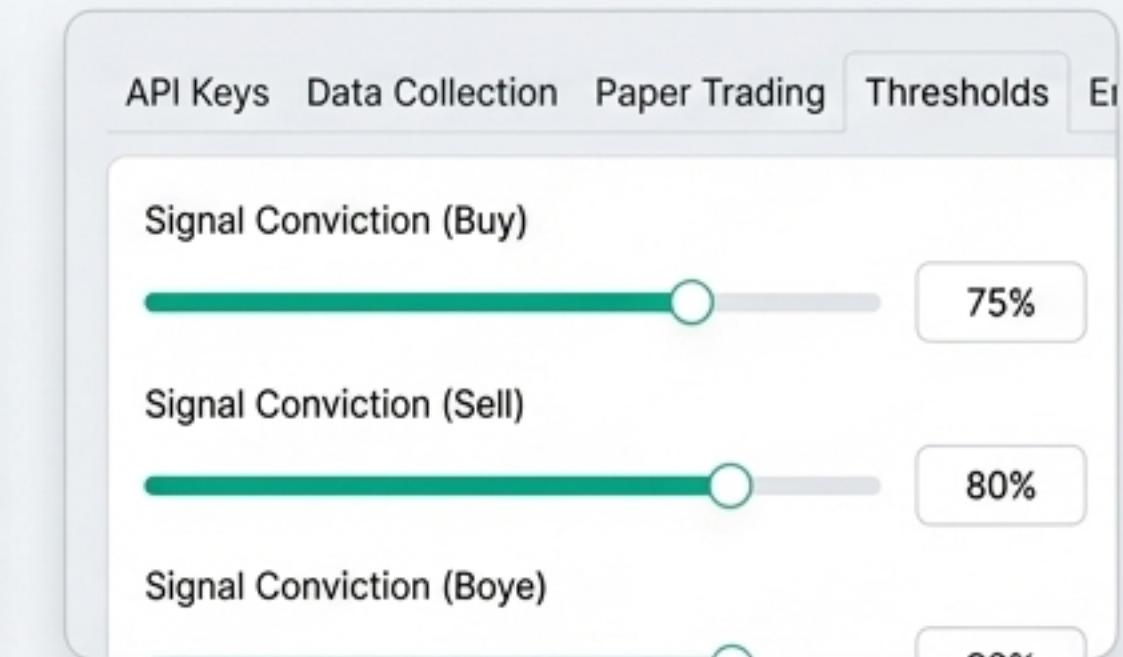
Your dashboard is just the start. The GUI gives you full control to fine-tune your strategy. Explore the other tabs to unlock the full power of the system.



Test Signals Risk-Free with the Paper Trading System.



Validate Your Strategy against historical data using the Backtester.



Fine-Tune Your Signals by adjusting the detection thresholds.

You've Gone From Zero to a Live Dashboard. What's Next?

You have successfully set up a powerful, automated stock analysis pipeline. Congratulations!



Explore

Dive into your dashboard. Analyze the signals and see how they correspond to market movements.



Experiment

Adjust settings in the GUI. Try enabling [Paper Trading](#) or [changing signal thresholds](#) and re-run the pipeline to see how your results change.



Engage

Have questions or ideas? Join the discussion or report issues on the project's [GitHub](#) page.