

MVP

Team Name: [insert name here] (Group 4)

Team Members (NetIDs): harinis4, jangres2, sangini2, alvins3, mka7

Team PM (NetIDs): shrestd2

What problem is your project trying to solve?

(What audience is it for? How will your project solve the issue?)

- Targeting people with little knowledge about trading
 - Help give trading advice that corresponds to general market sentiment to boost profitability
- Too much data and understanding required to trade effectively
 - Most traders lose money due to a lack of a chance to trade effectively
 - Fix this disparity and level the field by congregating market data in a user-friendly experience for trading advice
- By having computers go through the large spread of data available (including market sentiments from news sites, discussion forums, shareholder meetings), we can make create a generalized report for beginner traders to use for stocks

;

What features make up your MVP (Minimum Viable Product)?

(What features do you hope to accomplish by the end of the semester?)

- Scraping of some news source (tbd specific source)
- Sentiment analysis for parsed data
- Leverage LLM understanding of sentiments to create detailed trading strategies
- User Preferences (risk-tolerance, investment capital, activity level, interested stocks/sectors, targeted return, etc.)
- Trading frontend dashboard for users
- User login to see previous data

What are some additional features outside the MVP?

(A list of features that would be nice to add after finishing the MVP.)

- Confidence in analysis (looking at source material ethos)
- Different trading strategies (401k, HYSA, Roth IRA, Bonds, 529)
- Give platform user move (eg. 80% of users are buying Google)
- Advanced Ordering (OTTO, Stop/Loss, OCTO, options, arbitrage)
- Intra-day trading advice and notifications/alerts
- Portfolio statistics
- Diversification advice
- Profitability (make money through trading commissions)
- Daily recommendations (stock of the day)

Which does the tech stack look like and why did you choose these over alternatives?

(Feel free to discuss with your PM! Examples: React, Python, Java, etc. You do not need to know how to use these right now.)

- Python, Flask, Javascript, React, Gemini (or other LLM agents), BeautifulSoup, RESTful architecture, Reddit/Yahoo Finance API (or others), VsCode, Github, Google OAuth, Vercel or other host, Huggingface ABSA (or other)
 - We chose this tech stack as they integrate cleanly and are easy to learn and develop skills in

What will the project timeline look like?

SPRINT 0 (09/15):

- Project Ideas

SPRINT 1:

- MVP + environment setup

SPRINT 2:

- Getting used to tech stack

SPRINT 3:

- Building basic functionality from tutorial

SPRINT 4:

- Create base working model for front and back end

MIDTERM PRESENTATION (10/18-10/19):

SPRINT 5:

- Clean up front end and back end design

SPRINT 6:

- Create API architecture

SPRINT 7:

- refine

SPRINT 8:

- refine

SPRINT 9:

- refine

FINAL PRESENTATION (12/6-12/7)

(Discuss this with your PM as well! You don't have to stick to it, but this should give you a general guideline for how the project should progress.)