Project Proposal

Jasmina Frederico, Gia Mazza, Claudia Espejo

Department of Computer Science, Lafayette College

CS 470: Senior Project

February 13, 2026

Table of Contents

OVERVIEW

Title

TrustMeBro Analytics Betting Markets Dashboard and Chatbot

Team

TrustMeBro Analytics. **Members:** Jasmina Frederico, Gia Mazza, Claudia Espejo

Client

Lafayette College Policy Studies Department - Christo Maheras

Abstract

Polymarket is the world's largest prediction market, and our client hypothesizes that inefficiencies exist within it. Our project aims to use Polymarket's and Kalshi's publicly available data to identify, track, measure, and ultimately exploit these inefficiencies for research and, potentially, financial purposes. We will create a visually appealing website featuring live, data-driven dashboards and an integrated artificial intelligence (AI)-powered chatbot. This will allow our client to quickly see important information and discuss with the chatbot to learn more. Ultimately, our solution will allow our client to perform efficient, impactful research before committing to any critical financial decisions.

Motivation

Our Client's Problem: There are no current tools to educate prediction market betters on historical data trends from various events and provide users with signals to optimize their bets. There is no publicly available information about insider trades and market manipulation on sites such as Polymarket, putting regular users at a disadvantage. With the development of this site, naive users of prediction markets can educate themselves on their bets in real-time without navigating to multiple sites.

STAKEHOLDERS & SCOPE

Intended Users

Our project is designed for multiple audiences, including

- Individual traders seeking to make more informed decisions
- Investment clubs (or similar organizations) and other groups exploring collective strategies
- Policy analysts and researchers studying public expectations
- Academic or media audiences interested in how prediction markets process information

User involvement can range from passive explanations to active hypothesis testing.

Other Stakeholders

Key stakeholders include the development team (TrustMeBro Analytics), our team leader (Gia Mazza), our manager (Professor Xia), the client (Christo Maheras), and end-users such as individual traders, investment clubs, and policy analysts. ‘
Christo Maheras and the Lafayette College Policy Studies Department will own the project when it is completed in May 2026.

Goals

1. Create a website with data visualizations relating to the top Polymarket events/markets.
2. Create a visually appealing dashboard highlighting prediction market trends over time.
3. Create an integrated AI chatbot that can answer user questions in real time.
4. Educate application users on data trends on Polymarket.
5. Provide real-time assistance while making bets on Polymarket.
6. Explicitly and clearly evaluate inefficiencies for users to understand.

Non-Goals

1. We will not be tracking data regarding sports betting, per our client’s request.
2. Our chatbot will not autonomously place bets on behalf of the user or itself.
3. Our chatbot will not provide direct financial advice for ethical reasons.

Deliverables

Our solution is to create a single website where users can educate themselves on the top Polymarket events/markets in order to make more informed bets. Our final website will include:

1. Current data and historical data trends visualization dashboard based on real-time data from both Polymarket and Kalshi.
2. An artificial intelligence chatbot integrated into the dashboard to recommend bets to place based on the data we track (see *Data Collection*).

Our website will be entirely unit-tested and beta-tested before handoff to the client.

PROJECT REQUIREMENTS

Functional Requirements

Project Requirements

1. The website will display data that relates to relevant Polymarket events/markets.
2. The website will have an AI chatbot integrated into the page.
3. Clients can enter text into the chatbot as input and will see the chatbot’s output as text and/or relevant visualizations.
4. Clients can filter events/markets by category (e.g., politics, economy, entertainment).
5. The website will display data in clear and easily readable visualizations.

Nonfunctional Requirements

Accessibility

- The site should be accessible via both desktop and mobile.
- The site will use a color scheme that is accessible to most viewers.

Performance

- The site should load data onto the site in under a second.

Security

- Users cannot upload their own data into the site or edit site data.
- Users cannot manipulate the AI agent.
- The AI agent cannot make transactions.

PROPOSED SOLUTION & PROJECT SPECIFICATIONS

Data Collection

- Popular markets
- Extremely large bets
  - Could indicate an insider trade
- Number of posts with related hashtags to the event on social media
- Sentiment analysis on Reddit or subreddits
- Current news articles
- A series of incremental bets indicating increasing confidence
- Kalshi versus Polymarket to find differences between prediction apps

Identifying Market Inefficiencies

We will seek to examine several inefficiencies, including

- Related markets whose prices imply logically inconsistent outcomes (economic outcomes vs election results)
- Overconfidence or excessive certainty reflected in extreme probabilities
- Incremental bet-making
- Betting “whales” – extremely large bets
- Possible insider trades
- Unusual bet volume manipulating a market’s result; for example, Polymarket’s $7M Ukraine Mineral Deal Debacle Traced to Oracle Whale

Ethical Data Collection Statement

This project only uses publicly available data and official APIs, and all data will comply with the platform's terms of service. Sources such as Polymarket, Kalshi, news outlets, and social media platforms will be accessed through authorized, documented endpoints. We will not collect private information, bypass access controls, or attempt to identify individual users. Large bets and trading patterns are examined to understand market dynamics, but not to track personal identities or private information.

Backup Data Plan

If we are unable to access the data we need due to extenuating circumstances, we will still attempt to deliver the best solution possible with accurate data. Since our development is iterative, we can manually add data to the site to show the client our plan for future iterations. If needed, we can extrapolate data from past data available online on sites such as Kaggle.

Data Sources

- Polymarket API
    - Gamma
        - List events
        - Get events by id
        - List markets
        - List market by id
    - CLOB
        - Historical Timeseries Data
        - All historical trades can be fetched via the API
    - Data
        - Download an accounting snapshot (ZIP of CSVs)
    - WebSocket
        - Real-time views of data
- Kalshi API: https://docs.kalshi.com/api-reference/
    - Get trades
    - Get orders
    - Get positions
    - Get settlements
- News sources
    - The New York Times (political)
        - https://developer.nytimes.com/
            - Top Stories API
    - The Wall Street Journal (financial)
        - https://github.com/janlukasschroeder/realtime-newsapi
            - Query API: search for articles by title, description, or stock ticker
    - Washington Post (political)
        - https://rapidapi.com/thosedev-thosedev-default/api/washington-post
            - Retrieve news by category
- Reddit API
    - Append .json to subreddit URLs for the simplest usage
    - https://www.reddit.com/dev/api/#GET_{sort}
        - Get top posts from a subreddit

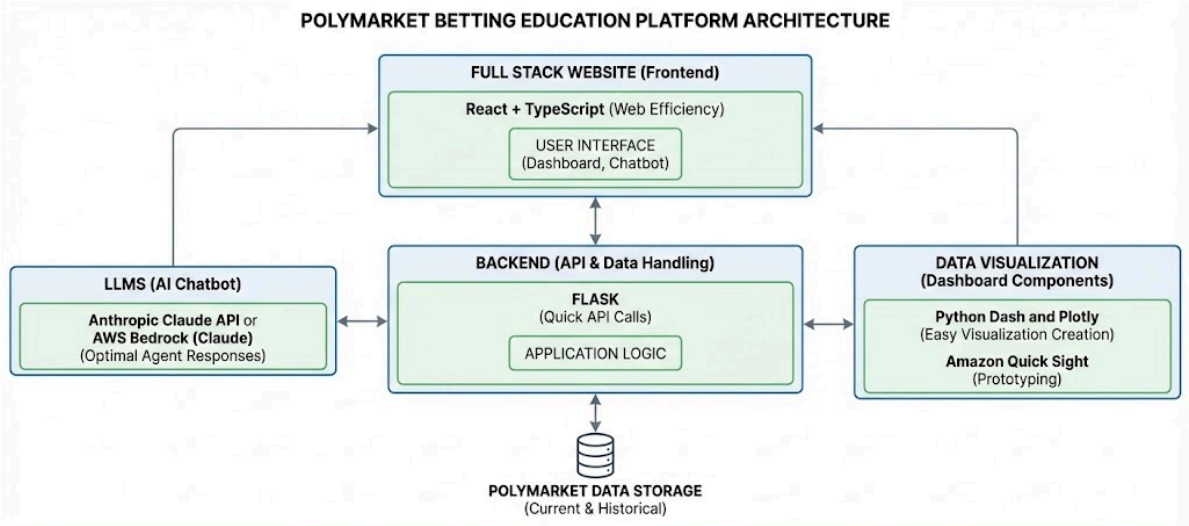
- Machine-generated financial prediction data https://www.synthdata.co/
    - A potential future iteration of this project could use Synthdata's API to fetch machine-generated predictions on different markets

Project Architecture

At the beginning of our project, we are not specifying our technical specifications in great detail, so as to prepare for changes throughout the project. We decided to create a simple stack to simplify.

- Full Stack Website
    - Designed with React + TypeScript for web efficiency
    - Flask backend for quick API calls
- Large Language Models (LLMs)
    - Anthropic Claude API or AWS Bedrock (Claude) for optimal agent responses
- Data Visualization
    - Python Dash and Plotly for easy visualization creation
    - Amazon Quick Sight for Prototyping
- Data Storage
    - Supabase

Architecture Diagram



This image was generated using Google's Nano Banana Pro.

System Design

Our project aims to leverage an advisor AI Agent paired with a dashboard, designed for both technical and non-technical users. Our key components include

- An interactive dashboard consisting of market timelines and inefficiency indicators
- AI-assisted analysis that uses natural language explanations of market movements and summaries of relevant external information

- Filtering mechanism to distinguish efficient markets from those with exploitable inefficiencies

This system is decision-support focused and not aimed at completing fully autonomous trading.

Unknowns

- Does the data have null values?
- How frequently does data update
- Will the data have conflicting values across sources?
- Will the LLM correctly interpret structured API data?

COMPETING SITES

Polymarket Analytics: https://polymarketanalytics.com/

- Displays live prices for active markets, lists, and ranks top traders by performance metrics
- Offers research, analysis articles, and insights into prediction markets and trader behavior
- Not a trading platform and does not offer financial recommendations

Polywhaler: https://www.polywhaler.com/

- Monitors high-value trades (over 10k), tracks wallet activity from large stakes to where large sums of money are being deployed
- Offers an AI-assisted section that highlights possible inefficiencies or informed trading
- Users can see the dashboard of metrics, a watchlist of users, and a whale leaderboard

Polymarket Tracker: https://polymarket-tracker.com/

- Site displays a dashboard to track users, markets, and comments

Poly Thoughts: https://polythoughts.com/

- Site for community discussion about events, aggregated sentiment, and watchlists
- Currently in development

PolyWallets: https://polywallets.xyz/

- Displays the largest Polymarket wallets and a full profile, consisting of all of their positions and previous data

Competitors vs Our Innovations

Throughout these different platforms, the emphasis on live dashboards and visibility into traders and market behavior was an aspect we wanted to incorporate into ours as well. While these platforms help users complete different tasks, our project stands out from the rest. Most competitors just track whales, display wallet-level performance, and lack educational explanations or assistance.

Our project analyzes structural inefficiencies, cross-market inconsistency, and incorporates relevant news from a wide variety of perspectives, all aiming to educate and help foster more transparent and effective market participation with the assistance of an AI agent.

VALIDATION PLAN

Evaluation

Measuring Success

Metrics

- The hallucination rate for the AI chatbot is < 10%
- Website loads data in < 1 second
- Chatbot response time is < 3 seconds

Test Plan

- Unit testing of all API functionality
- Test chatbot guardrails
- Verify dashboard updates correctly when API data changes
- Test failure handling when APIs are not functional
- Test mobile and desktop compatibility

User Study Plan

We will gather 8–10 users (representing a variety of financial and technical skill levels) to complete 3 predefined tasks each, testing dashboard navigation and chatbot insights. Then we will ask for qualitative feedback from each participant. We will also track the overall task completion rate, task completion time, and error rate. We will classify our work as success if over 80% of participants complete the tasks successfully and they feel the time spent was reasonable and productive. We will also note any issues to resolve going forward.

Baselines

We will compare our dashboard load speed to other popular web-based dashboards, and we will compare manual data lookup time to dashboard trend lookup time (regarding the same result).

Acceptance

- Dashboard successfully integrates with all specified APIs (see *Data Sources*).
- Data updates automatically and displays correctly.
- Users can filter and visualize data dynamically.
- Chatbot retrieves accurate data from the dashboard backend and APIs.
- Chatbot declines out-of-scope or inappropriate queries.
- Error handling displays meaningful feedback to users.
- Security best practices are implemented.
- Client reviews the final demo and approves the product.

Risks

Risk Register

1. API usage limits
    a. Likelihood: High

    b. Impact: High
    c. Mitigation: Implement caching or a forced wait time
2. Chatbot hallucinations
    a. Likelihood: Moderate
    b. Impact: High
    c. Mitigation: Implement guardrails and utilize accurate information in prompting
3. Poor site performance
    a. Likelihood: Moderate
    b. Impact: Moderate
    c. Mitigation: Implement caching or a forced wait time and test this early to have enough time to optimize
4. Unhelpful/Uncorrelated external data
    a. Likelihood: Moderate
    b. Impact: Low
    c. Mitigation: Discontinue use of APIs providing the data
5. Client does not approve of the final product
    a. Likelihood: Low
    b. Impact: High
    c. Mitigation: Include the client throughout the development process and integrate feedback as needed

PROJECT PLAN & CLIENT FEEDBACK

Timeline

January:

- Learn about the project and client
- Ask initial questions

February:

Week of 2/2: Connect with our client, begin developing a project proposal, conduct research

Week of 2/9: Meet with client, proposal presentation, finalize proposal after incorporating feedback

Week of 2/16: Follow-up with client after he reviews the proposal, begin back end development work

Week of 2/23: Continue back end development, begin developing database

March:

Week of 3/2: Begin front end development, implement API connections, test progress so far

Week of 3/9: Complete first iteration of project, deliver prototype to client for feedback *(midterm demo)*

Week of 3/16: Spring break
Week of 3/23: Integrate client feedback, begin work on AI chatbot, begin connecting the front end and the back end

April:

Week of 3/30: Expand front-end to include complete dashboard, begin implementing data filtering
Week of 4/6: Pass all unit tests, Demo to client to receive more feedback, Begin to finalize documentation for a non-technical audience, Select users for beta testing
Week of 4/13: Begin beta testing, optimize performance, integrate client feedback
Week of 4/20: Collect tester feedback, integrate feedback from testers, pass all unit tests

May:

Week of 4/27: Beta testing continues, complete documentation, final edits
Week of 4/4: Final presentation, project completion, product and documentation turned over to client

Response

Client Questions & Our Responses

How are we going to keep track of so much data?

- We will utilize a database and APIs

Is the AI going to help suggest a bet, i.e." you should make this bet because it's likely you'll win."

- We stated no. The main goal of our project is to educate, not direct.

How are we determining an "insider" trade?

- We will utilize trend tracking over time and a variety of data points to point towards possible inside traders. Our product does not offer any guarantees of this, as this is out of scope.

General feedback: try to not have too broad of a focus.

- We will start our project by focusing only on political bets (per client interest) to keep the data scope narrow. As we progress, if we are successful in politics, we will expand to include additional categories.

FINAL THOUGHTS

This project presents a sophisticated platform combining data visualization and AI-powered analysis to support informed betting decisions. With targeted features for diverse user groups and a robust technological foundation, it addresses market inefficiencies and enhances strategic insights, fostering more transparent and effective market participation.