Prior Work

There are 2 platforms we want to focus on that do Social Media Sport Betting, Pikkit and DubClub. DubClub offers a marketplace for selling sports picks while Pikkit offers a social aspect of seeing friends' bets and tracking users betting history. These platforms show us that there is a need / want for this service and clients are willing to pay subscriptions to copy the bets of high performing bettors or just to see interesting bets from their community and friends. The difference between our platform, PixRUs, and these other platforms is we want to combine both these features and at our core we offer transparency. We intend to incorporate advanced metrics to give users all the information needed to choose a subscription that aligns with their interests. DubClub and Pikkit do not offer performance metrics for posters that incorporate the level of risk of a bet in comparison to the payout, they mostly show ROI (return on investment), profit / loss, and then they narrow those metrics down into categories such as ROI on specific sports league or ROI on types of bets. We intend to use the following metrics:

ROI = (Net Profit / Total bet amount) * 100
Win Percentage = (# Winning Bets / Total # Bets) * 100
Risk Adjusted ROI = (Expected Payout / Risk(odds))

Rating = ROI + Win Percentage + Risk Adjusted ROI

After creating the basic application a Machine learning model could be trained to learn which factors contribute more to long term success and dynamically adjust the weights of these factors in the rating. Also could be used to group users into categories of High-risk High-reward bettors or Low-risk bettors that generate consistent gains.

Research Report: Technologies for Building PixRUs

In developing PixRUs, we have selected a tech stack that ensures scalability, flexibility, and ease of use. Our choices include Django for the backend, PostgreSQL for the database, and TailwindCSS with JavaScript for the frontend.

1. Backend Framework: Django (vs. FastAPI)

Django is a Python-based web framework known for its "batteries-included" philosophy, providing a wide array of built-in features like authentication, ORM (Object-Relational Mapping), and an admin panel. These features will allow us to build PixRUs rapidly and maintain scalability.

Advantages:

Rapid development: Django's robust framework includes tools that allow developers to focus on application logic without reinventing the wheel.

Scalability: Django can handle high-traffic applications and has proven scalability in many production systems.

Security: It offers built-in security features

Community and Ecosystem: Django has a large community and a plethora of third-party packages, making it easier to extend the application.

We chose Django over FastAPI because of its comprehensive feature set and ease of integration for the types of tasks we need for PixRUs. While FastAPI excels in API-first projects, Django's maturity and extensive libraries align with our focus on rapid development and long-term maintainability.

2. Database: PostgreSQL

PostgreSQL, a highly stable and robust relational database, is our choice for managing PIX's data, particularly for handling the structured data involved in storing bets, user data, and transaction history.

Advantages:

ACID Compliance: PostgreSQL ensures that all transactions are processed reliably, maintaining the integrity of the betting data.

Data Integrity: Relational databases like PostgreSQL are ideal for applications with structured data, ensuring referential integrity and complex relationships between tables.

Advanced Features: PostgreSQL supports JSON storage, full-text search, and other modern features that offer flexibility.

Scalability: It can handle complex queries and high volumes of data efficiently, which is crucial as our user base grows.

3. Frontend: TailwindCSS with JavaScript

For the frontend, we are using TailwindCSS in combination with basic JavaScript.TailwindCSS is a utility-first CSS framework that allows us to build custom designs rapidly without writing custom CSS.

Advantages:

Simplicity and Speed: TailwindCSS is fast to set up and eliminates the need to write custom CSS, speeding up the development process.

Customizable and Lightweight: The framework allows us to create a unique design while keeping the frontend lightweight and performant.

Control over UI: Unlike component-based frameworks, TailwindCSS offers more control over styling, making it easier to create pixel-perfect designs.

We chose TailwindCSS and JavaScript because our focus is on simplicity, customizability, and performance. React, Vue, or Angular would introduce unnecessary complexity for the initial version of the app, which requires a minimal yet efficient UI.