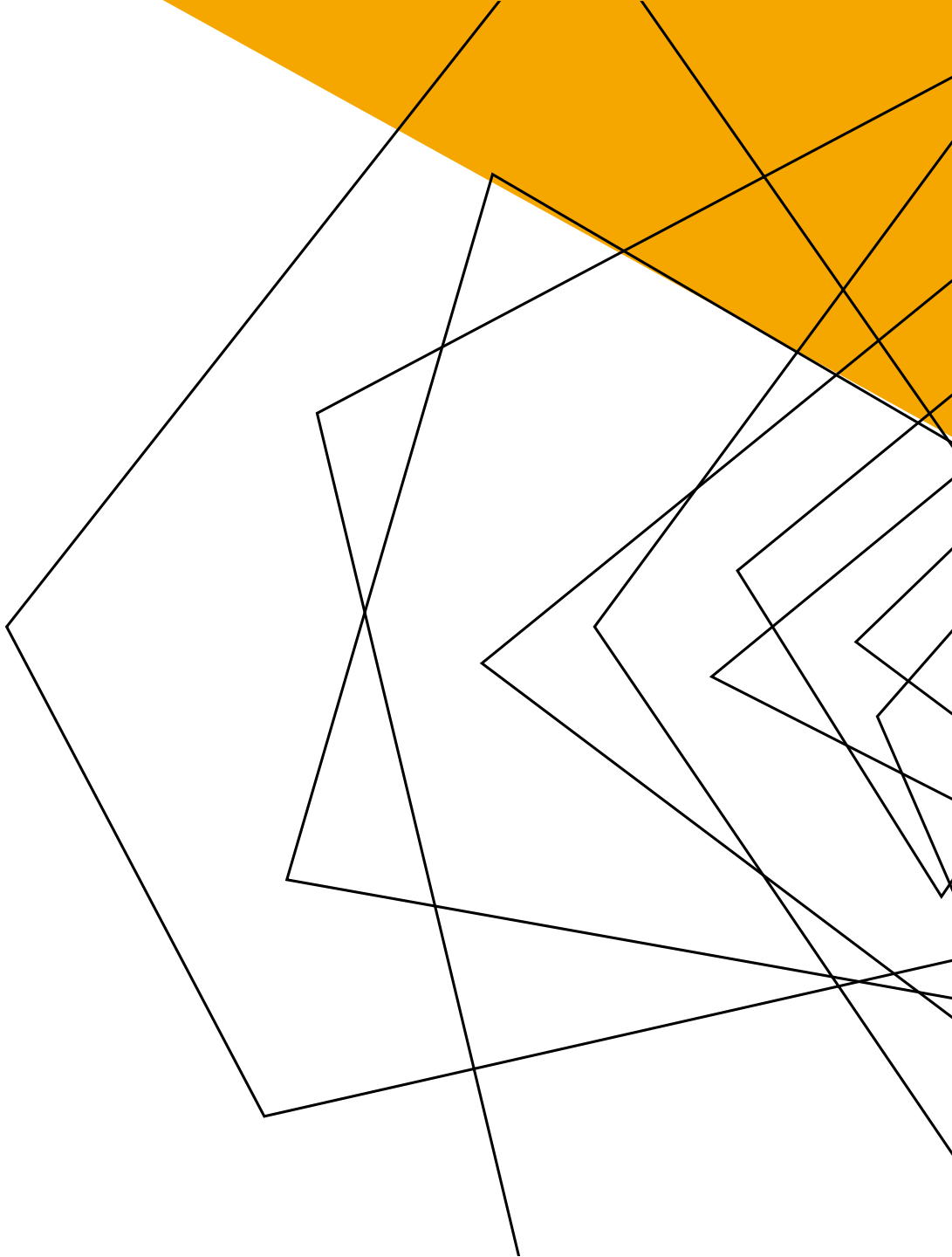
Buff Bets

CSCI 3308

BY: NOAH OSTEROOS, DAVID DE HIGES, DANIEL BAKULA, ABDULRAHMAN ALMUTAIRI, SAMUEL WENGERT, ANDREW ROBERTS

Description of Buff Bets

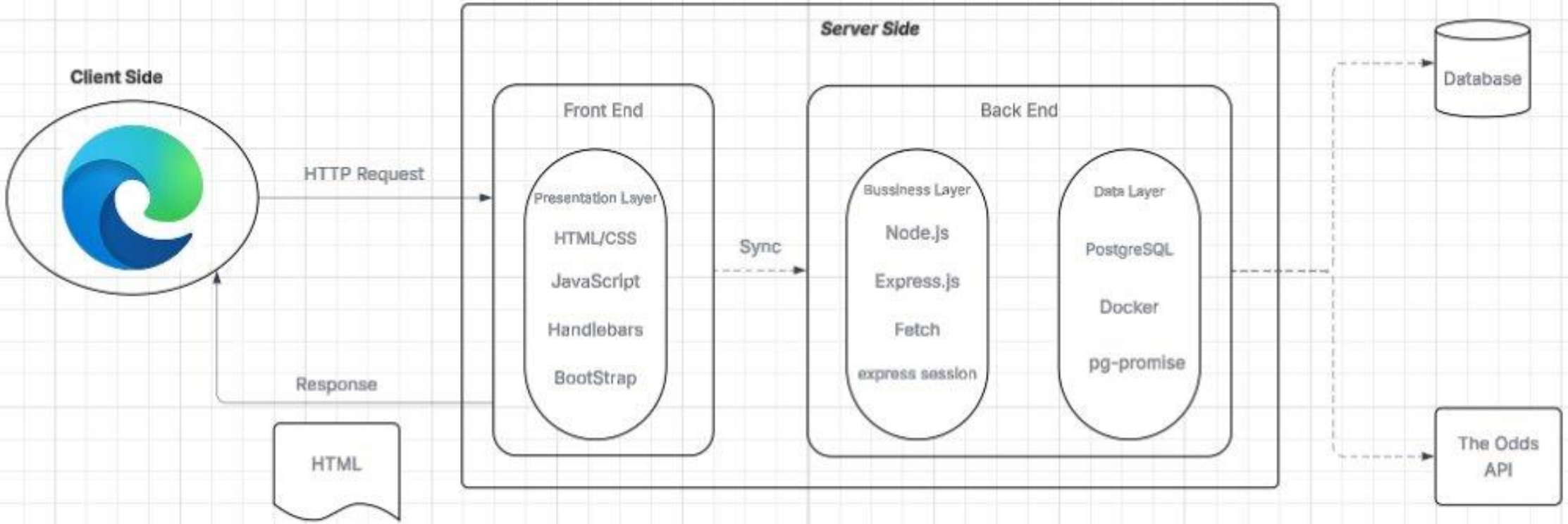
Buff Bets is a CU Boulder-themed sports betting platform where users can place virtual bets on everything from CU athletics to major pro sports like the NFL, NBA, and MLB. Inspired by DraftKings but designed for students, Buff Bets offers a risk-free, competitive environment where you can connect with friends, compare bets, and show off your sports knowledge. With real-time odds, a sleek interface, and social features that make it easy to challenge your friends, Buff Bets brings the excitement of sports betting to campus without the risk, but with all the pride! SKO BUFFS!



Tools Used

Tool Name	Purpose	Rating	Methodology
GitHub	Version control system to manage source code. Project repository	5/5	Branching, Pull Requests, Peer Code Reviews
GitHub Projects	Project tracking, issue planning and progress updates	5/5	Agile, Iterative development
PostgreSQL	Relational database to store users, bets, friendships, and event data	5/5	Normalization, SQL
Visual Studio	Integrated environment for editing, testing, and debugging both frontend and backend code	5/5	Pair Programming, Debugging, Peer review
HTML, CSS, JavaScript, Bootstrap	Used to create and render UI with Handlebars and responsive layouts	5/5	UI design, formatting, templating
Node.js	JavaScript runtime for handling backend logic and APIs endpoints	5/5	Event-Driven programming
Render	Hosting the live backend and frontend of the application	4/5	Continuous deployment
The Odds API	Provides real-time sports event odds to display available bets	4/5	API Integration
Mocha/Chai	Testing backend logic, endpoints, and unit tests	3/5	Use case and unit testing
Docker/Docker Compose	Containerization for development and multi-container setup for the app and database	3/5	DevOps, Environment testing
pg-promise	JavaScript interface library for PostgreSQL queries in Node	5/5	Database layer
bcrypt	Password hashing for secure user authentication	5/5	Authentication
express-session	Manages user login sessions through cookies at the backend	5/5	Session Management
Axios	Used for API calls	5/5	API communication
Express.js	Web application framework for routing, middleware, and APIs in Node.js	5/5	Middleware

ARCHITECTURE DIAGRAM



CHALLENGES FACED

API

- For obtaining the scores, you can only access scores by sports, and not events, which led to a restructure of the API endpoint. Solving this issue led to an API call for each sport rather than each event.

Friends

- With a solid foundation of a database, the majority of the friend's implementation was pretty straightforward, however it took a considerable amount of time workshopping to find just the right queries to find a friend of user, then find their id, then find their bets and finally tally those bets. Integrating between three tables of information was no easy task but through determination and long hours we managed to get the information exactly how we needed it. Of course, there were more hiccups along the way including syntax errors that wouldn't raise an error which became hard to find and difficulties with the displaying of the information, but all was resolved and now we have a working friend database and page.

Database Design

- Initially, our database schema was overly simplistic and lacked proper normalization. Too much data was stored in a single table, resulting in redundancy and inefficient queries. To resolve this, we redesigned the schema with a more structured and normalized approach, reducing duplication and minimizing join-related issues.

Docker

- Setting up Docker for local development was initially difficult. The Node.js server wasn't connecting to the PostgreSQL container. Environment variables weren't loading correctly, causing connection failures. We resolved the issue by properly configuring our docker-compose.yaml file to ensure both services shared the same network and confirmed the DB_HOST was set to the database container's name. Running docker-compose down -v helped reset volumes when the containers became desynchronized, and docker-compose up --build ensured that rebuilds consistently applied updates.

FUTURE SCOPE/ENHANCEMENTS

Real time Odds/ Scores:

- Integrate WebSocket support to display live scores and have live odds as game progresses
- This will make the betting experience more dynamic and captivating for our users

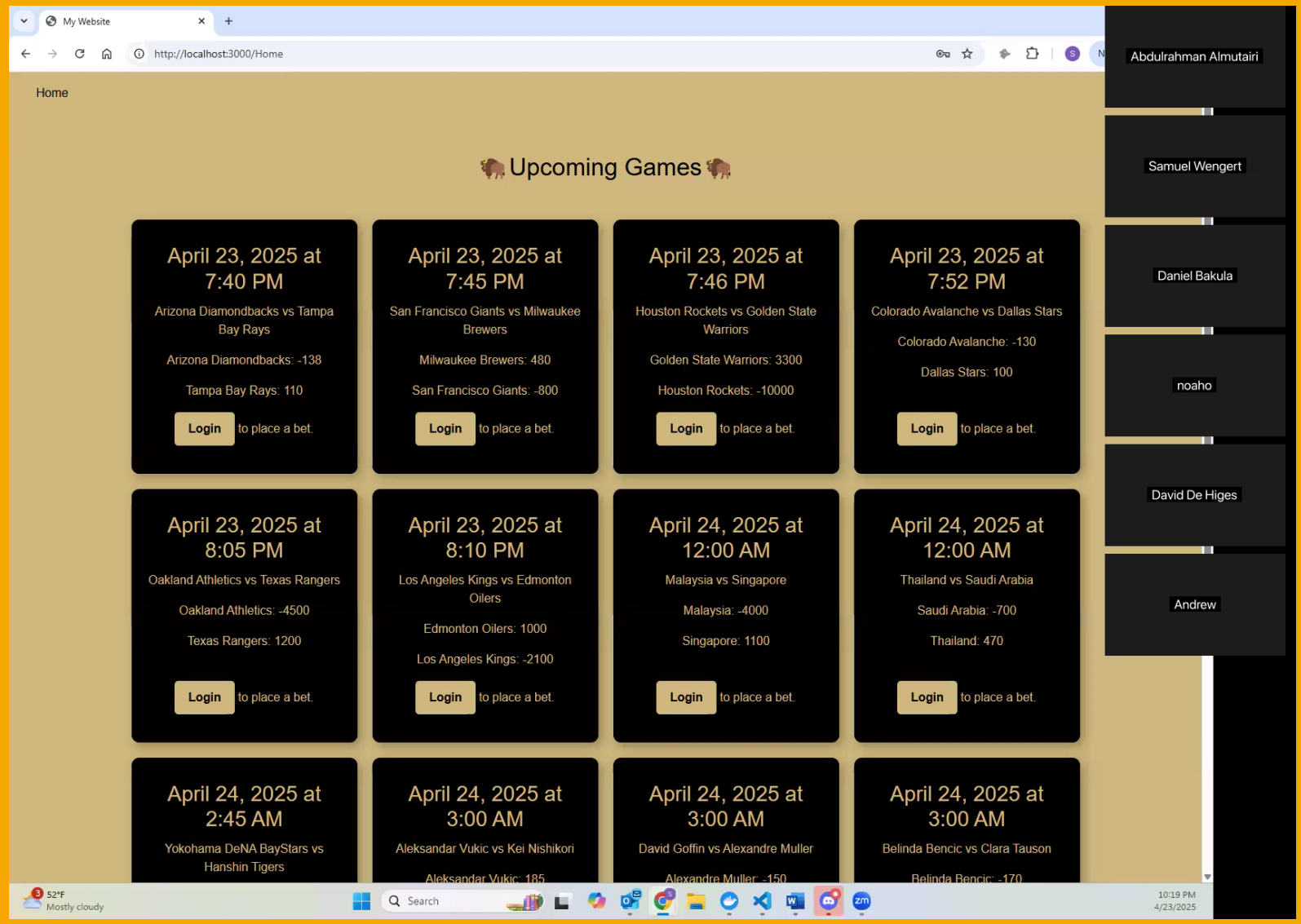
Leaderboard and enhanced social features:

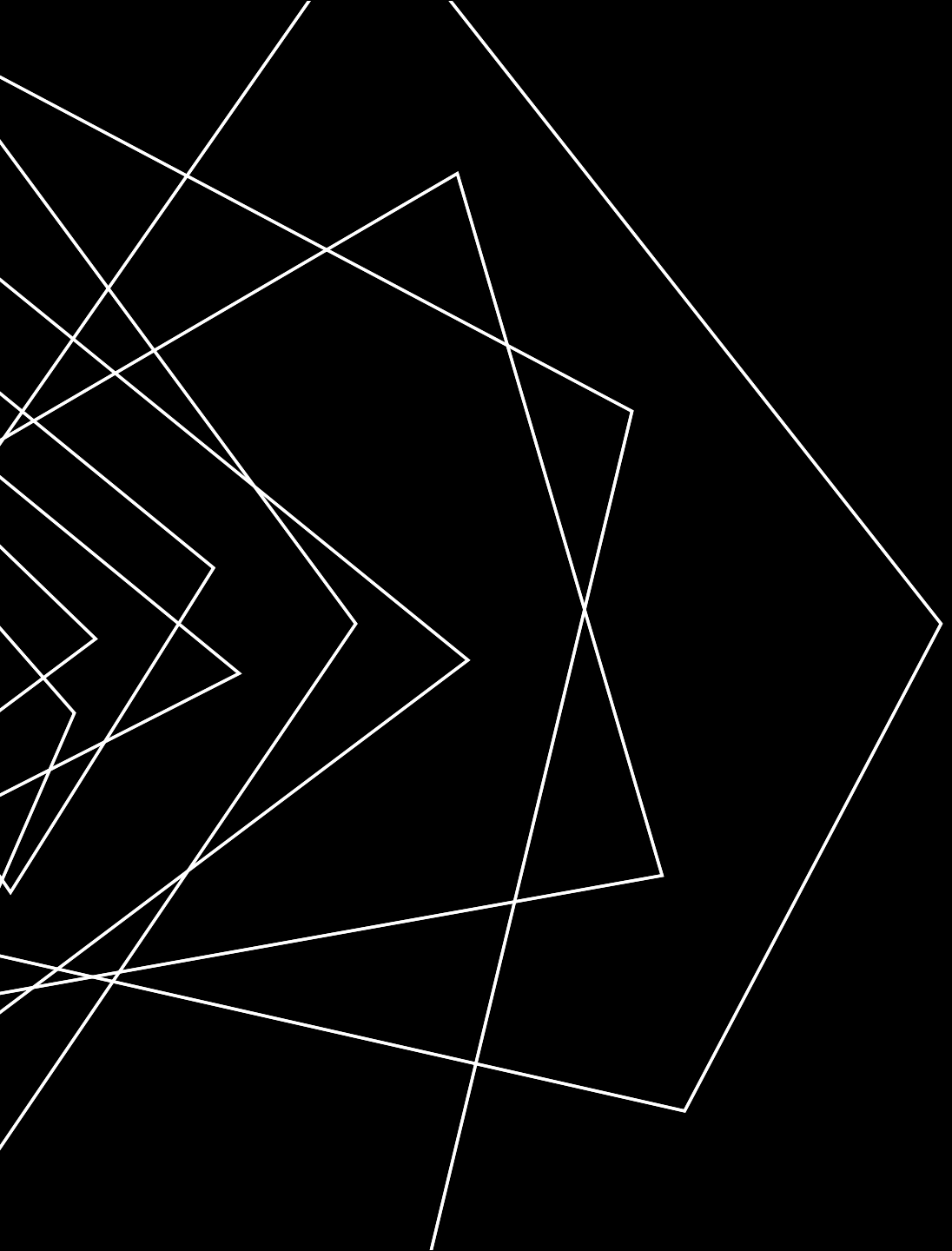
- Add leaderboards that ranks users based on a variety of options (Bets won, Winnings) Additionally can add visual cues for ranks
- Boosts user captivation, promotes friendly competition and promotes long-term usage

Mobile App:

- Make the application accessible from mobile devices. Allow users to place bets, view friends' activity, and track their virtual earnings from anywhere.
- Will make application more accessible to more users, as well as attract more users

Project demo





Thank you!

ANY QUESTIONS?