

# RANKED ROYALE: COMPETITIVE MATCHMAKING & LEAGUE PROGRESSION

Group 510: Dan Nguyen, Jacob Shechter, Mika Tums, Arshia Verma

Northeastern University

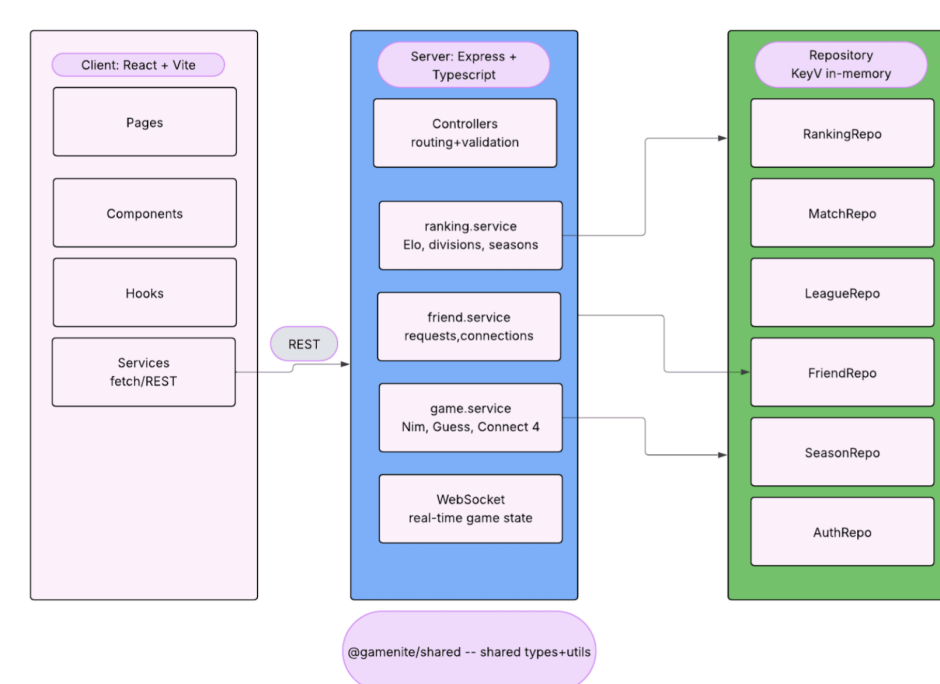


## Introduction/Abstract

GameNite is a multiplayer gaming platform that supports casual play across several skill-based games. While the platform provided a solid foundation for social gaming, it lacked any structured competitive system: wins and losses had no lasting meaning, and players had no incentive to return regularly. This project introduces **Ranked Royale**, a competitive ecosystem built on top of GameNite featuring performance-based *Elo matchmaking*, *seasonal league divisions*, *a new game (Connect 4)*, and *a friends system with social play features*. The goal was to transform GameNite from a casual platform into one where competitive players can track progression, climb divisions, and challenge friends, while preserving the existing casual experience for players who prefer it.

## System Design/ Methodology

The system was built as a three-layer architecture: a React + Vite client communicating via REST API to an Express server, which delegates to an in-memory KeyV repository layer. The ranking system uses the Elo formula with a flat K-factor of 32, a well-established standard for players that provides meaningful rating movement per match. Players are assigned to one of five divisions: Bronze, Silver, Gold, Platinum, and Diamond based on their Elo rating, with separate ranking records maintained per game per season to prevent cross-game contamination.



To the left here is the architecture diagram. We have the Client -> Server -> Repository.

Matchmaking enforces division-based pairings so players only compete against opponents of similar skill, with an exception for friend challenges which bypass this restriction to preserve social flexibility. The friends system uses WebSocket push notifications rather than REST polling, eliminating the server overload caused by repeated HTTP requests.

**Division-Based Matchmaking:** Players can only join open ranked matches created by players in their own division. This ensures competitive fairness, for example: a Bronze player cannot be matched against a Diamond player. This also mirrors standard competitive design. Friend challenges, however, bypass this restriction intentionally. For example: if Yao is friends with Knight of Games (KOG), and Yao is in the Gold division, and KOG is in Silver, KOG can't load a match that Yao made in the Ranked Match section. However, if Yao deliberately sends a ranked match game to KOG through Friends, KOG can join and play with Yao. *This feature preserves social play flexibility.*

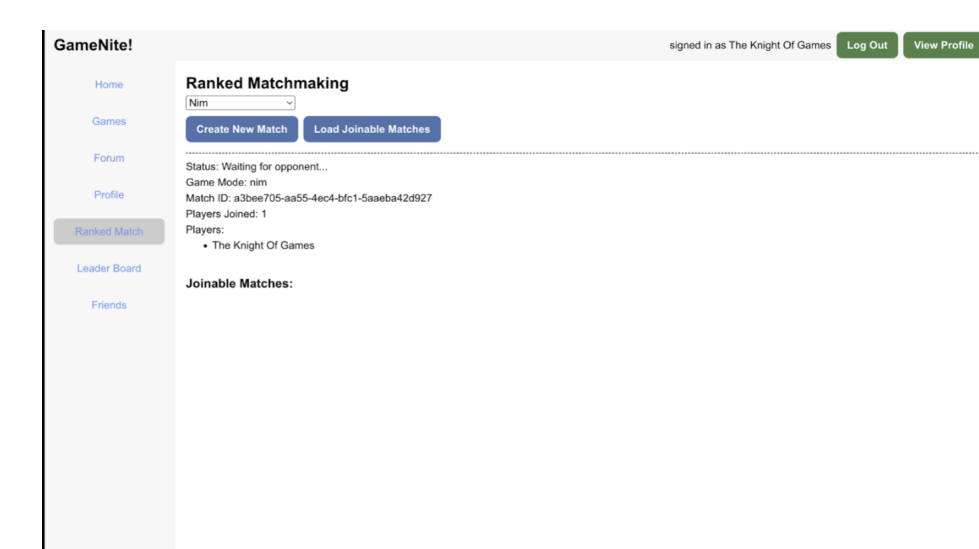


Fig. 1: Creating and joining a Ranked Match.



## Collaboration

The team followed an agile sprint-based workflow across four sprints, with two recurring weekly meetings: one internal team sync and one with the course TA to review progress and surface blockers early. All development was managed through GitHub Projects using a Kanban board with issues assigned per sprint, giving the team shared visibility into what was in progress, blocked, or complete at any time.

All code was developed on feature branches and merged via pull requests, with a required review from at least one teammate before merging into main. Code reviews were used not just for catching bugs but for knowledge sharing. Reviewers left inline comments explaining design concerns, suggesting refactors, and flagging edge cases. This kept the entire team aware of changes happening outside their own tasks and prevented integration surprises at the end of each sprint.

## Features & Implementation

Three user stories were fully implemented. For ranked matchmaking and league progression, players can create or join ranked matches, track their Elo rating per game, view division standings, and see their full match history with win/loss indicators and Elo changes. For new competitive games, Connect 4 was implemented with full game logic, real-time WebSocket state updates, and both ranked and casual play modes. For the friends system, players can send and accept friend requests, view a friend's competitive profile including ranked stats and match history, and directly challenge friends to ranked matches regardless of division. All features are live at the deployed URL.

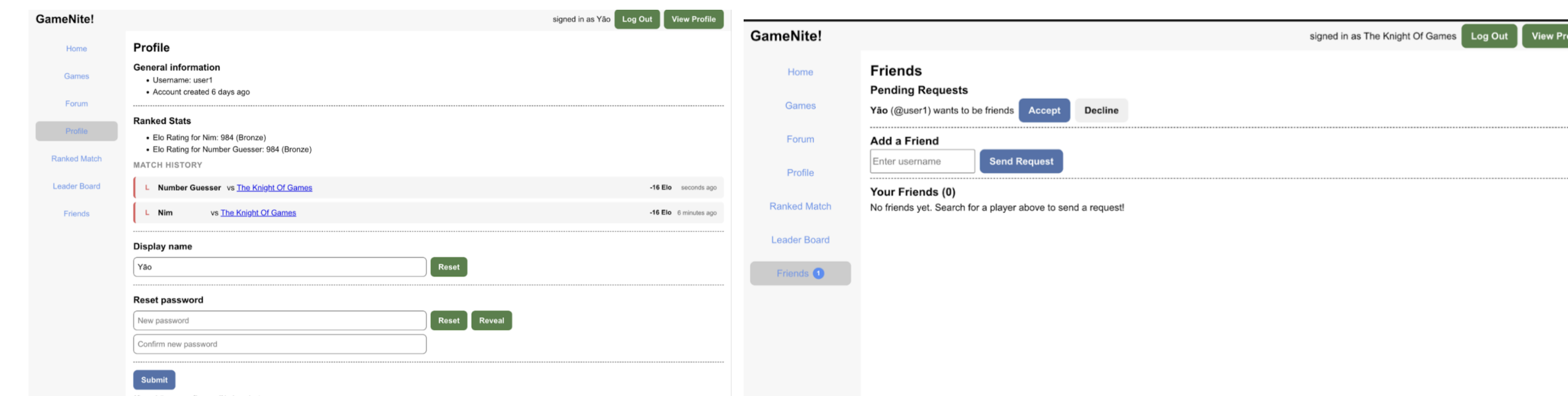


Fig. 3: Profile page showing Elo ratings and match history, ranked matchmaking page, friend profile page showing ranked stats

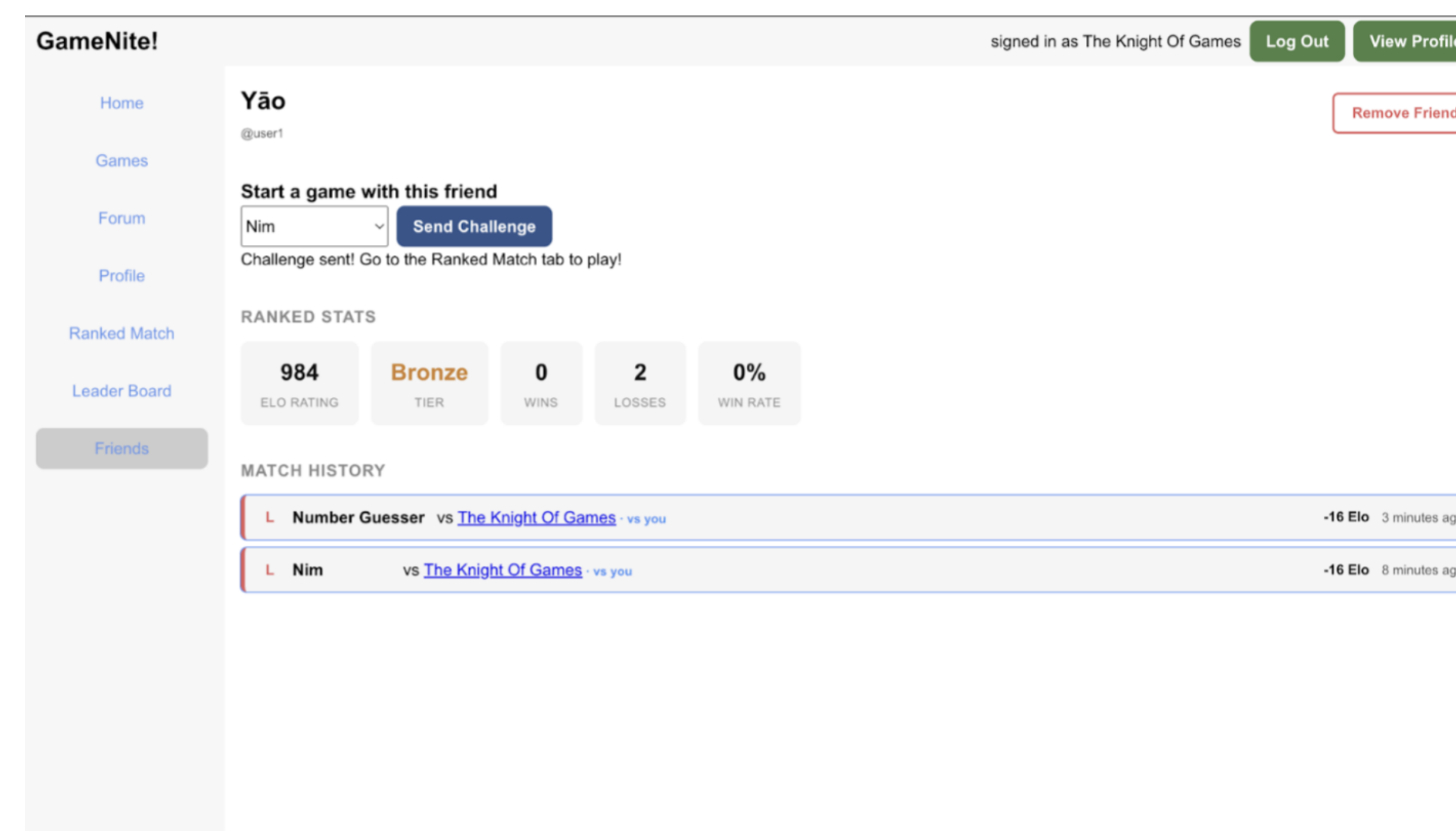


Fig. 4: View Friend Match History + Send Challenge

### LINKS TO GITHUB REPO AND HOSTED APP

Hosted Render app: [Here](#)

GitHub repo: [Here](#)

## Discussion/Future Work/Web App

The project successfully delivered a competitive layer on top of GameNite with all essential conditions of satisfaction met across three user stories. Several desirable features were descoped due to time constraints, including automatic season rotation, end-of-season division movement, a friends-only leaderboard, and side-by-side stat comparisons. A notable mid-sprint deviation was the refactor of the friends polling system to WebSockets after identifying that repeated REST polling was causing 502 errors on the free-tier Render deployment: a real-world constraint that shaped the final architecture. Future work would focus on activating the existing season infrastructure with a scheduled rotation job, adding rank badges to player profiles, and implementing a bracket-style tournament mode that was originally planned but replaced by the friends system.

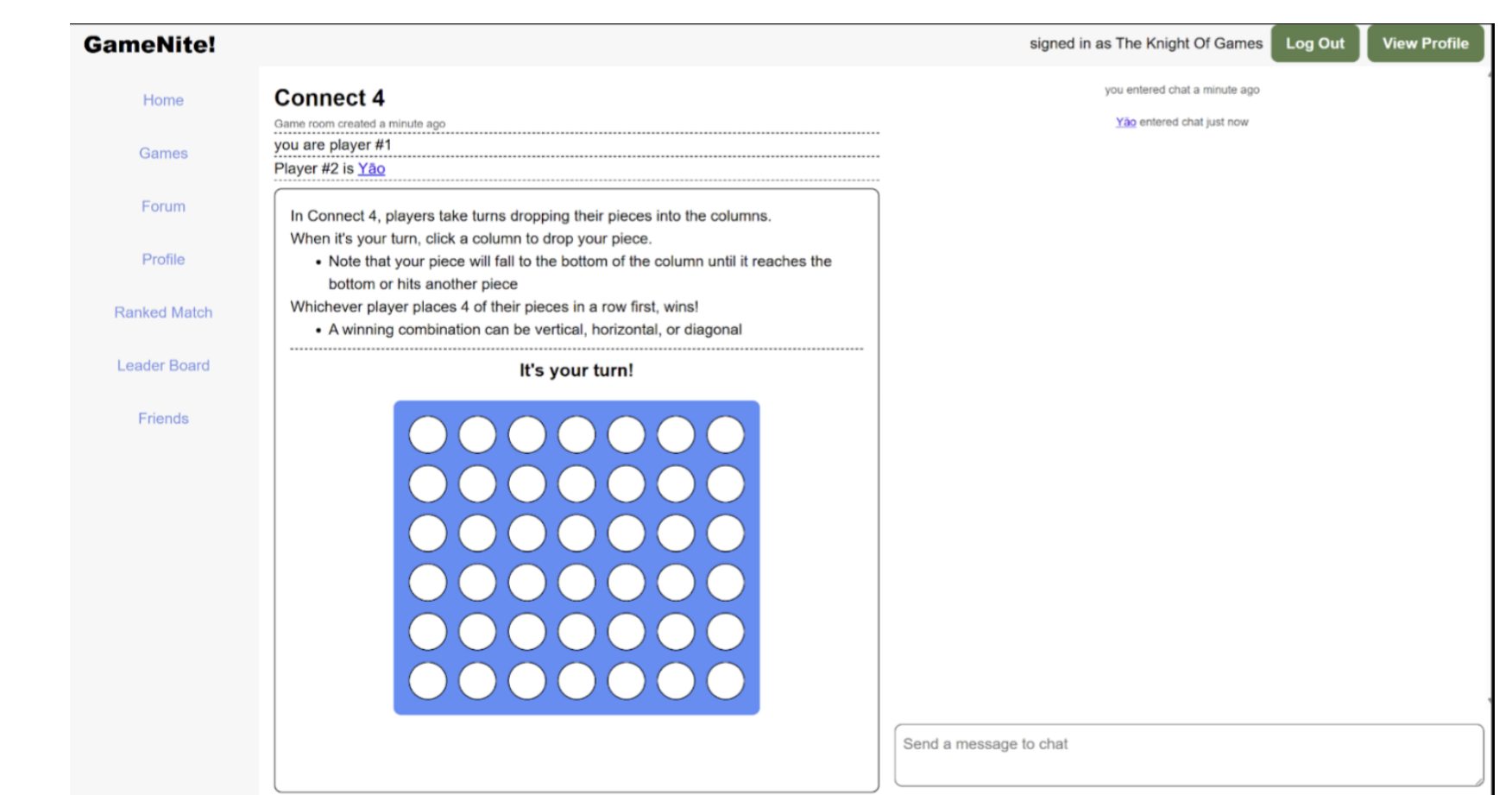
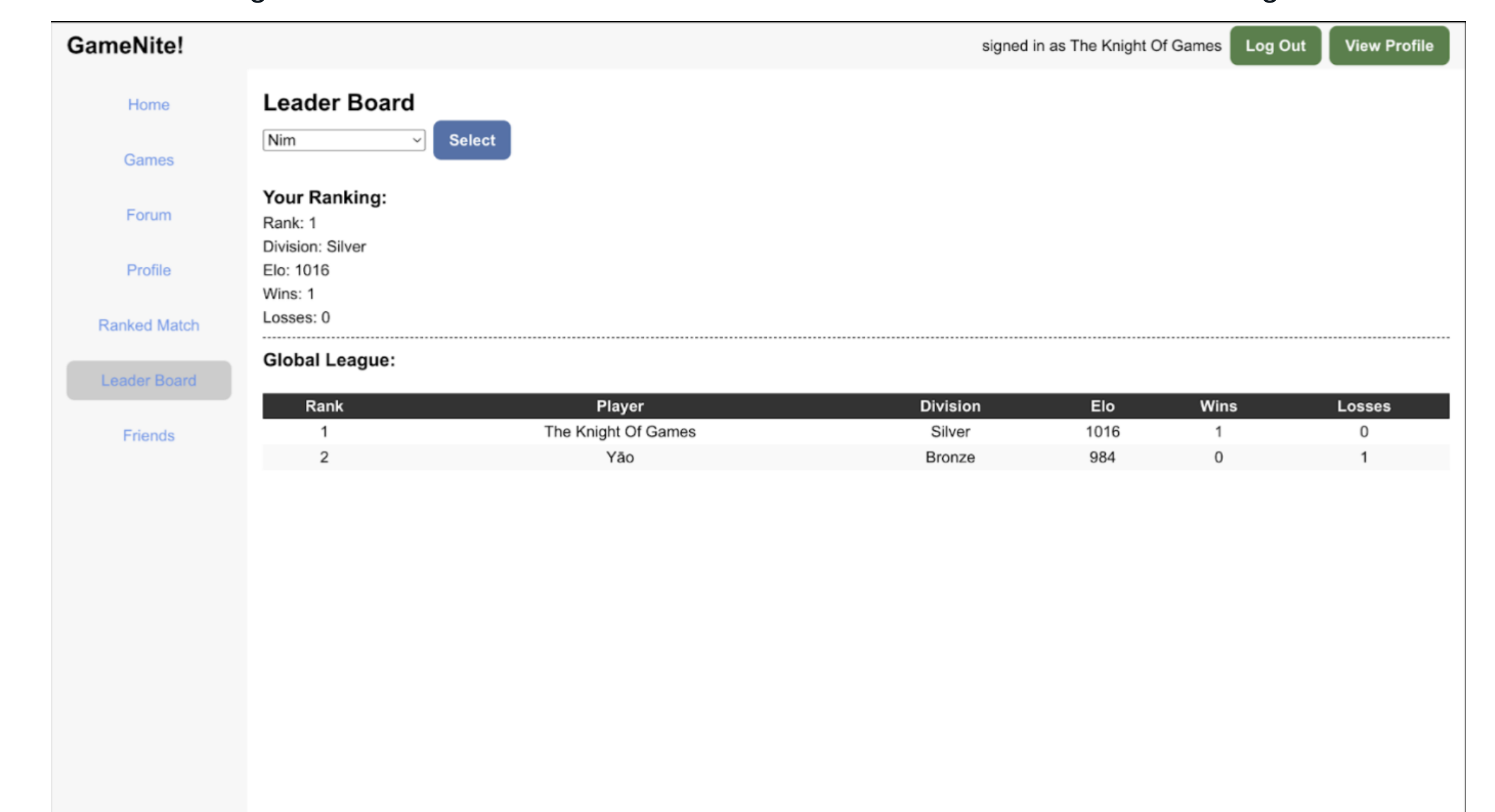


Fig. 5: Connect 4 Frontend & Leaderboard to Select and View Standings



## User Stories

### User Story 1: Ranked Mode/Matchmaking + League Progression

• As a competitive player, I want to play ranked matches that affect my statistics and rating, so that I can measure my skill progression and compare myself to other players.

### User Story 2: New Competitive Games

• As a competitive player, I want access to a variety of skill-based games to compete in, namely Connect Four, so that ranked play remains diverse and engaging

### User Story 3: Friends System

• As a social and competitive player, I want to add friends and view their activity, so that I can keep up with their competitive progress, compare rankings, and play games against them.