

KIMANITE!

CS4530 GROUP 111: ETHAN, DHRUV, DIO, NEIRIT

PROFILE CUSTOMIZE

Users can upload avatars, write bios, add personal links, connect Google/GitHub/Discord accounts through SSO, set profile visibility to public/friends/private, and rearrange profile sections.

LEADERBOARD

Global leaderboard ranked by wins with filtering by game type and time range, personal match history with filters, stat tracking on profiles, and 8 badges awarded automatically for milestones like win streaks and total games played.

SOCIAL FEATURES

Friend request system, direct messaging with a chat, user search by username or display name, online/idle/DND/invisible status indicators, user blocking that prevents DMs and friend requests, and DM privacy settings tied to profile visibility.

CATAN

Full implementation of Settlers of Catan with lobby creation and joining, turn-based gameplay with dice rolling, resource collection, building, trading, dev cards, robber mechanics, and real-time board state sync so all players see the same game state.

Problem Statement

The original GameNite website had almost no user personalization or community features. Users could only change their display name which made profiles feel plain and generic. There was no way to track improvement over time or compare yourself to other players, no friend system or direct messaging, and no way to search for people you've played with before. The gaming experience felt isolated. We also wanted to add a multiplayer strategy game so users could play something more complex together. Our goal was to turn GameNite into a personalized and social gaming platform for both competitive and casual players.

Future Work

The two extension features we didn't get to were performance graphs showing improvement over time and friends-only leaderboards. Adding charts to the profile page would let competitive players visualize their trends across different game types. We'd also want to improve real-time stability around edge cases like disconnects and duplicate events in the leaderboard and messaging systems. On the Catan side, handling mid-game disconnects gracefully (forfeit, skip turns, or pause) and supporting private invite-only lobbies would make the experience a lot better. A notification system for friend requests, game invites, and badge awards would also tie everything together.

Technology Stack & Design

We built on the existing GameNite codebase using TypeScript across the full stack. The frontend uses React with Chakra UI and the backend runs on Express with Socket.io for real-time communication. All data is stored through Keyv which provides an in-memory key-value store with a generic repository abstraction layer. Avatar images are uploaded to Cloudinary. SSO authentication uses Passport.js with Google, GitHub, and Discord OAuth strategies. The leaderboard uses a dual-storage approach with cached all-time stats for fast lookups and raw match history records for time-filtered queries, with Socket.io rooms for live push updates. Catan game state lives entirely on the server and syncs to all connected clients through Socket.io events. We used Vitest for unit testing, Playwright for end-to-end tests, and the app is deployed on Render.

