CS4530 Final Project: Code Commons

Group 603: Kyle Sung, Karishma Bhargava, Junyan Dai, Zhicheng Liu

Our Features

🔍 **Smarter Profiles & Activity Tracking**
To boost engagement and recognize contributions, we're enhancing user profiles with detailed activity tracking and a dynamic badge system. Users can view their questions, answers, and comments in a tabbed layout, favorite important posts, and follow others to stay informed. A real-time badge system will celebrate achievements, from tag-specific contributions to identity-based milestones, with rarity levels and progress tracking.

🎮 **Gamified Participation**
We're introducing a gamified point system to make participation more rewarding. Users can earn points by contributing to the platform. These points can be used in competitive games and challenges, with leaderboards and fair matchmaking to encourage friendly rivalry. Personalized goals and point decay will keep users active and motivated.

💬 **Advanced Developer Chat**
To support real-time collaboration, we're upgrading the chat system with features tailored for developers. Users can share and run code snippets, edit or delete messages, and react with emojis, New tools like typing indicators and file uploading will make chat more dynamic, efficient, and suited for fast-paced problem-solving.

Technology Stack & Design

We enhanced the user profile system by extending the user schema to include followers, follow requests, privacy settings, points, and badge collections. Real-time updates for follower interactions and badge achievements were implemented using Socket.IO, ensuring that users are instantly notified of activity. The profile UI includes a tabbed React interface and uses backend endpoints to fetch and display user-specific questions, answers, and comments.

For the gaming system, we introduced a WagerConfig object to manage game type, wagered points, and player data. Points were integrated into the user schema and are updated through hooks in key contribution functions such as posting questions, comments, or answers. Matchmaking logic supports skill-based challenges, while leaderboards and team modes are dynamically updated via server-side logic.

To upgrade chat functionality, we integrated an AWS S3 bucket for file uploads, supporting both local and production environments. We added code execution support using the Judge0 API, enabling users to run code snippets directly within chat. Reactions, message edits, and typing indicators are all synced in real time through Socket.IO, creating a seamless and interactive messaging experience.

Our continuous integration pipeline runs a comprehensive test suite across frontend and backend components, and automatically deploys the application using Render.

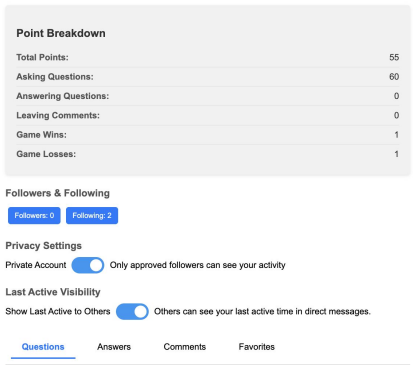
Future Work

There are several exciting opportunities to expand and refine our system. For user profiles, future updates could include achievement streak tracking - such as daily contributions or consecutive answer streaks - with visible multipliers to reward consistent activity. Expanding the badge system to allow users to compare their progress with friends or users they follow would further enhance community interaction and motivation.

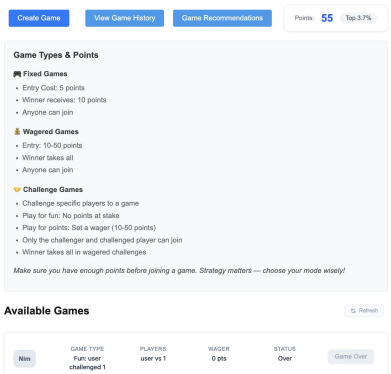
On the gaming side, we currently support only a single game type (Nim). Future iterations could introduce a wider variety of games, along with features like tournament hosting or time-limited special events to encourage engagement. These enhancements would give users more opportunities to participate, strategize, and compete within the community.

For chat, additional developer-friendly features could include expanded Markdown support for rich text formatting (e.g., bold, italics, inline code, and blockquotes). Showing online status for users would also make real-time collaboration more efficient by helping users identify who is available for help or discussion at any given time.

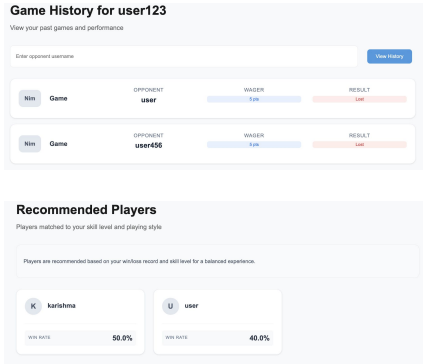
Demo Site: https://cs4530-s25-603.onrender.com/
Source Code: https://github.com/neu-cs4530/spring25-project-group-603



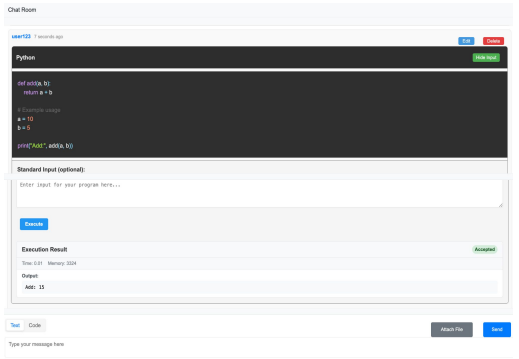
User profiles show a users' point breakdown, their followers and following, privacy settings, and contribution activity. If a user sets their profile to private, their activity will only be visible to approved followers.



The games page shows the different game options a user can play. Creating a wagered game allows a user to pick their wager amount, and users can either challenge specific users to games for fun or games for a wager.



Users can see their game history, including the type of game, who their opponent was, the wager, and if they won or lost. Users can also see which players they are recommended to play with according to their win rate.



In global chat and direct message, users can send text as well as upload files and code snippets. One code snippets are sent, they can be executed within chat to see the output (multiple languages are supported). Users have the ability to edit or delete their messages as well.