

# CS4530 FINAL PROJECT: "COLLABRIX"

## OUR FEATURE

Collabrix enhances the traditional Q&A forum experience by adding a proper feedback ecosystem, gamified learning tools, and modern UI designs while maintaining user safety. Users can like or dislike questions and answers, view engagement levels of other users, and gauge the reliability of fellow users by looking at our real time leaderboards, which is sorted by a built in scoring system based on activity and upvotes. To differentiate our site from other code based forums, we introduced a dedicated Games section featuring 2 coding related activities such as Code Golf and Bug Hunt as well as a discussion activity in the form of Daily Trivia. These games encourage problem solving, critical thinking and community collaboration, with score tracking and leaderboards to encourage competition. Additionally we implemented safety features such as a report button on questions and answers, allowing the community to flag inappropriate or harmful content, which will be checked on by admins to ensure a safe and respectful environment.

## TECH STACK

Our project was built using a modern full-stack JavaScript ecosystem. The backend uses Node.js, Express, MongoDB, and Mongoose, providing a clean API layer and persistent data storage for questions, answers, voting behavior, badges, and leaderboard scores. We also use synchronizied across all clients. On the frontend, we use React with functional components and custom hooks to manage state, API calls, and live socket events. Shared TypeScript types are stored in a separate/shared directory, ensuring type-safety between frontend and backend. The project is structured around a REST-plus-WebSocket hybrid design, automated with modular services, controllers, and hooks, to keep the code maintainable as features scale.

## DESIGN

The system's overall design prioritizes usability, modularity, and extensibility. The UI follows a clean, accessible layout with reusable components for buttons, cards, tables and input elements, enabling uniform behaviors across all pages. Each feature such as question, answer, voting, badges and multiple leaderboards is implemented as a self-contained module with its own API endpoints, React pages, and CSS styling. Real time interactions (voting, leaderboard update, badges assignment, and games) are handled through event-based patterns, while the server enforces validation, edge cases and ranking logic. The design intentionally mirrors real-worlds Q&A platform like Stack Overflow but scaled down into a mobile friendly, classroom appropriate architecture that is easy to understand and expand.

## OUR FUTURE WORK

For future development, we plan to expand the platform's gamification and introduce personalization features by introducing a point based reward system tied to the leaderboard performance. Users who consistently rank highly would earn points that could be used to spend on cosmetic upgrades such as customizable avatars, profile frames etc. This system would encourage engagement and long term user retention.

## DEMO & SOURCE

Our Demo Site: <https://fall25-project-fall25-project-group-205.onrender.com/>

Our Codebase: <https://github.com/neu-cs4530/fall25-project-fall25-project-group-205>

## SCREENSHOT SHOWCASING ANSWER PAGE

The screenshot shows the 'Collabrix' interface with a search bar, 'Log out', and 'View Profile' buttons. A sidebar on the left contains navigation links: Questions, Tags, Messaging, Users, Games, Leaderboard, Communities, and My Collections. The main content area displays a question: "What specific changes do I need to make to prevent memory leaks?". Below the question is an 'Add Comment' button. The answer section is titled "Users found this answer to be the most helpful:" and features a user profile for 'reactinia' (Apr 17 at 10:38:56) with an 'Add Comment' button. The answer text reads: "The issue with your code is a classic case of a memory leak in your React application. You're creating event listeners or subscriptions but not cleaning them up when components unmount. Here's how to properly handle this:" followed by a code block. The code block contains a React component using useState and useEffect to manage a data source and handle window/document resize events. The code includes comments explaining the cleanup function and the use of window.removeEventListener. Below the code, there is a note: "Always remember to return a cleanup function from useEffect when you're setting up subscriptions, timers, or event listeners." At the bottom of the answer, there is an 'Answer Question' button and a section for 'Other Answers:'.

GROUP 205: CHARLES PLAMER, VIET HUYNH, ARJUN VELLIMEDU, ZAY YAR MIN (SAM)

## SCREENSHOT SHOWCASING GAMES PAGE

The screenshot shows the 'Collabrix' interface with a search bar, 'Log out', and 'View Profile' buttons. A sidebar on the left contains navigation links: Questions, Tags, Messaging, Users, Games, Leaderboard, Communities, and My Collections. The main content area is titled "Games" and features the text "Challenge yourself with our interactive programming games!". Below this text is a link "View Game Leaderboards ->". There are three game cards: "Code Golf" (Solve a problem with the least amount of characters possible.), "Bug Hunt" (Look at a code excerpt to find the bug.), and "Daily Trivia" (Answer a theoretical programming question of varying difficulty.). Each card has a "Play Now ->" button.

## SCREENSHOT SHOWCASING LEADERBOARD

The screenshot shows the 'Collabrix' interface with a search bar, 'Log out', and 'View Profile' buttons. A sidebar on the left contains navigation links: Questions, Tags, Messaging, Users, Games, Leaderboard, Communities, and My Collections. The main content area is titled "Main Leaderboard (Top 20 Active Users)" and features two tabs: "Main Leaderboard" (selected) and "Game Leaderboards". Below the tabs are four buttons: "Daily", "Weekly", "All-time" (selected), and "Refresh". The leaderboard table has columns for "#", "User", and "Points".

#	User	Points
1	reactexpert	9
2	newbiereactdev	6
3	datascientist	5
4	nodejsdev	5
5	dbmaster	4
6	dbnovice	4
7	devopsengineer	4
8	dockerbeginner	4