# Threadscape

Group 109 | Sama Zaki, Nitsa Kanungo, Annabelle Sanok, Kyle Gold

Demo Site | **Source Repo**

Added Features

Threadscape is a discussion board platform enhanced with features designed to boost engagement, personalization, and social interactions. The initial platform lets users signup/login, post tagged questions, answers, and comments about different technical issues, direct or global message others, play Nim against other users, and customize their profile with a biography.

Our expansion was based on three core user stories: an achievement-based motivation system, profile and user experience customization, and personalized recommendations. Our platform now allows users to earn badges and coins, see their ranking among other users on a leaderboard, create collections of questions saved to their profile, change their profile photo, receive notifications, follow/unfollow other users, see a personalized feed of questions based on following activity, and see recommended Youtube videos.
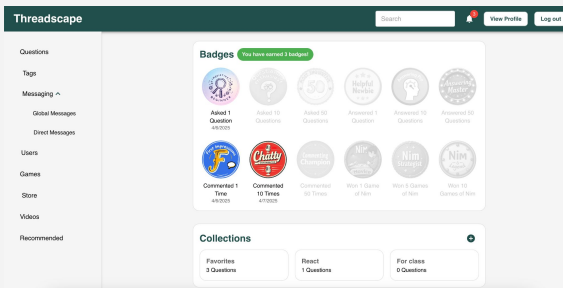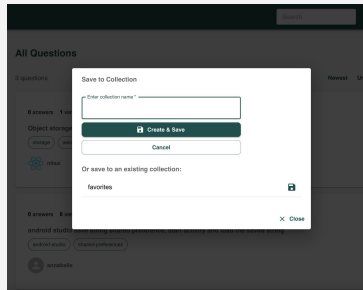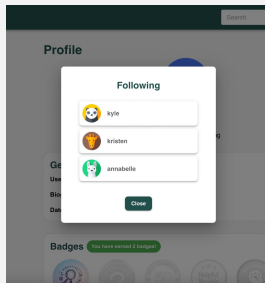
Tech Stack and Design Decisions

We extended the existing codebase to support our new features by defining Mongoose schemas and models for new objects, then added service files to hold backend logic, and controller files to handle API routing. We also integrated the Youtube API to get recommended videos, and Multer to let users upload and store custom profile photos.

On the frontend, we used React and CSS along with Material UI (MUI) to design our interface. We created custom React hooks to manage state and handle calls to our backend controllers, and used reusable React components to render data. Socket.io enabled live updates and communication between the client and server.

Our CI/CD pipeline uses GitHub Actions for continuous integration (linting & testing) and Render.com for continuous deployment of the frontend and backend. MongoDB Atlas holds our production database.

Future Work

For the future, we've proposed further enhancements to our badging system – where users can see their progress towards earning certain badges, and choose to toggle the visibility of badges on their profile. We can also extend our notification logic, to allow users to enable/disable specific types of notifications in their profile settings, and render notifications as a popup on the website as they are received.

Our personalized feed can also be enhanced with a more complex recommendation algorithm. For the sake of this timeline, our recommended feed selects posts based on what a user's followers have posted or upvoted, but this can be extended to show posts based on keywords or tags a user has interacted with. We could go beyond this and use user-similarity (matrix or k-NN algorithm) to recommend posts that similar users have liked, even if they don't follow them.