

CS4530 Final Project: "Stack UnderFlow"

Group 103: Benjamin, Daniel, Nathaniel, Shubh

Our Features

We significantly expanded the existing Stack Underflow codebase with a complete social layer, real-time messaging, community features, and a full moderation pipeline. The platform now behaves like a hybrid of Stack Overflow, Discord, and a lightweight social network.

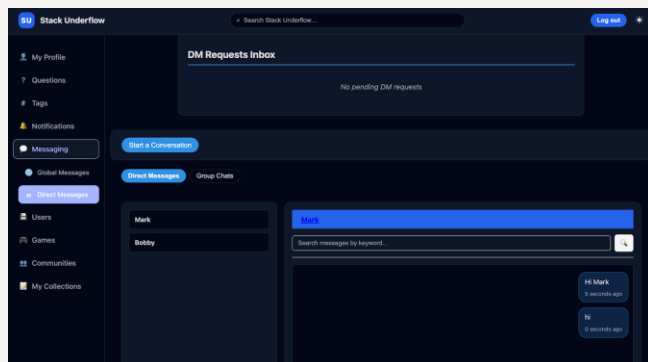
The major additions include:

- **Firestore Authentication** (secure email/password login, persistent sessions)
- **Direct Messaging** (one-on-one messaging, attachment support, DM request system)
- **Group Messaging** (up to 10 participants, admin add/remove controls, typing indicators)
- **User Profiles** (bios, banner images, activity tabs, follower/following lists)
- **Follow/ Block/ Mute System**
- **Collections** (save questions into custom lists)
- **Communities** (topic-based public or private groups)
- **Notifications** (real-time, quiet hours, batching)
- **Moderation Dashboard** (flag flow, appeal system, action logs, rule system)

All features integrate into a unified platform that maintains the Q&A core while enabling modern social interaction.

Demo and Source

Our demo site is available on <https://cs4530-f25-103.onrender.com/>, and repo on <https://github.com/neu-cs4530/fal25-project-fall25-project-group-103>



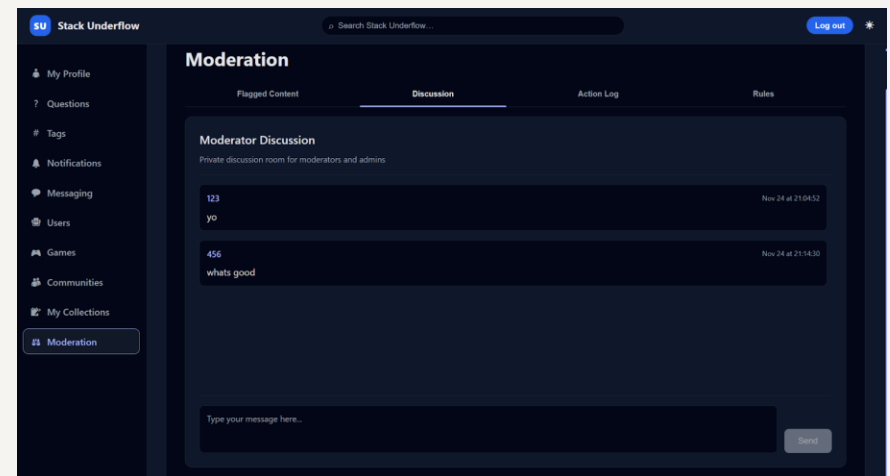
Messaging interface displaying group chats

Technology Stack and Design

We implemented our new social, messaging, and moderation features directly in the Stack Underflow codebase. User profiles, chats, DM requests, follows, collections, communities, notifications, and flags are represented as MongoDB schemas that are created and updated through Mongoose. Real-time messaging and notifications are synced across clients using Socket.IO, with React rendering the DM and group chat interfaces and Chakra-UI modals handling actions like profile edits, message requests, and flag submissions. Firebase Authentication manages secure login and session persistence. Moderation tools—including flag review, appeals, and action logs—are exposed through a dedicated React dashboard that communicates with backend controllers. A notification batching service reduces alert spam while keeping the UI updated through WebSocket events. Our continuous integration pipeline runs automated tests across frontend and backend components, then deploys the platform via Render.

Future Work

While the system is stable and feature-rich, there are several meaningful extensions that could further enhance the platform. Planned improvements include threaded replies in group chats to support more organized discussions, a full private/public profile mode for greater user control, and AI-assisted moderation to help identify toxicity or spam at scale. Moderators would also benefit from context-viewing tools that surface related threads, along with a more unified and polished design system across all pages. Communities could expand with features like pinned posts or event-style announcements, and the follower-based feed could be refined to deliver smarter, more personalized content. With the architecture now in place, the platform is well positioned to support rapid development of these next steps.



Moderation page accessible only to moderators