

Husky Sync: CS4530 Final Project

Team 104: Arielle Bello, Basilia Oferbia, Sathvik Charugundla, Alyssa Mckenzie Griffith

Project Features & Description

Group chat & Notification

To keep users engaged and informed on HuskyOverflow, we introduced a comprehensive notification system that alerts them when their posts receive answers, comments, upvotes, or when someone joins a community they're part of, with updates displayed in a dedicated, context-rich, chronological tab that also batches activity to prevent overload. We also expanded communication tools by enabling multi-participant group conversations where admins can create chats, manage members, assign roles like observers, and maintain organized discussions, while users receive real-time message updates and can actively participate—enhancing collaboration, clarity, and overall community interaction on the platform.

Accessibility

To support users with diverse accessibility needs on HuskyOverflow, we introduced a personalized accessibility experience that allows users to zoom into interface elements, adjust text size, resize the application window responsively, and easily switch between languages through a clear settings menu. These preferences sync across the platform, can be reset to default at any time, and are thoroughly tested to ensure reliability. We also included helpful features such as confirmation messages, downloadable accessibility shortcuts, and optional saved settings, enabling users to navigate the platform comfortably, independently, and with a fully customizable experience.

Security

To ensure users can participate on HuskyOverflow with confidence, we strengthened the platform's security and privacy features by protecting personal data, enforcing strict authentication rules, and validating all incoming requests. Users benefit from safeguards such as login attempt limits, two-factor authentication, session management, password strength guidance, and sanitized inputs that reduce the risk of unauthorized access. Additional measures—including strict credential validation, role-based permissions, and account recovery options—further enhance trust and allow users to engage on the platform without fear of data breaches or compromised accounts.

Tech Stack & Design

Back-End

In the back end, we built a secure and scalable infrastructure that protects user data while powering key HuskyOverflow features such as notifications, accessibility settings, and group conversations. We implemented strict request validation, login attempt limits, two-factor authentication, session management, and role-based permissions to ensure accounts remain protected at all times. User preferences—such as accessibility settings and language options—are stored in dedicated schemas and updated through sanitized, validated API requests. Real-time notifications for answers, comments, upvotes, and group messages are delivered through Socket.IO, which maintains persistent WebSocket connections for instant updates. For group chats, the backend manages participant roles, message handling, admin permissions, and dynamic membership changes, ensuring conversations stay synchronized and secure across all clients.

Front-End

For front-end development we built a user-focused interface with protected routing that guards all app routes, layers in session timeout handling (token refresh and modal warnings) along with a two-step login (2FA), user sign up, forgot/reset password feature and user profile driven account deletion/reset functionality. Login steps, password resetting, group chat messaging and biography editing all have centralized API calls in axios modules with enhanced UI that has error handling (ex. Prevents those titled "observers" to send messages). Theme and locale state are applied app-wide via context-backed hooks so typography and spacing respond instantly to user toggled UI changes. For language features, i18n is an integration framework used to support user languages options. Core page implementations include sortable question lists, tagged browsing, ask/answer pages with user voting and thread comments, along with collection-aware buttons that let users save/unsave their questions or create new collections within their accounts. Real-time collaboration uses Socket.io for direct messages and group chats. Including creating messaging flows, admin/observable roles, participant add/removal, mute/unmute per chat, and notification streaming with a filter notification page and a batch pop-up that appears when a new event is triggered. Cypress testing is also used to verify page behaviors, UI interactions and overall responsiveness

Future Work

Future work on HuskyOverflow can focus on enhancing both security and user experience by expanding several advanced features. Strengthening backend protections through stricter CORS enforcement will ensure that only approved origins can access the application. Additional accessibility improvements—such as enabling or disabling a Voice Command Mode, along with immediate visual or auditory feedback for recognized commands—can make the platform more inclusive. On the communication side, future updates may introduce mobile push notifications, group-chat enhancements like renaming chats, pinning messages, adding emoji support, and creating observer-based "Channels" for one-way announcements. Finally, we plan to integrate an AI-generated summary system that condenses long question threads into key points, unresolved issues, and differing approaches, helping users quickly grasp discussions and contribute more effectively.

