CS4530 FINAL PROJECT

"GROUP 411: MARIA, ISHA, ROBERT, AND ANNE"

Our Features

Our main focus was to enhance the usability of current chatting, posting, and overall functionality of the website. We outline the features below. For our first user story was focused on enhancing chat by allowing users to delete and edit their own messages (private and global), adding people and removing them from group chats, and allowing users to add and send attachments.

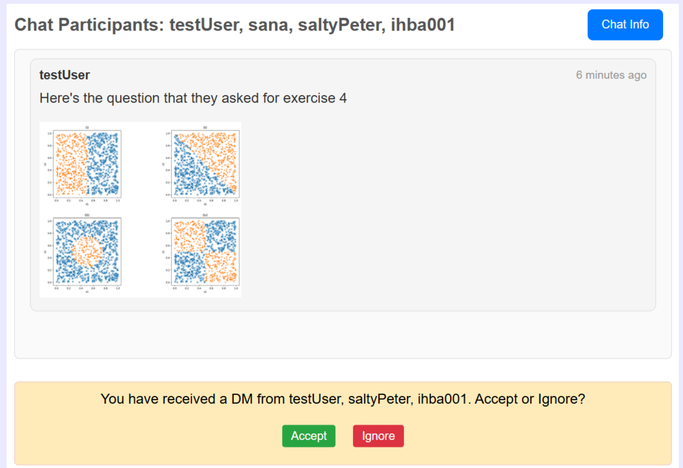
For our second user story, we added the ability to report posts (questions) and tracking the number of reports on a question to remove it, keeping track of total upvotes for a user to mark them as certified and displaying the number on their profile page, allowed users to sort posts by certified users, added the ability for users to search through global chat using specific keywords, and we added a blocking functionality where users are able to block other users not allowing them to create a chat with the users that were blocked.

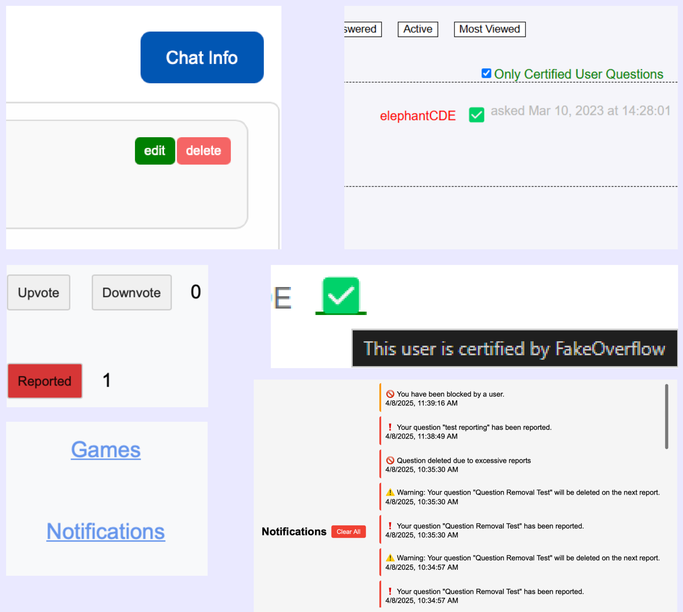
For our third user story, we enhanced the blocking user functionality allowing users to unblock as well, we tracked the number of blocks to suspend a user and not allow them to sign back in, we added a warning that allows a user to accept or decline chats with unfamiliar users. We also added a notifications page to allow users to see if their posts have been reported warning them that it will be deleted, notifying them once they get certified, and allowing them to also clear all notifications.

Demo and Source

Our **demo site** is available at https://cs4530-s25-411.onrender.com/, and our **source code** is available at https://github.com/neu-cs4530/spring25-team-project-spring25-project-group-411.

Images





Our Technology Stack & Design

We implemented our additional features in the Stack Overflow starter code and used the tech stack that was recommended. The only additional service we used was AWS S3 to store files for our sending attachments feature.

Generally we made our edits to the data types and structure of the starter code minimal in order to follow the continuous integration/ continuous deployment framework. When adding/deleting participants or blocked users, displaying notifications, and certifying users, we edited the corresponding global messages, direct messages, and profile components to get the updated user data type to store relevant information for the feature and utilized existing update functions.

For sorting posts and tracking upvotes, the process was similar, but the messages files were edited on the backend and global messages components on the front end.

For editing/deleting messages and sending attachments, messages was also edited for both global and direct. New messages service functions had to be made for updating the message. For sending attachments, we had to set up an AWS S3 database and store the links of the files in a message.

Future Work

Future work could involve implementing additional features outlined in our "extension" conditions of satisfactions as per the user stories. For our first user story, these enhancements include receiving notifications for unread messages, enabling/disabling notifications, and setting the chat modes to private (invite-only) or public. The second user story has conditions for seeing the most popular/active posts and saving and sharing posts for later viewing. Lastly, for our third user story, future improvements involve limiting daily user blocks/reporting and integrating content warnings for posts and messages.