

FAKESTACKOVERFLOW CLARIFICATIONS

OUR FEATURE: CLARIFICATIONS

In our final project, we implemented the Clarifications feature for FakeStackOverflow. Clarifications lets users ask follow-up questions on problem posts without detracting from the original question. They allow users and admins to reply, resolve, and track side conversations as they open and close.

The original Question/Answer application did not have a feature for those adjacent, tangentially related questions that prioritized quick resolutions. Users would rely on the small single thread below the original post if they had a question about the question. Our feature ensures that those small asides don't go forgotten and un-clarified.

Our Clarifications feature introduces a dedicated, real-time single-thread, single-topic discussion system with role-based permissions. Regular logged in users can ask clarifications and resolve their own threads, others can reply, and admins can resolve and freeze/lock threads when needed. Clarifications are accessible through question pages and update live across multiple clients

TECHNOLOGY STACK

We implemented the Clarification feature as a real-time discussion system in the existing Fake Stack Overflow codebase using a Controller-Service-Model architecture. Each clarification is represented as a MongoDB document with fields for text, status, endorsement, threading, and soft deletion. During interactions, React hooks and Socket.IO enable simultaneous backend-to-frontend communication, updating and rendering clarifications, status changes, and admin actions in real time across all users. The backend implements four-tier role-based access control through middleware (Non Authenticated, Authenticated Users, Authenticated users who can post answers, Admin). Our testing infrastructure includes Vitest for frontend tests, Jest for backend tests with 95%+ coverage, and Cypress for end-to-end testing. The system uses a hybrid REST API and WebSocket architecture, achieving sub-100ms real-time updates via Socket.IO, sub-200ms queries through database indexing, and sub-500ms responses via pagination. Built entirely in TypeScript with NPM Workspaces monorepo, OpenAPI documentation, and continuous integration validating quality before deployment.

DEMO + SOURCE

render link: <https://fall25-project-fall25-project-group-802-1.onrender.com/>
github repo: <https://github.com/neu-cs4530/fall25-project-fall25-project-group-802>

FUTURE WORK

For future work, the biggest thing we'd want to improve is scaling and reliability, especially for the real-time part of our Clarifications feature. Right now it works great for a class project, but if more users are online at once, we'd want to make Socket.IO scale across multiple server instances (ex: using a shared adapter) so updates stay instant and consistent for everyone. It's important because real-time features are the first thing that break when you go from a few users to many users. Currently we might have like 2-10 people online, one server instance, and everything feels instant. But if this ever ran in a real environment with many users, the system has to handle two big realities: more traffic and multiple servers.

Admin Clarification Management

My Clarifications

Reported Queue

Audit Log

User Management

Filter by Status:

All Statuses

My Clarifications (8)

Clarifications on your questions, answers, or created by you

The screenshot shows three clarification cards for user Janvi. Each card has a status indicator in the top right corner: 'Resolve Pending' (blue), 'Unresolved' (orange), and 'Resolve Pending' (blue). The first card has a 'Verified' badge (green checkmark). The text of the cards is: 'This is a test.', 'limit', and 'One more clarification'. Below each card is a 'Question:' field with a unique ID and a status dropdown menu.

Preventing memory leaks in React applications

1 answers

Ask a Question

View Clarifications

The screenshot shows a question page for 'Preventing memory leaks in React applications'. The 'Clarifications' section is highlighted with a purple header. It includes a 'Back to Questions' button, a 'Jump to Question' button, and a 'Clarifications' header with a search icon. Below the header, there is a section for 'Authenticated User' with a 'View' button and a 'Post' button. A 'Ask Clarification' button is also present. A summary bar shows 'Unresolved: 1', 'Pending: 3', and 'Resolved: 2'. A 'Filter by Status:' dropdown is set to 'All', with options for 'Unresolved', 'Resolve Pending', and 'Resolved'. The bottom of the screenshot shows a user comment from 'user123' on Nov 27 at 15:08:25, with a 'Copy Text' button, a 'Share Link' button, and a note that '0 replies remaining'.