



# HuskyFlow

CS4530 | GROUP 502: JOSH, ISSA, ANDRE, RODRIGO

[Source Repo](#)

[Public Demo](#)

## DESCRIPTION

HuskyFlow is an enhanced implementation of a Stack Overflow-like question-and-answer platform, developed using TypeScript, React, Node.js, and MongoDB. Our team implemented three major user stories focusing on content management, security infrastructure, and social features. The platform enables users to ask questions, provide answers, organize content through communities and collections, and engage with others through messaging and notifications—all while maintaining robust security through two-factor authentication, role-based access control, and comprehensive input validation.

## DEMO/SOURCE

Our demo site is available at <https://cs4530-f25-502.onrender.com/>, and our code at <https://github.com/neu-cs4530/fall25-project-fall25-project-group-502>

To test admin control features, use these admin credentials:  
Username: admin  
Password: adminuser!

### Load Drafts and Markdown Editor

Editing draft: DRAFT

Save Draft Load Draft (1) Clear Draft

Your Question Drafts

DRAFT

```
What's wrong with my code? ""typescript const newPosts = posts.splice(1, 3) setPosts(newPosts) ""...
```

Load Delete

Question Text\*

Add details

What's wrong with my code?

```
""typescript const newPosts = posts.splice(1, 3) setPosts(newPosts)
```

Markdown formatting is supported.

### Profile management

@joshuakung

Joined December 2025  
0 Friends

About

I am here: 11x engineer, CEO of fake stack overflow

Recent Posts

Questions: 1 Answers: 3

Recent Questions

### 2FA and Email Verification

#### Sign up for FakeStackOverflow!

Please enter your username.

TestUser

Please enter your password.

\*\*\*\*\*

Password must contain:

- ✓ At least 8 characters
- ✓ At least one number
- ✓ At least one symbol

Confirm your password

Show Password

Continue

Have an account? [Login here](#)

#### Sign up for FakeStackOverflow!

##### Email Verification Required

Please enter your email address to receive a verification code.

Please enter your email address.

Enter your email address

Send Verification Code

## TECH AND STACK

### Frontend

- React 19 with TypeScript for type-safe UI components
- Vite for fast development and optimized production builds
- React Router for client-side routing
- React Markdown with rehype-highlight for GitHub Flavored rendering
- Lucide React for modern, consistent iconography
- Axios for API communication

### Backend

- Socket.IO for real-time WebSocket notifications
- Mongoose for MongoDB object modeling
- Argon2 for password hashing
- SendGrid for email-based 2FA verification
- Helmet for security headers
- Express Validator + DOMPurify + XSS for input sanitization

### Storage & Infra

- MongoDB for flexible document-based data storage
- Cloudinary for secure profile image hosting
- Render for deployment and hosting

## DESIGN CHOICES

Snapshot-based version history instead of diffs for predictable restoration and querying

Separate Drafts collection to isolate unpublished content from production data

Client-side Markdown rendering for improved security and reduced server load

WebSocket-based notifications for low-latency, event-driven real-time updates

Device fingerprinting for trusted device management, reducing 2FA friction

Tag-driven recommendations using collection data instead of invasive tracking

Three-tier architecture (React → Express → MongoDB) with indexed queries for fast discovery

## FUTURE WORK

Planned features include a code playground for testing snippets, AI-powered duplicate detection, real-time collaborative editing, and enhanced moderation tools. Additional improvements include an analytics dashboard, multi-language support, reputation system with badges, advanced search filters, React Native mobile app, and integrations with GitHub, Slack, and Discord.

Technical enhancements focus on scalability and performance: migrating to GraphQL, implementing Redis caching, adopting microservices architecture, and expanding E2E test coverage with Playwright. Infrastructure improvements include database sharding, image optimization, automated Cloudinary cleanup, service workers for offline PWA capabilities, and CDC for real-time data synchronization.