

CS5430 Final Project: Connecting Code, Curating Content, Controlling Quality: Elevating HuskyFlow

Group 905: Jonathan Blinder, Curie Cha, Addison Seay, James Sweeney

Our Feature

While HuskyFlow is a functional Q&A platform, we noticed users had trouble finding related content, running code blocks in posts, and admins lacked proper management and analytics tools. We built four interconnected features to address these gaps:

Content Recommendation System

Users viewing a question now see related questions ranked by relevance. A feedback system lets users rate recommendations. The feedback is shown to admins.

Runnable Code Blocks

Users select a language and execute code directly in the browser, seeing output, errors, and execution time immediately.

Moderation Queue

When content or a user receives 3 reports, it automatically enters a moderation queue where admins review and decide whether to reinstate or remove it.

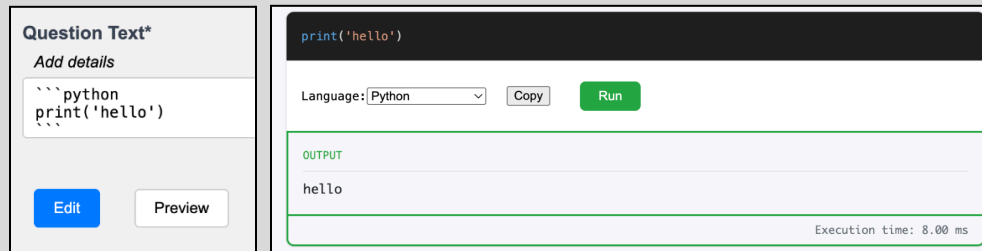
Admin Dashboard

Admins see a dashboard with three tools: moderation queue for reviewing reported content and users, active users analytics tracking platform activity in day/week/month, and recommendation analytics showing feedback data.

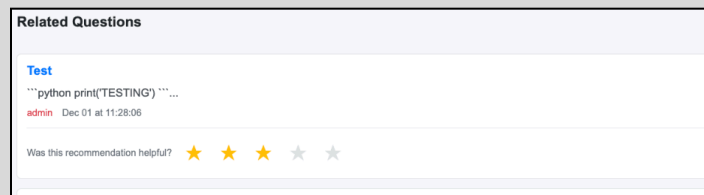
Demo and Source:

Demo site: <https://cs4530-f25-905.onrender.com>

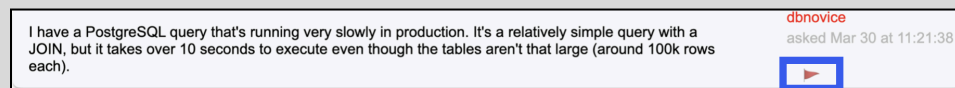
Code: <https://github.com/neu-cs4530/fal25-project-group-905>



Shown on the left, when creating a question/comment/answer, type code in Markdown format. When created, the runnable code block is displayed, shown on the right.



When viewing a question, a list of related questions are shown below. Each one can be rated 1-5 stars.



Questions and users can be reported by clicking on the flag icon.

Our Technology & Stack Design:

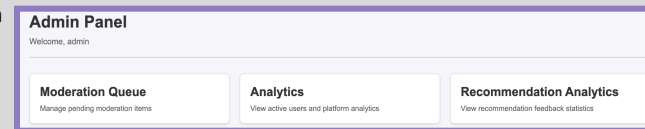
We implemented these features in the existing HuskyFlow codebase.

- The recommendation system uses TF-IDF vectorization to compute similarity scores between questions. Vectors and similarity scores are cached in memory to avoid redundancy. Recommendations are fetched through a React hook that calls the backend service, which returns the most similar questions.
- Runnable code blocks use the Judge0 API through a backend controller that handles language detection and submission management.
- The moderation system tracks report counts in MongoDB, automatically queuing and hiding content when it reaches 3 reports. Moderation items are created by a queue service and displayed in the admin panel through a React component. When content is hidden or reinstated, socket-io emits real-time update events, ensuring users see changes immediately.
- Analytics are logged to MongoDB through an analytics event model that tracks user activity and recommendation interactions. The active users service computes statistics over time windows. Recommendation analytics track ratings and click events. Visualizations are rendered using Chart.js.

Our CI pipeline runs automated tests, and then deploys the site on Render.

Future Work:

We struggled to display as many insights as we wanted on the admin dashboard because implementing the data types and infrastructure to track all the necessary events took longer than expected. Future work might expand the analytics system to track system performance and user behavior trends. We could add predictive analytics that detect unusual spikes and drops in the data. The recommendation system could be improved with modern ML techniques for more nuanced and accurate similarity scoring. Personalization would tailor recommendations to each user based on their history or community memberships. For the runnable code blocks, we could add execution history so users can see past runs, or support multi-file projects instead of just single blocks. The moderation system could use dynamic thresholds that adjust based on content type, user reputation, or report severity, rather than the fixed threshold we currently use.



Admins can land on the Admin Panel via the navigation sidebar. Admins can navigate to Moderation Queue, User Analytics, and Recommendation Analytics Dashboards from this main panel.

