

## OUR NEW FEATURES

In the world of media, there is often more to consume than we have time for. To ease this process and provide a more optimized user experience, we provide features that increase the efficiency of the user by streamlining tedious features on our forum, BeyondTheBinary.

Highly technical posts are difficult to read, so we implemented markdown to improve the readability of posts, comments, and answers. Since it is hard to trust users' responses to questions, we also implemented a reputation system to help users easily find credible answers to questions. Finally, technical jargon can separate related questions or answers, so we implemented semantic search to relate all questions to an associated answer while not requiring the use of similar terminology as well as the same technology to recommend between existing tags.

## FUTURE WORK

Future development would be in line with our goal of increasing user efficiency. This has already been done to a good extent with our added features, and they could be improved upon to further increase user efficiency. For instance, more languages could be added to runnable Markdown code blocks, semantic search could be optimized and potentially utilize other algorithms, reputation could be presented differently to ideally engage users, and more options (like Facebook, etc.) could be added to SSO to improve the likelihood that any particular new user can get started with our site right away.

We could also explore new features to work towards our goal such as autocomplete/editor suggestions for quicker writing and posting, support for LaTeX formatting, and a preview of frontend (HTML/CSS/JS) code block code ran inside of the site. Above all else, we'd want to keep developing a seamless user interface that gives the user lots of power while also staying out of their way.

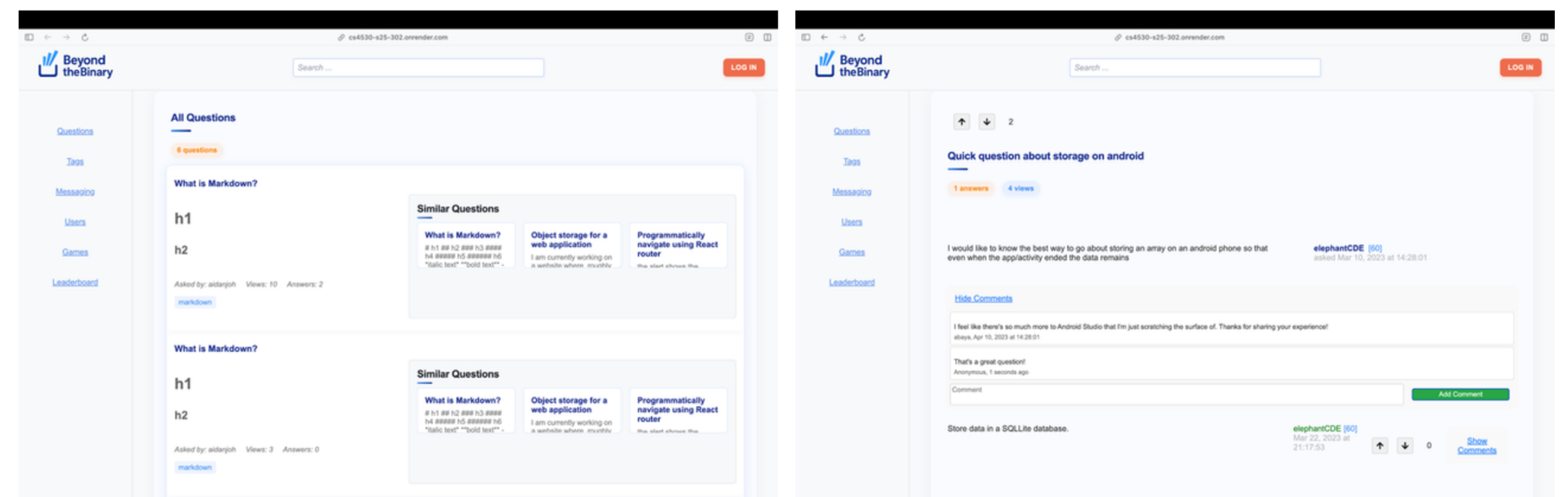
## DEMO AND SOURCE

Our demo site is available at <https://cs4530-s25-302.onrender.com/>, and our code at <https://github.com/neu-cs4530/spring25-project-group-302>

## TECHNOLOGY STACK AND DESIGN

Our system integrates reputation tracking, session management, semantic search, and Markdown processing across frontend and backend layers. The reputation system uses React-based VoteComponents and Leaderboard for user interactions, with backend QuestionService/AnswerService processing votes and UserService calculating weighted reputation (applying  $1 + \log_{10}(\text{voter\_rep} + 1)$  scaling and milestone bonuses) while WebSocket broadcasts ensure real-time updates. Session handling combines Google SSO via React's GoogleAuthService with secure backend SessionService-generated 256-bit tokens, stored in CORS cookies and validated every 5 minutes. Semantic search leverages Hugging Face's all-MiniLM-L6-v2 model for client-side embeddings, enabling MongoDB Atlas vector similarity searches via SearchBar/TagInput components. Markdown support features react-markdown/GFM rendering in Q&A views with syntax-highlighted previews via rehype-raw, while react-shiki powers executable Python code blocks (via AWS Lambda /coderun/getRunCode) with theme toggling and serverless execution. Backend ORMs store raw Markdown content, and Lambda URLs/environment variables secure code execution workflows.

Our CI/CD pipeline runs an automated test suite on our frontend and backend, then deploys the site using Render and Heroku.



All Question Page, displaying our new UI with questions featuring markdown, and our similar questions displayed

Specific Question Page, displaying our new UI with votes, reputation, anonymous comments, and more