



The Giants of Whose Shoulders We Stand

While several platforms currently serve the academic community, each has its limitations. For instance, Piazza.com, a popular choice among STEM majors, offers a forum-like space for students to discuss course material. However, its outdated user interface and poor user experience make it feel isolated and disconnected from the broader digital learning environment. This underscores the need for a more integrated and user-friendly platform like KitaHub.

Similarly, CodePost.io is another commonly used platform, especially for introductory programming assignments in computer science courses. While it's effective for simple assignments like "hello world" programs or basic algorithms, it struggles to scale for more complex, real-world projects. CodePost's limited flexibility becomes apparent when dealing with assignments that require complex testing or multiple configurations. This is where KitaHub comes in, aiming to support a wide range of assignments—from basic scripts to comprehensive full-stack web applications—offering a seamless experience for both students and instructors across various programming languages and frameworks.

On the technical side, KitaHub's front end is being built with Next.js and will be deployed via Vercel. This combination brings significant benefits, such as server-side rendering and static site generation, which optimize performance and SEO. Vercel's seamless integration with Next.js also provides rapid deployment pipelines and excellent support for scaling with minimal infrastructure management, making it the ideal choice for a dynamic platform like KitaHub, which requires frequent updates and high performance.

For the Q&A board system, KitaHub will leverage Redis Cloud for its pub/sub capabilities to handle real-time updates. Clients will connect through WebSockets to maintain instant communication between students and the



platform. This setup ensures that new questions and answers appear immediately without requiring page refreshes, providing a truly real-time interactive experience that significantly improves over traditional platforms like Piazza.

One of the key challenges in building KitaHub is ensuring the safety of potentially malicious code submissions. To address this, KitaHub will integrate with the Sphere Engine API, a secure environment designed to handle untrusted code submissions. This API provides an isolated environment to run, compile, and test code across multiple programming languages, ensuring that any potentially harmful operations are sandboxed. This robust security feature is essential for KitaHub to scale across a wide variety of coding assignments without compromising safety.

The backend will utilize AWS API Gateway and Lambda functions to manage HTTP API requests in a serverless manner. This ensures that the infrastructure scales dynamically with user demand, reducing overhead and costs while maintaining high availability. KitaHub will adopt Neon-hosted Postgres, a serverless, autoscaling database that adapts to varying loads for data storage. Neon offers high performance and cost-effectiveness, as it can automatically scale resources up or down based on the needs of the platform, making it an ideal choice for a dynamic, fast-growing platform like KitaHub.

KitaHub will employ Redis as a general cache to further optimize performance, speeding up frequent data retrieval operations and reducing the load on the primary database. DynamoDB will be used for miscellaneous configuration settings, providing scalable and flexible storage for less critical but necessary information such as user preferences or system configurations. This architecture ensures that KitaHub is both robust and scalable, capable of handling the diverse needs of its users efficiently.