



Innovative Solutions for Modern Education: The Kitahub Story

The Problem

In the realm of STEM education, students often face significant challenges. One prevalent issue is the discrepancy in assignment performance, where code that functions correctly on a student's computer fails when tested on an instructor's computer. This not only impacts grades but also fails to accurately reflect a student's programming capabilities. Furthermore, traditional communication platforms used in education do not adequately support the dynamic and interactive needs of STEM related discussions, leading to delays in feedback and gaps in learning.

Additionally, the current recruitment process in the tech industry heavily relies on contrived coding challenges that do not accurately represent a candidate's real-world coding ability or problem-solving skills. This outdated model overlooks the rich insights that can be gleaned from a student's actual coursework and interactions on educational platforms.

The Solution

Kitahub addresses these issues by offering an integrated solution that combines robust code testing with a modern discussion board, designed specifically for educational environments. Our platform ensures that students' code is tested in a standardized environment, mirroring the instructor's setup, thereby eliminating discrepancies in code execution. The discussion component enhances real-time communication and collaboration, fostering deeper understanding and more efficient learning.

Moreover, Kitahub innovatively extends its utility to the recruitment process. By collecting and analyzing data from actual assignments and discussions, Kitahub enables company recruiters to assess potential candidates based on genuine academic performances and collaborative interactions. This approach allows



recruiters to evaluate talent through a more authentic lens, focusing on real assignments and meaningful contributions in academic discussions rather than solely on performance in contrived coding tests. Through this, Kitahub not only enriches the educational journey but also bridges the gap between academic achievement and career advancement, making it a pivotal tool for students, educators, and recruiters alike.

KitaHub Roadmap

Sprint Overview

1. **Sprint 1 (Sept 30th – Oct 14th):** Foundation Development
2. **Sprint 2 (Oct 14th – Nov 6th):** Dashboard Implementation
3. **Sprint 3 (Nov 6th – Nov 20th):** Pub/Sub System Development
4. **Sprint 4 (Nov 20th – Dec 4th):** Production Environment Setup
5. **Sprint 5 (Dec 4th – Dec 18th):** System Testing and User Engagement

Sprint 1: Foundation Development

Timeline: September 30th – October 14th

Objectives

- **System Design Diagram:**
 - Develop a high-level system architecture [diagram](#).
 - Outline components such as frontend, backend, database, and third-party services.
- **PostgreSQL Data Model:**
 - Create the initial data model for PostgreSQL.
 - Publish the data model diagram on [DbDiagram.io](#).
- **API Design:**
 - Design RESTful APIs for the dashboard and question-and-answer board.
 - Document endpoints, request/response formats, and authentication methods.



- **Authentication System:**
 - Implement a secure authentication system.
 - Include user registration, login, password hashing, and session management.

Deliverables

- High-level system design diagram.
- Published PostgreSQL data model on DbDiagram.
- API design documentation.
- Functional authentication system integrated into the application.

Sprint 2: Dashboard Implementation

Timeline: October 14th – November 6th

Objectives

- **Dashboard UI:**
 - Develop the [user interface](#) for the dashboard using Next.js.
 - Ensure responsive design and intuitive navigation.
- **Redis Pub/Sub Data Model:**
 - Design the data model for the Redis Pub/Sub system.
 - Outline channels, message formats, and subscription patterns.

Deliverables

- Fully implemented dashboard UI.
- Redis Pub/Sub data model documentation.

Sprint 3: Pub/Sub System Development

Timeline: November 6th – November 20th

Objectives

- **Local Pub/Sub System:**
 - Implement the Redis Pub/Sub system running locally.



- Test message publishing and subscription functionalities.
- **Development Environment Setup:**
 - Create a development environment using Docker and LocalStack.
 - Orchestrate Next.js app, Redis Pub/Sub system, PostgreSQL database, and other services.

Deliverables

- Functional local Redis Pub/Sub system.
- Docker-compose files for development environment.
- Documentation for setting up and running the development environment.

Sprint 4: Production Environment Setup

Timeline: November 20th – December 4th

Objectives

- **Production Deployment:**
 - Set up the production environment on AWS, Redis Cloud, and Neon.
 - Deploy the Next.js app, databases, and services to the cloud.
- **Networking and Security:**
 - Configure Virtual Private Cloud (VPC) and networking settings.
 - Ensure secure communication between services.
- **Quality Assurance Testing:**
 - Perform end-to-end testing in the production environment.
 - Identify and fix bugs or performance issues.

Deliverables

- Live production environment with all components deployed.
- VPC and networking configurations.
- QA test reports and issue logs.

Sprint 5: System Testing and User Engagement

Timeline: December 4th – December 18th



Objectives

- **Stress Testing:**
 - Conduct performance and load testing to evaluate system scalability.
 - Optimize system based on test results.
- **User Onboarding:**
 - Coordinate with Ramesh to create a class on KitaHub.
 - Facilitate classmates' sign-up and interaction with the system.
- **Feedback Collection:**
 - Gather user feedback on functionality and usability.
 - Prioritize enhancements and fixes based on feedback.

Deliverables

- Stress test results and optimization reports.
- Successful user onboarding and interaction records.
- Compiled user feedback and action plan for improvements.