CS4530 Final Project: “Stack Overflaux”

Group 204: Thomas Walewski, Jackson Green,Arnav Mahale, Selwyn George

Our Feature: Stack Overfaux

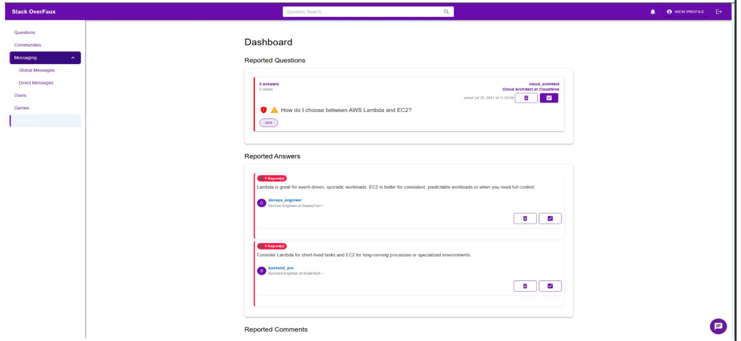
Stack Overfaux is a full-stack Q&A web application inspired by Stack Overflow. We reimagined the Q&A experience by introducing features tailored to three distinct user types: **Novices**, **Experts**, and **Moderators**. Novices can ask questions, receive AI-generated answers via Gemini, and invite credentialed experts to respond. Experts can showcase credentials, contribute tutorials/demos, and engage in communities. Moderators manage reported content, pin high-quality answers, and monitor site analytics through an interactive dashboard.

User Experience Highlights

🔰 Novice Features - see most upvoted answers, ask gemeni for help, colorblind mode
🎓 Expert Features - Credentials displayed on profile + answer, create markdown based demos,
🛡️ Moderator Features -View and manage reports (questions/comments/users), live analytics, pin

Demo & Source Code

👉 **Live Demo**:
https://cs4530-s25-204.onrender.com
📂 **GitHub**: spring25-project-group-204



Dashboard

Reported Questions

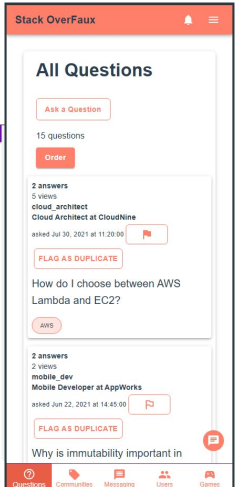
2 answers
5 views
cloud_architect
Cloud Architect at CloudNine
asked Jul 30, 2021 at 11:20:00
How do I choose between AWS Lambda and EC2?
AWS

Reported Answers

1 Report(s)
Lambda is great for event-driven, sporadic workloads. EC2 is better for consistent, predictable workloads or when you need full control.
devops_engineer
DevOps Engineer at DeployFast •

1 Report(s)
Consider Lambda for short-lived tasks and EC2 for long-running processes or specialized environments.
backend_pro
Backend Engineer at ScaleTech •

Reported Comments



All Questions

15 questions

2 answers
5 views
cloud_architect
Cloud Architect at CloudNine
asked Jul 30, 2021 at 11:20:00
How do I choose between AWS Lambda and EC2?
AWS

2 answers
2 views
mobile_dev
Mobile Developer at AppWorks
asked Jun 22, 2021 at 14:45:00
Why is immutability important in

Our Technology Stack & Design

Our tech stack includes a React frontend styled using the Material UI (MUI) component library, which allowed us to build a clean, accessible, and responsive user interface. For the backend, we used Node.js with Express.js to build scalable REST APIs, and MongoDB as our NoSQL database to store all user data, questions, answers, comments, and community structures. Real-time communication between clients and the server is handled using Socket.io, enabling dynamic updates for features like expert invites, pinned answers, and notification delivery.

A major engineering effort went into building the moderator feature, which we implemented by extending the existing user model to include a isModerator feature flag. With this flag, we were able to introduce a full moderation suite: moderators can delete users, flag or clear reports on questions, answers, and comments, pin high-quality answers, and view live analytics through a custom dashboard. The analytics system provides insights like total questions answered, user report counts, top contributors, and other engagement metrics, all pulled in via custom API endpoints and rendered with Material UI chart components.

We also designed modular, maintainable frontend and backend architectures, following a controller-service-repository pattern on the server to keep logic organized and testable. On the frontend, we utilized custom React hooks for managing socket events and API interactions. To support accessibility, theming, and user preferences, we implemented dynamic theming in MUI based on user profile settings like colorblind mode and high contrast view.

Future Work

We see huge potential in expanding the Communities feature. Right now, experts can join communities and view engagement analytics — but we want to take this further.
We aim to build a more fleshed-out community system that allows users to create their own spaces, either private (invite-only) or public (open to anyone). Each community would have its own rules, moderators, and thread-style Q&A boards.
Future updates will support role-based permissions within communities — think moderators, contributors, and viewers — enabling better content management and collaboration.
Longer term, users could customize the look and feel of their communities and even host community-led tutorials or events — creating a more interactive and educational space.

We also want to allow users to play more games with each other, including Wordle and possibly chess!