Technical assignment (take home)

Objective:

Create a simplified version of a status page application similar to services like StatusPage or Cachet or Betterstack or Openstatus. The application should allow administrators to manage services and their statuses, and provide a public-facing page for users to view the current status of all services.

Project Scope:

The primary goal is to create a working application where anyone can log, view, and manage status of multiple applications. The application should include public page where customers and end users can know the status of the application. You may choose any front-end framework, such as React, Vue.js, or others you are comfortable with.

Evaluation Criteria:

- Code Quality: Clean, well-organized, and commented code
- Architecture: Proper separation of concerns and scalable design
- Frontend Skills: Responsive design, state management, and component structure
- Backend Skills: API design, database integration, and authentication implementation
- Problem-Solving: How they approach and solve challenges during development
- AI first development: Ability to use AI LLMs/IDEs and be able to fast-track development

Submission:

- Provide a GitHub repository with the complete code
- Include a README with setup instructions and any necessary documentation
- Deploy the application to a free hosting service (e.g., Heroku, Vercel, Replit) for easy review
- Create a short loom showing the demo of the app and a quick walkthrough of the codebase

IMPORTANT: We would want you to to use AI tools (Cursor,v0, github copilot, claude sonet etc), web frameworks, ORMs, and/or SaaS tools like Auth0/Clerk(user & team management) etc for building this app to be able to fast track development .

Key Features:

1. User Authentication
2. Team management
3. Organization (multi-tenant)
4. Service Management:
    - CRUD operations for services (e.g., "Website", "API", "Database")
    - Ability to set and update the status of each service (e.g., "Operational", "Degraded Performance", "Partial Outage", "Major Outage")

5. Incident/Maintenance Management:

    - Create, update, and resolve incidents or scheduled maintenances
    - Associate incidents with specific services
    - Add updates to ongoing incidents

6. Real-time Status Updates:

    - Implement WebSocket connection to push status changes to connected clients in real-time

7. Public Status Page:

    - Display current status of all services
    - Show active incidents and maintenances
    - Display a timeline of recent incidents and status changes

Basic Styling:

Clean, minimalistic UI similar to Linear. You can use a framework like ShadcnUI

Optional Stretch Goals (if time permits):

- Implement email notifications for status changes
- Add metric graphs for service uptime percentage
- Create a simple API for external status checks

Timeline:

Please complete the assignment within 3 days of receiving it. If there are any questions or if additional time is needed, feel free to reach out.

We look forward to reviewing your innovative solutions and understanding how you approach and solve real-world problems. Good luck!