**High-Level Design Document**

XOrithm - Server Status Dashboard

Written by: Mayer Mamdouh

**Project Overview**

The goal of this project was to build a simple and clean server status dashboard for XOrithm using Next.js.

The dashboard shows the current status of multiple servers, and lets users log in to access it using either Google or GitHub.

I wanted the app to be easy to use, lightweight, and responsive. I also focused on making the UI clear, so anyone can tell right away if something's wrong with any of the servers.

**Tech Stack Used**

* Framework: Next.js (App Router)
* Styling: Tailwind CSS
* Icons: react-icons
* Auth: NextAuth.js with Google and GitHub providers
* Data: Manually created static JSON-like data (not using any API)

**Authentication Flow**

I used NextAuth to handle all login logic. On the homepage, users see two buttons to log in either with Google or GitHub.

* If the user is not logged in and tries to go to /dashboard, they are redirected back to /.
* Once the user logs in, they are sent straight to /dashboard.
* The session is stored using JWT strategy.

I added middleware to protect the /dashboard route and prevent unauthenticated access.

**Dashboard Design & Logic**

* The dashboard is the main part of the app. Here's how I approached it:
* Data structure: I created static data manually for the servers, including:
* Name
* Status (up, degraded, down)
* IP
* Response time
* Uptime %
* Uptime history by day
* Filtering: There's a dropdown that filters servers by status (all, up, degraded, down).
* Sorting: Another dropdown lets users sort by uptime: low to high or high to low.
* Details: When a server is clicked, a popup opens showing full server details. Inside the popup, the uptime history is shown with a color-coded bar and a tooltip on hover for extra info.

**Component Structure**

* /app/page.tsx 🡪 Homepage with login options
* /app/dashboard/page.tsx 🡪 Main dashboard page
* /app/layout.js🡪 Shared layout with AppBar and session provider
* /app/api/auth/[...nextauth]/route.js 🡪 NextAuth route handler for login providers
* /middleware.js 🡪 Protects /dashboard from unauthorized access
* /components/AppBar.jsx 🡪 Top navigation bar with logo and sign out
* /components/Popup.jsx 🡪 Modal for showing full server details on click
* /components/SignInButton.jsx 🡪 Reusable login button with Google or GitHub
* /components/SiginOut.jsx 🡪 Button to log out the user
* /data/server.js 🡪 Static manually-written server data
* /lib/nextAuth.js 🡪 NextAuth configuration (Google, GitHub)
* /providers/NextAuthProvider.jsx 🡪 Wraps app in session provider to enable auth context

**Design**

* Simple responsive layout -- works well on both desktop and mobile.
* Tailwind was used because it's quick and easy for layout/styling.
* I used useSession() in the dashboard to welcome the user with their name.
* The filtering and sorting were implemented using React useState() and array methods.

**To enhance the user experience and show attention to detail, additional features were added such as:**

* A "Last Updated" timestamp to simulate real-time data awareness.
* A 404 page for invalid routes to handle broken URLs gracefully.
* Live filtering through a search bar and server count display.