

Full Stack Development Intern Assignment: Interactive Dashboard with Next.js

Objective

Develop a full-stack interactive dashboard application using Next.js, Recharts, and Supabase. The dashboard should display various metrics, allow user authentication, and provide data visualization capabilities.

Requirements

1. Setup and Authentication (25%)

- Set up a new Next.js project using the App Router.
- Implement user authentication using Supabase Auth.
- Create a login and signup page.
- Implement protected routes that are only accessible to authenticated users.

2. Dashboard Implementation (30%)

- Design and implement a main dashboard page with multiple widgets.
- Use Recharts to create at least three different types of charts (e.g., line chart, bar chart, pie chart).
- Implement real-time data updates for at least one widget.

3. Data Management (20%)

- Set up a Supabase database to store relevant data for your dashboard.
- Implement API routes in Next.js to handle data operations (CRUD).
- Create a form to allow users to input new data that updates the dashboard.

4. Responsive Design (15%)

- Ensure the dashboard is fully responsive and works well on mobile devices.
- Implement a collapsible sidebar for navigation on smaller screens.

5. Deployment and Documentation (10%)

- Deploy the application to Vercel.
- Write clear documentation on how to set up and run the project locally.
- Include API documentation for any backend services you create.

Bonus Features (Up to 15% extra credit)

- Implement dark mode toggle for the dashboard.
- Add data export functionality (e.g., CSV download of chart data).
- Implement server-side rendering (SSR) for initial dashboard load.

Evaluation Criteria

- Code quality and organization
- Proper use of Next.js, Recharts, and Supabase
- User experience and interface design
- Effective implementation of data visualization
- Security considerations, especially regarding authentication
- Documentation quality

Submission Guidelines

- Submit your code via a GitHub repository.
- Include a README.md with setup instructions and any necessary documentation.
- Provide a link to the deployed application on Vercel.
- Submit a brief report (max 2 pages) discussing your approach, challenges faced, and potential improvements.