A·SAFE DIGITAL

Technical Test: React Developer with Next.js, TypeScript and Tailwind CSS

Task 1: Authentication with Next.js

- Secure routes using NextAuth.js for authentication.

Task 2: Component Development

UI Components

- Create a UI Kit for the project.
- Implement a theme system with multiple theme options.

Custom Hook

- Create a custom hook for handling complex logic to be reused across the application.

Task 3: Dashboard Generation

- Develop a dashboard to show a summary of information.
- This dashboard should include a graph, you can use libraries such as D3.js, Chart.js, etc.

Task 4: Large Data Set Handling

- Using a set of at least 1000 records, create a view that allows efficient querying and data retrieval.
- Implements an optimal way of displaying information to the user (for example using pagination).

Task 5: Performance Optimisation

- Make the most of the Next.js 14 features by using server-side rendering, server-components and the streaming technique.
- Optimize the application for mobile performance using Google Lighthouse metrics.

Task 6: Testing

- Perform unit testing of the application with jest and/or with React Testing Library.
- Perform E2E testing of the application with cypress.

Task 7: Documentation

- Provide a detailed README with instructions, architecture overview, and any additional information.

Submission Guidelines:

- Share the link to the Git repository containing your code.
- Ensure that your application is deployed, and share the deployment link (e.g., Netlify, Vercel).