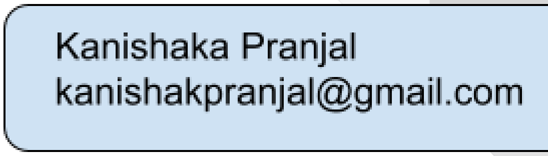
Report on Full Stack Development Intern Assignment: Interactive Dashboard



Kanishaka Pranjal
kanishakpranjal@gmail.com

October 17, 2024

Introduction

The goal of this project was to develop an interactive dashboard application utilizing Next.js, Recharts, and Supabase. The dashboard aimed to provide a user-friendly interface for data visualization, user authentication, and effective data management.

Approach

1. **Framework Selection:**
   a. I chose Next.js for its powerful features such as server-side rendering and static site generation, which enhance performance and SEO.
   b. Integrated Supabase as the backend solution for user authentication and data storage, offering a straightforward way to manage user sessions and database operations.
2. **User Authentication:**
   a. Implemented Supabase Auth to handle user registration and login
   b. Protected routes were established to ensure that only authenticated users could access the dashboard features.
3. **Dashboard Design:**
   a. The main dashboard page was designed to include multiple widgets, each displaying key metrics.
   b. Leveraged **Recharts** for creating interactive charts, including line, bar, and pie charts, providing diverse visualization options
4. **Real-time Data Updates:** Incorporated real-time data updates using Supabase subscriptions, enabling users to see changes reflected immediately on the dashboard.
5. **Responsive Design:**
   a. Focused on ensuring the dashboard was fully responsive, allowing it to adapt seamlessly to different screen sizes.
   b. Implemented a collapsible sidebar for improved navigation on mobile devices.
6. **Deployment:** Deployed the application on **Vercel**, which provides easy integration with Next.js and optimizes performance

Data Management

To facilitate effective data management:

- Set up a Supabase database to store relevant data for the dashboard.
- Developed API routes within Next.js to handle CRUD operations.
- Created a form for users to input new data, which updates the dashboard in real time.

Challenges Faced

1. **Authentication Complexity:**
   - Implementing user authentication posed several challenges, particularly in managing user sessions and protected routes. It required multiple iterations to ensure a seamless user experience.
2. **Real-time Data Integration:**
   - Implementing real-time updates for the dashboard was complex, necessitating a solid understanding of how to leverage Supabase's features effectively. I had to invest time in learning about Supabase subscriptions and managing state updates across components.
3. **Responsive Design Issues:**
   - Achieving a fully responsive design was challenging. Testing on various devices revealed inconsistencies in how elements scaled, leading to several rounds of adjustments to ensure usability on all screen sizes.
4. **API Management:**
   - Managing API routes for CRUD operations required careful planning, especially regarding error handling and data validation. Debugging these routes was time-consuming but ultimately necessary for a reliable application.

Potential Improvements

1. **Advanced Data Visualization:**
   - Future iterations could benefit from more complex visualizations, such as heatmaps or interactive dashboards. This would provide users with deeper insights into their data.
2. **Performance Enhancements:**
   - As user data scales, optimizing the application for performance will be critical. Implementing server-side rendering (SSR) for specific components could enhance initial load times and improve user experience.
3. **User Customization:**
   - Allowing users to customize their dashboards, such as selecting preferred metrics or themes, could increase user engagement and satisfaction.
4. **Data Export Options:**
   - Implementing features for users to export their data (e.g., to CSV or Excel) would enhance the application’s utility, especially for users who require offline data analysis.
5. **Enhanced Documentation:**
   - Improving the documentation for both the frontend and backend components would assist future developers in understanding the codebase. Clear API documentation is especially critical for maintaining and expanding the application.

Conclusion

This project has been an invaluable learning experience in full-stack development, particularly in utilizing Next.js and Supabase effectively. Although challenges arose, each provided opportunities for growth and development. With the potential improvements identified, this interactive dashboard has the capacity to evolve into a more robust and user-friendly application, serving the needs of its users effectively.