AnswersAi Frontend Engineer Take-Home

Assessment Overview

Create 3 screens of a data visualization platform. The focus will be on implementing three key screens and three specific interactions as detailed below. This assessment evaluates your ability to implement modern web interfaces, demonstrate expertise in React and TypeScript, and handle complex interactive behaviors. **Follow the interactions listed out in the Figma.**

Design Specifications: Figma

The Figma design serves as the source of truth for:

- Layout and spacing
- Typography
- Color values
- Component states
- Interactions
- Responsive behavior

Required Screens to Implement

1. Dashboard Screen

- Displays a primary data visualization (e.g., chart, graph, or map).
- Includes a “Variables Panel” with a list of adjustable parameters for the data visualization.
- Features action buttons, including one to “Edit Variables.”

2. Variable editing Slide-Over Card Screen

- Accessed by clicking the “Edit Variables” button on the dashboard.

- A slide-over card overlays the dashboard, allowing users to adjust data visualization parameters.

- Includes a form or interactive elements for editing variables.

3. Details Screen

- Displays information about a specific data point.

- This screen is accessed via hovering over a data point in the dashboard visualization.

- Shows detailed context about the selected data point.

Required Interactions to Implement

1. **Slide-Over Variable editing Card Interaction**

   - Triggered by clicking the “Edit Variables” button on the dashboard.

   - Opens a slide-over card with smooth transitions.

   - Allows the user to modify the variables used in the visualization.

2. **Data Point Hover Interaction**

   - Triggered by hovering over a data point in the main graph visualization.

   - Reveals a card with detailed information about that specific data point.

   - Includes a fade-in animation for the card. (feel free to make this beautiful and intuitive how you see fit).

3. **Variable Selection Interaction**

   - Users can select a variable from the “Variables Panel.”

   - Hovering over a variable displays contextual information to help the user decide whether to include it in the visualization.

   - Includes state management for variables (e.g., active/inactive states).

Technical Requirements

Frontend Stack

- React 18+ with TypeScript

- State management (Redux Toolkit, Zustand, or similar)
- React Router for navigation
- Styled-components or Tailwind CSS
- Firebase Authentication or similar service for auth

Core Requirements

1. Responsive Implementation

- Desktop-first approach as shown in designs.
- Tablet and mobile adaptations with proper breakpoint handling.

2. Interactions

- Smooth transitions for slide-over and hover effects.
- Loading and error states.

4. Component Architecture

- Reusable components such as navigation headers, buttons, tooltips, and form elements.
- Clear separation of concerns for shared and page-specific components.

Evaluation Criteria

1. Design Implementation (40%)

- Pixel-perfect recreation of provided Figma designs.
- Accurate interaction states and animations.
- Responsiveness across devices.

2. Code Quality (35%)

- Clean, maintainable, and well-documented code.
- Effective use of TypeScript for type safety.

- Scalable component and state management architecture.

3. Technical Implementation (25%)

- Smooth and intuitive interactions.
- Robust authentication and routing setup.
- Handling of edge cases (e.g., invalid data, network errors).

Getting Started

1. **Create a new React project:**

```
npm create vite@latest data-viz-platform -- --template react-ts
```

2. **Set up Firebase Authentication (or similar):**
   - Configure Google OAuth and email/password authentication.
3. **Start Implementation in this Order:**
   - Dashboard screen.
   - Slide-over card interaction.
   - Details screen.
   - Variable selection interaction.

Submission Guidelines

1. Create a new GitHub repository.
2. Include a detailed README with:
   - Setup instructions (including environment variables).
   - Features implemented.
   - Technical decisions and trade-offs.
   - Known limitations.
   - Time spent.
   - Local development instructions.

3. Email your repository URL to siddhant@answersai.com.

Additional Notes

- Use dummy data for all content.
- Ensure smooth navigation and state transitions.
- Comment your code thoroughly.
- Focus on core functionality over additional features.
- Test edge cases for robust implementation.

If you have any questions on the Design side, feel free to contact shaurya@answersai.ai

Or for general/engineering questions ask: siddhant@answersai.com