Frontend Developer Technical Assessment: User Management Application

Project Overview

Develop a responsive React application that allows users to view and manage user data with a focus on implementing user limits. This assessment evaluates your ability to create a functional React application with proper state management, API integration, and component structure.

Required Tech Stack

- React
- State Management: Choose either Redux Toolkit Query or Tanstack Query
- UI Library: choose any you are familiar with, for example, Material UI (MUI)
- TypeScript is preferred but not required

Application Structure

1. Users List Page

- Create a table displaying all users fetched from https://dummyjson.com/users?select=id,firstName,lastName,age,email,username,bank. Table should include columns for:
    - First Name,
    - Last Name,
    - Age,
    - Email,
    - Username
- Each row should be clickable and navigate to the user's detail page
- Implement proper loading states and error handling

2. User Details Page

This page consists of three main sections:

2.1. User Details Section

- Fetch and display user information from https://dummyjson.com/users/{userID}?select=id,firstName,lastName,age,email,username,bank.
- Display user's id, first name, last name, age, email, and username in a well-organized layout

- Do not display the bank information in this section
- Add a user avatar by randomly selecting an image from https://picsum.photos/v2/list?page=2&limit=10
- Implement proper loading states and error handling

2.2. User Limits Section

- Fetch limits data from https://dummyjson.com/c/a022-21ef-4179-910f and store in Redux/Query state
- Display the limits in a table with columns for: Limit Period, Limit Type, Limit Value, Limit Value Type, Status, Created Date
- Format all monetary values using the user's bank currency information with 2 decimal places
- Implement proper loading states and error handling

2.3. Add Limit Section

- Create a form with the following fields:
    - Limit Period (dropdown): "daily", "weekly", "monthly"
    - Limit Type (dropdown): "bet", "deposit"
    - Limit Value (number input): Positive numbers only
    - Limit Value Type (radio buttons): "percent", "amount"
    - Status (toggle): true/false
- Add validation for all fields
- When "Save" button is pressed:
    - Generate a unique ID for the new limit
    - Add a "created" field with the current timestamp
- Update the store with the new limit
- Ensure the new limit appears in the limits table

Your submission will be evaluated based on:

- Code quality, organization, and best practices
- Proper implementation of React components and hooks
- API integration
- Error handling and edge cases
- Anything extra, that could demonstrate how your previous experiences can be translated into the work

Submission

Please provide a GitHub repository link with your solution

Good luck!