Context Management Project

User Authentication Context

1. Create User Authentication Context:

Task: Develop a `UserAuthContext` to manage authentication state.
Guidance: Utilize the `createContext` function. Provide a skeleton for the context with an initial state representing non-authenticated users.

2. Context Provider:

Task: Implement a provider for the `UserAuthContext` around your main application.
Guidance: Use the `Context.Provider` component. Ensure it receives and provides the authentication state.

3. Consumer Components:

Task: Identify two components consuming the `UserAuthContext`.
Guidance: Choose components where user authentication affects the rendering. Demonstrate how to conditionally display content based on authentication status.

4. Login and Logout Actions:

Task: Define actions for user login and logout inside the context.
Guidance: Create functions (`login` and `logout`) that modify the authentication state. Provide examples of how these actions are used.

Theme Context

5. Implement Theme Context:

Task: Set up a `ThemeContext` to manage application themes.
Guidance: Use the `createContext` function to create the context. Include light and dark themes in the initial state.

6. Styling with Context:

Task: Apply styles dynamically based on the selected theme.
Guidance: Show how components use the `ThemeContext` to apply styles conditionally. Provide CSS examples for light and dark themes.

Cart Context

7. Create Cart Context:

Task: Establish a `CartContext` to manage shopping cart state.
Guidance: Use `createContext` to set up the context with an initial cart state (e.g., an empty array).

8. Adding and Removing Items:

Task: Implement actions for adding and removing items from the cart.
Guidance: Define functions (`addToCart` and `removeFromCart`) that modify the cart state. Illustrate their usage.