

Topic : Higher Order Component

Assignment: User Authentication Status Checker

Scenario:

You are tasked with building a simple application that checks if a user is authenticated or not before accessing certain features.

The application will use a Higher-Order Component to wrap other components and add authentication logic.

This assignment will help you practice creating and using Higher-Order Components in React with TypeScript.

Requirements:

➤ Project Setup:

- Create a new React project with a TypeScript template and name it AuthCheckerApp.

➤ Component Structure:

○ Create a UserProfile Component:

- This component should display a message like "Welcome, [User's Name]" if the user is authenticated.
- If the user is not authenticated, it should display "Please log in to view this page."

○ Create a Higher-Order Component (HOC) named withAuth:

- This HOC should take a component as an argument and return a new component that checks if the user is authenticated.
- The HOC should accept an isAuthenticated prop to determine if the user is logged in or not.
- If the user is authenticated, the HOC should render the wrapped component.
- If the user is not authenticated, the HOC should render a message like "Access Denied: You must log in to view this page."

- Create an App Component:
 - This component should manage the isAuthenticated state, which will be passed to the withAuth HOC.
 - It should include the UserProfile component wrapped with the withAuth HOC.
 - Provide buttons or links to simulate logging in and out, which will toggle the isAuthenticated state.

Tasks:

1. Create the UserProfile Component:
 - Implement the UserProfile component that displays a welcome message if the user is authenticated.
 2. Create the withAuth HOC:
 - Implement the Higher-Order Component that adds authentication logic to the wrapped component.
 3. Manage Authentication State in App Component:
 - Implement state management in the App component to track whether the user is authenticated.
 - Provide UI elements to toggle the isAuthenticated state.
 4. Expected Outcome:
 - A simple application where:
 - The UserProfile component displays a welcome message when the user is authenticated.
 - The UserProfile component is protected by the withAuth HOC, which blocks access if the user is not authenticated.
 - The App component allows users to simulate logging in and out.
-