Trabajo Práctico integrador grupal de presentación obligatoria.

Continuando con el desarrollo del Trabajo Práctico Integrador, se requiere agregar la siguiente funcionalidad a la aplicación React.

Requerimiento

Incorporar Autenticación que abarque Registro, login y logout simulado de usuarios.
El objetivo es comprender cómo integrar un flujo básico de registro → login → logout utilizando únicamente React, Context API/Redux, React Router y localStorage (sin backend).

Autenticación de Usuario (Simulada)

Registro de usuario.

1 Crear una vista “Registro” para el registro del usuario.

2. Formulario mínimo, debe contener: correo electrónico, contraseña y confirmar contraseña (se aceptan campos extras, por ejemplo. nombre).

3. Validaciones en el front-end:

- Correo con formato válido.
- Contraseña ≥ 6 caracteres.
- Contraseña y confirmación deben coincidir.

4. Al enviar y pasar validaciones, guardar el usuario en localStorage (clave sugerida: "users"; puede ser un array de objetos).

5. Mostrar notificación de éxito y redirigir automáticamente a la ruta “Login” (/login).

Inicio de sesión de usuario

1. Crear una vista “Login”.

2. Formulario con componentes para ingresar correo y contraseña.

3. Al enviar, comparar contra los datos almacenados en localStorage:

- Si coinciden, crear una clave "sessionUser" (o similar) en localStorage con la información básica del usuario (al menos el correo).
- Redirigir a la Home (/) y mostrar mensaje “Bienvenido, <correo>” en el Nav.
- Si no coinciden, mostrar error (“Credenciales inválidas”).

Mantener sesión activa

- Mientras exista "sessionUser" en localStorage, la Home y el resto de páginas privadas deben ser accesibles.

- Ante refresco de página, leer "sessionUser" para rehidratar el contexto/Redux y conservar el estado de usuario logueado.

Rehidratar el estado de usuario logueado quiere decir que cuando un usuario inicia sesión, sus datos se guardan en el localStorage (clave: "sessionUser"). Esto permite que, aunque se recargue la página, la aplicación pueda:

Leer esos datos al inicio.

Reinicializar el estado global de Redux con la información del usuario.

Evitar que el usuario tenga que volver a loguearse.

En initialState del slice creado se debe recuperar el usuario desde localStorage al iniciar:

user: getSession()

Cierre de sesión

- Incluir en el Nav un botón “Cerrar sesión”.

- Al hacer clic:

- Eliminar "sessionUser" del localStorage.
- Limpiar el estado de usuario en el contexto/Redux.
- Redirigir a la ruta “Login”.

Rutas protegidas

- Proteger las rutas internas (Home, Favoritos, Detalle, etc.) mediante un componente personalizado PrivateRoute.jsx

- Si el usuario no está autenticado, redirigir siempre a /login.

React Router, en App.jsx

- Definir rutas públicas (register, login) y privadas (home, favoritos, detalle).

- Implementar el componente <PrivateRoute> que verifique la sesión.