DAY 27 REACT ROUTER

1. Job Portal Web App

Objective:

Build a full-featured Job Portal for users to browse, apply, and manage jobs — reinforcing **dynamic routes, nested routes, and protected routes**.

Requirements:

1. **Public Routes:**
    - / → Home (Intro, featured jobs)
    - /jobs → All Jobs List (fetch data dynamically)
    - /jobs/:jobId → Job Details Page (**useParams**)
    - /about → About Page
    - /contact → Contact Page
2. **Private Routes (Protected):**
    - /dashboard → Requires login (use **ProtectedRoute**)
    - /dashboard/applications → List of user's applied jobs (**Nested route under Dashboard**)
    - /dashboard/profile → User profile (update details)
3. **Router Features to Include:**
    - Use NavLink to highlight active nav item.
    - Use useNavigate() after login to redirect to /dashboard.
    - Add NotFound (*) page for unknown URLs.
    - Use React.lazy() + Suspense for lazy loading JobDetails and Dashboard.
    - Show a breadcrumb bar using useLocation() to indicate the current path (e.g. Home / Jobs / Job Detail).
4. **Bonus:**
    - Add query params for job filters (e.g. /jobs?location=remote&type=full-time).
    - Use Outlet for dashboard's nested routes.

Concepts Reinforced:
Routing hierarchy, dynamic params, protected routes, useNavigate, useParams, NavLink, nested routing, lazy loading.

---

2. E-Commerce Store

Objective:

Simulate a real e-commerce experience using **dynamic product pages, nested routes, and conditional navigation**.

Requirements:

1. **Main Routes:**
   - / → Home with featured products
   - /products → Product listing page
   - /products/:id → Dynamic Product Detail page (**useParams**)
   - /cart → Shopping cart (Protected)
   - /checkout → Checkout page (Protected)
   - /profile → User account (Nested routes: /profile/orders, /profile/settings)
2. **Router Concepts Used:**
   - BrowserRouter, Routes, Route
   - NavLink in navbar with active states
   - useNavigate for redirect after checkout
   - Outlet for /profile nested pages
   - useLocation for showing success messages based on navigation state (e.g. "Order placed successfully!")
   - Lazy load product details page
3. **Bonus:**
   - Search bar updates the route with query params (/products?category=electronics).
   - Global 404 NotFound route.

Concepts Reinforced:
Dynamic routes, nested routing, redirects, protected routes, query params, route-based state management.

---

3. Movie Review & Rating App

Objective:

Users can browse movies, view details, and add reviews — demonstrating **nested, dynamic, and conditional routes**.

Requirements:

1. **Public Routes:**
   - / → Home page
   - /movies → Movie listing page
   - /movies/:id → Movie detail (title, trailer, cast)
   - /about → About app
2. **Protected Routes:**
   - /dashboard → User dashboard (only logged-in users)
     - /dashboard/reviews → Manage submitted reviews
     - /dashboard/settings → Profile settings (use Outlet)
3. **Router Features to Cover:**
   - NavLink with active class for navigation.
   - useParams() → Get movie ID.
   - useNavigate() → Redirect after submitting a review.
   - useLocation() → Pass movie info when redirecting from list to detail.
   - Lazy load movie details (suspense fallback "Loading movie…").
   - Handle 404 for invalid movie IDs.

Concepts Reinforced:
Dynamic nested routes, parameter handling, lazy loading, route guards, redirect logic, and conditional rendering.

---

4. Hotel Booking System

Objective:

A hotel booking application to browse hotels, check details, and manage bookings — showcasing **nested routes, layouts, and route-based redirection.**

Requirements:

1. **Main Routes:**
    - / → Landing page (Hero section, CTA to view hotels)
    - /hotels → List all hotels
    - /hotels/:hotelId → Hotel details with gallery and booking form
    - /login → Login page
    - /register → Register page
2. **Protected (Authenticated) Routes:**
    - /dashboard → Layout with sidebar (contains Outlet)
        - /dashboard/bookings → All user bookings
        - /dashboard/favorites → Saved hotels
        - /dashboard/settings → Account settings
3. **Router Concepts:**
    - Outlet for nested dashboard layout.
    - useNavigate() → Redirect to /dashboard/bookings after booking success.
    - NavLink for active tab styling in sidebar.
    - useLocation() → Preserve "back to hotel list" link.
    - Lazy load /dashboard module.
    - Handle expired session using a protected route wrapper.
4. **Bonus:**
    - Include search query params (/hotels?city=chennai&price=low).

Concepts Reinforced:
Nested routes, protected routing, query parameters, layout composition, route transitions, lazy imports.

---

5. Learning Management System (LMS)

Objective:

An education platform with student and admin roles to explore **advanced nested and protected routing**.

Requirements:

1. **Public Routes:**

    - / → Home page
    - /courses → List of available courses
    - /courses/:id → Course details (lessons, instructor info)
    - /about, /contact → Info pages
2. **Protected Routes:**
    - /dashboard → Different for Student and Admin
        - Student routes:
            - /dashboard/my-courses
            - /dashboard/assignments
        - Admin routes:
            - /dashboard/manage-users
            - /dashboard/manage-courses
3. **Router Features:**
    - Role-based protected routes (student vs admin).
    - Outlet for different dashboard layouts.
    - useNavigate() for redirect after login or logout.
    - useParams() for course ID.
    - useLocation() to show messages (e.g., “You’ve successfully enrolled!”)
    - Lazy load course detail pages.
4. **Bonus:**
    - 404 and custom Unauthorized page.
    - Breadcrumb navigation for deeper routes.

Concepts Reinforced:
Full control over route guards, nested layouts, role-based access, dynamic params, and conditional redirects.
