**Exercise 1: Basic Routing**

**Problem**

Create a React application with three pages: Home, About, and Contact. Use React Router to navigate between these pages using links in the navigation bar.

**Exercise 2: Nested Routes**

**Problem**

Add a nested route to the About page. For example, /about/team should display a "Team Page".

**Exercise 3: Dynamic Routes**

**Problem**

Create a dynamic route for viewing a user's profile by ID. For example, /user/1 should display "User 1's Profile".

**Exercise 4: Redirects**

**Problem**

Create a route that redirects /old-home to /.

**Exercise 5: Passing State Between Routes**

**Problem**

Create a route that passes state to the destination route. For example, clicking on a link should pass a message to the next page.

**Exercise 6: 404 Not Found Page**

**Problem**

Create a fallback route that displays a "404 Not Found" page when the user navigates to a non-existent route.

**Exercise 7: Private Routes**

**Problem**

Create a private route that redirects users to a login page if they are not authenticated.

**Exercise 8: Multi-Level Nested Routing**

**Problem**

Create a React app with the following structure:

* /: Home Page
* /about: About Page with sub-pages /about/team and /about/history.
* /contact: Contact Page.

Each sub-page should have its own content.

**Exercise 9: Product Catalog with Dynamic Routes**

**Problem**

Create a product catalog app:

* /: Home Page.
* /products: Product List Page with links to individual products.
* /products/:id: Product Details Page.

**Exercise 10: User Dashboard with Private Route**

**Problem**

Create a dashboard app with the following structure:

* /: Home Page.
* /dashboard: Dashboard Page (Private).
* /profile: Profile Page (Private).  
  Redirect unauthenticated users to /login.

**Exercise 11: Blog Application with Nested Routes**

**Problem**

Create a blog application:

* /: Home Page.
* /blog: Blog List Page.
* /blog/:postId: Individual Blog Post Page.
* /blog/:postId/comments: Comments for the Blog Post.

**Exercise 12: E-Commerce Platform with Categories and Product Details**

**Problem**

Create an e-commerce app with the following structure:

* /: Home Page
* /categories: Categories Page with a list of categories (e.g., Electronics, Fashion, Home).
* /categories/:categoryId: Products within a category.
* /products/:productId: Product Details Page.
* /cart: Shopping Cart Page.

**Exercise 13: Multi-Step Form Application**

**Problem**

Build a multi-step form application:

1. /: Home Page
2. /form: Form Start Page
3. /form/step1: Step 1 (Personal Info)
4. /form/step2: Step 2 (Address Info)
5. /form/step3: Step 3 (Review and Submit)

**Exercise 14: University Application with Nested Routes**

**Problem**

Create a university app with:

* /: Home Page
* /departments: Departments Page
* /departments/:deptId: Courses in a Department
* /departments/:deptId/courses/:courseId: Course Details Page
* /about: About Page

**Exercise 15: E-Learning Platform**

Create an application for an E-Learning Platform that includes the following components:

* Home: The main landing page.
* Categories: A page listing different course categories (e.g., Programming, Design, Marketing).
* CategoryDetails: A page listing courses under a specific category.
* CourseDetails: A detailed view of a selected course.
* Enroll: A page to enroll in a selected course.

**Routes**:

1. / - Displays the Home component.
2. /categories - Displays the Categories component.
3. /categories/:categoryId - Displays the CategoryDetails component.
4. /categories/:categoryId/courses/:courseId - Displays the CourseDetails component.
5. /enroll - Displays the Enroll component.

**Exercise 16: E-Commerce Website with Nested Routes**

Create an application for an E-Commerce Website with the following components:

* Home: A landing page.
* Shop: Displays a list of product categories.
* Category: Displays the products in a specific category.
* ProductDetails: Displays detailed information about a selected product.
* Cart: A shopping cart page to review items before checkout.
* Checkout: A page to complete the purchase.

**Routes**:

1. / - Displays the Home component.
2. /shop - Displays the Shop component.
3. /shop/:categoryId - Displays the Category component.
4. /shop/:categoryId/:productId - Displays the ProductDetails component.
5. /cart - Displays the Cart component.
6. /checkout - Displays the Checkout component.

**Exercise 17: HR Management System with Dynamic Routes**

Build an HR Management System with the following components:

* Dashboard: Displays HR statistics (e.g., total employees, open positions).
* Employees: Lists all employees.
* EmployeeDetails: Displays detailed information about a selected employee.
* Departments: Lists all company departments.
* DepartmentDetails: Displays employees in a specific department.

**Routes**:

1. /dashboard - Displays the Dashboard component.
2. /employees - Displays the Employees component.
3. /employees/:employeeId - Displays the EmployeeDetails component.
4. /departments - Displays the Departments component.
5. /departments/:departmentId - Displays the DepartmentDetails component.

**Exercise 18: Blog Platform with Nested and Parameterized Routes**

Create a blogging platform with the following components:

* Home: Displays a list of blog categories.
* Category: Displays a list of blog posts under a category.
* PostDetails: Displays a single blog post with its details.
* Author: Displays details about the author of a post.
* About: Displays information about the blog platform.

**Routes**:

1. / - Displays the Home component.
2. /category/:categoryId - Displays the Category component.
3. /category/:categoryId/post/:postId - Displays the PostDetails component.
4. /author/:authorId - Displays the Author component.
5. /about - Displays the About component.

**Exercise 19: Travel Management System**

Develop a travel management system with these components:

* Home: Landing page showcasing destinations.
* Destinations: Lists all destinations.
* DestinationDetails: Displays information about a specific destination.
* Booking: Allows users to book a destination.
* UserDashboard: Displays all bookings for the user.

**Routes**:

1. / - Displays the Home component.
2. /destinations - Displays the Destinations component.
3. /destinations/:destinationId - Displays the DestinationDetails component.
4. /booking - Displays the Booking component.
5. /dashboard - Displays the UserDashboard component.

**Exercise 20: University Portal with Multiple Features**

Create a university portal with the following components:

* Home: Displays the university introduction.
* Admissions: Contains information about admission processes.
* Programs: Lists all academic programs.
* ProgramDetails: Displays details about a specific academic program.
* Faculty: Lists all faculty members.
* FacultyDetails: Displays information about a specific faculty member.

**Routes**:

1. / - Displays the Home component.
2. /admissions - Displays the Admissions component.
3. /programs - Displays the Programs component.
4. /programs/:programId - Displays the ProgramDetails component.
5. /faculty - Displays the Faculty component.
6. /faculty/:facultyId - Displays the FacultyDetails component.