**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

function Home() {

return <h1>Home Page</h1>;

}

function About() {

return <h1>About Page</h1>;

}

function Contact() {

return <h1>Contact Page</h1>;

}

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/about">About</Link> |{" "}

<Link to="/contact">Contact</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about" element={<About />} />

<Route path="/contact" element={<Contact />} />

</Routes>

</Router>

);

}

**Exercise 2: Nested Routes**

**Problem**

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

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

function Home() {

return <h1>Home Page</h1>;

}

function About() {

return (

<div>

<h1>About Page</h1>

<Link to="team">Team</Link>

<Routes>

<Route path="team" element={<h2>Team Page</h2>} />

</Routes>

</div>

);

}

function Contact() {

return <h1>Contact Page</h1>;

}

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/about">About</Link> |{" "}

<Link to="/contact">Contact</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about/\*" element={<About />} />

<Route path="/contact" element={<Contact />} />

</Routes>

</Router>

);

}

**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".

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link, useParams } from "react-router-dom";

function UserProfile() {

const { id } = useParams();

return <h1>User {id}'s Profile</h1>;

}

export default function App() {

return (

<Router>

<nav>

<Link to="/user/1">User 1</Link> | <Link to="/user/2">User 2</Link>

</nav>

<Routes>

<Route path="/user/:id" element={<UserProfile />} />

</Routes>

</Router>

);

}

**Exercise 4: Redirects**

**Problem**

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

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Navigate } from "react-router-dom";

function Home() {

return <h1>Home Page</h1>;

}

export default function App() {

return (

<Router>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/old-home" element={<Navigate to="/" replace />} />

</Routes>

</Router>

);

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link, useLocation } from "react-router-dom";

function MessagePage() {

const location = useLocation();

const message = location.state?.message || "No message";

return <h1>{message}</h1>;

}

export default function App() {

return (

<Router>

<nav>

<Link to="/message" state={{ message: "Hello from Home Page!" }}>

Send Message

</Link>

</nav>

<Routes>

<Route path="/message" element={<MessagePage />} />

</Routes>

</Router>

);

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

function Home() {

return <h1>Home Page</h1>;

}

function NotFound() {

return <h1>404 - Page Not Found</h1>;

}

export default function App() {

return (

<Router>

<Routes>

<Route path="/" element={<Home />} />

<Route path="\*" element={<NotFound />} />

</Routes>

</Router>

);

}

**Exercise 7: Private Routes**

**Problem**

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

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Navigate } from "react-router-dom";

const isAuthenticated = false;

function PrivateRoute({ children }) {

return isAuthenticated ? children : <Navigate to="/login" replace />;

}

function Dashboard() {

return <h1>Dashboard - Private Page</h1>;

}

function Login() {

return <h1>Login Page</h1>;

}

export default function App() {

return (

<Router>

<Routes>

<Route path="/dashboard" element={<PrivateRoute><Dashboard /></PrivateRoute>} />

<Route path="/login" element={<Login />} />

</Routes>

</Router>

);

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import About from "./About";

import Contact from "./Contact";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/about">About</Link> | <Link to="/contact">Contact</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/about/\*" element={<About />} />

<Route path="/contact" element={<Contact />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Home Page</h1>;

}

// About.js

import React from "react";

import { Routes, Route, Link } from "react-router-dom";

export default function About() {

return (

<div>

<h1>About Page</h1>

<Link to="team">Team</Link> | <Link to="history">History</Link>

<Routes>

<Route path="team" element={<h2>Our Team</h2>} />

<Route path="history" element={<h2>Our History</h2>} />

</Routes>

</div>

);

}

// Contact.js

export default function Contact() {

return <h1>Contact Page</h1>;

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Products from "./Products";

import ProductDetails from "./ProductDetails";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/products">Products</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/products" element={<Products />} />

<Route path="/products/:id" element={<ProductDetails />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Welcome to the Product Catalog</h1>;

}

// Products.js

import React from "react";

import { Link } from "react-router-dom";

export default function Products() {

const productList = [

{ id: 1, name: "Product 1" },

{ id: 2, name: "Product 2" },

{ id: 3, name: "Product 3" },

];

return (

<div>

<h1>Products</h1>

<ul>

{productList.map((product) => (

<li key={product.id}>

<Link to={`/products/${product.id}`}>{product.name}</Link>

</li>

))}

</ul>

</div>

);

}

// ProductDetails.js

import React from "react";

import { useParams } from "react-router-dom";

export default function ProductDetails() {

const { id } = useParams();

return <h1>Details of Product {id}</h1>;

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Navigate } from "react-router-dom";

import Home from "./Home";

import Login from "./Login";

import Dashboard from "./Dashboard";

import Profile from "./Profile";

const isAuthenticated = true;

function PrivateRoute({ children }) {

return isAuthenticated ? children : <Navigate to="/login" replace />;

}

export default function App() {

return (

<Router>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/login" element={<Login />} />

<Route

path="/dashboard"

element={<PrivateRoute><Dashboard /></PrivateRoute>}

/>

<Route

path="/profile"

element={<PrivateRoute><Profile /></PrivateRoute>}

/>

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Home Page</h1>;

}

// Login.js

export default function Login() {

return <h1>Login Page</h1>;

}

// Dashboard.js

export default function Dashboard() {

return <h1>Dashboard Page</h1>;

}

// Profile.js

export default function Profile() {

return <h1>Profile Page</h1>;

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Blog from "./Blog";

import BlogPost from "./BlogPost";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/blog">Blog</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/blog/\*" element={<Blog />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Home Page</h1>;

}

// Blog.js

import React from "react";

import { Routes, Route, Link } from "react-router-dom";

import BlogPost from "./BlogPost";

export default function Blog() {

const posts = [

{ id: 1, title: "First Blog Post" },

{ id: 2, title: "Second Blog Post" },

];

return (

<div>

<h1>Blog</h1>

<ul>

{posts.map((post) => (

<li key={post.id}>

<Link to={`/blog/${post.id}`}>{post.title}</Link>

</li>

))}

</ul>

<Routes>

<Route path=":postId/\*" element={<BlogPost />} />

</Routes>

</div>

);

}

// BlogPost.js

import React from "react";

import { useParams, Routes, Route, Link } from "react-router-dom";

export default function BlogPost() {

const { postId } = useParams();

return (

<div>

<h1>Blog Post {postId}</h1>

<Link to="comments">View Comments</Link>

<Routes>

<Route path="comments" element={<h2>Comments for Post {postId}</h2>} />

</Routes>

</div>

);

}

**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.

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Categories from "./Categories";

import Products from "./Products";

import ProductDetails from "./ProductDetails";

import Cart from "./Cart";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/categories">Categories</Link> | <Link to="/cart">Cart</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/categories" element={<Categories />} />

<Route path="/categories/:categoryId" element={<Products />} />

<Route path="/products/:productId" element={<ProductDetails />} />

<Route path="/cart" element={<Cart />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Welcome to Our E-Commerce Platform</h1>;

}

// Categories.js

import React from "react";

import { Link } from "react-router-dom";

export default function Categories() {

const categories = [

{ id: 1, name: "Electronics" },

{ id: 2, name: "Fashion" },

{ id: 3, name: "Home" },

];

return (

<div>

<h1>Categories</h1>

<ul>

{categories.map((category) => (

<li key={category.id}>

<Link to={`/categories/${category.id}`}>{category.name}</Link>

</li>

))}

</ul>

</div>

);

}

// Products.js

import React from "react";

import { Link, useParams } from "react-router-dom";

export default function Products() {

const { categoryId } = useParams();

const products = [

{ id: 1, name: "Laptop", categoryId: "1" },

{ id: 2, name: "Smartphone", categoryId: "1" },

{ id: 3, name: "Shoes", categoryId: "2" },

];

const filteredProducts = products.filter((p) => p.categoryId === categoryId);

return (

<div>

<h1>Products in Category {categoryId}</h1>

<ul>

{filteredProducts.map((product) => (

<li key={product.id}>

<Link to={`/products/${product.id}`}>{product.name}</Link>

</li>

))}

</ul>

</div>

);

}

// ProductDetails.js

import React from "react";

import { useParams } from "react-router-dom";

export default function ProductDetails() {

const { productId } = useParams();

return <h1>Details of Product {productId}</h1>;

}

// Cart.js

export default function Cart() {

return <h1>Your Shopping Cart</h1>;

}

**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)

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import FormStart from "./FormStart";

import Step1 from "./Step1";

import Step2 from "./Step2";

import Step3 from "./Step3";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/form">Start Form</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/form" element={<FormStart />} />

<Route path="/form/step1" element={<Step1 />} />

<Route path="/form/step2" element={<Step2 />} />

<Route path="/form/step3" element={<Step3 />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Welcome to the Multi-Step Form App</h1>;

}

// FormStart.js

import React from "react";

import { Link } from "react-router-dom";

export default function FormStart() {

return (

<div>

<h1>Form Start</h1>

<Link to="/form/step1">Begin Step 1</Link>

</div>

);

}

// Step1.js

import React from "react";

import { Link } from "react-router-dom";

export default function Step1() {

return (

<div>

<h1>Step 1: Personal Info</h1>

<form>

<input type="text" placeholder="Name" />

<input type="email" placeholder="Email" />

</form>

<Link to="/form/step2">Next</Link>

</div>

);

}

// Step2.js

import React from "react";

import { Link } from "react-router-dom";

export default function Step2() {

return (

<div>

<h1>Step 2: Address Info</h1>

<form>

<input type="text" placeholder="Address" />

<input type="text" placeholder="City" />

</form>

<Link to="/form/step3">Next</Link>

</div>

);

}

// Step3.js

import React from "react";

export default function Step3() {

return (

<div>

<h1>Step 3: Review and Submit</h1>

<p>Review your details and submit the form!</p>

</div>

);

}

**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

**Solution**

// App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Departments from "./Departments";

import Courses from "./Courses";

import CourseDetails from "./CourseDetails";

import About from "./About";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/departments">Departments</Link> |{" "}

<Link to="/about">About</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/departments" element={<Departments />} />

<Route path="/departments/:deptId" element={<Courses />} />

<Route path="/departments/:deptId/courses/:courseId" element={<CourseDetails />} />

<Route path="/about" element={<About />} />

</Routes>

</Router>

);

}

// Home.js

export default function Home() {

return <h1>Welcome to the University Portal</h1>;

}

// Departments.js

import React from "react";

import { Link } from "react-router-dom";

export default function Departments() {

const departments = [

{ id: "cs", name: "Computer Science" },

{ id: "math", name: "Mathematics" },

{ id: "eng", name: "English" },

];

return (

<div>

<h1>Departments</h1>

<ul>

{departments.map((dept) => (

<li key={dept.id}>

<Link to={`/departments/${dept.id}`}>{dept.name}</Link>

</li>

))}

</ul>

</div>

);

}

// Courses.js

import React from "react";

import { Link, useParams } from "react-router-dom";

export default function Courses() {

const { deptId } = useParams();

const courses = [

{ id: "cs101", name: "Intro to Programming", deptId: "cs" },

{ id: "cs102", name: "Data Structures", deptId: "cs" },

{ id: "math101", name: "Calculus I", deptId: "math" },

{ id: "eng101", name: "English Literature", deptId: "eng" },

];

const filteredCourses = courses.filter((course) => course.deptId === deptId);

return (

<div>

<h1>Courses in {deptId.toUpperCase()}</h1>

<ul>

{filteredCourses.map((course) => (

<li key={course.id}>

<Link to={`/departments/${deptId}/courses/${course.id}`}>{course.name}</Link>

</li>

))}

</ul>

</div>

);

}

// CourseDetails.js

import React from "react";

import { useParams } from "react-router-dom";

export default function CourseDetails() {

const { deptId, courseId } = useParams();

return (

<div>

<h1>Details of {courseId.toUpperCase()} in {deptId.toUpperCase()}</h1>

<p>Here you can add detailed information about the course.</p>

</div>

);

}

// About.js

export default function About() {

return <h1>About the University</h1>;

}

**Explanation**

1. **Routes and Navigation**:
   * The App.js file defines routes for the homepage (/), departments (/departments), courses (/departments/:deptId), course details (/departments/:deptId/courses/:courseId), and about page (/about).
2. **Dynamic Routing with Parameters**:
   * Departments.js dynamically generates links for departments.
   * Courses.js uses the deptId parameter to filter and display courses belonging to a department.
   * CourseDetails.js uses both deptId and courseId to display specific course information.
3. **Component Details**:
   * Home and About are static components for simplicity.
   * Departments, Courses, and CourseDetails demonstrate dynamic routing with React Router.

**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.

This application includes the following components:

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

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Categories from "./Categories";

import CategoryDetails from "./CategoryDetails";

import CourseDetails from "./CourseDetails";

import Enroll from "./Enroll";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/categories">Categories</Link> |{" "}

<Link to="/enroll">Enroll</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/categories" element={<Categories />} />

<Route path="/categories/:categoryId" element={<CategoryDetails />} />

<Route path="/categories/:categoryId/courses/:courseId" element={<CourseDetails />} />

<Route path="/enroll" element={<Enroll />} />

</Routes>

</Router>

);

}

**2. Home.js (Landing Page)**

export default function Home() {

return <h1>Welcome to the E-Learning Platform</h1>;

}

**3. Categories.js (List of Categories)**

import React from "react";

import { Link } from "react-router-dom";

export default function Categories() {

const categories = [

{ id: "programming", name: "Programming" },

{ id: "design", name: "Design" },

{ id: "marketing", name: "Marketing" },

];

return (

<div>

<h1>Course Categories</h1>

<ul>

{categories.map((category) => (

<li key={category.id}>

<Link to={`/categories/${category.id}`}>{category.name}</Link>

</li>

))}

</ul>

</div>

);

}

**4. CategoryDetails.js (Courses in a Specific Category)**

import React from "react";

import { Link, useParams } from "react-router-dom";

export default function CategoryDetails() {

const { categoryId } = useParams();

const courses = [

{ id: "course1", name: "Intro to Programming", categoryId: "programming" },

{ id: "course2", name: "Advanced JavaScript", categoryId: "programming" },

{ id: "course3", name: "Graphic Design Basics", categoryId: "design" },

{ id: "course4", name: "Marketing Strategies", categoryId: "marketing" },

];

const filteredCourses = courses.filter((course) => course.categoryId === categoryId);

return (

<div>

<h2>Courses in {categoryId}</h2>

<ul>

{filteredCourses.map((course) => (

<li key={course.id}>

<Link to={`/categories/${categoryId}/courses/${course.id}`}>

{course.name}

</Link>

</li>

))}

</ul>

</div>

);

}

**5. CourseDetails.js (Details of a Specific Course)**

import React from "react";

import { useParams } from "react-router-dom";

export default function CourseDetails() {

const { categoryId, courseId } = useParams();

return (

<div>

<h1>Details of {courseId} in {categoryId}</h1>

<p>

Here you can add more detailed information about the course such as

course syllabus, prerequisites, and instructor details.

</p>

</div>

);

}

**6. Enroll.js (Enroll in a Course)**

export default function Enroll() {

return (

<div>

<h1>Enroll in a Course</h1>

<p>To enroll, please select a course and complete the enrollment form.</p>

</div>

);

}

**Explanation:**

1. **App Component (App.js)**:
   * The App component contains a simple navigation bar with Link components to navigate between different pages. The Routes define the paths for each component (Home, Categories, CategoryDetails, CourseDetails, and Enroll).
2. **Categories Component (Categories.js)**:
   * Displays a list of categories (Programming, Design, Marketing). Each category links to its detailed page, which lists the courses in that category.
3. **CategoryDetails Component (CategoryDetails.js)**:
   * Displays a list of courses specific to the category passed through the URL. It filters the courses by category (categoryId) using the useParams hook from react-router-dom.
4. **CourseDetails Component (CourseDetails.js)**:
   * Displays detailed information about a selected course based on the courseId and categoryId passed through the URL.
5. **Enroll Component (Enroll.js)**:
   * A placeholder for the enroll page. Here, users can enroll in a course (could be expanded with form inputs in the future).

**How to Run:**

1. Create a React app using create-react-app or any other method.
2. Replace the contents of App.js, Home.js, Categories.js, CategoryDetails.js, CourseDetails.js, and Enroll.js with the code provided above.
3. Run npm start to see the application in action.

**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.

This application consists of the following components:

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

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Shop from "./Shop";

import Category from "./Category";

import ProductDetails from "./ProductDetails";

import Cart from "./Cart";

import Checkout from "./Checkout";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> | <Link to="/shop">Shop</Link> |{" "}

<Link to="/cart">Cart</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/shop" element={<Shop />} />

<Route path="/shop/:categoryId" element={<Category />} />

<Route path="/shop/:categoryId/:productId" element={<ProductDetails />} />

<Route path="/cart" element={<Cart />} />

<Route path="/checkout" element={<Checkout />} />

</Routes>

</Router>

);

}

**2. Home.js (Landing Page)**

export default function Home() {

return <h1>Welcome to the E-Commerce Website</h1>;

}

**3. Shop.js (List of Categories)**

import React from "react";

import { Link } from "react-router-dom";

export default function Shop() {

const categories = [

{ id: "electronics", name: "Electronics" },

{ id: "fashion", name: "Fashion" },

{ id: "home-appliances", name: "Home Appliances" },

];

return (

<div>

<h1>Shop by Category</h1>

<ul>

{categories.map((category) => (

<li key={category.id}>

<Link to={`/shop/${category.id}`}>{category.name}</Link>

</li>

))}

</ul>

</div>

);

}

**4. Category.js (Products in a Specific Category)**

import React from "react";

import { useParams, Link } from "react-router-dom";

export default function Category() {

const { categoryId } = useParams();

const products = [

{ id: "product1", name: "Smartphone", categoryId: "electronics" },

{ id: "product2", name: "Laptop", categoryId: "electronics" },

{ id: "product3", name: "T-shirt", categoryId: "fashion" },

{ id: "product4", name: "Jeans", categoryId: "fashion" },

{ id: "product5", name: "Washing Machine", categoryId: "home-appliances" },

];

const filteredProducts = products.filter((product) => product.categoryId === categoryId);

return (

<div>

<h2>Products in {categoryId}</h2>

<ul>

{filteredProducts.map((product) => (

<li key={product.id}>

<Link to={`/shop/${categoryId}/${product.id}`}>{product.name}</Link>

</li>

))}

</ul>

</div>

);

}

**5. ProductDetails.js (Detailed Information about a Product)**

import React from "react";

import { useParams } from "react-router-dom";

export default function ProductDetails() {

const { categoryId, productId } = useParams();

return (

<div>

<h1>{productId} in {categoryId}</h1>

<p>

Here you can add detailed information about the product, such as description, price, images, and reviews.

</p>

</div>

);

}

**6. Cart.js (Shopping Cart Page)**

import React from "react";

export default function Cart() {

return (

<div>

<h1>Your Shopping Cart</h1>

<p>Here you can view the products you've added to the cart.</p>

<ul>

<li>Smartphone</li>

<li>Jeans</li>

</ul>

<Link to="/checkout">Proceed to Checkout</Link>

</div>

);

}

**7. Checkout.js (Checkout Page)**

export default function Checkout() {

return (

<div>

<h1>Checkout</h1>

<p>Enter your payment details and shipping information to complete the purchase.</p>

</div>

);

}

**Explanation:**

1. **App Component (App.js)**:
   * The App component contains a simple navigation bar with Link components to navigate between different pages. The Routes define the paths for each component (Home, Shop, Category, ProductDetails, Cart, and Checkout).
2. **Home Component (Home.js)**:
   * The Home component is a simple welcome page.
3. **Shop Component (Shop.js)**:
   * Displays a list of categories (Electronics, Fashion, Home Appliances). Each category links to a page that shows products in that category.
4. **Category Component (Category.js)**:
   * Displays products that belong to a specific category. It uses useParams to get the categoryId and filters the products based on that.
5. **ProductDetails Component (ProductDetails.js)**:
   * Displays detailed information about a specific product. It uses useParams to get both the categoryId and productId from the URL.
6. **Cart Component (Cart.js)**:
   * Displays a list of products in the shopping cart. It includes a link to the Checkout page where the user can complete the purchase.
7. **Checkout Component (Checkout.js)**:
   * The Checkout component contains a simple placeholder for entering payment and shipping information to complete the purchase.

**How to Run:**

1. Create a React app using create-react-app or any other method.
2. Replace the contents of App.js, Home.js, Shop.js, Category.js, ProductDetails.js, Cart.js, and Checkout.js with the code provided above.
3. Run npm start to see the application in action.

**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.

In this solution, we'll build an HR Management System with the specified components and dynamic routes. We'll structure the project with the following components:

1. **Dashboard**: Displays HR statistics like total employees and open positions.
2. **Employees**: Lists all employees.
3. **EmployeeDetails**: Displays detailed information about a specific employee.
4. **Departments**: Lists all departments.
5. **DepartmentDetails**: Displays a list of employees belonging to a specific department.

We'll implement the following 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.

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Dashboard from "./Dashboard";

import Employees from "./Employees";

import EmployeeDetails from "./EmployeeDetails";

import Departments from "./Departments";

import DepartmentDetails from "./DepartmentDetails";

export default function App() {

return (

<Router>

<nav>

<Link to="/dashboard">Dashboard</Link> |

<Link to="/employees">Employees</Link> |

<Link to="/departments">Departments</Link>

</nav>

<Routes>

<Route path="/dashboard" element={<Dashboard />} />

<Route path="/employees" element={<Employees />} />

<Route path="/employees/:employeeId" element={<EmployeeDetails />} />

<Route path="/departments" element={<Departments />} />

<Route path="/departments/:departmentId" element={<DepartmentDetails />} />

</Routes>

</Router>

);

}

**2. Dashboard.js (HR Statistics)**

import React from "react";

const statistics = {

totalEmployees: 100,

openPositions: 5,

};

export default function Dashboard() {

return (

<div>

<h1>HR Dashboard</h1>

<p>Total Employees: {statistics.totalEmployees}</p>

<p>Open Positions: {statistics.openPositions}</p>

</div>

);

}

**3. Employees.js (List of Employees)**

import React from "react";

import { Link } from "react-router-dom";

const employees = [

{ id: 1, name: "John Doe", position: "Software Engineer" },

{ id: 2, name: "Jane Smith", position: "HR Manager" },

{ id: 3, name: "Alice Johnson", position: "Product Manager" },

{ id: 4, name: "Bob Brown", position: "UX Designer" },

// Add more employees as needed

];

export default function Employees() {

return (

<div>

<h1>Employees</h1>

<ul>

{employees.map((employee) => (

<li key={employee.id}>

<Link to={`/employees/${employee.id}`}>{employee.name} - {employee.position}</Link>

</li>

))}

</ul>

</div>

);

}

**4. EmployeeDetails.js (Employee Details)**

import React from "react";

import { useParams } from "react-router-dom";

const employees = [

{ id: 1, name: "John Doe", position: "Software Engineer", department: "Engineering", email: "john@example.com" },

{ id: 2, name: "Jane Smith", position: "HR Manager", department: "Human Resources", email: "jane@example.com" },

{ id: 3, name: "Alice Johnson", position: "Product Manager", department: "Product", email: "alice@example.com" },

{ id: 4, name: "Bob Brown", position: "UX Designer", department: "Design", email: "bob@example.com" },

// Add more employees as needed

];

export default function EmployeeDetails() {

const { employeeId } = useParams();

const employee = employees.find((e) => e.id === parseInt(employeeId));

if (!employee) {

return <h2>Employee not found!</h2>;

}

return (

<div>

<h1>{employee.name}</h1>

<p>Position: {employee.position}</p>

<p>Department: {employee.department}</p>

<p>Email: {employee.email}</p>

</div>

);

}

**5. Departments.js (List of Departments)**

import React from "react";

import { Link } from "react-router-dom";

const departments = [

{ id: 1, name: "Engineering" },

{ id: 2, name: "Human Resources" },

{ id: 3, name: "Product" },

{ id: 4, name: "Design" },

// Add more departments as needed

];

export default function Departments() {

return (

<div>

<h1>Departments</h1>

<ul>

{departments.map((department) => (

<li key={department.id}>

<Link to={`/departments/${department.id}`}>{department.name}</Link>

</li>

))}

</ul>

</div>

);

}

**6. DepartmentDetails.js (Employees in a Department)**

import React from "react";

import { useParams } from "react-router-dom";

const employees = [

{ id: 1, name: "John Doe", department: "Engineering" },

{ id: 2, name: "Jane Smith", department: "Human Resources" },

{ id: 3, name: "Alice Johnson", department: "Product" },

{ id: 4, name: "Bob Brown", department: "Design" },

{ id: 5, name: "Charlie White", department: "Engineering" },

{ id: 6, name: "David Green", department: "Product" },

// Add more employees as needed

];

export default function DepartmentDetails() {

const { departmentId } = useParams();

const departmentNames = ["Engineering", "Human Resources", "Product", "Design"];

const departmentName = departmentNames[parseInt(departmentId) - 1];

const departmentEmployees = employees.filter((employee) => employee.department === departmentName);

return (

<div>

<h1>{departmentName} Department</h1>

<ul>

{departmentEmployees.map((employee) => (

<li key={employee.id}>{employee.name}</li>

))}

</ul>

</div>

);

}

**Explanation:**

1. **App.js**:
   * Routes are configured for the dashboard, employees, employee details, departments, and department details. We use useParams to dynamically fetch data based on the route.
2. **Dashboard.js**:
   * Displays basic HR statistics, including total employees and open positions. You can update these values based on the actual data.
3. **Employees.js**:
   * Displays a list of all employees. Each employee is clickable, and clicking on an employee name will navigate to their details page.
4. **EmployeeDetails.js**:
   * Displays detailed information about a selected employee. It uses the useParams hook to fetch the employee ID from the URL and displays the corresponding details.
5. **Departments.js**:
   * Displays a list of all departments. Each department is clickable, and clicking on a department name will navigate to a page that lists employees from that department.
6. **DepartmentDetails.js**:
   * Displays a list of employees in a specific department. It uses the useParams hook to fetch the department ID from the URL and filters the employees based on the department.

**How to Run:**

1. Set up a React application (using create-react-app or another setup).
2. Replace the contents of App.js, Dashboard.js, Employees.js, EmployeeDetails.js, Departments.js, and DepartmentDetails.js with the code provided above.
3. Run npm start to start the application.

**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.

In this solution, we'll build a **Blogging Platform** with nested and parameterized routes. The platform will include the following components:

1. **Home**: Displays a list of blog categories.
2. **Category**: Displays a list of blog posts under a selected category.
3. **PostDetails**: Displays detailed information about a selected blog post.
4. **Author**: Displays details about the author of a blog post.
5. **About**: Displays information about the blog platform.

We'll implement the following routes:

1. /: Displays the **Home** component.
2. /category/:categoryId: Displays the **Category** component, showing blog posts under that category.
3. /category/:categoryId/post/:postId: Displays the **PostDetails** component, showing the details of a specific post.
4. /author/:authorId: Displays the **Author** component, showing details about the post's author.
5. /about: Displays the **About** component.

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Category from "./Category";

import PostDetails from "./PostDetails";

import Author from "./Author";

import About from "./About";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> |

<Link to="/about">About</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/category/:categoryId" element={<Category />} />

<Route path="/category/:categoryId/post/:postId" element={<PostDetails />} />

<Route path="/author/:authorId" element={<Author />} />

<Route path="/about" element={<About />} />

</Routes>

</Router>

);

}

**2. Home.js (List of Categories)**

import React from "react";

import { Link } from "react-router-dom";

const categories = [

{ id: 1, name: "Technology" },

{ id: 2, name: "Lifestyle" },

{ id: 3, name: "Travel" },

{ id: 4, name: "Food" },

// Add more categories as needed

];

export default function Home() {

return (

<div>

<h1>Welcome to the Blog Platform</h1>

<h2>Categories</h2>

<ul>

{categories.map((category) => (

<li key={category.id}>

<Link to={`/category/${category.id}`}>{category.name}</Link>

</li>

))}

</ul>

</div>

);

}

**3. Category.js (List of Posts under a Category)**

import React from "react";

import { useParams, Link } from "react-router-dom";

const posts = {

1: [

{ id: 1, title: "Tech Post 1" },

{ id: 2, title: "Tech Post 2" },

{ id: 3, title: "Tech Post 3" },

],

2: [

{ id: 1, title: "Lifestyle Post 1" },

{ id: 2, title: "Lifestyle Post 2" },

{ id: 3, title: "Lifestyle Post 3" },

],

3: [

{ id: 1, title: "Travel Post 1" },

{ id: 2, title: "Travel Post 2" },

],

4: [

{ id: 1, title: "Food Post 1" },

{ id: 2, title: "Food Post 2" },

],

};

export default function Category() {

const { categoryId } = useParams();

const categoryPosts = posts[categoryId] || [];

return (

<div>

<h1>Category {categoryId}</h1>

<h2>Posts</h2>

<ul>

{categoryPosts.map((post) => (

<li key={post.id}>

<Link to={`/category/${categoryId}/post/${post.id}`}>{post.title}</Link>

</li>

))}

</ul>

</div>

);

}

**4. PostDetails.js (Blog Post Details)**

import React from "react";

import { useParams } from "react-router-dom";

const posts = {

1: {

1: { title: "Tech Post 1", content: "This is the content of Tech Post 1", authorId: 1 },

2: { title: "Tech Post 2", content: "This is the content of Tech Post 2", authorId: 2 },

3: { title: "Tech Post 3", content: "This is the content of Tech Post 3", authorId: 3 },

},

2: {

1: { title: "Lifestyle Post 1", content: "This is the content of Lifestyle Post 1", authorId: 1 },

2: { title: "Lifestyle Post 2", content: "This is the content of Lifestyle Post 2", authorId: 2 },

3: { title: "Lifestyle Post 3", content: "This is the content of Lifestyle Post 3", authorId: 3 },

},

3: {

1: { title: "Travel Post 1", content: "This is the content of Travel Post 1", authorId: 1 },

2: { title: "Travel Post 2", content: "This is the content of Travel Post 2", authorId: 2 },

},

4: {

1: { title: "Food Post 1", content: "This is the content of Food Post 1", authorId: 1 },

2: { title: "Food Post 2", content: "This is the content of Food Post 2", authorId: 2 },

},

};

export default function PostDetails() {

const { categoryId, postId } = useParams();

const post = posts[categoryId]?.[postId];

if (!post) {

return <h2>Post not found!</h2>;

}

return (

<div>

<h1>{post.title}</h1>

<p>{post.content}</p>

<Link to={`/author/${post.authorId}`}>Read about the author</Link>

</div>

);

}

**5. Author.js (Author Details)**

import React from "react";

import { useParams } from "react-router-dom";

const authors = {

1: { name: "John Doe", bio: "John is a technology enthusiast and writer." },

2: { name: "Jane Smith", bio: "Jane is a lifestyle blogger and content creator." },

3: { name: "Alice Johnson", bio: "Alice is a travel writer and photographer." },

};

export default function Author() {

const { authorId } = useParams();

const author = authors[authorId];

if (!author) {

return <h2>Author not found!</h2>;

}

return (

<div>

<h1>{author.name}</h1>

<p>{author.bio}</p>

</div>

);

}

**6. About.js (About the Blog Platform)**

import React from "react";

export default function About() {

return (

<div>

<h1>About the Blog Platform</h1>

<p>This blog platform allows users to browse and read blog posts across different categories.</p>

</div>

);

}

**Explanation:**

1. **App.js**:
   * The main app has routes set up for each component. It uses react-router-dom to navigate between pages.
2. **Home.js**:
   * Displays a list of categories. Clicking on a category name links to /category/:categoryId.
3. **Category.js**:
   * Displays a list of posts under the selected category. Clicking on a post title links to /category/:categoryId/post/:postId.
4. **PostDetails.js**:
   * Displays the details of a selected blog post. It retrieves the post using the useParams hook with both categoryId and postId.
5. **Author.js**:
   * Displays the details of the author of a post. It retrieves the author’s information based on the authorId parameter from the URL.
6. **About.js**:
   * Displays information about the blog platform, explaining its purpose.

**How to Run:**

1. Set up a React application (using create-react-app or another setup).
2. Replace the contents of App.js, Home.js, Category.js, PostDetails.js, Author.js, and About.js with the code provided above.
3. Run npm start to start the application.

**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.

In this exercise, we'll develop a **Travel Management System** with the following components:

1. **Home**: A landing page that showcases available destinations.
2. **Destinations**: A page that lists all available destinations.
3. **DestinationDetails**: A page that displays detailed information about a specific destination.
4. **Booking**: A page that allows users to book a selected destination.
5. **UserDashboard**: A page that displays all bookings for a user.

We will implement the following routes:

1. /: Displays the **Home** component.
2. /destinations: Displays the **Destinations** component, listing available destinations.
3. /destinations/:destinationId: Displays the **DestinationDetails** component, showing detailed information about a specific destination.
4. /booking: Displays the **Booking** component, where users can book a destination.
5. /dashboard: Displays the **UserDashboard** component, listing the user's bookings.

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Destinations from "./Destinations";

import DestinationDetails from "./DestinationDetails";

import Booking from "./Booking";

import UserDashboard from "./UserDashboard";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> |

<Link to="/destinations">Destinations</Link> |

<Link to="/booking">Book a Destination</Link> |

<Link to="/dashboard">Your Bookings</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/destinations" element={<Destinations />} />

<Route path="/destinations/:destinationId" element={<DestinationDetails />} />

<Route path="/booking" element={<Booking />} />

<Route path="/dashboard" element={<UserDashboard />} />

</Routes>

</Router>

);

}

**2. Home.js (Landing Page)**

import React from "react";

import { Link } from "react-router-dom";

const destinations = [

{ id: 1, name: "Paris" },

{ id: 2, name: "New York" },

{ id: 3, name: "Tokyo" },

];

export default function Home() {

return (

<div>

<h1>Welcome to the Travel Management System</h1>

<h2>Explore Popular Destinations</h2>

<ul>

{destinations.map((destination) => (

<li key={destination.id}>

<Link to={`/destinations/${destination.id}`}>{destination.name}</Link>

</li>

))}

</ul>

</div>

);

}

**3. Destinations.js (List of Destinations)**

import React from "react";

import { Link } from "react-router-dom";

const destinations = [

{ id: 1, name: "Paris", description: "The City of Light" },

{ id: 2, name: "New York", description: "The Big Apple" },

{ id: 3, name: "Tokyo", description: "The Capital of Japan" },

];

export default function Destinations() {

return (

<div>

<h1>All Destinations</h1>

<ul>

{destinations.map((destination) => (

<li key={destination.id}>

<Link to={`/destinations/${destination.id}`}>{destination.name}</Link>

</li>

))}

</ul>

</div>

);

}

**4. DestinationDetails.js (Destination Details)**

import React from "react";

import { useParams } from "react-router-dom";

const destinationDetails = {

1: { name: "Paris", description: "The City of Light", imageUrl: "/paris.jpg" },

2: { name: "New York", description: "The Big Apple", imageUrl: "/newyork.jpg" },

3: { name: "Tokyo", description: "The Capital of Japan", imageUrl: "/tokyo.jpg" },

};

export default function DestinationDetails() {

const { destinationId } = useParams();

const destination = destinationDetails[destinationId];

if (!destination) {

return <h2>Destination not found!</h2>;

}

return (

<div>

<h1>{destination.name}</h1>

<p>{destination.description}</p>

<img src={destination.imageUrl} alt={destination.name} />

</div>

);

}

**5. Booking.js (Booking Form)**

import React, { useState } from "react";

import { useNavigate } from "react-router-dom";

const destinations = [

{ id: 1, name: "Paris" },

{ id: 2, name: "New York" },

{ id: 3, name: "Tokyo" },

];

export default function Booking() {

const [selectedDestination, setSelectedDestination] = useState("");

const navigate = useNavigate();

const handleBooking = () => {

// Normally, you'd save booking data to a backend or local storage

navigate("/dashboard");

};

return (

<div>

<h1>Book Your Destination</h1>

<form onSubmit={(e) => e.preventDefault()}>

<div>

<label>Select Destination: </label>

<select

value={selectedDestination}

onChange={(e) => setSelectedDestination(e.target.value)}

>

<option value="">--Select a Destination--</option>

{destinations.map((destination) => (

<option key={destination.id} value={destination.id}>

{destination.name}

</option>

))}

</select>

</div>

<button onClick={handleBooking}>Book</button>

</form>

</div>

);

}

**6. UserDashboard.js (User's Bookings)**

import React from "react";

const bookings = [

{ id: 1, destination: "Paris", date: "2025-06-01" },

{ id: 2, destination: "Tokyo", date: "2025-07-15" },

];

export default function UserDashboard() {

return (

<div>

<h1>Your Bookings</h1>

<ul>

{bookings.map((booking) => (

<li key={booking.id}>

{booking.destination} - {booking.date}

</li>

))}

</ul>

</div>

);

}

**Explanation:**

1. **App.js**:
   * This is the main application file that uses react-router-dom to create routes for all components: Home, Destinations, DestinationDetails, Booking, and UserDashboard.
2. **Home.js**:
   * The **Home** component displays a landing page with links to explore popular destinations. It gives a preview of what destinations are available.
3. **Destinations.js**:
   * The **Destinations** component displays all available destinations. Each destination is clickable, linking to the **DestinationDetails** component.
4. **DestinationDetails.js**:
   * The **DestinationDetails** component displays detailed information about a specific destination when the user clicks on it.
5. **Booking.js**:
   * The **Booking** component allows the user to select a destination from a dropdown and book it. Once booked, the user is redirected to the **UserDashboard**.
6. **UserDashboard.js**:
   * The **UserDashboard** component displays a list of the user's bookings.

**How to Run:**

1. Set up a React application (using create-react-app or another setup).
2. Replace the contents of App.js, Home.js, Destinations.js, DestinationDetails.js, Booking.js, and UserDashboard.js with the code provided above.
3. Run npm start to start the application.

**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.

In this exercise, we'll develop a **University Portal** with the following components:

1. **Home**: Displays the introduction to the university.
2. **Admissions**: Contains information about the admission process.
3. **Programs**: Lists all academic programs offered by the university.
4. **ProgramDetails**: Displays details about a specific academic program.
5. **Faculty**: Lists all faculty members.
6. **FacultyDetails**: Displays information about a specific faculty member.

We will implement the following routes:

1. /: Displays the **Home** component.
2. /admissions: Displays the **Admissions** component.
3. /programs: Displays the **Programs** component, listing all academic programs.
4. /programs/:programId: Displays the **ProgramDetails** component, showing details about a specific program.
5. /faculty: Displays the **Faculty** component, listing all faculty members.
6. /faculty/:facultyId: Displays the **FacultyDetails** component, showing details about a specific faculty member.

**Solution Code**

**1. App.js (Main Application)**

import React from "react";

import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

import Home from "./Home";

import Admissions from "./Admissions";

import Programs from "./Programs";

import ProgramDetails from "./ProgramDetails";

import Faculty from "./Faculty";

import FacultyDetails from "./FacultyDetails";

export default function App() {

return (

<Router>

<nav>

<Link to="/">Home</Link> |

<Link to="/admissions">Admissions</Link> |

<Link to="/programs">Programs</Link> |

<Link to="/faculty">Faculty</Link>

</nav>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/admissions" element={<Admissions />} />

<Route path="/programs" element={<Programs />} />

<Route path="/programs/:programId" element={<ProgramDetails />} />

<Route path="/faculty" element={<Faculty />} />

<Route path="/faculty/:facultyId" element={<FacultyDetails />} />

</Routes>

</Router>

);

}

**2. Home.js (University Introduction)**

import React from "react";

export default function Home() {

return (

<div>

<h1>Welcome to Our University</h1>

<p>We offer world-class education across a variety of disciplines.</p>

</div>

);

}

**3. Admissions.js (Admission Information)**

import React from "react";

export default function Admissions() {

return (

<div>

<h1>Admissions</h1>

<p>Find out everything you need to know about our admission process, including deadlines and requirements.</p>

</div>

);

}

**4. Programs.js (List of Programs)**

import React from "react";

import { Link } from "react-router-dom";

const programs = [

{ id: 1, name: "Computer Science" },

{ id: 2, name: "Mechanical Engineering" },

{ id: 3, name: "Business Administration" },

];

export default function Programs() {

return (

<div>

<h1>Academic Programs</h1>

<ul>

{programs.map((program) => (

<li key={program.id}>

<Link to={`/programs/${program.id}`}>{program.name}</Link>

</li>

))}

</ul>

</div>

);

}

**5. ProgramDetails.js (Details of a Specific Program)**

import React from "react";

import { useParams } from "react-router-dom";

const programDetails = {

1: { name: "Computer Science", description: "Learn the fundamentals of computing and software development." },

2: { name: "Mechanical Engineering", description: "Focus on designing, analyzing, and manufacturing mechanical systems." },

3: { name: "Business Administration", description: "Prepare for leadership roles in the corporate world." },

};

export default function ProgramDetails() {

const { programId } = useParams();

const program = programDetails[programId];

if (!program) {

return <h2>Program not found!</h2>;

}

return (

<div>

<h1>{program.name}</h1>

<p>{program.description}</p>

</div>

);

}

**6. Faculty.js (List of Faculty Members)**

import React from "react";

import { Link } from "react-router-dom";

const faculty = [

{ id: 1, name: "Dr. John Doe", department: "Computer Science" },

{ id: 2, name: "Dr. Jane Smith", department: "Mechanical Engineering" },

{ id: 3, name: "Dr. Robert Brown", department: "Business Administration" },

];

export default function Faculty() {

return (

<div>

<h1>Our Faculty</h1>

<ul>

{faculty.map((member) => (

<li key={member.id}>

<Link to={`/faculty/${member.id}`}>{member.name}</Link>

</li>

))}

</ul>

</div>

);

}

**7. FacultyDetails.js (Details of a Specific Faculty Member)**

import React from "react";

import { useParams } from "react-router-dom";

const facultyDetails = {

1: { name: "Dr. John Doe", department: "Computer Science", bio: "Dr. John Doe is a leading expert in artificial intelligence." },

2: { name: "Dr. Jane Smith", department: "Mechanical Engineering", bio: "Dr. Jane Smith specializes in robotics and automation." },

3: { name: "Dr. Robert Brown", department: "Business Administration", bio: "Dr. Robert Brown focuses on leadership and management strategies." },

};

export default function FacultyDetails() {

const { facultyId } = useParams();

const member = facultyDetails[facultyId];

if (!member) {

return <h2>Faculty member not found!</h2>;

}

return (

<div>

<h1>{member.name}</h1>

<p>Department: {member.department}</p>

<p>{member.bio}</p>

</div>

);

}

**Explanation:**

1. **App.js**:
   * This is the main application file that defines routes for each component using react-router-dom. It includes links for navigation and specifies routes for the home page, admissions, programs, program details, faculty list, and faculty details.
2. **Home.js**:
   * The **Home** component provides a welcome message about the university and an introduction to what the portal offers.
3. **Admissions.js**:
   * The **Admissions** component provides detailed information about the university's admission process, deadlines, and requirements.
4. **Programs.js**:
   * The **Programs** component lists all available academic programs, each linking to the **ProgramDetails** component.
5. **ProgramDetails.js**:
   * The **ProgramDetails** component shows detailed information about a specific program, fetched dynamically based on the program ID from the URL.
6. **Faculty.js**:
   * The **Faculty** component lists all faculty members and links to their individual details.
7. **FacultyDetails.js**:
   * The **FacultyDetails** component shows detailed information about a specific faculty member, including their department and biography.

**How to Run:**

1. Set up a React application (using create-react-app or another setup).
2. Replace the contents of App.js, Home.js, Admissions.js, Programs.js, ProgramDetails.js, Faculty.js, and FacultyDetails.js with the code provided above.
3. Run npm start to start the application.