# Lab - Week 5

PART A

(PART A: TO BE REFERRED BY STUDENTS)

**Objective** Develop a multi-page React app using React Router.

# Theory:

React Router is a standard library for routing in React. It enables the creation of single-page applications with navigation among multiple views without reloading the page. In this lab, we will use React Router to build a multi-page application.

Key Concepts:

1. **React Router**: A library to manage routes and navigation in a React application.
2. **URL Parameters**: Used to pass dynamic data in the URL and display specific content on a page.
3. **Navigation Links**: Components like <Link> and <NavLink> enable smooth navigation.
4. **Dynamic Content Rendering**: Use URL parameters to render content based on the user’s navigation.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the portal at the end of the practical. The filename should be **AWT\_batch\_rollno\_experimentno Example: AWT\_A1\_A001\_P1**

|  |  |
| --- | --- |
| **Roll No.: A046** | **Name: TEJAS PADMAKAR** |
| **Prog/Yr/Sem: 2** | **Batch: 2** |
| **Date of Experiment: 13/02/25** | **Date of Submission:** |

# Task 1: Build a Multi-Page React App

1. **Setup React Environment**:
   * Create a new React project using create-react-app.
   * Install react-router-dom for routing.

# Create Pages:

* + **Home Page**: A welcome page with some basic content.
  + **About Page**: A page describing the application or developer.
  + **Contact Page**: A form to collect user inquiries (name, email, and message).

# Navigation Links:

* + Add a navigation bar or menu with links to Home, About, and Contact pages.
  + Use the <Link> or <NavLink> components for navigation.

# Details Page with URL Parameters:

* + Create a Details page that accepts a parameter from the URL (e.g.,

/details/:id).

* + Use the parameter to display specific content (e.g., product details, user profile).

# Dynamic Content Rendering:

* + Extract the parameter using useParams from react-router-dom.
  + Fetch or display dynamic content based on the parameter value.

// Home.jsx

import React from 'react'

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

const items = [

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

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

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

]

function Home() {

return (

<div>

<h1>This is our home page</h1>

<p>Product List</p>

<ul>

{items.map((*item*) => (

<li *key*={*item*.id}>

<Link *to*={`/details/${*item*.id}`}>{*item*.name}</Link>

</li>

))}

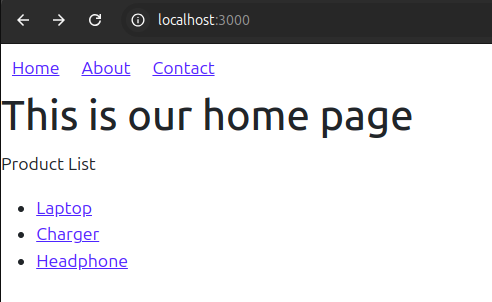
</ul>

</div>

)

}

export default Home



// About.jsx

import React from 'react'

function About() {

return (

<div>

<h2>Hello from about page</h2>

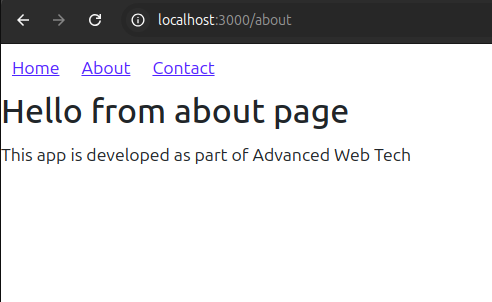
<p>This app is developed as part of Advanced Web Tech</p>

</div>

)

}

export default About



// Contact.jsx

import React, { useState } from 'react'

function Contact() {

const[formData, setFormData] = useState({ name: "", email: "", message: ""})

const handleChange = (*e*) => {

setFormData({...formData, [*e*.target.name]: *e*.target.value })

}

const handleSubmit = (*e*) => {

*e*.preventDefault();

console.log("Submitted:", formData);

alert("Submitted!")

}

return (

<div>

<h1>Contact us</h1>

<form *onSubmit*={handleSubmit}>

<input *type*="text" *name*='name' *placeholder*='Your name' *onChange*={handleChange} *required* />

<input *type*="email" *name*='name' *placeholder*='Your email' *onChange*={handleChange} *required* />

<input *type*="message" *placeholder*='Your message' *onChange*={handleChange} *required* />

<button *type*='submit'>Submit</button>

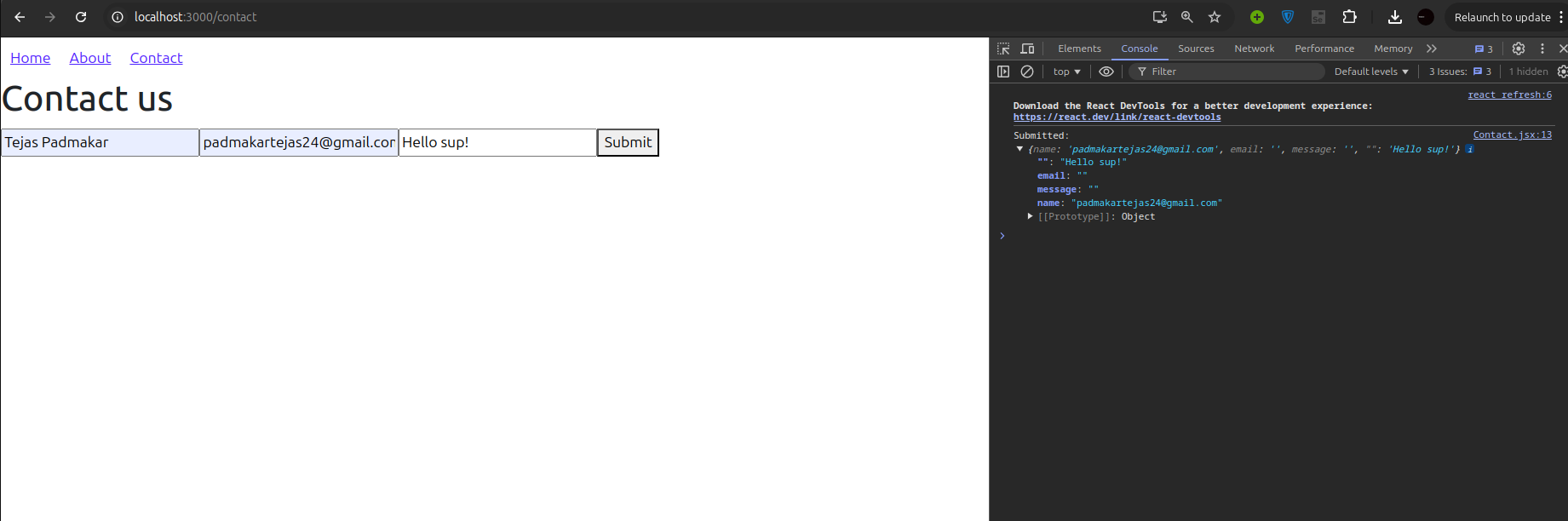
</form>

</div>

)

}

export default Contact



// Details.jsx

import React from 'react'

import { useParams } from 'react-router-dom'

function Details() {

const { id } = useParams();

return (

<div>

<h1>Details</h1>

<p>Displaying details for item ID: {id} </p>

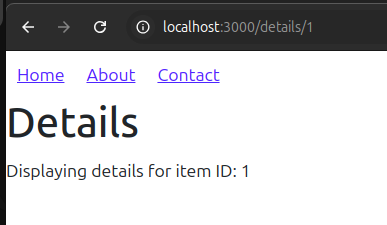
</div>

)

}

export default Details

When clicked on Laptop from HomePage



# Task 2: Bonus Scenarios

1. **Blog Application**:
   * Create a blog with a list of articles on the Home page.
   * Use the Details page to display the content of a selected article.



// ArticleList.jsx

import React from 'react';

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

const articles = [

{ id: 1, title: 'React Basics' },

{ id: 2, title: 'Advanced React' },

{ id: 3, title: 'React Router Deep Dive' }

];

function ArticleList() {

return (

<div *className*="container mt-4">

<h1 *className*="mb-4">Articles</h1>

<div *className*="row">

{articles.map((*article*) => (

<div *key*={*article*.id} *className*="col-md-4 mb-4">

<div *className*="card">

<div *className*="card-body">

<h5 *className*="card-title">{*article*.title}</h5>

<Link *to*={`/article/${*article*.id}`} *className*="btn btn-primary">

Read More

</Link>

</div>

</div>

</div>

))}

</div>

</div>

);

}

export default ArticleList;

// ArticleDetails.jsx

import React from 'react';

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

const articles = [

{ id: 1, title: 'React Basics', content: 'Hi letsss learn basics of reacttttt' },

{ id: 2, title: 'Advanced React', content: 'Advanceddd reacttttt' },

{ id: 3, title: 'React Router Deep Dive', content: 'Less learn react routerr deep diveee' }

];

function ArticleDetail() {

const { id } = useParams();

const article = articles.find((*a*) => *a*.id === parseInt(id));

if (!article) {

return <h2 *className*="text-center mt-4">Article not found</h2>;

}

return (

<div *className*="container mt-4">

<div *className*="card">

<div *className*="card-body">

<h1 *className*="card-title">{article.title}</h1>

<p *className*="card-text">{article.content}</p>

</div>

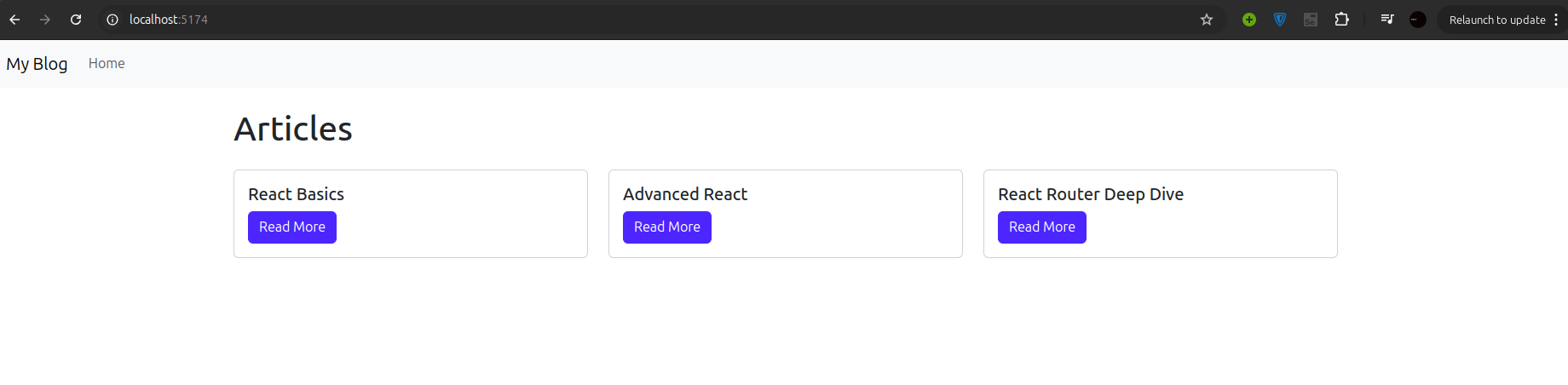
</div>

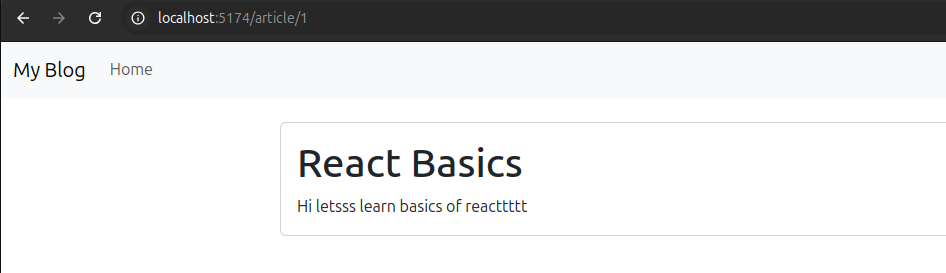
</div>

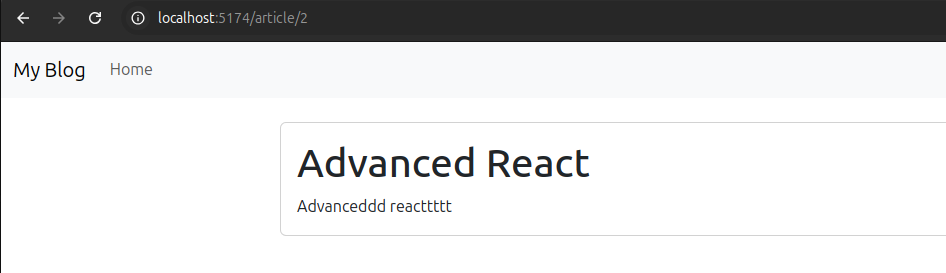
);

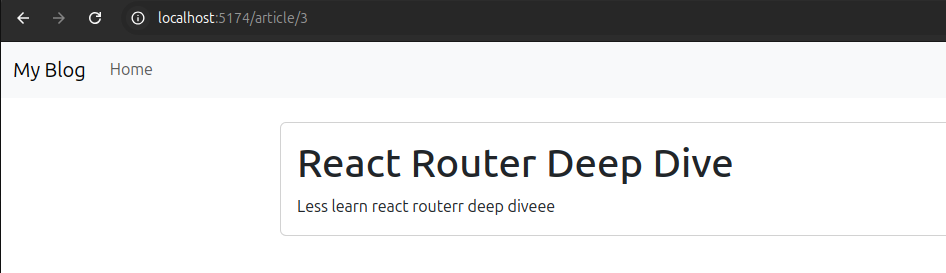
}

export default ArticleDetail;









# Portfolio Navigation:

* + Develop a portfolio site with pages for Projects, Skills, and Contact.
  + Include a Projects page that links to individual project details.

import React from 'react';

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

import Projects from './components/Projects';

import Skills from './components/Skills';

import Contact from './components/Contact';

import ProjectDetail from './components/ProjectDetail';

function App() {

return (

<Router>

<div *className*="container">

<nav *className*="navbar navbar-expand-lg navbar-light bg-light">

<div *className*="container-fluid">

<Link *className*="navbar-brand" *to*="/">Portfolio</Link>

<div *className*="collapse navbar-collapse">

<ul *className*="navbar-nav">

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/projects">Projects</Link>

</li>

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/skills">Skills</Link>

</li>

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/contact">Contact</Link>

</li>

</ul>

</div>

</div>

</nav>

<Routes>

<Route *path*="/" *element*={<h1>Welcome to My Portfolio</h1>} />

<Route *path*="/projects" *element*={<Projects />} />

<Route *path*="/skills" *element*={<Skills />} />

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

<Route *path*="/project/:id" *element*={<ProjectDetail />} />

</Routes>

</div>

</Router>

);

}

export default App;

// Projects.jsx

import React from 'react';

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

const Projects = () => {

const projects = [

{ id: 1, name: 'Project 1' },

{ id: 2, name: 'Project 2' },

{ id: 3, name: 'Project 3' },

];

return (

<div>

<h2>My Projects</h2>

<ul>

{projects.map(*project* => (

<li *key*={*project*.id}>

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

</li>

))}

</ul>

</div>

);

};

export default Projects;

// Skills.jsx

import React from 'react';

const Skills = () => {

const skills = ['JavaScript', 'React', 'Node.js', 'CSS', 'HTML'];

return (

<div>

<h2>My Skills</h2>

<ul>

{skills.map(*skill* => (

<li *key*={*skill*}>{*skill*}</li>

))}

</ul>

</div>

);

};

export default Skills;

// Contact.jsx

import React from 'react';

const Contact = () => {

return (

<div>

<h2>Contact Me</h2>

<p>Email: example@domain.com</p>

<p>Phone: +123 456 7890</p>

</div>

);

};

export default Contact;

// ProjectDetail.jsx

import React, { useEffect, useState } from 'react';

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

const ProjectDetail = () => {

const { id } = useParams();

const [project, setProject] = useState(null);

useEffect(() => {

const fetchedProject = { id, name: `Project ${id}`, description: `Details of project ${id}.` };

setProject(fetchedProject);

}, [id]);

if (!project) return <div>Loading...</div>;

return (

<div>

<h2>{project.name}</h2>

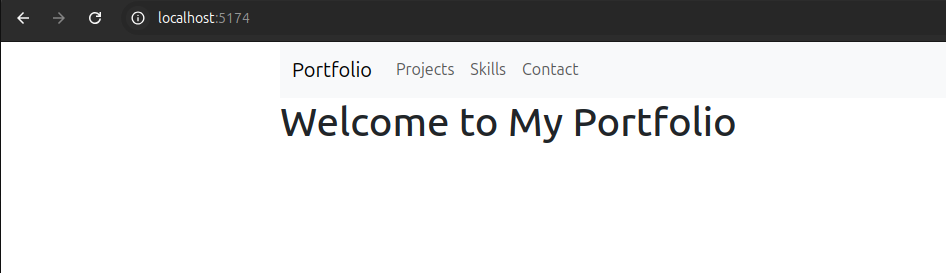
<p>{project.description}</p>

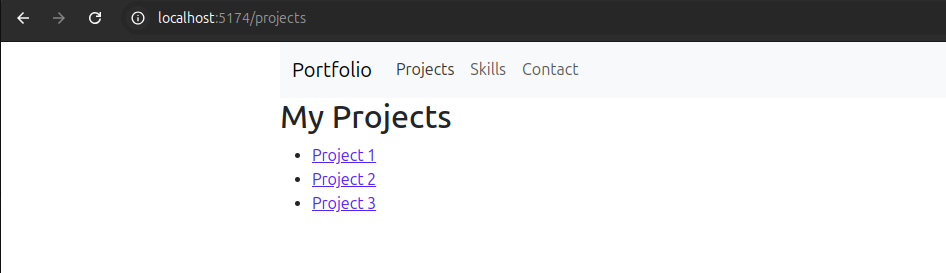
</div>

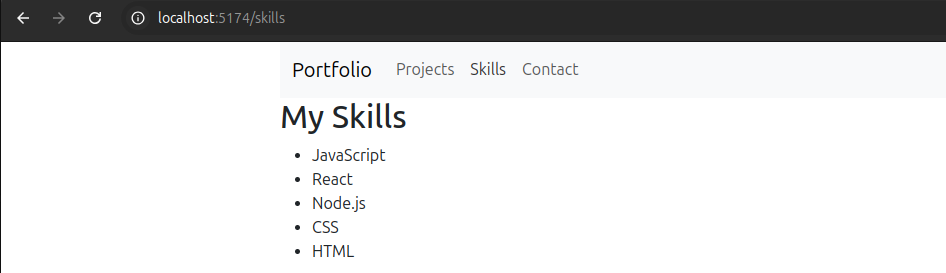
);

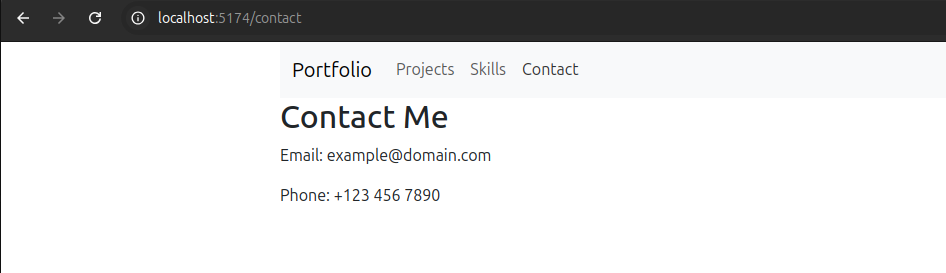
};

export default ProjectDetail;









# Enhance the Multi-Page App:

* + Add a "Not Found" page for undefined routes.
  + Use query parameters (e.g., ?sort=asc) to sort or filter content dynamically.

import React from 'react';

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

import Projects from './components/Projects';

import Skills from './components/Skills';

import Contact from './components/Contact';

import ProjectDetail from './components/ProjectDetail';

import NotFound from './components/NotFound';

function App() {

return (

<Router>

<div *className*="container">

<nav *className*="navbar navbar-expand-lg navbar-light bg-light">

<div *className*="container-fluid">

<Link *className*="navbar-brand" *to*="/">Portfolio</Link>

<div *className*="collapse navbar-collapse">

<ul *className*="navbar-nav">

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/">Home</Link>

</li>

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/projects">Projects</Link>

</li>

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/skills">Skills</Link>

</li>

<li *className*="nav-item">

<Link *className*="nav-link" *to*="/contact">Contact</Link>

</li>

</ul>

</div>

</div>

</nav>

<Routes>

<Route *path*="/" *element*={<h1>Welcome to My Portfolio</h1>} />

<Route *path*="/projects" *element*={<Projects />} />

<Route *path*="/skills" *element*={<Skills />} />

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

<Route *path*="/project/:id" *element*={<ProjectDetail />} />

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

</Routes>

</div>

</Router>

);

}

export default App;

// NotFound.jsx

import React from 'react';

const NotFound = () => {

return (

<div>

<h2>404 - Page Not Found</h2>

<p>The page you're looking for doesn't exist.</p>

</div>

);

};

export default NotFound;

// Projects.jsx

import React, { useState, useEffect } from 'react';

import { Link, useLocation } from 'react-router-dom';

const Projects = () => {

const [projects, setProjects] = useState([

{ id: 1, name: 'Project W' },

{ id: 2, name: 'Project E' },

{ id: 3, name: 'Project S' },

{ id: 4, name: 'Project A' },

]);

const location = useLocation();

const queryParams = new URLSearchParams(location.search);

const sortOrder = queryParams.get('sort') || 'asc';

useEffect(() => {

const sortedProjects = [...projects].sort((*a*, *b*) => {

if (sortOrder === 'asc') {

return *a*.name.localeCompare(*b*.name);

} else {

return *b*.name.localeCompare(*a*.name);

}

});

setProjects(sortedProjects);

}, [sortOrder]);

return (

<div>

<h2>My Projects</h2>

<div>

<Link *to*="/projects?sort=asc">Sort Ascending</Link> |

<Link *to*="/projects?sort=desc">Sort Descending</Link>

</div>

<ul>

{projects.map(*project* => (

<li *key*={*project*.id}>

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

</li>

))}

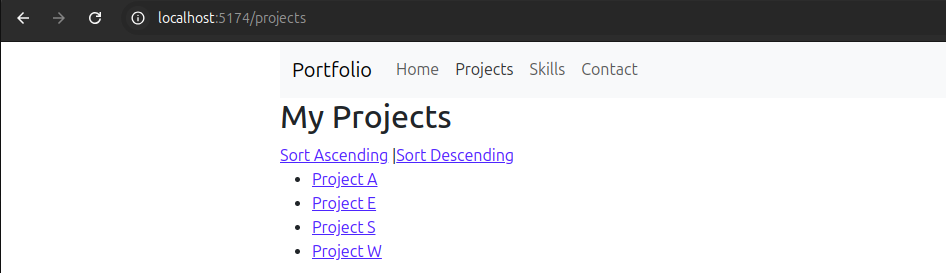
</ul>

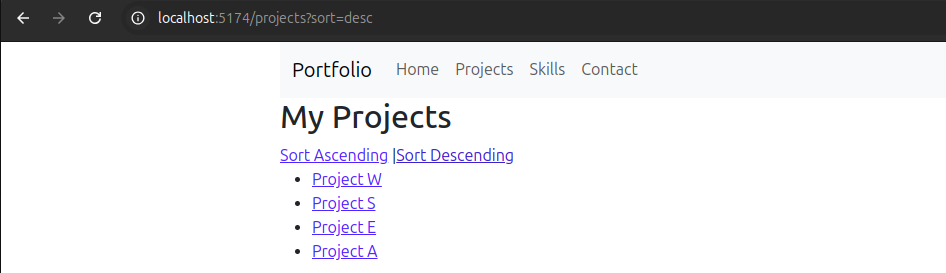
</div>

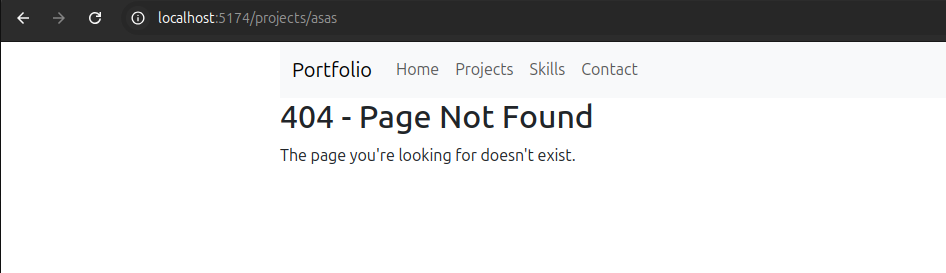
);

};

export default Projects;







# Enhancements (Optional but Encouraged):

* Use CSS frameworks like Bootstrap or Tailwind for styling.
* Implement animations for page transitions.
* Add breadcrumbs to enhance navigation.

# Deliverables:

1. Functional multi-page React app with routing and navigation.
2. Dynamic content displayed using URL parameters.
3. Bonus: Implement one additional scenario from Task 2.

# Submission Checklist:

* + Source code with appropriate comments.
  + Screenshot of the application in the browser.
  + Explanation of the use of React Router and URL parameters.

# Expected Learning Outcomes:

By the end of this practical, students will be able to:

1. Use React Router to create a multi-page React application.
2. Implement navigation using <Link> and <NavLink>.
3. Utilize URL parameters to render dynamic content.
4. Extend React applications with routing capabilities for real-world scenarios.