**React Forms Exercises with Router Integration**

Here are some exercises focused on building **React forms** with **React Router** integration. These exercises will help you practice creating forms, handling user inputs, managing form state, and navigating between different components using routing.

**Exercise 1: Simple Contact Form**

**Objective**: Create a contact form that captures user input and displays a confirmation message upon submission.

**Requirements:**

1. Create a form with the following fields:
   * Name
   * Email
   * Message
2. Validate that **Name** and **Email** are required.
3. After submission, redirect to a **Thank You** page using React Router.

**Routes:**

1. /contact - Displays the Contact Form.
2. /thank-you - Displays a Thank You message after form submission.

**Solution for Exercise 1**

**Project Structure:**

/src

/components

ContactForm.js

ThankYouPage.js

/css

ContactForm.css

ThankYouPage.css

App.js

index.js

**1. Install dependencies**

If you don't have react-router-dom installed, you can install it with the following command:

npm install react-router-dom

**2. ContactForm.js**

This component handles the contact form and redirects to the thank-you page after submission.

// src/components/ContactForm.js

import React, { useState } from 'react';

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

import './ContactForm.css';

const ContactForm = () => {

const [name, setName] = useState('');

const [email, setEmail] = useState('');

const [message, setMessage] = useState('');

const [error, setError] = useState('');

const history = useHistory();

const handleSubmit = (e) => {

e.preventDefault();

if (!name || !email) {

setError('Name and Email are required');

} else {

// You can submit the form data here if needed

history.push('/thank-you');

}

};

return (

<div className="contact-form">

<h2>Contact Us</h2>

{error && <p className="error">{error}</p>}

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

value={name}

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

/>

</div>

<div>

<label>Email:</label>

<input

type="email"

value={email}

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

/>

</div>

<div>

<label>Message:</label>

<textarea

value={message}

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

/>

</div>

<button type="submit">Submit</button>

</form>

</div>

);

};

export default ContactForm;

**3. ThankYouPage.js**

This component displays the Thank You message after form submission.

// src/components/ThankYouPage.js

import React from 'react';

import './ThankYouPage.css';

const ThankYouPage = () => {

return (

<div className="thank-you-page">

<h2>Thank You!</h2>

<p>Your message has been submitted successfully. We will get back to you soon.</p>

</div>

);

};

export default ThankYouPage;

**4. ContactForm.css**

CSS for styling the contact form.

/\* src/css/ContactForm.css \*/

.contact-form {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

.contact-form h2 {

text-align: center;

}

.contact-form form {

display: flex;

flex-direction: column;

}

.contact-form label {

margin-bottom: 5px;

font-weight: bold;

}

.contact-form input,

.contact-form textarea {

padding: 8px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

.contact-form button {

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

.contact-form button:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 14px;

text-align: center;

}

**5. ThankYouPage.css**

CSS for styling the Thank You page.

/\* src/css/ThankYouPage.css \*/

.thank-you-page {

max-width: 500px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

text-align: center;

}

.thank-you-page h2 {

font-size: 2em;

color: #4CAF50;

}

.thank-you-page p {

font-size: 1.2em;

color: #333;

}

**6. App.js**

This is the main component that defines the routes for /contact and /thank-you.

// src/App.js

import React from 'react';

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

import ContactForm from './components/ContactForm';

import ThankYouPage from './components/ThankYouPage';

const App = () => {

return (

<Router>

<div className="App">

<Switch>

<Route path="/contact" component={ContactForm} />

<Route path="/thank-you" component={ThankYouPage} />

</Switch>

</div>

</Router>

);

};

export default App;

**7. index.js**

This is the entry point where we render the app.

// src/index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

**8. index.css**

Global CSS for the application.

/\* src/index.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

**How It Works:**

1. The user accesses the /contact route to fill out the contact form.
2. Upon form submission, the user is redirected to /thank-you, which displays a confirmation message.
3. If the user attempts to submit the form with missing Name or Email, an error message is shown.

**Running the Project:**

1. Run the project using npm start.
2. Go to /contact to fill out the form. After submitting the form, you'll be redirected to /thank-you.

**Exercise 2: User Registration Form with Profile View**

**Objective**: Create a user registration form that captures user details and displays the submitted information on a profile page.

**Requirements:**

1. Create a registration form with the following fields:
   * Username
   * Password
   * Email
2. Validate that all fields are required.
3. Upon successful registration, redirect the user to a profile page showing the submitted details.

**Routes:**

1. /register - Displays the Registration Form.
2. /profile - Displays the Profile page with user information.

**Solution for Exercise 2**

**Project Structure:**

/src

/components

RegistrationForm.js

ProfilePage.js

/css

RegistrationForm.css

ProfilePage.css

App.js

index.js

**1. Install dependencies**

Make sure you have react-router-dom installed:

npm install react-router-dom

**2. RegistrationForm.js**

This component handles the registration form and redirects to the profile page upon successful form submission.

// src/components/RegistrationForm.js

import React, { useState } from 'react';

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

import './RegistrationForm.css';

const RegistrationForm = () => {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const [email, setEmail] = useState('');

const [error, setError] = useState('');

const history = useHistory();

const handleSubmit = (e) => {

e.preventDefault();

if (!username || !password || !email) {

setError('All fields are required');

} else {

const user = { username, password, email };

localStorage.setItem('user', JSON.stringify(user));

history.push('/profile');

}

};

return (

<div className="registration-form">

<h2>Register</h2>

{error && <p className="error">{error}</p>}

<form onSubmit={handleSubmit}>

<div>

<label>Username:</label>

<input

type="text"

value={username}

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

/>

</div>

<div>

<label>Password:</label>

<input

type="password"

value={password}

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

/>

</div>

<div>

<label>Email:</label>

<input

type="email"

value={email}

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

/>

</div>

<button type="submit">Register</button>

</form>

</div>

);

};

export default RegistrationForm;

**3. ProfilePage.js**

This component displays the profile with the registered user details.

// src/components/ProfilePage.js

import React from 'react';

import './ProfilePage.css';

const ProfilePage = () => {

const user = JSON.parse(localStorage.getItem('user'));

if (!user) {

return <p>User not registered. Please register first.</p>;

}

return (

<div className="profile-page">

<h2>Profile Page</h2>

<p><strong>Username:</strong> {user.username}</p>

<p><strong>Email:</strong> {user.email}</p>

</div>

);

};

export default ProfilePage;

**4. RegistrationForm.css**

CSS for the registration form.

/\* src/css/RegistrationForm.css \*/

.registration-form {

max-width: 400px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

.registration-form h2 {

text-align: center;

}

.registration-form form {

display: flex;

flex-direction: column;

}

.registration-form label {

margin-bottom: 5px;

font-weight: bold;

}

.registration-form input {

padding: 8px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

.registration-form button {

padding: 10px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

.registration-form button:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 14px;

text-align: center;

}

**5. ProfilePage.css**

CSS for the profile page.

/\* src/css/ProfilePage.css \*/

.profile-page {

max-width: 500px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

.profile-page h2 {

text-align: center;

}

.profile-page p {

font-size: 18px;

}

**6. App.js**

This is the main component where we define routes for registration and profile pages.

// src/App.js

import React from 'react';

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

import RegistrationForm from './components/RegistrationForm';

import ProfilePage from './components/ProfilePage';

const App = () => {

return (

<Router>

<div className="App">

<Switch>

<Route path="/register" component={RegistrationForm} />

<Route path="/profile" component={ProfilePage} />

</Switch>

</div>

</Router>

);

};

export default App;

**7. index.js**

This is where we render the app and set up routing.

// src/index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

**8. index.css**

Global styles for the application.

/\* src/index.css \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

**How It Works:**

1. The user accesses /register to fill out the registration form.
2. Upon successful registration, the form data is stored in localStorage, and the user is redirected to /profile.
3. The /profile route displays the submitted details.

**Running the Project:**

1. Run the project using npm start.
2. Navigate to /register to register, and then you'll be redirected to /profile to see the user details.

This structure should meet your requirements for a user registration form with validation, redirection, and a profile view.

**Exercise 3: Multi-Step Job Application Form**

**Objective**: Create a **multi-step job application form** where users fill in their personal information, education details, and work experience. After completing all steps, the user should see a summary of the application and confirm submission.

**Requirements:**

1. **Step 1: Personal Information**
   * Full Name (Required)
   * Email (Required)
   * Phone Number (Required)
2. **Step 2: Education**
   * Degree (Required)
   * University (Required)
   * Graduation Year (Required, must be a valid year)
3. **Step 3: Work Experience**
   * Company Name
   * Position
   * Years of Experience (Must be a positive number)
4. **Step 4: Review & Submit**
   * Display all the entered information.
   * Allow users to edit any section before final submission.
   * After submission, redirect to a **Thank You** page.

**Routes:**

1. /apply/personal - Displays the Personal Information form.
2. /apply/education - Displays the Education form.
3. /apply/experience - Displays the Work Experience form.
4. /apply/review - Displays the Review & Submit page.
5. /apply/thank-you - Displays the Thank You message after form submission.

**Solution**:

**App.js**

import React from 'react';

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

import PersonalInfoForm from './components/PersonalInfoForm';

import EducationForm from './components/EducationForm';

import WorkExperienceForm from './components/WorkExperienceForm';

import ReviewSubmit from './components/ReviewSubmit';

import ThankYou from './components/ThankYou';

import './css/App.css';

export default function App() {

return (

<Router>

<div className="app-container">

<Routes>

<Route path="/apply/personal" element={<PersonalInfoForm />} />

<Route path="/apply/education" element={<EducationForm />} />

<Route path="/apply/experience" element={<WorkExperienceForm />} />

<Route path="/apply/review" element={<ReviewSubmit />} />

<Route path="/apply/thank-you" element={<ThankYou />} />

<Route path="\*" element={<Navigate to="/apply/personal" replace />} />

</Routes>

</div>

</Router>

);

}

/\* App.css \*/

.app-container {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

min-height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

/\* components/PersonalInfoForm.css \*/

.personal-info-form {

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.personal-info-form input {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

.personal-info-form button {

background-color: #007bff;

color: white;

border: none;

padding: 10px 15px;

cursor: pointer;

border-radius: 4px;

}

/\* components/EducationForm.css \*/

.education-form {

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.education-form input {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

.education-form button {

background-color: #007bff;

color: white;

border: none;

padding: 10px 15px;

cursor: pointer;

border-radius: 4px;

}

/\* components/WorkExperienceForm.css \*/

.work-experience-form {

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.work-experience-form input {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

.work-experience-form button {

background-color: #007bff;

color: white;

border: none;

padding: 10px 15px;

cursor: pointer;

border-radius: 4px;

}

/\* components/ReviewSubmit.css \*/

.review-submit {

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.review-submit button {

background-color: #28a745;

color: white;

border: none;

padding: 10px 15px;

cursor: pointer;

border-radius: 4px;

margin-top: 20px;

}

/\* components/ThankYou.css \*/

.thank-you {

text-align: center;

background-color: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

width: 300px;

}

.thank-you h1 {

color: #28a745;

}

**components/PersonalInfoForm.js**

import React, { useState } from 'react';

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

import '../css/PersonalInfoForm.css';

export default function PersonalInfoForm() {

const navigate = useNavigate();

const [formData, setFormData] = useState({

firstName: '',

lastName: '',

email: '',

phone: ''

});

const handleChange = (e) => {

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

};

const handleSubmit = (e) => {

e.preventDefault();

localStorage.setItem('personalInfo', JSON.stringify(formData));

navigate('/apply/education');

};

return (

<form className="personal-info-form" onSubmit={handleSubmit}>

<h2>Personal Information</h2>

<input name="firstName" placeholder="First Name" value={formData.firstName} onChange={handleChange} required />

<input name="lastName" placeholder="Last Name" value={formData.lastName} onChange={handleChange} required />

<input name="email" type="email" placeholder="Email" value={formData.email} onChange={handleChange} required />

<input name="phone" placeholder="Phone Number" value={formData.phone} onChange={handleChange} required />

<button type="submit">Next</button>

</form>

);

}

**components/EducationForm.js**

import React, { useState } from 'react';

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

import '../css/EducationForm.css';

export default function EducationForm() {

const navigate = useNavigate();

const [educationData, setEducationData] = useState({

highestDegree: '',

university: '',

graduationYear: ''

});

const handleChange = (e) => {

setEducationData({ ...educationData, [e.target.name]: e.target.value });

};

const handleSubmit = (e) => {

e.preventDefault();

localStorage.setItem('educationInfo', JSON.stringify(educationData));

navigate('/apply/experience');

};

return (

<form className="education-form" onSubmit={handleSubmit}>

<h2>Education</h2>

<input name="highestDegree" placeholder="Highest Degree" value={educationData.highestDegree} onChange={handleChange} required />

<input name="university" placeholder="University" value={educationData.university} onChange={handleChange} required />

<input name="graduationYear" placeholder="Graduation Year" value={educationData.graduationYear} onChange={handleChange} required />

<button type="submit">Next</button>

</form>

);

}

**components/WorkExperienceForm.js**

import React, { useState } from 'react';

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

import '../css/WorkExperienceForm.css';

export default function WorkExperienceForm() {

const navigate = useNavigate();

const [experienceData, setExperienceData] = useState({

companyName: '',

position: '',

yearsOfExperience: ''

});

const handleChange = (e) => {

setExperienceData({ ...experienceData, [e.target.name]: e.target.value });

};

const handleSubmit = (e) => {

e.preventDefault();

localStorage.setItem('workExperience', JSON.stringify(experienceData));

navigate('/apply/review');

};

return (

<form className="work-experience-form" onSubmit={handleSubmit}>

<h2>Work Experience</h2>

<input name="companyName" placeholder="Company Name" value={experienceData.companyName} onChange={handleChange} required />

<input name="position" placeholder="Position" value={experienceData.position} onChange={handleChange} required />

<input name="yearsOfExperience" placeholder="Years of Experience" value={experienceData.yearsOfExperience} onChange={handleChange} required />

<button type="submit">Next</button>

</form>

);

}

**components/ReviewSubmit.js**

import React from 'react';

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

import '../css/ReviewSubmit.css';

export default function ReviewSubmit() {

const navigate = useNavigate();

const personalInfo = JSON.parse(localStorage.getItem('personalInfo'));

const educationInfo = JSON.parse(localStorage.getItem('educationInfo'));

const workExperience = JSON.parse(localStorage.getItem('workExperience'));

const handleSubmit = () => {

// Here you might send the data to a server

localStorage.clear(); // Clear data after submission

navigate('/apply/thank-you');

};

return (

<div className="review-submit">

<h2>Review Your Information</h2>

<h3>Personal Information</h3>

<p><strong>Name:</strong> {personalInfo.firstName} {personalInfo.lastName}</p>

<p><strong>Email:</strong> {personalInfo.email}</p>

<p><strong>Phone:</strong> {personalInfo.phone}</p>

<h3>Education</h3>

<p><strong>Degree:</strong> {educationInfo.highestDegree}</p>

<p><strong>University:</strong> {educationInfo.university}</p>

<p><strong>Graduation Year:</strong> {educationInfo.graduationYear}</p>

<h3>Work Experience</h3>

<p><strong>Company:</strong> {workExperience.companyName}</p>

<p><strong>Position:</strong> {workExperience.position}</p>

<p><strong>Years:</strong> {workExperience.yearsOfExperience}</p>

<button onClick={handleSubmit}>Submit Application</button>

</div>

);

}

**components/ThankYou.js**

import React from 'react';

import '../css/ThankYou.css';

export default function ThankYou() {

return (

<div className="thank-you">

<h1>Thank You!</h1>

<p>Your application has been submitted successfully.</p>

</div>

);

}

**Exercise 4: Multi-Step Registration Form with Confirmation**

**Objective:**

Create a multi-step user registration form that captures details such as personal information, contact details, and account settings. After submission, display a confirmation page showing the submitted data.

**Requirements:**

1. **Step 1:** Capture the user's personal information (name, age, gender).
2. **Step 2:** Capture contact information (email, phone).
3. **Step 3:** Account settings (username, password).
4. After all steps are completed, display a confirmation page showing the submitted information.
5. Use React Router to handle navigation between the steps.
6. Display appropriate validation messages for each step.

**Project Structure:**

/src

/components

Step1.js

Step2.js

Step3.js

Confirmation.js

ProgressBar.js

/css

Step1.css

Step2.css

Step3.css

Confirmation.css

ProgressBar.css

App.js

index.js

**Solution:**

**1. Install Dependencies:**

If you haven't already installed react-router-dom, use the following command:

npm install react-router-dom

**2. Step1.js (Personal Information):**

This component captures the user's personal information.

// src/components/Step1.js

import React, { useState } from 'react';

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

import './Step1.css';

const Step1 = ({ setStepData }) => {

const [name, setName] = useState('');

const [age, setAge] = useState('');

const [gender, setGender] = useState('');

const history = useHistory();

const handleNext = () => {

if (!name || !age || !gender) {

alert('All fields are required');

return;

}

setStepData({ name, age, gender });

history.push('/step2');

};

return (

<div className="step1">

<h2>Step 1: Personal Information</h2>

<div>

<label>Name:</label>

<input type="text" value={name} onChange={(e) => setName(e.target.value)} />

</div>

<div>

<label>Age:</label>

<input type="number" value={age} onChange={(e) => setAge(e.target.value)} />

</div>

<div>

<label>Gender:</label>

<select value={gender} onChange={(e) => setGender(e.target.value)}>

<option value="">Select Gender</option>

<option value="male">Male</option>

<option value="female">Female</option>

</select>

</div>

<button onClick={handleNext}>Next</button>

</div>

);

};

export default Step1;

**3. Step2.js (Contact Information):**

This component captures the user's contact details.

// src/components/Step2.js

import React, { useState } from 'react';

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

import './Step2.css';

const Step2 = ({ setStepData }) => {

const [email, setEmail] = useState('');

const [phone, setPhone] = useState('');

const history = useHistory();

const handleNext = () => {

if (!email || !phone) {

alert('All fields are required');

return;

}

setStepData({ email, phone });

history.push('/step3');

};

return (

<div className="step2">

<h2>Step 2: Contact Information</h2>

<div>

<label>Email:</label>

<input type="email" value={email} onChange={(e) => setEmail(e.target.value)} />

</div>

<div>

<label>Phone:</label>

<input type="text" value={phone} onChange={(e) => setPhone(e.target.value)} />

</div>

<button onClick={handleNext}>Next</button>

</div>

);

};

export default Step2;

**4. Step3.js (Account Settings):**

This component captures the account settings (username and password).

// src/components/Step3.js

import React, { useState } from 'react';

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

import './Step3.css';

const Step3 = ({ setStepData }) => {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const history = useHistory();

const handleNext = () => {

if (!username || !password) {

alert('All fields are required');

return;

}

setStepData({ username, password });

history.push('/confirmation');

};

return (

<div className="step3">

<h2>Step 3: Account Settings</h2>

<div>

<label>Username:</label>

<input type="text" value={username} onChange={(e) => setUsername(e.target.value)} />

</div>

<div>

<label>Password:</label>

<input type="password" value={password} onChange={(e) => setPassword(e.target.value)} />

</div>

<button onClick={handleNext}>Finish</button>

</div>

);

};

export default Step3;

**5. Confirmation.js (Display Submitted Data):**

This component shows the final confirmation with all the submitted data.

// src/components/Confirmation.js

import React from 'react';

import './Confirmation.css';

const Confirmation = ({ userData }) => {

return (

<div className="confirmation">

<h2>Registration Complete</h2>

<h3>Your Submitted Data</h3>

<p><strong>Name:</strong> {userData.name}</p>

<p><strong>Age:</strong> {userData.age}</p>

<p><strong>Gender:</strong> {userData.gender}</p>

<p><strong>Email:</strong> {userData.email}</p>

<p><strong>Phone:</strong> {userData.phone}</p>

<p><strong>Username:</strong> {userData.username}</p>

</div>

);

};

export default Confirmation;

**6. ProgressBar.js (Optional, Track Progress):**

This component displays the progress of the form steps.

// src/components/ProgressBar.js

import React from 'react';

import './ProgressBar.css';

const ProgressBar = ({ step }) => {

return (

<div className="progress-bar">

<div className={`step ${step >= 1 ? 'completed' : ''}`}>Step 1</div>

<div className={`step ${step >= 2 ? 'completed' : ''}`}>Step 2</div>

<div className={`step ${step >= 3 ? 'completed' : ''}`}>Step 3</div>

</div>

);

};

export default ProgressBar;

**7. App.js (Main App with Routing):**

This file handles routing and state management between the steps.

// src/App.js

import React, { useState } from 'react';

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

import Step1 from './components/Step1';

import Step2 from './components/Step2';

import Step3 from './components/Step3';

import Confirmation from './components/Confirmation';

import ProgressBar from './components/ProgressBar';

import './App.css';

const App = () => {

const [userData, setUserData] = useState({});

const [step, setStep] = useState(1);

const setStepData = (data) => {

setUserData({ ...userData, ...data });

setStep(step + 1);

};

return (

<Router>

<div className="App">

<ProgressBar step={step} />

<Switch>

<Route path="/step1" render={() => <Step1 setStepData={setStepData} />} />

<Route path="/step2" render={() => <Step2 setStepData={setStepData} />} />

<Route path="/step3" render={() => <Step3 setStepData={setStepData} />} />

<Route path="/confirmation" render={() => <Confirmation userData={userData} />} />

</Switch>

</div>

</Router>

);

};

export default App;

**8. index.js (Entry Point):**

The entry point where the React app is rendered.

// src/index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

ReactDOM.render(

<React.StrictMode>

<App />

</React.StrictMode>,

document.getElementById('root')

);

**9. CSS (for each component):**

You can customize the CSS files (Step1.css, Step2.css, Step3.css, Confirmation.css, and ProgressBar.css) to style each component as desired.

**How It Works:**

1. The user navigates through three steps of the form (/step1, /step2, /step3), entering personal, contact, and account information.
2. After completing all the steps, the user is redirected to /confirmation, where all the data they entered is displayed.
3. A progress bar tracks the user's position in the registration process.

This solution involves 5 components (Step1, Step2, Step3, Confirmation, and ProgressBar), and it uses React Router for navigation between the steps and confirmation page.

Here are the CSS files for each of the components mentioned in the solution. Each CSS file is designed to provide basic styling for the components, ensuring that they are visually distinct and organized.

**1. Step1.css (Personal Information)**

/\* src/css/Step1.css \*/

.step1 {

max-width: 500px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.step1 h2 {

text-align: center;

color: #333;

}

.step1 label {

font-weight: bold;

display: block;

margin-bottom: 8px;

}

.step1 input, .step1 select {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

.step1 button {

width: 100%;

padding: 12px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.step1 button:hover {

background-color: #45a049;

}

**2. Step2.css (Contact Information)**

/\* src/css/Step2.css \*/

.step2 {

max-width: 500px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.step2 h2 {

text-align: center;

color: #333;

}

.step2 label {

font-weight: bold;

display: block;

margin-bottom: 8px;

}

.step2 input {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

.step2 button {

width: 100%;

padding: 12px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.step2 button:hover {

background-color: #45a049;

}

**3. Step3.css (Account Settings)**

/\* src/css/Step3.css \*/

.step3 {

max-width: 500px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.step3 h2 {

text-align: center;

color: #333;

}

.step3 label {

font-weight: bold;

display: block;

margin-bottom: 8px;

}

.step3 input {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

.step3 button {

width: 100%;

padding: 12px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.step3 button:hover {

background-color: #45a049;

}

**4. Confirmation.css (Confirmation Page)**

/\* src/css/Confirmation.css \*/

.confirmation {

max-width: 600px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

text-align: center;

}

.confirmation h2 {

color: #4CAF50;

}

.confirmation h3 {

margin-top: 20px;

font-size: 20px;

}

.confirmation p {

font-size: 16px;

margin: 5px 0;

}

**5. ProgressBar.css (Progress Bar for Steps)**

/\* src/css/ProgressBar.css \*/

.progress-bar {

display: flex;

justify-content: space-between;

margin-bottom: 20px;

}

.progress-bar .step {

padding: 8px 12px;

background-color: #ddd;

border-radius: 5px;

text-align: center;

width: 30%;

}

.progress-bar .step.completed {

background-color: #4CAF50;

color: white;

}

.progress-bar .step:hover {

cursor: pointer;

background-color: #76c76f;

}

**Exercise 5: Job Application Form**

**Objective:**

Create a Job Application Form that captures user information and displays the submitted details on a confirmation page.

**Requirements:**

1. Create a form with the following fields:
   * Full Name
   * Email
   * Phone Number
   * Position Applied For
   * Resume (file upload)
2. Validate that Full Name, Email, and Position Applied For are required.
3. Upon successful submission, redirect the user to a confirmation page that displays the submitted information.

**Routes:**

1. /job-application - Displays the Job Application Form.
2. /application-confirmation - Displays a confirmation page with the submitted application details.

**Components:**

1. **JobApplicationForm.js** - The form where users enter their job application details.
2. **ConfirmationPage.js** - Displays a confirmation page with the submitted data.
3. **ProgressBar.js** - A simple progress bar that shows the current step in the application process.
4. **Navbar.js** - A simple navigation bar for accessing the form and confirmation page.
5. **Footer.js** - A footer component with basic copyright information.

**1. JobApplicationForm.js (Job Application Form)**

import React, { useState } from 'react';

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

import './JobApplicationForm.css';

const JobApplicationForm = () => {

const [formData, setFormData] = useState({

fullName: '',

email: '',

phone: '',

position: '',

resume: null,

});

const [errors, setErrors] = useState({});

const history = useHistory();

const handleInputChange = (e) => {

const { name, value } = e.target;

setFormData((prevData) => ({

...prevData,

[name]: value,

}));

};

const handleFileChange = (e) => {

setFormData((prevData) => ({

...prevData,

resume: e.target.files[0],

}));

};

const validateForm = () => {

const newErrors = {};

if (!formData.fullName) newErrors.fullName = 'Full Name is required';

if (!formData.email) newErrors.email = 'Email is required';

if (!formData.position) newErrors.position = 'Position Applied For is required';

return newErrors;

};

const handleSubmit = (e) => {

e.preventDefault();

const newErrors = validateForm();

if (Object.keys(newErrors).length === 0) {

history.push('/application-confirmation', { formData });

} else {

setErrors(newErrors);

}

};

return (

<div className="job-application-form">

<h2>Job Application Form</h2>

<form onSubmit={handleSubmit}>

<label>

Full Name

<input

type="text"

name="fullName"

value={formData.fullName}

onChange={handleInputChange}

/>

{errors.fullName && <span className="error">{errors.fullName}</span>}

</label>

<label>

Email

<input

type="email"

name="email"

value={formData.email}

onChange={handleInputChange}

/>

{errors.email && <span className="error">{errors.email}</span>}

</label>

<label>

Phone Number

<input

type="text"

name="phone"

value={formData.phone}

onChange={handleInputChange}

/>

</label>

<label>

Position Applied For

<input

type="text"

name="position"

value={formData.position}

onChange={handleInputChange}

/>

{errors.position && <span className="error">{errors.position}</span>}

</label>

<label>

Resume

<input type="file" onChange={handleFileChange} />

</label>

<button type="submit">Submit</button>

</form>

</div>

);

};

export default JobApplicationForm;

**2. ConfirmationPage.js (Confirmation Page)**

import React from 'react';

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

import './ConfirmationPage.css';

const ConfirmationPage = () => {

const { state } = useLocation();

const { formData } = state;

return (

<div className="confirmation-page">

<h2>Application Submitted Successfully!</h2>

<div className="confirmation-details">

<p><strong>Full Name:</strong> {formData.fullName}</p>

<p><strong>Email:</strong> {formData.email}</p>

<p><strong>Phone:</strong> {formData.phone}</p>

<p><strong>Position Applied For:</strong> {formData.position}</p>

<p><strong>Resume:</strong> {formData.resume ? formData.resume.name : 'No file uploaded'}</p>

</div>

</div>

);

};

export default ConfirmationPage;

**3. ProgressBar.js (Progress Bar Component)**

import React from 'react';

import './ProgressBar.css';

const ProgressBar = ({ currentStep }) => {

return (

<div className="progress-bar">

<div className={`step ${currentStep >= 1 ? 'completed' : ''}`}>Step 1</div>

<div className={`step ${currentStep >= 2 ? 'completed' : ''}`}>Step 2</div>

<div className={`step ${currentStep >= 3 ? 'completed' : ''}`}>Step 3</div>

</div>

);

};

export default ProgressBar;

**4. Navbar.js (Navbar Component)**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<Link to="/job-application">Job Application</Link>

<Link to="/application-confirmation">Confirmation</Link>

</nav>

);

};

export default Navbar;

**5. Footer.js (Footer Component)**

import React from 'react';

import './Footer.css';

const Footer = () => {

return (

<footer className="footer">

<p>&copy; 2025 Job Application Inc.</p>

</footer>

);

};

export default Footer;

**1. JobApplicationForm.css (Styling for the Form)**

/\* src/css/JobApplicationForm.css \*/

.job-application-form {

max-width: 600px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.job-application-form h2 {

text-align: center;

color: #333;

}

.job-application-form label {

font-weight: bold;

display: block;

margin-bottom: 8px;

}

.job-application-form input {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

.job-application-form button {

width: 100%;

padding: 12px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.job-application-form button:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 12px;

}

**2. ConfirmationPage.css (Styling for the Confirmation Page)**

/\* src/css/ConfirmationPage.css \*/

.confirmation-page {

max-width: 600px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

text-align: center;

}

.confirmation-page h2 {

color: #4CAF50;

}

.confirmation-page p {

font-size: 16px;

margin: 5px 0;

}

**3. ProgressBar.css (Styling for the Progress Bar)**

/\* src/css/ProgressBar.css \*/

.progress-bar {

display: flex;

justify-content: space-between;

margin-bottom: 20px;

}

.progress-bar .step {

padding: 8px 12px;

background-color: #ddd;

border-radius: 5px;

text-align: center;

width: 30%;

}

.progress-bar .step.completed {

background-color: #4CAF50;

color: white;

}

.progress-bar .step:hover {

cursor: pointer;

background-color: #76c76f;

}

**4. Navbar.css (Styling for the Navbar)**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

color: white;

padding: 10px 0;

text-align: center;

}

.navbar a {

color: white;

padding: 10px 15px;

text-decoration: none;

font-weight: bold;

}

.navbar a:hover {

background-color: #444;

border-radius: 5px;

}

**5. Footer.css (Styling for the Footer)**

/\* src/css/Footer.css \*/

.footer {

background-color: #333;

color: white;

padding: 10px 0;

text-align: center;

}

**How to Use:**

1. Ensure that the components are imported and used in your App.js file.
2. Import the CSS files into their respective component files.
3. Ensure you have React Router set up to manage the /job-application and /application-confirmation routes.

This exercise allows users to submit job application data, which is then confirmed on the next page, making it a practical exercise for handling form validation and data submission with React.

**Exercise 6: Task Management Application**

**Objective:**

Create a Task Management Application that allows users to add tasks, mark them as completed, and view all their tasks in a list.

**Requirements:**

1. Create a form with the following fields:
   * Task Name
   * Due Date
   * Priority Level (Low, Medium, High)
2. Display a list of tasks below the form with the ability to:
   * Mark tasks as completed.
   * Delete tasks.
3. Task list should be filtered by priority levels (e.g., Low, Medium, High).

**Routes:**

1. /task-manager - Displays the Task Manager form and task list.
2. /task-detail - Displays the details of a selected task.

**Components:**

1. **TaskForm.js** - A form for adding new tasks.
2. **TaskList.js** - Displays the list of tasks and handles task actions like marking as completed and deleting.
3. **TaskItem.js** - Represents each individual task in the list.
4. **TaskDetail.js** - Shows the details of a selected task.
5. **Navbar.js** - A navigation bar for navigating between pages.

**1. TaskForm.js (Task Management Form)**

import React, { useState } from 'react';

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

import './TaskForm.css';

const TaskForm = ({ addTask }) => {

const [taskData, setTaskData] = useState({

taskName: '',

dueDate: '',

priority: 'Low',

});

const [errors, setErrors] = useState({});

const history = useHistory();

const handleInputChange = (e) => {

const { name, value } = e.target;

setTaskData((prevData) => ({

...prevData,

[name]: value,

}));

};

const validateForm = () => {

const newErrors = {};

if (!taskData.taskName) newErrors.taskName = 'Task Name is required';

if (!taskData.dueDate) newErrors.dueDate = 'Due Date is required';

return newErrors;

};

const handleSubmit = (e) => {

e.preventDefault();

const newErrors = validateForm();

if (Object.keys(newErrors).length === 0) {

addTask(taskData);

history.push('/task-manager');

} else {

setErrors(newErrors);

}

};

return (

<div className="task-form">

<h2>Add a New Task</h2>

<form onSubmit={handleSubmit}>

<label>

Task Name

<input

type="text"

name="taskName"

value={taskData.taskName}

onChange={handleInputChange}

/>

{errors.taskName && <span className="error">{errors.taskName}</span>}

</label>

<label>

Due Date

<input

type="date"

name="dueDate"

value={taskData.dueDate}

onChange={handleInputChange}

/>

{errors.dueDate && <span className="error">{errors.dueDate}</span>}

</label>

<label>

Priority

<select

name="priority"

value={taskData.priority}

onChange={handleInputChange}

>

<option value="Low">Low</option>

<option value="Medium">Medium</option>

<option value="High">High</option>

</select>

</label>

<button type="submit">Add Task</button>

</form>

</div>

);

};

export default TaskForm;

**2. TaskList.js (Task List)**

import React from 'react';

import TaskItem from './TaskItem';

import './TaskList.css';

const TaskList = ({ tasks, markCompleted, deleteTask }) => {

return (

<div className="task-list">

<h2>Task List</h2>

<div className="task-filters">

<label>

Filter by Priority:

<select>

<option value="All">All</option>

<option value="Low">Low</option>

<option value="Medium">Medium</option>

<option value="High">High</option>

</select>

</label>

</div>

<ul>

{tasks.map((task) => (

<TaskItem

key={task.id}

task={task}

markCompleted={markCompleted}

deleteTask={deleteTask}

/>

))}

</ul>

</div>

);

};

export default TaskList;

**3. TaskItem.js (Individual Task)**

import React from 'react';

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

import './TaskItem.css';

const TaskItem = ({ task, markCompleted, deleteTask }) => {

return (

<li className={`task-item ${task.completed ? 'completed' : ''}`}>

<div>

<h3>{task.taskName}</h3>

<p>Due: {task.dueDate}</p>

<p>Priority: {task.priority}</p>

</div>

<div className="task-actions">

<button onClick={() => markCompleted(task.id)}>

{task.completed ? 'Unmark' : 'Mark as Completed'}

</button>

<button onClick={() => deleteTask(task.id)}>Delete</button>

<Link to={`/task-detail/${task.id}`}>View Details</Link>

</div>

</li>

);

};

export default TaskItem;

**4. TaskDetail.js (Task Detail Page)**

import React from 'react';

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

import './TaskDetail.css';

const TaskDetail = ({ tasks }) => {

const { taskId } = useParams();

const task = tasks.find((task) => task.id === parseInt(taskId));

return (

<div className="task-detail">

<h2>Task Details</h2>

{task ? (

<div>

<p><strong>Task Name:</strong> {task.taskName}</p>

<p><strong>Due Date:</strong> {task.dueDate}</p>

<p><strong>Priority:</strong> {task.priority}</p>

<p><strong>Status:</strong> {task.completed ? 'Completed' : 'Pending'}</p>

</div>

) : (

<p>Task not found!</p>

)}

</div>

);

};

export default TaskDetail;

**5. Navbar.js (Navbar Component)**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<Link to="/task-manager">Task Manager</Link>

<Link to="/task-detail">Task Detail</Link>

</nav>

);

};

export default Navbar;

**1. TaskForm.css (Styling for the Task Form)**

/\* src/css/TaskForm.css \*/

.task-form {

max-width: 600px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.task-form h2 {

text-align: center;

color: #333;

}

.task-form label {

font-weight: bold;

display: block;

margin-bottom: 8px;

}

.task-form input,

.task-form select {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

.task-form button {

width: 100%;

padding: 12px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

.task-form button:hover {

background-color: #45a049;

}

.error {

color: red;

font-size: 12px;

}

**2. TaskList.css (Styling for the Task List)**

/\* src/css/TaskList.css \*/

.task-list {

max-width: 800px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.task-list h2 {

text-align: center;

color: #333;

}

.task-list ul {

list-style-type: none;

padding: 0;

}

.task-list .task-actions {

display: flex;

justify-content: space-between;

}

.task-list button {

padding: 8px 12px;

background-color: #ff6666;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.task-list button:hover {

background-color: #ff4d4d;

}

.task-list select {

padding: 8px;

}

**3. TaskItem.css (Styling for Individual Task Items)**

/\* src/css/TaskItem.css \*/

.task-item {

display: flex;

justify-content: space-between;

padding: 15px;

border-bottom: 1px solid #ddd;

}

.task-item.completed {

background-color: #d4edda;

}

.task-item h3 {

margin: 0;

}

.task-actions {

display: flex;

justify-content: space-between;

align-items: center;

}

.task-actions button {

padding: 8px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.task-actions button:hover {

background-color: #0056b3;

}

.task-actions a {

text-decoration: none;

color: #007bff;

}

**4. TaskDetail.css (Styling for Task Detail)**

/\* src/css/TaskDetail.css \*/

.task-detail {

max-width: 600px;

margin: 20px auto;

padding: 20px;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

}

.task-detail h2 {

text-align: center;

color: #333;

}

.task-detail p {

font-size: 16px;

margin: 10px 0;

}

**5. Navbar.css (Styling for the Navbar)**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

padding: 15px;

display: flex;

justify-content: space-around;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 18px;

}

.navbar a:hover {

color: #ddd;

}

**TaskManagerPage.js**

import React, { useState } from 'react';

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

import TaskForm from './TaskForm';

import TaskList from './TaskList';

import Navbar from './Navbar';

import './TaskManagerPage.css';

const TaskManagerPage = () => {

const [tasks, setTasks] = useState([]);

const history = useHistory();

const addTask = (taskData) => {

setTasks([

...tasks,

{

...taskData,

id: tasks.length + 1,

completed: false,

},

]);

};

const markCompleted = (taskId) => {

setTasks(tasks.map((task) =>

task.id === taskId

? { ...task, completed: !task.completed }

: task

));

};

const deleteTask = (taskId) => {

setTasks(tasks.filter((task) => task.id !== taskId));

};

return (

<div>

<Navbar />

<TaskForm addTask={addTask} />

<TaskList tasks={tasks} markCompleted={markCompleted} deleteTask={deleteTask} />

</div>

);

};

export default TaskManagerPage;

This setup should give you a functional task manager application with task addition, completion, and deletion, styled in a clean, user-friendly way!

**App.js**

import React from 'react';

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

import TaskManagerPage from './components/TaskManagerPage';

import TaskDetail from './components/TaskDetail';

import './App.css';

const App = () => {

return (

<Router>

<div className="app">

<Switch>

<Route path="/task-manager" exact component={TaskManagerPage} />

<Route path="/task-detail/:taskId" component={TaskDetail} />

<Route path="/" exact component={TaskManagerPage} />

</Switch>

</div>

</Router>

);

};

export default App;

**App.css (Basic Styling)**

/\* src/css/App.css \*/

.app {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

padding: 20px;

}

h1 {

text-align: center;

color: #333;

}

This App.js file sets up your React Router with routes for /task-manager and /task-detail/:taskId, making sure to handle navigation between pages. The TaskManagerPage route renders your task management system, and the TaskDetail route renders the details of an individual task. The default route (/) also points to the task manager page.

Make sure to adjust the imports according to your file structure.

**Exercise 7:**

Create a simple Blog application with multiple components to display blog posts, allow adding new posts, and view post details.

**Features:**

1. **HomePage** - Display a list of blog posts.
2. **PostPage** - Display the details of a single post.
3. **AddPostPage** - A page with a form to add a new post.
4. **Navbar** - A navigation bar for the application.
5. **PostCard** - A component that displays a blog post preview in the list.

**App.js**

import React from 'react';

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

import HomePage from './components/HomePage';

import PostPage from './components/PostPage';

import AddPostPage from './components/AddPostPage';

import Navbar from './components/Navbar';

import './App.css';

const App = () => {

return (

<Router>

<div className="app">

<Navbar />

<Switch>

<Route path="/" exact component={HomePage} />

<Route path="/post/:postId" component={PostPage} />

<Route path="/add-post" component={AddPostPage} />

</Switch>

</div>

</Router>

);

};

export default App;

**Navbar.js**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<ul>

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

<li><Link to="/add-post">Add Post</Link></li>

</ul>

</nav>

);

};

export default Navbar;

**Navbar.css**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

padding: 10px;

}

.navbar ul {

list-style-type: none;

display: flex;

justify-content: space-around;

}

.navbar li {

margin: 0 15px;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 18px;

}

.navbar a:hover {

color: #ddd;

}

**HomePage.js**

import React, { useState } from 'react';

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

import PostCard from './PostCard';

import './HomePage.css';

const HomePage = () => {

const [posts, setPosts] = useState([

{ id: 1, title: 'My First Blog Post', preview: 'This is a preview of my first post...', content: 'Full content of the first post.' },

{ id: 2, title: 'Another Blog Post', preview: 'Here is a preview of another blog post...', content: 'Full content of the second post.' },

]);

return (

<div className="home-page">

<h1>Blog Posts</h1>

<div className="post-list">

{posts.map(post => (

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

<PostCard post={post} />

</Link>

))}

</div>

</div>

);

};

export default HomePage;

**HomePage.css**

/\* src/css/HomePage.css \*/

.home-page {

text-align: center;

}

.post-list {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 20px;

}

**PostCard.js**

import React from 'react';

import './PostCard.css';

const PostCard = ({ post }) => {

return (

<div className="post-card">

<h3>{post.title}</h3>

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

</div>

);

};

export default PostCard;

**PostCard.css**

/\* src/css/PostCard.css \*/

.post-card {

background-color: #fff;

padding: 15px;

border-radius: 8px;

width: 200px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

transition: transform 0.2s;

}

.post-card:hover {

transform: scale(1.05);

}

**PostPage.js**

import React from 'react';

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

import './PostPage.css';

const PostPage = () => {

const { postId } = useParams();

const post = {

1: { title: 'My First Blog Post', content: 'Full content of the first post.' },

2: { title: 'Another Blog Post', content: 'Full content of the second post.' },

}[postId];

return (

<div className="post-page">

<h2>{post.title}</h2>

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

</div>

);

};

export default PostPage;

**PostPage.css**

/\* src/css/PostPage.css \*/

.post-page {

text-align: center;

max-width: 800px;

margin: 0 auto;

}

h2 {

margin-bottom: 20px;

}

**AddPostPage.js**

import React, { useState } from 'react';

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

import './AddPostPage.css';

const AddPostPage = () => {

const [title, setTitle] = useState('');

const [content, setContent] = useState('');

const history = useHistory();

const handleSubmit = (e) => {

e.preventDefault();

// Here, you could add a new post to the list (in this example, we're just redirecting)

history.push('/');

};

return (

<div className="add-post-page">

<h2>Add a New Post</h2>

<form onSubmit={handleSubmit}>

<div>

<label htmlFor="title">Title</label>

<input

id="title"

type="text"

value={title}

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

required

/>

</div>

<div>

<label htmlFor="content">Content</label>

<textarea

id="content"

value={content}

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

required

/>

</div>

<button type="submit">Add Post</button>

</form>

</div>

);

};

export default AddPostPage;

**AddPostPage.css**

/\* src/css/AddPostPage.css \*/

.add-post-page {

text-align: center;

max-width: 600px;

margin: 0 auto;

}

form {

display: flex;

flex-direction: column;

gap: 20px;

}

input, textarea {

padding: 10px;

border: 1px solid #ddd;

border-radius: 4px;

}

button {

padding: 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

}

button:hover {

background-color: #555;

}

**App.css**

/\* src/css/App.css \*/

.app {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

padding: 20px;

}

h1 {

color: #333;

}

This application now consists of 5 main components:

1. **HomePage**: Displays a list of blog posts with a preview.
2. **PostPage**: Shows the full content of a single post.
3. **AddPostPage**: A form to add new blog posts.
4. **Navbar**: A navigation bar for easy routing.
5. **PostCard**: Displays a preview of each post.

Make sure to organize your components and CSS files according to the provided file structure. This example offers a blog application with routing, form handling, and a simple task management feature for displaying and adding posts!

**Exercise 8**

**Objective:**

Create a To-Do List application with multiple components such as adding tasks, viewing tasks, and marking tasks as completed. This will involve routing, forms, and dynamic content rendering.

**Features:**

1. **App**: Main application file with routing.
2. **Navbar**: A navigation bar with links to Home and About pages.
3. **HomePage**: Displays the list of tasks and allows for task creation.
4. **TaskList**: Displays a list of tasks.
5. **TaskItem**: Displays each task with options to mark as completed or delete.
6. **AddTaskForm**: A form to add a new task.
7. **TaskFilter**: Allows users to filter tasks based on their status (all, completed, or pending).
8. **TaskCounter**: Displays the count of tasks (all, completed, and pending).
9. **AboutPage**: Displays information about the app.
10. **Footer**: Displays a footer with basic information.

**App.js**

import React from 'react';

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

import HomePage from './components/HomePage';

import AboutPage from './components/AboutPage';

import Navbar from './components/Navbar';

import Footer from './components/Footer';

import './App.css';

const App = () => {

return (

<Router>

<div className="app">

<Navbar />

<Switch>

<Route path="/" exact component={HomePage} />

<Route path="/about" component={AboutPage} />

</Switch>

<Footer />

</div>

</Router>

);

};

export default App;

**Navbar.js**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<ul>

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

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

</ul>

</nav>

);

};

export default Navbar;

**Navbar.css**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

padding: 10px;

}

.navbar ul {

list-style-type: none;

display: flex;

justify-content: center;

gap: 20px;

}

.navbar li {

margin: 0;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 18px;

}

.navbar a:hover {

color: #ddd;

}

**HomePage.js**

import React, { useState } from 'react';

import AddTaskForm from './AddTaskForm';

import TaskList from './TaskList';

import TaskFilter from './TaskFilter';

import TaskCounter from './TaskCounter';

import './HomePage.css';

const HomePage = () => {

const [tasks, setTasks] = useState([]);

const [filter, setFilter] = useState('all');

const addTask = (task) => {

setTasks([...tasks, task]);

};

const deleteTask = (id) => {

setTasks(tasks.filter(task => task.id !== id));

};

const toggleTaskCompletion = (id) => {

setTasks(

tasks.map(task =>

task.id === id ? { ...task, completed: !task.completed } : task

)

);

};

const filteredTasks = tasks.filter(task =>

filter === 'all' ? true : filter === 'completed' ? task.completed : !task.completed

);

return (

<div className="home-page">

<h1>To-Do List</h1>

<TaskCounter tasks={tasks} />

<TaskFilter setFilter={setFilter} />

<AddTaskForm addTask={addTask} />

<TaskList tasks={filteredTasks} toggleTaskCompletion={toggleTaskCompletion} deleteTask={deleteTask} />

</div>

);

};

export default HomePage;

**HomePage.css**

/\* src/css/HomePage.css \*/

.home-page {

text-align: center;

max-width: 800px;

margin: 0 auto;

}

h1 {

color: #333;

}

button {

padding: 10px 20px;

margin-top: 20px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

**AddTaskForm.js**

import React, { useState } from 'react';

import './AddTaskForm.css';

const AddTaskForm = ({ addTask }) => {

const [taskText, setTaskText] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (taskText.trim()) {

addTask({ id: Date.now(), text: taskText, completed: false });

setTaskText('');

}

};

return (

<form onSubmit={handleSubmit} className="add-task-form">

<input

type="text"

placeholder="Add a new task"

value={taskText}

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

/>

<button type="submit">Add Task</button>

</form>

);

};

export default AddTaskForm;

**AddTaskForm.css**

/\* src/css/AddTaskForm.css \*/

.add-task-form {

display: flex;

justify-content: center;

margin-bottom: 20px;

}

.add-task-form input {

padding: 10px;

width: 60%;

margin-right: 10px;

border: 1px solid #ddd;

border-radius: 4px;

}

.add-task-form button {

padding: 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

.add-task-form button:hover {

background-color: #555;

}

**TaskList.js**

import React from 'react';

import TaskItem from './TaskItem';

import './TaskList.css';

const TaskList = ({ tasks, toggleTaskCompletion, deleteTask }) => {

return (

<ul className="task-list">

{tasks.map(task => (

<TaskItem

key={task.id}

task={task}

toggleTaskCompletion={toggleTaskCompletion}

deleteTask={deleteTask}

/>

))}

</ul>

);

};

export default TaskList;

**TaskList.css**

/\* src/css/TaskList.css \*/

.task-list {

list-style-type: none;

padding: 0;

}

.task-list li {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px;

background-color: #f9f9f9;

border-bottom: 1px solid #ddd;

}

.task-list li.completed {

text-decoration: line-through;

background-color: #d4edda;

}

**TaskItem.js**

import React from 'react';

import './TaskItem.css';

const TaskItem = ({ task, toggleTaskCompletion, deleteTask }) => {

return (

<li className={task.completed ? 'completed' : ''}>

<span onClick={() => toggleTaskCompletion(task.id)}>{task.text}</span>

<button onClick={() => deleteTask(task.id)}>Delete</button>

</li>

);

};

export default TaskItem;

**TaskItem.css**

/\* src/css/TaskItem.css \*/

.task-item {

padding: 10px;

display: flex;

justify-content: space-between;

background-color: #f4f4f4;

border-radius: 4px;

}

.task-item button {

background-color: #e53e3e;

color: white;

border: none;

padding: 5px 10px;

cursor: pointer;

}

.task-item button:hover {

background-color: #c53030;

}

**TaskFilter.js**

import React from 'react';

import './TaskFilter.css';

const TaskFilter = ({ setFilter }) => {

return (

<div className="task-filter">

<button onClick={() => setFilter('all')}>All</button>

<button onClick={() => setFilter('completed')}>Completed</button>

<button onClick={() => setFilter('pending')}>Pending</button>

</div>

);

};

export default TaskFilter;

**TaskFilter.css**

/\* src/css/TaskFilter.css \*/

.task-filter {

margin: 20px;

}

.task-filter button {

margin: 5px;

padding: 10px;

background-color: #444;

color: white;

border: none;

cursor: pointer;

}

.task-filter button:hover {

background-color: #666;

}

**TaskCounter.js**

import React from 'react';

import './TaskCounter.css';

const TaskCounter = ({ tasks }) => {

const totalTasks = tasks.length;

const completedTasks = tasks.filter(task => task.completed).length;

const pendingTasks = totalTasks - completedTasks;

return (

<div className="task-counter">

<p>Total Tasks: {totalTasks}</p>

<p>Completed: {completedTasks}</p>

<p>Pending: {pendingTasks}</p>

</div>

);

};

export default TaskCounter;

**TaskCounter.css**

/\* src/css/TaskCounter.css \*/

.task-counter {

margin: 20px;

font-size: 18px;

}

.task-counter p {

margin: 5px 0;

}

**AboutPage.js**

import React from 'react';

import './AboutPage.css';

const AboutPage = () => {

return (

<div className="about-page">

<h2>About To-Do List App</h2>

<p>This app helps you manage your tasks efficiently.</p>

</div>

);

};

export default AboutPage;

**AboutPage.css**

/\* src/css/AboutPage.css \*/

.about-page {

text-align: center;

margin-top: 50px;

}

.about-page h2 {

color: #333;

}

.about-page p {

color: #666;

}

**Footer.js**

import React from 'react';

import './Footer.css';

const Footer = () => {

return (

<footer className="footer">

<p>&copy; 2025 To-Do List App</p>

</footer>

);

};

export default Footer;

**Footer.css**

/\* src/css/Footer.css \*/

.footer {

text-align: center;

padding: 20px;

background-color: #333;

color: white;

}

**Exercise 9:**

**Objective:**

Create a Blog Application that allows users to view, add, and edit posts.

**Features:**

1. **App**: Main application with routing.
2. **Navbar**: A navigation bar with links to Home and About pages.
3. **HomePage**: Displays a list of blog posts.
4. **PostItem**: Displays each blog post in the list.
5. **PostDetails**: Displays the full content of a single post when clicked.
6. **AddPostForm**: Form to add a new blog post.
7. **EditPostForm**: Form to edit an existing blog post.
8. **PostCounter**: Displays the total number of posts.
9. **AboutPage**: Displays information about the blog.
10. **Footer**: Displays basic footer information.

**App.js**

import React from 'react';

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

import HomePage from './components/HomePage';

import AboutPage from './components/AboutPage';

import Navbar from './components/Navbar';

import Footer from './components/Footer';

import './App.css';

const App = () => {

return (

<Router>

<div className="app">

<Navbar />

<Switch>

<Route path="/" exact component={HomePage} />

<Route path="/about" component={AboutPage} />

</Switch>

<Footer />

</div>

</Router>

);

};

export default App;

**Navbar.js**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<ul>

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

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

</ul>

</nav>

);

};

export default Navbar;

**Navbar.css**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

padding: 10px;

}

.navbar ul {

list-style-type: none;

display: flex;

justify-content: center;

gap: 20px;

}

.navbar li {

margin: 0;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 18px;

}

.navbar a:hover {

color: #ddd;

}

**HomePage.js**

import React, { useState } from 'react';

import PostList from './PostList';

import AddPostForm from './AddPostForm';

import PostCounter from './PostCounter';

import './HomePage.css';

const HomePage = () => {

const [posts, setPosts] = useState([]);

const addPost = (post) => {

setPosts([...posts, post]);

};

return (

<div className="home-page">

<h1>Blog</h1>

<PostCounter posts={posts} />

<AddPostForm addPost={addPost} />

<PostList posts={posts} />

</div>

);

};

export default HomePage;

**HomePage.css**

/\* src/css/HomePage.css \*/

.home-page {

text-align: center;

max-width: 800px;

margin: 0 auto;

}

h1 {

color: #333;

}

button {

padding: 10px 20px;

margin-top: 20px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

**AddPostForm.js**

import React, { useState } from 'react';

import './AddPostForm.css';

const AddPostForm = ({ addPost }) => {

const [title, setTitle] = useState('');

const [content, setContent] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (title.trim() && content.trim()) {

addPost({ id: Date.now(), title, content });

setTitle('');

setContent('');

}

};

return (

<form onSubmit={handleSubmit} className="add-post-form">

<input

type="text"

placeholder="Title"

value={title}

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

/>

<textarea

placeholder="Content"

value={content}

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

/>

<button type="submit">Add Post</button>

</form>

);

};

export default AddPostForm;

**AddPostForm.css**

/\* src/css/AddPostForm.css \*/

.add-post-form {

display: flex;

flex-direction: column;

justify-content: center;

max-width: 600px;

margin: 0 auto;

}

.add-post-form input,

.add-post-form textarea {

padding: 10px;

margin-bottom: 10px;

width: 100%;

border: 1px solid #ddd;

border-radius: 4px;

}

.add-post-form button {

padding: 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

.add-post-form button:hover {

background-color: #555;

}

**PostList.js**

import React from 'react';

import PostItem from './PostItem';

import './PostList.css';

const PostList = ({ posts }) => {

return (

<div className="post-list">

{posts.map(post => (

<PostItem key={post.id} post={post} />

))}

</div>

);

};

export default PostList;

**PostList.css**

/\* src/css/PostList.css \*/

.post-list {

display: flex;

flex-direction: column;

gap: 20px;

margin-top: 20px;

}

**PostItem.js**

import React from 'react';

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

import './PostItem.css';

const PostItem = ({ post }) => {

return (

<div className="post-item">

<h3>{post.title}</h3>

<p>{post.content.substring(0, 100)}...</p>

<Link to={`/post/${post.id}`} className="read-more">Read More</Link>

</div>

);

};

export default PostItem;

**PostItem.css**

/\* src/css/PostItem.css \*/

.post-item {

background-color: #f9f9f9;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.post-item h3 {

margin-top: 0;

}

.read-more {

text-decoration: none;

color: #007bff;

}

.read-more:hover {

text-decoration: underline;

}

**PostDetails.js**

import React from 'react';

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

import './PostDetails.css';

const PostDetails = ({ posts }) => {

const { id } = useParams();

const post = posts.find(post => post.id === parseInt(id));

if (!post) return <p>Post not found</p>;

return (

<div className="post-details">

<h2>{post.title}</h2>

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

</div>

);

};

export default PostDetails;

**PostDetails.css**

/\* src/css/PostDetails.css \*/

.post-details {

max-width: 800px;

margin: 20px auto;

padding: 20px;

background-color: #f9f9f9;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.post-details h2 {

margin-top: 0;

}

**PostCounter.js**

import React from 'react';

import './PostCounter.css';

const PostCounter = ({ posts }) => {

return (

<div className="post-counter">

<p>Total Posts: {posts.length}</p>

</div>

);

};

export default PostCounter;

**PostCounter.css**

/\* src/css/PostCounter.css \*/

.post-counter {

margin: 20px 0;

font-size: 18px;

}

**AboutPage.js**

javascript

CopyEdit

import React from 'react';

import './AboutPage.css';

const AboutPage = () => {

return (

<div className="about-page">

<h2>About This Blog</h2>

<p>This is a simple blog application built with React to manage posts.</p>

</div>

);

};

export default AboutPage;

**AboutPage.css**

/\* src/css/AboutPage.css \*/

.about-page {

text-align: center;

margin-top: 50px;

}

.about-page h2 {

color: #333;

}

.about-page p {

color: #666;

}

**Footer.js**

import React from 'react';

import './Footer.css';

const Footer = () => {

return (

<footer className="footer">

<p>&copy; 2025 Blog Application</p>

</footer>

);

};

export default Footer;

**Footer.css**

/\* src/css/Footer.css \*/

.footer {

text-align: center;

padding: 20px;

background-color: #333;

color: white;

}

**Final Remarks:**

This blog application includes 10 components with individual CSS for each component, as well as routing to navigate between pages. The app supports adding posts, viewing post details, and displays a simple counter for the number of posts.

**Exercise 10:**

**Objective:**

Create a **Task Manager Application** where users can manage tasks by adding, editing, and deleting them.

**Features:**

1. **App**: Main application with routing.
2. **Navbar**: A navigation bar with links to Home, Add Task, and About pages.
3. **HomePage**: Displays a list of tasks.
4. **TaskItem**: Displays each task in the list with options to edit and delete.
5. **AddTaskForm**: Form to add a new task.
6. **EditTaskForm**: Form to edit an existing task.
7. **TaskCounter**: Displays the total number of tasks.
8. **AboutPage**: Displays information about the Task Manager.
9. **Footer**: Displays footer information.
10. **TaskList**: A container to display the list of tasks.

**App.js**

import React from 'react';

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

import HomePage from './components/HomePage';

import AboutPage from './components/AboutPage';

import Navbar from './components/Navbar';

import Footer from './components/Footer';

import './App.css';

const App = () => {

return (

<Router>

<div className="app">

<Navbar />

<Switch>

<Route path="/" exact component={HomePage} />

<Route path="/about" component={AboutPage} />

</Switch>

<Footer />

</div>

</Router>

);

};

export default App;

**Navbar.js**

import React from 'react';

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

import './Navbar.css';

const Navbar = () => {

return (

<nav className="navbar">

<ul>

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

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

</ul>

</nav>

);

};

export default Navbar;

**Navbar.css**

/\* src/css/Navbar.css \*/

.navbar {

background-color: #333;

padding: 10px;

}

.navbar ul {

list-style-type: none;

display: flex;

justify-content: center;

gap: 20px;

}

.navbar li {

margin: 0;

}

.navbar a {

color: white;

text-decoration: none;

font-size: 18px;

}

.navbar a:hover {

color: #ddd;

}

**HomePage.js**

import React, { useState } from 'react';

import TaskList from './TaskList';

import AddTaskForm from './AddTaskForm';

import TaskCounter from './TaskCounter';

import './HomePage.css';

const HomePage = () => {

const [tasks, setTasks] = useState([]);

const addTask = (task) => {

setTasks([...tasks, task]);

};

const editTask = (id, updatedTask) => {

setTasks(tasks.map(task => task.id === id ? updatedTask : task));

};

const deleteTask = (id) => {

setTasks(tasks.filter(task => task.id !== id));

};

return (

<div className="home-page">

<h1>Task Manager</h1>

<TaskCounter tasks={tasks} />

<AddTaskForm addTask={addTask} />

<TaskList tasks={tasks} editTask={editTask} deleteTask={deleteTask} />

</div>

);

};

export default HomePage;

**HomePage.css**

/\* src/css/HomePage.css \*/

.home-page {

text-align: center;

max-width: 800px;

margin: 0 auto;

}

h1 {

color: #333;

}

button {

padding: 10px 20px;

margin-top: 20px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

**AddTaskForm.js**

import React, { useState } from 'react';

import './AddTaskForm.css';

const AddTaskForm = ({ addTask }) => {

const [title, setTitle] = useState('');

const [description, setDescription] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (title.trim() && description.trim()) {

addTask({ id: Date.now(), title, description });

setTitle('');

setDescription('');

}

};

return (

<form onSubmit={handleSubmit} className="add-task-form">

<input

type="text"

placeholder="Task Title"

value={title}

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

/>

<textarea

placeholder="Task Description"

value={description}

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

/>

<button type="submit">Add Task</button>

</form>

);

};

export default AddTaskForm;

**AddTaskForm.css**

/\* src/css/AddTaskForm.css \*/

.add-task-form {

display: flex;

flex-direction: column;

justify-content: center;

max-width: 600px;

margin: 0 auto;

}

.add-task-form input,

.add-task-form textarea {

padding: 10px;

margin-bottom: 10px;

width: 100%;

border: 1px solid #ddd;

border-radius: 4px;

}

.add-task-form button {

padding: 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

.add-task-form button:hover {

background-color: #555;

}

**TaskList.js**

import React from 'react';

import TaskItem from './TaskItem';

import './TaskList.css';

const TaskList = ({ tasks, editTask, deleteTask }) => {

return (

<div className="task-list">

{tasks.map(task => (

<TaskItem

key={task.id}

task={task}

editTask={editTask}

deleteTask={deleteTask}

/>

))}

</div>

);

};

export default TaskList;

**TaskList.css**

/\* src/css/TaskList.css \*/

.task-list {

display: flex;

flex-direction: column;

gap: 20px;

margin-top: 20px;

}

**TaskItem.js**

import React, { useState } from 'react';

import EditTaskForm from './EditTaskForm';

import './TaskItem.css';

const TaskItem = ({ task, editTask, deleteTask }) => {

const [isEditing, setIsEditing] = useState(false);

const handleEdit = (updatedTask) => {

editTask(task.id, updatedTask);

setIsEditing(false);

};

return (

<div className="task-item">

{!isEditing ? (

<>

<h3>{task.title}</h3>

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

<button onClick={() => setIsEditing(true)}>Edit</button>

<button onClick={() => deleteTask(task.id)}>Delete</button>

</>

) : (

<EditTaskForm task={task} handleEdit={handleEdit} />

)}

</div>

);

};

export default TaskItem;

**TaskItem.css**

/\* src/css/TaskItem.css \*/

.task-item {

background-color: #f9f9f9;

padding: 20px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.task-item h3 {

margin-top: 0;

}

button {

padding: 5px 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

margin-right: 10px;

}

button:hover {

background-color: #555;

}

**EditTaskForm.js**

import React, { useState } from 'react';

import './EditTaskForm.css';

const EditTaskForm = ({ task, handleEdit }) => {

const [title, setTitle] = useState(task.title);

const [description, setDescription] = useState(task.description);

const handleSubmit = (e) => {

e.preventDefault();

handleEdit({ ...task, title, description });

};

return (

<form onSubmit={handleSubmit} className="edit-task-form">

<input

type="text"

value={title}

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

/>

<textarea

value={description}

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

/>

<button type="submit">Save</button>

</form>

);

};

export default EditTaskForm;

**EditTaskForm.css**

/\* src/css/EditTaskForm.css \*/

.edit-task-form {

display: flex;

flex-direction: column;

justify-content: center;

gap: 10px;

max-width: 600px;

margin: 0 auto;

}

.edit-task-form input,

.edit-task-form textarea {

padding: 10px;

margin-bottom: 10px;

width: 100%;

border: 1px solid #ddd;

border-radius: 4px;

}

.edit-task-form button {

padding: 10px;

background-color: #333;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

.edit-task-form button:hover {

background-color: #555;

}

**TaskCounter.js**

import React from 'react';

import './TaskCounter.css';

const TaskCounter = ({ tasks }) => {

return (

<div className="task-counter">

<p>Total Tasks: {tasks.length}</p>

</div>

);

};

export default TaskCounter;

**TaskCounter.css**

/\* src/css/TaskCounter.css \*/

.task-counter {

margin: 20px 0;

font-size: 18px;

}

**AboutPage.js**

import React from 'react';

import './AboutPage.css';

const AboutPage = () => {

return (

<div className="about-page">

<h1>About Task Manager</h1>

<p>This is a simple task manager application where you can add, edit, and delete tasks.</p>

</div>

);

};

export default AboutPage;

**AboutPage.css**

/\* src/css/AboutPage.css \*/

.about-page {

padding: 20px;

text-align: center;

}

**Footer.js**

import React from 'react';

import './Footer.css';

const Footer = () => {

return (

<footer className="footer">

<p>&copy; 2025 Task Manager Application</p>

</footer>

);

};

export default Footer;

**Footer.css**

/\* src/css/Footer.css \*/

.footer {

text-align: center;

padding: 20px;

background-color: #333;

color: white;

}