**Event Handling**

In React, **event handling** refers to the way you respond to user interactions and other events (like clicks, form submissions, and keyboard inputs) within your components.

React provides a declarative way to manage these interactions by defining event handlers that are triggered in response to specific events.

**Key Concepts of Event Handling in React**

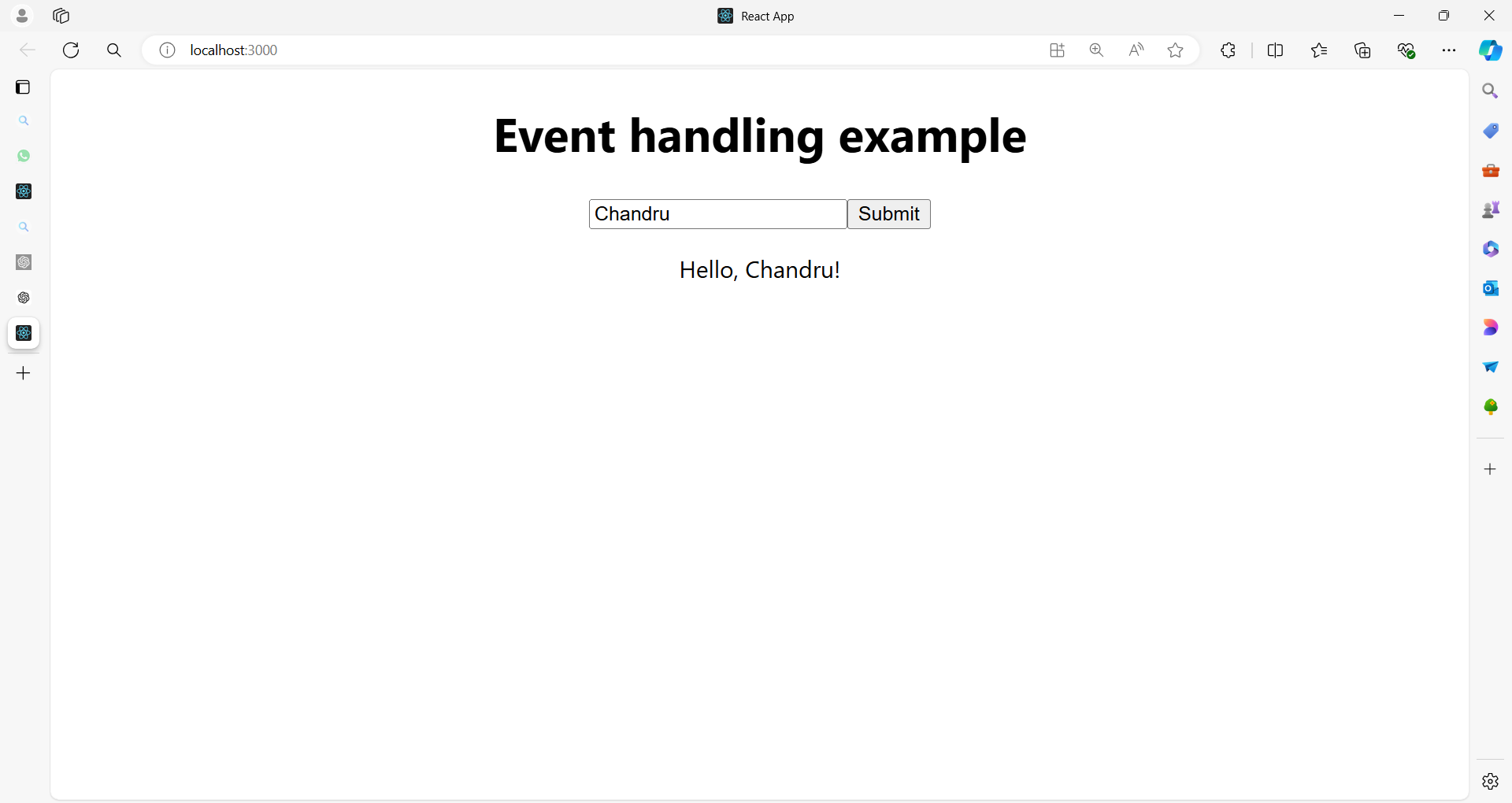
1. **Event Handlers**: Functions that are called in response to events, such as user actions (clicking a button) or browser events (loading a page).
2. **Event Binding**: Attaching event handlers to React components or elements.
3. **Synthetic Events**: React wraps native DOM events in its own **SyntheticEvent** wrapper. This wrapper ensures that events work consistently across different browsers and is part of React’s event delegation system.
4. **Event Propagation**: React’s event handling system uses event delegation to handle events at a higher level in the DOM tree, which improves performance by reducing the number of event listeners attached to individual elements.

**Common Event Types**

* **Click Events**: onClick
* **Form Events**: onChange, onSubmit
* **Keyboard Events**: onKeyDown, onKeyUp
* **Mouse Events**: onMouseEnter, onMouseLeave
* **Focus Events**: onFocus, onBlur

**Example of Event Handling in React**

Here’s a simple example demonstrating how to handle a button click event and form submission in React:



**src/App.js**

import React, { useState } from 'react';

import './App.css';

function App() {

// State to manage the input value

const [inputValue, setInputValue] = useState('');

// State to manage the message displayed

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

// Event handler for button click

const handleClick = () => {

setMessage(`Hello, ${inputValue}!`);

};

// Event handler for form submission

const handleSubmit = (e) => {

e.preventDefault(); // Prevent the default form submission behavior

handleClick();

};

// Event handler for input change

const handleChange = (e) => {

setInputValue(e.target.value);

};

return (

<div className="App">

<h1>Event Handling Example</h1>

<form onSubmit={handleSubmit}>

<input

type="text"

value={inputValue}

onChange={handleChange}

placeholder="Enter your name"

/>

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

</form>

<p>{message}</p>

</div>

);

}

export default App;

**Breakdown of the Example**

1. **State Management**:
   * inputValue: Manages the value of the text input.
   * message: Stores the greeting message to be displayed.
2. **Event Handlers**:
   * **handleClick**: Updates the message state with a greeting message that includes the inputValue.
   * **handleSubmit**: Prevents the default form submission action (which would reload the page) and calls handleClick to update the message.
   * **handleChange**: Updates the inputValue state whenever the user types in the input field.
3. **Event Binding**:
   * **onClick**: Not used directly here but can be used in other components.
   * **onSubmit**: Binds to the form's submit event to trigger handleSubmit.
   * **onChange**: Binds to the input field’s change event to trigger handleChange.

**Summary**

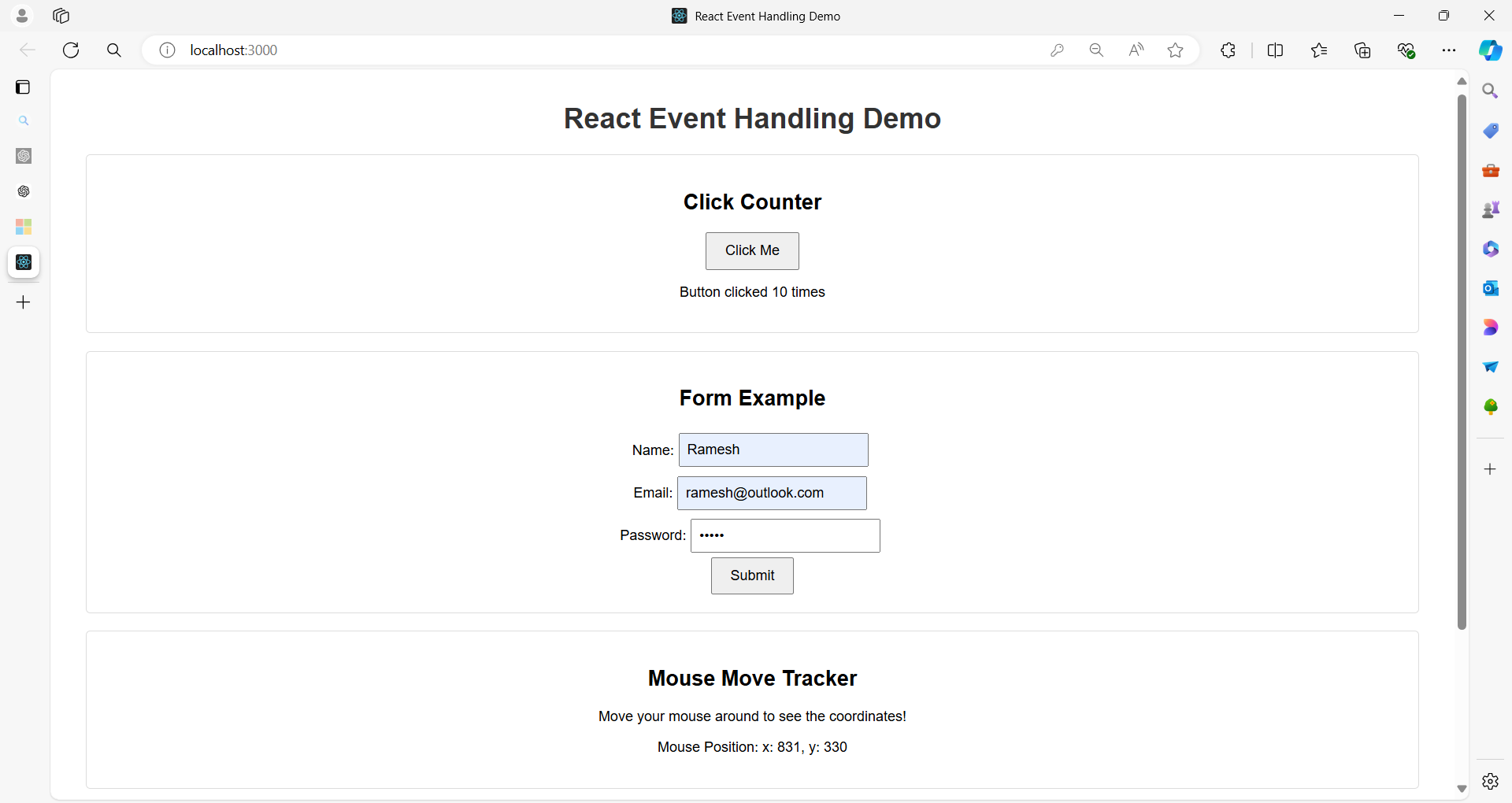
Event handling in React involves defining functions to respond to user actions or browser events and binding these functions to the appropriate events on React elements.

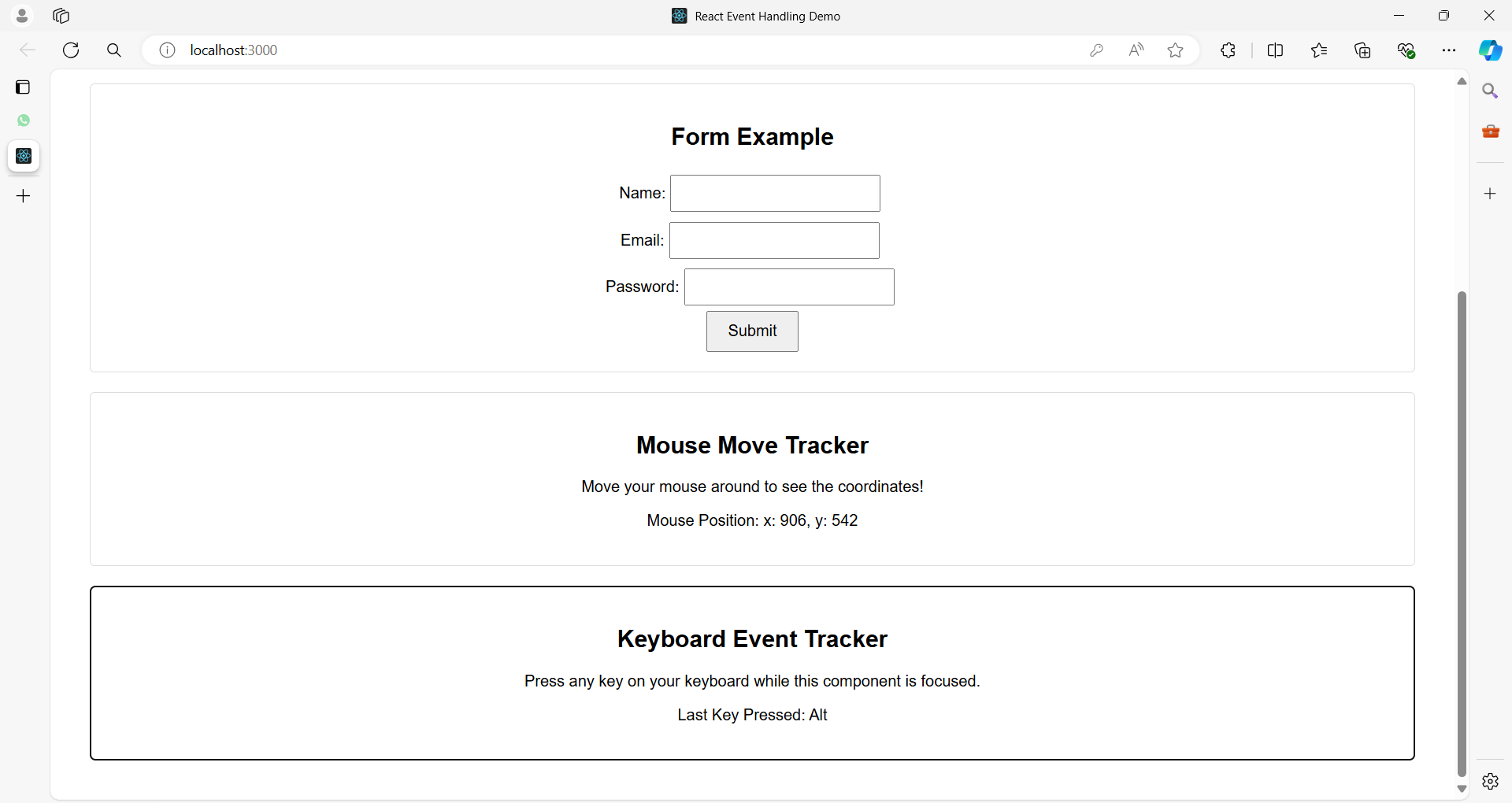
React’s synthetic event system and event delegation enhance performance and ensure consistent behavior across different browsers.

**Example 01**

Here's a comprehensive example of a React application that demonstrates various types of events, including click events, form events, mouse events, keyboard events, and more.

The program will include different components to illustrate how these events can be handled in a React application.





**Project Structure**

**event-demo-app/**

|-- public/

| |-- index.html

|-- src/

| |-- App.js

| |-- ButtonComponent.js

| |-- FormComponent.js

| |-- MouseEventComponent.js

| |-- KeyboardEventComponent.js

| |-- App.css

|-- package.json

|-- README.md

**Code for Each File**

**src/App.css**

.App {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.button-component, .form-component, .mouse-event-component, .keyboard-event-component {

margin: 20px;

padding: 20px;

border: 1px solid #ddd;

border-radius: 5px;

}

button {

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

}

input {

padding: 8px;

font-size: 16px;

margin: 5px;

}

form {

display: flex;

flex-direction: column;

align-items: center;

}

h1 {

color: #333;

}

**public/index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>React Event Handling Demo</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

**src/App.js**

import React from 'react';

import './App.css';

import ButtonComponent from './ButtonComponent';

import FormComponent from './FormComponent';

import MouseEventComponent from './MouseEventComponent';

import KeyboardEventComponent from './KeyboardEventComponent';

function App() {

return (

<div className="App">

<h1>React Event Handling Demo</h1>

<ButtonComponent />

<FormComponent />

<MouseEventComponent />

<KeyboardEventComponent />

</div>

);

}

export default App;

**src/ButtonComponent.js**

import React, { useState } from 'react';

import './App.css';

function ButtonComponent() {

const [count, setCount] = useState(0);

const handleClick = () => {

setCount(count + 1);

};

return (

<div className="button-component">

<h2>Click Counter</h2>

<button onClick={handleClick}>Click Me</button>

<p>Button clicked {count} times</p>

</div>

);

}

export default ButtonComponent;

**src/FormComponent.js**

import React, { useState } from 'react';

import './App.css';

function FormComponent() {

const [formData, setFormData] = useState({

name: '',

email: '',

password: ''

});

const handleChange = (e) => {

const { name, value } = e.target;

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

};

const handleSubmit = (e) => {

e.preventDefault();

alert(`Submitted! \nName: ${formData.name} \nEmail: ${formData.email}`);

};

return (

<div className="form-component">

<h2>Form Example</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</div>

<div>

<label>Email:</label>

<input

type="email"

name="email"

value={formData.email}

onChange={handleChange}

required

/>

</div>

<div>

<label>Password:</label>

<input

type="password"

name="password"

value={formData.password}

onChange={handleChange}

required

/>

</div>

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

</form>

</div>

);

}

export default FormComponent;

**src/MouseEventComponent.js**

import React, { useState } from 'react';

import './App.css';

function MouseEventComponent() {

const [position, setPosition] = useState({ x: 0, y: 0 });

const handleMouseMove = (e) => {

setPosition({ x: e.clientX, y: e.clientY });

};

return (

<div className="mouse-event-component" onMouseMove={handleMouseMove}>

<h2>Mouse Move Tracker</h2>

<p>Move your mouse around to see the coordinates!</p>

<p>Mouse Position: {`x: ${position.x}, y: ${position.y}`}</p>

</div>

);

}

export default MouseEventComponent;

**src/KeyboardEventComponent.js**

import React, { useState } from 'react';

import './App.css';

function KeyboardEventComponent() {

const [key, setKey] = useState('');

const handleKeyDown = (e) => {

setKey(e.key);

};

return (

<div className="keyboard-event-component" tabIndex="0" onKeyDown={handleKeyDown}>

<h2>Keyboard Event Tracker</h2>

<p>Press any key on your keyboard while this component is focused.</p>

<p>Last Key Pressed: {key}</p>

</div>

);

}

export default KeyboardEventComponent;

**Summary**

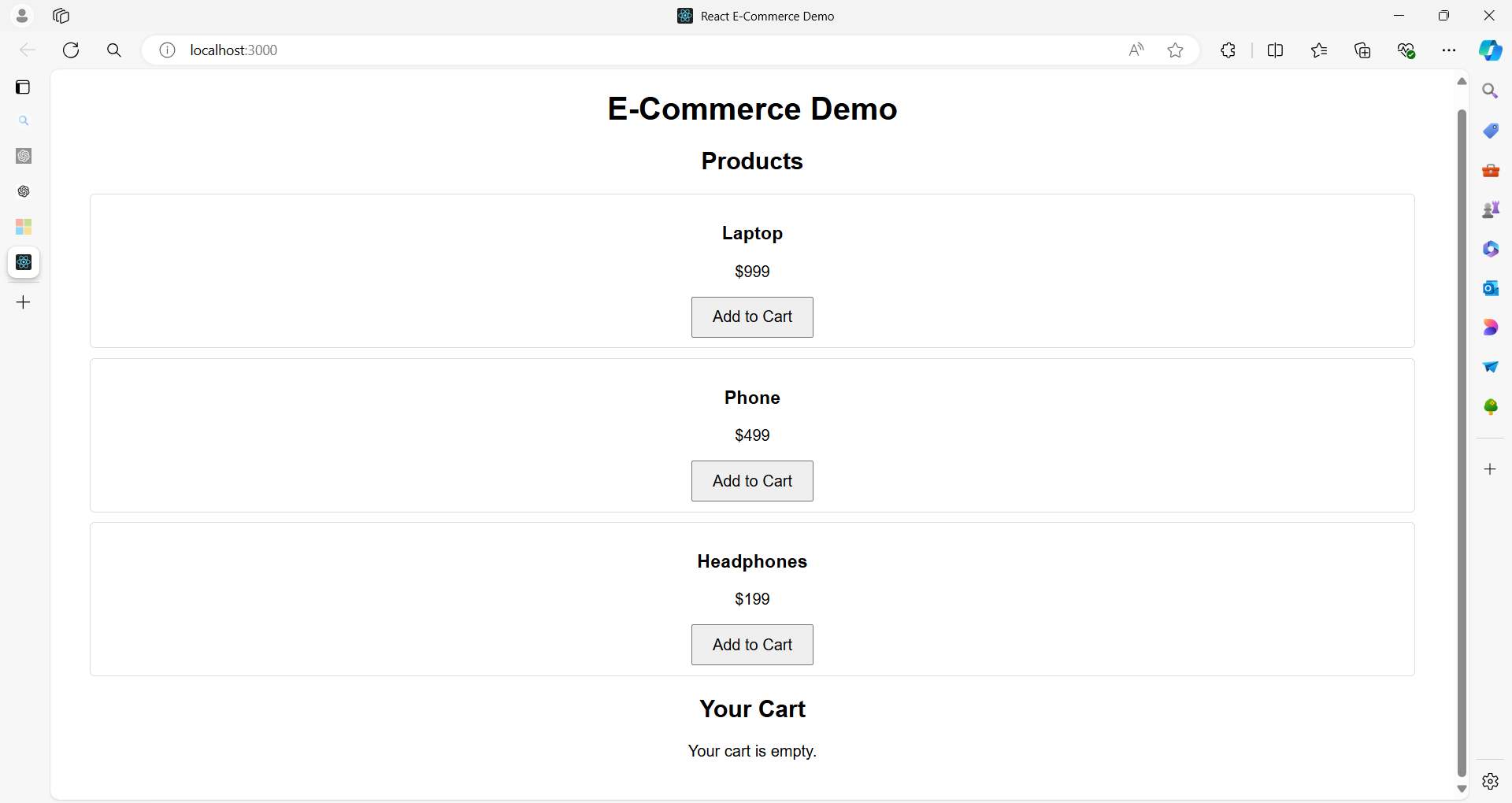
* **ButtonComponent**: Demonstrates handling a click event. The button click increments a counter.
* **FormComponent**: Shows handling form events including input changes and form submission.
* **MouseEventComponent**: Illustrates handling mouse movement events to track the mouse position.
* **KeyboardEventComponent**: Captures keyboard events to display the last key pressed.

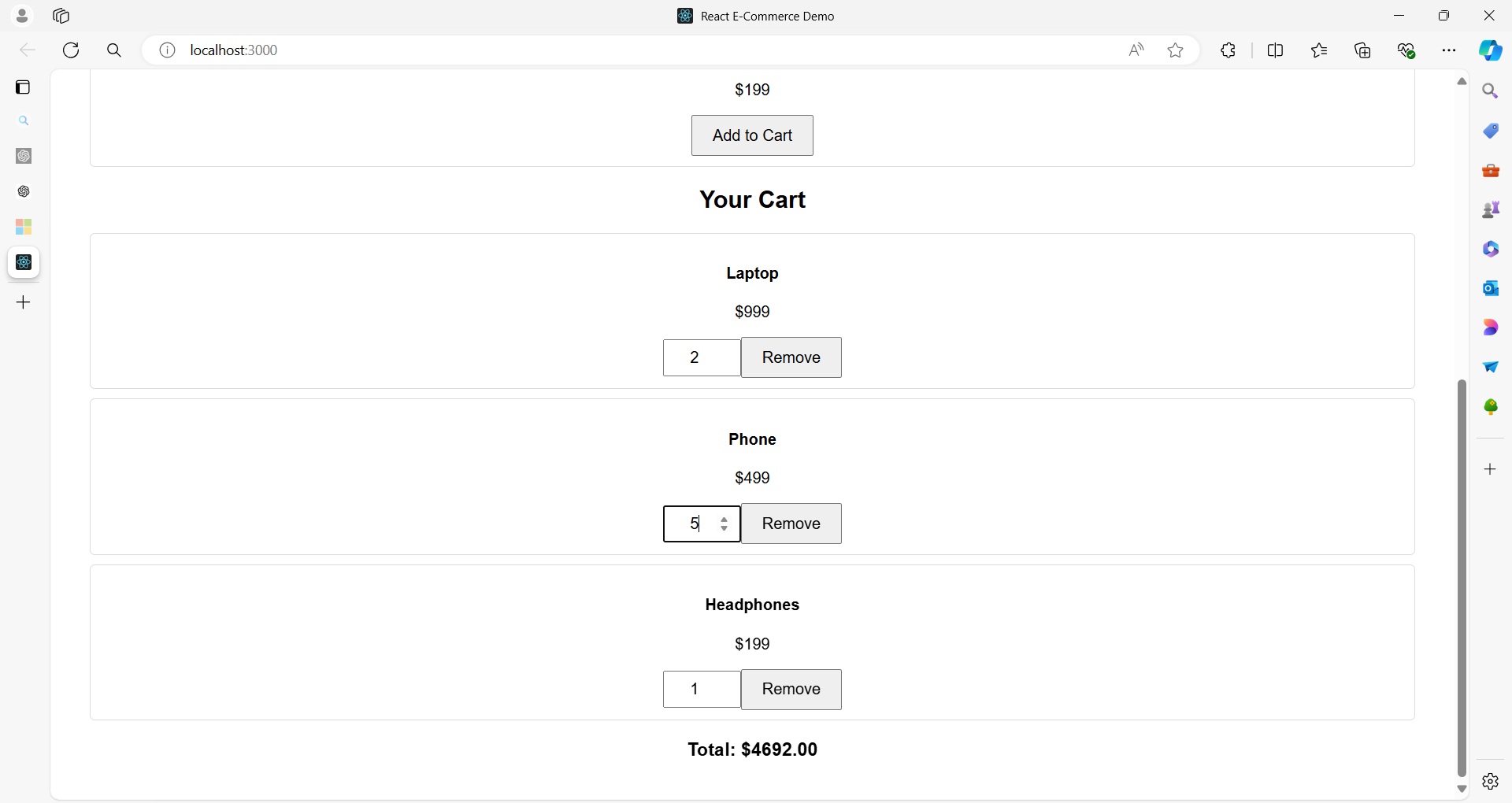
This example provides a comprehensive overview of handling various types of events in React, demonstrating how to use event handlers to manage user interactions effectively.

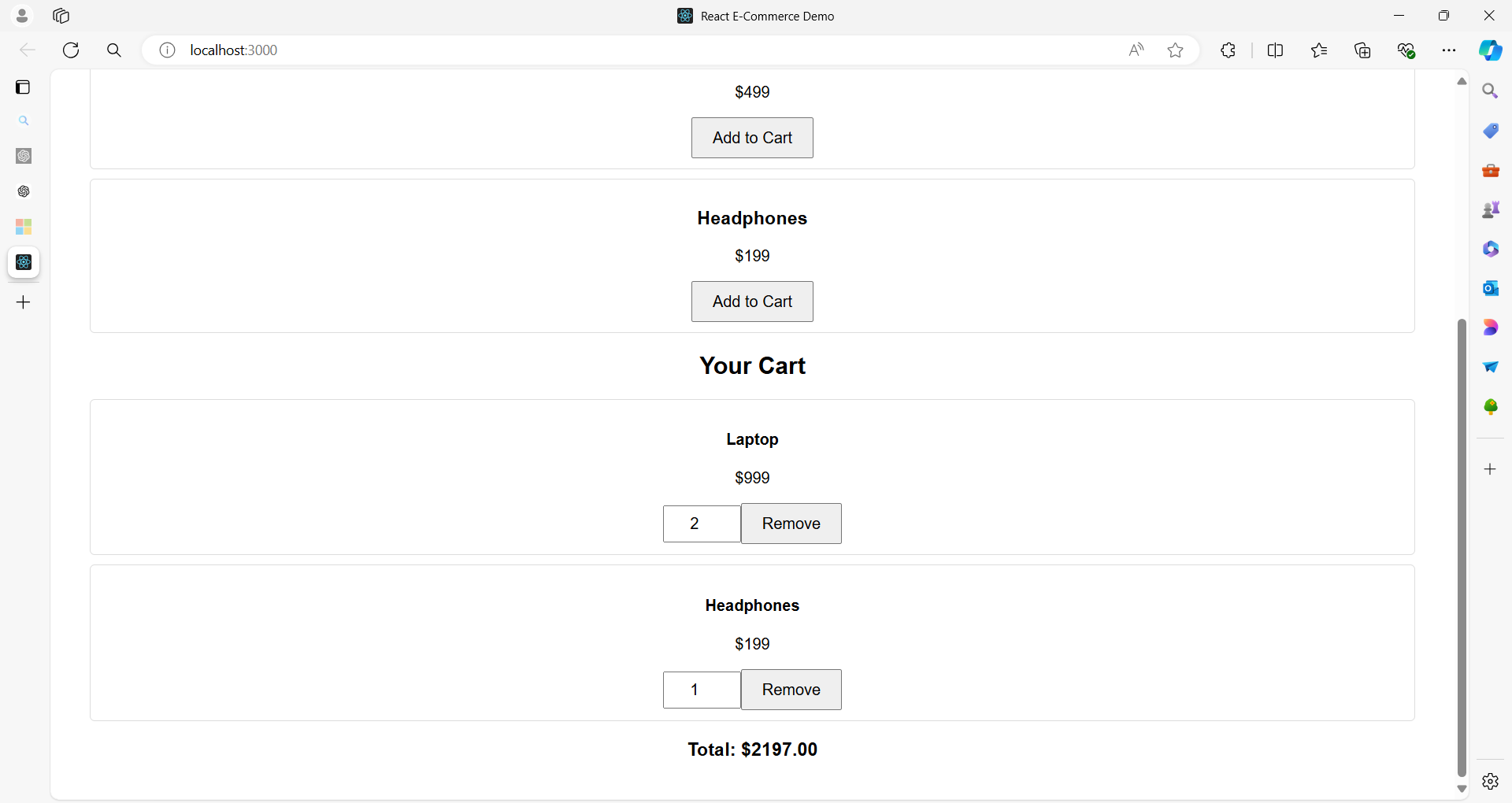
**Example 02**

Here's an example of a more complex React application for an e-commerce website.

This application demonstrates various event handling techniques, including adding items to the cart, updating quantities, and removing items.







**Project Structure**

**ecommerce-demo/**

|-- public/

| |-- index.html

|-- src/

| |-- App.js

| |-- ProductList.js

| |-- Cart.js

| |-- ProductItem.js

| |-- CartItem.js

| |-- App.css

|-- package.json

|-- README.md

**Code for Each File**

**src/App.css**

.App {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.product-list, .cart {

margin: 20px;

}

.product-item, .cart-item {

margin: 10px 0;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

button {

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

}

input {

padding: 8px;

font-size: 16px;

width: 60px;

text-align: center;

}

**public/index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>React E-Commerce Demo</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

**src/App.js**

import React, { useState } from 'react';

import './App.css';

import ProductList from './ProductList';

import Cart from './Cart';

const initialProducts = [

{ id: 1, name: 'Laptop', price: 999 },

{ id: 2, name: 'Phone', price: 499 },

{ id: 3, name: 'Headphones', price: 199 }

];

function App() {

const [cart, setCart] = useState([]);

const addToCart = (product) => {

setCart((prevCart) => {

const existingProduct = prevCart.find(item => item.id === product.id);

if (existingProduct) {

return prevCart.map(item =>

item.id === product.id ? { ...item, quantity: item.quantity + 1 } : item

);

}

return [...prevCart, { ...product, quantity: 1 }];

});

};

const updateQuantity = (id, quantity) => {

setCart((prevCart) =>

prevCart.map(item =>

item.id === id ? { ...item, quantity: Math.max(1, quantity) } : item

)

);

};

const removeFromCart = (id) => {

setCart((prevCart) => prevCart.filter(item => item.id !== id));

};

return (

<div className="App">

<h1>E-Commerce Demo</h1>

<ProductList products={initialProducts} onAddToCart={addToCart} />

<Cart

cartItems={cart}

onUpdateQuantity={updateQuantity}

onRemoveFromCart={removeFromCart}

/>

</div>

);

}

export default App;

**src/ProductList.js**

import React from 'react';

import ProductItem from './ProductItem';

function ProductList({ products, onAddToCart }) {

return (

<div className="product-list">

<h2>Products</h2>

{products.map(product => (

<ProductItem key={product.id} product={product} onAddToCart={onAddToCart} />

))}

</div>

);

}

export default ProductList;

**src/ProductItem.js**

import React from 'react';

function ProductItem({ product, onAddToCart }) {

return (

<div className="product-item">

<h3>{product.name}</h3>

<p>${product.price}</p>

<button onClick={() => onAddToCart(product)}>Add to Cart</button>

</div>

);

}

export default ProductItem;

**src/Cart.js**

import React from 'react';

import CartItem from './CartItem';

function Cart({ cartItems, onUpdateQuantity, onRemoveFromCart }) {

const total = cartItems.reduce((sum, item) => sum + item.price \* item.quantity, 0);

return (

<div className="cart">

<h2>Your Cart</h2>

{cartItems.length === 0 ? (

<p>Your cart is empty.</p>

) : (

<div>

{cartItems.map(item => (

<CartItem

key={item.id}

item={item}

onUpdateQuantity={onUpdateQuantity}

onRemoveFromCart={onRemoveFromCart}

/>

))}

<h3>Total: ${total.toFixed(2)}</h3>

</div>

)}

</div>

);

}

export default Cart;

**src/CartItem.js**

import React from 'react';

function CartItem({ item, onUpdateQuantity, onRemoveFromCart }) {

const handleQuantityChange = (e) => {

onUpdateQuantity(item.id, parseInt(e.target.value, 10));

};

return (

<div className="cart-item">

<h4>{item.name}</h4>

<p>${item.price}</p>

<input

type="number"

value={item.quantity}

onChange={handleQuantityChange}

min="1"

/>

<button onClick={() => onRemoveFromCart(item.id)}>Remove</button>

</div>

);

}

export default CartItem;

**Summary**

This e-commerce demo application consists of:

* **App.js**: The main component that holds the application state for the cart and provides functions to add, update, and remove items.
* **ProductList.js**: Displays a list of products with buttons to add items to the cart.
* **ProductItem.js**: Represents an individual product with a button to add it to the cart.
* **Cart.js**: Displays the items in the cart with the ability to update quantities or remove items.
* **CartItem.js**: Represents an individual cart item with quantity control and a remove button.
* **App.css**: Basic styling for the components.

This setup demonstrates various event handling scenarios including:

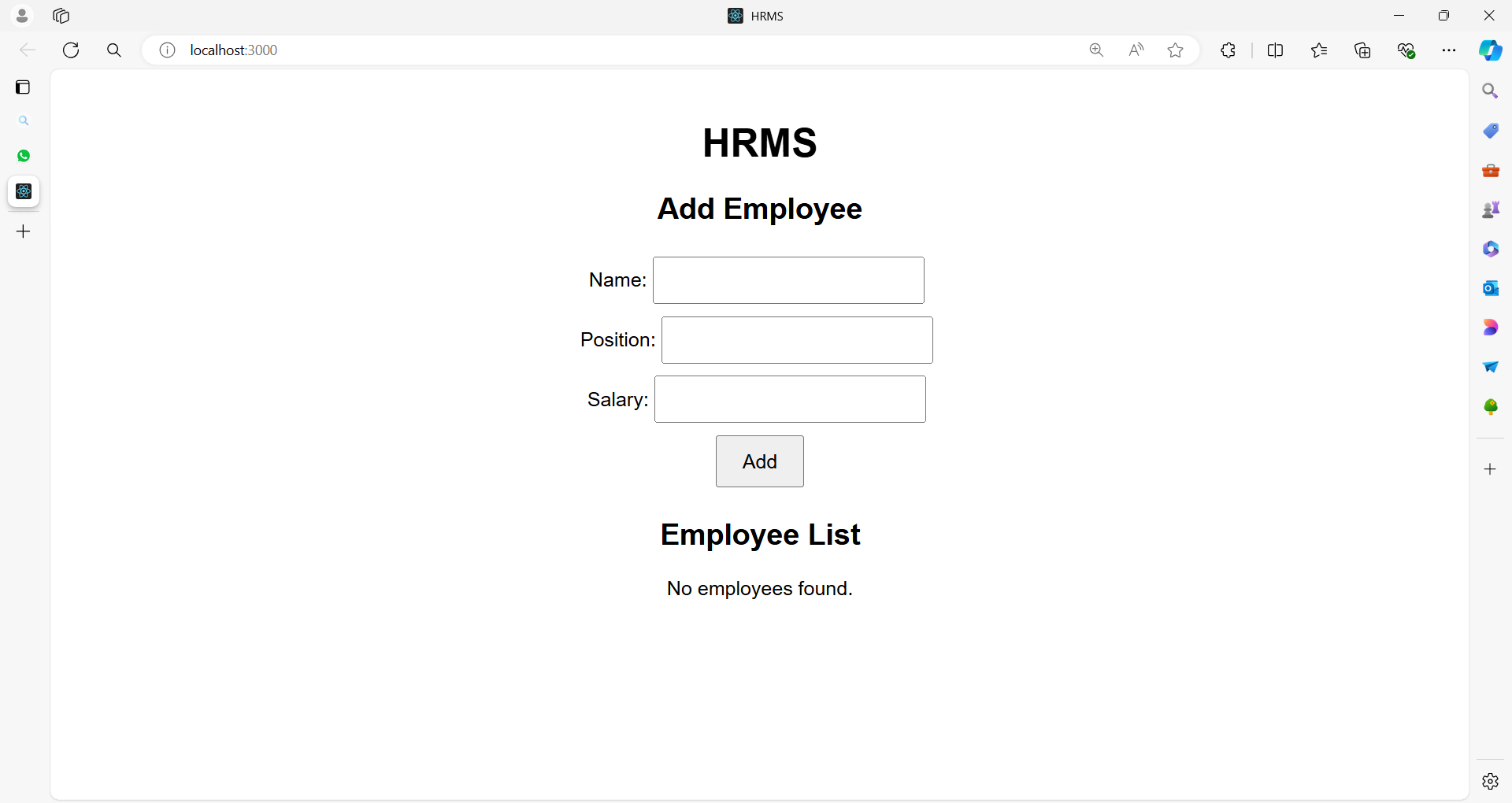
* **Button Clicks**: Adding items to the cart.
* **Input Changes**: Updating item quantities.
* **Button Clicks**: Removing items from the cart.

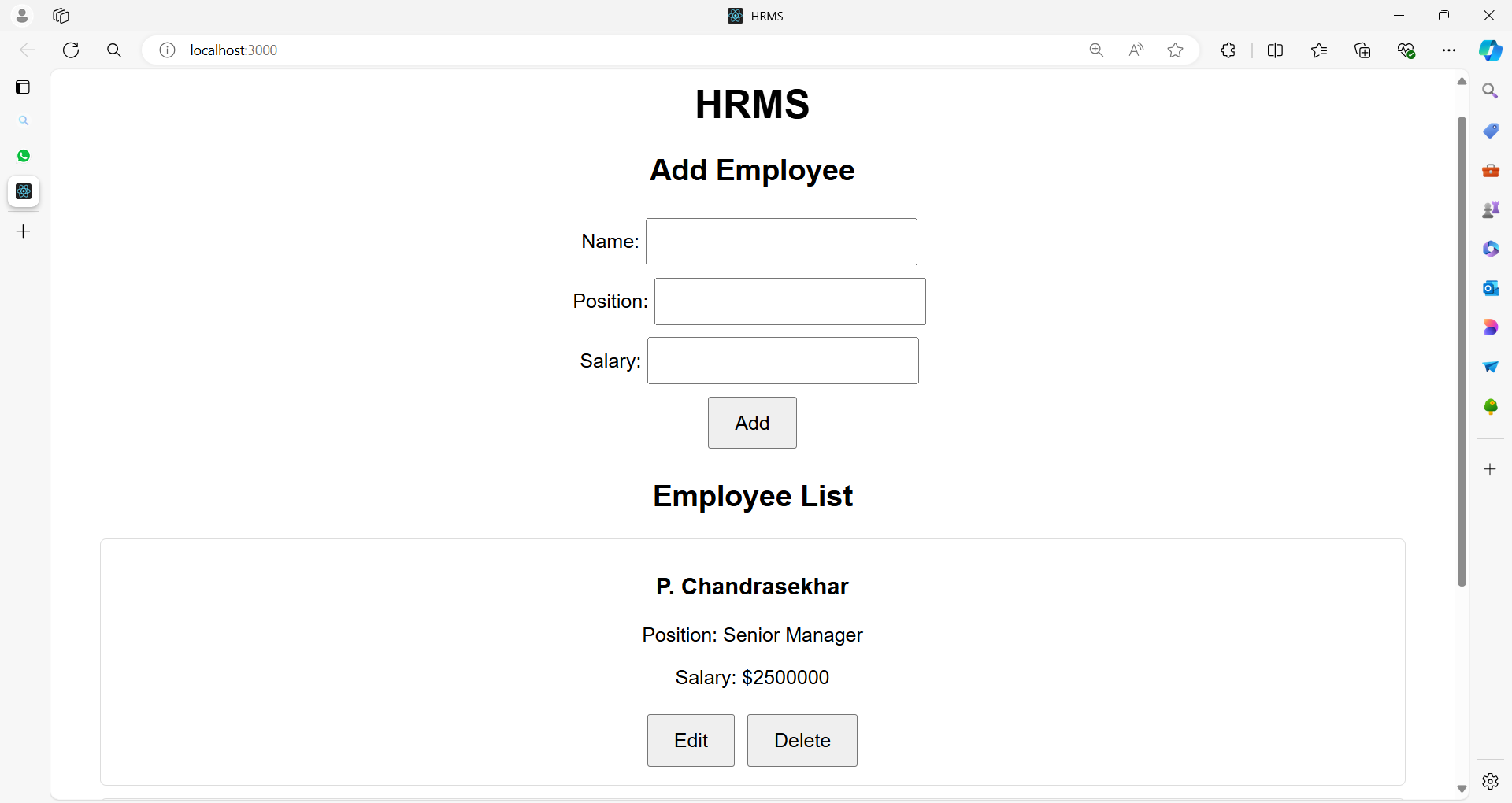
The application maintains a clear separation of concerns between different components and demonstrates how to manage state and events in a React application.

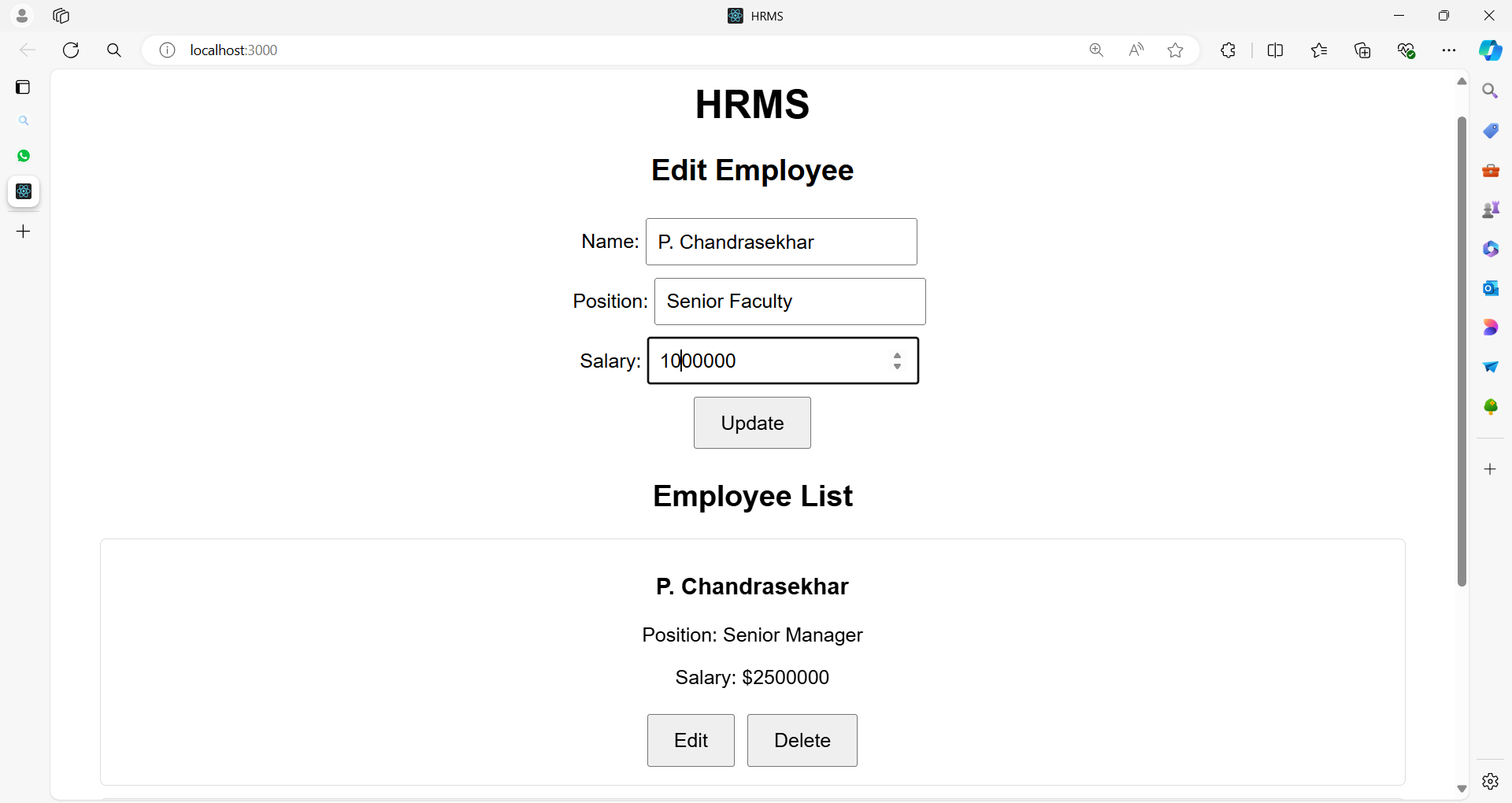
**Example 03**

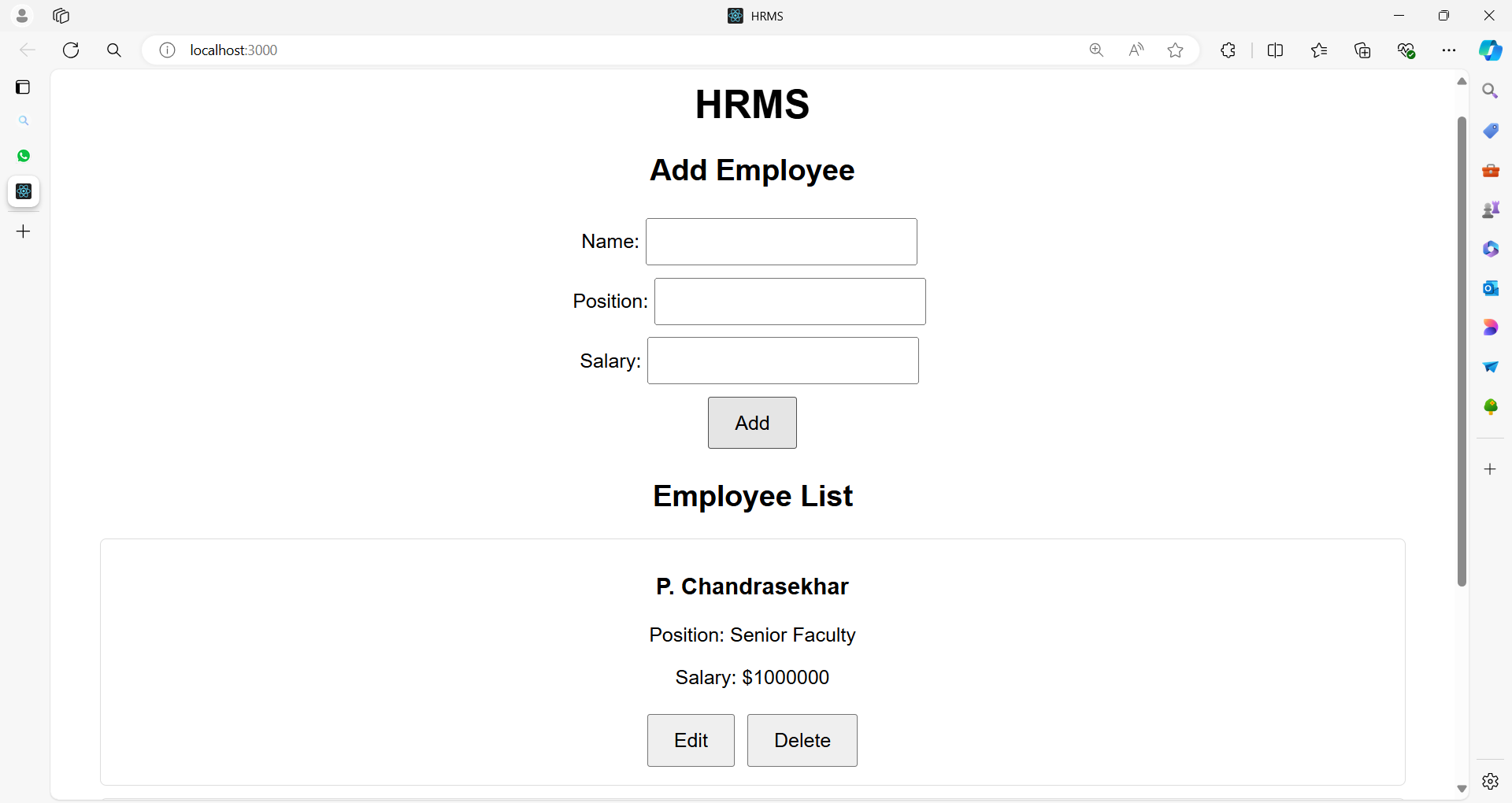
Here's a comprehensive example of a React application for a Human Resource Management System (HRMS).

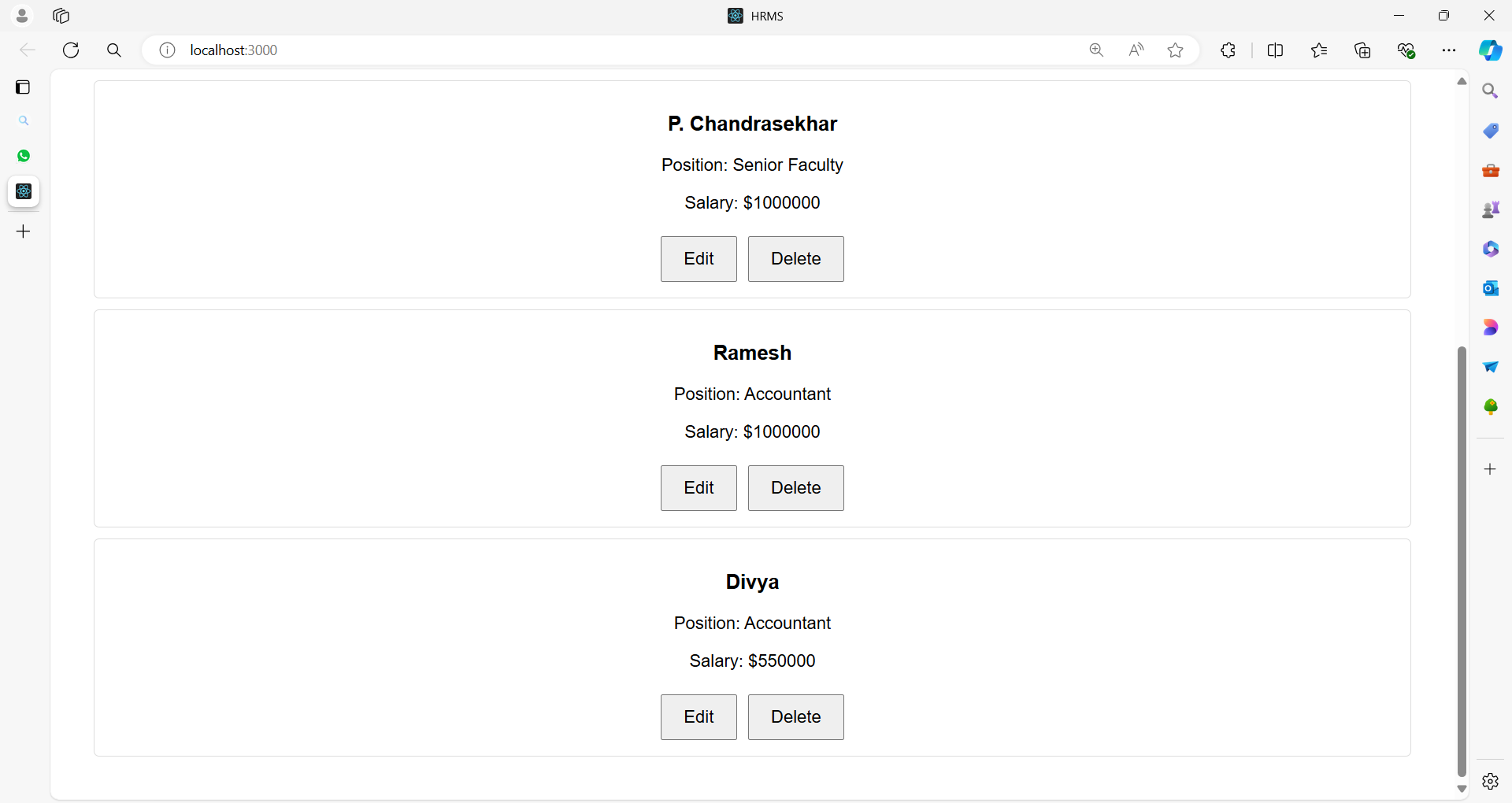
This example demonstrates various event handling techniques, including managing employee records, adding new employees, editing existing records, and deleting records.



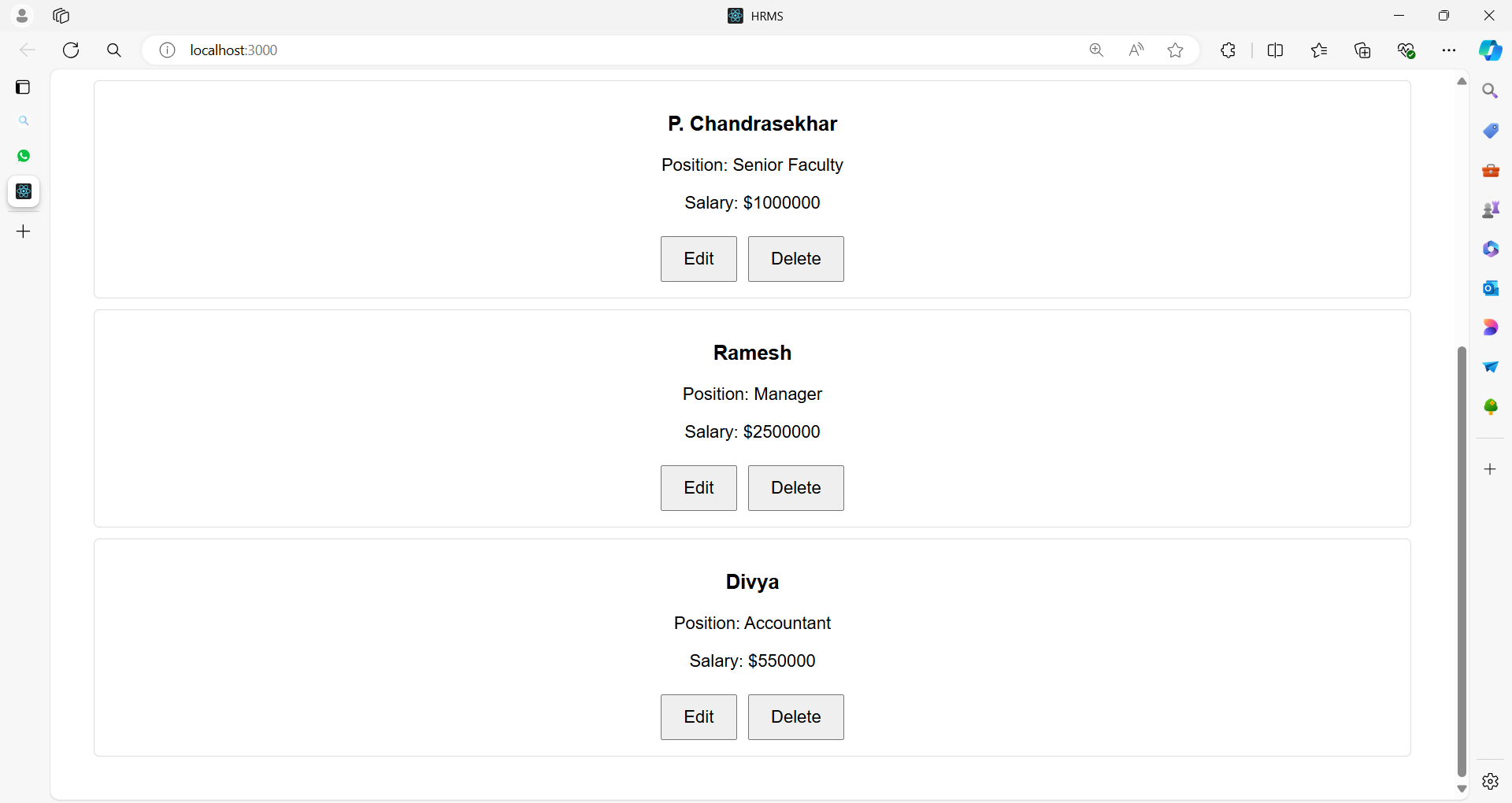




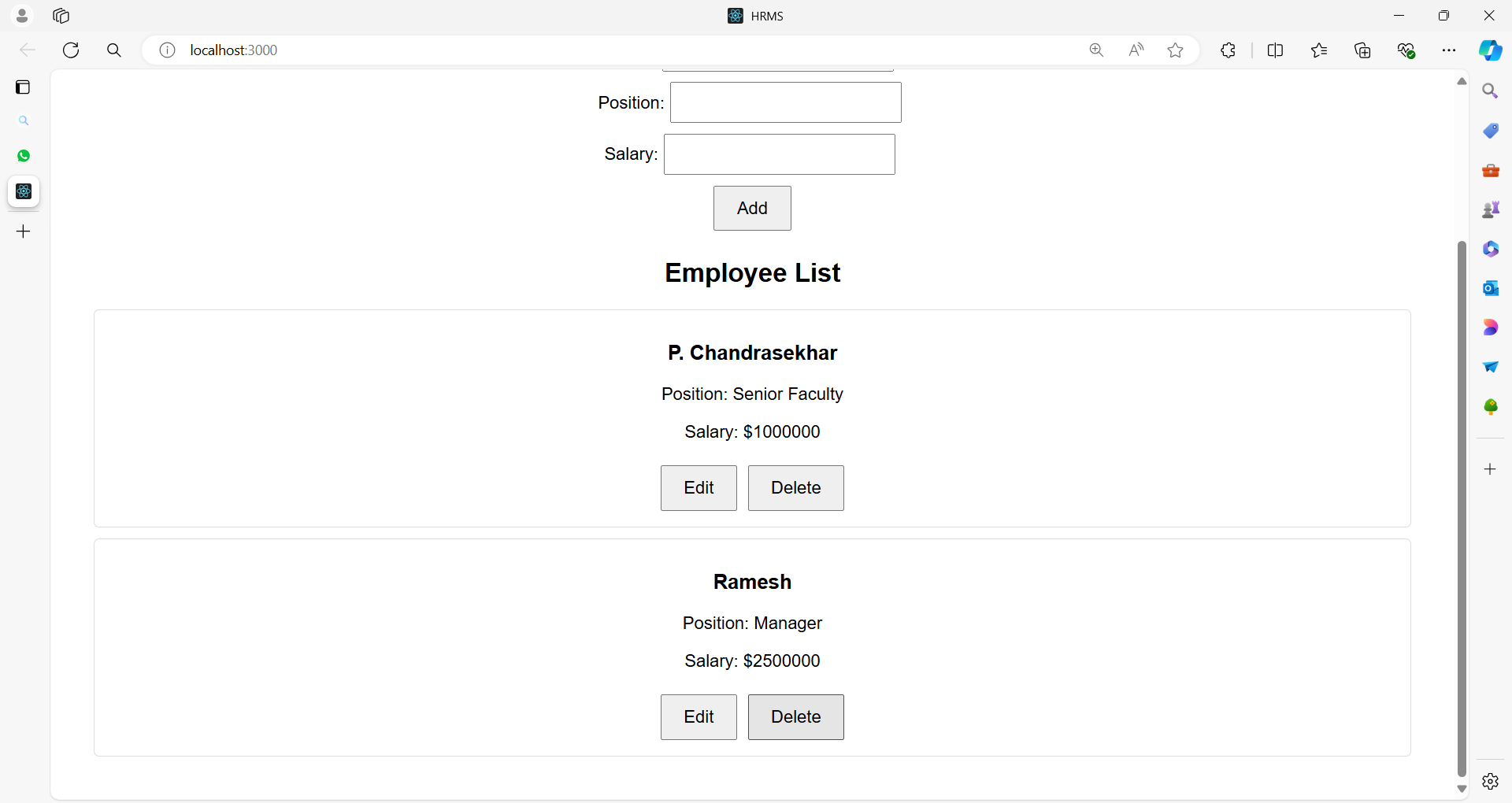




After updating the Ramesh details



After deleting Divya



**Project Structure**

**hrms-demo/**

|-- public/

| |-- index.html

|-- src/

| |-- App.js

| |-- EmployeeList.js

| |-- EmployeeForm.js

| |-- EmployeeItem.js

| |-- App.css

|-- package.json

|-- README.md

**Code for Each File**

**public/index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>HRMS</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

**src/App.js**

import React, { useState } from 'react';

import './App.css';

import EmployeeList from './EmployeeList';

import EmployeeForm from './EmployeeForm';

function App() {

const [employees, setEmployees] = useState([]);

const [editingEmployee, setEditingEmployee] = useState(null);

const addEmployee = (employee) => {

setEmployees((prevEmployees) => [...prevEmployees, { ...employee, id: Date.now() }]);

setEditingEmployee(null);

};

const updateEmployee = (updatedEmployee) => {

setEmployees((prevEmployees) =>

prevEmployees.map((emp) =>

emp.id === updatedEmployee.id ? updatedEmployee : emp

)

);

setEditingEmployee(null);

};

const deleteEmployee = (id) => {

setEmployees((prevEmployees) => prevEmployees.filter((emp) => emp.id !== id));

};

const startEditEmployee = (employee) => {

setEditingEmployee(employee);

};

return (

<div className="App">

<h1>HRMS</h1>

<EmployeeForm

onAddEmployee={addEmployee}

onUpdateEmployee={updateEmployee}

editingEmployee={editingEmployee}

/>

<EmployeeList

employees={employees}

onDeleteEmployee={deleteEmployee}

onEditEmployee={startEditEmployee}

/>

</div>

);

}

export default App;

**src/EmployeeList.js**

import React from 'react';

import EmployeeItem from './EmployeeItem';

function EmployeeList({ employees, onDeleteEmployee, onEditEmployee }) {

return (

<div className="employee-list">

<h2>Employee List</h2>

{employees.length === 0 ? (

<p>No employees found.</p>

) : (

employees.map((employee) => (

<EmployeeItem

key={employee.id}

employee={employee}

onDeleteEmployee={onDeleteEmployee}

onEditEmployee={onEditEmployee}

/>

))

)}

</div>

);

}

export default EmployeeList;

**src/EmployeeForm.js**

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

function EmployeeForm({ onAddEmployee, onUpdateEmployee, editingEmployee }) {

const [formData, setFormData] = useState({

name: '',

position: '',

salary: ''

});

useEffect(() => {

if (editingEmployee) {

setFormData({

name: editingEmployee.name,

position: editingEmployee.position,

salary: editingEmployee.salary

});

}

}, [editingEmployee]);

const handleChange = (e) => {

const { name, value } = e.target;

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

};

const handleSubmit = (e) => {

e.preventDefault();

if (editingEmployee) {

onUpdateEmployee({ ...editingEmployee, ...formData });

} else {

onAddEmployee(formData);

}

setFormData({ name: '', position: '', salary: '' });

};

return (

<div className="employee-form">

<h2>{editingEmployee ? 'Edit Employee' : 'Add Employee'}</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</div>

<div>

<label>Position:</label>

<input

type="text"

name="position"

value={formData.position}

onChange={handleChange}

required

/>

</div>

<div>

<label>Salary:</label>

<input

type="number"

name="salary"

value={formData.salary}

onChange={handleChange}

required

/>

</div>

<button type="submit">{editingEmployee ? 'Update' : 'Add'}</button>

</form>

</div>

);

}

export default EmployeeForm;

**src/EmployeeItem.js**

import React from 'react';

function EmployeeItem({ employee, onDeleteEmployee, onEditEmployee }) {

return (

<div className="employee-item">

<h3>{employee.name}</h3>

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

<p>Salary: ${employee.salary}</p>

<button onClick={() => onEditEmployee(employee)}>Edit</button>

<button onClick={() => onDeleteEmployee(employee.id)}>Delete</button>

</div>

);

}

export default EmployeeItem;

**src/App.css**

.App {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.employee-list, .employee-form {

margin: 20px;

}

.employee-item {

margin: 10px 0;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

button {

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

margin: 5px;

}

input {

padding: 8px;

font-size: 16px;

margin: 5px;

}

form {

display: flex;

flex-direction: column;

align-items: center;

}

**Summary**

This HRMS demo application consists of:

* **App.js**: The main component managing the application state for employees and providing functions to add, update, and delete employees.
* **EmployeeList.js**: Displays the list of employees and provides options to edit or delete them.
* **EmployeeItem.js**: Represents an individual employee with options to edit or delete.
* **EmployeeForm.js**: Handles the form for adding and updating employee records.
* **App.css**: Basic styling for the components.

**Key Event Handling Demonstrations:**

1. **Form Handling**:
   * **Add Employee**: Handle form submission to add new employees.
   * **Update Employee**: Handle form submission to update existing employee details.
2. **Button Clicks**:
   * **Edit**: Pre-fill the form with the selected employee’s details for editing.
   * **Delete**: Remove the employee from the list.
3. **Form Inputs**:
   * **Change Events**: Update form state when user types into input fields.

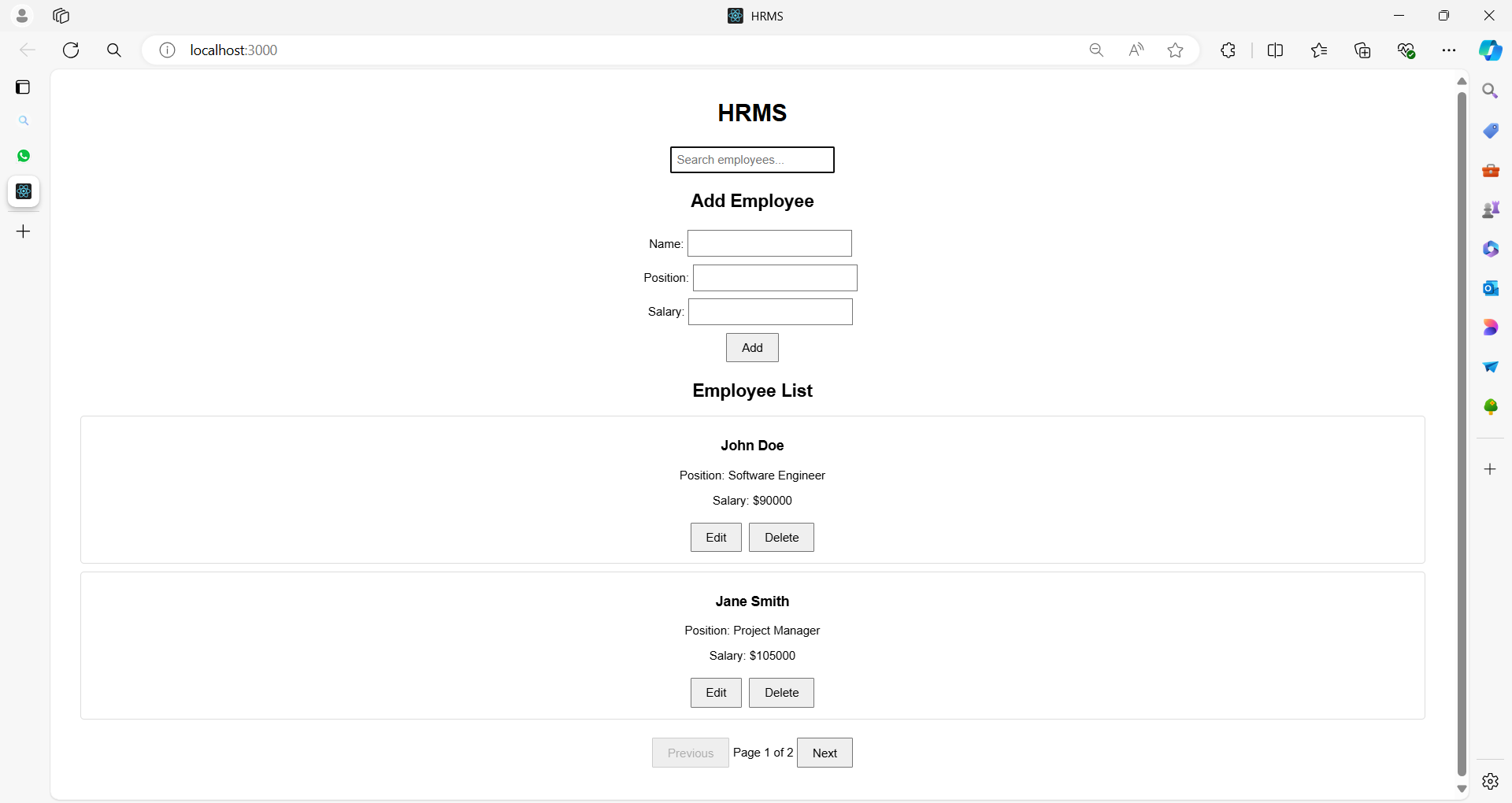
This setup provides a robust example of handling various types of events in a React application, demonstrating how to manage a more complex state and user interactions effectively.

**Example 04:**

The same HRMS example with additional components to make it more comprehensive.

We'll include:

1. **Search Component**: To filter the employee list based on the search query.
2. **Employee Detail Component**: To display detailed information about an employee when clicked.
3. **Pagination Component**: To manage pagination for large employee lists.
4. **Confirmation Dialog**: For confirming deletion of employee records.



**Expanded Project Structure**

**hrms-demo/**

|-- public/

| |-- index.html

|-- src/

| |-- App.js

| |-- EmployeeList.js

| |-- EmployeeForm.js

| |-- EmployeeItem.js

| |-- EmployeeDetail.js

| |-- Search.js

| |-- Pagination.js

| |-- ConfirmationDialog.js

| |-- App.css

|-- package.json

|-- README.md

**Code for Each File**

**src/App.js**

import React, { useState } from 'react';

import './App.css';

import EmployeeList from './EmployeeList';

import EmployeeForm from './EmployeeForm';

import Search from './Search';

import ConfirmationDialog from './ConfirmationDialog';

import Pagination from './Pagination';

const initialEmployees = [

{ id: 1, name: 'John Doe', position: 'Software Engineer', salary: 90000 },

{ id: 2, name: 'Jane Smith', position: 'Project Manager', salary: 105000 },

{ id: 3, name: 'Alice Johnson', position: 'UX Designer', salary: 75000 }

];

function App() {

const [employees, setEmployees] = useState(initialEmployees);

const [editingEmployee, setEditingEmployee] = useState(null);

const [searchQuery, setSearchQuery] = useState('');

const [currentPage, setCurrentPage] = useState(1);

const [itemsPerPage] = useState(2);

const [showConfirmDialog, setShowConfirmDialog] = useState(false);

const [employeeToDelete, setEmployeeToDelete] = useState(null);

const addEmployee = (employee) => {

setEmployees((prevEmployees) => [

...prevEmployees,

{ ...employee, id: Date.now() }

]);

setEditingEmployee(null);

};

const updateEmployee = (updatedEmployee) => {

setEmployees((prevEmployees) =>

prevEmployees.map((emp) =>

emp.id === updatedEmployee.id ? updatedEmployee : emp

)

);

setEditingEmployee(null);

};

const deleteEmployee = (id) => {

setEmployees((prevEmployees) => prevEmployees.filter((emp) => emp.id !== id));

setShowConfirmDialog(false);

};

const startEditEmployee = (employee) => {

setEditingEmployee(employee);

};

const handleSearchChange = (query) => {

setSearchQuery(query);

setCurrentPage(1); // Reset to the first page on search

};

const handlePageChange = (page) => {

setCurrentPage(page);

};

const handleDeleteClick = (employee) => {

setEmployeeToDelete(employee);

setShowConfirmDialog(true);

};

const filteredEmployees = employees.filter((employee) =>

employee.name.toLowerCase().includes(searchQuery.toLowerCase())

);

const paginatedEmployees = filteredEmployees.slice(

(currentPage - 1) \* itemsPerPage,

currentPage \* itemsPerPage

);

return (

<div className="App">

<h1>HRMS</h1>

<Search onSearchChange={handleSearchChange} />

<EmployeeForm

onAddEmployee={addEmployee}

onUpdateEmployee={updateEmployee}

editingEmployee={editingEmployee}

/>

<EmployeeList

employees={paginatedEmployees}

onDeleteEmployee={handleDeleteClick}

onEditEmployee={startEditEmployee}

/>

<Pagination

currentPage={currentPage}

totalItems={filteredEmployees.length}

itemsPerPage={itemsPerPage}

onPageChange={handlePageChange}

/>

{showConfirmDialog && (

<ConfirmationDialog

message={`Are you sure you want to delete ${employeeToDelete?.name}?`}

onConfirm={() => deleteEmployee(employeeToDelete.id)}

onCancel={() => setShowConfirmDialog(false)}

/>

)}

</div>

);

}

export default App;

**src/EmployeeList.js**

import React from 'react';

import EmployeeItem from './EmployeeItem';

function EmployeeList({ employees, onDeleteEmployee, onEditEmployee }) {

return (

<div className="employee-list">

<h2>Employee List</h2>

{employees.length === 0 ? (

<p>No employees found.</p>

) : (

employees.map((employee) => (

<EmployeeItem

key={employee.id}

employee={employee}

onDeleteEmployee={onDeleteEmployee}

onEditEmployee={onEditEmployee}

/>

))

)}

</div>

);

}

export default EmployeeList;

**src/EmployeeForm.js**

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

function EmployeeForm({ onAddEmployee, onUpdateEmployee, editingEmployee }) {

const [formData, setFormData] = useState({

name: '',

position: '',

salary: ''

});

useEffect(() => {

if (editingEmployee) {

setFormData({

name: editingEmployee.name,

position: editingEmployee.position,

salary: editingEmployee.salary

});

}

}, [editingEmployee]);

const handleChange = (e) => {

const { name, value } = e.target;

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

};

const handleSubmit = (e) => {

e.preventDefault();

if (editingEmployee) {

onUpdateEmployee({ ...editingEmployee, ...formData });

}

else {

onAddEmployee(formData);

}

setFormData({ name: '', position: '', salary: '' });

};

return (

<div className="employee-form">

<h2>{editingEmployee ? 'Edit Employee' : 'Add Employee'}</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</div>

<div>

<label>Position:</label>

<input

type="text"

name="position"

value={formData.position}

onChange={handleChange}

required

/>

</div>

<div>

<label>Salary:</label>

<input

type="number"

name="salary"

value={formData.salary}

onChange={handleChange}

required

/>

</div>

<button type="submit">{editingEmployee ? 'Update' : 'Add'}</button>

</form>

</div>

);

}

export default EmployeeForm;

**src/EmployeeItem.js**

import React from 'react';

function EmployeeItem({ employee, onDeleteEmployee, onEditEmployee }) {

return (

<div className="employee-item">

<h3>{employee.name}</h3>

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

<p>Salary: ${employee.salary}</p>

<button onClick={() => onEditEmployee(employee)}>Edit</button>

<button onClick={() => onDeleteEmployee(employee)}>Delete</button>

</div>

);

}

export default EmployeeItem;

**src/EmployeeDetail.js**

import React from 'react';

function EmployeeDetail({ employee, onClose }) {

if (!employee) return null;

return (

<div className="employee-detail">

<h2>Employee Details</h2>

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

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

<p><strong>Salary:</strong> ${employee.salary}</p>

<button onClick={onClose}>Close</button>

</div>

);

}

export default EmployeeDetail;

**src/Search.js**

import React from 'react';

function Search({ onSearchChange }) {

const handleChange = (e) => {

onSearchChange(e.target.value);

};

return (

<div className="search">

<input

type="text"

placeholder="Search employees..."

onChange={handleChange}

/>

</div>

);

}

export default Search;

**src/Pagination.js**

import React from 'react';

function Pagination({ currentPage, totalItems, itemsPerPage, onPageChange }) {

const totalPages = Math.ceil(totalItems / itemsPerPage);

const handlePageClick = (page) => {

if (page >= 1 && page <= totalPages) {

onPageChange(page);

}

};

return (

<div className="pagination">

<button onClick={() => handlePageClick(currentPage - 1)} disabled={currentPage === 1}>

Previous

</button>

<span>Page {currentPage} of {totalPages}</span>

<button onClick={() => handlePageClick(currentPage + 1)} disabled={currentPage === totalPages}>

Next

</button>

</div>

);

}

export default Pagination;

**src/ConfirmationDialog.js**

import React from 'react';

function ConfirmationDialog({ message, onConfirm, onCancel }) {

return (

<div className="confirmation-dialog">

<p>{message}</p>

<button onClick={onConfirm}>Confirm</button>

<button onClick={onCancel}>Cancel</button>

</div>

);

}

export default ConfirmationDialog;

**src/App.css**

.App {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

}

.employee-list, .employee-form {

margin: 20px;

}

.employee-item {

margin: 10px 0;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

}

button {

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

margin: 5px;

}

input {

padding: 8px;

font-size: 16px;

margin: 5px;

}

form {

display: flex;

flex-direction: column;

align-items: center;

}

.employee-detail {

padding: 20px;

border: 1px solid #ddd;

border-radius: 5px;

}

.search {

margin: 20px;

}

.pagination {

margin: 20px;

}

**Summary**

This expanded HRMS application includes:

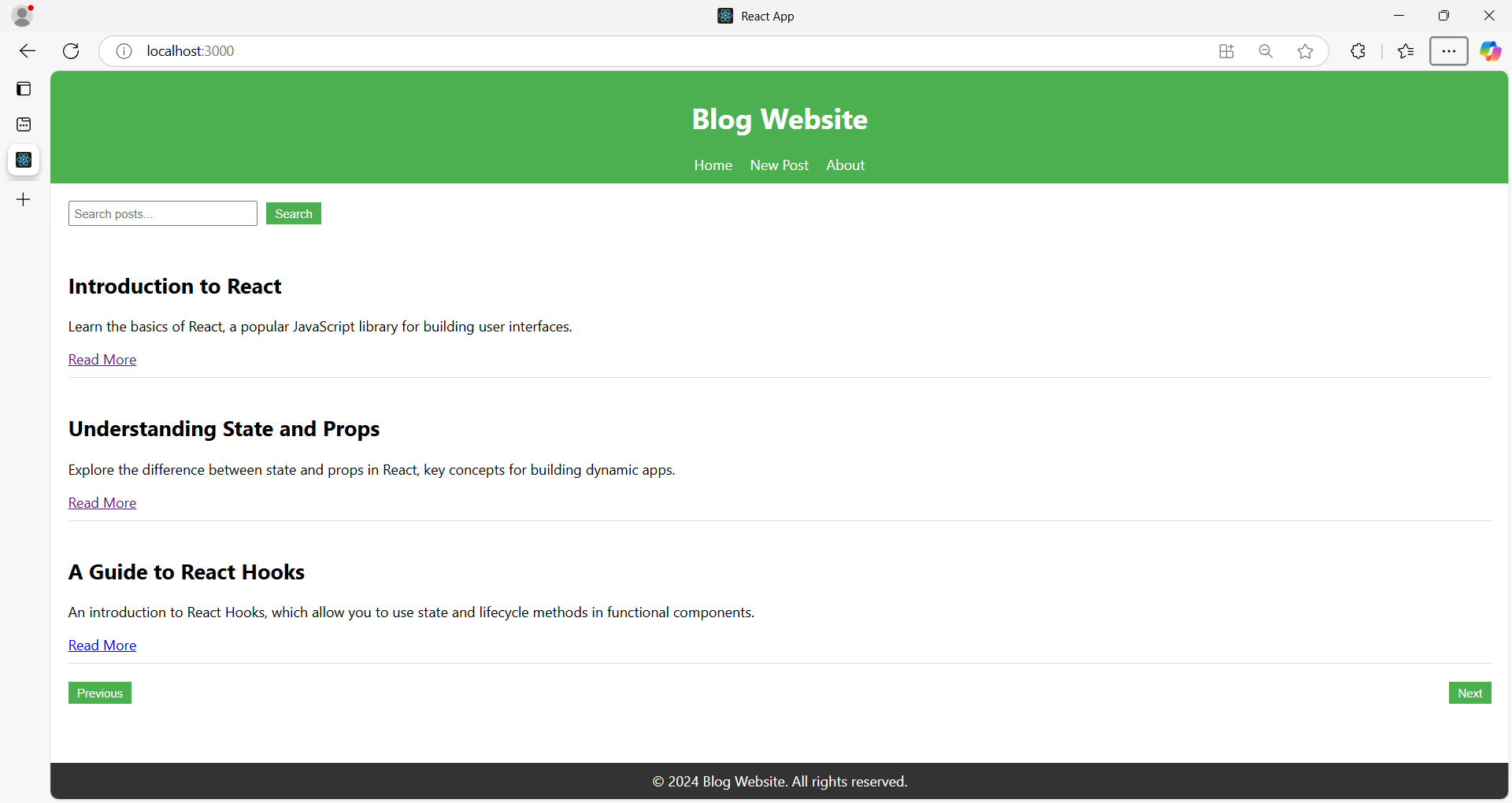
* **Search Component**: Allows users to search for employees.
* **Pagination Component**: Manages pagination for the employee list.
* **Confirmation Dialog**: Confirms deletion of an employee.
* **Employee Detail Component**: Displays detailed information about an employee.

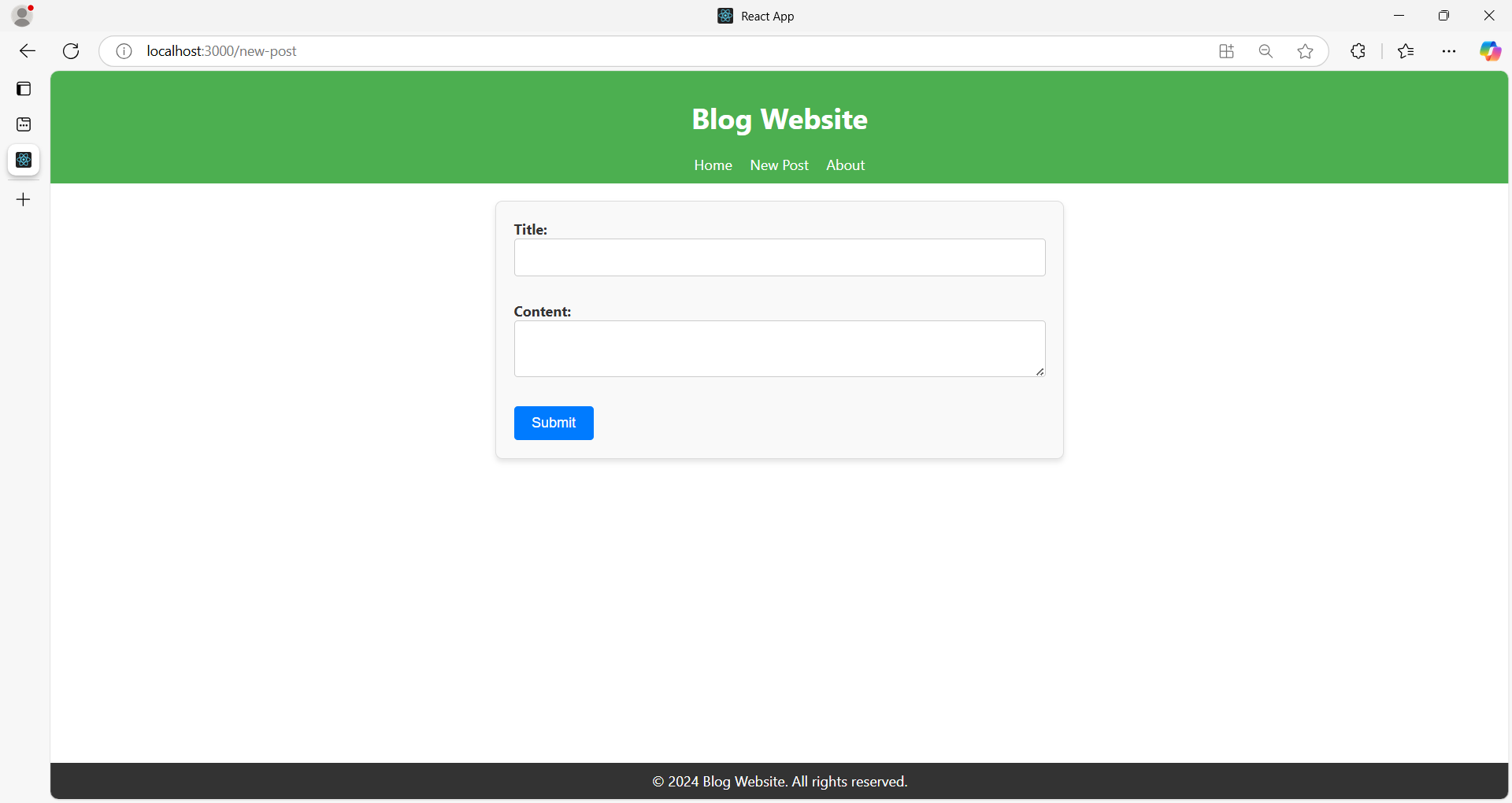
The App.js component orchestrates the application by managing the state for employees, handling form submissions, managing pagination, and handling search queries. The new components are integrated into this system to provide additional functionality and improve the overall user experience.

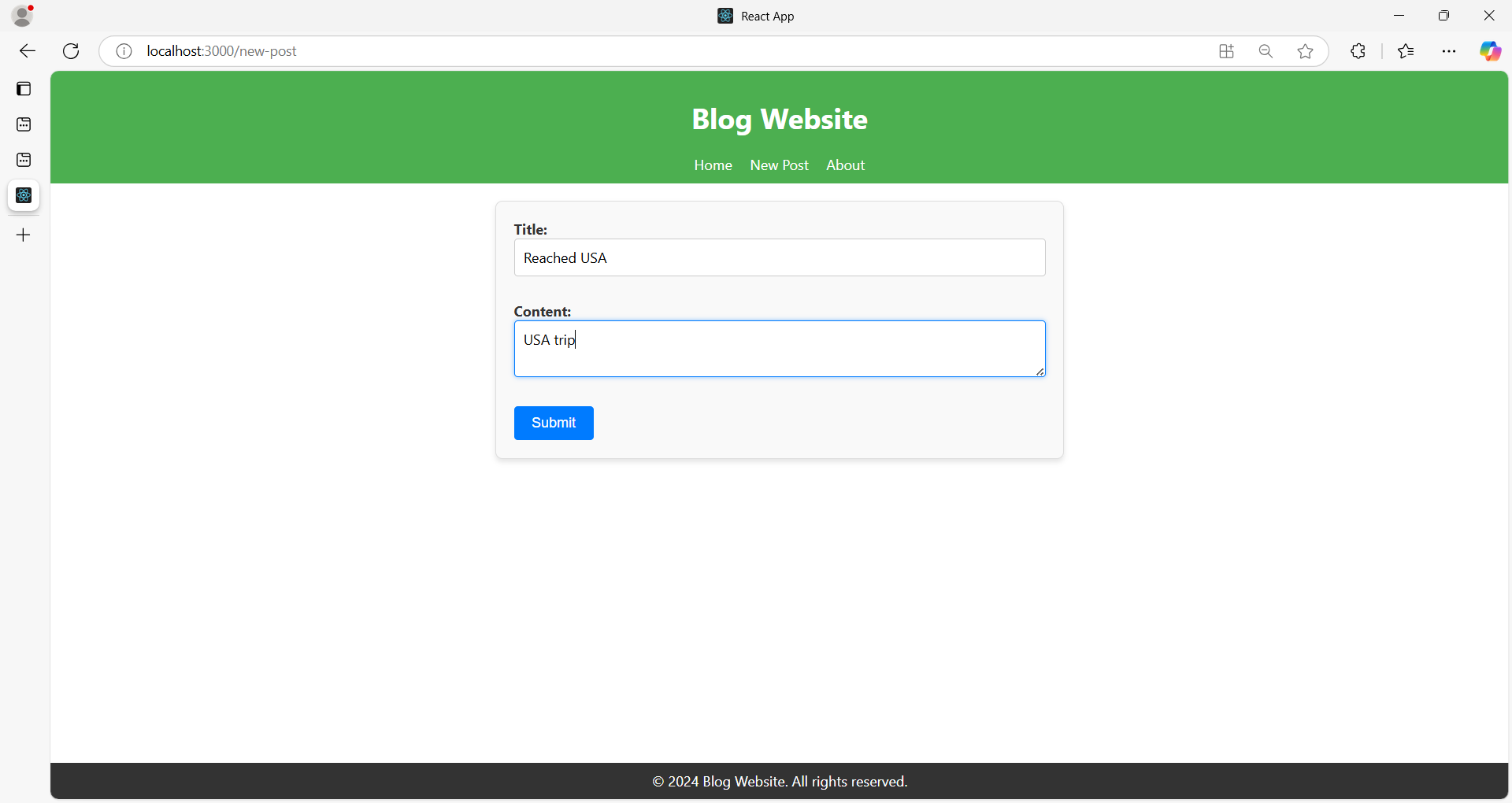
**Example 05**

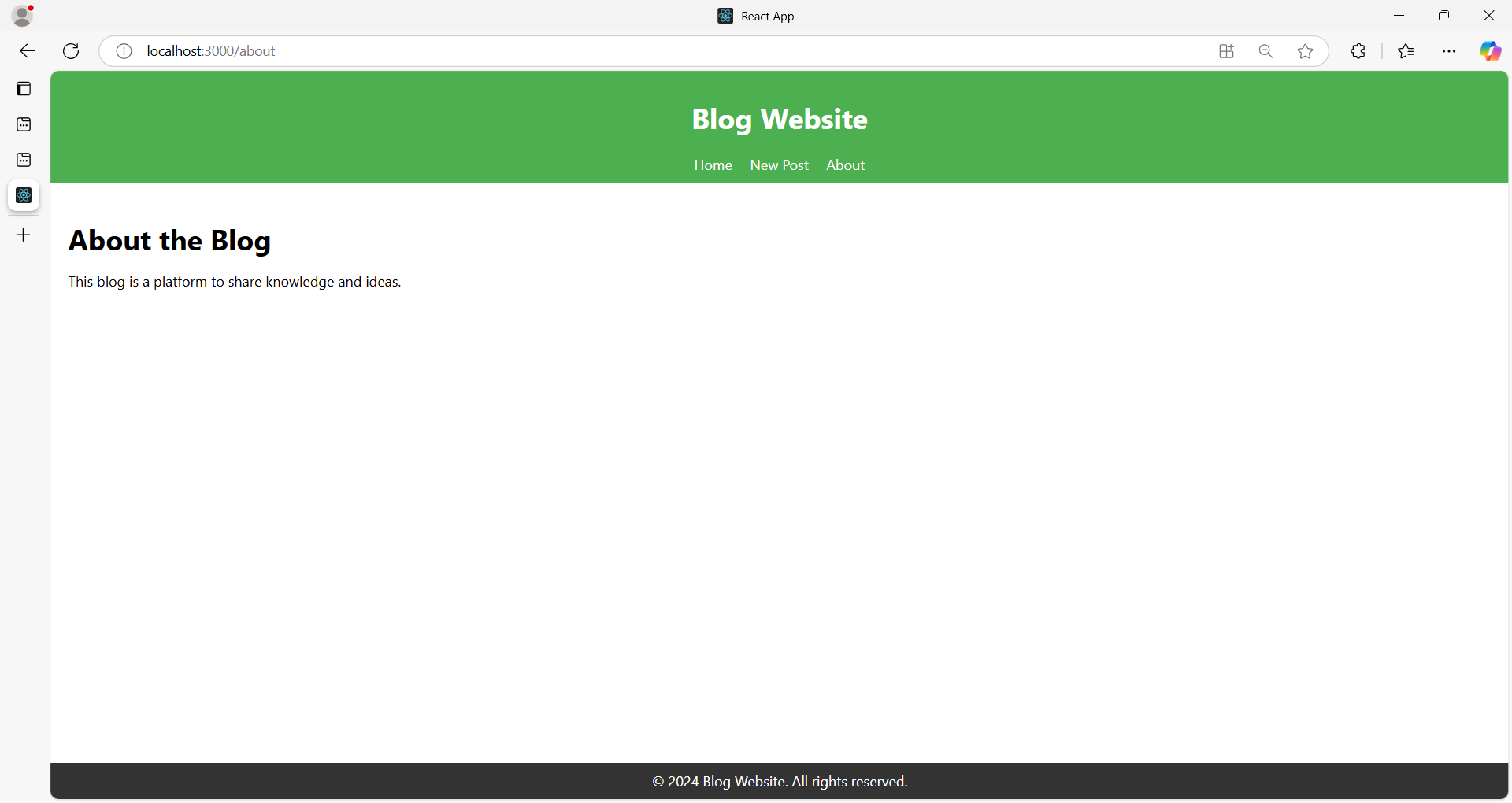
Let's create a blog website with a robust structure and multiple components.

The blog will include features like displaying posts, creating new posts, editing posts, deleting posts, user comments, user profiles, a search bar, pagination, and an about page.









**Project Structure**

**blog-website/**

|-- public/

| |-- index.html

|-- src/

| |-- components/

| | |-- PostList.js

| | |-- PostItem.js

| | |-- PostForm.js

| | |-- PostDetail.js

| | |-- CommentSection.js

| | |-- SearchBar.js

| | |-- Pagination.js

| | |-- UserProfile.js

| | |-- About.js

| | |-- Header.js

| | |-- Footer.js

| |-- styles/

| | |-- App.css

| | |-- PostList.css

| | |-- PostItem.css

| | |-- PostForm.css

| | |-- PostDetail.css

| | |-- CommentSection.css

| | |-- SearchBar.css

| | |-- Pagination.css

| | |-- UserProfile.css

| | |-- About.css

| | |-- Header.css

| | |-- Footer.css

|-- App.js

|-- package.json

|-- README.md

**Code Implementation**

**1. App.js**

import React from "react";

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

import Header from "./components/Header";

import Footer from "./components/Footer";

import PostList from "./components/PostList";

import PostDetail from "./components/PostDetail";

import PostForm from "./components/PostForm";

import About from "./components/About";

import UserProfile from "./components/UserProfile";

import "./styles/App.css";

function App() {

return (

<Router>

<div className="app">

<Header />

<Routes>

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

<Route path="/post/:id" element={<PostDetail />} />

<Route path="/new-post" element={<PostForm />} />

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

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

</Routes>

<Footer />

</div>

</Router>

);

}

export default App;

**2. Header.js**

import React from "react";

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

import "../styles/Header.css";

const Header = () => {

return (

<header className="header">

<h1>Blog Website</h1>

<nav>

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

<Link to="/new-post">New Post</Link>

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

</nav>

</header>

);

};

export default Header;

**Header.css**

.header {

background-color: #4caf50;

padding: 10px;

color: white;

text-align: center;

}

.header nav a {

margin: 0 10px;

color: white;

text-decoration: none;

}

**3. Footer.js**

import React from "react";

import "../styles/Footer.css";

const Footer = () => {

return <footer className="footer">© 2024 Blog Website. All rights reserved.</footer>;

};

export default Footer;

**Footer.css**

.footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px 0;

position: fixed;

width: 100%;

bottom: 0;

}

**4. PostList.js**

import React from "react";

import PostItem from "./PostItem";

import SearchBar from "./SearchBar";

import Pagination from "./Pagination";

import "../styles/PostList.css";

const PostList = () => {

const posts = [

    {

      "id": 1,

      "title": "Introduction to React",

      "summary": "Learn the basics of React, a popular JavaScript library for building user interfaces."

    },

    {

      "id": 2,

      "title": "Understanding State and Props",

      "summary": "Explore the difference between state and props in React, key concepts for building dynamic apps."

    },

    {

      "id": 3,

      "title": "A Guide to React Hooks",

      "summary": "An introduction to React Hooks, which allow you to use state and lifecycle methods in functional components."

    },

    {

      "id": 4,

      "title": "Styling in React",

      "summary": "Discover various styling techniques in React, including CSS, CSS-in-JS, and inline styles."

    },

    {

      "id": 5,

      "title": "Using React Router",

      "summary": "Understand how to navigate between views in your React application using React Router."

    }

  ]

return (

<div className="post-list">

<SearchBar />

{posts.map((post) => (

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

))}

<Pagination />

</div>

);

};

export default PostList;

**PostList.css**

.post-list {

padding: 20px;

}

**5. PostItem.js**

import React from "react";

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

import "../styles/PostItem.css";

const PostItem = ({ post }) => {

return (

<div className="post-item">

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

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

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

</div>

);

};

export default PostItem;

**PostItem.css**

.post-item {

border-bottom: 1px solid #ddd;

margin-bottom: 10px;

padding: 10px 0;

}

**6. PostForm.js**

import React, { useState } from "react";

import "../styles/PostForm.css";

const PostForm = () => {

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

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

const handleSubmit = (e) => {

e.preventDefault();

console.log({ title, content });

};

return (

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

<label>

Title:

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

</label>

<label>

Content:

<textarea value={content} onChange={(e) => setContent(e.target.value)} />

</label>

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

</form>

);

};

export default PostForm;

**PostForm.css**

.post-form {

  max-width: 600px;

  margin: 20px auto;

  padding: 20px;

  border: 1px solid #ddd;

  border-radius: 8px;

  background-color: #f9f9f9;

  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.post-form label {

  display: block;

  font-size: 1rem;

  font-weight: bold;

  color: #333;

  margin-bottom: 8px;

}

.post-form input[type="text"],

.post-form textarea {

  width: 100%;

  padding: 10px;

  margin-bottom: 20px;

  border: 1px solid #ccc;

  border-radius: 4px;

  font-size: 1rem;

  font-family: inherit;

  box-sizing: border-box;

}

.post-form input[type="text"]:focus,

.post-form textarea:focus {

  border-color: #007bff;

  outline: none;

  box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

.post-form button {

  display: inline-block;

  padding: 10px 20px;

  font-size: 1rem;

  color: #fff;

  background-color: #007bff;

  border: none;

  border-radius: 4px;

  cursor: pointer;

  transition: background-color 0.3s ease;

}

.post-form button:hover {

  background-color: #0056b3;

}

**7. PostDetail.js**

import React from "react";

import CommentSection from "./CommentSection";

import "../styles/PostDetail.css";

const PostDetail = () => {

const post = {

id: 1,

title: "First Post",

content: "This is the detailed content of the first post.",

};

return (

<div className="post-detail">

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

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

<CommentSection />

</div>

);

};

export default PostDetail;

**PostDetail.css**

.post-detail {

padding: 20px;

border: 1px solid #ddd;

border-radius: 5px;

}

**8. CommentSection.js**

import React, { useState } from "react";

import "../styles/CommentSection.css";

const CommentSection = () => {

const [comments, setComments] = useState(["Great post!", "Very helpful."]);

const [newComment, setNewComment] = useState("");

const handleAddComment = () => {

if (newComment.trim()) {

setComments([...comments, newComment]);

setNewComment("");

}

};

return (

<div className="comment-section">

<h3>Comments</h3>

<ul>

{comments.map((comment, index) => (

<li key={index}>{comment}</li>

))}

</ul>

<input

type="text"

placeholder="Add a comment"

value={newComment}

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

/>

<button onClick={handleAddComment}>Add Comment</button>

</div>

);

};

export default CommentSection;

**CommentSection.css**

.comment-section {

margin-top: 20px;

}

.comment-section ul {

list-style-type: none;

padding: 0;

}

.comment-section li {

background-color: #f9f9f9;

padding: 5px;

margin-bottom: 5px;

border-radius: 3px;

}

**9. SearchBar.js**

import React from "react";

import "../styles/SearchBar.css";

const SearchBar = () => {

return (

<div className="search-bar">

<input type="text" placeholder="Search posts..." />

<button>Search</button>

</div>

);

};

export default SearchBar;

**SearchBar.css**

.search-bar {

margin-bottom: 20px;

}

.search-bar input {

padding: 5px;

width: 200px;

margin-right: 10px;

}

.search-bar button {

padding: 5px 10px;

background-color: #4caf50;

color: white;

border: none;

cursor: pointer;

}

**10. Pagination.js**

import React from "react";

import "../styles/Pagination.css";

const Pagination = () => {

return (

<div className="pagination">

<button>Previous</button>

<button>Next</button>

</div>

);

};

export default Pagination;

**Pagination.css**

.pagination {

margin-top: 20px;

display: flex;

justify-content: space-between;

}

.pagination button {

padding: 5px 10px;

background-color: #4caf50;

color: white;

border: none;

cursor: pointer;

}

**11. UserProfile.js**

import React from "react";

import "../styles/UserProfile.css";

const UserProfile = () => {

const user = {

name: "John Doe",

bio: "A passionate blogger.",

};

return (

<div className="user-profile">

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

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

</div>

);

};

export default UserProfile;

**UserProfile.css**

.user-profile {

padding: 20px;

text-align: center;

}

.user-profile h1 {

margin-bottom: 10px;

}

.user-profile p {

color: #555;

}

**12. About.js**

import React from "react";

import "../styles/About.css";

const About = () => {

return (

<div className="about">

<h1>About the Blog</h1>

<p>This blog is a platform to share knowledge and ideas.</p>

</div>

);

};

export default About;

**About.css**

.about {

padding: 20px;

}

.about h1 {

margin-bottom: 10px;

}

To install it, run the following command in your project directory:

**npm install react-router-dom**