**Assignment 1**

**HTML(index.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Todo App</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="app">

<h1>Todo App</h1>

<form id="todoForm">

<input type="text" id="todoInput" placeholder="Add a new todo">

<button type="button" onclick="addTodo()">Add Todo</button>

</form>

<ul id="todoList"></ul>

</div>

<script src="app.js"></script>

</body>

</html>

**CSS(Style.css)**

/\* Add your styling here if needed \*/

.completed {

text-decoration: line-through;

}

**JS(app.js)**

// Get references to elements

const todoForm = document.getElementById('todoForm');

const todoInput = document.getElementById('todoInput');

const todoList = document.getElementById('todoList');

// Function to add a new todo

function addTodo() {

// Get the input value

const todoText = todoInput.value.trim();

// Check if the input is not empty

if (todoText !== '') {

// Create a new list item

const listItem = document.createElement('li');

listItem.textContent = todoText;

// Add click event listener to toggle strikethrough

listItem.addEventListener('click', function() {

this.classList.toggle('completed');

});

// Append the new list item to the todo list

todoList.appendChild(listItem);

// Clear the input field

todoInput.value = '';

}

}

**Assignment 2**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Todo App</title>

</head>

<body>

<div id="app"></div>

<script src="https://unpkg.com/react@17/umd/react.development.js"></script>

<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6.26.0/babel.min.js"></script>

<script src="https://unpkg.com/prop-types@15/umd/prop-types.js"></script>

<script type="text/babel" src="app.jsx"></script>

</body>

</html>

**App.js**

// Define a functional component HelpForm

function HelpForm() {

// Define state to keep track of the form value

const [formValue, setFormValue] = React.useState('');

// Event handler to update the form value in state

const handleInputChange = (event) => {

setFormValue(event.target.value);

};

return (

<div>

<h1>Todo App</h1>

{/\* Form with an input and a paragraph \*/}

<form>

<label>

Type something:

<input type="text" value={formValue} onChange={handleInputChange} />

</label>

</form>

{/\* Display what the user has typed \*/}

<p>You typed: {formValue}</p>

</div>

);

}

// Render the HelpForm component to the root div in the HTML

ReactDOM.render(<HelpForm />, document.getElementById('app'));

**Assignment 3**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>React Document Title App</title>

</head>

<body>

<div id="app"></div>

<script src="https://unpkg.com/react@17/umd/react.development.js"></script>

<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6.26.0/babel.min.js"></script>

<script type="text/babel" src="app.jsx"></script>

</body>

</html>

**React (App.js)**

// Custom hook for setting document title

function useDocumentTitle(title) {

React.useEffect(() => {

document.title = title;

}, [title]);

}

// Main App component

function App() {

// State variable to store the title

const [title, setTitle] = React.useState("React Document Title App");

// Custom hook to set the document title

useDocumentTitle(title);

// Event handler to update the title state

const handleInputChange = (event) => {

setTitle(event.target.value);

};

return (

<div>

<h1>React Document Title App</h1>

{/\* Input to set the title \*/}

<label>

Set Document Title:

<input type="text" value={title} onChange={handleInputChange} />

</label>

</div>

);

}

// Render the App component to the root div in the HTML

ReactDOM.render(<App />, document.getElementById('app'));

**Assignment 4**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>React Form Validation</title>

</head>

<body>

<div id="app"></div>

<script src="https://unpkg.com/react@17/umd/react.development.js"></script>

<script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6.26.0/babel.min.js"></script>

<script type="text/babel" src="app.jsx"></script>

</body>

</html>

**App.js**

// Main App component

function App() {

// State variables for form inputs and errors

const [formData, setFormData] = React.useState({

username: '',

password: '',

email: '',

mobile: '',

});

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

// Function to handle form submission

const handleSubmit = (event) => {

event.preventDefault();

// Validate form data

const validationErrors = validateForm(formData);

setErrors(validationErrors);

// If there are no validation errors, log the data to the console

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

console.log('Form Data:', formData);

}

};

// Function to handle input changes

const handleInputChange = (event) => {

const { name, value } = event.target;

setFormData({

...formData,

[name]: value,

});

};

// Function to validate form data

const validateForm = (data) => {

const errors = {};

// Validate username

if (!data.username.trim()) {

errors.username = 'Username is required';

}

// Validate password

if (!data.password.trim()) {

errors.password = 'Password is required';

}

// Validate email

if (!data.email.trim()) {

errors.email = 'Email is required';

} else if (!isValidEmail(data.email)) {

errors.email = 'Invalid email format';

}

// Validate mobile number

if (!data.mobile.trim()) {

errors.mobile = 'Mobile number is required';

} else if (!isValidMobile(data.mobile)) {

errors.mobile = 'Invalid mobile number';

}

return errors;

};

// Function to validate email format

const isValidEmail = (email) => {

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return emailRegex.test(email);

};

// Function to validate mobile number format

const isValidMobile = (mobile) => {

const mobileRegex = /^[0-9]{10}$/;

return mobileRegex.test(mobile);

};

return (

<div>

<h1>React Form Validation</h1>

{/\* Form \*/}

<form onSubmit={handleSubmit}>

{/\* Username \*/}

<div>

<label>Username:</label>

<input

type="text"

name="username"

value={formData.username}

onChange={handleInputChange}

/>

{errors.username && <p>{errors.username}</p>}

</div>

{/\* Password \*/}

<div>

<label>Password:</label>

<input

type="password"

name="password"

value={formData.password}

onChange={handleInputChange}

/>

{errors.password && <p>{errors.password}</p>}

</div>

{/\* Email \*/}

<div>

<label>Email:</label>

<input

type="text"

name="email"

value={formData.email}

onChange={handleInputChange}

/>

{errors.email && <p>{errors.email}</p>}

</div>

{/\* Mobile Number \*/}

<div>

<label>Mobile Number:</label>

<input

type="text"

name="mobile"

value={formData.mobile}

onChange={handleInputChange}

/>

{errors.mobile && <p>{errors.mobile}</p>}

</div>

{/\* Submit Button \*/}

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

</form>

</div>

);

}

**Assignment 5**

npx create-react-app my-react-app

cd my-react-app

npm install react-router-dom

**App.js**

// src/App.js

import React from 'react';

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

const users = [

{ id: 1, name: 'John Doe' },

{ id: 2, name: 'Jane Doe' },

// Add more users as needed

];

const posts = [

{ userId: 1, id: 1, title: 'Post 1', body: 'Body of Post 1' },

{ userId: 1, id: 2, title: 'Post 2', body: 'Body of Post 2' },

{ userId: 2, id: 3, title: 'Post 3', body: 'Body of Post 3' },

// Add more posts as needed

];

function App() {

return (

<Router>

<div>

<h1>Single Page App</h1>

<Switch>

{/\* Route for user information \*/}

<Route path="/user/:userId" component={UserDetails} />

{/\* Default route \*/}

<Route path="/" exact render={() => <div>Home Page</div>} />

{/\* Route for user not found \*/}

<Route render={() => <div>No User Found</div>} />

</Switch>

</div>

</Router>

);

}

export default App;

**Create two components, one for displaying user details and another for displaying post details**

// src/components/UserDetails.js

import React from 'react';

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

const UserDetails = ({ match }) => {

const userId = parseInt(match.params.userId);

const user = users.find((user) => user.id === userId);

if (!user) {

return <div>No User Found</div>;

}

return (

<div>

<h2>User Details</h2>

<p>ID: {user.id}</p>

<p>Name: {user.name}</p>

<h3>Posts</h3>

<ul>

{posts

.filter((post) => post.userId === userId)

.map((post) => (

<li key={post.id}>

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

</li>

))}

</ul>

{/\* Nested route for displaying post details \*/}

<Route path={`${match.path}/post/:postId`} component={PostDetails} />

</div>

);

};

export default UserDetails;

// src/components/PostDetails.js

import React from 'react';

const PostDetails = ({ match }) => {

const postId = parseInt(match.params.postId);

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

if (!post) {

return <div>No Post Found</div>;

}

return (

<div>

<h4>Post Details</h4>

<p>ID: {post.id}</p>

<p>Title: {post.title}</p>

<p>Body: {post.body}</p>

</div>

);

};

export default PostDetails;

**Update the src/App.js file to import and use the components:**

// src/App.js

import React from 'react';

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

import UserDetails from './components/UserDetails';

function App() {

return (

<Router>

<div>

<h1>Single Page App</h1>

<Switch>

{/\* Route for user information \*/}

<Route path="/user/:userId" component={UserDetails} />

{/\* Default route \*/}

<Route path="/" exact render={() => <div>Home Page</div>} />

{/\* Route for user not found \*/}

<Route render={() => <div>No User Found</div>} />

</Switch>

</div>

</Router>

);

}

export default App;

**Assignment 6**

npx create-react-app my-react-app

cd my-react-app

In the **src** folder, create five components: **Main**, **Cart**, **Product**, **ProductList**, and **TotalPrice**.

// src/components/Main.js

import React from 'react';

import ProductList from './ProductList';

import Cart from './Cart';

import TotalPrice from './TotalPrice';

function Main() {

return (

<div>

<h1>Main Component</h1>

<ProductList />

<Cart />

<TotalPrice />

</div>

);

}

export default Main;

// src/components/Cart.js

import React from 'react';

function Cart({ cartItems }) {

return (

<div>

<h2>Cart Component</h2>

<ul>

{cartItems.map((item) => (

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

))}

</ul>

</div>

);

}

export default Cart;

// src/components/Product.js

import React from 'react';

function Product({ product, onAddToCart }) {

return (

<div>

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

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

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

</div>

);

}

export default Product;

// src/components/ProductList.js

import React from 'react';

import Product from './Product';

const products = [

{ id: 1, name: 'Product 1', price: 10 },

{ id: 2, name: 'Product 2', price: 20 },

{ id: 3, name: 'Product 3', price: 30 },

{ id: 4, name: 'Product 4', price: 40 },

{ id: 5, name: 'Product 5', price: 50 },

];

function ProductList({ onAddToCart }) {

return (

<div>

<h2>Product List Component</h2>

{products.map((product) => (

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

))}

</div>

);

}

export default ProductList;

// src/components/TotalPrice.js

import React from 'react';

function TotalPrice({ totalPrice }) {

return (

<div>

<h2>Total Price Component</h2>

<p>Total Price: ${totalPrice}</p>

</div>

);

}

export default TotalPrice;

**Replace the content of src/App.js with the following:**

// src/App.js

import React, { useState } from 'react';

import Main from './components/Main';

function App() {

const [cartItems, setCartItems] = useState([]);

const [totalPrice, setTotalPrice] = useState(0);

const addToCart = (product) => {

setCartItems([...cartItems, product]);

setTotalPrice(totalPrice + product.price);

};

return (

<div className="App">

<Main

onAddToCart={addToCart}

cartItems={cartItems}

totalPrice={totalPrice}

/>

</div>

);

}

export default App;

npm start