React Model Lab Solutions

1.

ccomponent.js:

import React from 'react'

class Ccomponent extends React.Component {state = { }

render() {

return (

<div>

<h1>Class Component:</h1>

<h2>Hello Everyone,Welcome To React</h2>

</div>

);

}

}

export default Ccomponent;

render it in index.js

2.

conditionalrendering.js:

import React, { Component } from 'react';import 'bootstrap/dist/css/bootstrap.css';

class Conditionalrendering extends Component {state = { count: 0 ,name:"Zero"}

changeCount(){ if(this.state.count===0 )

return"Zero";

else if(this.state.count===10)return"Ten";

else

return this.state.count;

}

changeColor1(){ let str="btn btn-";

if(this.state.count===0){str+="danger";

}

else {

str+="success";

}

return str;

}

changeColor2(){

let str="btn btn-"; if(this.state.count===10){str+="danger";

}

else {

str+="success";

}

return str;

}

render() {

return (<div style={{textAlign:'center',padding:50}}>

<h1 style={{padding:100}}> Conditional Rendering:</h1>

<button className={this.changeColor1()}style={{width:100,height:50}}

disabled={this.state.count===0} onClick={() =>

{ this.setState({ count: this.state.count - 1 }) }}>- </button>

<span style={{padding:50,fontSize:27}}>{this.changeCount()}</span>

<button className={this.changeColor2()}

style={{width:100,height:50}}disabled={this.state.count===10}

onClick={() => { this.setState({ count: this.state.count + 1 }) }}>+</button>

</div>);

}

}

export default Conditionalrendering;

Index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';import Conditional from './conditional';

const root = ReactDOM.createRoot(document.getElementById('root'));root.render(

<React.StrictMode>

<Conditional/>

</React.StrictMode>);

3.

parent.js:

import React from 'react'; import Child from './child';

class Parent extends React.Component{state = {

name: "",

}

handleCallback = (childData) =>{ this.setState({name: childData})

}

render(){

const {name} = this.state;return(

<div style={{textAlign:'center'}}>

<h1>Child and Parent Components</h1>

<Child parentCallback = {this.handleCallback}/>

{name}

</div>

)

}

}

export default Parent;

Child.js:

import React from 'react'

class Child extends React.Component{

onTrigger = (event) => {

this.props.parentCallback(event.target.myname.value);event.preventDefault();

}

render(){

return(

<div>

</div>

)

<form onSubmit = {this.onTrigger}>

<input type = "text"

name = "myname" placeholder = "Enter Name"/>

<br></br><br></br>

<input type = "submit" value = "Submit"/>

<br></br><br></br>

</form>

4.

import \* as React from 'react';

import Card from '@mui/material/Card';

import CardActions from '@mui/material/CardActions';import CardContent from

'@mui/material/CardContent';import CardMedia from '@mui/material/CardMedia'; import Button

from '@mui/material/Button';

import Typography from '@mui/material/Typography';

export default function MediaCard() {

return (

<Card sx={{ maxWidth: 345 }}>

<CardMedia component="img"height="140"

image="

0eQQqbmJJyyw5uaqPyjJYna8yYStYSTmNCA+agBA+EsBXjrWNR0VxSpiJqlAODLDgubLLU

o1vHdFxb9CHlUqXubGSaJ5DygjhNkonoClKUChbhm+HIqrg6iA8nEpDB/A7o1HR+ozCu938Ij

mTMIp82Ckl+ /rjE3wp5woUA1wVtTHRbthQoZPY2ZeI4UKGhM//2Q=="

alt="green iguana"

/>

<CardContent>

<Typography gutterBottom variant="h5" component="div">SKCET

</Typography>

<Typography variant="body2" color="text.secondary">

SKCET is one of the finest colleges in South India.It is well known for it's infrastructure,

academics and the placements offered. With Asia's second-largest library, SKCET is well -knownfor

it's architecture and maintenance.

</Typography>

</CardContent>

<CardActions>

<Button size="small">Share</Button>

<Button size="small">Learn More</Button>

</CardActions>

</Card>

);

}

5.

import \* as React from 'react';

import AppBar from '@mui/material/AppBar';

import Box from '@mui/material/Box';

import Toolbar from '@mui/material/Toolbar';

import IconButton from '@mui/material/IconButton';

import Typography from '@mui/material/Typography';import Menu from '@mui/material/Menu';

import MenuIcon from '@mui/icons-material/Menu';import Container from

'@mui/material/Container';

import Avatar from '@mui/material/Avatar'; import Button from '@mui/material/Button';

import Tooltip from '@mui/material/Tooltip'; import MenuItem from '@mui/material/MenuItem';

import AdbIcon from '@mui/icons-material/Adb';

const pages = ['Products', 'Pricing', 'Blog'];

const settings = ['Profile', 'Account', 'Dashboard', 'Logout'];

function ResponsiveAppBar() {

const [anchorElNav, setAnchorElNav] = React.useState(null);const [anchorElUser, setAnchorElUser]

= React.useState(null);

const handleOpenNavMenu = (event) => {setAnchorElNav(event.currentTarget);

};

const handleOpenUserMenu = (event) => {setAnchorElUser(event.currentTarget);

};

const handleCloseNavMenu = () => {setAnchorElNav(null);

};

const handleCloseUserMenu = () => {setAnchorElUser(null);

};

return (

<AppBar position="static">

<Container maxWidth="xl">

<Toolbar disableGutters>

<AdbIcon sx={{ display: { xs: 'none', md: 'flex' }, mr: 1 }} />

<Typography variant="h6" noWrapcomponent="a" href="/" sx={{ mr:

2, display: { xs: 'none', md: 'flex' },fontFamily: 'monospace', fontWeight: 700, letterSpacing:

'.3rem', color:'inherit', textDecoration:'none',

}}> SKCET

</Typography>

<Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>

<IconButtonsize="large"

aria-label="account of current user" aria-controls="menu-appbar" ariahaspopup="true"

onClick={handleOpenNavMenu} color="inherit">

<MenuIcon />

</IconButton>

<Menu id="menuappbar"anchorEl={anchorElNav}anchorOrigin={{ vertical:

'bottom', horizontal:

'left',

}}

keepMounted transformOrigin={{ vertical:

'top', horizontal:

'left',

}}

open={Boolean(anchorElNav)} onClose={handleCloseNavMenu} sx={{ display: { xs: 'block',md:

'none' },

}}>

{pages.map((page) => (

<MenuItem key={page} onClick={handleCloseNavMenu}>

<Typography textAlign="center">{page}</Typography>

</MenuItem>

))}

</Menu>

</Box>

<AdbIcon sx={{ display: { xs: 'flex', md: 'none' }, mr: 1 }} />

<Typography variant="h5" noWrapcomponent="a" href="" sx={{ mr:

2,display: { xs: 'flex', md: 'none' }, flexGrow: 1, fontFamily: 'monospace',fontWeight: 700,

letterSpacing:'.3rem',color: 'inherit', textDecoration: 'none',

}}>

SKCET

</Typography>

<Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' } }}>

{pages.map((page) => (

<Button key={page}

onClick={handleCloseNavMenu}

sx={{ my: 2, color: 'white', display: 'block' }}

>

{page}

</Button>

))}

</Box>

<Box sx={{ flexGrow: 0 }}>

<Tooltip title="Open settings">

<IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>

<Avatar alt="Remy Sharp" src="/static/images/avatar/2.jpg" />

</IconButton>

</Tooltip>

<Menu sx={{ mt:

'45px' }} id="menuappbar"anchorEl={anchorElUser}

anchorOrigin={{ vertical:

'top', horizontal: 'right',

}}

keepMounted transformOrigin={{ vertical:

'top', horizontal:

'right',

}}

open={Boolean(anchorElUser)} onClose={handleCloseUserMenu}

>

{settings.map((setting) => (

<MenuItem key={setting} onClick={handleCloseUserMenu}>

<Typography textAlign="center">{setting}</Typography>

</MenuItem>

))}

</Menu>

</Box>

</Toolbar>

</Container>

</AppBar>

);

}

export default ResponsiveAppBar;

App.js

import React from 'react' import Hello from'./over'; import Card from './button';

import UpdatedComponent from './update';import './style.css';

import ResponsiveAppBar from './navbar';function App(){

return(

<ResponsiveAppBar/>

);

}

export default App

6.

Programfetch.js:

import { useEffect, useState } from "react";

export default function FetchAPI(){

const [user, setUser] = useState([]);useEffect(() => {

fetch('https://jsonplaceholder.typicode.com/users')

.then(res => res.json())

.then(res => setUser(res))

})

console.log(user)return(

<div className="main">

{user.map(u => (

<div>{u.name}, {u.id}, {u.email}</div>

))}

</div>

)

}

App.js:

import './App.css';

import FetchAPI from './fetch';function App() {

return (<>

<FetchAPI />

</>);

}

export default App;

7.

App.js

import Dropdown from "./main";export default function App() { return (

<div className="App">

<Dropdown placeHolder="Select..." />

</div>

);

}

Dropdown.css

.dropdown-container {text-align: left;

border: 2px solid rgb(57, 19, 196);position: relative;

border-radius: 5px;

}

.dropdown-input {padding: 20px; display: flex;

align-items: center;

justify-content: space-between;user-select: none;

}

Dropdown.js

import \* as React from 'react';const App = () => {

return (

<div>

<select>

<option value="fruit">Fruit</option

<option value="vegetable">Vegetable</option>

<option value="meat">Meat</option>

<option value="Groceries">Groceries</option>

<option value="Snacks">Snacks</option>

<option value="Footwears">Footwears</option>

</select>

</div>

);

};

export default App;

8.

App.js

import './mainpage.css'; import './returnpage.css';

import Mainpage from './mainpage' import Returnpage from './returnpage'

import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';function App() {

return(

<Router>

<div className='App'>

<Routes>

<Route exact path='/' element={<Mainpage />}></Route>

<Route exact path='/mainpage' element={<Mainpage />}></Route>

<Route exact path='/returnpage' element={<Returnpage />}></Route>

</Routes>

</div>

</Router>

)

}

export default App;

App.test.js

import React from 'react';

import ReactDOM from 'react-dom';import App from './App';

it('renders without crashing', () => {

const div = document.createElement('div');ReactDOM.render(<App />, div);

});

Index.css

html, body, #root, #root>div {height: 100%;

}

body { margin: 0;

padding: 0;

font-family: sans-serif;height: 100%;

}

Index.js

import React from 'react';

import ReactDOM from 'react-dom';import App from './App';

import './index.css';

ReactDOM.render(

<App />, document.getElementById('root')

);

Mainpage.js

//import React from 'react';import "./mainpage.css" import "./returnpage.css"

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

const Mainpage = () => {return (<>

<div>

<h1>This is Mainpage</h1>

<Link to="/returnpage" className="v2\_21">Next Page</Link>

</div>

</>

);

};

export default Mainpage;

mainpage.css

.v2\_21 {

top: 100px;left: 50vh;

position: absolute;font-family: Inter;font-weight: Bold;font-size: 30px; text-align: center;

}

Returnpage.js

import "./mainpage.css" import "./returnpage.css"

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

const Returnpage = () => {return (<>

<div>

<h1>This is Next Page</h1>

<Link to='/mainpage' class="v2\_21">Back to mainpage</Link>

</div>

</>

);

};

export default Returnpage;

returnpage.css

.v2\_21 {

top: 100px;left: 50vh;

position: absolute;font-family: Inter;font-weight: Bold;font-size: 30px; text-align: center;

}

9.

Form validation in react Form.js

import React from 'react'; import './form.css';

class RegisterForm extends React.Component { constructor() {super(); this.state = {

fields: {},

errors: {}

}

this.handleChange = this.handleChange.bind(this); this.submituserRegistrationForm =

this.submituserRegistrationForm.bind(this);

};

handleChange(e) {

let fields = this.state.fields; fields[e.target.name] = e.target.value; this.setState({fields

});

}

submituserRegistrationForm(e) { e.preventDefault();

if (this.validateForm()) { let fields = {}; fields["username"] = "";fields["mobileno"] = "";

fields["password"] = ""; this.setState({fields:fields}); alert("Form submitted");

}

}

validateForm() {

let fields = this.state.fields; let errors = {};

let formIsValid = true; if (!fields["username"]) {formIsValid = false;

errors["username"] = "\*Please enter your username.";

}

if (typeof fields["username"] !== "undefined") {

if (!fields["username"].match(/^[a-zA-Z ]\*$/)) { formIsValid = false;errors["username"] = "\*Please

enter alphabet characters only.";

}

}

if (!fields["emailid"]) { formIsValid = false; errors["emailid"] = "\*Please enter your email-ID.";

}

if (!fields["mobileno"]) { formIsValid = false; errors["mobileno"] = "\*Please enter your mobile no.";

}

if (typeof fields["mobileno"] !== "undefined") { if (!fields["mobileno"].match(/^[0-9]{10}$/)) {

formIsValid = false;

errors["mobileno"] = "\*Please enter valid mobile no.";

}

}

if (!fields["password"]) { formIsValid = false; errors["password"] = "\*Please enter your password.";

}

if (typeof fields["password"] !== "undefined") {

if (!fields["password"].match("^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[0-9])(?=.\*[!@#\$%\^&\\*])(?=.{8,})") {

formIsValid = false;

errors["password"] = "\*Please enter secure and strong password.";

}

}

this.setState({ errors: errors

});

return formIsValid;

}

render() { return (

<div id="main-registration-container">

<div id="register">

<h3>Registration page</h3>

<form method="post" name="userRegistrationForm" onSubmit=

{this.submituserRegistrationForm} >

<label>Name</label>

<input type="text" name="username" value={this.state.fields.username}

onChange={this.handleChange} />

<div className="errorMsg">{this.state.errors.username}</div>

<label>Mobile No:</label>

<input type="text" name="mobileno" value={this.state.fields.mobileno}

onChange={this.handleChange} />

<div className="errorMsg">{this.state.errors.mobileno}</div>

<label>Password</label>

<input type="password" name="password" value={this.state.fields.password}

onChange={this.handleChange} />

<div className="errorMsg">{this.state.errors.password}</div>

<input type="submit" className="button" value="Register"/>

</form>

</div>

</div>

);

}

}

export default RegisterForm;

App.js

import './App.css';

import RegisterForm from '.form'; function App() {return (

<div className="App">

<RegisterForm/>

</div>

);

}

export default App;

CSS

#register, #login {width: 300px;

border: 1px solid #d6d7da; padding: 0px 15px 15px 15px; border-radius: 5px;font-family: arial; line-

height: 16px; color: #333333; font-size: 14px; background: #ffffff; margin: 100px auto;

}

form label, form input { display: block;margin-bottom: 10px; width: 90%

}

form input { padding: 10px; border: solid 1px #BDC7D8;

}

.button {

background-color: #00BFFF; border-color: #3ac162; font-weight: bold; padding: 12px 15px;color:

#ffffff;

}

.errorMsg { color: #cc0000;margin-bottom: 12px;

}

.sucessMsg { color: #6B8E23;margin-bottom: 10px;

10.

Coding:

// Import necessary React and Material-UI components

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

import { Table, TableBody, TableCell, TableContainer, TableHead, TableRow, Paper } from

'@mui/material';

// Functional component

function UserList() {

// State to store the fetched user data

const [users, setUsers] = useState([]);

// useEffect hook to fetch data when the component mounts

useEffect(() => {

const fetchData = async () => {

try {

const response = await fetch('https://jsonplaceholder.typicode.com/users');

const data = await response.json();

setUsers(data);

} catch (error) {

console.error('Error fetching user data:', error);

}

};

fetchData();

}, []); // Empty dependency array ensures useEffect runs only once when the component

mounts

// Render the component

return (

<TableContainer component={Paper}>

<Table>

<TableHead>

<TableRow>

<TableCell>User ID</TableCell>

<TableCell>Name</TableCell>

<TableCell>Email</TableCell>

<TableCell>Phone</TableCell>

</TableRow>

</TableHead>

<TableBody>

{users.map((user) => (

<TableRow key={user.id}>

<TableCell>{user.id}</TableCell>

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

<TableCell>{user.email}</TableCell>

<TableCell>{user.phone}</TableCell>

</TableRow>

))}

</TableBody>

</Table>

</TableContainer>

);

}

export default UserList;

11.

// greeting.js component:

import React, { useState } from "react";

function Appp() {

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

const fetchData = async () => {

try {

const response = await fetch(

https://jsonplaceholder.typicode.com/songs

);

if (!response.ok) {

throw new Error("Invalid Endpoint");

}

} catch (error) {

setError(

<h1 style={{ display: "flex", justifyContent: "center" }}>

Invalid endpoint or error occurred.

</h1>

);

console.error(error);

}

};

return (

<div>

<button

onClick={fetchData}

style={{

padding: "10px",

margin: "10px",

color: "white",

display: "flex",

justifyContent: "center",

backgroundColor: "blue",

}}

>

Fetch Data

</button>

{error && <h1>Error: {error}</h1 >}

</div>

);

}

export default Appp;

//App.Js component:

import './App.css';

import Message from './components/greeting';

function App() {

return (

<>

<App />

</>

);

}

export default App;

12.

// src/navbar.js

import React from 'react'

import { useFont } from './FontContext'

import AppBar from '@mui/material/AppBar';

import Toolbar from '@mui/material/Toolbar';

import Typography from '@mui/material/Typography';

import Button from '@mui/material/Button';

const Navbar = () => {

const { font, toggleFont } = useFont()

return (

<AppBar position="static">

<Toolbar>

<Typography

variant="h6"

sx={{ fontFamily: font}}

>

App

</Typography>

<Button color="inherit" onClick={toggleFont} sx={{ fontFamily: font}}>

Toggle Font

</Button>

</Toolbar>

</AppBar>

)

}

export default Navbar

// src/home.js

import React from 'react'

import { useFont } from './FontContext'

const Home = () => {

const { font } = useFont()

return (

<div style={{ fontFamily: font,textAlign: 'center' }}>

<h2>Home Component</h2>

<p>This is the home content with dynamic font.</p>

</div>

)

}

export default Home

// src/footer.js

import React from 'react'

import { useFont } from './FontContext'

import { Typography } from '@mui/material'

const Footer = () => {

const { font } = useFont()

return (

<footer style={{backgroundColor: 'purple',color:'white'}}>

<Typography variant="body1" style={{ fontFamily: font }}>

&copy; 2023 My React App. All rights reserved.

</Typography>

</footer>

)

}

export default Footer

//App.js

import React from 'react';

import Navbar from './components/navbar';

import Home from './components/home';

import Footer from './components/footer';

import { FontProvider } from './components/FontContext';

import './App.css';

function App() {

return (

<FontProvider>

<div className="App">

<Navbar />

<Home />

<Footer />

</div>

</FontProvider>

);

}

export default App;

13.

import React, { useRef } from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import {

TextField,

Button,

Container,

Typography,

CssBaseline,

Paper,

} from "@mui/material";

import { createTheme, ThemeProvider } from "@mui/material/styles";

// MUI theme

const theme = createTheme();

// Functional component definition

const InputForm = () => {

// useRef to manage input references

const inputRef1 = useRef();

const inputRef2 = useRef();

// Function to handle form submission

const handleSubmit = (event) => {

event.preventDefault();

// Log values to the console

console.log({

input1: inputRef1.current.value,

input2: inputRef2.current.value,

});

};

// JSX for the form

return (

<form onSubmit={handleSubmit}>

{/\* Input 1 \*/}

<TextField

variant="outlined"

margin="normal"

fullWidth

label="Input 1"

inputRef={inputRef1}

/>

{/\* Input 2 \*/}

<TextField

variant="outlined"

margin="normal"

fullWidth

label="Input 2"

inputRef={inputRef2}

/>

{/\* Submit button \*/}

<Button type="submit" fullWidth variant="contained" color="primary">

Submit

</Button>

</form>

);

};

// Parent component definition

const App = () => {

return (

<ThemeProvider theme={theme}>

<CssBaseline />

<Container

sx={{ display: "grid", placeItems: "center", minHeight: "100vh" }}

maxWidth="xs"

>

<Paper

sx={{ px: 4, py: 5, my: "auto" }}

elevation={4}

component={Paper}

>

<Typography align="center" component="h1" variant="h4">

Input Form Example

</Typography>

<InputForm />

</Paper>

</Container>

</ThemeProvider>

);

};

const root = document.getElementById("root");

ReactDOM.createRoot(root).render(<App />);

14.

LifecycleComponent.js

import React, { useEffect } from 'react';

const LifecycleComponent = () => {

// Component Mount Phase

useEffect(() => {

console.log('Component is mounted');

// Cleanup function (Unmount Phase)

return () => {

console.log('Component will unmount');

};

}, []); // Empty dependency array means this effect runs only once on mount

// Component Update Phase (every render)

useEffect(() => {

console.log('Component is updated');

});

return <div>Component Lifecycle Example</div>;

};

export default LifecycleComponent;

App.js

import React from 'react';

import LifecycleComponent from "./components/LifecycleComponent";

const App = () => {

return (

<div>

<h1>React Lifecycle Example</h1>

<LifecycleComponent />

</div>

);

};

export default App;

15.

ControlledForm.js:

import React, { useState } from 'react';

const ControlledForm = () => {

const [formData, setFormData] = useState({

firstName: '',

lastName: '',

email: '',

message: '',

});

const formContainer = {

maxWidth: "400px",

border: "1px solid #ccc",

}

const form = {

display: "flex",

flexDirection: "column"

}

const label = { marginBottom: "8px; }

const textField = {

padding: "8px",

border: "1px solid #ccc",

}

const buttonStyle = {

backgroundColor: "#4caf50",

color: "white",

cursor: "pointer"

}

const handleInputChange = (e) => {

const { name, value } = e.target;

setFormData({

...formData,

[name]: value,

});

};

const handleSubmit = (e) => {

e.preventDefault();

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

};

return (

<div className="form-container" style={formContainer}>

<h2>Contact Us</h2>

<form onSubmit={handleSubmit} style={form}>

<label style={label}>First Name</label>

<input type="text" name="firstName" value={formData.firstName}

onChange={handleInputChange} style={textField}/>

<br />

<label style={label}>Last Name</label>

<input type="text" name="lastName" value={formData.lastName}

onChange={handleInputChange} style={textField}/>

<br />

<label style={label}>Email</label>

<input style={textField} type="email" name="email" value={formData.email}

onChange={handleInputChange}/>

<br />

<label style={label}>Message</label>

<textarea name="message" value={formData.message} onChange={handleInputChange}

style={textField}></textarea>

<br />

<button type="submit" style={buttonStyle}>Submit</button>

</form>

</div>

);

};

export default ControlledForm;

16.

// App.js

import React from 'react';

import { Provider } from 'react-redux';

import './App.css';

import Counter from './components/redux/counter'

import store from './components/redux/store';

import Fetching from './components/fetchingjson/fetching';

function App() {

return (

<Provider store={store}>

<Counter/>

</Provider>

);

}

export default App;

// action.js

function increment() {

return {

type: 'inc'

};

}

function decrement() {

return {

type: 'dec'

};

}

export { increment, decrement };

// counter.js

import { connect } from "react-redux";

import { increment, decrement } from "./action";

function Counter({ count, increment, decrement }) {

return (

<div>

<h1>count: {count}</h1>

<button onClick={increment}>increment</button>

<button onClick={decrement}>decrement</button>

</div>

);

}

function Mapto(state) {

return {

count: state.count

};

}

const dis = { increment, decrement };

export default connect(Mapto, dis)(Counter);

// Reducer.js

function Reducer(state = { count: 0 }, action) {

switch (action.type) {

case 'inc':

return { count: state.count + 1 };

case 'dec':

return { count: state.count - 1 };

default:

return state;

}

}

export default Reducer;

// store.js

import { createStore } from "redux";

import Reducer from "./reducer";

const Store = createStore(Reducer);

export default Store;

17.

Form.jsx

import React, { useState } from "react";

import "./RegistrationForm.css";

function RegistrationForm() {

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

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

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

const handleSubmit = (e) => {

e.preventDefault();

alert(`Name: ${name}\nEmail: ${email}\nPassword: ${password}`);

};

return (

<div className="registration-form">

<h1>Registration Form</h1>

<form onSubmit={handleSubmit}>

<div className="form-group">

<label htmlFor="name">Name:</label>

<input

type="text"

id="name"

value={name}

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

required

/>

</div>

<div className="form-group">

<label htmlFor="email">Email:</label>

<input

type="email"

id="email"

value={email}

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

required

/>

</div>

<div className="form-group">

<label htmlFor="password">Password:</label>

<input

type="password"

id="password"

value={password}

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

required

/>

</div>

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

</form>

</div>

);

}

export default RegistrationForm;

style.css

.registration-form{

margin: 200px auto;

width: 400px;

}

.registration-form h1 {

margin-bottom: 20px;

font-family: Arial, sans-serif;

font-size: 24px;

font-weight: bold;

text-align: center;

}

.registration-form form {

padding: 20px;

border: 1px solid #ccc;}

.registration-form label {

display: block;

margin-bottom: 5px;

}

.registration-form input {

display: block;

width: 100%;

margin-bottom: 10px;

padding: 10px;

border: 1px solid #ccc;

}

.registration-form button {

margin-top: 10px;

padding: 10px;

border: none;

background-color: #00a2ed;

color: white;

cursor: pointer;

}

.registration-form button:hover {

background-color: #008ac9;

}

18.

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

const ErrorBoundary = ({ children }) => {

const [hasError, setHasError] = useState(false);

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

const [errorInfo, setErrorInfo] = useState(null);

useEffect(() => {

const handleError = (error, errorInfo) => {

setHasError(true);

setError(error);

setErrorInfo(errorInfo);

};

if (hasError) {

setHasError(false);

setError(null);

setErrorInfo(null);

}

window.addEventListener('error', handleError);

return () => {

window.removeEventListener('error', handleError);

};

}, [hasError]);

if (hasError) {

return (

<div>

<h2>Something went wrong!</h2>

<p>{error.toString()}</p>

<div>{errorInfo.componentStack}</div>

</div>

);

}

return <>{children}</>;

};

const ExampleComponent = () => {

const throwError = () => {

throw new Error('This is a deliberate error for demonstration purposes');

};

return (

<div>

<h1>Example Component</h1>

<button onClick={throwError}>Click me to throw an error</button>

</div>

);

};

const App = () => (

<ErrorBoundary>

<ExampleComponent />

</ErrorBoundary>

);

export default App;

19.

import React, { useState } from 'react';

const Counter = () => {

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

const increment = () => {

setCount(count + 1);

};

const decrement = () => {

setCount(count - 1);

};

const reset = () => {

setCount(0);

};

return (

<div>

<p>Count: {count}</p>

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

<button onClick={reset}>Reset</button>

</div>

);

};

export default Counter;

App.js:

import Counter from "./Project/demo1";

function App() {

return (

<div>

<h1>Counter App</h1>

<Counter/>

</div>

);

}

export default App;

20.

//Theme.js:

import React, { createContext, useContext, useState } from 'react';

import { ThemeProvider, createMuiTheme } from '@mui/material/styles';

import Button from '@mui/material/Button';

import { CssBaseline } from '@mui/material';

const ThemeContext = createContext();

const lightTheme = createMuiTheme({

palette: {

mode: 'light',

},

});

const darkTheme = createMuiTheme({

palette: {

mode: 'dark',

},

});

export const ThemeApp = ({ children }) => {

const [theme, setTheme] = useState(lightTheme);

const toggleTheme = () => {

setTheme(theme === lightTheme ? darkTheme : lightTheme);

};

return (

<ThemeContext.Provider value={{ theme, toggleTheme }}>

<ThemeProvider theme={theme}>

<CssBaseline />

{children}

</ThemeProvider>

</ThemeContext.Provider>

);

};

// Component that uses the theme context

const ThemeToggle = () => {

const { theme, toggleTheme } = useContext(ThemeContext);

return (

<div style={{textAlign:'center',position:'absolute',top:'40%',left:'42%'}}>

<Button variant="contained" color="primary" onClick={toggleTheme}>

{theme === lightTheme ? 'Switch to Dark Theme' : 'Switch to Light Theme'}

</Button>

</div>

);

};

export default ThemeToggle;

//App.js:

import ThemeToggle, { ThemeApp } from "./components/lab";

const App = () => {

return (

<ThemeApp>

<ThemeToggle />

</ThemeApp>

);

};

export default App;

21.

//bookcontexts.

import React, { createContext, useState, useContext } from "react";

const BookContext = createContext();

export const BookProvider = ({ children }) => {

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

const value = {

searchQuery,

setSearchQuery,

};

return <BookContext.Provider value={value}>{children}</BookContext.Provider>;

};

export const useBookContext = () => {

return useContext(BookContext);

};

//BookSerach.js

import React, { useEffect } from "react";

import axios from "axios";

import { useBookContext } from "./bookContexts";

const BookSearch = () => {

const { books, setBooks, searchTerm, setSearchTerm } = useBookContext();

useEffect(() => {

const fetchData = async () => {

try {

const response = await axios.get(

`https://jsonplaceholder.typicode.com/posts`

);

setBooks(response.data);

} catch (error) {

console.error("Error fetching data:" + error);

}

};

if (searchTerm) {

fetchData();

} else {

setBooks([]);

}

}, [searchTerm, setBooks]);

return (

<div>

<input

type="text"

placeholder="Search books"

value={searchTerm}

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

/>

<ul>

{books.map((book) => (

<li key={book.id}>{book.title}</li>

))}

</ul>

</div>

);

};

export default BookSearch;

//BookList

import React, { useEffect } from "react";

import axios from "axios";

import {

List,

ListItem,

ListItemText,

TextField,

Container,

Typography,

Box,

} from "@mui/material";

import { useBookContext } from "./bookContexts";

const BookList = () => {

const { searchQuery, setSearchQuery } = useBookContext();

const [books, setBooks] = React.useState([]);

useEffect(() => {

const fetchData = async () => {

try {

const response = await axios.get(

`https://jsonplaceholder.typicode.com/posts`

);

setBooks(response.data);

} catch (error) {

console.error("Error fetching data:" + error);

}

};

fetchData();

}, []);

const filteredBooks = books.filter((book) =>

book.title.toLowerCase().includes(searchQuery.toLowerCase())

);

return (

<Container>

<Box sx={{ my: 4 }}>

<Typography

variant="h4"

component="h1"

gutterBottom

sx={{ color: "red", display: "flex", justifyContent: "center" }}

>

Book Search

</Typography>

<TextField

label="Search Books"

value={searchQuery}

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

variant="outlined"

fullWidth

/>

<List>

{filteredBooks.map((book) => (

<ListItem key={book.id}>

<ListItemText primary={book.title} />

</ListItem>

))}

</List>

</Box>

</Container>

);

};

export default BookList;

22.

import { useState, useMemo } from "react";

import ReactDOM from "react-dom/client";

const App = () => {

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

const [todos, setTodos] = useState([]);

const calculation = useMemo(() => expensiveCalculation(count), [count]);

const increment = () => {

setCount((c) => c + 1);

};

const addTodo = () => {

setTodos((t) => [...t, "New Todo"]);

};

return (

<div>

<div>

<h2>My Todos</h2>

{todos.map((todo, index) => {

return <p key={index}>{todo}</p>;

})}

<button onClick={addTodo}>Add Todo</button>

</div>

<hr />

<div>

Count: {count}

<button onClick={increment}>+</button>

<h2>Expensive Calculation</h2>

{calculation}

</div>

</div>

);

};

const expensiveCalculation = (num) => {

console.log("Calculating...");

for (let i = 0; i < 1000000000; i++) {

num += 1;

}

return num;

};

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);