|  |
| --- |
|  |
|  | **Questions** |
|  |  |
|  | **1)Calculator** |
|  | **Design a simple Calculator example. Introduce the Stateful Component,** |
|  | **which are defined using a class and make it easier to keep track of the** |
|  | **application state provided component. In this component example use** |
|  | **add onClick event in JSX. Add the multiply and divide functionalies.** |
|  |  |
|  | **2)Voting System** |
|  | **Design a simple vote calculator example which increments count on every** |
|  | **click. Introduce the Stateful Component, which are defined using a class** |
|  | **and make it easier to keep track of the application state provided** |
|  | **component. In this component example use add onClick event in JSX.** |
|  | **3)Compass Clock** |
|  | **Design a clock to show current date time in compass clock format. In order** |
|  | **to perform this task reasonably well, need to modify the App.js file and also** |
|  | **the App.css file. Show the system time on the compass clock.** |
|  | **4)Form Validation** |
|  | **Design a form to show various different types of required validations for** |
|  | **Form. It should include use of email and phone validation based on regex.** |
|  | **5)Tic Tac Toe game** |
|  | **Develop a tic tac toe game** |
|  | **6)BMI Calculator** |
|  | **A React Hooks app for calculating BMI** |
|  | **7)Counter App** |
|  | **small shopping cart example** |
|  | **8)temperature control app** |
|  | **You will build a temperature control app where users can** |
|  | **increment/decrement temperature and the background color changes** |
|  | **based on temperature.** |
|  | **9)Search Filter** |
|  | **You will build an app that allows users to enter search keys and it filters the** |
|  | **user list from the hardcode list based on the entered key.** |
|  | **10)Basic Registration Form** |
|  | **You will build a basic registration form in React where users can enter first** |
|  | **name, last name, and email. Post-registration, an error message is** |
|  | **displayed for each field if validation fails else a success message is shown.** |
|  | **11)React Quiz App** |
|  | **You will build a simple React quiz app where users can click on option** |
|  | **buttons to answer questions, next question rendered each time option is** |
|  | **selected and at the end, the final score is displayed.** |
|  | **12)React Navbar** |
|  | **You will create a fully responsive navbar on React using CSS grid and** |
|  | **flexbox. Additionally, display breadcrumbs with the menu dropdown on** |
|  | **mobile screen size.** |
|  | **13)Photo gallery app** |
|  | **You will build a photo gallery app where users can upload a new file that is** |
|  | **stored in the Firebase Realtime database and view a list of uploaded files** |
|  | **in a grid format. Users can click on any image to view it as a popup along** |
|  | **with animations added to all the transitions.**  **ANSWERS:**   |  | | --- | | **1)Calculator App.js** | |  | import React, { Component } from "react"; | |  | import "./App.css"; | |  |  | |  | class Calculator extends Component { | |  | constructor(props) { | |  | super(props); | |  | this.state = { | |  | value: 0, | |  | displayValue: "0", | |  | waitingForOperand: false, | |  | operator: null | |  | }; | |  | } | |  |  | |  | inputDigit(digit) { | |  | const { displayValue, waitingForOperand } = this.state; | |  |  | |  | if (waitingForOperand) { | |  | this.setState({ | |  | displayValue: String(digit), | |  | waitingForOperand: false | |  | }); | |  | } else { | |  | this.setState({ | |  | displayValue: | |  | displayValue === "0" ? String(digit) : displayValue + digit | |  | }); | |  | } | |  | } | |  |  | |  | inputDot() { | |  | const { displayValue, waitingForOperand } = this.state; | |  |  | |  | if (waitingForOperand) { | |  | this.setState({ | |  | displayValue: ".", | |  | waitingForOperand: false | |  | }); | |  | } else if (displayValue.indexOf(".") === -1) { | |  | this.setState({ | |  | displayValue: displayValue + ".", | |  | waitingForOperand: false | |  | }); | |  | } | |  | } | |  |  | |  | clearDisplay() { | |  | this.setState({ | |  | displayValue: "0" | |  | }); | |  | } | |  |  | |  | toggleSign() { | |  | const { displayValue } = this.state; | |  | this.setState({ | |  | displayValue: | |  | displayValue.charAt(0) === "-" | |  | ? displayValue.substr(1) | |  | : "-" + displayValue | |  | }); | |  | } | |  |  | |  | inputPercent() { | |  | const { displayValue } = this.state; | |  | const value = parseFloat(displayValue); | |  |  | |  | this.setState({ | |  | displayValue: String(value / 100) | |  | }); | |  | } | |  |  | |  | performOperation(nextOperator) { | |  | const { value, displayValue, operator } = this.state; | |  | const nextValue = parseFloat(displayValue); | |  |  | |  | const operations = { | |  | "/": (prevValue, nextValue) => prevValue / nextValue, | |  | "\*": (prevValue, nextValue) => prevValue \* nextValue, | |  | "+": (prevValue, nextValue) => prevValue + nextValue, | |  | "-": (prevValue, nextValue) => prevValue - nextValue, | |  | "=": (prevValue, nextValue) => nextValue | |  | }; | |  |  | |  | if (value == null) { | |  | this.setState({ | |  | value: nextValue | |  | }); | |  | } else if (operator) { | |  | const currentValue = value || 0; | |  | const computedValue = operations[operator](currentValue, nextValue); | |  |  | |  | this.setState({ | |  | value: computedValue, | |  | displayValue: String(computedValue) | |  | }); | |  | } | |  |  | |  | this.setState({ | |  | waitingForOperand: true, | |  | operator: nextOperator | |  | }); | |  | } | |  |  | |  | render() { | |  | const { displayValue } = this.state; | |  |  | |  | return ( | |  | <div className="calculator"> | |  | <div className="calculator-display">{displayValue}</div> | |  | <div className="calculator-keypad"> | |  | <div className="input-keys"> | |  | <div className="function-keys"> | |  | <button className="calculator-key key-clear" onClick={() => this.clearDisplay()}> | |  | AC | |  | </button> | |  | <button className="calculator-key key-sign" onClick={() => this.toggleSign()}> | |  | ± | |  | </button> | |  | <button className="calculator-key key-percent" onClick={() => this.inputPercent()}> | |  | % | |  | </button> | |  | </div> | |  | <div className="digit-keys"> | |  | <button className="calculator-key key-0" onClick={() => this.inputDigit(0)}> | |  | 0 | |  | </button> | |  | <button className="calculator-key key-dot" onClick={() => this.inputDot()}> | |  | ● | |  | </button> | |  | <button className="calculator-key key-1" onClick={() => this.inputDigit(1)}> | |  | 1 | |  | </button> | |  | <button className="calculator-key key-2" onClick={() => this.inputDigit(2)}> | |  | 2 | |  | </button> | |  | <button className="calculator-key key-3" onClick={() => this.inputDigit(3)}> | |  | 3 | |  | </button> | |  | <button className="calculator-key key-4" onClick={() => this.inputDigit(4)}> | |  | 4 | |  | </button> | |  | <button className="calculator-key key-5" onClick={() => this.inputDigit(5)}> | |  | 5 | |  | </button> | |  | <button className="calculator-key key-6" onClick={() => this.inputDigit(6)}> | |  | 6 | |  | </button> | |  | <button className="calculator-key key-7" onClick={() => this.inputDigit(7)}> | |  | 7 | |  | </button> | |  | <button className="calculator-key key-8" onClick={() => this.inputDigit(8)}> | |  | 8 | |  | </button> | |  | <button className="calculator-key key-9" onClick={() => this.inputDigit(9)}> | |  | 9 | |  | </button> | |  | </div> | |  | </div> | |  | <div className="operator-keys"> | |  | <button className="calculator-key key-divide" onClick={() => this.performOperation("/")}> | |  | ÷ | |  | </button> | |  | <button className="calculator-key key-multiply" onClick={() => this.performOperation("\*")}> | |  | × | |  | </button> | |  | <button className="calculator-key key-subtract" onClick={() => this.performOperation("-")}> | |  | − | |  | </button> | |  | <button className="calculator-key key-add" onClick={() => this.performOperation("+")}> | |  | + | |  | </button> | |  | <button className="calculator-key key-equals" onClick={() => this.performOperation("=")}> | |  | = | |  | </button> | |  | </div> | |  | </div> | |  | </div> | |  | ); | |  | } | |  | } | |  |  | |  | export default Calculator; | |  |  | |  | **// App.css** | |  | .calculator { | |  | font-family: sans-serif; | |  | width: 320px; | |  | background: #fcfcfc; | |  | padding: 20px; | |  | border-radius: 10px; | |  | box-shadow: 0px 5px 10px 0px rgba(0, 0, 0, 0.2); | |  | margin: auto; | |  | } | |  | .calculator-display { | |  | width: 100%; | |  | margin-bottom: 20px; | |  | font-size: 2.5em; | |  | text-align: right; | |  | color: #222; | |  | background: #e0e0e0; | |  | padding: 12px 20px; | |  | border-radius: 3px; | |  | box-shadow: 0px 5px 10px 0px rgba(0, 0, 0, 0.2); | |  | } | |  | .calculator-key { | |  | float: left; | |  | width: 25%; | |  | padding: 10px 0; | |  | cursor: pointer; | |  | font-size: 1.5em; | |  | background: #f0f0f0; | |  | border: none; | |  | outline: none; | |  | border-radius: 3px; | |  | box-shadow: 0px 2px 5px 0px rgba(0, 0, 0, 0.2); | |  | margin-bottom: 10px; | |  | } | |  | .calculator-key.key-clear, | |  | .calculator-key.key-sign { | |  | background: #fa9595; | |  | color: #fff; | |  | } | |  | .calculator-key.key-percent { | |  | background: #a7bfe8; | |  | color: #fff; | |  | } | |  | .calculator-key.operator { | |  | background: #ffcd75; | |  | color: #fff; | |  | } | |  | .calculator-key.key-equals { | |  | background: #55efc4; | |  | color: #fff; | |  | } |  |  | | --- | |  | |  |  | |  | **2) Voting System**  **App.js** | |  |  | |  | import React, { Component } from 'react'; | |  | import './App.css'; | |  |  | |  | class VoteCalculator extends Component { | |  | constructor(props) { | |  | super(props); | |  | this.state = { | |  | count: 0 | |  | }; | |  | } | |  |  | |  | incrementCounter() { | |  | this.setState({ | |  | count: this.state.count + 1 | |  | }); | |  | } | |  |  | |  | render() { | |  | return ( | |  | <div className="App"> | |  | <header className="App-header"> | |  | <h1>Vote Calculator</h1> | |  | </header> | |  | <p>You have voted {this.state.count} times.</p> | |  | <button onClick={() => this.incrementCounter()}> | |  | Vote Now | |  | </button> | |  | </div> | |  | ); | |  | } | |  | } | |  |  | |  | export default VoteCalculator; | |  |  | |  | **App.css** | |  |  | |  | .App { | |  | text-align: center; | |  | } | |  |  | |  | .App-header { | |  | background-color: #282c34; | |  | min-height: 100vh; | |  | display: flex; | |  | flex-direction: column; | |  | align-items: center; | |  | justify-content: center; | |  | font-size: calc(10px + 2vmin); | |  | color: white; | |  | } | |  |  | |  | button { | |  | background-color: #282c34; | |  | font-size: 1rem; | |  | padding: 0.5rem 1.5rem; | |  | border: none; | |  | color: #fff; | |  | width: 100px; | |  | margin-top: 2rem; | |  | cursor: pointer; | |  | transition: all 0.3s ease; | |  | &:hover { | |  | background-color: #1f212b; | |  | } | |  | } |  |  | | --- | |  | |  |  | |  | **3) Compass Clock**  **// App.js** | |  | import React, { Component } from 'react'; | |  | import './App.css'; | |  |  | |  | class App extends Component { | |  | constructor(props) { | |  | super(props); | |  | this.state = { | |  | date: new Date() | |  | }; | |  | } | |  | componentDidMount() { | |  | this.timerID = setInterval( | |  | () => this.tick(), | |  | 1000 | |  | ); | |  | } | |  |  | |  | componentWillUnmount() { | |  | clearInterval(this.timerID); | |  | } | |  |  | |  | tick() { | |  | this.setState({ | |  | date: new Date() | |  | }); | |  | } | |  |  | |  | render() { | |  | return ( | |  | <div className="App"> | |  | <h1>Compass Clock</h1> | |  | <div className="clock"> | |  | <div className="center-circle"></div> | |  | <div className="second-hand hand" | |  | style={{transform: `rotate(${this.state.date.getSeconds() \* 6}deg)`}}></div> | |  | <div className="minute-hand hand" | |  | style={{transform: `rotate(${this.state.date.getMinutes() \* 6}deg)`}}></div> | |  | <div className="hour-hand hand" | |  | style={{transform: `rotate(${this.state.date.getHours() \* 30 + this.state.date.getMinutes() \* 0.5}deg)`}}></div> | |  | </div> | |  | </div> | |  | ); | |  | } | |  | } | |  |  | |  | export default App; | |  |  | |  | **// App.css** | |  | .App { | |  | text-align: center; | |  | font-family: sans-serif; | |  | background-color: #e6e6e6; | |  | } | |  |  | |  | .clock { | |  | position: relative; | |  | width: 500px; | |  | height: 500px; | |  | margin: auto; | |  | border-radius: 50%; | |  | border: 5px solid #000000; | |  | background-color: #ffffff; | |  | } | |  |  | |  | .center-circle { | |  | position: absolute; | |  | width: 10px; | |  | height: 10px; | |  | border-radius: 50%; | |  | background-color: #000000; | |  | top: 50%; | |  | left: 50%; | |  | transform: translate(-50%, -50%); | |  | } | |  |  | |  | .hand { | |  | position: absolute; | |  | width: 50%; | |  | height: 5px; | |  | background-color: #000000; | |  | top: 50%; | |  | left: 50%; | |  | transform-origin: 100%; | |  | transition: all 0.05s ease-in-out; | |  | } | |  |  | |  | .second-hand { | |  | transform: rotate(90deg); | |  | transform-origin: 0% 100%; | |  | } | |  |  | |  | .minute-hand { | |  | transform: rotate(90deg); | |  | transform-origin: 0% 100%; | |  | } | |  |  | |  | .hour-hand { | |  | transform: rotate(90deg); | |  | transform-origin: 0% 100%; | |  | } |  |  | | --- | |  | |  | |  |  | |  | **4) Form Validation**  **App.js** | |  |  | |  | import React, { Component } from 'react'; | |  | import './App.css'; | |  |  | |  | //Variables to store data | |  | var emailRegex = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/; | |  | var phoneRegex = /^\(\d{3}\) \d{3}-\d{4}$/; | |  |  | |  | class App extends Component { | |  | state = { | |  | formData: { | |  | name: '', | |  | email: '', | |  | phone: '', | |  | date: '' | |  | }, | |  | errors: { | |  | name: '', | |  | email: '', | |  | phone: '', | |  | date: '' | |  | } | |  | }; | |  |  | |  | //Function to validate form | |  | validateForm = () => { | |  | let errors = {}; | |  | let formIsValid = true; | |  |  | |  | //Validate name | |  | if (!this.state.formData.name) { | |  | formIsValid = false; | |  | errors['name'] = 'Name field cannot be empty'; | |  | } | |  |  | |  | //Validate email | |  | if (!this.state.formData.email) { | |  | formIsValid = false; | |  | errors['email'] = 'Email field cannot be empty'; | |  | } | |  | if (typeof this.state.formData.email !== 'undefined') { | |  | if (!emailRegex.test(this.state.formData.email)) { | |  | formIsValid = false; | |  | errors['email'] = 'Please enter a valid email address'; | |  | } | |  | } | |  |  | |  | //Validate phone | |  | if (!this.state.formData.phone) { | |  | formIsValid = false; | |  | errors['phone'] = 'Phone field cannot be empty'; | |  | } | |  | if (typeof this.state.formData.phone !== 'undefined') { | |  | if (!phoneRegex.test(this.state.formData.phone)) { | |  | formIsValid = false; | |  | errors['phone'] = 'Please enter a valid phone number'; | |  | } | |  | } | |  |  | |  | //Validate date | |  | if (!this.state.formData.date) { | |  | formIsValid = false; | |  | errors['date'] = 'Date field cannot be empty'; | |  | } | |  |  | |  | //Update errors in state | |  | this.setState({ | |  | errors: errors | |  | }); | |  | return formIsValid; | |  | }; | |  |  | |  | //Function to handle input change | |  | handleInputChange = e => { | |  | const { name, value } = e.target; | |  | let formData = this.state.formData; | |  | formData[name] = value; | |  | this.setState({ | |  | formData: formData | |  | }); | |  | }; | |  |  | |  | //Function to handle form submission | |  | handleFormSubmit = e => { | |  | e.preventDefault(); | |  | if (this.validateForm()) { | |  | console.log('Form is valid. Submitted successfully!'); | |  | } else { | |  | console.log('Form is invalid'); | |  | } | |  | }; | |  |  | |  | //Render the form | |  | render() { | |  | return ( | |  | <div className='App'> | |  | <form> | |  | <label>Name</label> | |  | <input | |  | name='name' | |  | type='text' | |  | value={this.state.formData.name} | |  | onChange={this.handleInputChange} | |  | /> | |  | <span className='error'>{this.state.errors.name}</span> | |  | <label>Email</label> | |  | <input | |  | name='email' | |  | type='text' | |  | value={this.state.formData.email} | |  | onChange={this.handleInputChange} | |  | /> | |  | <span className='error'>{this.state.errors.email}</span> | |  | <label>Phone</label> | |  | <input | |  | name='phone' | |  | type='text' | |  | value={this.state.formData.phone} | |  | onChange={this.handleInputChange} | |  | /> | |  | <span className='error'>{this.state.errors.phone}</span> | |  | <label>Date</label> | |  | <input | |  | name='date' | |  | type='text' | |  | value={this.state.formData.date} | |  | onChange={this.handleInputChange} | |  | /> | |  | <span className='error'>{this.state.errors.date}</span> | |  | <button onClick={this.handleFormSubmit}>Submit</button> | |  | </form> | |  | </div> | |  | ); | |  | } | |  | } | |  |  | |  | export default App; | |  |  | |  | **App.css** | |  |  | |  | .App { | |  | width: 700px; | |  | margin: 0 auto; | |  | padding: 20px; | |  | } | |  |  | |  | label { | |  | display: block; | |  | margin-top: 10px; | |  | font-size: 20px; | |  | } | |  |  | |  | input { | |  | margin-top: 5px; | |  | border: 1px solid #ccc; | |  | border-radius: 3px; | |  | padding: 5px; | |  | font-size: 16px; | |  | } | |  |  | |  | button { | |  | background-color: #4CAF50; | |  | border: none; | |  | color: white; | |  | padding: 10px; | |  | text-align: center; | |  | text-decoration: none; | |  | display: inline-block; | |  | font-size: 16px; | |  | margin-top: 15px; | |  | margin-bottom: 15px; | |  | cursor: pointer; | |  | border-radius: 3px; | |  | } | |  |  | |  | .error { | |  | color: red; | |  | font-size: 14px; | |  | margin-top: 5px; | |  | } | |

|  |  |
| --- | --- |
|  | **5) Tic Tac Toe game** |
|  | **App.js** |
|  | import React from 'react'; |
|  | import { connect } from 'react-redux'; |
|  | import { moveSquare, gameWon, gameDraw } from '../actions'; |
|  |  |
|  | // Square component |
|  | const Square = (props) => { |
|  | const { value, moveSquare, rowIndex, colIndex, gameWon, gameDraw } = props; |
|  | const handleClick = () => { |
|  | moveSquare(rowIndex, colIndex); |
|  | gameWon(); |
|  | gameDraw(); |
|  | } |
|  | return ( |
|  | <div className="square" onClick={handleClick}> |
|  | {value} |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | // Board component |
|  | const Board = (props) => { |
|  | const { squares, moveSquare, gameWon, gameDraw } = props; |
|  | return ( |
|  | <div className="board"> |
|  | { |
|  | squares.map((row, rowIndex) => { |
|  | return ( |
|  | <div key={rowIndex} className="row"> |
|  | { |
|  | row.map((value, colIndex) => { |
|  | return ( |
|  | <Square |
|  | key={colIndex} |
|  | value={value} |
|  | moveSquare={moveSquare} |
|  | rowIndex={rowIndex} |
|  | colIndex={colIndex} |
|  | gameWon={gameWon} |
|  | gameDraw={gameDraw} |
|  | /> |
|  | ); |
|  | }) |
|  | } |
|  | </div> |
|  | ) |
|  | }) |
|  | } |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | // App component |
|  | const App = (props) => { |
|  | const { squares, moveSquare, gameWon, gameDraw } = props; |
|  |  |
|  | return ( |
|  | <div> |
|  | <h1>Tic Tac Toe</h1> |
|  | <Board |
|  | squares={squares} |
|  | moveSquare={moveSquare} |
|  | gameWon={gameWon} |
|  | gameDraw={gameDraw} |
|  | /> |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | const mapStateToProps = (state) => { |
|  | return { |
|  | squares: state.squares |
|  | }; |
|  | }; |
|  |  |
|  | const mapDispatchToProps = (dispatch) => { |
|  | return { |
|  | moveSquare: (row, col) => dispatch(moveSquare(row, col)), |
|  | gameWon: () => dispatch(gameWon()), |
|  | gameDraw: () => dispatch(gameDraw()) |
|  | }; |
|  | }; |
|  |  |
|  | export default connect(mapStateToProps, mapDispatchToProps)(App); |
|  |  |
|  | **App.css** |
|  | \* { |
|  | box-sizing: border-box; |
|  | } |
|  |  |
|  | body { |
|  | margin: 0; |
|  | padding: 0; |
|  | font-family: sans-serif; |
|  | background-color: #f1f1f1; |
|  | } |
|  |  |
|  | .container { |
|  | margin: 0 auto; |
|  | width: 600px; |
|  | background-color: #fff; |
|  | padding: 20px; |
|  | text-align: center; |
|  | } |
|  |  |
|  | .board { |
|  | margin-top: 20px; |
|  | position: relative; |
|  | width: 100%; |
|  | height: 0; |
|  | padding-top: 100%; |
|  | background-color: #ccc; |
|  | border-radius: 10px; |
|  | } |
|  |  |
|  | .board:after { |
|  | content: ''; |
|  | position: absolute; |
|  | top: 17.5%; |
|  | left: 17.5%; |
|  | width: 65%; |
|  | height: 65%; |
|  | background-color: #fff; |
|  | border-radius: 10px; |
|  | z-index: -1; |
|  | } |
|  |  |
|  | .board > div { |
|  | position: absolute; |
|  | width: 33.33%; |
|  | height: 33.33%; |
|  | background-color: #ccc; |
|  | border-radius: 10px; |
|  | display: flex; |
|  | justify-content: center; |
|  | align-items: center; |
|  | font-size: 2rem; |
|  | color: #fff; |
|  | font-weight: bold; |
|  | cursor: pointer; |
|  | transition: all .3s; |
|  | } |
|  |  |
|  | .board > div:hover { |
|  | background-color: #888; |
|  | } |
|  |  |
|  | .board > div.x { |
|  | background-color: #0099ff; |
|  | } |
|  |  |
|  | .board > div.o { |
|  | background-color: #ff9999; |
|  | } |
|  |  |
|  | .board > div.active { |
|  | background-color: #888; |
|  | } |
|  |  |
|  | .turn { |
|  | font-size: 1.2rem; |
|  | font-weight: bold; |
|  | color: #0099ff; |
|  | margin-top: 20px; |
|  | } |
|  |  |
|  | .game-end { |
|  | font-size: 1.2rem; |
|  | font-weight: bold; |
|  | color: #ff9999; |
|  | margin-top: 20px; |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **6) BMI Calculator** |
|  | **App.js** |
|  |  |
|  | import React, { useState } from "react"; |
|  |  |
|  | const App = () => { |
|  | const [height, setHeight] = useState(0); |
|  | const [weight, setWeight] = useState(0); |
|  | const [bmi, setBmi] = useState(0); |
|  |  |
|  | const calculateBMI = e => { |
|  | e.preventDefault(); |
|  | const bmiCalculation = (weight / (height \* height)) \* 10000; |
|  | setBmi(bmiCalculation); |
|  | }; |
|  |  |
|  | return ( |
|  | <div> |
|  | <form onSubmit={calculateBMI}> |
|  | <label> |
|  | Height (cm): |
|  | <input |
|  | type="number" |
|  | value={height} |
|  | onChange={e => setHeight(parseInt(e.target.value))} |
|  | /> |
|  | </label> |
|  | <label> |
|  | Weight (kg): |
|  | <input |
|  | type="number" |
|  | value={weight} |
|  | onChange={e => setWeight(parseInt(e.target.value))} |
|  | /> |
|  | </label> |
|  | <input type="submit" value="Calculate BMI" /> |
|  | </form> |
|  | <p>Your BMI is: {bmi}</p> |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | export default App; |
|  |  |
|  |  |
|  | **App.css** |
|  | .container { |
|  | width: 100%; |
|  | max-width: 500px; |
|  | margin: 0 auto; |
|  | padding: 10px; |
|  | text-align: center; |
|  | } |
|  |  |
|  | .title { |
|  | font-size: 20px; |
|  | font-weight: bold; |
|  | margin-bottom: 10px; |
|  | } |
|  |  |
|  | .input-group { |
|  | margin-top: 10px; |
|  | margin-bottom: 10px; |
|  | width: 100%; |
|  | display: flex; |
|  | flex-direction: row; |
|  | justify-content: center; |
|  | align-items: center; |
|  | } |
|  |  |
|  | .input-group > label { |
|  | padding-right: 10px; |
|  | width: 30%; |
|  | } |
|  |  |
|  | .input-group > input { |
|  | width: 70%; |
|  | padding: 5px 10px; |
|  | border-radius: 5px; |
|  | border: 1px solid #ccc; |
|  | } |
|  |  |
|  | .btn { |
|  | background-color: #4CAF50; |
|  | color: #fff; |
|  | padding: 10px; |
|  | text-align: center; |
|  | border-radius: 10px; |
|  | width: 100%; |
|  | margin-top: 10px; |
|  | margin-bottom: 10px; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | .result { |
|  | font-size: 18px; |
|  | font-weight: bold; |
|  | margin-top: 10px; |
|  | margin-bottom: 10px; |
|  | } |
|  |  |
|  |  |
|  | **7) Counter App** |
|  | **// app.js** |
|  |  |
|  | import React from 'react'; |
|  | import ReactDOM from 'react-dom'; |
|  | import { createStore } from 'redux'; |
|  | import { Provider } from 'react-redux'; |
|  | import App from './App'; |
|  | import reducer from './reducers'; |
|  |  |
|  | const store = createStore(reducer); |
|  |  |
|  | ReactDOM.render( |
|  | <Provider store={store}> |
|  | <App /> |
|  | </Provider>, |
|  | document.getElementById('root') |
|  | ); |
|  |  |
|  | **// app.css** |
|  |  |
|  | .app { |
|  | display: flex; |
|  | flex-direction: column; |
|  | width: 500px; |
|  | margin: 0 auto; |
|  | background-color: #f2f2f2; |
|  | font-family: sans-serif; |
|  | } |
|  |  |
|  | .app-header { |
|  | background-color: #24292e; |
|  | color: white; |
|  | padding: 10px; |
|  | font-size: 20px; |
|  | } |
|  |  |
|  | .app-body { |
|  | padding: 10px; |
|  | } |
|  |  |
|  | .app-item { |
|  | display: flex; |
|  | justify-content: space-between; |
|  | padding: 10px 0; |
|  | border-bottom: 1px solid #e2e2e2; |
|  | } |
|  |  |
|  | .app-item-name { |
|  | font-weight: bold; |
|  | } |
|  |  |
|  | .app-item-quantity { |
|  | font-weight: bold; |
|  | color: #24292e; |
|  | } |
|  |  |
|  | .app-footer { |
|  | padding: 10px; |
|  | display: flex; |
|  | flex-direction: row; |
|  | justify-content: space-between; |
|  | background-color: #24292e; |
|  | color: white; |
|  | } |
|  |  |
|  | .app-total-price { |
|  | font-weight: bold; |
|  | } |
|  |  |
|  |  |
|  | **8) temperature control app** |
|  | **/\* app.js \*/** |
|  |  |
|  | import React from 'react'; |
|  | import { connect } from 'react-redux'; |
|  | import { incrementTemp, decrementTemp } from './actions'; |
|  |  |
|  | const App = ({ temperature, incrementTemp, decrementTemp }) => { |
|  | const bgColor = temperature > 0 ? 'red' : 'blue'; |
|  | return ( |
|  | <div className="App" style={{ backgroundColor: bgColor }}> |
|  | <h1>Temperature: {temperature}</h1> |
|  | <button onClick={incrementTemp}>+</button> |
|  | <button onClick={decrementTemp}>-</button> |
|  | </div> |
|  | ); |
|  | } |
|  |  |
|  | const mapStateToProps = state => ({ |
|  | temperature: state.temperature |
|  | }); |
|  |  |
|  | const mapDispatchToProps = dispatch => ({ |
|  | incrementTemp: () => dispatch(incrementTemp()), |
|  | decrementTemp: () => dispatch(decrementTemp()) |
|  | }); |
|  |  |
|  | export default connect( |
|  | mapStateToProps, |
|  | mapDispatchToProps |
|  | )(App); |
|  |  |
|  | **/\* app.css \*/** |
|  |  |
|  | .App { |
|  | padding: 20px; |
|  | text-align: center; |
|  | font-family: sans-serif; |
|  | } |
|  |  |
|  | button { |
|  | padding: 10px; |
|  | margin-top: 10px; |
|  | font-size: 20px; |
|  | color: white; |
|  | background-color: black; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | **9) Search Filter** |
|  | **// App.js** |
|  |  |
|  | import React, { Component } from 'react'; |
|  | import { connect } from 'react-redux'; |
|  | import { updateSearch } from './actions'; |
|  |  |
|  | class App extends Component { |
|  | constructor(props) { |
|  | super(props); |
|  | this.state = { |
|  | searchValue: '' |
|  | }; |
|  | } |
|  |  |
|  | handleChange = (e) => { |
|  | this.setState({ |
|  | searchValue: e.target.value |
|  | }); |
|  | } |
|  |  |
|  | handleSubmit = (e) => { |
|  | e.preventDefault(); |
|  | this.props.updateSearch(this.state.searchValue); |
|  | } |
|  |  |
|  | render() { |
|  | const { users } = this.props; |
|  | const usersList = users.map((user) => { |
|  | return ( |
|  | <li key={user.id}> |
|  | <p>{user.name}</p> |
|  | </li> |
|  | ); |
|  | }); |
|  | return ( |
|  | <div> |
|  | <form onSubmit={this.handleSubmit}> |
|  | <input |
|  | type="text" |
|  | value={this.state.searchValue} |
|  | onChange={this.handleChange} |
|  | /> |
|  | <button type="submit">Search</button> |
|  | </form> |
|  | <ul> |
|  | {usersList} |
|  | </ul> |
|  | </div> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | const mapStateToProps = (state) => { |
|  | return { |
|  | users: state.users |
|  | }; |
|  | }; |
|  |  |
|  | const mapDispatchToProps = { |
|  | updateSearch |
|  | }; |
|  |  |
|  | export default connect(mapStateToProps, mapDispatchToProps)(App); |
|  |  |
|  |  |
|  |  |
|  | **// App.css** |
|  |  |
|  | form { |
|  | margin-bottom: 25px; |
|  | } |
|  |  |
|  | input { |
|  | width: 250px; |
|  | padding: 10px; |
|  | margin-right: 10px; |
|  | font-size: 16px; |
|  | border-radius: 4px; |
|  | border: 1px solid #ccc; |
|  | } |
|  |  |
|  | button { |
|  | background-color: #4CAF50; |
|  | color: white; |
|  | padding: 10px 20px; |
|  | font-size: 16px; |
|  | border-radius: 4px; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  | li { |
|  | padding: 10px; |
|  | list-style: none; |
|  | font-size: 16px; |
|  | border-bottom: 1px solid #ccc; |
|  | } |
|  |  |
|  |  |
|  | **10) Basic Registration Form** |
|  | **// app.js** |
|  |  |
|  | import React from 'react'; |
|  | import { connect } from 'react-redux'; |
|  | import { registerUser } from './actions/registrationAction'; |
|  |  |
|  | class RegistrationForm extends React.Component { |
|  | constructor(props) { |
|  | super(props); |
|  |  |
|  | this.state = { |
|  | firstName: '', |
|  | lastName: '', |
|  | email: '' |
|  | }; |
|  | } |
|  |  |
|  | handleInputChange = (event) => { |
|  | const { name, value } = event.target; |
|  | this.setState({ |
|  | [name]: value |
|  | }); |
|  | } |
|  |  |
|  | handleFormSubmit = (event) => { |
|  | event.preventDefault(); |
|  |  |
|  | const { firstName, lastName, email } = this.state; |
|  | this.props.registerUser({ firstName, lastName, email }); |
|  | } |
|  |  |
|  | render() { |
|  | const { isError, errorMessage } = this.props; |
|  |  |
|  | return ( |
|  | <form onSubmit={this.handleFormSubmit}> |
|  | <div className="form-group"> |
|  | <label>First Name</label> |
|  | <input |
|  | type="text" |
|  | name="firstName" |
|  | value={this.state.firstName} |
|  | onChange={this.handleInputChange} |
|  | /> |
|  | </div> |
|  | <div className="form-group"> |
|  | <label>Last Name</label> |
|  | <input |
|  | type="text" |
|  | name="lastName" |
|  | value={this.state.lastName} |
|  | onChange={this.handleInputChange} |
|  | /> |
|  | </div> |
|  | <div className="form-group"> |
|  | <label>Email</label> |
|  | <input |
|  | type="email" |
|  | name="email" |
|  | value={this.state.email} |
|  | onChange={this.handleInputChange} |
|  | /> |
|  | </div> |
|  | <button type="submit">Submit</button> |
|  | {isError && <p className="error-message">{errorMessage}</p>} |
|  | </form> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | const mapStateToProps = (state) => ({ |
|  | isError: state.registration.isError, |
|  | errorMessage: state.registration.errorMessage, |
|  | }); |
|  |  |
|  | const mapDispatchToProps = { registerUser }; |
|  |  |
|  | export default connect(mapStateToProps, mapDispatchToProps)(RegistrationForm); |
|  |  |
|  | **// app.css** |
|  |  |
|  | .form-group { |
|  | margin-bottom: 1rem; |
|  | } |
|  |  |
|  | .error-message { |
|  | color: red; |
|  | font-size: 0.9rem; |
|  | } |
|  |  |
|  |  |
|  | **11) React Quiz App** |
|  | **App.js:** |
|  |  |
|  | import React, { useState } from 'react'; |
|  | import { connect } from 'react-redux'; |
|  | import { setQuestions, setAnswers, setScore } from './actions'; |
|  |  |
|  | const App = (props) => { |
|  | const [questions, setQuestions] = useState([]); |
|  | const [answers, setAnswers] = useState([]); |
|  | const [score, setScore] = useState(0); |
|  |  |
|  | const handleSubmit = (e) => { |
|  | e.preventDefault(); |
|  | props.setQuestions(questions); |
|  | props.setAnswers(answers); |
|  | props.setScore(score); |
|  | } |
|  |  |
|  | const handleQuestion = (e) => { |
|  | setQuestions([...questions, e.target.value]); |
|  | } |
|  |  |
|  | const handleAnswer = (e) => { |
|  | setAnswers([...answers, e.target.value]); |
|  | } |
|  |  |
|  | const handleScore = (e) => { |
|  | setScore(score + 1); |
|  | } |
|  |  |
|  | return ( |
|  | <div> |
|  | <form onSubmit={handleSubmit}> |
|  | <h1>React Quiz App</h1> |
|  | <h2>Questions</h2> |
|  | <input type="text" onChange={handleQuestion} /> |
|  | <h2>Answers</h2> |
|  | <input type="radio" name="answer" value="1" onChange={handleAnswer} /> |
|  | <input type="radio" name="answer" value="2" onChange={handleAnswer} /> |
|  | <input type="radio" name="answer" value="3" onChange={handleAnswer} /> |
|  | <input type="submit" value="Submit" onClick={handleScore} /> |
|  | </form> |
|  | <h3>Your Score: {score}</h3> |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | const mapStateToProps = (state) => { |
|  | return { |
|  | questions: state.questions, |
|  | answers: state.answers, |
|  | score: state.score |
|  | } |
|  | } |
|  |  |
|  | const mapDispatchToProps = (dispatch) => { |
|  | return { |
|  | setQuestions: (questions) => { dispatch(setQuestions(questions)) }, |
|  | setAnswers: (answers) => { dispatch(setAnswers(answers)) }, |
|  | setScore: (score) => { dispatch(setScore(score)) } |
|  | } |
|  | } |
|  |  |
|  | export default connect(mapStateToProps, mapDispatchToProps)(App); |
|  |  |
|  | **App.css:** |
|  |  |
|  | .App { |
|  | font-family: sans-serif; |
|  | font-size: 1rem; |
|  | color: #222; |
|  | padding: 1rem; |
|  | max-width: 400px; |
|  | margin: 0 auto; |
|  | background: #F7F7F7; |
|  | border-radius: 5px; |
|  | box-shadow: 0 0 4px #ccc; |
|  | } |
|  |  |
|  | h1 { |
|  | font-size: 1.5rem; |
|  | margin: 0 0 0.5rem 0; |
|  | } |
|  |  |
|  | h2 { |
|  | font-size: 1.2rem; |
|  | margin: 0 0 0.5rem 0; |
|  | } |
|  |  |
|  | input[type="text"], |
|  | input[type="radio"] { |
|  | font-family: sans-serif; |
|  | font-size: 1rem; |
|  | padding: 0.5rem; |
|  | margin: 0.5rem 0; |
|  | margin-right: 0.5rem; |
|  | border: 1px solid #ccc; |
|  | border-radius: 3px; |
|  | width: 100%; |
|  | max-width: 400px; |
|  | } |
|  |  |
|  | input[type="submit"] { |
|  | font-family: sans-serif; |
|  | font-size: 1rem; |
|  | padding: 0.5rem; |
|  | margin: 0.5rem 0; |
|  | margin-right: 0.5rem; |
|  | border: 1px solid #ccc; |
|  | border-radius: 3px; |
|  | width: 100%; |
|  | max-width: 400px; |
|  | background: #222; |
|  | color: #FFF; |
|  | cursor: pointer; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | **12) React Navbar** |
|  | **// App.js** |
|  | import React, { Component } from 'react'; |
|  | import { Link } from "react-router-dom"; |
|  | import './App.css'; |
|  |  |
|  | class App extends Component { |
|  | constructor() { |
|  | super(); |
|  | this.state = { |
|  | navbarItems: [ |
|  | { name: 'Home', link: '/' }, |
|  | { name: 'About', link: '/about' }, |
|  | { name: 'Services', link: '/services' }, |
|  | { name: 'Contact', link: '/contact' }, |
|  | ] |
|  | } |
|  | } |
|  |  |
|  | render() { |
|  | return ( |
|  | <div className="App"> |
|  | <nav className="navbar"> |
|  | <ul> |
|  | {this.state.navbarItems.map((item, index) => ( |
|  | <li key={index}> |
|  | <Link to={item.link}>{item.name}</Link> |
|  | </li> |
|  | ))} |
|  | </ul> |
|  | </nav> |
|  | <div className="breadcrumbs"> |
|  | <ul> |
|  | <li><Link to='/'>Home</Link></li> |
|  | <li><Link to='/about'>About</Link></li> |
|  | <li><Link to='/services'>Services</Link></li> |
|  | <li><Link to='/contact'>Contact</Link></li> |
|  | </ul> |
|  | </div> |
|  | </div> |
|  | ); |
|  | } |
|  | } |
|  |  |
|  | export default App; |
|  |  |
|  |  |
|  |  |
|  | **// App.css** |
|  | .navbar { |
|  | display: flex; |
|  | flex-direction: row; |
|  | justify-content: center; |
|  | background-color: #000; |
|  | width: 100%; |
|  | } |
|  |  |
|  | .navbar ul { |
|  | list-style: none; |
|  | padding: 0; |
|  | margin: 0; |
|  | display: flex; |
|  | flex-direction: row; |
|  | justify-content: space-around; |
|  | } |
|  |  |
|  | .navbar ul li { |
|  | margin: 0; |
|  | padding: 0; |
|  | } |
|  |  |
|  | .navbar ul li a { |
|  | color: #fff; |
|  | text-decoration: none; |
|  | padding: 10px 40px; |
|  | font-size: 16px; |
|  | } |
|  |  |
|  | .navbar ul li a:hover { |
|  | background-color: #999; |
|  | color: #000; |
|  | } |
|  |  |
|  | .breadcrumbs { |
|  | display: none; |
|  | } |
|  |  |
|  | @media (max-width: 600px) { |
|  | .navbar { |
|  | display: none; |
|  | } |
|  |  |
|  | .breadcrumbs { |
|  | display: flex; |
|  | flex-direction: column; |
|  | align-items: center; |
|  | background-color: #000; |
|  | width: 100%; |
|  | } |
|  |  |
|  | .breadcrumbs ul { |
|  | list-style: none; |
|  | padding: 0; |
|  | margin: 0; |
|  | display: flex; |
|  | flex-direction: row; |
|  | justify-content: center; |
|  | } |
|  |  |
|  | .breadcrumbs ul li { |
|  | margin: 0; |
|  | padding: 0; |
|  | } |
|  |  |
|  | .breadcrumbs ul li a { |
|  | color: #fff; |
|  | text-decoration: none; |
|  | padding: 10px 20px; |
|  | font-size: 16px; |
|  | } |
|  |  |
|  | .breadcrumbs ul li a:hover { |
|  | background-color: #999; |
|  | color: #000; |
|  | } |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **13) Photo gallery app** |
|  | **// app.js** |
|  | import React, { useState } from 'react'; |
|  | import { useSelector, useDispatch } from 'react-redux'; |
|  | import { |
|  | getPhotos, |
|  | uploadPhoto, |
|  | deletePhoto |
|  | } from './redux/actions/photoActions'; |
|  | import { |
|  | Grid, |
|  | Card, |
|  | CardActionArea, |
|  | CardMedia, |
|  | CardContent, |
|  | Typography, |
|  | Button, |
|  | Modal, |
|  | Input |
|  | } from '@material-ui/core'; |
|  | import './app.css'; |
|  |  |
|  | const App = () => { |
|  | const [open, setOpen] = useState(false); |
|  | const [image, setImage] = useState(null); |
|  | const [openDelete, setOpenDelete] = useState(false); |
|  | const [deleteId, setDeleteId] = useState(null); |
|  | const photos = useSelector(state => state.photos); |
|  | const dispatch = useDispatch(); |
|  |  |
|  | const handleOpen = () => { |
|  | setOpen(true); |
|  | }; |
|  |  |
|  | const handleClose = () => { |
|  | setOpen(false); |
|  | }; |
|  |  |
|  | const handleOpenDelete = id => { |
|  | setOpenDelete(true); |
|  | setDeleteId(id); |
|  | }; |
|  |  |
|  | const handleCloseDelete = () => { |
|  | setOpenDelete(false); |
|  | setDeleteId(null); |
|  | }; |
|  |  |
|  | const handleImageUpload = e => { |
|  | setImage(e.target.files[0]); |
|  | }; |
|  |  |
|  | const handleUpload = () => { |
|  | dispatch(uploadPhoto(image)); |
|  | handleClose(); |
|  | }; |
|  |  |
|  | const handleDelete = () => { |
|  | dispatch(deletePhoto(deleteId)); |
|  | handleCloseDelete(); |
|  | }; |
|  |  |
|  | React.useEffect(() => { |
|  | dispatch(getPhotos()); |
|  | }, [dispatch]); |
|  |  |
|  | return ( |
|  | <div className="app"> |
|  | <Button |
|  | variant="contained" |
|  | color="primary" |
|  | onClick={handleOpen} |
|  | > |
|  | Upload |
|  | </Button> |
|  | <Grid container spacing={2}> |
|  | {photos.map(photo => ( |
|  | <Grid item xs={3} key={photo.id}> |
|  | <Card> |
|  | <CardActionArea> |
|  | <CardMedia |
|  | component="img" |
|  | alt="Photo" |
|  | height="250" |
|  | image={photo.url} |
|  | /> |
|  | <CardContent> |
|  | <Typography gutterBottom variant="h5" component="h2"> |
|  | {photo.name} |
|  | </Typography> |
|  | <Button |
|  | variant="contained" |
|  | color="secondary" |
|  | onClick={() => handleOpenDelete(photo.id)} |
|  | > |
|  | Delete |
|  | </Button> |
|  | </CardContent> |
|  | </CardActionArea> |
|  | </Card> |
|  | </Grid> |
|  | ))} |
|  | </Grid> |
|  | <Modal open={open} onClose={handleClose}> |
|  | <div className="modal"> |
|  | <Input |
|  | type="file" |
|  | onChange={handleImageUpload} |
|  | /> |
|  | <Button |
|  | variant="contained" |
|  | color="primary" |
|  | onClick={handleUpload} |
|  | > |
|  | Upload |
|  | </Button> |
|  | </div> |
|  | </Modal> |
|  | <Modal open={openDelete} onClose={handleCloseDelete}> |
|  | <div className="modal"> |
|  | <Typography gutterBottom variant="h5" component="h2"> |
|  | Are you sure you want to delete this photo? |
|  | </Typography> |
|  | <Button |
|  | variant="contained" |
|  | color="primary" |
|  | onClick={handleDelete} |
|  | > |
|  | Delete |
|  | </Button> |
|  | </div> |
|  | </Modal> |
|  | </div> |
|  | ); |
|  | }; |
|  |  |
|  | export default App; |
|  |  |
|  | **/\* App.css \*/** |
|  |  |
|  | body { |
|  | font-family: 'Arial', sans-serif; |
|  | margin: 0; |
|  | padding: 0; |
|  | background: #f8f8f8; |
|  | } |
|  |  |
|  | #root { |
|  | display: flex; |
|  | flex-direction: column; |
|  | } |
|  |  |
|  | .App { |
|  | padding: 20px; |
|  | } |
|  |  |
|  | .gallery-grid { |
|  | display: flex; |
|  | flex-wrap: wrap; |
|  | justify-content: space-around; |
|  | } |
|  |  |
|  | .gallery-grid-item { |
|  | width: 20%; |
|  | margin: 5px; |
|  | } |
|  |  |
|  | .gallery-grid-item img { |
|  | width: 100%; |
|  | border-radius: 3px; |
|  | } |
|  |  |
|  | .gallery-modal { |
|  | position: fixed; |
|  | left: 0; |
|  | top: 0; |
|  | width: 100%; |
|  | height: 100%; |
|  | background: rgba(0, 0, 0, 0.9); |
|  | display: flex; |
|  | justify-content: center; |
|  | align-items: center; |
|  | z-index: 1000; |
|  | transition: all 0.2s ease; |
|  | opacity: 0; |
|  | visibility: hidden; |
|  | } |
|  |  |
|  | .gallery-modal.show { |
|  | opacity: 1; |
|  | visibility: visible; |
|  | } |
|  |  |
|  | .modal-img { |
|  | width: 90%; |
|  | max-height: 90%; |
|  | border-radius: 3px; |
|  | cursor: pointer; |
|  | transition: all 0.3s ease; |
|  | } |
|  |  |
|  | .modal-img:hover { |
|  | transform: scale(1.2); |
|  | } |