**Array-Objects**

**App.js:**

import React, { Component } from 'react';

import './index.css';

class TableView extends Component {

  renderTableHeader() {

    const header = Object.keys(this.props.data[0]);

    return header.map((key) => <th key={key}>{key}</th>);

  }

  renderTableData() {

    return this.props.data.map((item, index) => {

      const rowStyle = {

        backgroundColor: index % 2 === 0 ? '#f2f2f2' : 'white', // Alternating background colors

      };

      return (

        <tr key={index} style={rowStyle}>

          {Object.entries(item).map(([key, value], index) => {

            const cellStyle = {

              border: '1px solid black',

              textAlign: key === 'Qualification' ? 'left' : 'center',

            };

            return <td key={index} style={cellStyle}>{value}</td>;

          })}

        </tr>      );

    });

  }

  render() {

    const tableStyle = {

      borderCollapse: 'collapse',

      width: '70%',

      margin: '20px auto', // Center the table horizontally

      border: '1px solid blue',

    };

    const headerStyle = {

      backgroundColor: '#3498db',

    };

    return (

      <div>

        <table style={tableStyle}>

          <thead style={headerStyle}>

            <tr>{this.renderTableHeader()}</tr>

          </thead>

          <tbody>{this.renderTableData()}</tbody>

        </table>

      </div>

    );

  }

}

class App extends Component {

  render() {

    const data = [

      { ID: 1, Name: 'KBK', Qualification: 'B.Tech., M.Tech., MBA., DID., (Ph.D)' },

      { ID: 2, Name: 'TVC', Qualification: '(B.Tech)' },

      { ID: 3, Name: 'JRP', Qualification: '(B.Tech)' },

      { ID: 4, Name: 'NBS', Qualification: '(B.Tech)' },

      { ID: 5, Name: 'KMC', Qualification: '(B.Tech)' },

    ];

    return (

      <div>

        <h1 style={{ textAlign: 'center', borderBottom: '2px solid darkblue', paddingTop: '20px', marginBottom: '100px' }}>

          Array of Objects - Table View

        </h1>

        <TableView data={data} />

        <div className="watermark">

        <p style={{ textAlign: 'right', marginRight: '20px', fontWeight: 'bold'}}>

        </p>

        </div>

      </div>

    );

  }

}

export default App;

**Index.css:**

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

  background-color: rgb(217,230,173);

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

.watermark {

  position: fixed;

  bottom: 10px;

  right: 10px;

  color: rgba(255, 55, 0, 0.4);

  font-size: 15px; }

**Digital clock:**

**App.js:**

import React from "react";

import "./style.css";

import Clock from "./clock.js";

class DigitalClock extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      time: new Date()

    };

  }

  componentDidMount() {

    this.timerId = setInterval(() => {

      this.setState({

        time: new Date()

      });

    }, 1000);

  }

  componentWillUnmount() {

    clearInterval(this.timerId);

  }

  render() {

    const { time } = this.state;

    const hours = time.getHours().toString().padStart(2, "0");

    const minutes = time.getMinutes().toString().padStart(2, "0");

    const seconds = time.getSeconds().toString().padStart(2, "0"); // Corrected variable name

    const ampm = hours >= 12 ? "PM" : "AM";

    const displayHours = (hours % 12 || 12).toString().padStart(2, "0");

    const options = { day: "numeric", month: "short", year: "numeric" };

    const formattedDate = new Intl.DateTimeFormat("en-US", options).format(time);

    return (

      <div className="digital-clock">

        <h2>

  {displayHours}:{minutes}:{seconds}

  <sup style={{ fontSize: "0.4em", verticalAlign: "top" }}>{ampm}</sup>

</h2>

        <div className="date-container" style={{ textAlign: "center" }}>

          <p style={{ fontSize: "0.6em", margin: "0" }}>{formattedDate}</p>

        </div>

      </div>

    );

  }

}

function App() {

  return (

    <div>

      <h1 style={{ textAlign: "center", borderBottom: "2px solid orange", paddingTop: "20px", color: "blue" }}>

        Analog & Digital Clock {/\* Corrected text \*/}

      </h1>

      <div className="clocks-container">

        <Clock />

        <DigitalClock />

      </div>

    </div>

  );

}

export default App;

**clock.js:**

import React, { Component } from "react";

export default class Clock extends Component {

    constructor(props) {

        super(props);

        this.state = {

            time: new Date()

        };

    }

    componentDidMount() {

        this.timerId = setInterval(() => {

            this.setState({

                time: new Date()

            });

        }, 1000);

    }

    componentWillUnmount() {

        clearInterval(this.timerId);

    }

    render() {

        const hourRotation = this.state.time.getHours() \* 30;

        const minuteRotation = this.state.time.getMinutes() \* 6;

        const secondRotation = this.state.time.getSeconds() \* 6;

        return (

            <div className="clock">

                <div className="analog-clock">

                    <div

                        className="hour\_hand"

                        style={{ transform: `rotateZ(${hourRotation}deg)` }}

                    />

                    <div

                        className="min\_hand"

                        style={{ transform: `rotateZ(${minuteRotation}deg)` }}

                    />

                    <div

                        className="sec\_hand"

                        style={{ transform: `rotateZ(${secondRotation}deg)` }}

                    />

                    <span className="twelve">12</span>

                    <span className="one">1</span>

                    <span className="two">2</span>

                    <span className="three">3</span>

                    <span className="four">4</span>

                    <span className="five">5</span>

                    <span className="six">6</span>

                    <span className="seven">7</span>

                    <span className="eight">8</span>

                    <span className="nine">9</span>

                    <span className="ten">10</span>

                    <span className="eleven">11</span>

                </div>

            </div>

        );

    }

}

**Style.css:**

body{

  margin:0;

}

.App{

  font-family: sans-serif;

  position: relative;

}

.watermark{

  position:absolute;

  bottom: 10px;

  right: 10px;

  color: rgb(11, 9, 9);

  font-size: 20px;

}

.clocks-container{

  display: flex;

  justify-content: center;

  align-items: center;

}

.clock,.digital-clock{

  width: 300px;

  height: 300px;

  border-radius: 50%;

  position: relative;

  box-shadow: 0 2px 30px rgba(0,0,0,0.2);

  font-size: 24px;

  color:#444;

  text-align: center;

  margin: 50px;

  background-color: #f6fe00;

}

.clock::after{

  background: #aaa;

  content: "";

  width:12px;

  height:12px;

  border-radius: 50%;

  position: absolute;

  z-index:2;

  top: 50%;

  left: 50%;

  transform: translate(-50%,-50%);

   border: 2px solid #fff;

}

.digital-clock h2{

  font-family: "Perfograma","Source Sans Pro",sans-serif;

  font-weight: 700;

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%,-50%);

  margin: 0;

}

.hour\_hand{

  position: absolute;

  width: 6px;

  height: 60px;

  background:#222;

  top:30%;

  left: 49%;

  transform-origin: bottom;

}

.min\_hand{

  position: absolute;

  width: 4px;

  height: 80px;

  background:#444;

  top:22.5%;

  left: 49%;

  transform-origin: bottom;

}

.sec\_hand{

  position: absolute;

  width: 2px;

  height: 118px;

  background:red;

  top:10.5%;

  left: 50%;

  transform-origin: bottom;

}

.clock span{

  position: absolute;

  font-family: "Source Sans Pro",sans-serif;

  font-weight: 700;

}

.twelve{

  top: 10px;

  left:46%;

}

.one{

  top: 10%;

  right:26%;

}

.eleven{

  top: 10%;

  left:26%;

}

.two{

  top: 25%;

  right:10%;

}

.three{

  right: 10px;

  top:46%;

}

.four{

  right: 30px;

  top:67%;

}

.five{

  right: 78px;

  top:80%;

}

.six{

  bottom: 10px;

  left:50%;

}

.seven{

  left: 80px;

  top:82%;

}

.eight{

  left: 30px;

  top:67%;

}

.nine{

  left: 10px;

  top:46%;

}

.ten{

  top:25%;

  left:10%;

}

.clock\_\_text-ampm{

  font-size: var(--tiny-font-size);

  color: var(--title-color);

  font-weight: var(--font-medium);

  margin-left: var(--mb-0-25);

}

.date-container{

  position: absolute;

  bottom: 35%;

  left: 50%;

  transform: translate(-50%,-50%);

  margin: 0;

  font-size: 16px;

  color: #444;

}

.clock\_\_date{

  text-align: center;

  font-size: var(--small-font-size);

  font-weight: var(--font-medium);

}

**Textcolor:**

**ColorChanger.js:**

import { Component } from "react";

class ColorChanger extends Component {

    constructor(props) {

        super(props);

        this.state = {

            textColor: 'black',

            fontSize: '20px',

            fontStyle: 'normal',

            textDecoration: 'none'

        };

        this.watermarkStyle = {

            position: 'absolute',

            top: '50%',

            left: '50%',

            transform: 'translate(-50%, -50%)',

            zIndex: '-1',

            opacity: '0.3',

            fontFamily: 'Arial, sans-serif',

            fontSize: '16px',  // Fixed font size

            color: 'blue'

        };

    }

    handleColorChange = (event) => {

        this.setState({ textColor: event.target.value });

    };

    handleDecorationChange = (event) => {

        this.setState({ textDecoration: event.target.value });

    };

    handleStyleChange = (event) => {

        this.setState({ fontStyle: event.target.value });

    };

    handleSizeChange = (event) => {

        this.setState({ fontSize: `${event.target.value}px` });

    };

    render() {

        const { textColor, fontSize, fontStyle, textDecoration } = this.state;

        return (

            <div style={{ textAlign: 'center', marginTop: '50px' }}>

                {/\* Color Picker \*/}

                <label htmlFor='colorpicker'>Select Text Color:</label>

                <input

                    type='color'

                    id='colorpicker'

                    value={textColor}

                    onChange={this.handleColorChange}

                />

                {/\* Font Size \*/}

                <label htmlFor="fontSize">Select Text Size:</label>

                <input

                    type='number'

                    id='fontSize'

                    value={parseInt(fontSize)}

                    onChange={this.handleSizeChange}

                />

                {/\* Font Style \*/}

                <label htmlFor='fontStyle'>Select Text Style:</label>

                <select id='fontStyle' value={fontStyle} onChange={this.handleStyleChange}>

                    <option value='normal'>Normal</option>

                    <option value='italic'>Italic</option>

                </select>

                {/\* Text Decoration \*/}

                <label htmlFor='textDecoration'>Select Text Decoration:</label>

                <select id='textDecoration' value={textDecoration} onChange={this.handleDecorationChange}>

                    <option value="none">None</option>

                    <option value="underline">Underline</option>

                    <option value="overline">Overline</option>

                    <option value="line-through">Line Through</option>

                </select>

                {/\* Display the text with selected styles \*/}

                <div

                    style={{

                        color: textColor,

                        marginTop: '20px',

                        fontFamily: 'Arial, sans-serif',

                        fontSize: fontSize,

                        fontStyle,

                        textDecoration

                    }}

                >

                    <h1>It's Your Choice, See the Magic!</h1>

                </div>

            </div>

        );

    }

}

export default ColorChanger;

**Hooks**

**App.js:**

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

const ThemeContext = createContext();

const ThemeProvider = ({ children }) => {

  const [theme, setTheme] = useState("light");

  const toggleTheme = () => {

    setTheme((prevTheme) => (prevTheme === "light" ? "dark" : "light"));

  };

  return (

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

      {children}

    </ThemeContext.Provider>

  );

};

// Themed Component (Uses useContext)

const ThemedComponent = () => {

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

  const ThemedComponentStyle = {

    background: theme === "light" ? "#fff" : "#333",

    color: theme === "light" ? "#333" : "#fff",

    border: "1px solid #ccc",

    borderRadius: "8px",

    padding: "20px",

    margin: "20px 0",

    textAlign: "center",

  };

  return (

    <div style={ThemedComponentStyle}>

      <h2>Themed Component</h2>

      <p>Current Theme: {theme}</p>

      <button onClick={toggleTheme}>Toggle Theme</button>

    </div>

  );

};

// Counter Component (Uses useEffect)

const CounterComponent = () => {

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

  useEffect(() => {

    document.title = `Count: ${count}`;

  }, [count]);

  const counterStyle = {

    margin: "20px 0",

    padding: "20px",

    border: "1px solid #ccc",

    borderRadius: "8px",

    textAlign: "center",

  };

  return (

    <div style={counterStyle}>

      <h2>Counter Component</h2>

      <p>Count: {count}</p>

      <button onClick={() => setCount(count + 1)}>Increment</button>

      <button onClick={() => setCount(count - 1)}>Decrement</button>

    </div>

  );

};

// User Info Component

const UserInfoComponent = () => {

  const [userInfo, setUserInfo] = useState({ name: "", age: 0 });

  useEffect(() => {

    const fetchUserInfo = async () => {

      setTimeout(() => {

        setUserInfo({ name: "asari Harika", age: 20 });

      }, 1000);

    };

    fetchUserInfo();

  }, []);

  const userInfoStyle = {

    margin: "20px 0",

    padding: "20px",

    border: "1px solid #ccc",

    borderRadius: "8px",

    textAlign: "center",

  };

  return (

    <div style={userInfoStyle}>

      <h2>User Info Component</h2>

      <p>Name: {userInfo.name}</p>

      <p>Age: {userInfo.age}</p>

    </div>

  );

};

// Watermark Component

const WatermarkComponent = () => {

  const watermarkStyle = {

    position: "fixed",

    bottom: "10px",

    right: "10px",

    opacity: 1,

    fontSize: "12px",

  };

  return <div style={watermarkStyle}>DASARI HARIKA</div>;

};

// Main App Component

const App = () => {

  const appStyle = {

    fontFamily: "Arial, sans-serif",

    maxWidth: "600px",

    margin: "auto",

    padding: "20px",

    textAlign: "center",

  };

  return (

    <ThemeProvider>

      <div style={{ textAlign: "center", marginTop: "10px" }}>

        <h1 style={{ borderBottom: "2px solid orange", color: "blue" }}>

          All Hooks @ Once

        </h1>

      </div>

      <div style={appStyle}>

        <CounterComponent />

        <ThemedComponent />

        <UserInfoComponent />

      </div>

      <WatermarkComponent />s

    </ThemeProvider>

  );

};

export default App;