Cognizant - DN 4.0 I Deep Skilling

Week-7

**10] Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.**

**Create an element to display the heading of the page.**

**Attribute to display the image of the office space**

**Create an object of office to display the details like Name, Rent and Address.**

**Create a list of Object and loop through the office space item to display more data.**

**To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000**

import React from 'react';

import './App.css';

function App() {

// Heading JSX

const heading = <h1>🏢 Office Space Rental Portal</h1>;

// Office Image JSX

const officeImage = "https://images.unsplash.com/photo-1504384308090-c894fdcc538d"; // Sample image URL

// Single office object

const office = {

name: "Prestige Tech Park",

rent: 55000,

address: "Whitefield, Bangalore"

};

// List of office objects

const officeList = [

{

name: "WeWork Residency",

rent: 75000,

address: "MG Road, Bangalore"

},

{

name: "IndiQube Alpha",

rent: 50000,

address: "Koramangala, Bangalore"

},

{

name: "Regus RMZ EcoWorld",

rent: 85000,

address: "Outer Ring Road, Bangalore"

}

];

// Function to get rent color

const getRentStyle = (rent) => {

return { color: rent > 60000 ? 'green' : 'red' };

};

return (

<div className="App">

{heading}

{/\* Office Image \*/}

<img src={officeImage} alt="Office Space" width="400" />

{/\* Display Single Office Object \*/}

<div className="office-card">

<h2>{office.name}</h2>

<p><strong>Address:</strong> {office.address}</p>

<p><strong>Rent:</strong> <span style={getRentStyle(office.rent)}>{office.rent}</span></p>

</div>

{/\* Display List of Offices \*/}

<h2>Available Office Spaces</h2>

{officeList.map((item, index) => (

<div className="office-card" key={index}>

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

<p><strong>Address:</strong> {item.address}</p>

<p><strong>Rent:</strong> <span style={getRentStyle(item.rent)}>{item.rent}</span></p>

</div>

))}

</div>

);

}

export default App;

**11] Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.**

1. **Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.** 
   1. **To increment the value**
   2. **Say Hello followed by a static message.**

import React, { useState } from 'react';

import './App.css'

function App() {

const [counter, setCounter] = useState(0);

// Function to increment counter

const increment = () => {

setCounter(prev => prev + 1);

};

// Function to say hello and show static message

const sayHello = () => {

alert("Hello! This is a static message from Cognizant Academy.");

};

// Function to handle multiple actions on increment

const handleIncrementClick = () => {

increment();

sayHello();

};

// Function to decrement counter

const decrement = () => {

setCounter(prev => prev - 1);

};

return (

<div className="App">

<h1>🎯 Event Handling in React</h1>

<h2>Counter: {counter}</h2>

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

<button onClick={decrement} style={{ marginLeft: '10px' }}>Decrement</button>

</div>

);

}

export default App;

.App {

text-align: center;

padding: 50px;

font-family: 'Segoe UI', sans-serif;

}

button {

font-size: 16px;

padding: 10px 20px;

margin-top: 20px;

cursor: pointer;

}

**12] Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.**

**The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed**.

import React from 'react';

const flights = [

{ id: 1, from: "Delhi", to: "Mumbai", time: "10:00 AM", price: 5000 },

{ id: 2, from: "Chennai", to: "Bangalore", time: "1:30 PM", price: 3500 },

{ id: 3, from: "Kolkata", to: "Hyderabad", time: "5:00 PM", price: 4500 }

];

function FlightList({ showBooking }) {

return (

<div>

<h2>Available Flights</h2>

<ul>

{flights.map((flight) => (

<li key={flight.id}>

✈️ {flight.from} → {flight.to} at {flight.time} – ₹{flight.price}

{showBooking && <button style={{ marginLeft: '15px' }}>Book Ticket</button>}

</li>

))}

</ul>

</div>

);

}

export default FlightList;

import React from 'react';

import FlightList from './FlightList';

function GuestPage() {

return (

<div>

<h1>Welcome, Guest!</h1>

<p>Browse available flights below.</p>

<FlightList showBooking={false} />

</div>

);

}

export default GuestPage;

import React from 'react';

import FlightList from './FlightList';

function UserPage() {

return (

<div>

<h1>Welcome, Logged-in User!</h1>

<p>You can now book flights.</p>

<FlightList showBooking={true} />

</div>

);

}

export default UserPage;

import React, { useState } from 'react';

import GuestPage from './components/GuestPage';

import UserPage from './components/UserPage';

import './App.css';

function App() {

const [isLoggedIn, setIsLoggedIn] = useState(false);

const handleLogin = () => setIsLoggedIn(true);

const handleLogout = () => setIsLoggedIn(false);

return (

<div className="App">

<header>

<h1>🛫 Ticket Booking App</h1>

{isLoggedIn ? (

<button onClick={handleLogout}>Logout</button>

) : (

<button onClick={handleLogin}>Login</button>

)}

</header>

<main>

{isLoggedIn ? <UserPage /> : <GuestPage />}

</main>

</div>

);

}

export default App;

.App {

font-family: 'Segoe UI', sans-serif;

padding: 30px;

background-color: #f4f4f4;

}

header {

display: flex;

justify-content: space-between;

align-items: center;

}

button {

padding: 10px 15px;

background-color: #1976d2;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #125ba1;

}

ul {

list-style-type: none;

padding: 0;

}

li {

margin: 12px 0;

}

**13] Create a React App named “bloggerapp” in with 3 components.**

1. **Book Details**
2. **Blog Details**
3. **Course Details**

// src/components/BookDetails.js

import React from 'react';

function BookDetails() {

return (

<div>

<h2>📘 Book Details</h2>

<p>Title: The Pragmatic Programmer</p>

<p>Author: Andrew Hunt & David Thomas</p>

</div>

);

}

export default BookDetails;

// src/components/BlogDetails.js

import React from 'react';

function BlogDetails() {

return (

<div>

<h2>✍️ Blog Details</h2>

<p>Title: How to Build React Apps</p>

<p>Author: Violet King</p>

</div>

);

}

export default BlogDetails;

// src/components/CourseDetails.js

import React from 'react'

function CourseDetails() {

return (

<div>

<h2>📚 Course Details</h2>

<p>Course: Full Stack Web Development</p>

<p>Instructor: John Doe</p>

</div>

);

}

export default CourseDetails;

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

import './App.css';

function App() {

const [selected, setSelected] = useState("book");

// 1. Using if-else

let renderedComponent;

if (selected === "book") {

renderedComponent = <BookDetails />;

} else if (selected === "blog") {

renderedComponent = <BlogDetails />;

} else {

renderedComponent = <CourseDetails />;

}

return (

<div className="App">

<h1>📚 Blogger App</h1>

<div className="button-group">

<button onClick={() => setSelected("book")}>Book</button>

<button onClick={() => setSelected("blog")}>Blog</button>

<button onClick={() => setSelected("course")}>Course</button>

</div>

<hr />

{/\* 1. Conditional Rendering using if-else (above) \*/}

{renderedComponent}

<hr />

{/\* 2. Conditional Rendering using Ternary Operator \*/}

<h3>🔁 Ternary Method:</h3>

{selected === "book" ? <BookDetails /> : selected === "blog" ? <BlogDetails /> : <CourseDetails />}

<hr />

{/\* 3. Conditional Rendering using Logical AND \*/}

<h3>✅ Logical AND Method:</h3>

{selected === "book" && <BookDetails />}

{selected === "blog" && <BlogDetails />}

{selected === "course" && <CourseDetails />}

<hr />

{/\* 4. Conditional Rendering using Switch statement (inline function) \*/}

<h3>🧩 Switch Statement Method:</h3>

{(() => {

switch (selected) {

case "book": return <BookDetails />;

case "blog": return <BlogDetails />;

case "course": return <CourseDetails />;

default: return null;

}

})()}

</div>

);

}

export default App;

.App {

padding: 40px;

font-family: 'Segoe UI', sans-serif;

background: #f8f9fa;

color: #333;

}

button {

margin-right: 10px;

padding: 10px 20px;

background-color: #007bff;

border: none;

color: white;

cursor: pointer;

border-radius: 4px;

}

button:hover {

background-color: #0056b3;

}

hr {

margin: 20px 0;

}