**React**

**Handson 13**

**1.Explain various ways of conditional rendering:**

* **Using if-else statement:**

if (isLoggedIn) {

return <Dashboard />;

} else {

return <LoginPage />;

}

* **Using ternary operator (**? :**):**

{isLoggedIn ? <Dashboard /> : <LoginPage />}

* **Using logical AND (**&&**) operator:**

{hasPermission && <AdminPanel />}

**2.Explain how to render multiple components:**

You can render multiple components in React by placing them inside a wrapper (like a <div>, <section>, or <> fragment):

**Example:**

function App() {

return (

<div>

<Header />

<Content />

<Footer />

</div>

);

}

**3.Define list component:**

A List component in React is a component that displays a list of items by mapping over an array.

**Example:**

function NumberList(props) {

const numbers = props.numbers;

return (

<ul>

{numbers.map((num) => <li key={num}>{num}</li>)}

</ul>

);

}

**Usage:**

<NumberList numbers={[1, 2, 3, 4]} />

**4.Explain about keys in React applications:**

Keys help React identify which items have changed, are added, or removed in a list.

* Keys should be unique among siblings.
* Best practice is to use a unique id (not the index, unless no better option).

**Example:**

{items.map(item => <li key={item.id}>{item.name}</li>)}

**5.Explain how to extract components with keys:**

If you map items to components, ensure that the key is passed to the top-level element of the component being returned.

**Example:**

function Item(props) {

return <li>{props.value}</li>;

}

function ItemList() {

const items = ['Apple', 'Banana', 'Mango'];

return (

<ul>

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

<Item key={index} value={item} />

))}

</ul>

);

}

**6.Explain React Map, map() function:**

The map() function in JavaScript is commonly used in React to loop through an array and render JSX for each item.

**Example:**

const fruits = ['Apple', 'Banana', 'Mango'];

const listItems = fruits.map((fruit, index) =>

<li key={index}>{fruit}</li>

);

**Used inside a component:**

function FruitList() {

return (

<ul>

{fruits.map((fruit, index) => (

<li key={index}>{fruit}</li>

))}

</ul>

);

}

**BlogDetails.js:**

function BlogDetails() {

  return <h2>📰 Blog Details: Read the latest React blogs!</h2>;

}

export default BlogDetails;

**BookDetails.js:**

function BookDetails() {

  return <h2>📖 Book Details: Learn React through books!</h2>;

}

export default BookDetails;

**CourseDetails.js:**

import CourseItem from "./CourseItem";

import "./App.css";

function CourseDetails() {

  const courses = [

    { id: 1, title: "React Basics", duration: "3 weeks" },

    { id: 2, title: "Advanced React", duration: "4 weeks" },

    { id: 3, title: "MERN Full Stack", duration: "6 weeks" },

  ];

  return (

    <div className="course-list">

      <h2> Available Courses</h2>

      <ul>

        {courses.map((course) => (

          <CourseItem key={course.id} course={course} />

        ))}

      </ul>

    </div>

  );

}

export default CourseDetails;

**CourseItem.js:**

function CourseItem({ course }) {

  return (

    <li className="course-item">

      <strong>{course.title}</strong>

      <span>({course.duration})</span>

    </li>

  );

}

export default CourseItem;

**App.js:**

import React, { useState } from "react";

import BookDetails from "./BookDetails";

import BlogDetails from "./BlogDetails";

import CourseDetails from "./CourseDetails";

import "./App.css";

function App() {

  const [choice, setChoice] = useState("book");

  const renderComponent = () => {

    switch (choice) {

      case "book":

        return <BookDetails />;

      case "blog":

        return <BlogDetails />;

      case "course":

        return <CourseDetails />;

      default:

        return <h2>Please select a section</h2>;

    }

  };

  return (

    <div className="App">

      <h1>Blogger App</h1>

      <div className="buttons">

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

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

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

      </div>

      <div className="content">{renderComponent()}</div>

    </div>

  );

}

export default App;

**App.css:**

.App {

  font-family: Arial, sans-serif;

  text-align: center;

  padding: 20px;

}

h1 {

  color: #2c3e50;

  margin-bottom: 20px;

}

.buttons button {

  background-color: #3498db;

  border: none;

  color: white;

  padding: 10px 16px;

  margin: 5px;

  font-size: 16px;

  cursor: pointer;

  border-radius: 8px;

  transition: background 0.3s;

}

.buttons button:hover {

  background-color: #2980b9;

}

.content {

  margin-top: 30px;

}

.course-list ul {

  list-style: none;

  padding: 0;

}

.course-item {

  background: #f2f2f2;

  padding: 10px;

  margin: 8px auto;

  width: 50%;

  border-radius: 6px;

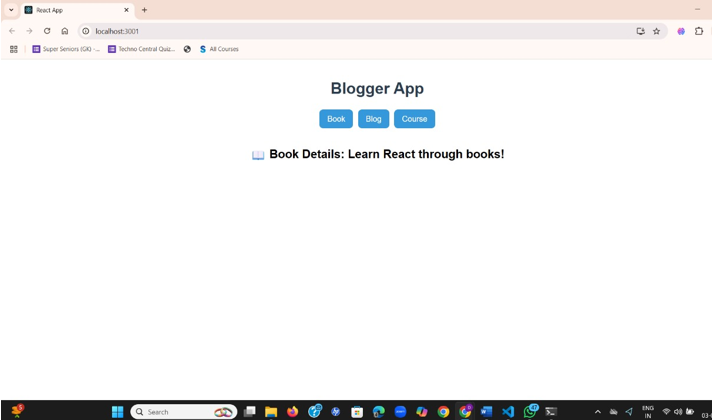
  display: flex;

  justify-content: space-between;

  font-size: 18px;

}

**Output:**

****

A screenshot of a computer

AI-generated content may be incorrect.