**Explain various ways of conditional rendering**

Conditional rendering means deciding what to display based on logic (like user login, form values, etc.).

**A. if/else statement:**

if (isLoggedIn) {

return <Dashboard />;

} else {

return <LoginPage />;

}

**B. Ternary operator:**

return (

<div>

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

</div>

);

**C. Logical AND (&&):**

{hasNotifications && <NotificationList />}

**D. Using Element Variables:**

let content;

if (isLoggedIn) {

content = <Dashboard />;

} else {

content = <LoginPage />;

}

return <div>{content}</div>;

**Explain how to render multiple components**

Just return them together in a single wrapper (like a <div>, React Fragment <>, or a custom parent component):

function App() {

return (

<>

<Header />

<Sidebar />

<Content />

<Footer />

</>

);

}

React **Fragments** (<> </>) are commonly used to avoid extra <div> wrappers.

**Define list component**

A List Component in React displays a collection of similar data items (like users, products, posts).

**Example:**

function NameList(props) {

return (

<ul>

{props.names.map((name) => (

<li key={name}>{name}</li>

))}

</ul>

);

}

Use map() to loop through the data and render each item.

**Explain about keys in React applications**

Keys are unique identifiers used by React to **track elements** in a list between renders.

**Why they matter:**

* Help React efficiently update or re-order items
* Prevents performance issues and bugs in dynamic lists

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

const listItems = items.map((item) => <li key={item}>{item}</li>);

Use unique IDs as keys when available. Avoid using array index unless there's no other choice.

**Explain how to extract components with keys**

When mapping over data, you can extract each item into its own component and pass a key:

function Item({ name }) {

return <li>{name}</li>;

}

function ItemList({ items }) {

return (

<ul>

{items.map((item) => (

<Item key={item.id} name={item.name} />

))}

</ul>

);

}

Item is reusable.

key is passed to the outer component (<Item />) during list rendering.

**Explain React Map, map() function**

map() is a **JavaScript** array method used in React to loop over arrays and return JSX for each item.

**Example:**

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

const fruitList = fruits.map((fruit) => <li key={fruit}>{fruit}</li>);

return <ul>{fruitList}</ul>;

map() transforms the array into a new array of React elements.

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

**Index.js :-**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

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

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

reportWebVitals();

**App.js :-**

import React from 'react';

import CourseDetails from './components/CourseDetails';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

function App() {

  const sectionStyle = {

    padding: '0 30px',

    minHeight: '100vh',

  };

  const borderStyle = {

    borderLeft: '5px solid green',

    paddingLeft: '30px',

    marginLeft: '30px'

  };

  return (

    <div style={{ display: 'flex', justifyContent: 'center' }}>

      <div style={sectionStyle}>

        <CourseDetails />

      </div>

      <div style={{ ...sectionStyle, ...borderStyle }}>

        <BookDetails />

      </div>

      <div style={{ ...sectionStyle, ...borderStyle }}>

        <BlogDetails />

      </div>

    </div>

  );

}

export default App;

**BlogDetails.js :-**

import React from 'react';

function BlogDetails() {

  const blogs = [

    {

      title: "React Learning",

      author: "Stephen Biz",

      content: "Welcome to learning React!"

    },

    {

      title: "Installation",

      author: "Schewzdenier",

      content: "You can install React from npm."

    }

  ];

  // Return early conditional rendering

  if (!blogs || blogs.length === 0) return <p>No blogs available</p>;

  return (

    <div>

      <h2>Blog Details</h2>

      {blogs.map((blog, index) => (

        <div key={index}>

          <h3>{blog.title}</h3>

          <strong>{blog.author}</strong>

          <p>{blog.content}</p>

        </div>

      ))}

    </div>

  );

}

export default BlogDetails;

**BookDetails.js :-**

import React from 'react';

function BookDetails() {

  const books = [

    { title: "Master React", price: 670 },

    { title: "Deep Dive into Angular 11", price: 800 },

    { title: "Mongo Essentials", price: 450 }

  ];

  // Using switch-case based conditional rendering

  const renderBook = (book, index) => {

    switch (book.title) {

      case "Master React":

      case "Deep Dive into Angular 11":

      case "Mongo Essentials":

        return (

          <div key={index}>

            <h3>{book.title}</h3>

            <p>{book.price}</p>

          </div>

        );

      default:

        return null;

    }

  };

  return (

    <div>

      <h2>Book Details</h2>

      {books.map(renderBook)}

    </div>

  );

}

export default BookDetails;

**CourseDetails.js :-**

import React from 'react';

function CourseDetails() {

  const courses = [

    { name: "Angular", date: "4/5/2021" },

    { name: "React", date: "6/3/20201" }  // Notice typo in date for demo

  ];

  return (

    <div>

      <h2>Course Details</h2>

      {courses.map((course, index) => {

        // Using if inside map

        if (!course.name || !course.date) return null;

        return (

          <div key={index}>

            <h3>{course.name}</h3>

            <p>{course.date}</p>

          </div>

        );

      })}

    </div>

  );

}

export default CourseDetails;

Output :-



