KONGU ENGINEERING COLLEGE, PERUNDURAI, ERODE - 638 060
DEPARTMENT OF INFORMATION TECHNOLOGY
Professional Elective
20ITE03 User Interface Design
Tutorial

S.No	Questions	CO Level	K Level
	Part - A		
1	What is JSX in React? How does it differ from standard HTML? **JSX is a JavaScript extension used in React for defining UI components. It resembles HTML but supports embedding JavaScript expressions, event handling in camelCase, custom components, and self-closing tags. It's transpiled into JavaScript for rendering, enabling dynamic UI updates.**	CO1	K1
2	Explain the purpose of using JSX in React applications. How does it contribute to the development process? **JSX simplifies UI creation in React by blending HTML-like syntax with JavaScript. It promotes component reusability, dynamic content generation, and clear event handling. This accelerates development, enhances code readability, and enables a more efficient and expressive UI creation process.**	CO1	K1
3	How do you embed expressions or variables inside JSX? Provide an example of how you might use this feature. **You can embed JavaScript expressions or variables within JSX by enclosing them in curly braces {}. This allows you to dynamically insert values, perform calculations, or incorporate logic directly into your JSX code.** **Ex** **import React from 'react';** **import ReactDOM from 'react-dom';** **const userName = 'Alice';** **const greeting = Hello, {userName}! ;** **ReactDOM.render(greeting, document.getElementById('root'));**	CO1	K2
4	In React, what is the role of the "ReactDOM.render()" method? Explain its significance in the context of rendering components. **The "ReactDOM.render()" method in React is pivotal for rendering components onto the real DOM. It initiates the virtual DOM comparison, updates the actual DOM efficiently, and triggers component lifecycle hooks, enabling dynamic UI updates and optimal performance.**	CO1	K2
5	Describe the process of creating a React component using JSX. Include the necessary steps from defining the component to rendering it on the page. **To create a React component using JSX:**	CO1	K1

<table>
<tr><td></td><td>1. Define component logic and structure using JSX tags and expressions.
2. Optionally, include state and event handling.
3. Return JSX within the component's function.
4. Use `ReactDOM.render(component, rootElement)` to render the component inside the root element on the web page.</td><td></td><td></td></tr>
<tr><td>6</td><td>Describe the role of curly braces {} in JSX. Give an example of how they're used.
Curly braces `{}` in JSX allow embedding JavaScript expressions. They're used to inject dynamic values, variables, or calculations into JSX elements. For example, `<p>Hello, {name}!</p>` embeds the value of the `name` variable into the JSX.</td><td>CO1</td><td>K1</td></tr>
<tr><td>7</td><td>Differentiate between JSX elements and JSX components.
<table>
<tr><th>JSX Elements</th><th>JSX Components</th></tr>
<tr><td>Created with HTML-like syntax</td><td>Created as functions or classes</td></tr>
<tr><td>Building blocks of UI</td><td>Reusable UI entities</td></tr>
<tr><td>Can be composed in JSX structure</td><td>Can be composed of multiple elements or other components</td></tr>
<tr><td>Limited reusability without repetition</td><td>Promotes reusability and modularity</td></tr>
<tr><td>Can have props (attributes)</td><td>Can receive props for customization</td></tr>
<tr><td>No lifecycle hooks</td><td>Can have lifecycle hooks for state and behavior management</td></tr>
<tr><td>`<div>`, `<p>`, `<button>`</td><td>`Header`, `ProductCard`, `UserForm`</td></tr>
<tr><td>Directly rendered by `ReactDOM.render()`</td><td>Rendered within other components or `ReactDOM.render()`</td></tr>
</table></td><td>CO1</td><td>K1</td></tr>
<tr><td>8</td><td>Describe how to use the map function to render an array of data into JSX elements.
To render an array of data as JSX elements using `map()`:
1. Create an array of data.
2. Use `map()` on the array, returning JSX for each item.
3. Render the resulting JSX array within your component's structure.
Remember to assign unique `key` props.</td><td>CO1</td><td>K3</td></tr>
<tr><td>9</td><td>Provide an example of rendering a list of items using the map function in JSX.
import React from 'react';
function App() {
const items = [</td><td>CO1</td><td>K1</td></tr>
</table>

	'Item 1', **'Item 2',** **'Item 3',** **];** **const itemList = items.map((item, index) => (** **<li key={index}>{item} ** **));** **return (** ** ** ** List of Items ** ** ** **{itemList}** ** ** ** ** **);** **}** **export default App;**		
10	Discuss best practices for writing clean and maintainable JSX code. Write clean and maintainable JSX code by: **1. Breaking complex components into smaller ones.** **2. Using descriptive variable and prop names.** **3. Organizing code logically.** **4. Applying consistent formatting.** **5. Adding comments for clarity.** **6. Avoiding deeply nested structures.** **7. Utilizing PropTypes or TypeScript for type safety.**	CO1	K1
11	What is a JSX fragment, and why is it useful? **A JSX fragment is an element that doesn't create an actual DOM element but groups multiple elements together. It's useful to avoid unnecessary wrapper elements and maintain clean component structures.**	CO1	K1
12	Provide an example of using a JSX fragment to wrap multiple elements. **import React from 'react';** **function App() {** **return (** **<React.Fragment>** ** Header ** ** This is a paragraph. ** **<button>Click me</button>** **</React.Fragment>** **);** **}** **export default App;**	CO1	K2
13	What is the spread operator in ES6 give an example. **The spread operator (`...`) is a feature introduced in ES6 that allows you to expand elements from an array or object into a new location. It's used for creating copies, merging arrays, and passing object properties.** **Ex:**	CO1	K3

	const originalArray = [1, 2, 3]; **const newArray = [...originalArray, 4, 5];** **console.log(newArray);** **// Output: [1, 2, 3, 4, 5]**		
14	Outline the methods of applying styles to the components. ☐ **Inline CSS.** ☐ **Normal CSS.** ☐ **CSS in JS.** ☐ **Styled Components.** ☐ **CSS module.** ☐ **Sass & SCSS.** ☐ **Less.** ☐ **Stylable.**	CO1	K1
15	Create a table element and render it into the root using JSX. i**mport React from 'react';** **import ReactDOM from 'react-dom';** **const table = (** **<table>** **<thead>** **<tr>** **<th>Name</th>** **<th>Age</th>** **</tr>** **</thead>** **<tbody>** **<tr>** **<td>John Doe</td>** **<td>30</td>** **</tr>** **<tr>** **<td>Jane Smith</td>** **<td>28</td>** **</tr>** **</tbody>** **</table>** **);** **ReactDOM.render(table, document.getElementById('root'));**	CO1	K3
Part -B			
21	i)Create a simple React component named "Greeting" that renders a greeting message using JSX. Render this component inside the root element of your application. **Sol:** **import React from 'react';** **import ReactDOM from 'react-dom';** **function Greeting() {** **return Hello, Welcome to My App! ;** **}**	CO1	K3

	ReactDOM.render(<Greeting />, document.getElementById('root')); ii)Create a React element using `React.createElement()` to render a simple `<p>` element with the text "Hello, React!". **Sol:** **import React from 'react';** **import ReactDOM from 'react-dom';** **const element = React.createElement('p', null, 'Hello, React!');** **ReactDOM.render(element, document.getElementById('root'));**		
22	i)Develop a "Profile" component that display user information such as name, age, bio . Render this information using JSX. sol: **import React from 'react';** **import ReactDOM from 'react-dom';** **function Profile() {** **const user = {** **name: 'John Doe',** **age: 30,** **bio: 'I love exploring new technologies and building cool things with code.'** **};** **return (** ** ** ** User Profile ** ** Name: {user.name} ** ** Age: {user.age} ** ** Bio: {user.bio} ** ** ** **);** **}** **ReactDOM.render(<Profile />, document.getElementById('root'));** ii)Create a React element that represents a `<div>` containing an `<h1>` element with the text "Welcome" and a `<p>` element with the text "This is a React app." **sol:** **import React from 'react';** **import ReactDOM from 'react-dom';** **const element = (** ** ** ** Welcome ** ** This is a React app. ** ** ** **);** **ReactDOM.render(element, document.getElementById('root'));**	CO1	K3
23	i)Design a "Product" component that display Product information such as ProductName, Price, Quantity,Description . Render this information using JSX. **sol:** **import React from 'react';** **import ReactDOM from 'react-dom';** **function Product() {** **const product = {** **productName: 'Example Product',**	CO1	K3

```
    price: 29.99,
    quantity: 10,
    description: 'This is an amazing product that will revolutionize your life!'
  };
  return (
    <div>
      <h2>Product Information</h2>
      <p>Product Name: {product.productName}</p>
      <p>Price: ${product.price}</p>
      <p>Quantity: {product.quantity}</p>
      <p>Description: {product.description}</p>
    </div>
  );
}
ReactDOM.render(<Product />, document.getElementById('root'));
```

ii)Create a React element using `React.createElement()` that represents a `<button>` element with the text "Click me" and an `onClick` event handler that alerts "Button clicked!".

Sol:

```
import React from 'react';
import ReactDOM from 'react-dom';
function handleClick() {
  alert('Button clicked!');
}
const element = React.createElement('button', { onClick: handleClick },
'Click me');
ReactDOM.render(element, document.getElementById('root'));
```