WEEK\_6

1. ReactJS-HOL

**1]Define SPA and Its Benefits**

Ans:A Single-Page Application (SPA) is a type of web application that loads a single HTML page and dynamically updates content as the user interacts with the application, without reloading the entire page.

**Benefits of SPA:**

* Faster navigation and user interaction after the initial load
* No full page reloads, which enhances user experience
* Responsive and smoother interface similar to desktop applications
* Reduced load on the server as only necessary data is exchanged
* Supports reusable component-based architecture

**2. Define React and Identify Its Working**

React is a JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications. It allows developers to create reusable UI components.

**Working of React:**

* Uses a component-based structure to build user interfaces
* Allows writing HTML-like code using JSX (JavaScript XML)
* Maintains a virtual DOM that improves performance
* Updates the real DOM only where changes have occurred, making rendering efficient

**3. Identify the Differences Between SPA and MPA**

| **Criteria** | **SPA (Single-Page Application)** | **MPA (Multi-Page Application)** |
| --- | --- | --- |
| Page Reload | No page reloads | Reloads entire page on navigation |
| Performance | Faster after initial load | Slower due to full page refresh |
| SEO Support | More difficult | Easier to optimize for search engines |
| Complexity | More complex routing logic | Simpler page-based routing |
| Examples | Gmail, Facebook | Amazon, Wikipedia |
|  |  |  |

**4. Explain Pros and Cons of Single-Page Application**

**Pros:**

* Faster and smoother user experience
* Better client-side rendering and interactivity
* Less bandwidth usage due to partial updates
* Reusable component-based code

**Cons:**

* Limited SEO optimization
* Large initial load time
* Requires JavaScript to be enabled
* May involve more complex development and routing setup

**5. Explain About React**

Ans:React is a front-end JavaScript library used for building interactive and dynamic web user interfaces. It is maintained by Facebook and a community of developers. React promotes the creation of reusable UI components and helps build single-page applications efficiently.

**6. Define Virtual DOM**

Ans:

The Virtual DOM is an in-memory representation of the real DOM elements. It is used by React to improve performance.

React creates a virtual DOM and compares it with the previous version when a change occurs. It identifies the differences (diffing) and applies only the necessary updates to the actual DOM. This makes rendering faster and more efficient.

**7. Explain Features of React**

**Ans:**

* Component-based architecture allowing code reusability
* Virtual DOM for efficient UI updates
* JSX syntax for writing HTML in JavaScript
* Unidirectional data flow for predictable data management
* Support for server-side rendering and client-side rendering
* Integration with third-party libraries like Redux and React Router
* Cross-platform development support with React Native

2. ReactJS-HOL

**1]Explain React Components**

Ans:React components are the building blocks of any React application. They allow developers to split the user interface into independent, reusable pieces that can be managed separately. Each component is a JavaScript function or class that may accept inputs called props and return React elements that describe how a section of the UI should appear.

**2]Identify the Differences Between Components and JavaScript Functions**

Ans:Although React function components are written like regular JavaScript functions, there are some key differences:

| **Aspect** | **JavaScript Function** | **React Component** |
| --- | --- | --- |
| Purpose | Executes logic or returns a value | Returns JSX to render UI |
| Return Value | Returns any JavaScript value | Returns JSX (React elements) |
| Usage | Called directly to perform logic | Used to define reusable UI elements |
| Side Effects | Generally avoids side effects | Can use React hooks like useEffect |
| Naming Convention | Any valid name | Must start with a capital letter |

**3]Identify the Types of Components**

Ans:There are two main types of React components:

1. Class Components
2. Function Components

**4]Explain Class Component**

Ans:A class component is a React component defined using an ES6 JavaScript class that extends React.Component. It can hold its own state and lifecycle methods such as componentDidMount().

**Example:**

jsx

class Welcome extends React.Component {

render() {

return <h1>Hello, {this.props.name}</h1>;

}

}

Class components are ideal when you need state management or access to lifecycle hooks

**5]Explain Function Component**

Function components are simpler and written as regular JavaScript functions. From React 16.8 onward, function components can use state and other features using hooks like useState and useEffect.

**Example:**

jsx

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

Function components are widely used today due to their simplicity and the power of React Hooks.

**6]Define Component Constructor**

The constructor in a class component is a special method used to initialize the component’s state and bind event handlers. It is called before the component is mounted.

**Example:**

jsx

constructor(props) {

super(props);

this.state = { count: 0 };

}

The super(props) call is necessary to access this.props inside the constructor.

**7]Define render() Function**

The render() function is a required method in class components. It returns the JSX that defines the UI of the component. React calls this method to determine what to display on the screen.

**Example:**

jsx

render() {

return <h1>Welcome to React</h1>;

}

The content returned by render() is used by React to update the DOM efficiently using the virtual DOM.

4.ReactJS-HOL

**1]Explain the Need and Benefits of Component Life Cycle**

In React, every component goes through a lifecycle — from its creation to updating and finally unmounting. Understanding the component lifecycle is important because it allows developers to:

* Control how and when code executes during different phases.
* Perform tasks like fetching data, initializing state, and cleaning up before a component is removed.
* Optimize performance by managing rendering and updates more efficiently.

**Benefits:**

* Helps in managing side effects (e.g., API calls, subscriptions).
* Enables resource cleanup to avoid memory leaks.
* Provides control over component behavior during mounting, updating, and unmounting phases.

**2]Identify Various Life Cycle Hook Methods**

React class components provide lifecycle methods that can be categorized into three main phases:

1. **Mounting Phase (Component is being created and inserted into the DOM):**
   * constructor()
   * static getDerivedStateFromProps()
   * render()
   * componentDidMount()
2. **Updating Phase (Component is being re-rendered due to state/prop changes):**
   * static getDerivedStateFromProps()
   * shouldComponentUpdate()
   * render()
   * getSnapshotBeforeUpdate()
   * componentDidUpdate()
3. **Unmounting Phase (Component is removed from the DOM):**
   * componentWillUnmount()
4. **Error Handling Phase:**
   * componentDidCatch()
   * getDerivedStateFromError()

**3]List the Sequence of Steps in Rendering a Component**

The typical sequence of lifecycle methods when a component is rendered (mounted) is:

1. **constructor()**  
   Initializes the component state and binds methods.
2. **getDerivedStateFromProps(props, state)**  
   Updates the state based on prop changes (rarely used).
3. **render()**  
   Returns JSX to display UI.
4. **componentDidMount()**  
   Called once after the component is mounted. Ideal for API calls or DOM manipulations.

For **updates**, the sequence is:

1. **getDerivedStateFromProps()**
2. **shouldComponentUpdate()**
3. **render()**
4. **getSnapshotBeforeUpdate()**
5. **componentDidUpdate()**

For **unmounting**, only:

* **componentWillUnmount()** is called.

5.ReactJS-HOL

**1]Understanding the Need for Styling React Components**

Styling in React is essential to improve the look and feel of the application and ensure a consistent user experience. React components are reusable, and applying styles correctly ensures that each component maintains its design standards across the application. Proper styling makes applications visually appealing, accessible, and user-friendly.

**Why styling is needed:**

* To provide a polished and professional UI.
* To maintain consistent layouts and themes across components.
* To enhance usability by improving readability, spacing, and responsiveness.
* To differentiate various sections or states (like active, disabled, error).
* To support branding through colors, fonts, and logos.

**2]Working with CSS Module and Inline Styles**

React supports multiple ways to style components. Two commonly used methods are CSS Modules and inline styles.

**CSS Modules:**

* CSS Modules help to scope styles locally to a component.
* File names typically end with .module.css.
* They prevent class name conflicts by automatically generating unique class names.
* Styles are imported into the component and used via object notation.

Example:

jsx

import styles from './Button.module.css';

function Button() {

return <button className={styles.primary}>Click Me</button>;

}

**Inline Styles:**

* Inline styles are applied directly to elements using the style attribute.
* Styles are written as JavaScript objects with camelCased properties.
* Useful for dynamic styling and quick changes.

Example:

jsx

function AlertBox() {

const style = {

color: 'white',

backgroundColor: 'red',

padding: '10px'

};

return <div style={style}>Warning!</div>;

}

Both methods can be used together based on the use case. CSS Modules are preferred for reusable components, while inline styles are effective for conditional and dynamic styling.