1. **ReactJS-HOL**

**Q1. Define SPA and its benefits**

SPA (Single-Page Application) is a web application that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from the server.

**Benefits :-**

* + Faster user experience after initial load
  + Reduces server load

**Q2. Define React and identify its working**

React is a JavaScript library developed by Facebook for building fast and interactive user interfaces for web and mobile applications.

**Working :-**

* Uses a component-based architecture where UIs are split into reusable parts.
* Maintains a virtual DOM for efficient updates.
* React compares changes using a diffing algorithm and updates only the necessary parts of the real DOM, making it fast and efficient.

**Q3. Identify the differences between SPA and MPA**

**SPA –**

* It stands for Single-Page App.
* It is faster.
* Examples – Gmail, Facebook

**MPA-**

* It stands for Multi-Page App.
* It is slower.
* Examples – Amazon, Wikipedia

**Q4 .Explain Pros & Cons of Single-Page Application**

**Pros –**

* Fast and responsive user experience.
* Reduced server load.
* Seamless navigation without page.

**Cons –**

* SEO is harder to implement.
* Initial loading time is higher.
* Can be complex to manage large SPAs.

**Q5. Explain about React**

React is a declarative, component-based JavaScript library used to build user interfaces.

**Key characteristics:**

* Developed and maintained by Facebook
* Promotes reusability through components
* Efficient DOM updates using virtual DOM
* Can be used with other libraries or frameworks

**Q6. Define virtual DOM**

Virtual DOM is a lightweight in-memory representation of the actual DOM.

**Working:**

* React keeps a copy of the real DOM (virtual DOM)
* When the UI changes, React compares the old and new virtual DOM
* Only the changed parts are updated in the real DOM

**Q7. Explain Features of React**

**Features:**

* **JSX:** JavaScript + HTML syntax for UI creation.
* **Component-Based:** Build encapsulated and reusable UI pieces.
* **Virtual DOM:** Optimized DOM manipulation.
* **Unidirectional Data Flow:** Predictable data structure.
* **Hooks:** Functional components with state and lifecycle methods.
* **Developer Tools:** React DevTools for debugging.

**Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.**

* **Creating React App**

A screenshot of a computer

AI-generated content may be incorrect.

* **Creating a React Application with the name of “myfirstreact”**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **Navigating into the folder of myfirstreact**

**A black background with white text

AI-generated content may be incorrect.**

* **VS Code - App.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **Executing the React Application**

**A screen shot of a computer

AI-generated content may be incorrect.**

* **Output – localhost:3000**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **ReactJS-HOL**

**Q1. Explain React components.**

React **components** are the building blocks of any React application. They are reusable, self-contained units of code that return a portion of the user interface (UI).

* A component can be a simple button, a form, a whole page, or even the entire application.
* Components can be either **function-based** or **class-based**.

**Q2. Identify the differences between components and JavaScript functions.**

**React Component –**

* Renders UI
* Returns JSX
* Can use lifecycle methods

**JavaScript Function –**

* Performs a task
* Returns any JS value
* No lifecycle methods

**Q3. Identify the types of components.**

There are two main types of components:

1. **Functional Components**
   * Written as JavaScript functions.
   * Can use React Hooks for state and side effects.
2. **Class Components**
   * Written as ES6 classes.
   * Use lifecycle methods for managing state and behaviour.

**Q4. Explain class component**

A **class component** is a React component defined using a JavaScript class that extends React.Component.

**Example :**

A screen shot of a computer program

AI-generated content may be incorrect.

**Q5. Explain function component**

A function component is a simpler way to write components using a JavaScript function.

**Example :**

A screen shot of a computer

AI-generated content may be incorrect.

**Q6. Define component constructor**

The constructor is a special method used only in class components. It is called when a component is created and is used to:

* Initialize state
* Bind methods

**Syntax:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Q7. Define render() function**

The render() function is a mandatory method in class components. It returns the JSX that should be displayed on the screen.

**Example :**

A screen shot of a computer

AI-generated content may be incorrect.

Create a react app for **Student Management Portal** named **StudentApp** and create a component named Home which will display the Message **“Welcome to the Home page of Student Management Portal”.** Create another component named About and display the Message **“Welcome to the About page of the Student Management Portal”.** Create a third component named Contact and display the Message **“Welcome to the Contact page of the Student Management Portal”.** Call all three components.

* **React Project named “studentapp”**

A screen shot of a computer program

AI-generated content may be incorrect.

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/Components/Home.js**

**A screen shot of a computer program

AI-generated content may be incorrect.**

* **src/Components/About.js**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **src/Components/Contact.js**

**A screen shot of a computer

AI-generated content may be incorrect.**

* **src/App.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **Executing the React Application**

**A black screen with white text

AI-generated content may be incorrect.**

* **Output – localhost:3000**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **ReactJS-HOL**

Create a react app for **Student Management Portal** named **scorecalculatorapp** and create a function component named **“CalculateScore”** which will accept Name, School, Total and goal to calculate the average score of a student and display the same.

* **React Project named “scorecalculatorapp”**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/Components/CalculateScore.js**

A screenshot of a computer program

AI-generated content may be incorrect.

* **src/App.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/Stylesheets/mystyle.css**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **Executing the React Application**

**A screen shot of a computer

AI-generated content may be incorrect.**

* **Output – localhost:3000**

**A screen shot of a computer

AI-generated content may be incorrect.**

1. **ReactJS-HOL**

**Q1. Explain the need and Benefits of component life cycle**

The component lifecycle in React represents the stages a component goes through from creation to destruction. Understanding it helps developers manage side effects, data fetching, DOM updates, and cleanup efficiently.

**Benefits :**

* Helps in fetching data right after rendering.
* Allows error handling within components.
* Ensures cleanup like removing event listeners.

**Q2. Identify various life cycle hook methods**

React Class Components have the following main lifecycle methods:

1. Mounting Phase (component is created and inserted into the DOM)
   * + constructor()
     + render()
2. Updating Phase (when props or state change)
   * + static getDerivedStateFromProps()
     + shouldComponentUpdate()
     + render()
     + getSnapshotBeforeUpdate()
     + componentDidUpdate()
3. Unmounting Phase (component is removed from DOM)
   * + componentWillUnmount()
4. Error Handling Phase
   * + componentDidCatch()
     + static getDerivedStateFromError()

**Q3. List the sequence of steps in rendering a component**

Here’s the **sequence React follows** when rendering a class component:

1. **Mounting Phase (First Render):**
   * + constructor()
     + static getDerivedStateFromProps()
     + render()
     + componentDidMount()
2. **Updating Phase (Re-render):**
   * + static getDerivedStateFromProps()
     + shouldComponentUpdate()
     + render()
     + getSnapshotBeforeUpdate()
     + componentDidUpdate()
3. **Unmounting Phase:**
   * + componentWillUnmount()
4. **Error Handling (Any Phase):**
   * + componentDidCatch()

In this application, we created a React based **blog viewer** that fetches and displays posts from a public **API** using a class component. We implemented the **componentDidMount()** lifecycle method to load data and used **componentDidCatch()** to handle any rendering errors gracefully.

* **React Project named “blogapp”**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/Post.js**

**A screen shot of a computer program

AI-generated content may be incorrect.**

* **src/Posts.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/App.js**

**A screen shot of a computer

AI-generated content may be incorrect.**

* **Executing React Application**

**A black screen with white text

AI-generated content may be incorrect.**

* **Output – localhost:3000**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **ReactJS-HOL**

We're building a **Cohort Dashboard** React app that displays information about multiple training cohorts. Each cohort is styled using a **CSS Module** and some inline styles based on the cohort status (ongoing or completed).

* **Creating React App - "cohortdashboard”**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/CohortDetails.module.css**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/CohortDetails.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **src/App.js**

**A screenshot of a computer program

AI-generated content may be incorrect.**

* **Executing the React Application**

**A black screen with white text

AI-generated content may be incorrect.**

* **Output – localhost:3000**

**A screenshot of a computer

AI-generated content may be incorrect.**