**React Assignment**

Question 1: What is React.js? How is it different from other JavaScript frameworks and libraries?

Answer 1: React is a library focused on the view layer whereas NodeJS provides a complete framework with its own set of tools and features. Reacts simplicity and ease of integration with existing projects make it a popular choice for many developers, especially for smaller projects.

Question 2: Explain the core principles of React such as the virtual DOM and component- based architecture.

Answer 2: The core principles of React.js are component-based architecture, reusability, and unidirectional data flow. React.js is a JavaScript library that allows developers to build user interfaces for web applications.

Core principles

* **Component-based architecture**: React.js uses reusable components to build user interfaces. This modular approach makes it easier to manage and maintain complex interfaces.
* **Reusability**: React.js allows developers to create modular UI elements and combine them to form complex interfaces.
* **Unidirectional data flow**: React.js uses a unidirectional data flow.

Other core concepts

* **Virtual DOM**: React.js uses a virtual DOM to optimize the rendering process.
* **JSX**: React.js uses JSX (JavaScript XML) to write HTML-like code directly within React components.
* **State and props**: React.js uses state and props to manage data within components.

Question 3: What are the advantages of using React.js in web development?

Answer 3 : React JS is a JavaScript library that can be used to create user interfaces (UI) for web applications. It has many advantages, including:

**Improved performance**

React JS uses a virtual DOM to manage changes efficiently, which reduces load times and improves user experience.

**Reusable components**

React JS allows developers to create reusable UI elements, which can be used to build complex user interfaces.

**SEO-friendly**

React JS can help make web applications more discoverable and indexable by search engines.

**Easy maintenance**

React JS makes it easy to maintain and update web applications.

**Strong community support**

React JS has a large and active community of developers who can provide support and guidance.

**Fast development**

React JS is fast and flexible, which can help developers create applications quickly.

**Declarative**

React JS is a declarative language, which means it's easy to read and understand.

**Simple**

React JS is simple to use and learn, making it a good choice for developers of all skill levels. React JS is especially useful for building single-page applications and interactive websites.

**JSX• (JavaScript Task 1: XML)**

Question 1: What is JSX in React.js? Why is it used?

Answer 1: JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

Question 2: How is JSX different from regular JavaScript? Can you write JavaScript inside

Answer 2: JSX lets you write HTML-like markup inside a JavaScript file, keeping rendering logic and content in the same place. Sometimes you will want to add a little JavaScript logic or reference a dynamic property inside that markup.

Question 3: Discuss the importance of using curly braces {} in JSX expressions.

Answer 3: Curly braces in JSX are the syntactic sugar that sweetens the development process. The indicators tell React to interpret the enclosed content as JavaScript, not plain text. This allows developers to embed variables, function calls, and other JavaScript expressions directly within their JSX code.

**Components(Functional &Class Components)**

Question 1: What are components in React? Explain the difference between functional component and class component

Answer 1: In React, components are isolated units that return HTML and are similar to JavaScript functions. There are two types of components in React: functional components and class components.

Functional components do not have a state or lifecycle methods. Class components have a state and can implement lifecycle methods like componentDidMount and componentDidUpdate. Faster as they do not have state and lifecycle, react needs to do less work to render these components.

Question 2: How do you pass data to a component using props?

Answer 2: To pass props, add them to the JSX, just like you would with HTML attributes. To read props, use the function Avatar({ person, size }) DE structuring syntax. You can specify a default value like size = 100 , which is used for missing and undefined props.

Question 3: What is the role of render() in class components?Answer 3: Render in React JS is a fundamental part of class components. It is used to display the component on the UI returned as HTML or JSX components. The React DOM. render() function takes two arguments, HTML code and an HTML element.

Prop and state

Question 1: What are props in React.js? How are props different from state?

Answer 1: Props are used to pass data from a parent component to a child component, while state is used to manage data within a component. Props are immutable and cannot be changed within a component, while state is mutable and can be updated using the set State function.

Question 2: Explain the concept of state in React and how it is used to manage component

Answer 2: State is a fundamental concept in ReactJS that plays an important role in building interactive and dynamic web applications. In this article, we will dive deep into **what state is in ReactJS**, how to use it and explore advanced techniques for managing state in large applications.

Question 3: Why is this.setState() used in class components, and how doesit work?

Answer 2: set State is a method used in React class components to update the component’s state that triggers a re-render with the updated values. All the React components can have a **state** associated with them. The state of a component can change either due to a response to an action performed by the user or an event triggered by the system.We use the **set State() method** to change the state object. It ensures that the component has been updated and calls for re-rendering of the component.