

# **ReactJS Assignment**

## **1. What is ReactJS and Why is it Used?**

**ReactJS is a JavaScript library developed by Facebook for building user interfaces, especially for single-page applications where data changes over time. It allows developers to create large web applications that can update and render efficiently in response to data changes.**

**Why it is used:**

- Fast performance with Virtual DOM**
- Reusable components**
- Easy to integrate with other frameworks**
- Enhances user experience with dynamic content**
- Popular for building modern, responsive web apps**

## **2. What are the Main Features of ReactJS?**

- JSX (JavaScript XML): A syntax extension that lets you write HTML-like code inside JavaScript.**
- Virtual DOM: React maintains a virtual DOM that improves performance by minimizing actual DOM manipulation.**
- Component-Based Architecture: Build encapsulated components that manage their own state.**
- One-Way Data Binding: Data flows in a single direction, making the app more predictable and easier to debug.**
- Declarative UI: Makes code more readable and easier to debug by describing what UI should look like for each state.**
- React Hooks: Allow the use of state and other React features without writing a**

**class.**

### **3. What is the Meaning of the Term SPA?**

**SPA (Single Page Application) is a web application that loads a single HTML page and dynamically updates content without refreshing the page. It improves performance and user experience by reducing page load times and server requests.**

### **4. What are React Components and Their Types?**

**React components are the building blocks of a React application. They define how a part of the UI should appear and behave.**

**There are two main types of components:**

#### **1. Functional Components:**

- JavaScript functions**
- Use hooks for state and side effects**
- Example:**

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

#### **2. Class Components:**

- ES6 classes that extend React.Component**
- Have lifecycle methods**
- Example:**

```
class Welcome extends React.Component {
```

```
render() {  
  return <h1>Hello, {this.props.name}</h1>;  
}  
}
```