**Explain React components**

**React components** are like reusable building blocks of a website. Each component makes a part of the user interface, such as a button, header, or card—just like LEGO bricks come together to make something big.

**Differences between components and JavaScript functions**

* **React components**: Special code made to display part of a web page, and they usually return **JSX** (which looks like HTML).
* **JavaScript functions**: Can do anything—calculations, logic, or anything else in programming—but don’t return HTML by default.

**Types of React components**

1. **Class Components** – Made with JavaScript classes; can have more features and older style.
2. **Function Components** – Made with regular functions; simpler, more modern, and used the most today.

**Explain class component**

A **class component** is made using a JavaScript class (with class). It can hold its own data (state) and use special functions called lifecycle methods. Example:

jsx

**import** React, { Component } **from** 'react';

**class** Hello **extends** Component {

render() {

**return** <h1>Hello!</h1>;

}

}

**Explain function component**

A **function component** is just a function that returns JSX. It’s shorter, easier to write, and is the recommended way in React today. Example:

jsx

**function** Hello() {

**return** <h1>Hello!</h1>;

}

**Define component constructor**

A **constructor** is a special function inside a class component. It runs once when the component is created and sets up starter data (state). Example:

jsx

constructor() {

**super**();

**this**.state = { name: 'Student' };

}

**Define render() function**

The **render()** function is part of a class component. It decides what gets shown on the page by returning JSX (looks like HTML). Without a render() function, nothing appears onscreen. Example:

jsx

render() {

**return** <h1>Hello!</h1>;

}