Industrial Training Daily Diary

MERN Stack Training

Company: Sensation Software Solutions
Student Name: Tajinder Kaur
Training Duration: 6 Months
Day: 11

Day 11 – React Components, JSX in Depth & Rendering

Objective of the Day

The objective of Day 11 was to gain a deeper understanding of **React components**, learn **JSX in detail**, and understand how components are rendered in a React application. This session focused on the core building blocks of React and how user interfaces are created using reusable components.

Understanding React Components

The trainer explained that **components** are the heart of any React application. A component is a reusable piece of UI that returns JSX and can accept data using props.

Key characteristics of React components:

- Reusable and modular
- Independent and maintainable
- Easy to test and debug

We learned that large applications are divided into smaller components to improve code organization.

Types of React Components

Functional Components

The trainer explained **functional components**, which are simple JavaScript functions that return JSX.

Advantages of functional components:

- Simple and readable syntax
- Better performance
- Widely used with React Hooks

Class Based Components (Overview)

We were briefly introduced to class-based components and learned that although they are still supported, functional components are preferred in modern React development.

JSX in Depth

We studied **JSX** in more detail and learned the following concepts:

- JSX expressions using `{}`
- Writing JavaScript logic inside JSX
- Conditional rendering using ternary operators
- Rendering lists using `map()`
- Importance of keys in lists

The trainer explained how JSX is transpiled into `React.createElement()` calls behind the scenes.

Rendering Components in React

The trainer explained how React renders components to the browser using:

- `ReactDOM.createRoot()`
- `root.render(<App />)`

We learned the role of the **App component** as the root component of a React application.

Components vs Elements

The difference between **components** and **elements** was explained:

- Components are functions or classes
- Elements are objects returned by components
- Elements are immutable

This helped clarify how React builds the UI internally.

Project Structure Understanding

We explored the React project folder structure:

- `src` folder
- `index.js`
- `App.js`
- `public` folder

Understanding the project structure helped in navigating and organizing React applications efficiently.

Practical Work Done

During the practical session, we:

- Created multiple functional components
- Rendered components inside App component
- Used JSX expressions and conditions
- Displayed lists dynamically using `map()`

This hands-on practice strengthened React basics.

Importance of Components in MERN Stack

The trainer explained that React components are responsible for displaying data fetched from backend APIs. Well-structured components make applications scalable and easier to maintain.

Conclusion of Day 11

Day 11 provided a strong foundation in React component-based development. I learned how JSX works internally and how React renders UI efficiently using components.

Outcome of the Day:

- Clear understanding of React components
- In-depth knowledge of JSX
- Ability to render components
- Confidence in