

Training Day 1 Report

Date: 25 June 2025

Topic Covered: Introduction to React.js

Today marked the first day of MERN stack training, focused on **React.js** — the frontend library used to build user interfaces.

Since HTML, CSS, and JavaScript are already known, the day was dedicated to understanding React's core concepts: **components**, **props**, and **JSX**.

What is React?

React is a JavaScript library developed by Facebook for building user interfaces in a *declarative* and *component-based* way.

Instead of manipulating the DOM directly, React allows developers to describe how the UI should look and handles the rendering efficiently using a **virtual DOM**.

Understanding Components

Components are the **building blocks** of a React application. They are reusable, encapsulated pieces of UI logic.

There are two main types:

- **Functional components** (preferred using Hooks)
- **Class components** (legacy but still useful)

Each component returns JSX, which represents the structure of the UI.

What is JSX?

JSX (JavaScript XML) allows writing *HTML-like code inside JavaScript*. It helps describe UI in a readable and declarative way.

JSX is not valid JavaScript by default — it is transpiled using **Babel**.

Example:

```
<h1>Hello , React !</h1>
```

This is converted into `React.createElement()` under the hood.

Props in React

Props (short for properties) are how components receive data from a parent component.

They are:

- **Read-only**
- Help make components dynamic and reusable

Example:

```
<Welcome name="Ananya" />
```

Here, `name="Ananya"` is passed as a prop to the `Welcome` component.

Outcome of the Day

- Understood how React **components** work and how to pass data using **props**.
- Learned **JSX** syntax and how it compiles into `React.createElement()`.

- Set up a basic **React app using Vite** and created sample components to practice.