**React**

**Day 3rd**

**What is react???**

React is a JavaScript library for building interactive user interface. It uses virtual DOM make application fast. It is a small library and it is following unidirectional data flow. And it has feature called data binding components routing.

ReactJS is a JS library used for building reusable UI components according to react official documentation following is the definition-

[React](https://www.react.dev) is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use react as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using node, and it can power native apps using React Native. React implements one way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

It supports component based architecture and it is **declarative** sometimes it is imperative but in a DOM way.

**Imperative**

Imperative programming is when you write code that explicitly describes how the UI should change step by step, by manually manipulating the DOM or component state.

Imperative code is often found when you’re using:

* Document.querySelector, document.createElement
* Direct DOM manipulation
* React refs (sometimes), or libraries that modify the DOM outside of React’s control

📝 You tell the browser exactly how to do everything.

**Declarative**

React makes it painless to create interactive UIs. Design simple view for each state in your application, and react will efficiently update and render the right components when your data changes. Declarative view make your code more predictable and easier to debug.

Declarative means you need to declare something and behind the scene execution will done by any other thing.

Declarative programming is when you describe what the UI should look like for a given state, and react handles the process of rendering and updating it efficiently

In react you define the components and what they should render based on the data (props/state). You don’t worry about how react updates the DOM – React handles that under the hood.

**Component – based**

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

**💀 Pre-requisites for learning ReactJS**

Basic understanding of HTML, CSS and JavaScript.

Basic understanding of ES6 features like **Let, Conts, Arrow functions, Modules, export and import, classes, spread and rest parameter.**

Basic understanding of **WebPack, Babel and NPM**