React is declarative (not imperative) framework, in fact its not a framework, it’s a library and it handle all the DOM manipulations

imperative focuses on how and declarative focuses on what. In a software engineering context, declarative programming means writing code to describe what the program should do. One describes what needs to happen; the minutiae for making it so are left to the system.

In contrast, imperative programming involves writing code that follows explicit steps to solving a problem, completing a task, or achieving a desired result. It’s telling the system specifically how to do something with the expectation that the desired outcome will result.

Web API is kind of imperative, you need to write code follows explicit steps to manipulate DOM, people on Facebook make this library (React) to handle all those DOM manipulations to get ride off tedious repetition in our workflow under the hood (declarative)

\*Composability its another important feature of React -> easy to change and swap in/out the web building blocks (components) -> prevent the code become very intertwined

(data as array of object 🡪 using .map🡪to React component

\*Jest -> test tools in order to 1- document your line of code to prevent other developer break your feature in future! 2- make new code resilience to all other part of project at the time of integration

React components🡪 always first letter would be capital (upper camel case – pascal casing)

Props would be attributes of those components

(Components can store the states)

Event occurs 🡪 React reacts to that change 🡪state get updated🡪change will go through the props and components

Babel🡪 JSX (using babel library to bring the html to JavaScript and switch to JS when {}