**Week 5**

**5.1 React Foundations**

* React compiler takes care of converting React to HTML, CSS, and JS.
* 3 things 🡪 components, state, re-rendering.
* We have to write state and components part only.
* Anything on the website that changes is called the state. You have to keep the things that would change in the state.
* Components help decide how to render the website given a specific state.
* When the DOM actually changes, that is re-rendering.
* Component takes some state and spits out some HTML.
* You have to do the first render.
* Array destructuring is used in defining state.
* 0 is the initial value in useState(0).
* .jsx is just a JS file in which you can write both JS and XML.
* npm run build converts our react project into HTML, CSS, and JS.
* React project has idex.html, main.jsx 🡪 app.jsx 🡪 hi there.
* Whenever we want to put any JS code in a React project, we put it inside curly braces.
* … spreads the array so that you can add values to the array.
* Any time a parent re-renders, it’s child re-renders as well.
* If you want to put styles, you would have to use two curly braces. One for JS, one for object.
* React and Vue are similar.
* Vite is a bundler.
* Bootstrap means giving you an initial code.
* If state variable changes, App would re-render and any time App re-renders, all of the children components also re-render.
* If your buttons need to have separate counts, you need to define the states inside the component.
* Curly braces inside xml is not for writing code only for running expressions.
* React can only change root.
* setCount is asynchronous, useState is not asynchronous.

**5.2 React Project - Todo App**

* First build the backend.
* If you are creating node JS backend, you need to initialize it and it must have package.json.
* npm init for initialising
* npm install installs all the node\_modules that are listed in dependencies.
* DB server information is stored in a .env file and not pushed onto GitHub.
* Components are always start with a capital letter.
* Cors library lets your frontend hit your backend. Any frontend can hit your backend now. Slightly insecure.
* useEffect hook solves the problem of infinite requests.
* You should never props from a child to a parent.
* Local state variable causes a lot of useless re-renders.
* Vercel for free frontend deployment. Render Network for free express deployment.