**Week4**

**4.1 DOM Introduction**

* DOM helps to make the website dynamic. Dynamic means that the website changes when the page has been loaded.
* After 2009, Reach came into the picture and made it easy for website to be dynamic.
* Classes are created with .
* You should not have same ids for separate elements.
* Classes for CSS, Ids for JS.
* To make the website you need DOM and JS.
* Script tag lets you write JS in an HTML file.
* # means something that is an Id. Usage is similar to dot for classes.
* document.querySelectorAll(“div”) gives back an array of all the divs.
* When sending response back to frontend, you can use .send instead of .json.
* Axios is much better than fetch.
* Debouncing is when you don’t send the request immediately 🡪 standard interview question. Debouncers are created using clearTimeout and declare a timer variable outside the function which is cleared each time, the clearTimeout function is called.
* Throttling is similar. When we get many request, the backend can throttle them. But the difference is that the backend always respond to them.
* Debouncing happens on the frontend while throttling happens on the backend.
* If the request is POST, then fetch takes a 3rd argument as json which has method : POST.
* The id attribute and onclick expects a string. The onclick needs a function call, not a function signature.

**4.2 React Foundation**

* With ` (backtick), you can write multiple line strings.
* In React, you have a state which is basically a list, a list of strings, an array of objects etc.
* Server returns a state, the heavy lifting of putting the data in the DOM is what React does.
* React asks you for a state and updates your DOM completely.
* React takes a state and some logic to let you put data on the screen as you want.
* First clearing the HTML and then adding to it is bad (when we’re not using React).
* The better way is to calculate the diff, and if there is a difference then update that specific element in DOM. This what React does.
* Frontend Developer – ReactDOM – React.
* React is a difference calculating framework.
* Any variable that React needs to consciously watch is called state variable.
* When we deploy, it’s just HTML, CSS, and JS. However, in development, it is helpful to have a backend server which does hot reloads for React. It uses the native http server.
* React onClick expects a function signature and not function calling like in HTML.
* Redux, recoil further helps in React.
* Express is a library, React is a framework.
* To create React project: npm create vite@latest 🡪 React 🡪 JS 🡪 npm install 🡪 npm run dev.

**4.3 Mongo Deep Dive**

* Granual access means database provides only your data. Only firebase does this, so browsers/mobile apps can directly talk to databases.
* Mongoose lets you define some schema.
* References point to another table that exists in MongoDB.
* If there’s an error in the async function and if you have used async await syntax, then the control stops at the line of async call.
* Bearer defines the type of token that you are sending to the backend. API-Key, Passwords are some other kinds of tokens. We could work without this, but it is standard practice.
* Irrespective of the case in the header, it will get converted to lower case.
* Middlewares end the request, forward the request, or pass data along to the next middleware.