React (Day-3)

→ Name our components with .js or .jsx extention?

* Doesn't matter! Till now, I personally prefer writting .js extentions. And it's better to adapt based on the project, your team mates and team leads. It hardly matters!

→ Folder/File structure?

Industry convention :-

Root/
:
src/
Header.js
(all components)
(or)
components/
↳ all components in components folder which is inside src.

We have diff. structures.

1) Feature based / Route based structuring

2) Component based structuring.

* Again, at the end you can even write your react code in a single file. It's going to be bundled into single file using parcel/babel.

* But since we are going to work on projects with other people. It is better to follow some principles while writing the code,

Like modularity. (diff components in different files).

* At the same time, too much folder nesting is also not suggested. Just find the suitable structure and work with it.

→ How to structure data and URLs in our project?

* : It's very bad way to include static/mock data inside our component files.

* Component file code should always be maintained neat, slim and lean.

→ So we can create a separate folder to store our static data / URL constants / API keys, etc.

→ create a folder - utils (same heirarchy as components Src)

Diff. ppl. name it diff ways.

common, config, utils, etc. we will follow utils folder naming.

And you can create data.js for data

constants.js for urls, etc.

Export & Import

Export & Import = Two types each.

Default Export & Import : export default CompName;

import CompName from "path";

Named Export & Import : export const name = --- (Just put export infront of variable)

import {name} from "path";

Named export is used when we want to export multiple entities from single file.

Default export can be only one in a file.

* while importing, other files need an extention in the path, whereas .js files don't need.

Ex: import "./index.css";
import Componame from "./../components/Home";

here we need not put Home.js though it will work with Home.js too.

React Hooks

→ These are normal JS utility functions written by facebook developers written inside react.

→ They are just function with some useful logic inside them.

Two imp React hooks :-

* useState() - superpowerful state variables in react
* useEffect()

useState variable binds & keeps the UI & data layer in sync.

As soon as the state of a component changes, the state associated with the component rerenders.

Reconciliation algorithm (React Fiber)

Virtual DOM → virtual representation of UI is kept in memory.

Diff algorithm → Checks for the differences/updates in virtual dom with Actual dom and only updates the required updates in the actual dom on every render cycle.

This is known as reconciliation or React Fibre concept.

→ why do we have setState() function when we can directly update the state variable?

✱ Whenever the setState() is called to modify the state, a trigger is made and the render cycle is processed.

→ What makes React fast?

✱ The ability of React to make fast DOM manipulations, using the concepts of virtual DOM, Reconciliation & React Fiber.