* Reactjs (it is a library of JavaScript)
* Working on API’s
* **We will build our own fake API within reactjs project**
* **We we can get the data from external API file which is still present in our reactjs project**
* **How to fetch the data from any live API**
* **How we can make our own API and put the data into that API using Postman and finally fetch the data from it.**
* There are 2 types of components in reactjs
* Class components
* Functional components
* What are the functions?

Functions are the block of codes that are re-usable

* All the hooks start from the word **use**
* useState() this hook is responsible for handling the component state/behavior
* Syntax of useState()

const [variable1, variable2] = useState(initial\_value);

variable1 (it holds all the data of the component)

variable2 (it acts like a function that perform the actions for the component)

* useEffect( ) this is the hook that is used for fetching the data immediately after the component renders
* If we want to use bootstrap in reactjs project, then there are 2 methods for it
* Include CDN of bootstrap
* Install bootstrap using npm package
* How to pass the data from one component to another component
* Just pass the variable to that component tag in which we want to show our data
* Props
* Syntax of map

variableName.map((newVariable) => (

code….

))

* There are 2 methods to get the data from API in Reactjs
* Using fetch() this method register our API
* Using Axios (npm package)
* Syntax of useEffect

useEffect(() => {

code….

},[])

* then{

//we get all the response/data of API

}catch(){

}

Steps of getting the data from any online API:

1. Make a component
2. Get useEffect() into it
3. Register the online API using **fetch()** method
4. After registering the API, get the response/data of API using **then** statement
5. Pass the API data to useState hook. Because for mapping the data, we must have to pass the data to useState hook
6. Type the statement **catch** just for getting the error if any.
7. Finally map the data using the 1st variable of useState hook