Namaste React Assignment5:

1. What is the difference between named export, default export and \* as export?

Ans: With named export we can export multiple values or functionalities from a single file and while importing them we have to use the {}. With default export we can export only one value or functionality from a file.

With named exports the name of the import and export must be same and with default export the name of the import and export can be different.

Example of named export:

App.js:

export const name=”Kalyan”;

export const Body = () => {

return (

<div>

<h1>Kalyan </h1>

</div>

)

}

Importing them in Header.js:

Import {name, Body} from App.js;

Example of default export:

App.js:

const Body= () =>{

return(

<div>

<h1>Kalyan</h1>

</div>

)

}

export default Body;

Importing in Haeder.js:

Import Body from App.js;

* Export Everything (\* as): The \* as syntax allows you to import all the exports from a module under a single namespace. This is very useful when a module has so many exports.

Example:

Calc.js:

export const AddTwo = (a,b) => a+b;

export const MulTwo = (a,b) => a\*b;

App.js:

import \* as Calculator from Calc.js;

console.log(Calculator.AddTwo(6+3));

console.log(Calculator.MulTwo(4+5));

1. What is the importance of config.js file?

Ans: config.js file allows developers to configure the values instead of hardcoding directly inside the components. Whenever we have some common values we can define them here and we can import this file in the required component and use the values.

1. What are React hooks?

Ans: They are just normal JavaScript utility functions. Before hooks was introduced the concept of state and React life cycle methods are only available in the class components, React hooks was introduced in the React16, with the help of hooks we can have state and React life cycle methods functionalities in functional components also.

1. Why do we need a useState() hook?

Ans: useState() hook is used to manage the state of a component in the functional components. If we create a state variable using useState() hook then when the value of the variable is changed then that component will be re-rendered, so when we want the UI to be in sync with the data then we need to use the useState() hook instead of normal JavaScript variables.