1. **What is the difference between Named Export, Default export and \* as export?**

*Named Export:* There are multiple named exports in a file. If we want to import a specific export we surround it with {} braces and import it from one file to another, this called named import.

Example: Components are in MyComponent.js file

export const MyComponent = () => {}

export const MyComponent2 = () => {}

And these components are imported like this.

// ex. Importing a single named comment

import { MyComponent } from “./MyComponent”;

// ex. Importing multiple single named comment

import { MyComponent, MyComponent2 } from “./MyComponent”

// ex. Giving a named import a diff name

Import { MyComponent2 as MyNewComponent } from “./MyComponent”

*Default Export:* In default export there is a only one default export per file. The component is exported like this:

const MyComponent = () => {}

export default MyComponent

And the component is imported like this: here we must omit {}

import MyComponent from "./MyComponent";

*\*as export:* It is used to import the whole module as a component. The component are exported like this:

export const MyComponent = () => {}

export const MyComponent2 = () => {}

export const MyComponent3 = () => {}

The components are imported like this:

import \* as MainComponents from “./MyComponent”;

We can use them in JSX as

<MainComponent.MyComponent />

<MainComponent.MyComponent2 />

<MainComponent.MyComponent3 />

We can use Named Export and Default Export together also

export const MyComponent2 = () => {}

const MyComponent = () => {}

export default MyComponent;

import MyComponent, { MyComponent2 } from “./MyComponent”

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

The config file is used to store configuration data of the application. It also contains API endpoints, database connection strings and other sensitive information. By keeping the data in separate file it is easier to update as well as access it throughout the project.

1. **What is State variable? Need of it? What is Hooks? Why do we need useState() hook ?**

A state variable is a special variable in react that is used to store data that changes over time. When the state of the component changes, it automatically gets rendered on the UI. To create a State variable we use Hooks. Hooks are just normal javascript functions. We use a useState() hook to create a local state variable inside functional component. It return an array with 2 values [current state, function to update the current state]. **Never** use a useState() outside functional component.