🪝

Assignment 05 - Let’s get Hooked

Owner	Pankaj Kumar

1: What is the difference between Named export , Default export and as export?

- **Named** export - In named export, the function is exported as:

  ```
  export <function>;
  ```

  and imported as:

  ```
  import { <function> } from '..';
  ```

  . The function is exportde inside `{}` to other modules.

- **Default** export: In default export the function is exported as:

  ```
  export default <function>
  ```

  and imported as:

  ```
  import <function> from '...'
  ```

- **as** export: * as export is used to import the whole module as a component and access the components inside the module.

for example: We have a module ABC.js and some components inide of it and a file named XYZ.js where we want to import the components.

ABC.js:

```
export const Comp1 = () => {...}
export const Comp2 = () => {...}
export const Comp3 = () => {...}
```

in **XYZ.js** we'll import then as

```
import * as  Module from 'ABC.js'
```

Now we can use them JSX as:

```
<Module.Comp1 />
<Module.Comp2 />
<Module.Comp3 />
```

2: What is the importance of config.js file?

- `config.js` file is used for the hard coded values used in our application. We can use it import a configration in side any component without having to copy it over and over again.

3: What are React Hooks?

- React Hooks are basically JavaScript functions which are provided by React. Hooks have some special capabilities that are usefull for the development. like managing state, memory etc.

▼ **React provides a bunch of standard in-built hooks:**

  - **useState:** To manage states. Returns a stateful value and an updater function to update it.
  - **useEffect:** To manage side-effects like API calls, subscriptions, timers, mutations, and more.

- **useContext:** To return the current value for a context.
- **useReducer:** A useState alternative to help with complex state management.
- **useCallback:** It returns a memorized version of a callback to help a child component not re-render unnecessarily.
- **useMemo:** It returns a memoized value that helps in performance optimizations.
- **useRef:** It returns a ref object with a current property. The ref object is mutable. It is mainly used to access a child component imperatively.
- **useLayoutEffect:** It fires at the end of all DOM mutations. It's best to use useEffect as much as possible over this one as the useLayoutEffect fires synchronously.
- **useDebugValue:** Helps to display a label in React DevTools for custom hooks.

4: Why do we need useState Hook?

- **useState** hook is used to maintain the state in our React application. It keeps track of the state changes. When a component is re-rendered it changes the state of our compoent.