* Why React :- Makes easy to manage complex front-end
* React is a ***Library***
* When we put our app in production then the build is served to the user an not the source folder
* Manifest.json = this file is used for mobile application
* React is the core Library while React-Dom is an implementation of React on web
* Whenver working with vite, always write name of the components as jsx and the first letter should be ***uppercase*** and not js.
* If you want to export multiple tags but don’t want to use the div tag, then return it using

<></>

* React has a special function called react.createElement

Eg: const ReactElement = React.createElement(

  'a',

  { href: "https://www.github.com", target: "\_Blank" },

  "Click me to fly"

)

* In React ‘ {} ‘ means an ***Evaluated Expression*** remember for interview, evaluated expression means the variable inside holds the final value of any operation. Hence if-else statements cannot be implied inside the {} expression
* In React software, the UI updation is controlled by React or more specifically React Hooks rather then user himself
* All the under the hood functions are performed by Babble hence importing React is not relevant
* React Props can only take reference to objects and not the objects themselves.
* For props you can use like this :-

btnValue = {props.btnValue || “Fuck this shit I’m out”}

* When we write like this :-

onClick = {setValue( “Habibi” )};

You are giving a return value to the onClick instead of a function reference and return value could be anything from string to array, hence it is suitable to use an arrow function or function reference

* useCallback :- if from the given set of array of dependencies, if anny dependency or method changes then optimise the code. Hence we talk about ***Optimization.***

***It basically memorizes the function, either in part or whole*** .

* useEffect :- If any dependencies from the given array of dependencies changes then rerun the given function
* Because we are locally creating on React, we can use ***window*** option, which is not possible in next.js because of server side rendering
* useRef :- give more feel to user by selecting the referenced part
* Whenever using hooks always keep the file name as ***JS*** and not JSX because the hooks file always return javascript and not jsx
* Standard approach, always use the keyword “use” whenever writing the hooks
* If you want user to pass additional css along with the given css then use

`bg-white p-3 rounded-lg text-sm flex ${className}`

* To check if a function reference or it’s value exist or not, we can use

(e) => onAmountChange && onAmountChange(Number(e.target.value))

This is a checker which checks whether the “onAmountChange” exists or not

* Number(e.target.value) :- In javascript, the event function always returns string while in this particular case
* Whenever iterating through the array using map, React often ignores repeatitive elements, Hence it’s necessary to assign a key to each iteration.
* For choosing a key, often choose ids if doing directly with database or else choose the field incoming.
* useId :- A React Hook for generating unique IDs that can be passed to accessibility attributes.