Contents

[2. React and JSX Fundamentals 2](#_Toc172296183)

[1. Create a react app using Vite: 2](#_Toc172296184)

[2. If-else statement in JSX using ternary Operator 3](#_Toc172296185)

[3. Anonymous Function 4](#_Toc172296186)

[4. JSX For-Loop 5](#_Toc172296187)

[5. JSX Conditional Rendering Using IF-Else: 6](#_Toc172296188)

[6. JSX Conditional Rendering Using Switch Statement: 7](#_Toc172296189)

[7. JSX Conditional Rendering Using && Operator: 7](#_Toc172296190)

[8. Passing Properties to Child Component: 8](#_Toc172296191)

[Passing a String 8](#_Toc172296192)

[Passing an Object: 9](#_Toc172296193)

[Passing a Function: 11](#_Toc172296194)

[9. Managing C lick Event: 13](#_Toc172296195)

[Wrong Way: 13](#_Toc172296196)

[Correct Way By Using Arrow Function: 14](#_Toc172296197)

[Correct Way By Using Regular Function: 14](#_Toc172296198)

[10. File structure of this chapter: 15](#_Toc172296199)

[11. App.jsx of this chapter: 16](#_Toc172296200)

[12. Main.jsx of this chapter: 16](#_Toc172296201)

[13. Output Result of this chapter: 17](#_Toc172296202)

[3. React Hook and State Manager 18](#_Toc172296203)

# React and JSX Fundamentals

## Create a react app using Vite:

My node version on the Ubuntu WSL was outdated so I ran this command to install the latest long term support version of it:

**

In order to create a Vite project this command should be used and the Vite should be followed:



In order not having to install npm packages every time for each project and using the computer hard disk unnecessary und inefficient, pnpm instead will be used. Pnpm uses hard links and symlinks to save one version of a module only ever once on a disk. for installing it, this command have been used:



After that, instead of npm install, pnpm install will be used.

For running the program using pnpm, the following command should be used:



For running the program using npm, the following command should be used:



Project can also be run by using the Vite command:



## If-else statement in JSX using ternary Operator



## Anonymous Function



## JSX For-Loop

Using map function and then calling an anonymous function inside the map-function:

* Don’t forgot to add ‘return’ to the map function!



## JSX Conditional Rendering Using IF-Else:

In this code, more efficient and cleaner way, of implementing If-Else by using functions is illustrated: 

## JSX Conditional Rendering Using Switch Statement:



## JSX Conditional Rendering Using && Operator:

If condition is true, it will execute the code after the ‘&&’ operator, otherwise it won’t execute something else.



## Passing Properties to Child Component:

It has a unidirectional flow, so you cannot pass components from child to parent.

### Passing a String

Like html we will use Attributes:

***Parent Component:***



***Child Component:***



### Passing an Object:

***Parent Component:***



***Child Component:*** 

### Passing a Function:

***Parent Component:***



***Child Component:*** 

## Managing C lick Event:

### Wrong Way:

If you implement a function in this way, as illustrated in the code below, the browser will constantly keep running this function whenever the user refreshes the page instead of running it only when the button is clicked.

***App Component:***



***Button function in ‘Managing\_Click\_Event’ Component:*** 

### Correct Way By Using Arrow Function:

***Button function in ‘Managing\_Click\_Event’ Component:*** 

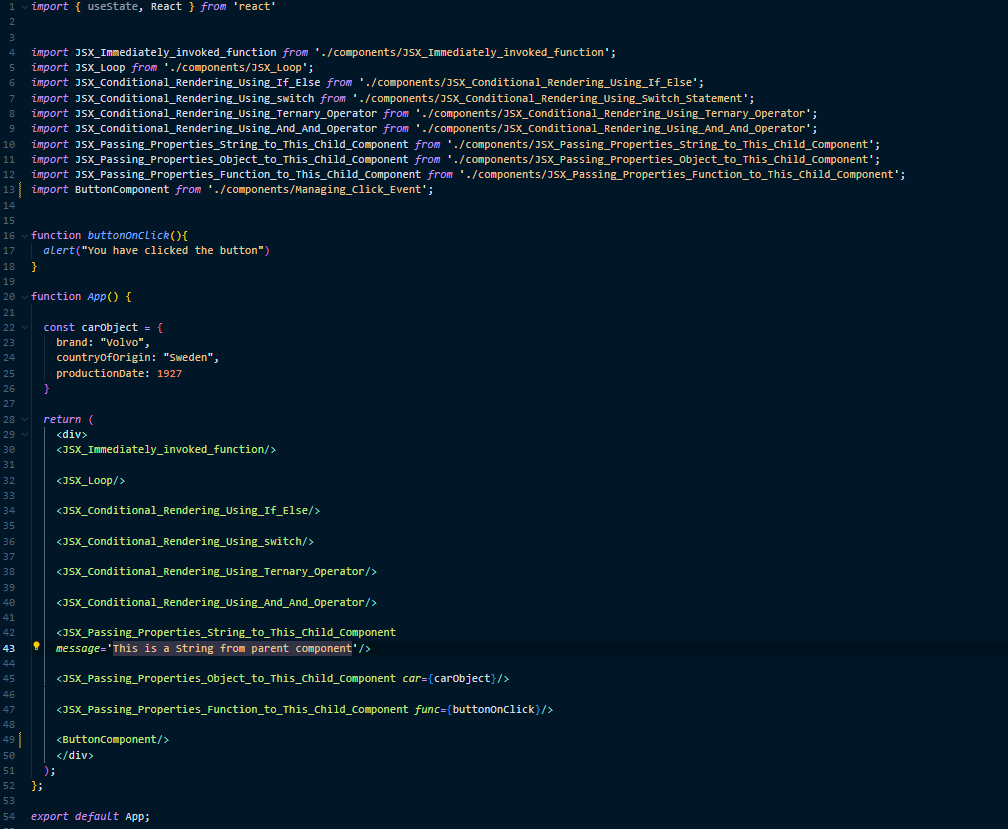
### Correct Way By Using Regular Function:



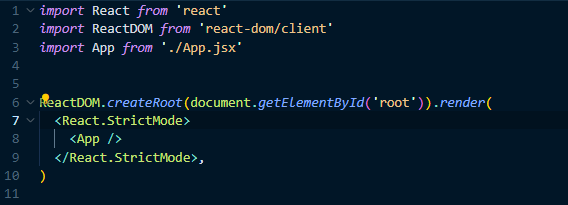
## File structure of this chapter:



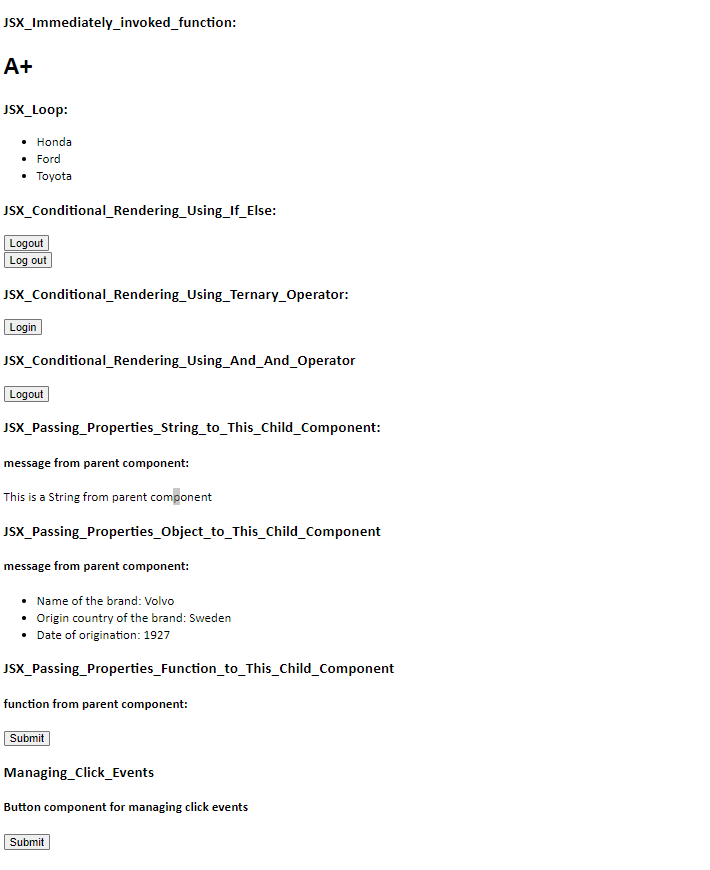
## App.jsx of this chapter:



## Main.jsx of this chapter:



## Output Result of this chapter:



# React Hook and State Manager: