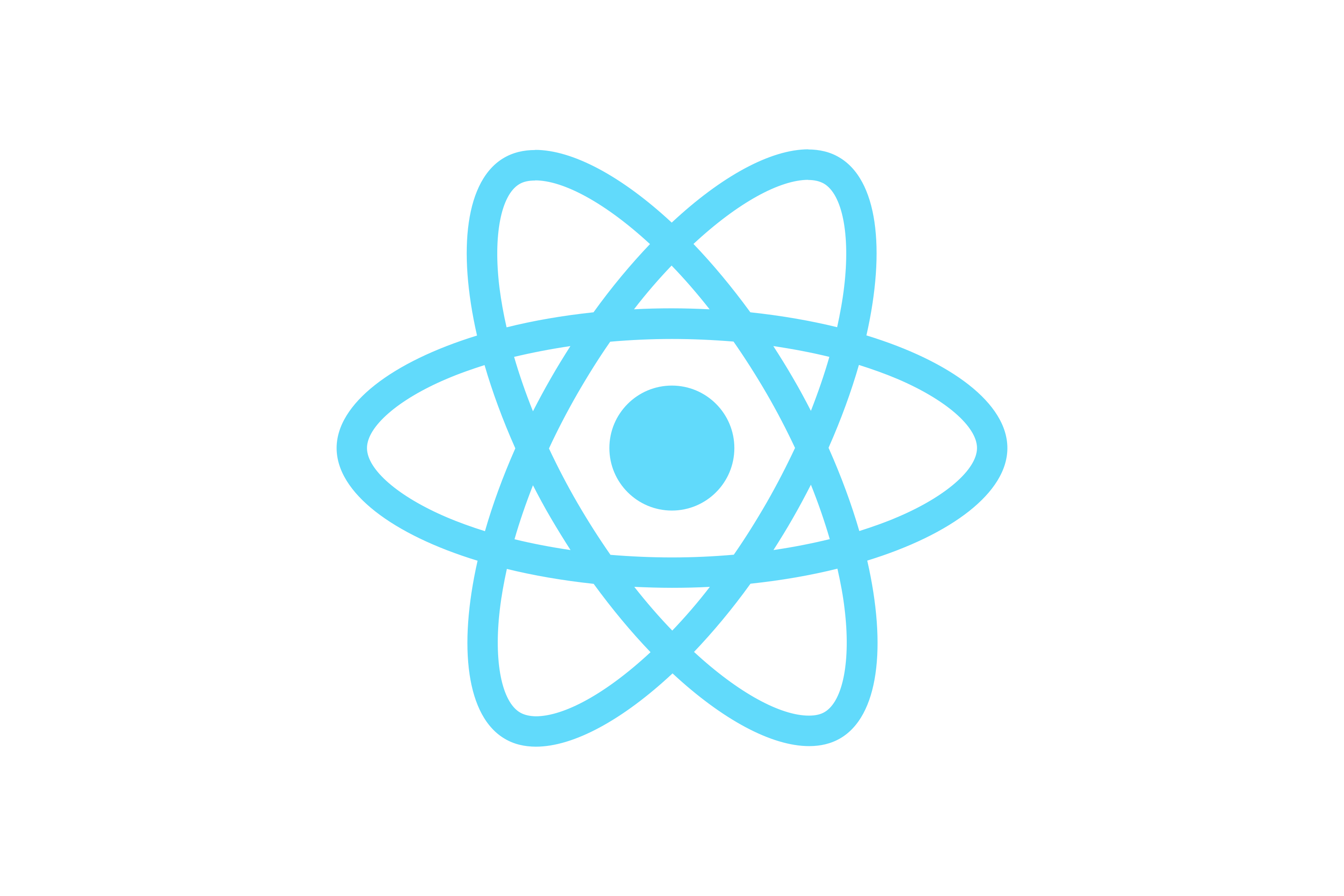
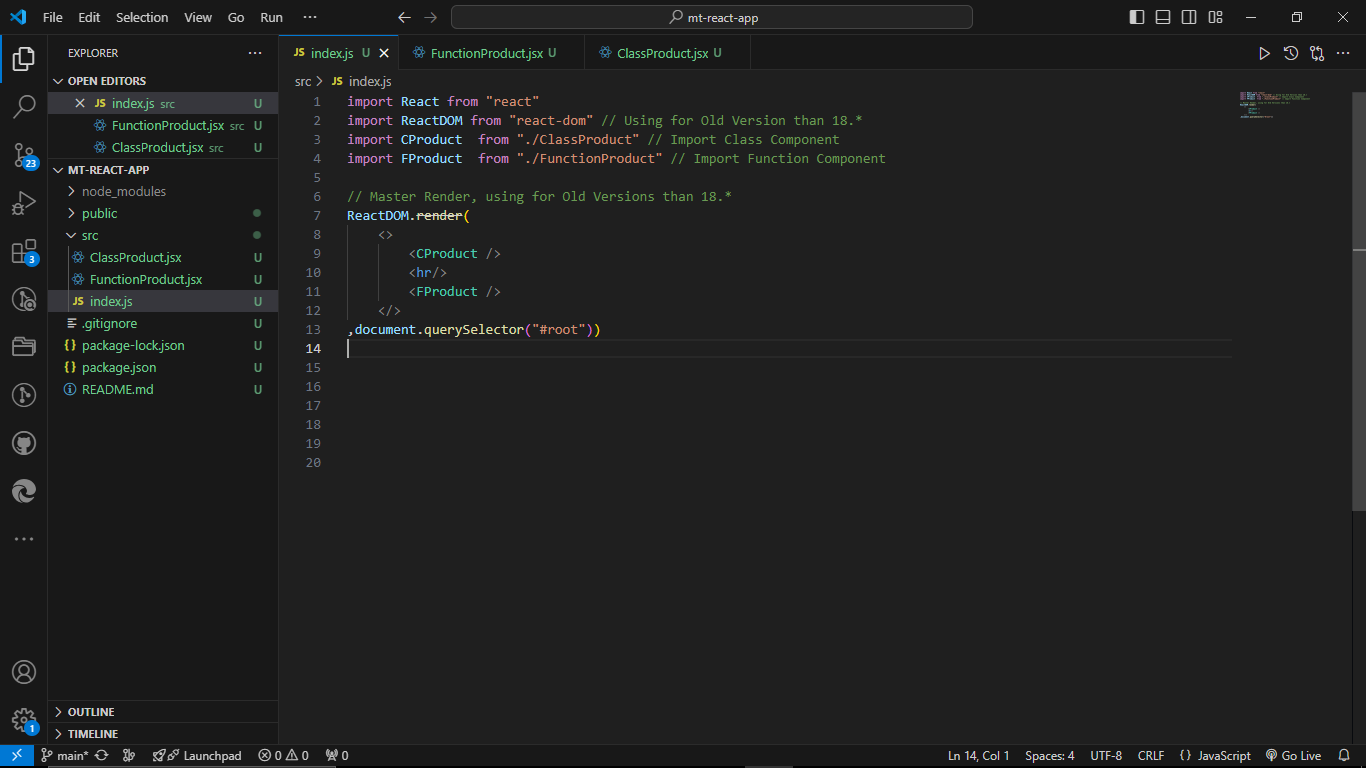
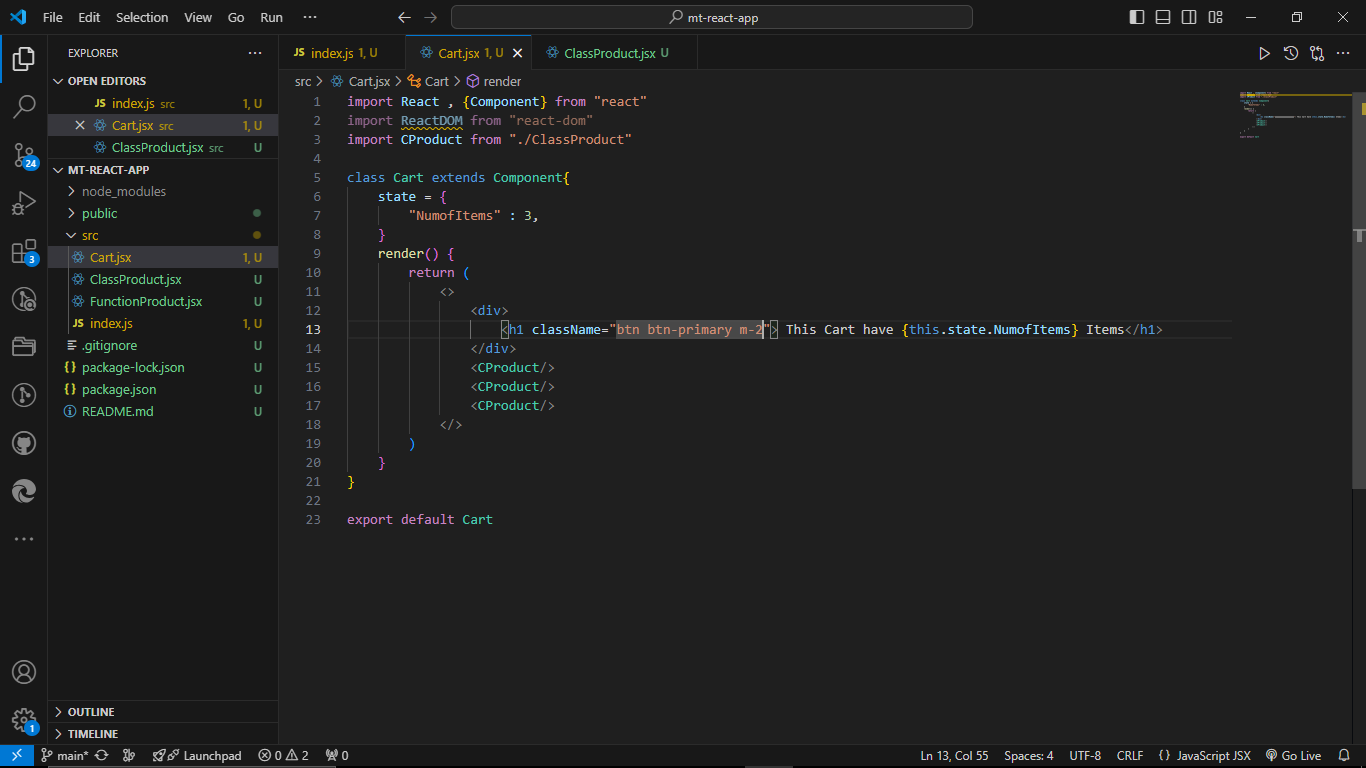
REACT JS FRAMEWORK



React Definition

* React is an open-source JavaScript Library Using for Creation Dynamic and Interactive User Interfaces for web applications
* React Using Virtual DOM, it Compare it Virtual DOM with Real DOM and Update the changed part only
* React Components
  + Component based architecture
  + Declarative Syntex
  + Virtual DOM Optimization
  + Onw Way Data Binding
  + JSX ‘JavaScript XML’ Integration
  + Hooks Employment
* React Installation Steps
  + Install node.js as CMD
  + Install VS Code as Editor
  + npm create vite@latest ‘CMD Command’
  + set name of project’s folder
  + choose “react” as framework
  + choose “JavaScript” as Working Language
  + cd “project’s folder name” ‘CMD Command’
  + npm install ‘CMD Command’
  + npm run dev ‘CMD Command’
    - to run project, will open the given link
* React Installation Steps
  + npx create-react-app “app-name-in-lowercase”
  + cd “app-name-in-lowercase”
  + npm start ‘to run the program’

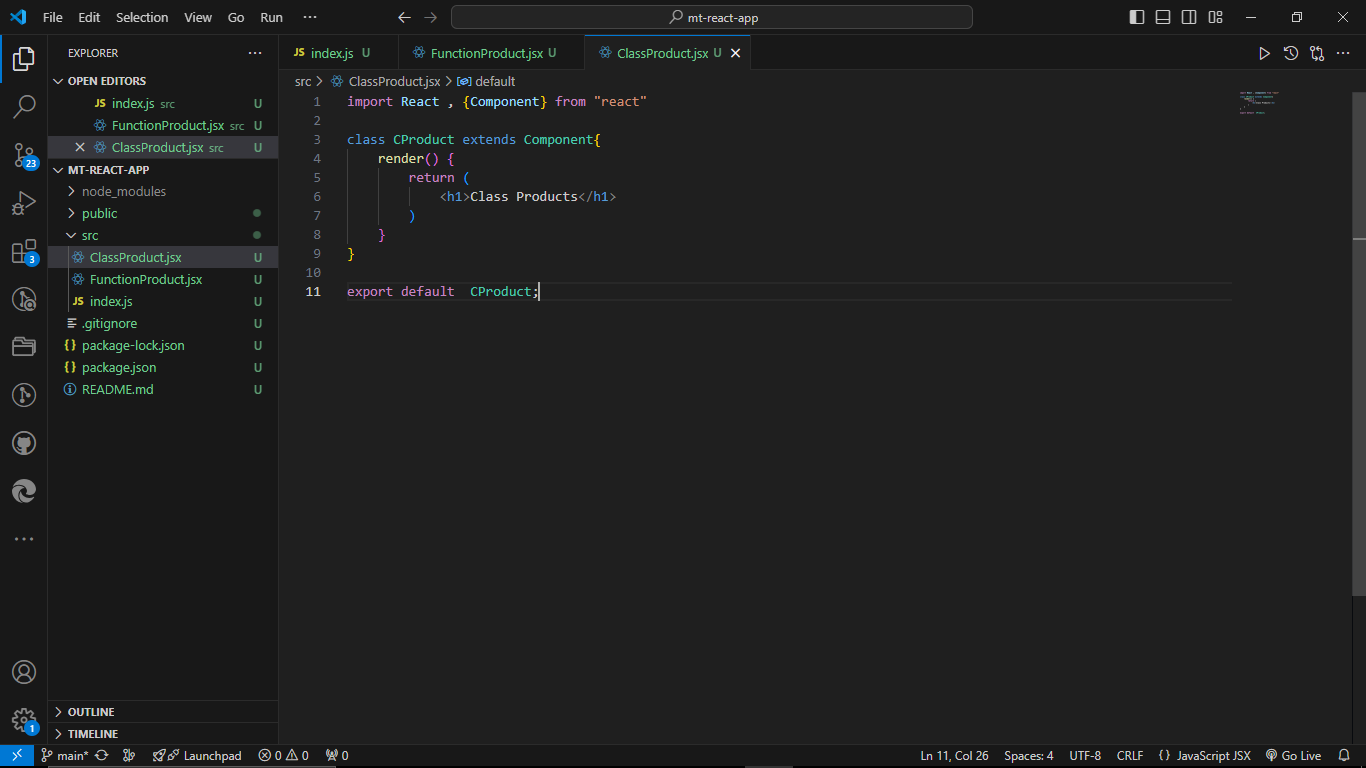
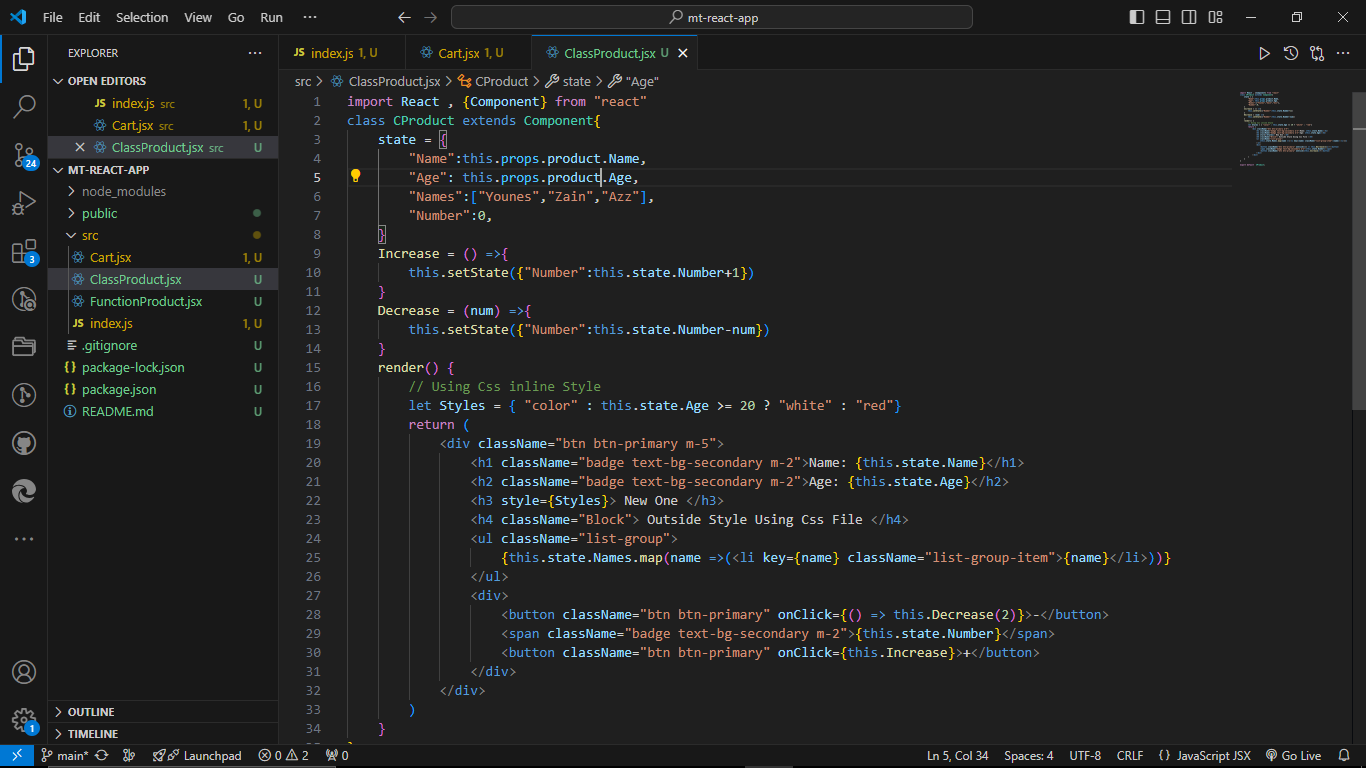
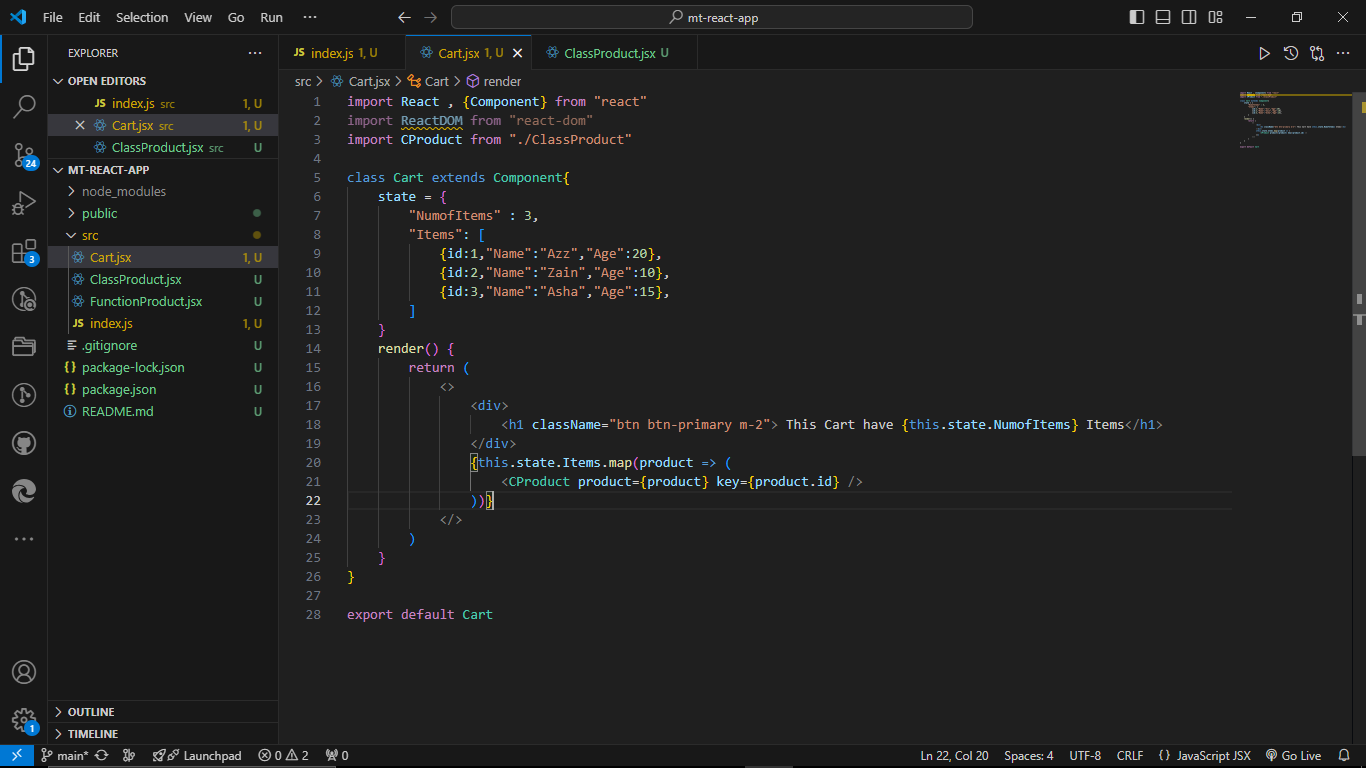
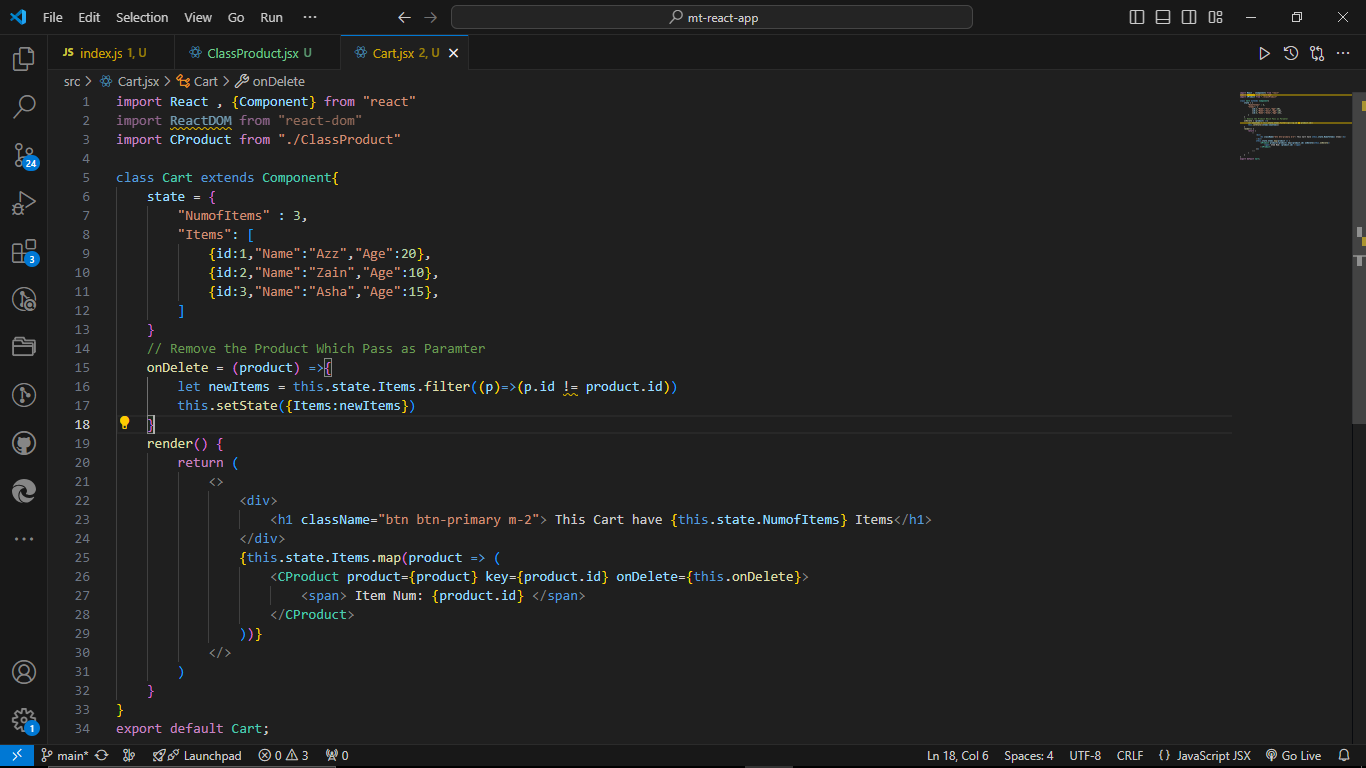
React Structure ‘src folder’

* create index.js file
* have all Render Components after importing it
* Render Component Before Version 18.\* as
  + 
* Render Component After Version 18.\* as
  + 

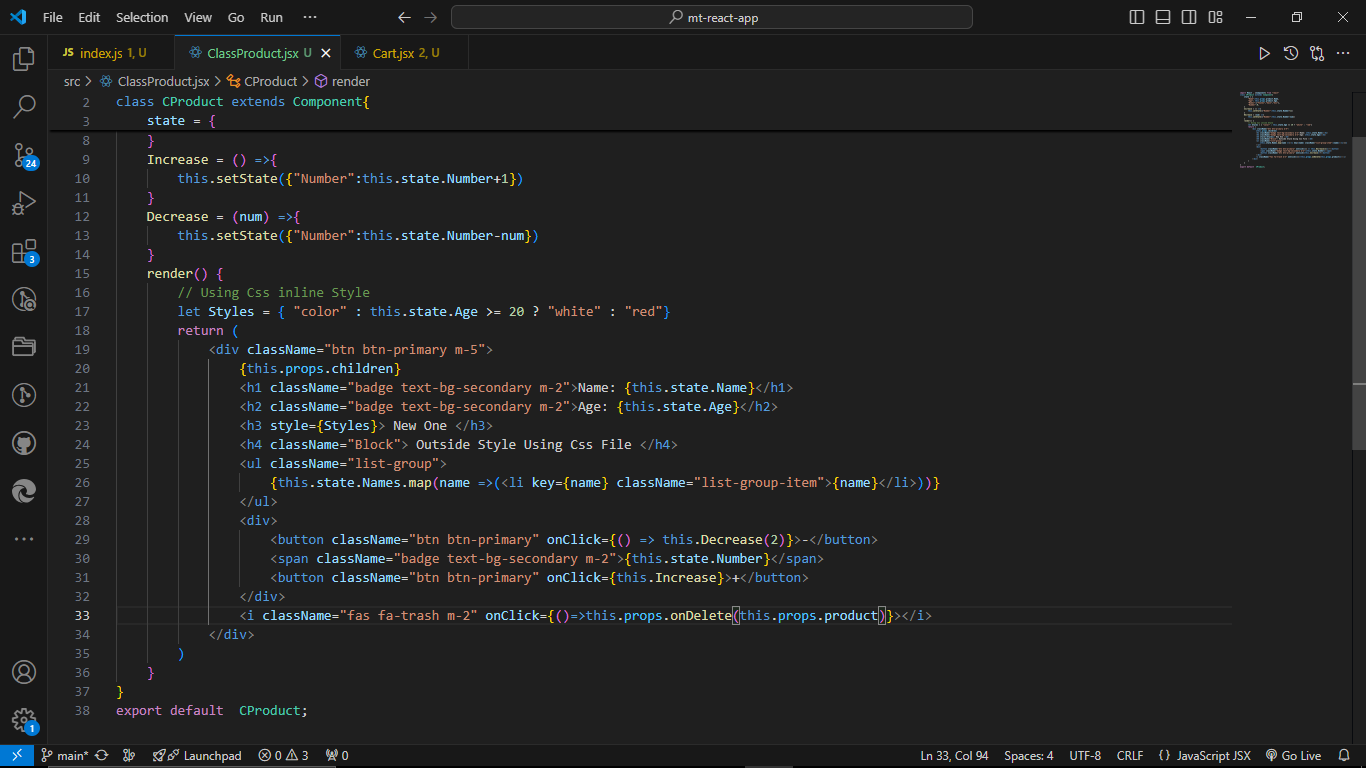
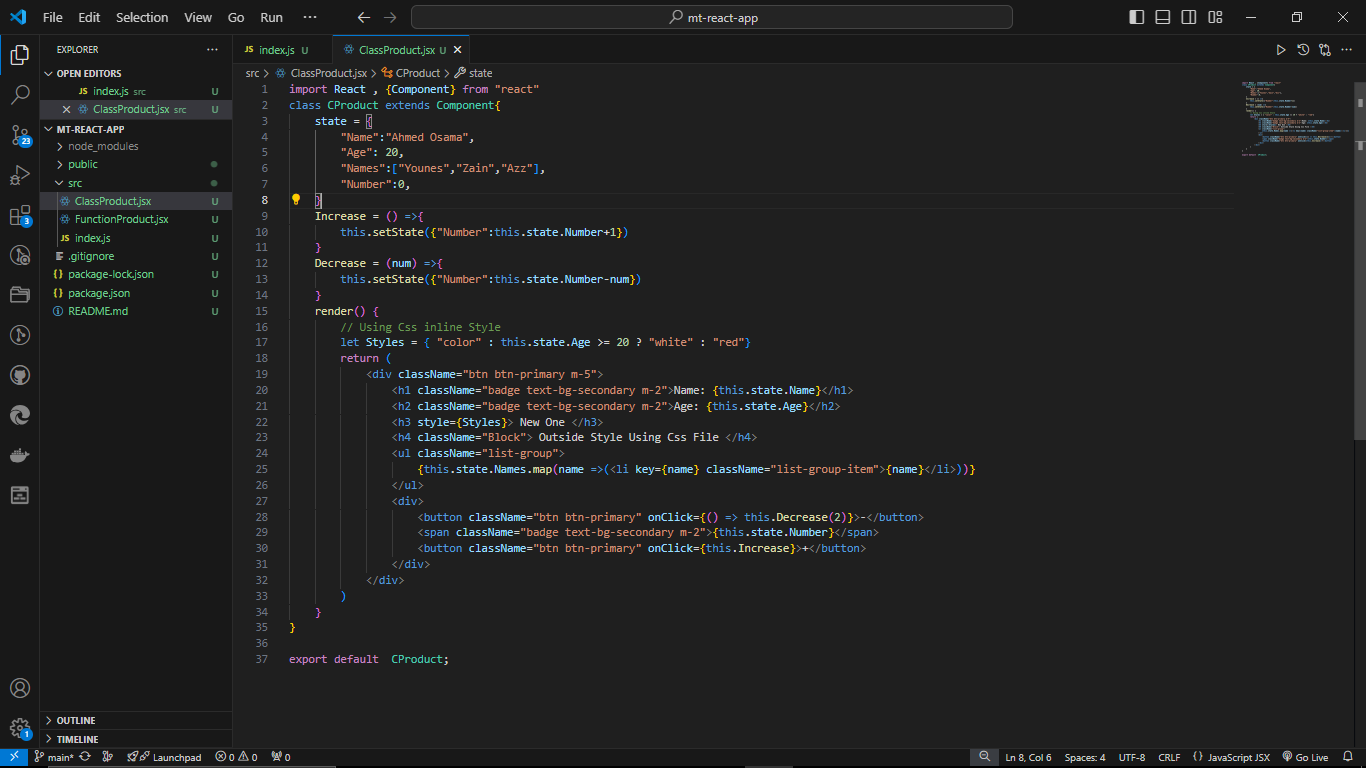
React Component

* The Main Block of the react framework
* Render Pieces of the Web Page
* Write in Private/Separated File with .jsx Extension
* Component features
  + Properties ‘Prop’
    - Manage Passing of data
  + Events
    - Manage DOM of Files
  + States
    - Updating the UI
* Types of Components
  + Functional
    - Written Like JS Functions
    - Return JSX
    - Enable Taken Arguments
  + Class Component
    - Written Using ES6
    - Data Passing from One to another
    - Use State, props, Lifecycle methods
    - More Complex than Functional
  + Higher Order Component ‘HOC’
    - Reuse Logic
    - Pass Component as Argument
    - Compose Components together

React Class Component

* It is Js Class
* Extends from React.Component class
* Manage State Lifecycle and Method, Events
* Props in Class Component
  + Using for Passing data
  + Read Only
  + Send data from parent to child
  + Methods are read only, not changeable
  + Can access methods and functions
  + Props Code as
    - 
    - 
  + Props Using for Shared Events between main class and called class
    - 



* + - 
* States in Class Component
  + Using for Manage Data
  + Allow Modify data within the component
  + Local State
    - Using for only its component to hide or show data
  + Shared State
    - Shared by Multiple Components, as ordering product
  + To Update State will foll these steps
    - Clone the piece which need to update
    - Edit the cloned piece
    - Using setState() to set the new value
  + State Code as
    - 
* Component Phases
  + Mounting ‘Create Component’
    - Methods Called in Order is

**constructor**

GetDerivedStateFromProps

ComponentWillMount

**render**

**componentDidMount**

* + Updating ‘Component Changing’
    - Methods Called in Order is

GetDerivedStateFromProps

ShouldComponentUpdate

**render**

GetSnapshotBeforeUpdate

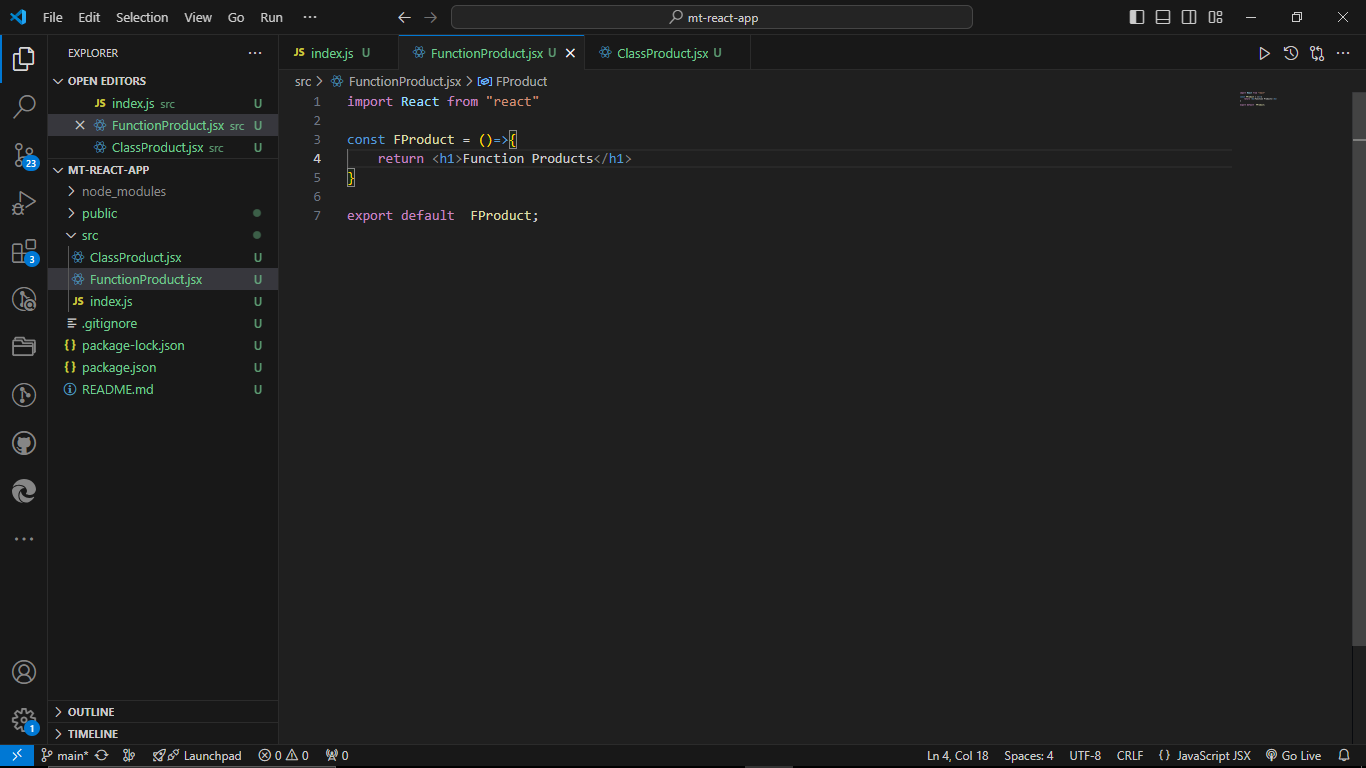
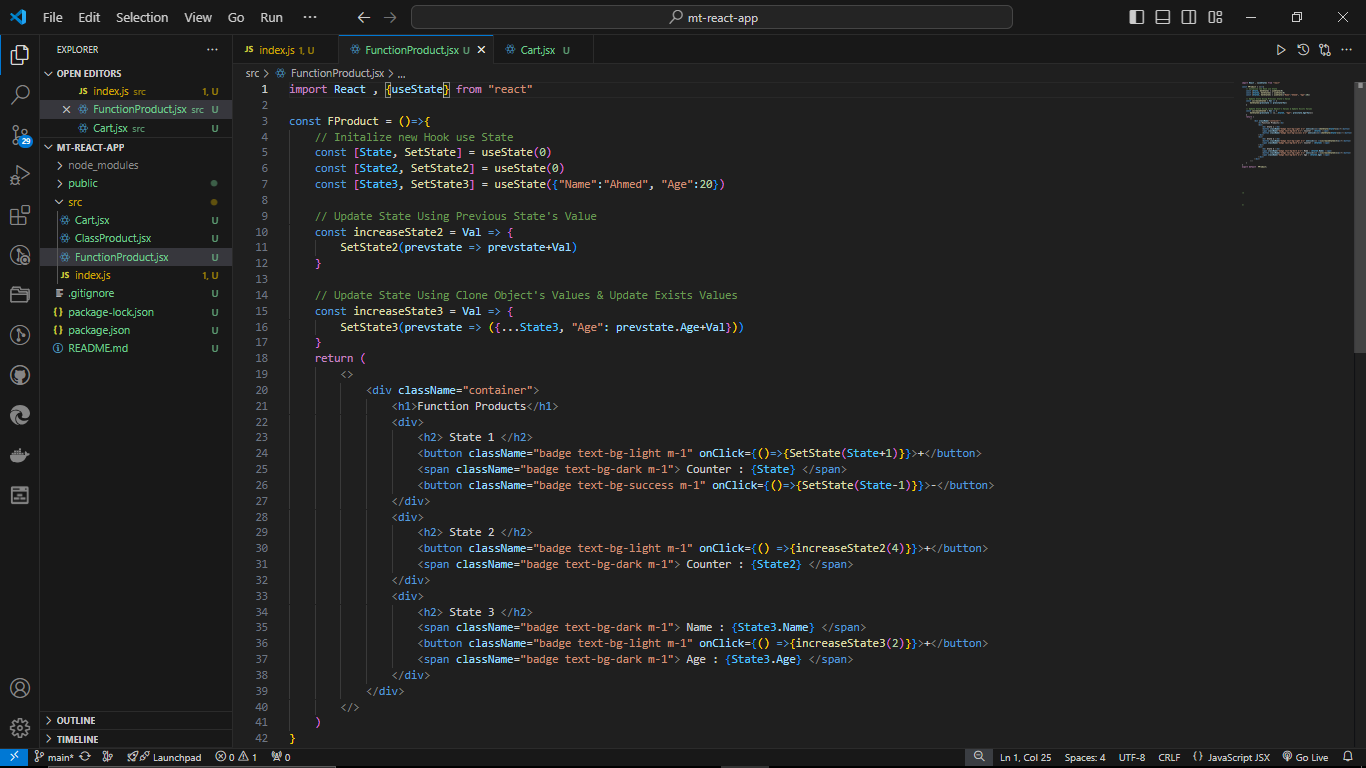
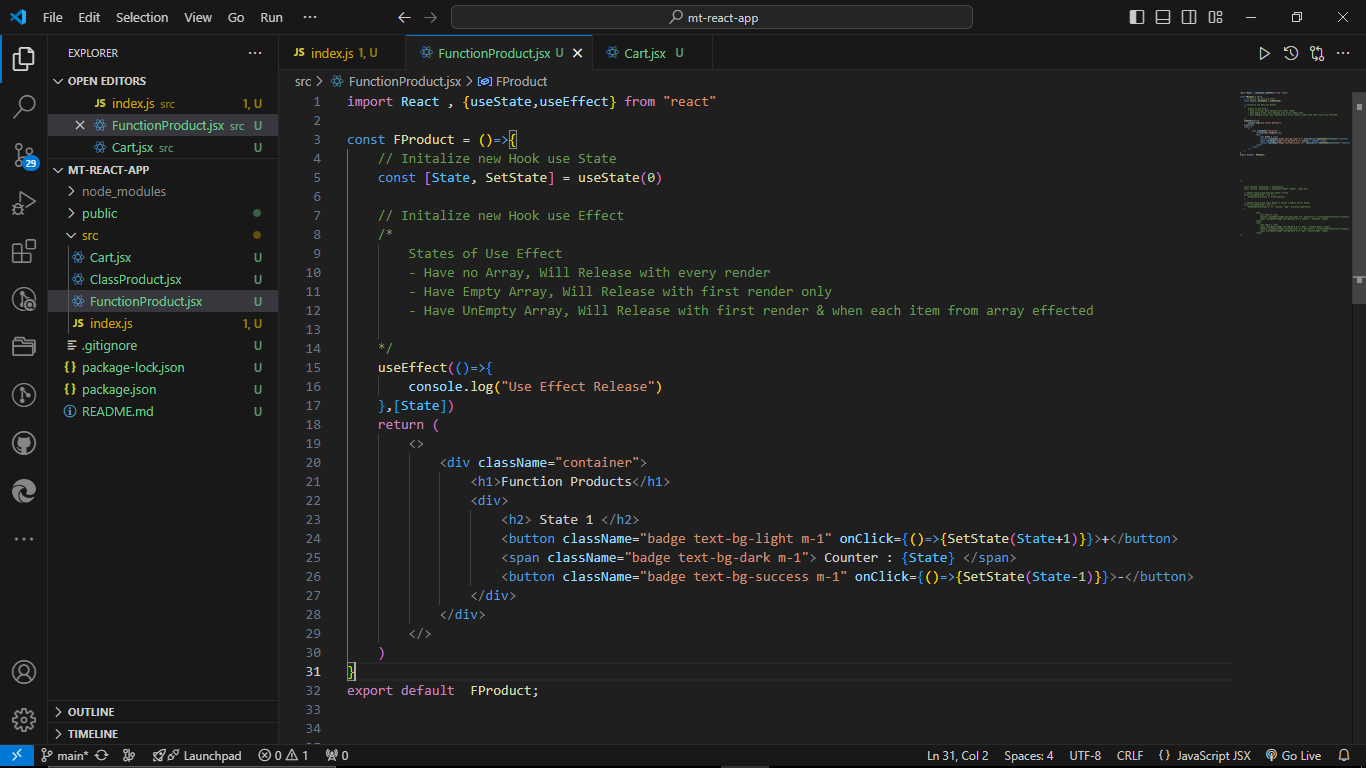
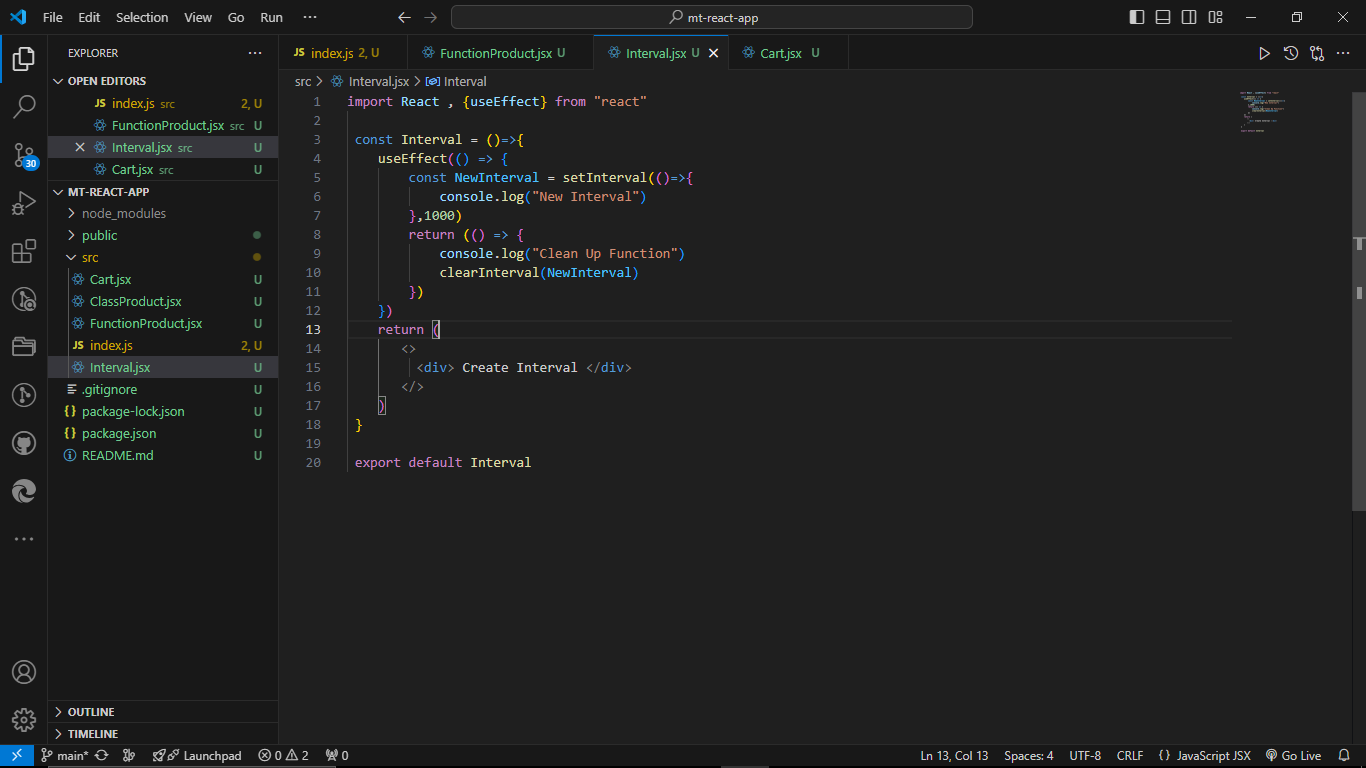
**componentDidUpdate**

* + Unmounting ‘Remove Component’
    - Methods Called in Order is

**componentWillUnmount**

* To Connect React with external servers uses so requests as
  + POST
    - To send data to server ‘hidden’
  + GET
    - To receive data from server ‘unhidden’
  + UPDATE
    - To update data in server
  + DELETE
    - To remove data
* Using Promises to handle the connection

React Function Component

* It is Js Function
* 
* Hooks
  + Rules of Hooks
    - Must not use in Class Component
    - Must be High Level in the Function Component
  + useState
    - 
  + useEffect
    - Use Effect Main Function
    - 
    - Use Effect as Clean up Function
    - 

React Routing