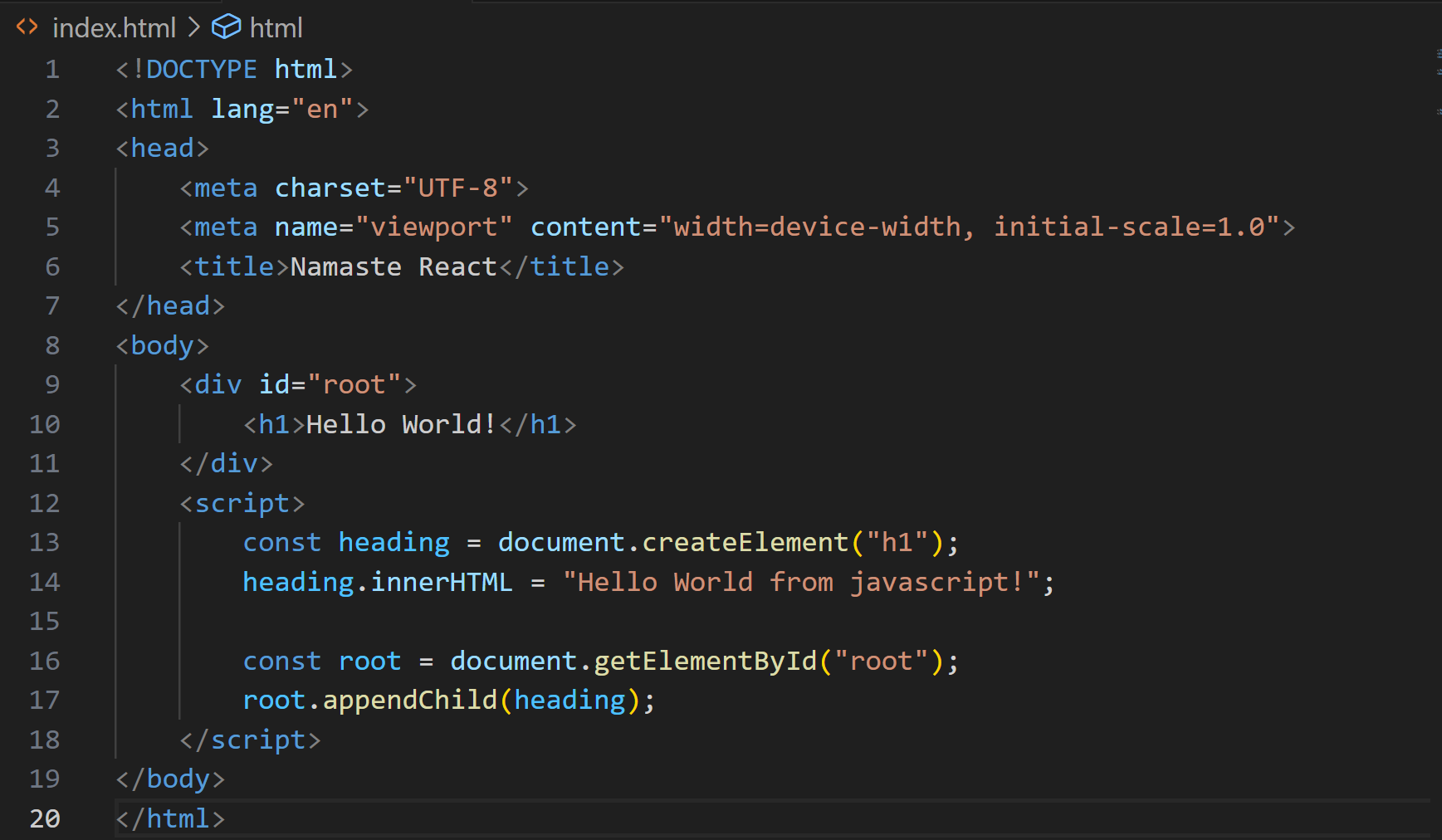
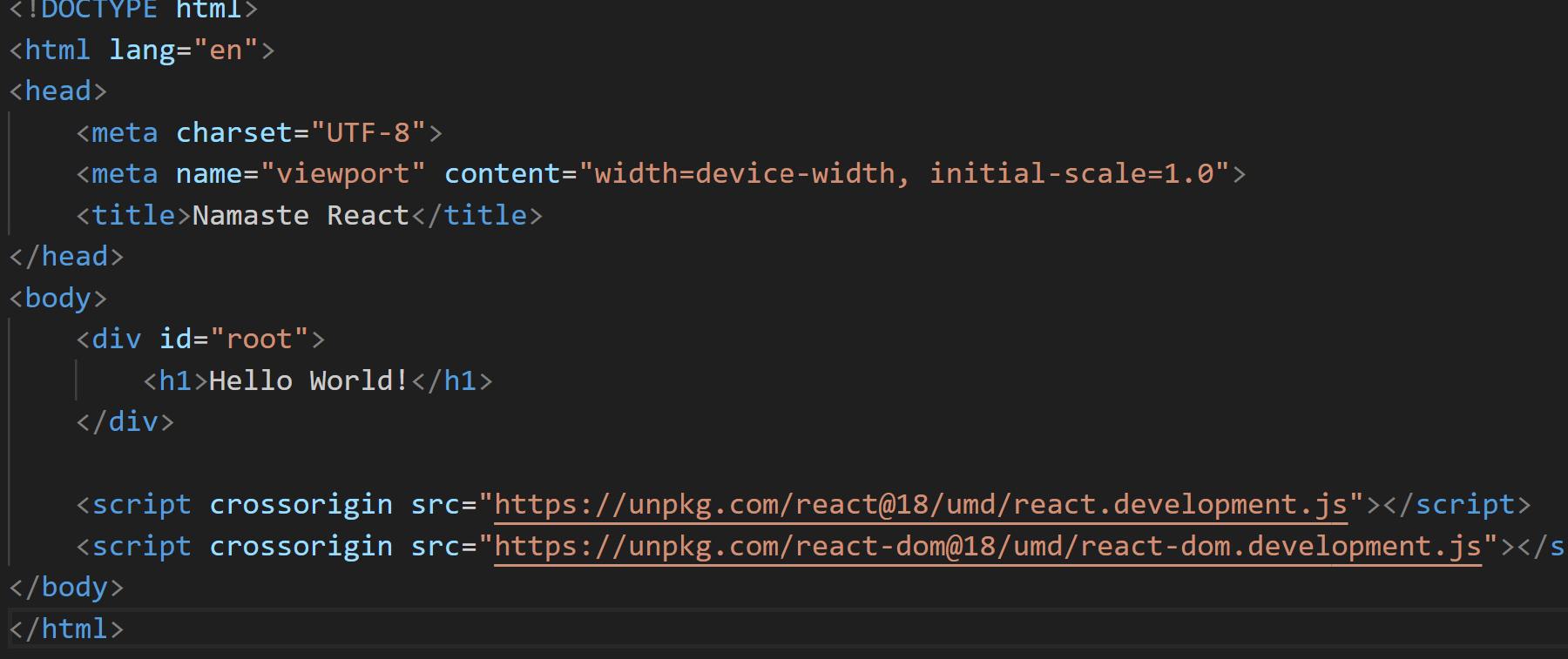
In visual Code editor if you write => Html:5 (Emmet Abbrevation) will give the basic skeleton for html.



In above hello world created using javascript.

Injecting react into project:

**Method 1 of injecting react into html**



A cross-origin request is a request for a resource (e.g. style sheets, iframes, images, fonts, or scripts) from another domain.

CORS is used to manage cross-origin requests.

CORS stands for Cross-Origin Resource Sharing, and is a mechanism that allows resources on a web page to be requested from another domain outside their own domain. It defines a way of how a browser and server can interact to determine whether it is safe to allow the cross-origin request. CORS allows servers to specify who can access the assets on the server, among many other things.

* Why we have two react files above?  
  First is the core react file which can be used in browser and mobile app.  
  Second file react dom is only used in browser to have dom operations.   
  So react is used at multiple places that’s why we have two different files here.

A black background with colorful text

Description automatically generated

A white background with colorful text

Description automatically generated

React comes with a philosophy to manipulate DOM which is most expensive operation in Web Pages , is to do it with javascript. It optimizes dom manipulation.

Segreagate code into js and css.



A screenshot of a computer program

Description automatically generated

**Create Nested Elements in react:**

Note: React cannot only be written in JSX, it can also be written in javascript only like we are doing right now. But this makes code unreadable and complex. So we use JSX.



As you can see that so much needs to be written to get simple nested structure. So here JSX comes into picture.   
Also in above console will throw error below which we will get rid of in future:  
A screenshot of a computer

Description automatically generated

Root.render **replaces** the existing content inside root element.

React only works inside your root element. So above and below portion of html will works the same. That’s why we call react as a library. Since it can be applied to a small portion of page. It is not a fully fledged framework.. its some piece of javascript code.

**NPM:** is a package manager. Packages are also called as dependency on which our project depends on.  
Package.json has a configuration required in our project

**Bundler:** helps us bundle our code together(html,css,js), minified code, compress the code so that it can push to production. Eg webpack, parcel, wheat etc.

Two types of dependency which we can install using npm.  
1)Dev dependency: required only for development phase  
2) Normal dependency: used in production as well.

"devDependencies": {

    "parcel": "^2.11.0"

  }

Carrot ^ is used to upgrade the minor version automatically but not the major version  
tilde ~ is used to upgrade the major version automatically. So its not advisable.

Package-lock.json keeps track of exact version which installed. And verifies whatever installed on local machine should be present on production as well.

Transitive dependency: project’s direct dependency may have multiple dependencies further. These indirect dependencies called transitive dependency.

So here parcel has dependency on babel etc.. babel is transitive dependency here for us.

If we have package and package-lock.json then we can re install our nodeModules again.  
using npm install. It will check the package.json and install all the dependencies.  
  
So whatever we can regenerate then don’t put things on github.   
NPM command mostly used for installation

Npx command used to execut a package  
  
CDN links are not a perfect way to bring react /react dom into ur project:  
-> It will always make a cdn network request and makes sites slower each time.

**Method 2 of adding react into html:**  
  
A black background with colorful lines

Description automatically generated

A screen shot of a computer program

Description automatically generated

Above throws error since script tag in html cannot have import or exports. It only understands normal javascript. It has to be module. Normal javascript cannot have import or export

A screenshot of a computer error

Description automatically generated

 parcel uses a file watching algorithm written in c++.

- HMR: Hot Module Replacement (make the change in any file and save it. Will reflect automatically on browser)

- Local Server

- Provides faster build because it does caching

- Image optimization (Showing image on browser is very expensive, parcel does optimization for us)

- Production build: do minification of files and bundling, compress as well

- Content Hashing

- Code splitting

- Differential bundling - support older browsers

- Diagnostic, Error Handling

- Https

- Tree SHaking - remove unused code automatically

Dist folder in parcel: parcel generates a dev build and put code in dist folder. Similarly prod build will also generated in dist folder

Autogenerated folders like dist, .parcel-cache shud not push in git repo. So add them in gitignore

To support older browsers we need to add browserList to package.json



GO to documentation.. Must read.  
  
  
**Now we will start REACT:**

If we do following configuration in package.json then we just need to start project:

A screen shot of a computer

Description automatically generated

Npm start or npm run start

To have production build:  
npm run build

JSX is separate and react is separate. JSX makes dev life easier to write complex code.

JSX is a convention to merge javascript and html code together. It is html like / XM-like syntax.  
It is not html in javascript. JSX is different syntax  
JS engine doesn’t understand JSX, it understands ES6 and all other pure javascript versions.   
So JSX is not a pure javascript. It is not understood by javascript  
So how it works?   
JSX code is getting transpilled before it goes to React or JS Engine. It is getting transpiled by Parcel which in turns uses Babel package**. Babel** is a transpiler (Javascript compiler)which converts JSX into something which is understood by React or JS engine.

A black screen with text

Description automatically generated

Its Babel job to convert ES6 code to backward compatible js code understood by older javascript

In HTML we do not use camelCasing in attribute name. For JSX we have to use camelCase

In html we do class as an attribute, but in jsx we use className.  
  
For single line statement in jsx we don’t need to wrap it with ().  
But if we have multiple line statement then we need to wrap it with (). So that babel will understand it.

React component: Functional component-> function which is returning JSX is a functional component  
Always name react component with first letter in capital letter

Component Composition: When components are injected into other component its called component composition

Always enclosed multiple React JSX element with one React JSX element when u return



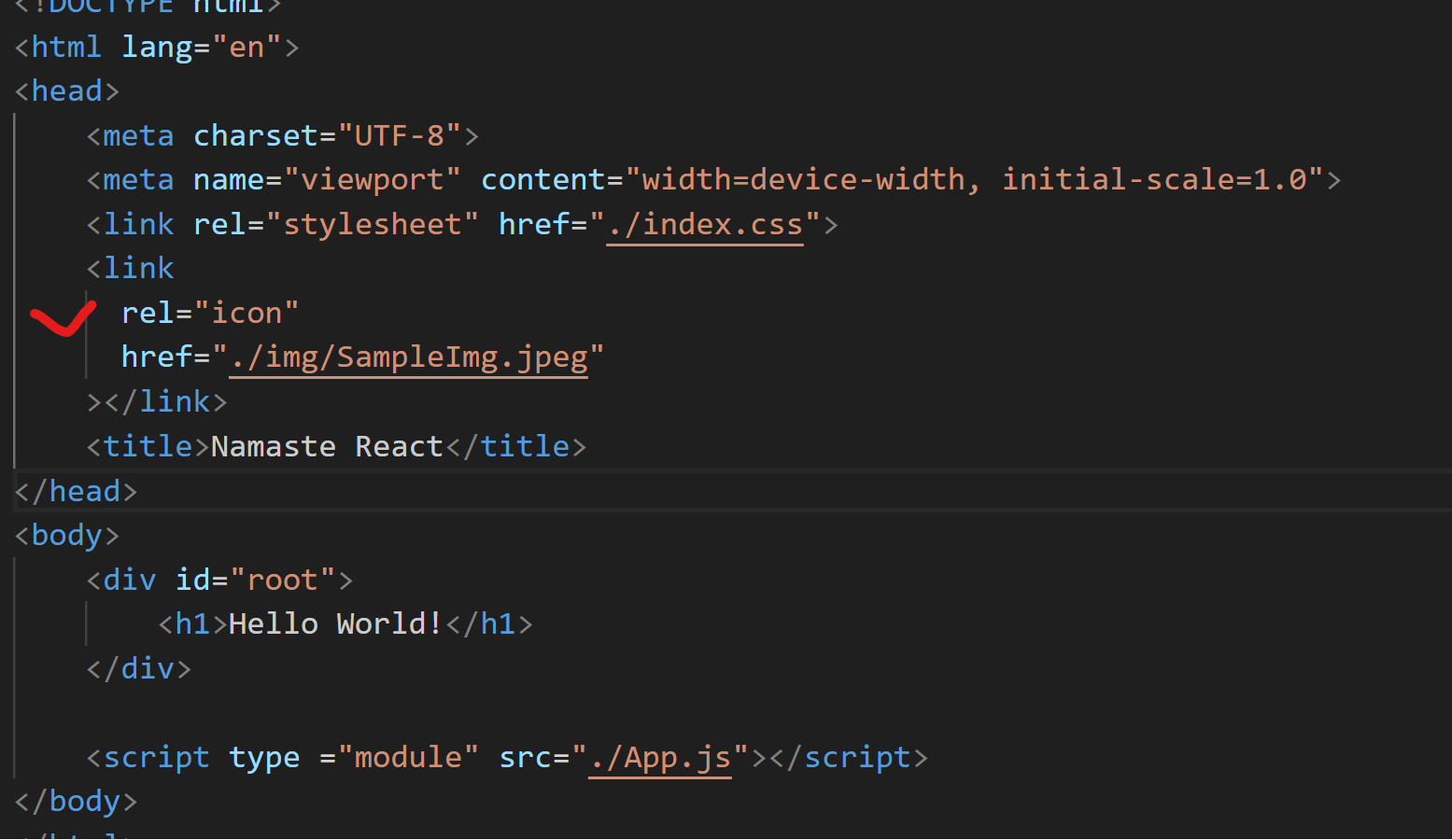
We can put any react element inside other react element, put react component inside react element and vice versa. That’s the beauty of JSX.

**JSX takes care of XXS attacks. It sanitizes the malicious injections to api responses.**

Components are basically a function so you can also inject component in following ways:  
**A screen shot of a computer

Description automatically generated**

Adding img as an icon to title of site:

Link tags defined the relationship between document and an external resource. Mostly used to put styple sheet or icon to title of page. Link tag is an empty element, it contains attributes only.  
****

**Add Img to body of html in react:**img tag is also empty element i.e it will only have attributes.



**Episode 4**

/\*

Header

-> Logo

-> Search Bar

-> Location

-> Nav Bar

Body

-> ProductsContainer (will have vertical scroll)

    -> ProductContainer(Will have left right scroll)

        -> ProductCard

Footer

-> AboutUS

-> CopyRightInfo

-> ContactUs

\*/

**ConfigDrivenUI:** Swiggy uses this approach. Website is driven by configs data. This config comes from backend. Config drives UI. Lie if something needs to be shown in some diff way then through data UI can read and display. Data changes and UI also changes accordingly

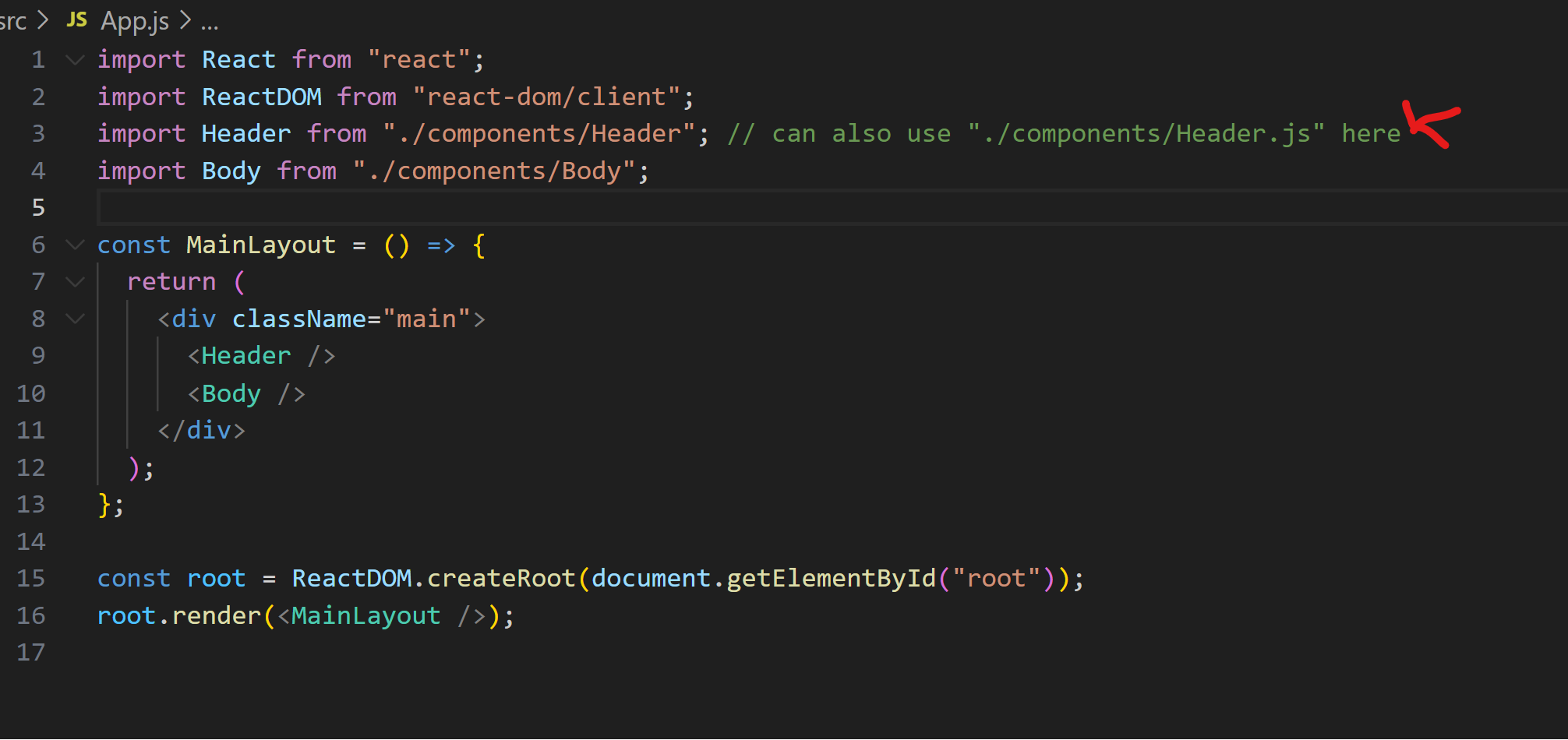
Whenever we use map function then we need to pass key as well as shown below:  


It gets applied to childs at same level in hierarchy.

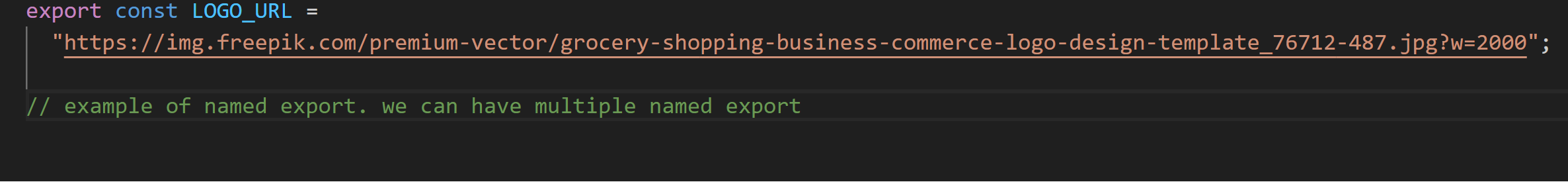
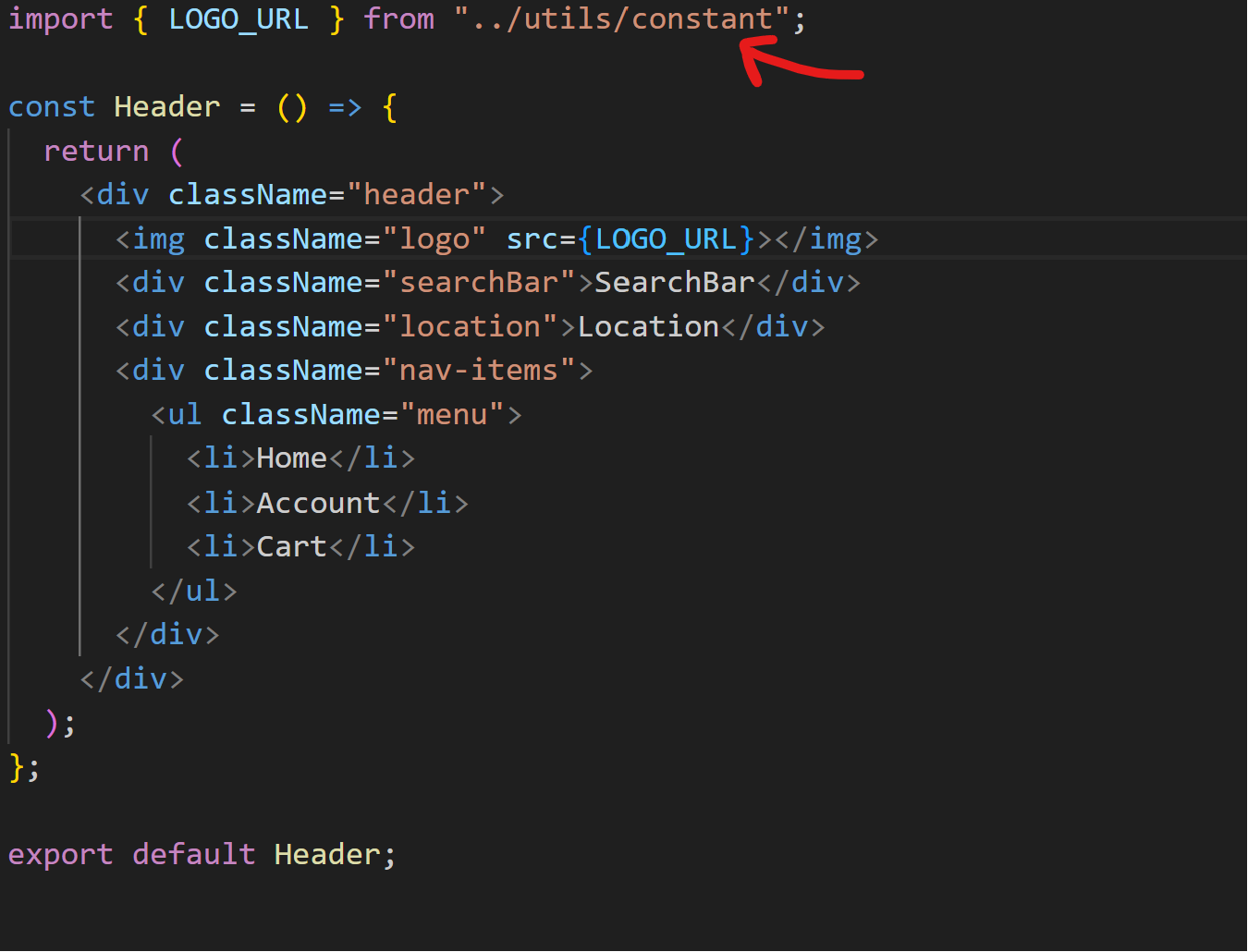
So if we provide key to list items then react will only render newly added keys and other already rendered items will be as it is. So it’s a huge optimization.

Some people use index as well as a key. Index is second param to map. But **react says don’t use index as key**. It’s a bad practise. Reason being if order of article changes then it will not work properly. **Key should always be uniquely identified.**

**Best practise is to make separate files for separate component  
Component name should be always start in capital letters otherwise react will not understand and it will not be rendered on UI.  
File name convention is to use exact same name with same caps and camelcasing we use for name the component. You can use .js or .jsx. don’t overthingk it doesn’t matter.**

****

As shown in above snippet we can import in any of the way with extension or without extension.  
Also to import first u need to do export in component file.  
**Named export**

We can also make named export as default export and import it directly instead of using {}  
  
**OnMouseOver** is the event which gets triggered when u hover to any element

**.**

**React Hook**

React solves the problem of updating UI layer whenever data gets changes.  
It keeps UI Layer and data layer in sync.  
React is fast bcz it can do faster dom manipulation.  
Hooks are normal javascript functions which are provided by React with some super powers.

useState(): gives super powerful react state variable These state variables keep UI in sync with data. Normal JS variables were not able to update the UI whenever their value got changed but state variable are able to.  
  
**Whenever a state variable updates, react re-renders the component.** So be cautious what u r writing in component.

hooks can only be called inside component