**Frontend Readme**

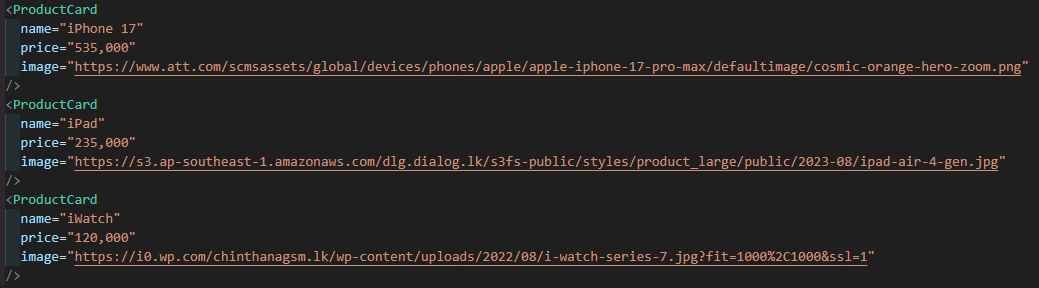
* npm create vite@latest
* npm install
* npm run dev
* when creating components, need to create a folder inside src named ‘components’
* component name must be start with a capital letter and the extension is ‘.jsx’

**props**

* inside a component we use props, so that we can maintain code redundancy. (inside html when we write JavaScript, we need to use {})

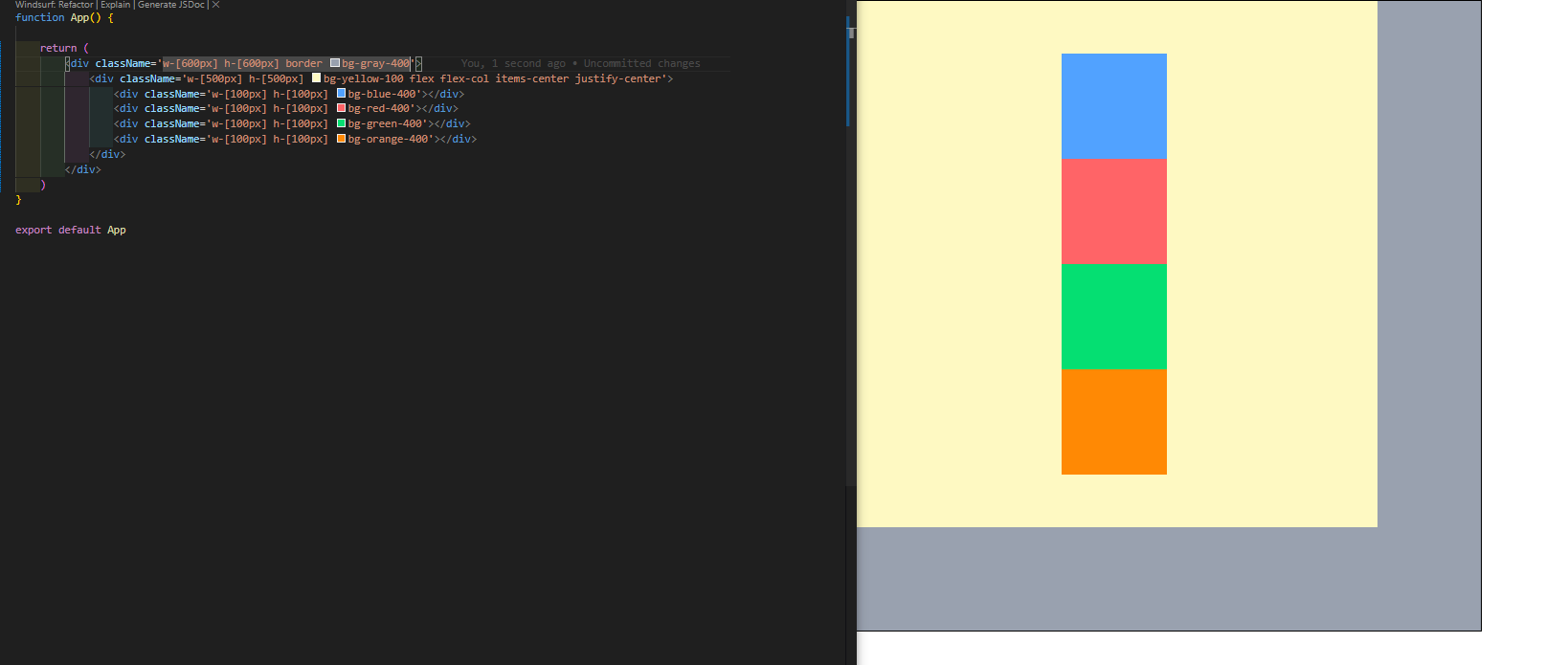


* When we import a component which include props, we need to pass relevant data with prop name.

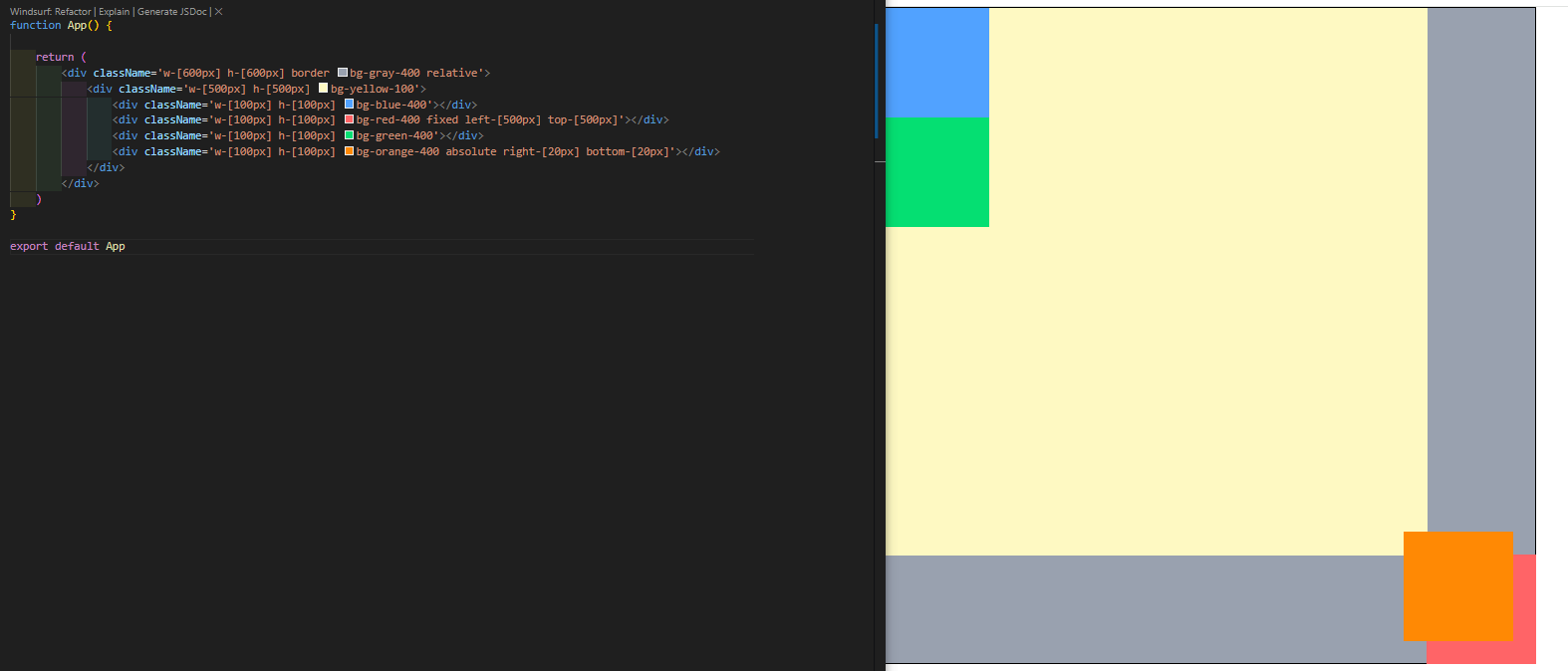


**Tailwind CSS**

* **Display – by default, display is in block** (one below another). First change display to **flex**.
  + When **display is in flex**, we can use **flex-col** to display it as one below another.
  + If we use **flex-row**, it displays as one after another.
  + **Horizontal/ Vertical center** -> item-center/justify-center.



* **Position - by default, position is in static.** 
  + When position is in **fixed (**it comes forward compared to other divs), we can move that div to anywhere using **left, right, top, bottom**. Fixed shows everywhere in fixed position.
  + When position is in **absolute (**it comes forward compared to other divs), we can move that div to anywhere using **left, right, top, bottom.** If there is a **position relative in a parent div**, **Absolute positioning in relation to that div**. Else it goes everywhere.



**Hooks**

1. Hooks must write in the beginning of the function.
2. Hooks must read according to an order. It shouldn’t be changed.
3. Hooks cannot be added conditionally.
4. **useState Hook**

* There is an array with two things
  + First one is a variable
  + Second one is a function
  + Value of the useState is determined by the type of the variable

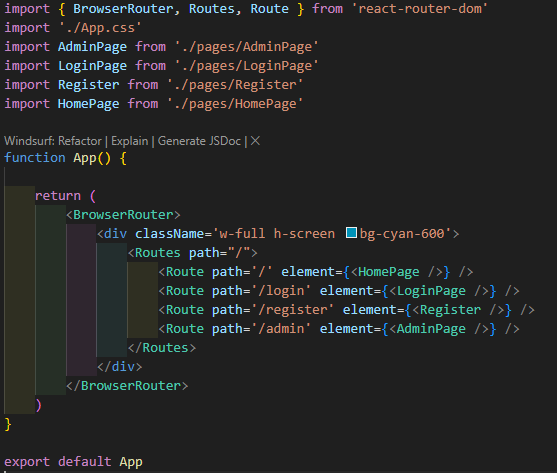
Ex: According the attached image count is a number, we use 0 as the value.

* + When set function is used jsx is rendering again and again when set function is used.



**Routing**

* npm install react-router-dom
* Wrap everything in App.jsx with BrowserRouter



**Color Palette**