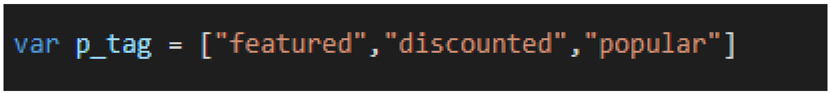
index.js

This JavaScript file is responsible of loading all the necessary element like the card, search bar and etc. It handles the layout of each element that is interactable like the category buttons and other interactable element in the DOM.

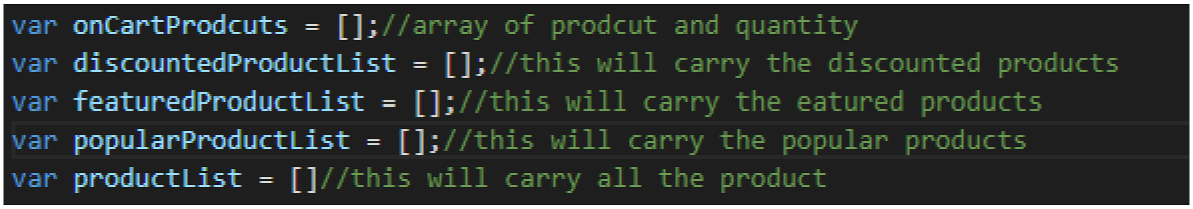
Code:

This code is temporary, this is for sorting the items in the home page.



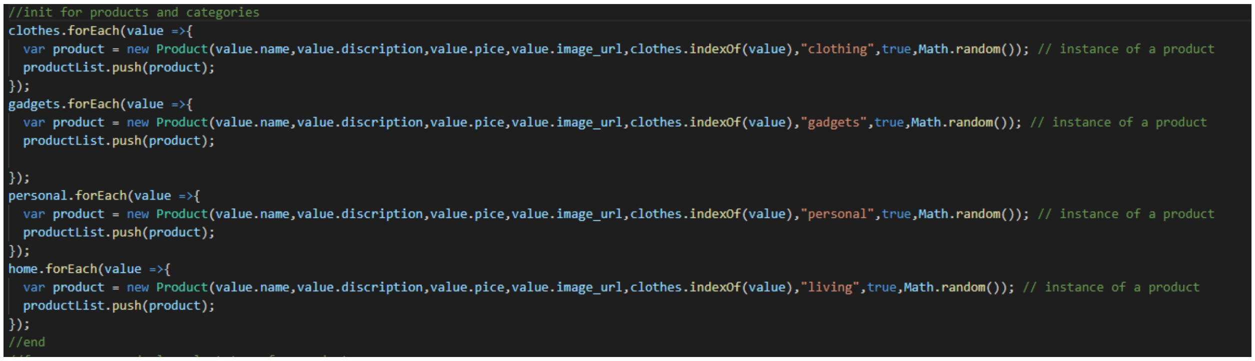
```
var p_tag = ["featured","discounted","popular"]
```

These arrays are responsible for storing a list of data.



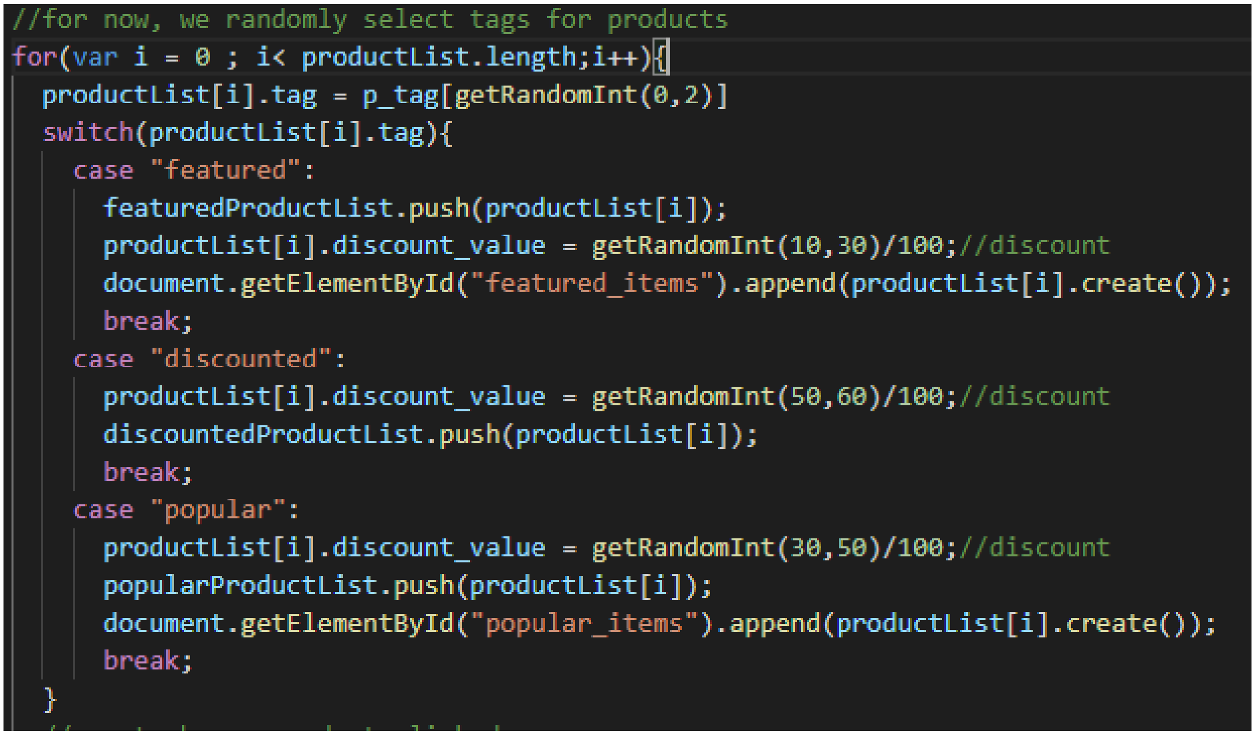
```
var onCartProdcuts = [];//array of prodcut and quantity
var discountedProductList = [];//this will carry the discounted products
var featuredProductList = [];//this will carry the eatured products
var popularProductList = [];//this will carry the popular products
var productList = []//this will carry all the product
```

Initializing all product from respective categories and storing it in the productList.



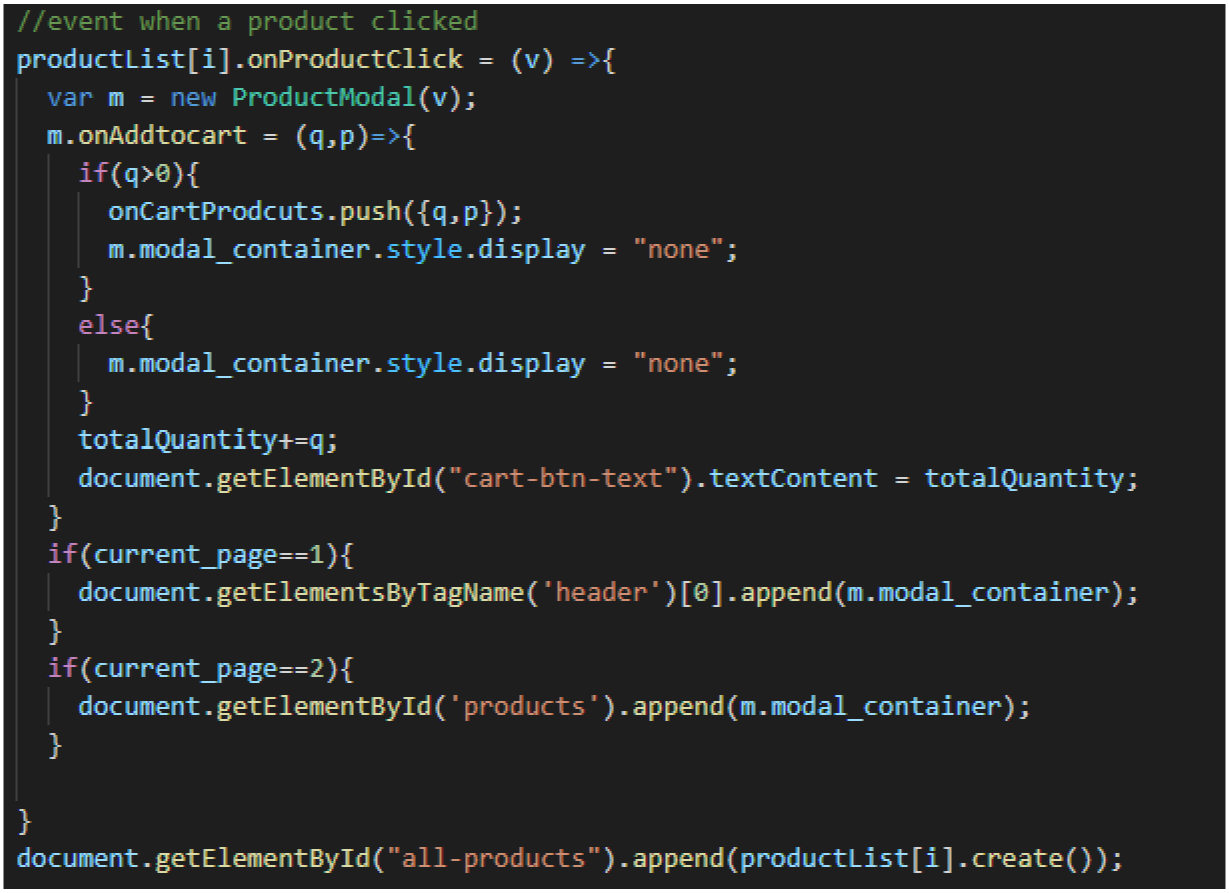
```
//init for products and categories
clothes.forEach(value =>{
  var product = new Product(value.name,value.discription,value.pice,value.image_url,clothes.indexOf(value),"clothing",true,Math.random()); // instance of a product
  productList.push(product);
});
gadgets.forEach(value =>{
  var product = new Product(value.name,value.discription,value.pice,value.image_url,clothes.indexOf(value),"gadgets",true,Math.random()); // instance of a product
  productList.push(product);

});
personal.forEach(value =>{
  var product = new Product(value.name,value.discription,value.pice,value.image_url,clothes.indexOf(value),"personal",true,Math.random()); // instance of a product
  productList.push(product);
});
home.forEach(value =>{
  var product = new Product(value.name,value.discription,value.pice,value.image_url,clothes.indexOf(value),"living",true,Math.random()); // instance of a product
  productList.push(product);
});
//end
```

Loop a – This code is responsible for displaying Items with random tags



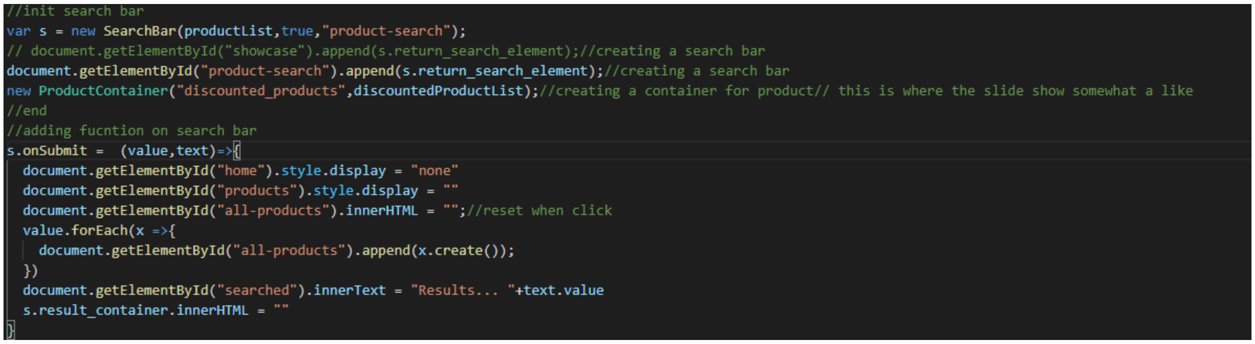
```
//for now, we randomly select tags for products
for(var i = 0 ; i< productList.length;i++){
  productList[i].tag = p_tag[getRandomInt(0,2)]
  switch(productList[i].tag){
    case "featured":
      featuredProductList.push(productList[i]);
      productList[i].discount_value = getRandomInt(10,30)/100;//discount
      document.getElementById("featured_items").append(productList[i].create());
      break;
    case "discounted":
      productList[i].discount_value = getRandomInt(50,60)/100;//discount
      discountedProductList.push(productList[i]);
      break;
    case "popular":
      productList[i].discount_value = getRandomInt(30,50)/100;//discount
      popularProductList.push(productList[i]);
      document.getElementById("popular_items").append(productList[i].create());
      break;
  }
```

Loop b – after displaying the randomly tag items, this code will give each Card(item) a function on click and as you can see we created an instance of a ProductModal class for the item to pop when the user clicked the Card(item)



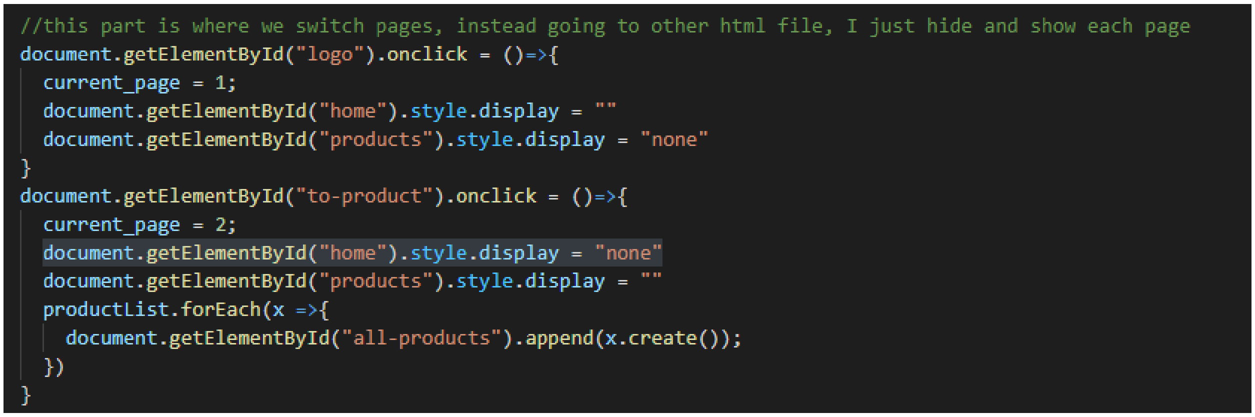
```
//event when a product clicked
productList[i].onProductClick = (v) =>{
    var m = new ProductModal(v);
    m.onAddtocart = (q,p)=>{
        if(q>0){
            onCartProdcuts.push({q,p});
            m.modal_container.style.display = "none";
        }
        else{
            m.modal_container.style.display = "none";
        }
        totalQuantity+=q;
        document.getElementById("cart-btn-text").textContent = totalQuantity;
    }
    if(current_page==1){
        document.getElementsByTagName('header')[0].append(m.modal_container);
    }
    if(current_page==2){
        document.getElementById('products').append(m.modal_container);
    }

}
document.getElementById("all-products").append(productList[i].create());
```

Creating an instance of SearchBar and display it in the DOM and adding onSubmit function.
Inside onSubmit function, (it is where the user hit enter of click on the search icon) we change the display by hiding the home page and displaying the product page and adding all the search product on the all-products div



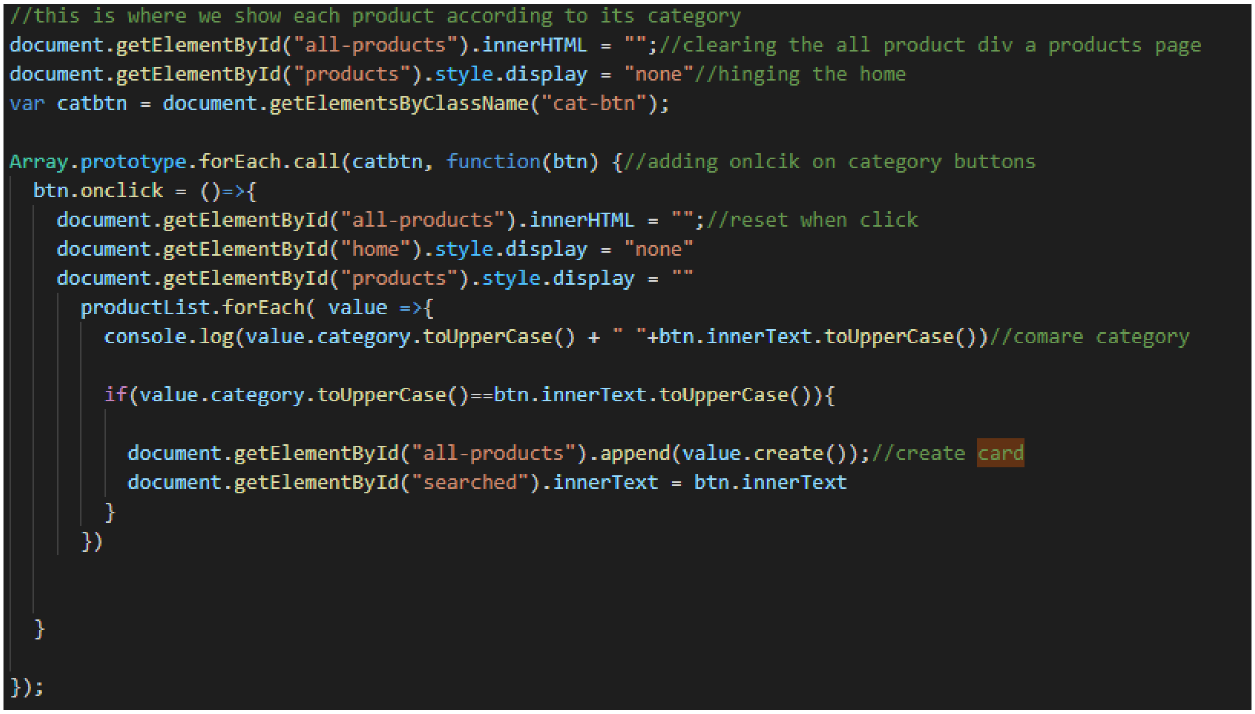
```
//init search bar
var s = new SearchBar(productList,true,"product-search");
// document.getElementById("showcase").append(s.return_search_element);//creating a search bar
document.getElementById("product-search").append(s.return_search_element);//creating a search bar
new ProductContainer("discounted_products",discountedProductList);//creating a container for product// this is where the slide show somewhat a like
//end
//adding fucntion on search bar
s.onSubmit =  (value,text)=>{
  document.getElementById("home").style.display = "none"
  document.getElementById("products").style.display = ""
  document.getElementById("all-products").innerHTML = "";//reset when click
  value.forEach(x =>{
    document.getElementById("all-products").append(x.create());
  })
  document.getElementById("searched").innerText = "Results... "+text.value
  s.result_container.innerHTML = ""
}
```

This codes is responsible for hiding and showing the homepage and the productpage



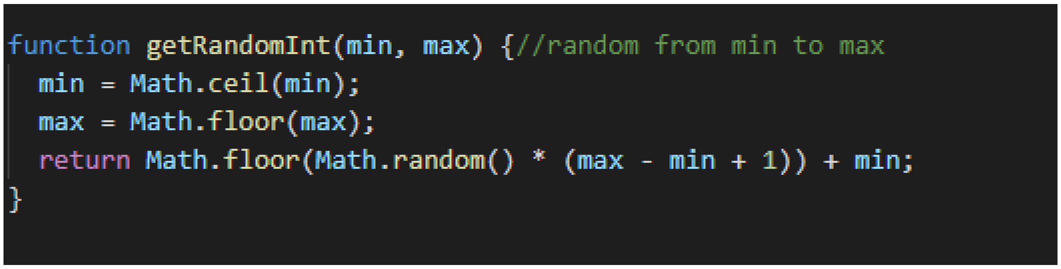
```
//this part is where we switch pages, instead going to other html file, I just hide and show each page
document.getElementById("logo").onclick = ()=>{
    current_page = 1;
    document.getElementById("home").style.display = ""
    document.getElementById("products").style.display = "none"
}
document.getElementById("to-product").onclick = ()=>{
    current_page = 2;
    document.getElementById("home").style.display = "none"
    document.getElementById("products").style.display = ""
    productList.forEach(x =>{
        document.getElementById("all-products").append(x.create());
    })
}
```

Displaying each product when the user clicked specific category



```
//this is where we show each product according to its category
document.getElementById("all-products").innerHTML = "";//clearing the all product div a products page
document.getElementById("products").style.display = "none"//hinging the home
var catbtn = document.getElementsByClassName("cat-btn");

Array.prototype.forEach.call(catbtn, function(btn) {//adding onlcik on category buttons
  btn.onclick = ()=>{
    document.getElementById("all-products").innerHTML = "";//reset when click
    document.getElementById("home").style.display = "none"
    document.getElementById("products").style.display = ""
      productList.forEach( value =>{
        console.log(value.category.toUpperCase() + " "+btn.innerText.toUpperCase())//comare category

        if(value.category.toUpperCase()==btn.innerText.toUpperCase()){

          document.getElementById("all-products").append(value.create());//create card
          document.getElementById("searched").innerText = btn.innerText
        }
      })


  }

});
```

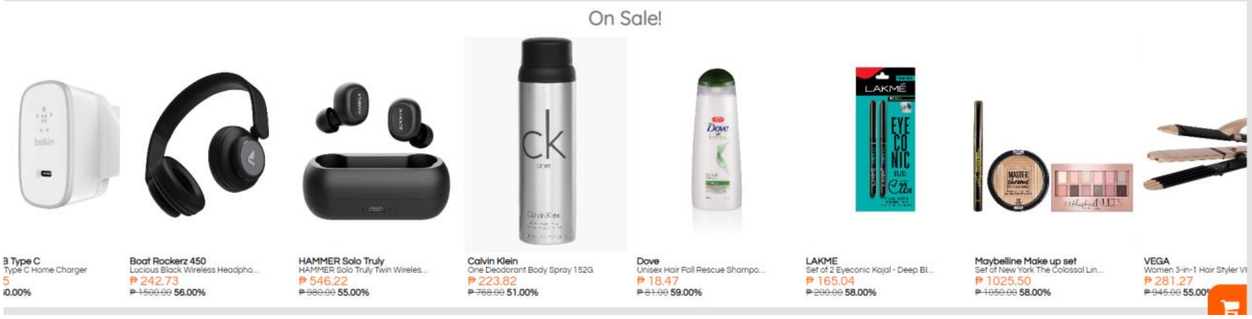
Just a random function that randomize a number from number min to number max



```
function getRandomInt(min, max) {//random from min to max
  min = Math.ceil(min);
  max = Math.floor(max);
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

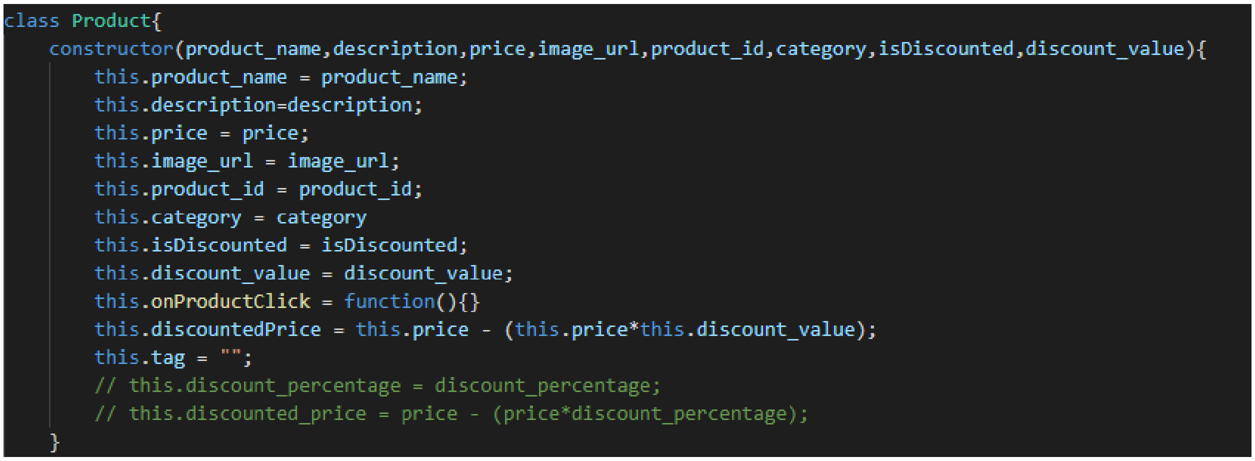
product.js

This JavaScript file is responsible for the Product class and the product card. We use this a blueprint for our products information such as name, description, price, image_url, id, category and discount value. We also have the function that is responsible for creating a card for our item. This file has also a ProductContainer which responsible for creating slidable products as you can see on the Home Page.



Code:

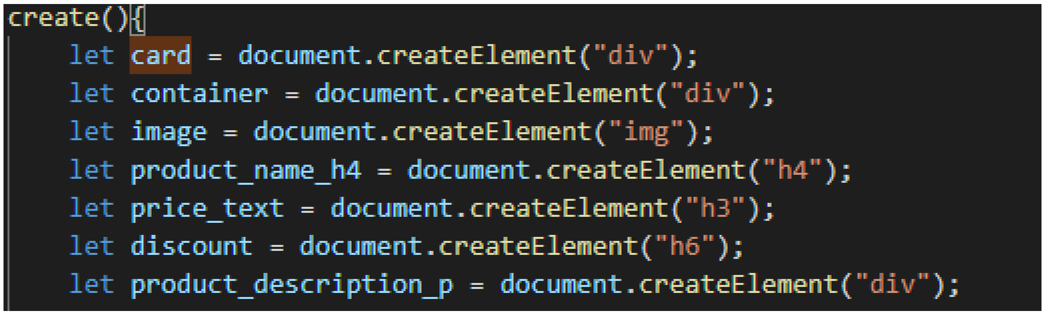
Product class – Creating a constructor



```
class Product{
    constructor(product_name,description,price,image_url,product_id,category,isDiscounted,discount_value){
        this.product_name = product_name;
        this.description=description;
        this.price = price;
        this.image_url = image_url;
        this.product_id = product_id;
        this.category = category
        this.isDiscounted = isDiscounted;
        this.discount_value = discount_value;
        this.onProductClick = function(){}
        this.discountedPrice = this.price - (this.price*this.discount_value);
        this.tag = "";
        // this.discount_percentage = discount_percentage;
        // this.discounted_price = price - (price*discount_percentage);
    }
```

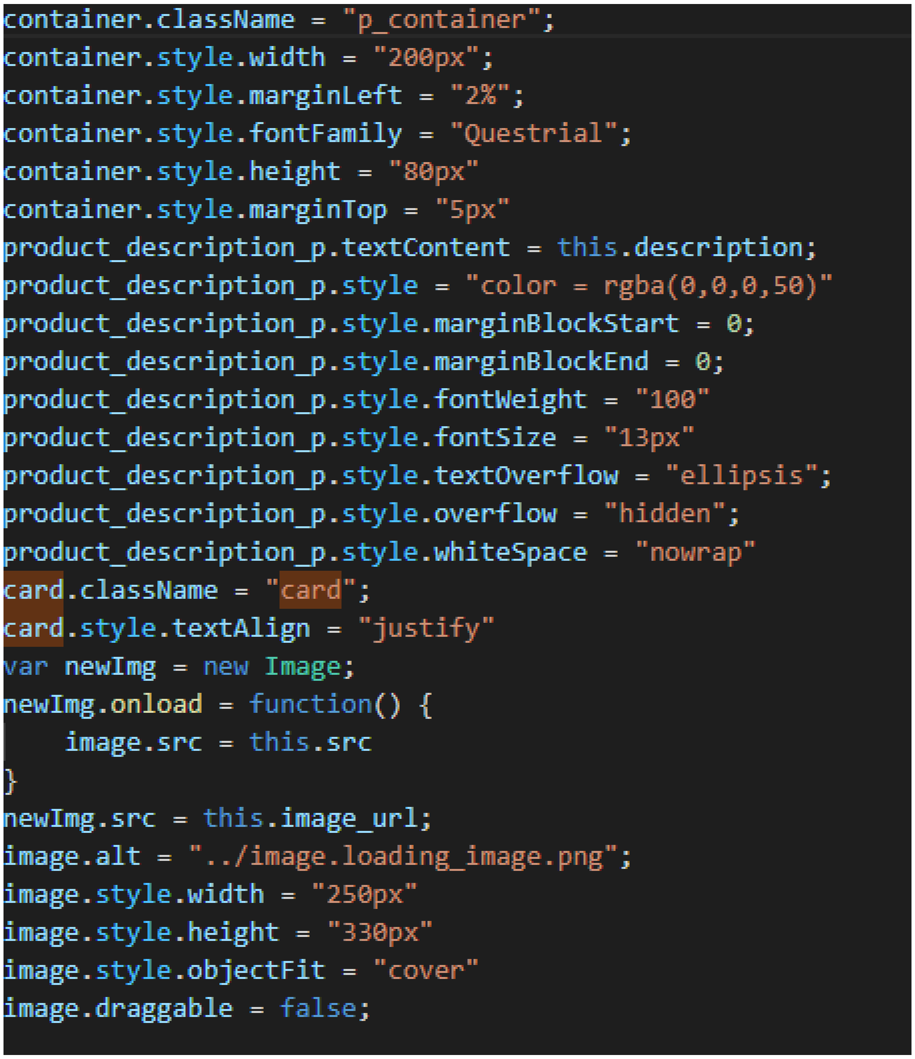
Product class - create() –

Creating elements for creating card for our product

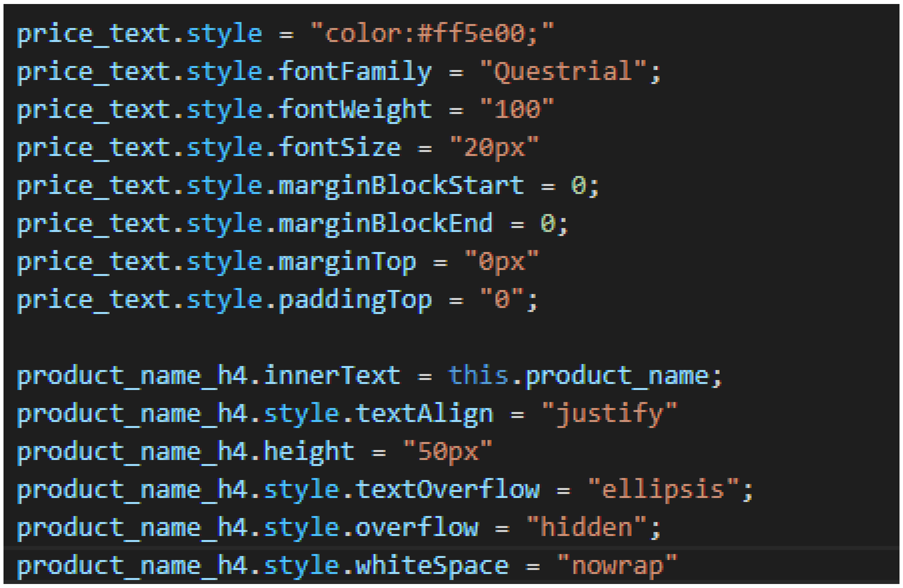


```
create(){
    let card = document.createElement("div");
    let container = document.createElement("div");
    let image = document.createElement("img");
    let product_name_h4 = document.createElement("h4");
    let price_text = document.createElement("h3");
    let discount = document.createElement("h6");
    let product_description_p = document.createElement("div");
```

Setting attributes

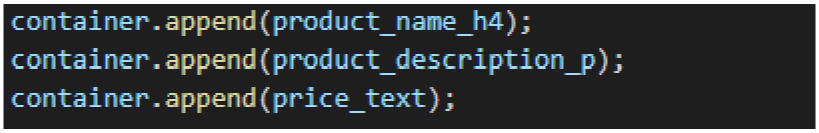


```
container.className = "p_container";
container.style.width = "200px";
container.style.marginLeft = "2%";
container.style.fontFamily = "Questrial";
container.style.height = "80px"
container.style.marginTop = "5px"
product_description_p.textContent = this.description;
product_description_p.style = "color = rgba(0,0,0,50)"
product_description_p.style.marginBlockStart = 0;
product_description_p.style.marginBlockEnd = 0;
product_description_p.style.fontWeight = "100"
product_description_p.style.fontSize = "13px"
product_description_p.style.textOverflow = "ellipsis";
product_description_p.style.overflow = "hidden";
product_description_p.style.whiteSpace = "nowrap"
card.className = "card";
card.style.textAlign = "justify"
var newImg = new Image;
newImg.onload = function() {
    image.src = this.src
}
newImg.src = this.image_url;
image.alt = "../image.loading_image.png";
image.style.width = "250px"
image.style.height = "330px"
image.style.objectFit = "cover"
image.draggable = false;
```



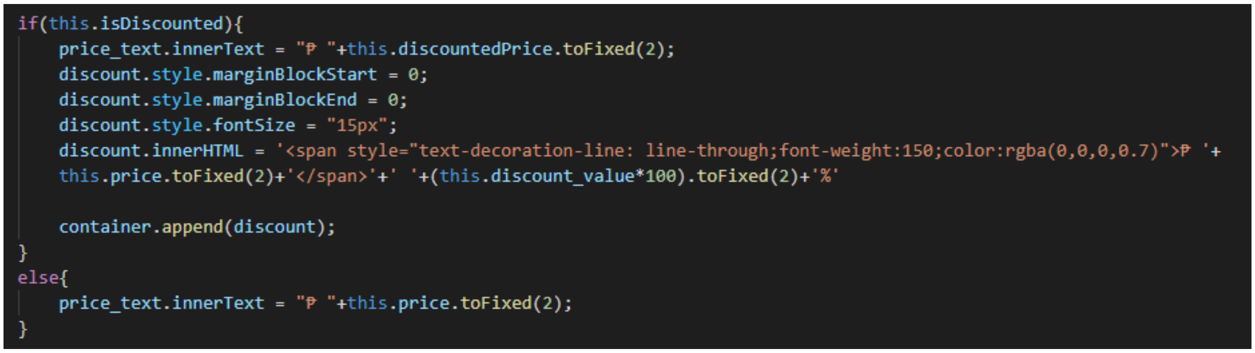
```
price_text.style = "color:#ff5e00;"
price_text.style.fontFamily = "Questrial";
price_text.style.fontWeight = "100"
price_text.style.fontSize = "20px"
price_text.style.marginBlockStart = 0;
price_text.style.marginBlockEnd = 0;
price_text.style.marginTop = "0px"
price_text.style.paddingTop = "0";

product_name_h4.innerText = this.product_name;
product_name_h4.style.textAlign = "justify"
product_name_h4.height = "50px"
product_name_h4.style.textOverflow = "ellipsis";
product_name_h4.style.overflow = "hidden";
product_name_h4.style.whiteSpace = "nowrap"
```

Adding elements to the container



```
container.append(product_name_h4);
container.append(product_description_p);
container.append(price_text);
```

Setting discount text for the product

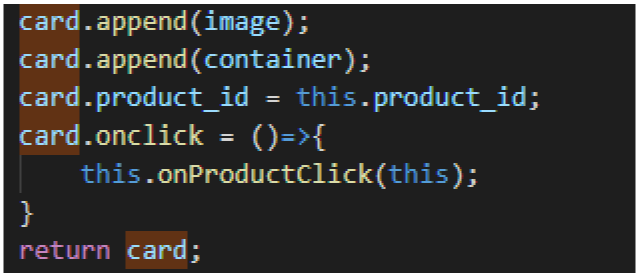


```
if(this.isDiscounted){
    price_text.innerText = "₱ "+this.discountedPrice.toFixed(2);
    discount.style.marginBlockStart = 0;
    discount.style.marginBlockEnd = 0;
    discount.style.fontSize = "15px";
    discount.innerHTML = '<span style="text-decoration-line: line-through;font-weight:150;color:rgba(0,0,0,0.7)">₱ '+
    this.price.toFixed(2)+'</span>'+' '+(this.discount_value*100).toFixed(2)+'%'

    container.append(discount);
}
else{
    price_text.innerText = "₱ "+this.price.toFixed(2);
}
```

Adding the image and the container to the card div element and return it .

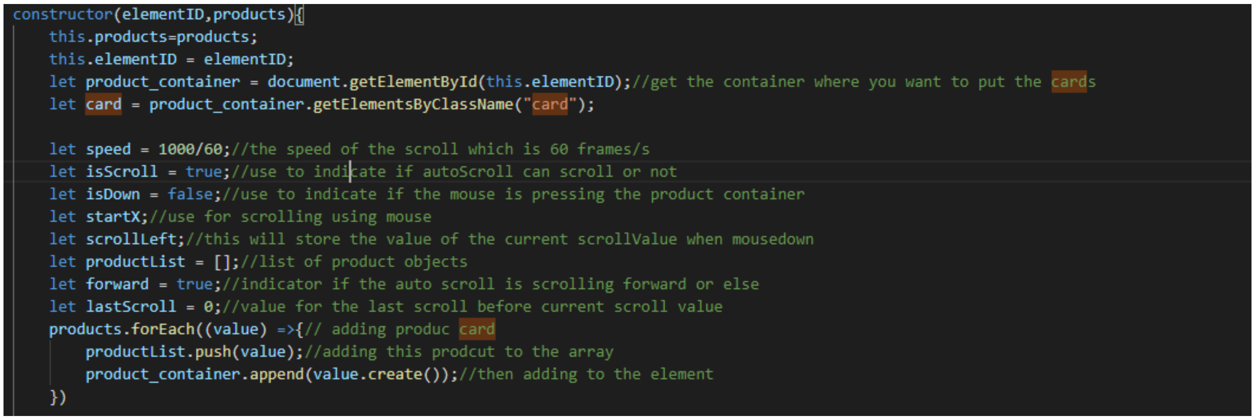
Therefore, you can call this fucntion inside of an appendChild or append.

Adding an onclick event and call the OnProductClick callback function that is use to something when this product is clicked.



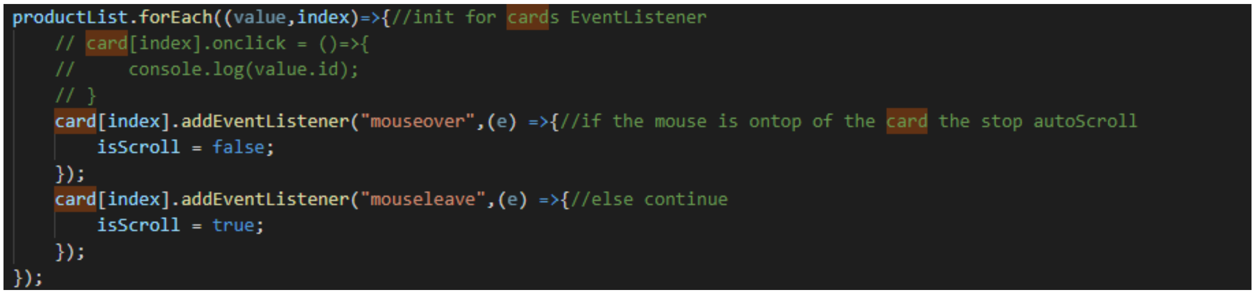
```
card.append(image);
card.append(container);
card.product_id = this.product_id;
card.onclick = ()=>{
    this.onProductClick(this);
}
return card;
```

ProductContainer class – Creating a constructor



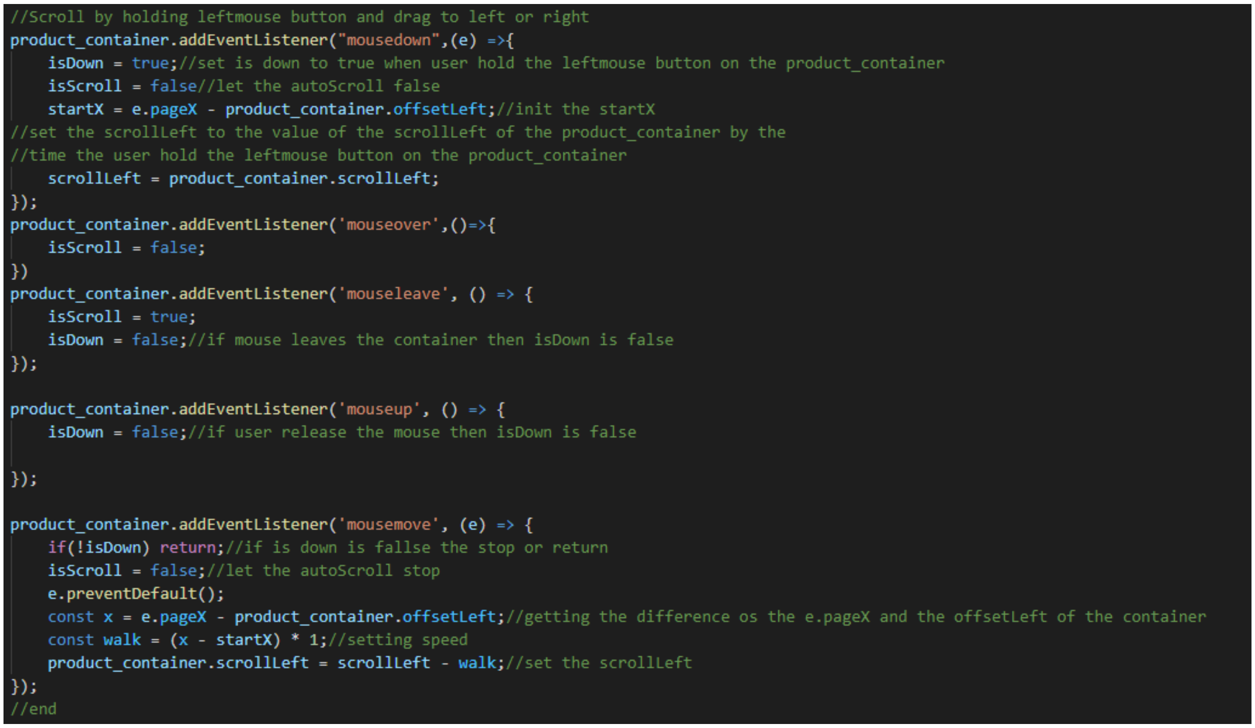
```
constructor(elementID,products){
    this.products=products;
    this.elementID = elementID;
    let product_container = document.getElementById(this.elementID);//get the container where you want to put the cards
    let card = product_container.getElementsByClassName("card");

    let speed = 1000/60;//the speed of the scroll which is 60 frames/s
    let isScroll = true;//use to indicate if autoScroll can scroll or not
    let isDown = false;//use to indicate if the mouse is pressing the product container
    let startX;//use for scrolling using mouse
    let scrollLeft;//this will store the value of the current scrollValue when mousedown
    let productList = [];//list of product objects
    let forward = true;//indicator if the auto scroll is scrolling forward or else
    let lastScroll = 0;//value for the last scroll before current scroll value
    products.forEach((value) =>{// adding produc card
        productList.push(value);//adding this prodcut to the array
        product_container.append(value.create());//then adding to the element
    })
```

This code is responsible for stopping the automatic scroll when the user hover a product card



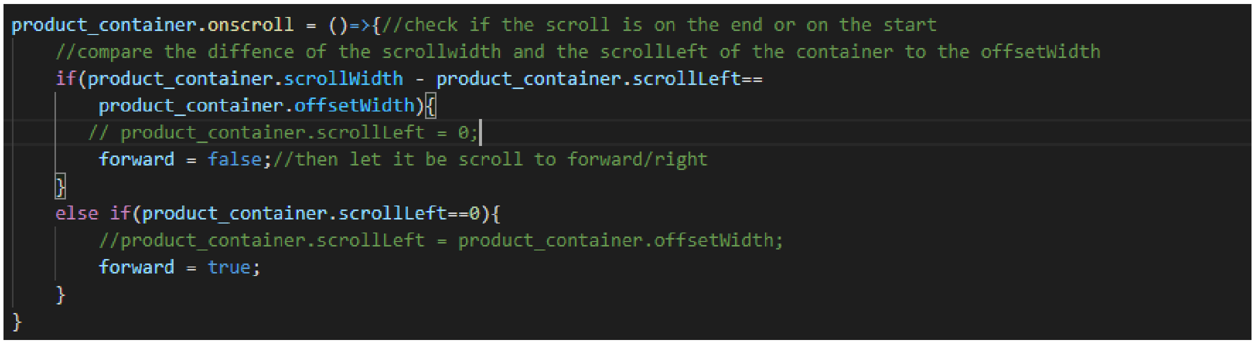
```
productList.forEach((value,index)=>{//init for cards EventListener
    // card[index].onclick = ()=>{
    //     console.log(value.id);
    // }
    card[index].addEventListener("mouseover",(e) =>{//if the mouse is ontop of the card the stop autoScroll
        isScroll = false;
    });
    card[index].addEventListener("mouseleave",(e) =>{//else continue
        isScroll = true;
    });
});
```

This code is responsible for dragging left an right



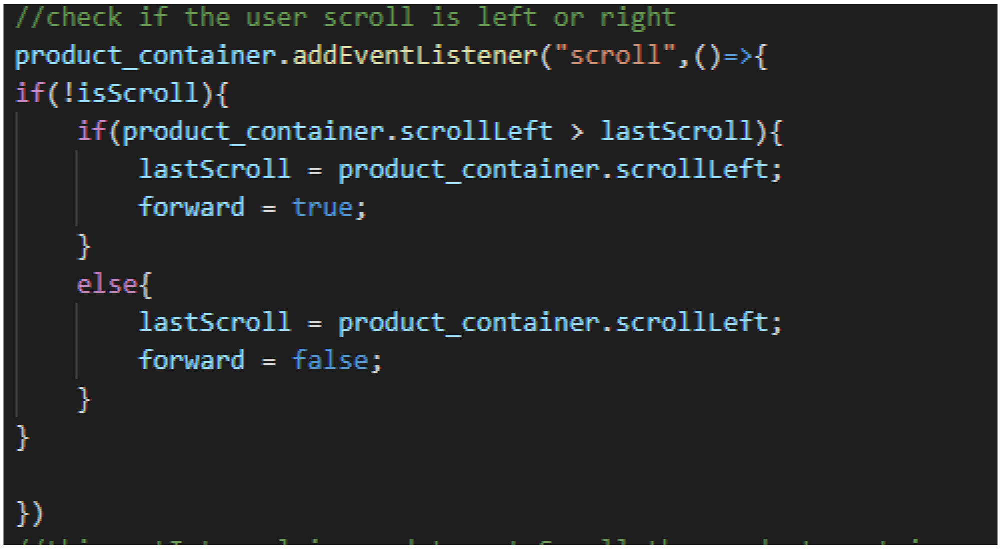
```
//Scroll by holding leftmouse button and drag to left or right
product_container.addEventListener("mousedown",(e) =>{
    isDown = true;//set is down to true when user hold the leftmouse button on the product_container
    isScroll = false//let the autoScroll false
    startX = e.pageX - product_container.offsetLeft;//init the startX
//set the scrollLeft to the value of the scrollLeft of the product_container by the
//time the user hold the leftmouse button on the product_container
    scrollLeft = product_container.scrollLeft;
});
product_container.addEventListener('mouseover',()=>{
    isScroll = false;
})
product_container.addEventListener('mouseleave', () => {
    isScroll = true;
    isDown = false;//if mouse leaves the container then isDown is false
});

product_container.addEventListener('mouseup', () => {
    isDown = false;//if user release the mouse then isDown is false

});

product_container.addEventListener('mousemove', (e) => {
    if(!isDown) return;//if is down is fallse the stop or return
    isScroll = false;//let the autoScroll stop
    e.preventDefault();
    const x = e.pageX - product_container.offsetLeft;//getting the difference os the e.pageX and the offsetLeft of the container
    const walk = (x - startX) * 1;//setting speed
    product_container.scrollLeft = scrollLeft - walk;//set the scrollLeft
});
//end
```

Checking if the user scrolled on end or start, if end then autoscroll move forward, if not then opposite



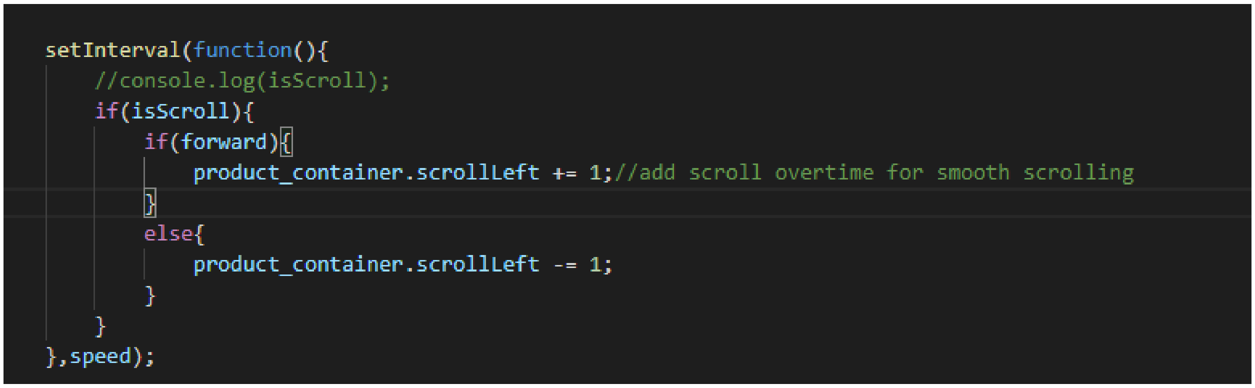
```
product_container.onscroll = ()=>{//check if the scroll is on the end or on the start
    //compare the diffence of the scrollwidth and the scrollLeft of the container to the offsetWidth
    if(product_container.scrollWidth - product_container.scrollLeft==
        product_container.offsetWidth){
       // product_container.scrollLeft = 0;
        forward = false;//then let it be scroll to forward/right
    }
    else if(product_container.scrollLeft==0){
        //product_container.scrollLeft = product_container.offsetWidth;
        forward = true;
    }
}
```

Checking if the user is scrolling to left or right, if right then auto scroll go right else left



```
//check if the user scroll is left or right
product_container.addEventListener("scroll",()=>{
if(!isScroll){
    if(product_container.scrollLeft > lastScroll){
        lastScroll = product_container.scrollLeft;
        forward = true;
    }
    else{
        lastScroll = product_container.scrollLeft;
        forward = false;
    }
}

})
```

Responsible for autoscrolling using setInterval. For each delta time passed then if forward product_container.scrollLeft add 1 else subtract 1

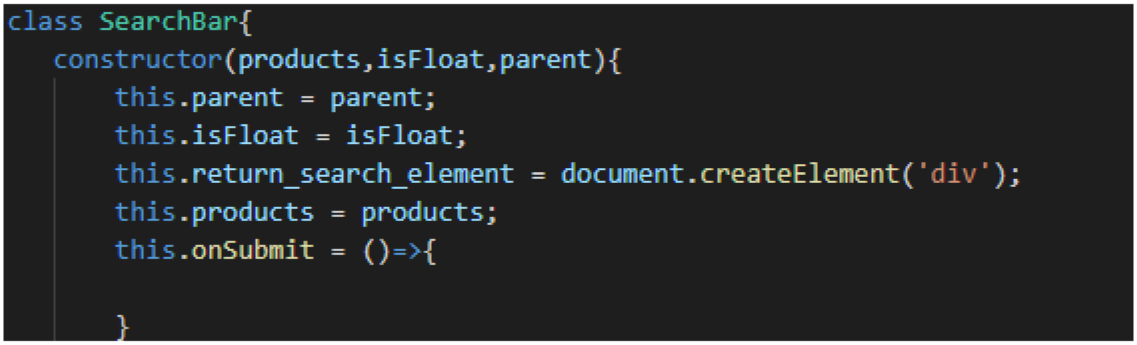


```
setInterval(function(){
    //console.log(isScroll);
    if(isScroll){
        if(forward){
            product_container.scrollLeft += 1;//add scroll overtime for smooth scrolling
        }
        else{
            product_container.scrollLeft -= 1;
        }
    }
},speed);
```

searchbar.js

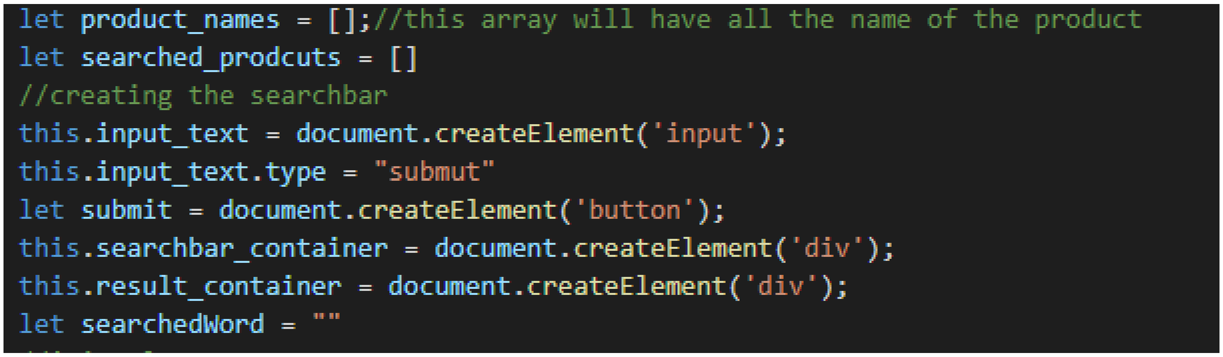
This JavaScript file is responsible for searching item on the system.

SearchBar class – Creating a constructor



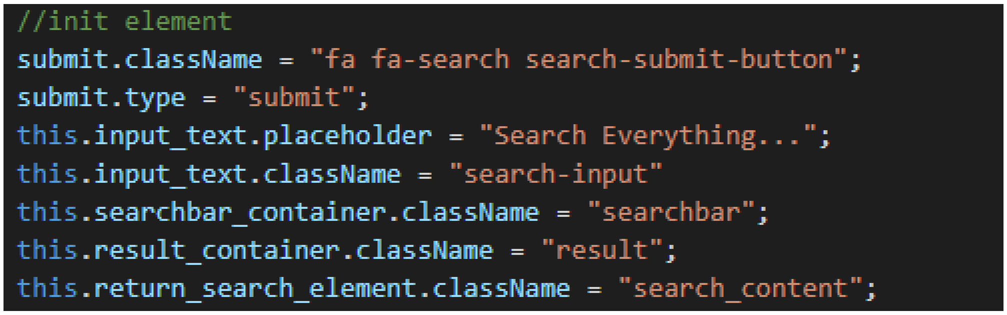
```
class SearchBar{
   constructor(products,isFloat,parent){
       this.parent = parent;
       this.isFloat = isFloat;
       this.return_search_element = document.createElement('div');
       this.products = products;
       this.onSubmit = ()=>{

       }
```

Creating search bar elements



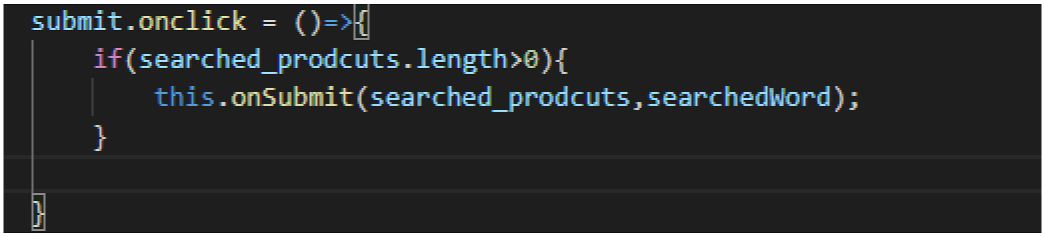
```
let product_names = [];//this array will have all the name of the product
let searched_prodcuts = []
//creating the searchbar
this.input_text = document.createElement('input');
this.input_text.type = "submut"
let submit = document.createElement('button');
this.searchbar_container = document.createElement('div');
this.result_container = document.createElement('div');
let searchedWord = ""
```

Init elements



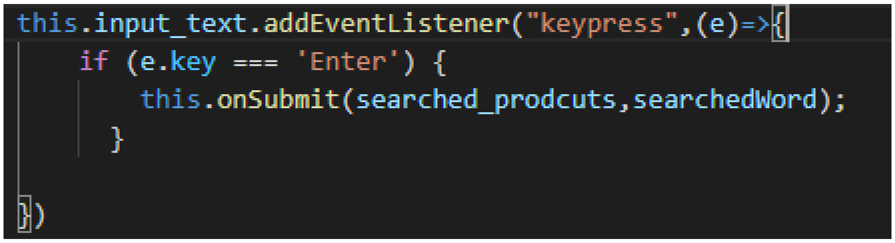
```
//init element
submit.className = "fa fa-search search-submit-button";
submit.type = "submit";
this.input_text.placeholder = "Search Everything...";
this.input_text.className = "search-input"
this.searchbar_container.className = "searchbar";
this.result_container.className = "result";
this.return_search_element.className = "search_content";
```

Adding on click on submit button(Search Icon) and calling the call back function onSubmit



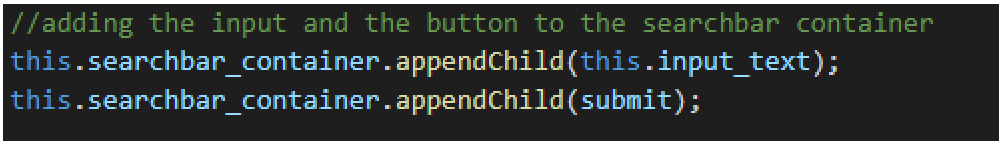
```
 submit.onclick = ()=>{
     if(searched_prodcuts.length>0){
         this.onSubmit(searched_prodcuts,searchedWord);
     }

 }
```

Check if the user hit enter



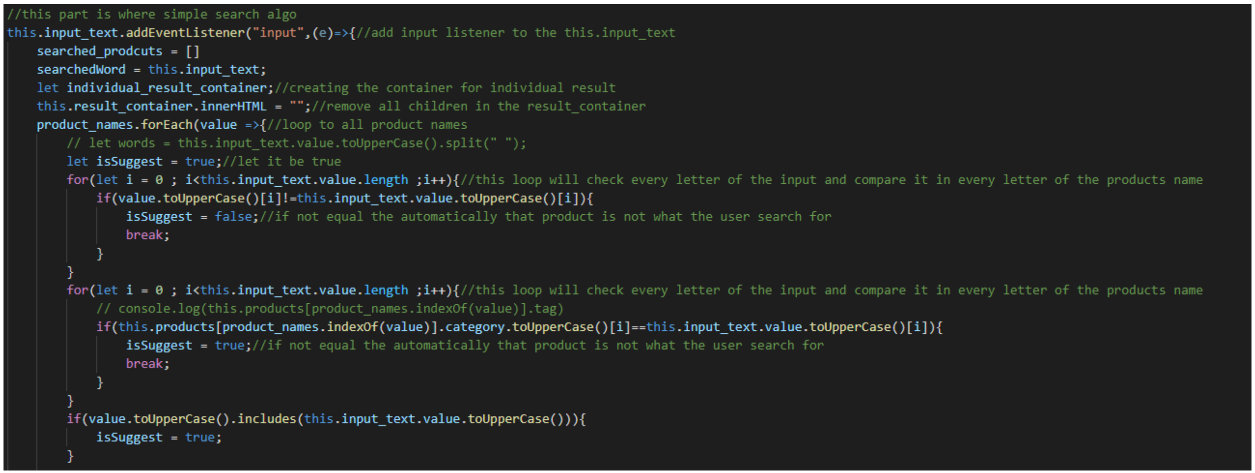
```
this.input_text.addEventListener("keypress",(e)=>{
    if (e.key === 'Enter') {
        this.onSubmit(searched_prodcuts,searchedWord);
     }

})
```

Adding the search input text and the search icon to the searchbar container

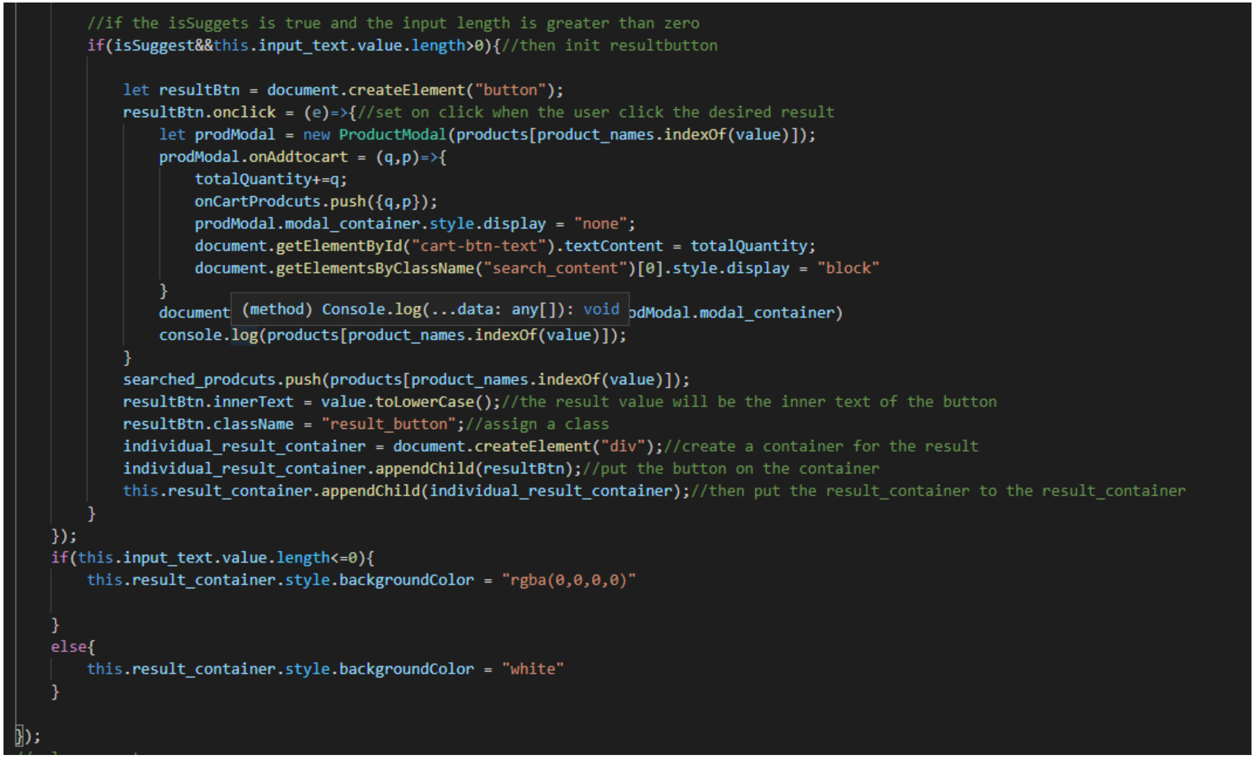


```
//adding the input and the button to the searchbar container
this.searchbar_container.appendChild(this.input_text);
this.searchbar_container.appendChild(submit);
```

This is code is responsible for search suggestions and on click suggested product

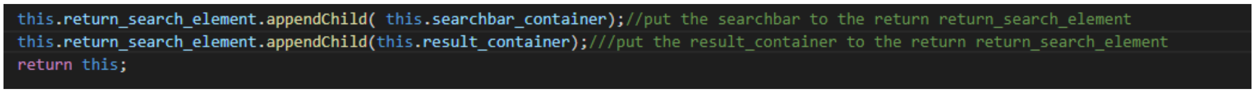


```
//this part is where simple search algo
this.input_text.addEventListener("input",(e)=>{//add input listener to the this.input_text
    searched_prodcuts = []
    searchedWord = this.input_text;
    let individual_result_container;//creating the container for individual result
    this.result_container.innerHTML = "";//remove all children in the result_container
    product_names.forEach(value =>{//loop to all product names
        // let words = this.input_text.value.toUpperCase().split(" ");
        let isSuggest = true;//let it be true
        for(let i = 0 ; i<this.input_text.value.length ;i++){//this loop will check every letter of the input and compare it in every letter of the products name
            if(value.toUpperCase()[i]!=this.input_text.value.toUpperCase()[i]){
                isSuggest = false;//if not equal the automatically that product is not what the user search for
                break;
            }
        }
        for(let i = 0 ; i<this.input_text.value.length ;i++){//this loop will check every letter of the input and compare it in every letter of the products name
            // console.log(this.products[product_names.indexOf(value)].tag)
            if(this.products[product_names.indexOf(value)].category.toUpperCase()[i]==this.input_text.value.toUpperCase()[i]){
                isSuggest = true;//if not equal the automatically that product is not what the user search for
                break;
            }
        }
        if(value.toUpperCase().includes(this.input_text.value.toUpperCase())){
            isSuggest = true;
        }
```



```
        //if the isSuggets is true and the input length is greater than zero
        if(isSuggest&&this.input_text.value.length>0){//then init resultbutton

            let resultBtn = document.createElement("button");
            resultBtn.onclick = (e)=>{//set on click when the user click the desired result
                let prodModal = new ProductModal(products[product_names.indexOf(value)]);
                prodModal.onAddtocart = (q,p)=>{
                    totalQuantity+=q;
                    onCartProdcuts.push({q,p});
                    prodModal.modal_container.style.display = "none";
                    document.getElementById("cart-btn-text").textContent = totalQuantity;
                    document.getElementsByClassName("search_content")[0].style.display = "block"
                }
                document (method) Console.log(...data: any[]): void odModal.modal_container)
                console.log(products[product_names.indexOf(value)]);
            }
            searched_prodcuts.push(products[product_names.indexOf(value)]);
            resultBtn.innerText = value.toLowerCase();//the result value will be the inner text of the button
            resultBtn.className = "result_button";//assign a class
            individual_result_container = document.createElement("div");//create a container for the result
            individual_result_container.appendChild(resultBtn);//put the button on the container
            this.result_container.appendChild(individual_result_container);//then put the result_container to the result_container
        }
    });
    if(this.input_text.value.length<=0){
        this.result_container.style.backgroundColor = "rgba(0,0,0,0)"

    }
    else{
        this.result_container.style.backgroundColor = "white"
    }

});
```

Adding the searchbar_container and result container to the return_search_element and return this. Therefore you can add this Class to anything using append or appendChild

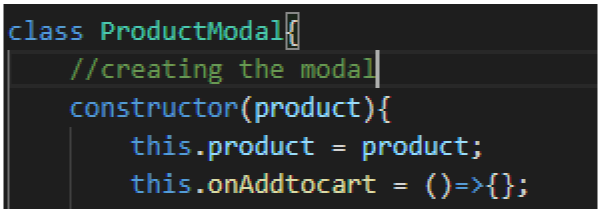


```
this.return_search_element.appendChild( this.searchbar_container);//put the searchbar to the return return_search_element
this.return_search_element.appendChild(this.result_container);///put the result_container to the return return_search_element
return this;
```

modal.js

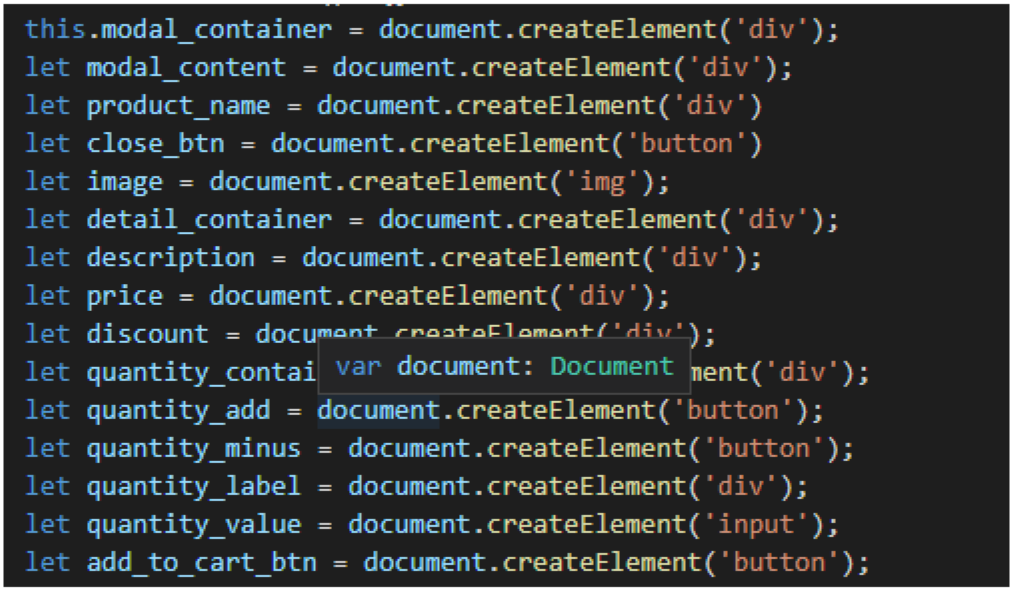
This JavaScript file is responsible for creating modal for product and product cart.

ProductModal class – Creating a constructor



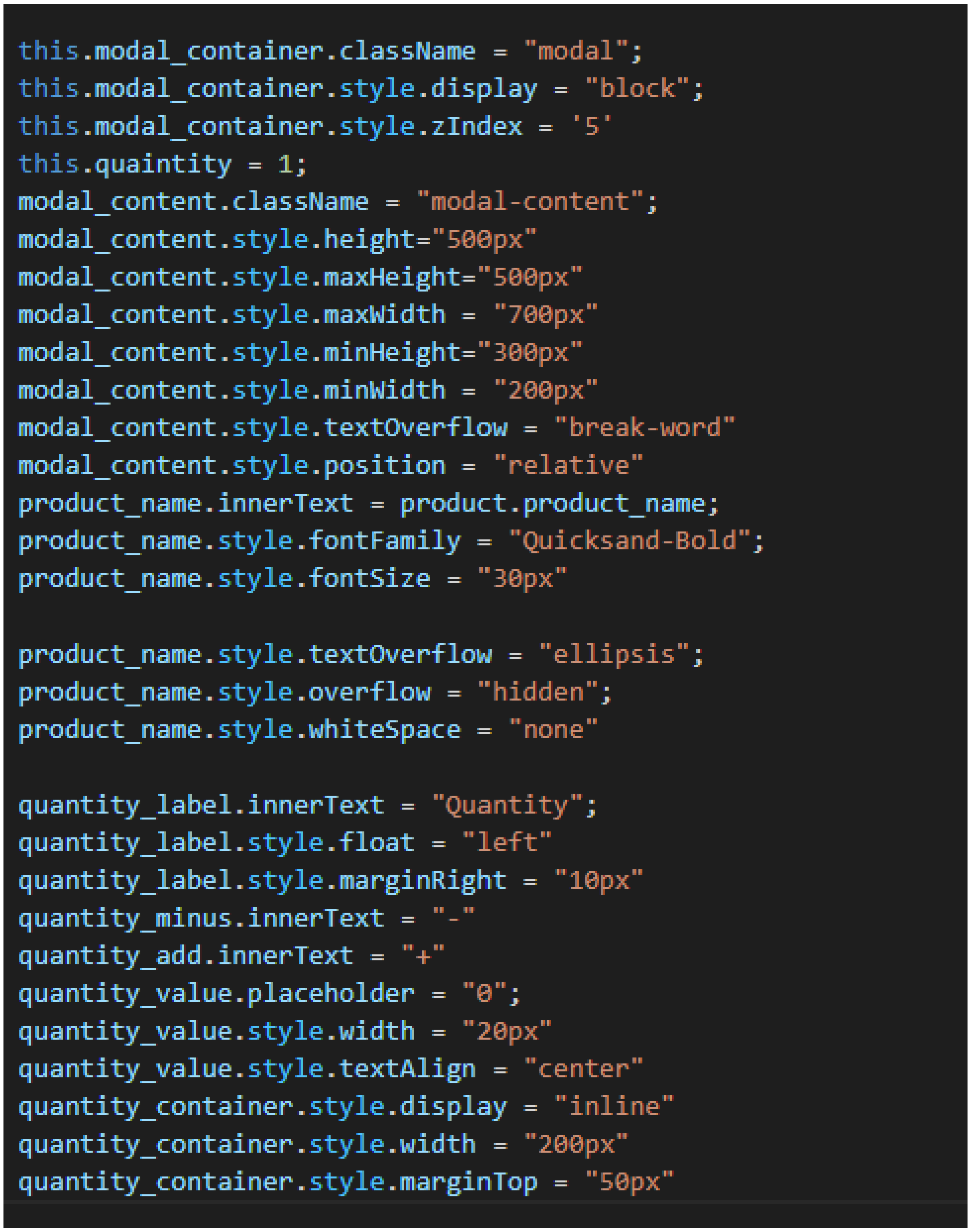
```
class ProductModal{
    //creating the modal
    constructor(product){
        this.product = product;
        this.onAddtocart = ()=>{};
```

Creating elements responsible for displaying modal



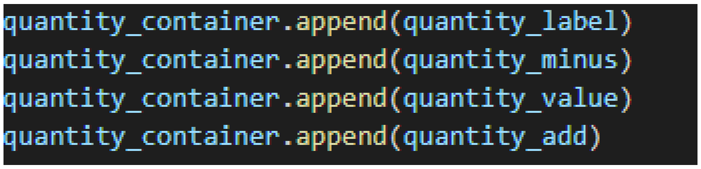
```
 this.modal_container = document.createElement('div');
 let modal_content = document.createElement('div');
 let product_name = document.createElement('div')
 let close_btn = document.createElement('button')
 let image = document.createElement('img');
 let detail_container = document.createElement('div');
 let description = document.createElement('div');
 let price = document.createElement('div');
 let discount = document.createElement('div');
 let quantity_contai var document: Document ment('div');
 let quantity_add = document.createElement('button');
 let quantity_minus = document.createElement('button');
 let quantity_label = document.createElement('div');
 let quantity_value = document.createElement('input');
 let add_to_cart_btn = document.createElement('button');
```

Setting up styles for each elements



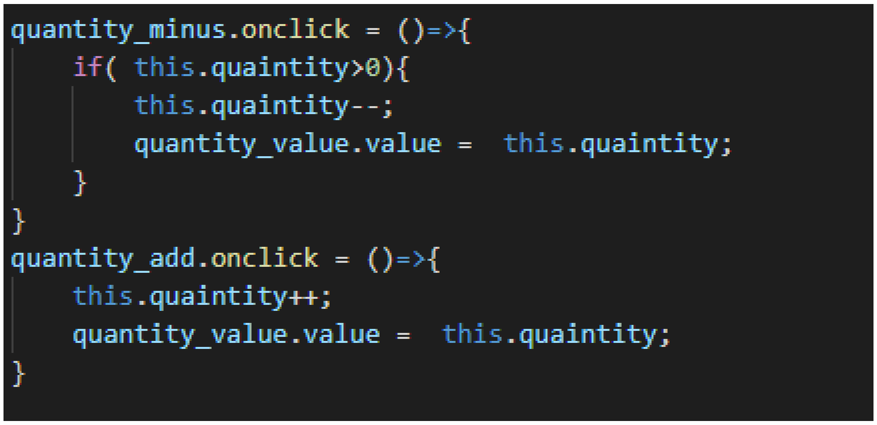
```
this.modal_container.className = "modal";
this.modal_container.style.display = "block";
this.modal_container.style.zIndex = '5'
this.quaintity = 1;
modal_content.className = "modal-content";
modal_content.style.height="500px"
modal_content.style.maxHeight="500px"
modal_content.style.maxWidth = "700px"
modal_content.style.minHeight="300px"
modal_content.style.minWidth = "200px"
modal_content.style.textOverflow = "break-word"
modal_content.style.position = "relative"
product_name.innerText = product.product_name;
product_name.style.fontFamily = "Quicksand-Bold";
product_name.style.fontSize = "30px"

product_name.style.textOverflow = "ellipsis";
product_name.style.overflow = "hidden";
product_name.style.whiteSpace = "none"

quantity_label.innerText = "Quantity";
quantity_label.style.float = "left"
quantity_label.style.marginRight = "10px"
quantity_minus.innerText = "-"
quantity_add.innerText = "+"
quantity_value.placeholder = "0";
quantity_value.style.width = "20px"
quantity_value.style.textAlign = "center"
quantity_container.style.display = "inline"
quantity_container.style.width = "200px"
quantity_container.style.marginTop = "50px"
```

Adding some element to the quantity container- responsible for displaying quantity on the modal



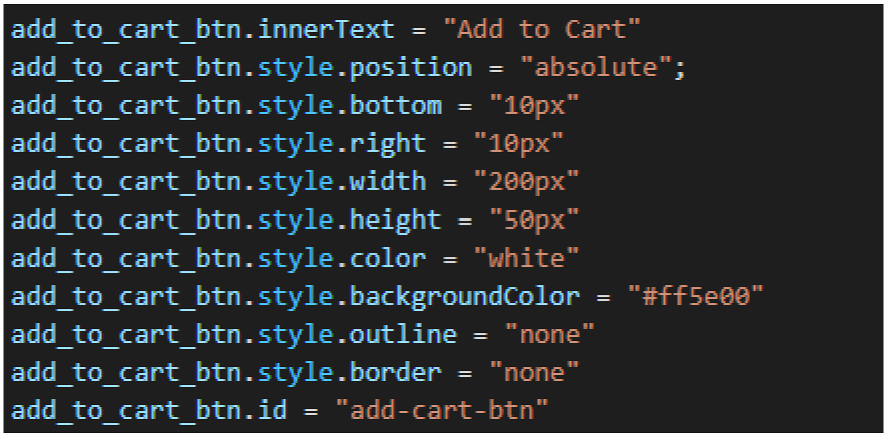
```
quantity_container.append(quantity_label)
quantity_container.append(quantity_minus)
quantity_container.append(quantity_value)
quantity_container.append(quantity_add)
```

adding function to add quantity and subtract



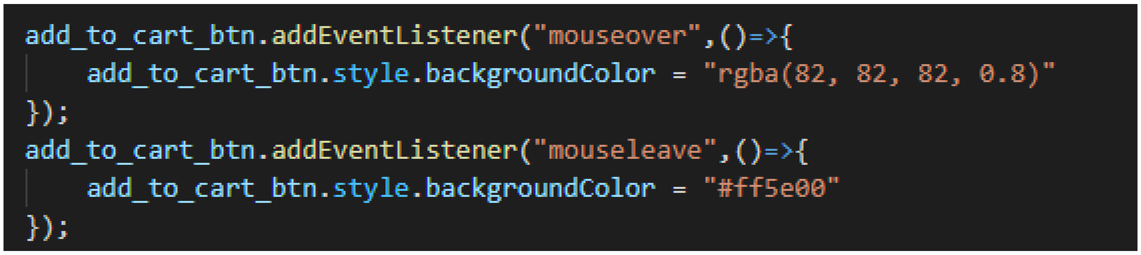
```
quantity_minus.onclick = ()=>{
    if( this.quaintity>0){
        this.quaintity--;
        quantity_value.value =  this.quaintity;
    }
}
quantity_add.onclick = ()=>{
    this.quaintity++;
    quantity_value.value =  this.quaintity;
}
```

Setting up style for add to cart button



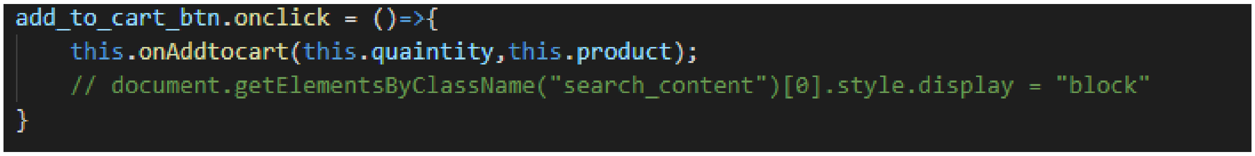
```
add_to_cart_btn.innerText = "Add to Cart"
add_to_cart_btn.style.position = "absolute";
add_to_cart_btn.style.bottom = "10px"
add_to_cart_btn.style.right = "10px"
add_to_cart_btn.style.width = "200px"
add_to_cart_btn.style.height = "50px"
add_to_cart_btn.style.color = "white"
add_to_cart_btn.style.backgroundColor = "#ff5e00"
add_to_cart_btn.style.outline = "none"
add_to_cart_btn.style.border = "none"
add_to_cart_btn.id = "add-cart-btn"
```

adding mouse hover to add to cart button



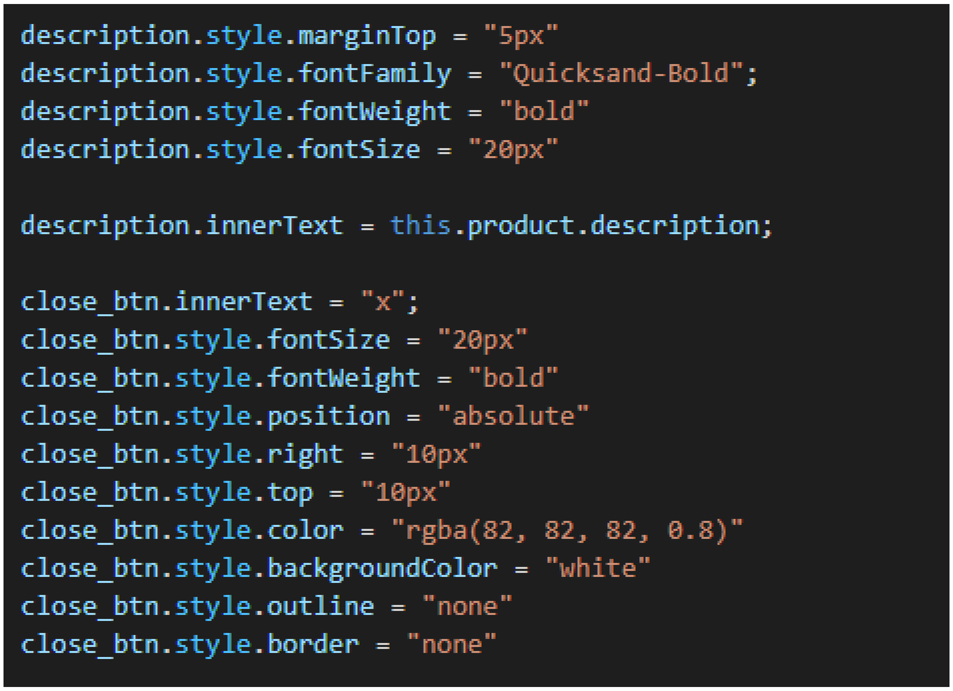
```
add_to_cart_btn.addEventListener("mouseover",()=>{
    add_to_cart_btn.style.backgroundColor = "rgba(82, 82, 82, 0.8)"
});
add_to_cart_btn.addEventListener("mouseleave",()=>{
    add_to_cart_btn.style.backgroundColor = "#ff5e00"
});
```

adding a call back function to add to cart button



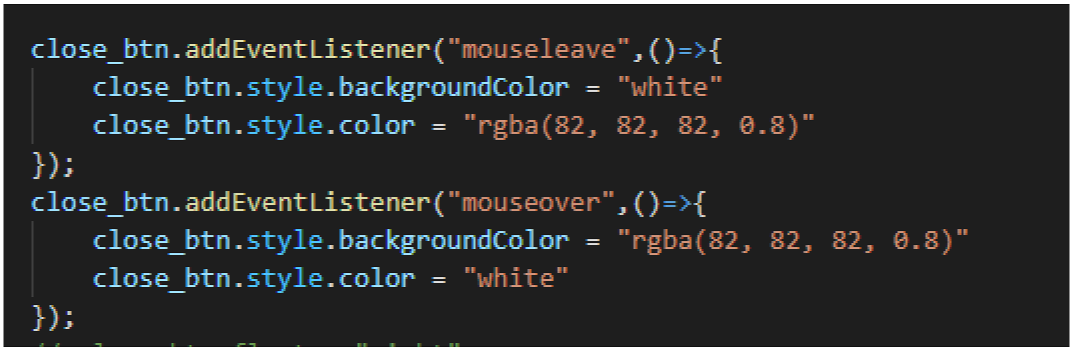
```
add_to_cart_btn.onclick = ()=>{
    this.onAddtocart(this.quaintity,this.product);
    // document.getElementsByClassName("search_content")[0].style.display = "block"
}
```

Adding style to descripton and close button



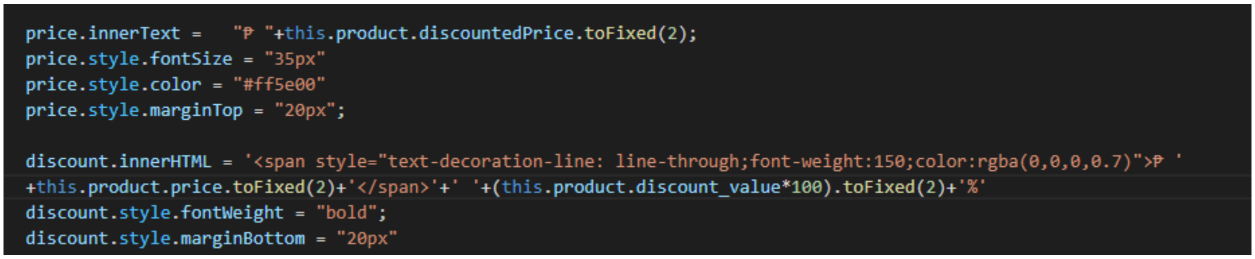
```
description.style.marginTop = "5px"
description.style.fontFamily = "Quicksand-Bold";
description.style.fontWeight = "bold"
description.style.fontSize = "20px"

description.innerText = this.product.description;

close_btn.innerText = "x";
close_btn.style.fontSize = "20px"
close_btn.style.fontWeight = "bold"
close_btn.style.position = "absolute"
close_btn.style.right = "10px"
close_btn.style.top = "10px"
close_btn.style.color = "rgba(82, 82, 82, 0.8)"
close_btn.style.backgroundColor = "white"
close_btn.style.outline = "none"
close_btn.style.border = "none"
```

Adding mouse leave and mouseover in close button



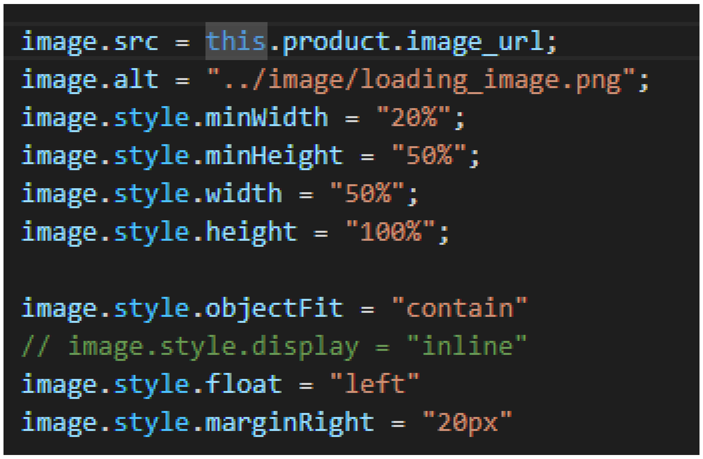
```
close_btn.addEventListener("mouseleave",()=>{
    close_btn.style.backgroundColor = "white"
    close_btn.style.color = "rgba(82, 82, 82, 0.8)"
});
close_btn.addEventListener("mouseover",()=>{
    close_btn.style.backgroundColor = "rgba(82, 82, 82, 0.8)"
    close_btn.style.color = "white"
});
```

Adding style to discount text



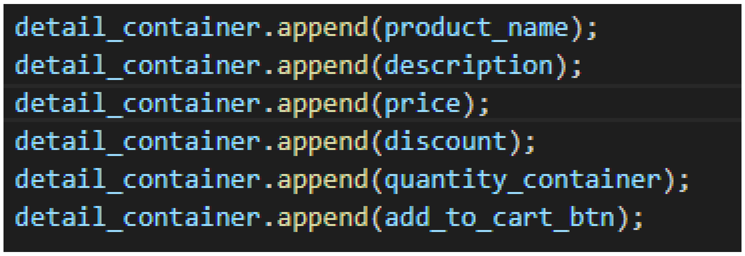
```
price.innerText =   "₱ "+this.product.discountedPrice.toFixed(2);
price.style.fontSize = "35px"
price.style.color = "#ff5e00"
price.style.marginTop = "20px";

discount.innerHTML = '<span style="text-decoration-line: line-through;font-weight:150;color:rgba(0,0,0,0.7)">₱ '
+this.product.price.toFixed(2)+'</span>'+' '+(this.product.discount_value*100).toFixed(2)+'%'
discount.style.fontWeight = "bold";
discount.style.marginBottom = "20px"
```

Adding style to image



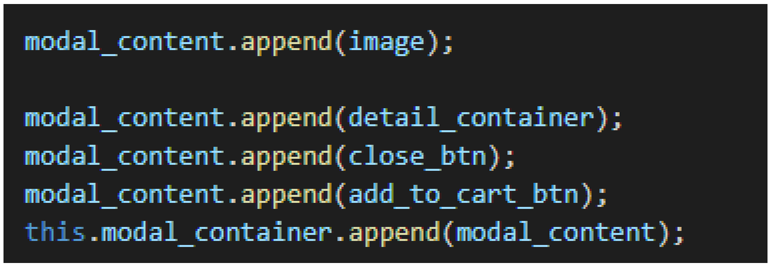
```
image.src = this.product.image_url;
image.alt = "../image/loading_image.png";
image.style.minWidth = "20%";
image.style.minHeight = "50%";
image.style.width = "50%";
image.style.height = "100%";

image.style.objectFit = "contain"
// image.style.display = "inline"
image.style.float = "left"
image.style.marginRight = "20px"
```

Adding elements to detail_container



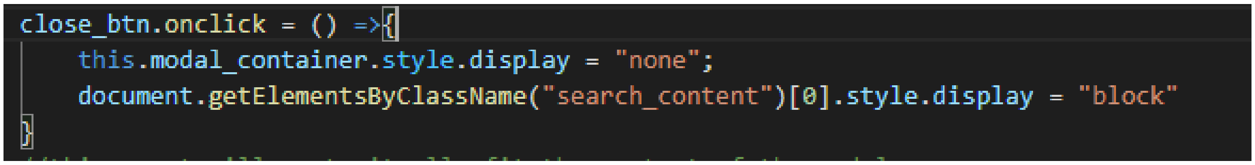
```
detail_container.append(product_name);
detail_container.append(description);
detail_container.append(price);
detail_container.append(discount);
detail_container.append(quantity_container);
detail_container.append(add_to_cart_btn);
```

Adding all the element to modal_content



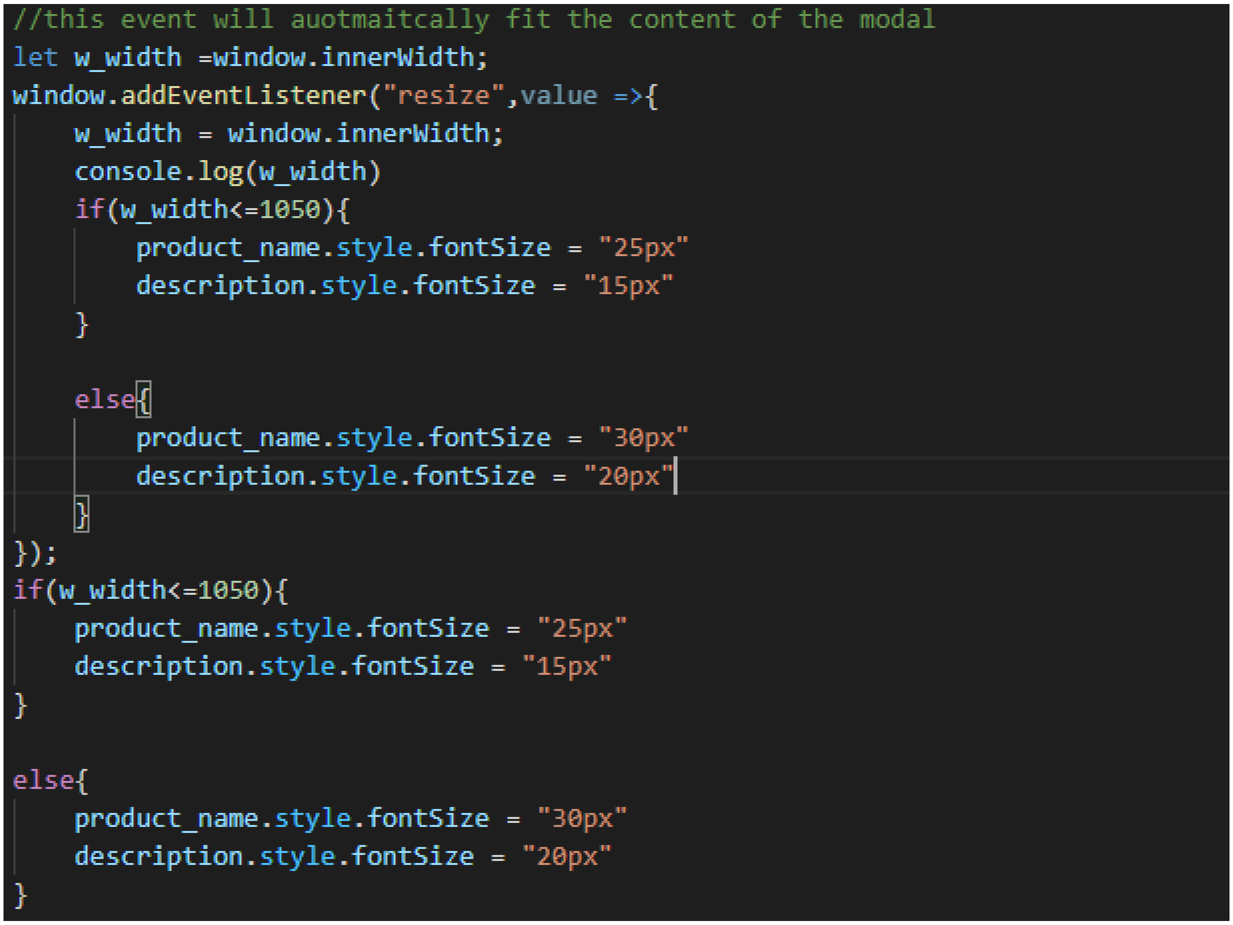
```
modal_content.append(image);

modal_content.append(detail_container);
modal_content.append(close_btn);
modal_content.append(add_to_cart_btn);
this.modal_container.append(modal_content);
```

adding on click function to close the modal



```
close_btn.onclick = () =>{
    this.modal_container.style.display = "none";
    document.getElementsByClassName("search_content")[0].style.display = "block"
}
```

This code is responsible for resizing the model regarding to the size of the window



```
//this event will auotmaitcally fit the content of the modal
let w_width =window.innerWidth;
window.addEventListener("resize",value =>{
    w_width = window.innerWidth;
    console.log(w_width)
    if(w_width<=1050){
        product_name.style.fontSize = "25px"
        description.style.fontSize = "15px"
    }

    else{
        product_name.style.fontSize = "30px"
        description.style.fontSize = "20px"
    }
});
if(w_width<=1050){
    product_name.style.fontSize = "25px"
    description.style.fontSize = "15px"
}

else{
    product_name.style.fontSize = "30px"
    description.style.fontSize = "20px"
}
```

Then return this class, therefore you can add this modal to any div by using append or appendChild

```
 return  this;
```

We also have cart modal which is responsible for the product cart, it is same as this but only all the product that the user add will be display.

This JavaScript file is responsible of storing product information.

clothing.js

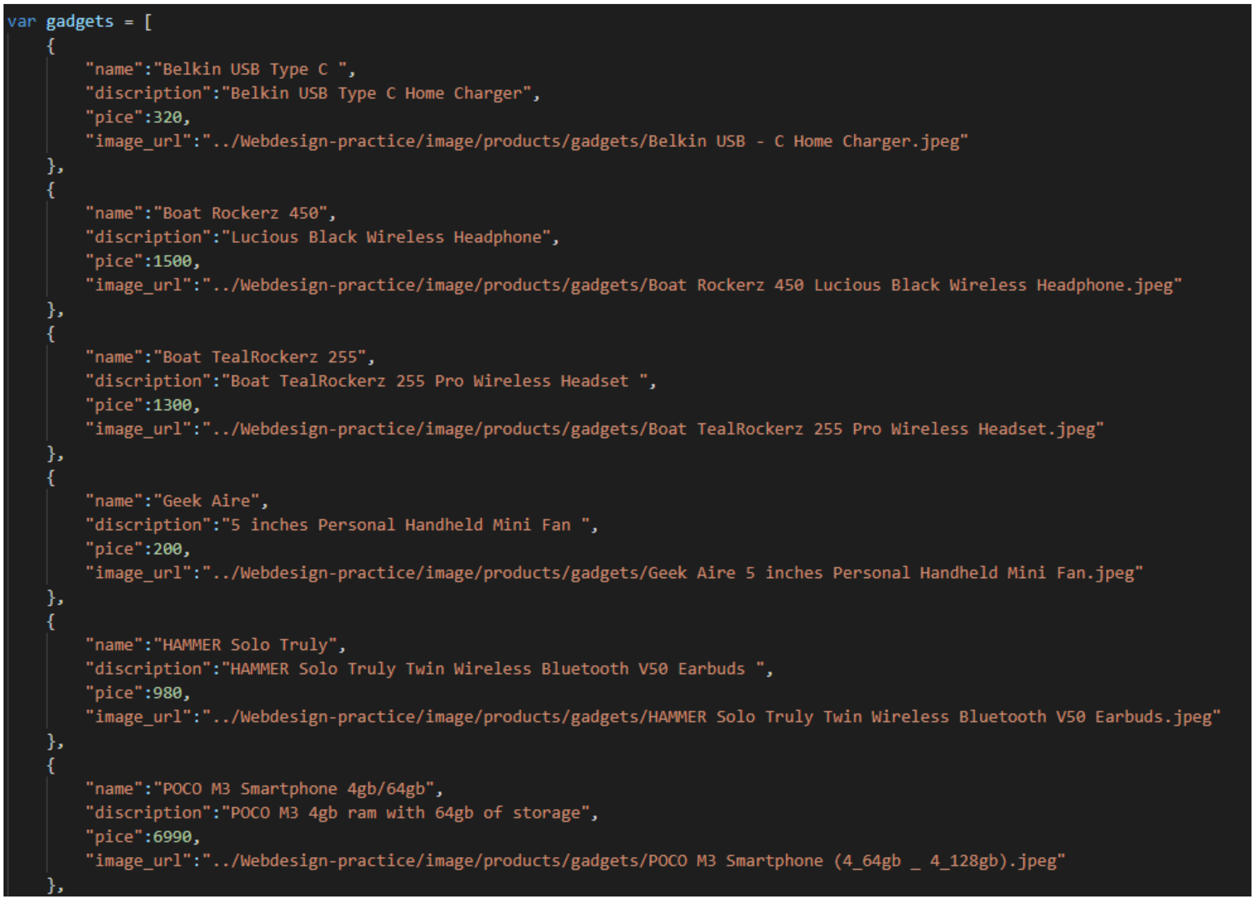
gadgets.js

home.js

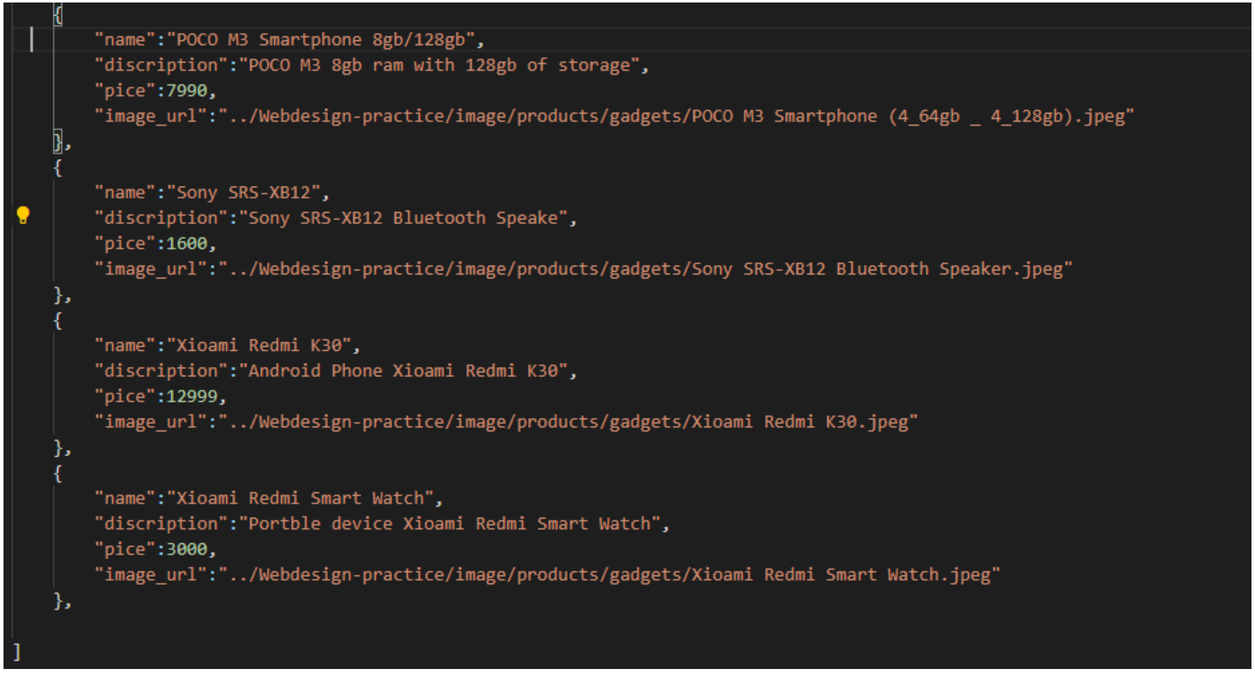
personal.js

This class is and array of jsons th has name,descripton, price and image url as a key

EXAMPLE – this is inside of gadget.js



```
var gadgets = [
    {
        "name":"Belkin USB Type C ",
        "discription":"Belkin USB Type C Home Charger",
        "pice":320,
        "image_url":"../Webdesign-practice/image/products/gadgets/Belkin USB - C Home Charger.jpeg"
    },
    {
        "name":"Boat Rockerz 450",
        "discription":"Lucious Black Wireless Headphone",
        "pice":1500,
        "image_url":"../Webdesign-practice/image/products/gadgets/Boat Rockerz 450 Lucious Black Wireless Headphone.jpeg"
    },
    {
        "name":"Boat TealRockerz 255",
        "discription":"Boat TealRockerz 255 Pro Wireless Headset ",
        "pice":1300,
        "image_url":"../Webdesign-practice/image/products/gadgets/Boat TealRockerz 255 Pro Wireless Headset.jpeg"
    },
    {
        "name":"Geek Aire",
        "discription":"5 inches Personal Handheld Mini Fan ",
        "pice":200,
        "image_url":"../Webdesign-practice/image/products/gadgets/Geek Aire 5 inches Personal Handheld Mini Fan.jpeg"
    },
    {
        "name":"HAMMER Solo Truly",
        "discription":"HAMMER Solo Truly Twin Wireless Bluetooth V50 Earbuds ",
        "pice":980,
        "image_url":"../Webdesign-practice/image/products/gadgets/HAMMER Solo Truly Twin Wireless Bluetooth V50 Earbuds.jpeg"
    },
    {
        "name":"POCO M3 Smartphone 4gb/64gb",
        "discription":"POCO M3 4gb ram with 64gb of storage",
        "pice":6990,
        "image_url":"../Webdesign-practice/image/products/gadgets/POCO M3 Smartphone (4_64gb _ 4_128gb).jpeg"
    },
```



```
    {
        "name":"POCO M3 Smartphone 8gb/128gb",
        "discription":"POCO M3 8gb ram with 128gb of storage",
        "pice":7990,
        "image_url":"../Webdesign-practice/image/products/gadgets/POCO M3 Smartphone (4_64gb _ 4_128gb).jpeg"
    },
    {
        "name":"Sony SRS-XB12",
        "discription":"Sony SRS-XB12 Bluetooth Speake",
        "pice":1600,
        "image_url":"../Webdesign-practice/image/products/gadgets/Sony SRS-XB12 Bluetooth Speaker.jpeg"
    },
    {
        "name":"Xioami Redmi K30",
        "discription":"Android Phone Xioami Redmi K30",
        "pice":12999,
        "image_url":"../Webdesign-practice/image/products/gadgets/Xioami Redmi K30.jpeg"
    },
    {
        "name":"Xioami Redmi Smart Watch",
        "discription":"Portble device Xioami Redmi Smart Watch",
        "pice":3000,
        "image_url":"../Webdesign-practice/image/products/gadgets/Xioami Redmi Smart Watch.jpeg"
    },

]
```