**index.js**

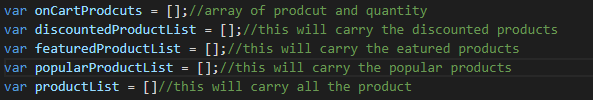
This JavaScript file is responsible of loading all the necessary element like the card, search bar and etc. It handles the layout of each element that is interactable like the category buttons and other interactable element in the DOM.

**Code:**

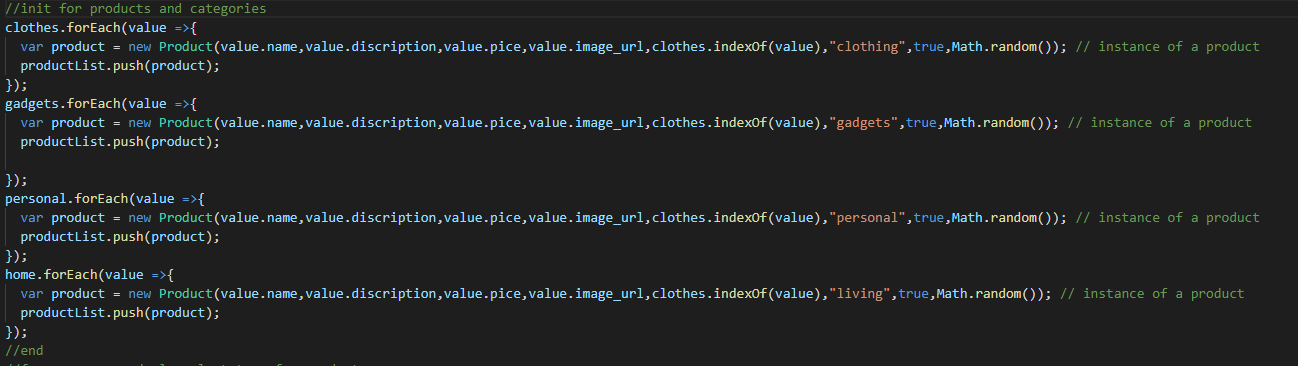
This code is temporary, this is for sorting the items in the home page.



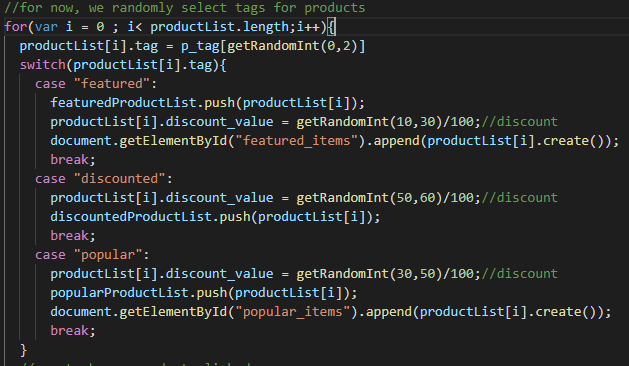
These arrays are responsible for storing a list of data.

****

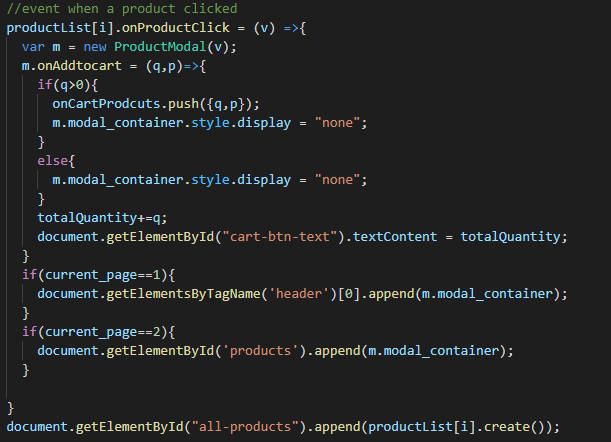
Initializing all product from respective categories and storing it in the productList.

****

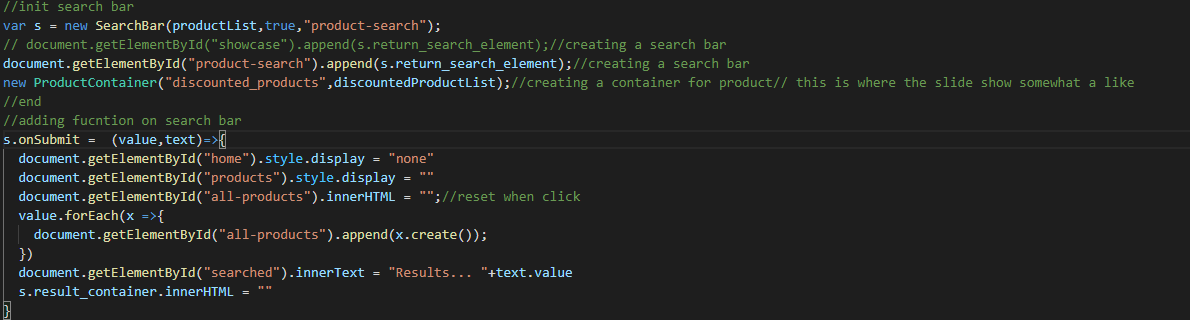
Loop a – This code is responsible for displaying Items with random tags



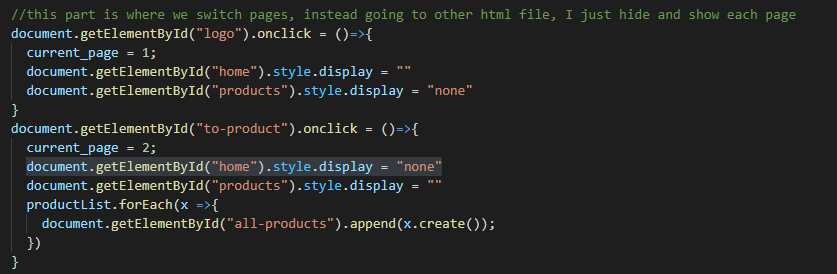
Loop b – after displaying the randomly tag items, this code will give each Card(item) a function on click  
and as you can see we created an instance of a ProductModal class for the item to pop when the user clicked the Card(item)



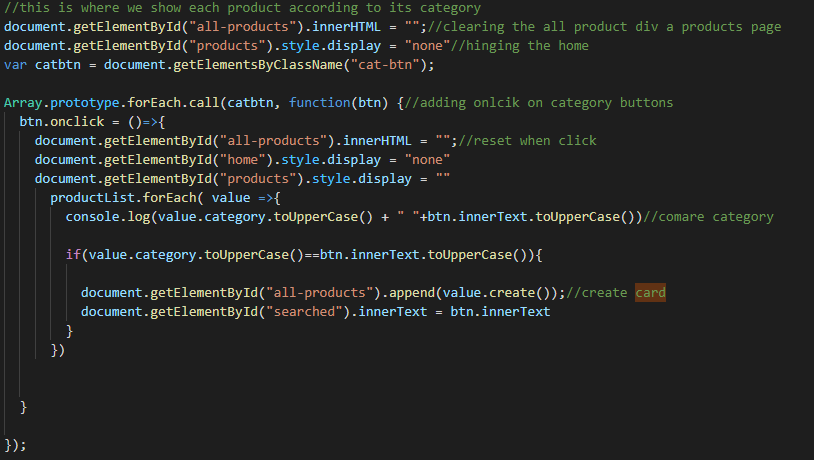
Creating an instance of SearchBar and display it in the DOM and adding onSubmit function.  
Inside onSubmit function, (it is where the user hit enter of click on the search icon) we change the display by hiding the home page and displaying the product page and adding all the search product on the all-products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



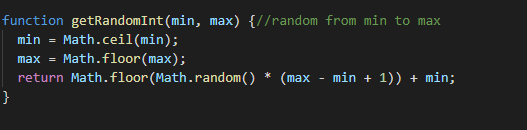
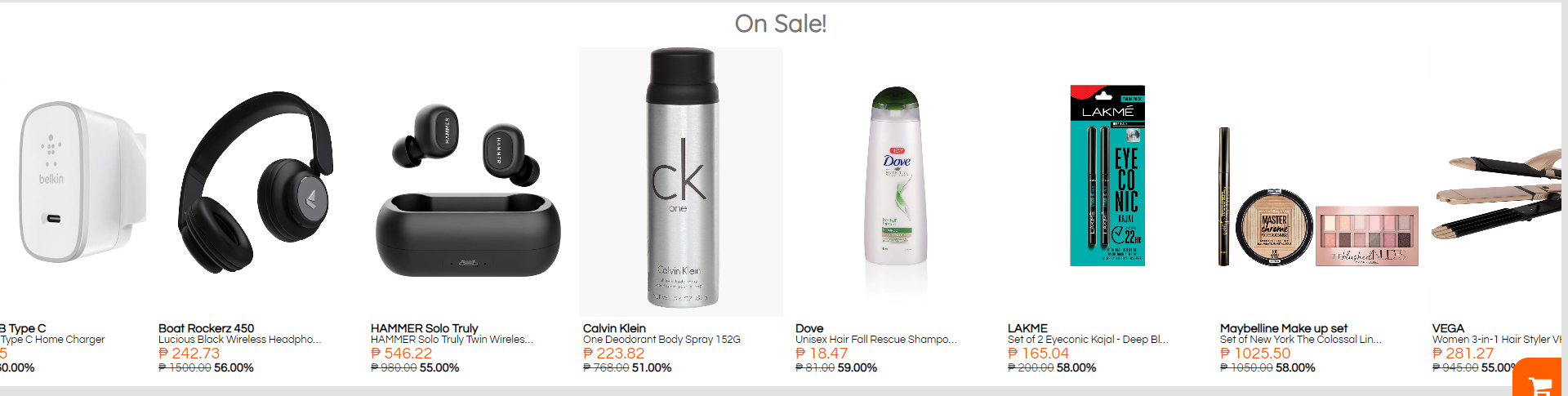
This codes is responsible for hiding and showing the homepage and the productpage



Displaying each product when the user clicked specific category

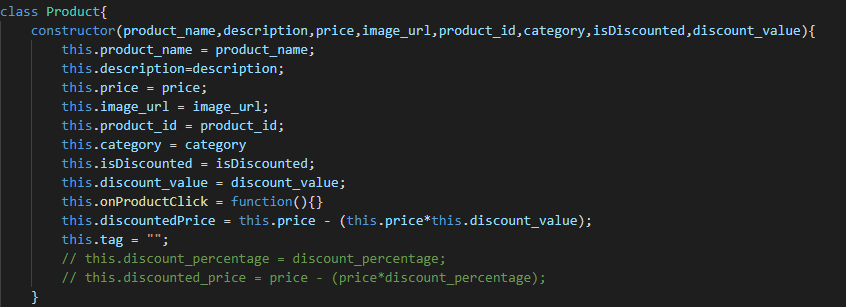


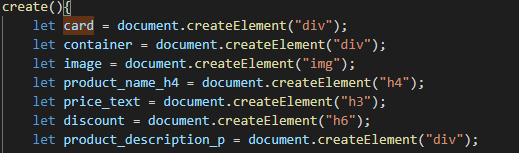
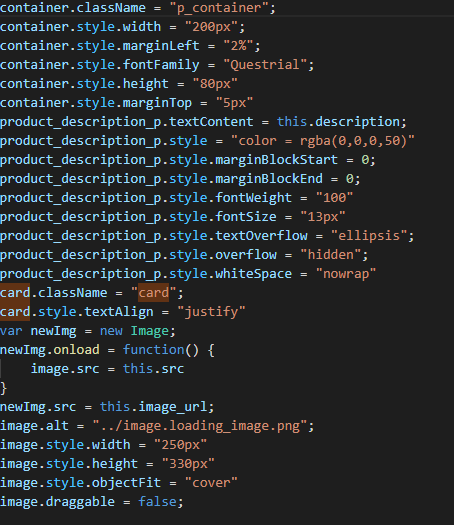
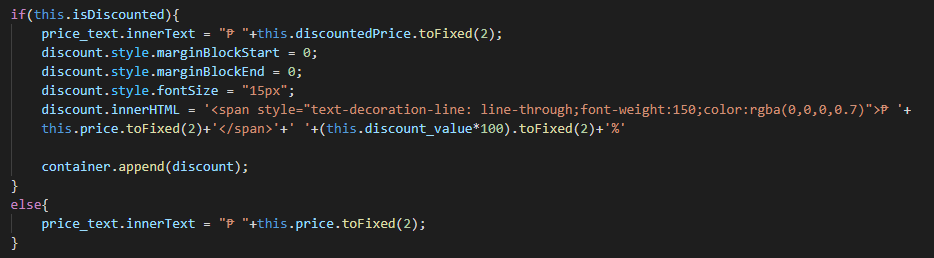
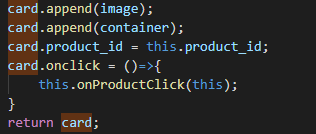
Just a random function that randomize a number from number min to number max

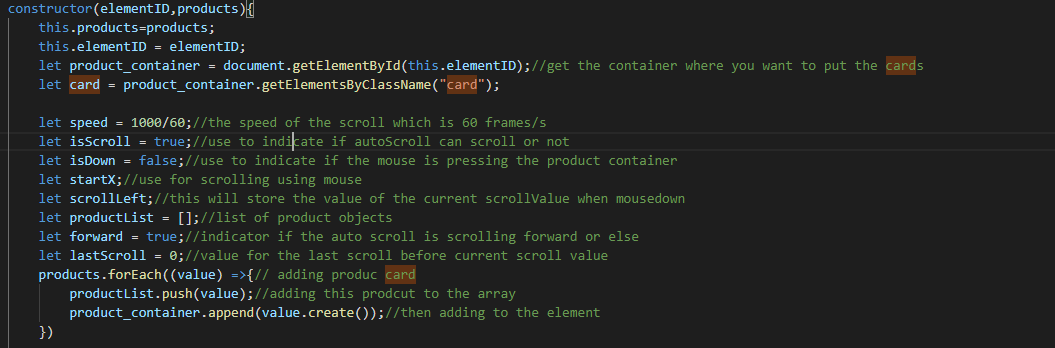
  
**product.js**This JavaScript file is responsible for the Product class and the product card. We use this a blueprint for our products information such as name, description, price, image\_url, id, category and discount value. We also have the function that is responsible for creating a card for our item.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


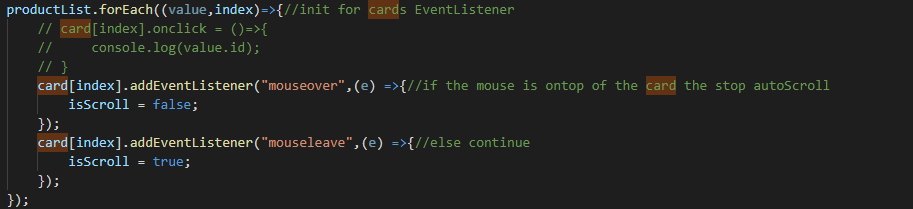
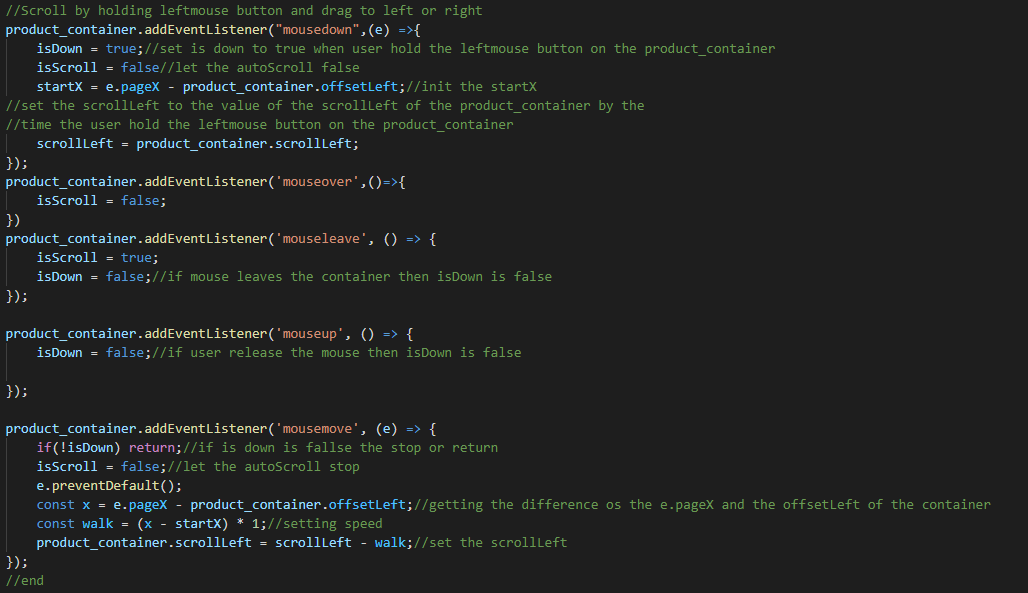
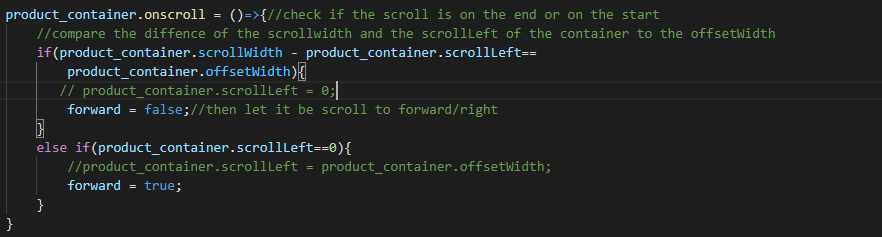
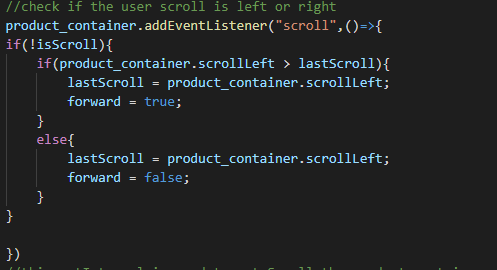
**Code:**

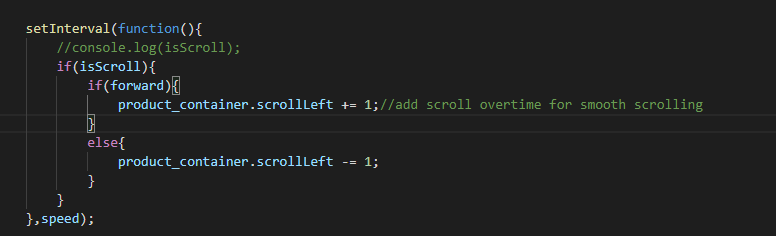
*Product class – Creating a constructor*



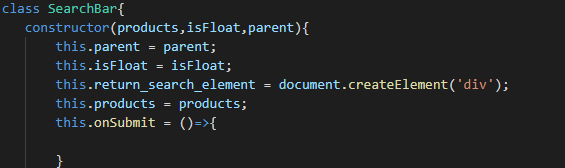
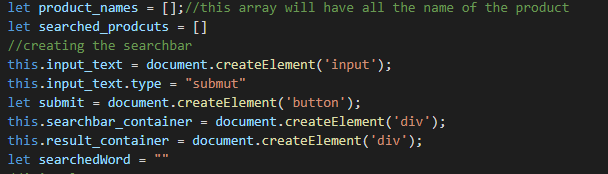
Product class - create() –  
Creating elements for creating card for our product  
  
Setting attributes  
  
  
  
Adding elements to the container  
  
Setting discount text for the product  
  
Adding the image and the container to the card div element and return it .  
Therefore, you can call this 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.  


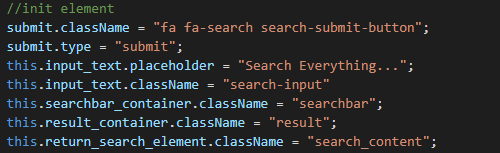
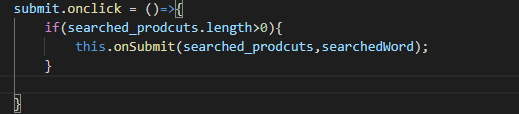
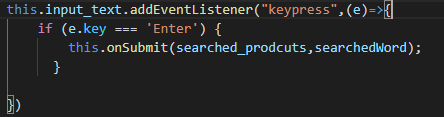
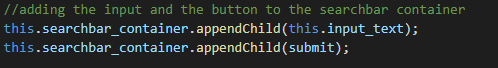
*ProductContainer class – Creating a constructor   
*

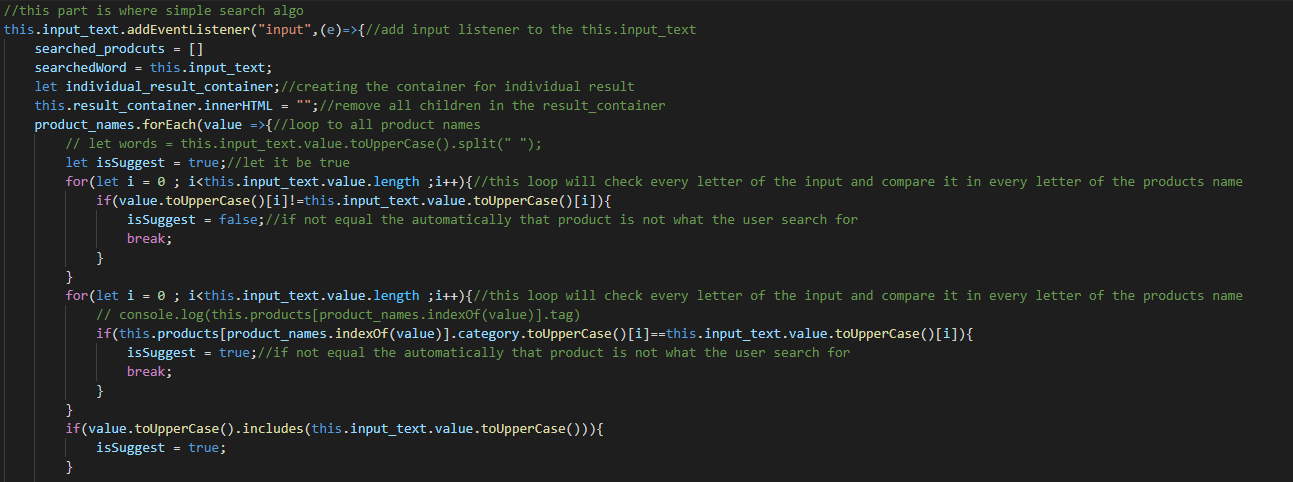
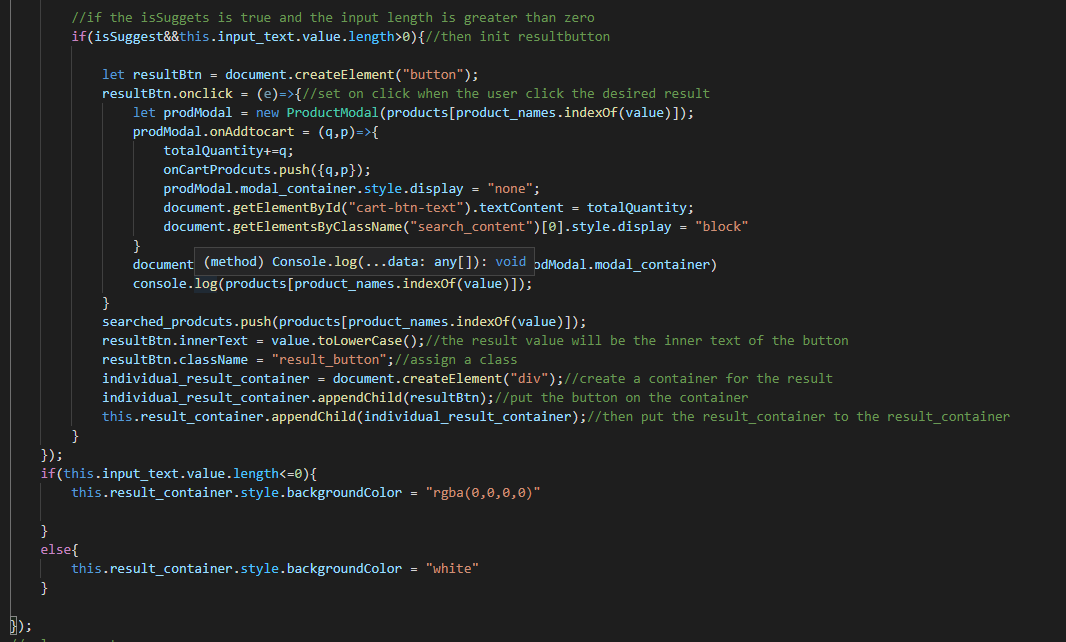
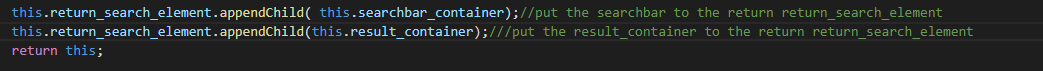
This code is responsible for stopping the automatic scroll when the user hover a product card  
  
This code is responsible for dragging left an right  
  
Checking if the user scrolled on end or start, if end then autoscroll move forward, if not then opposite  
  
Checking if the user is scrolling to left or right, if right then auto scroll go right else left  


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


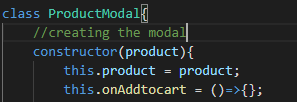
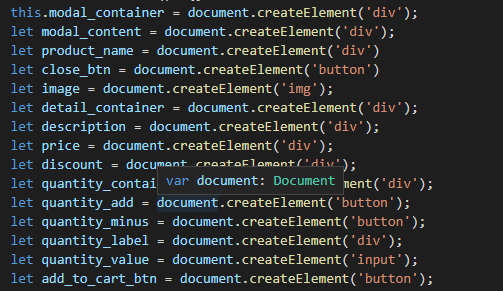
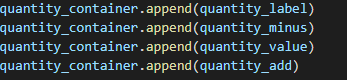
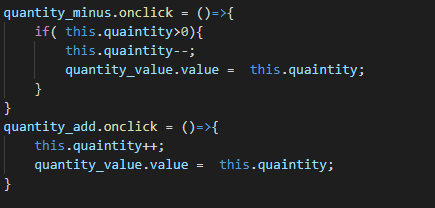
**searchbar.js**This JavaScript file is responsible for searching item on the system.

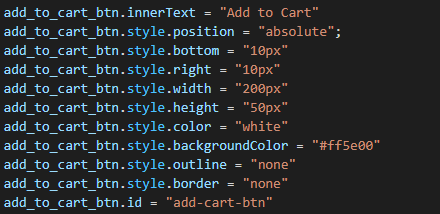
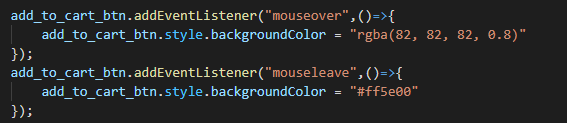
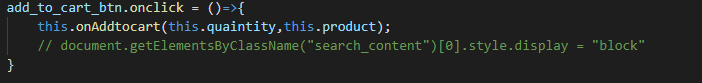
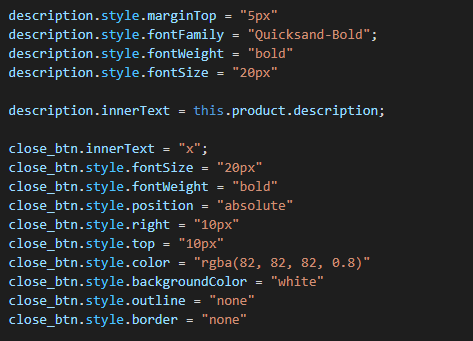
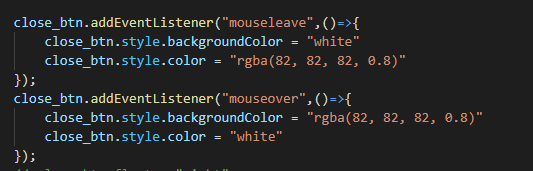
*SearchBar class – Creating a constructor*   
  
Creating search bar elements  


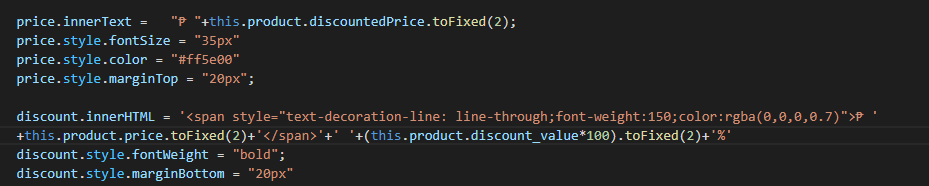
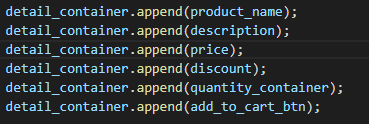
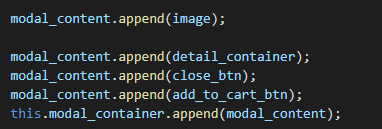
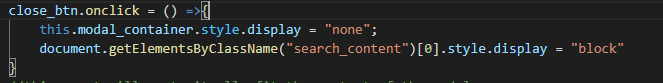
Init elements  
  
Adding on click on submit button(Search Icon) and calling the call back function onSubmit  
  
Check if the user hit enter   
  
Adding the search input text and the search icon to the searchbar container  


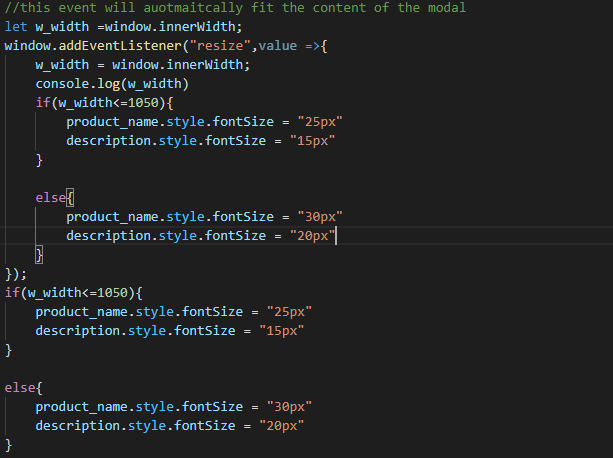
This is code is responsible for search suggestions and on click suggested product  
  
  
Adding the searchbar\_container and result container to the return\_search\_element and return this.  
Therefore you can add this Class to anything using append or appendChild  


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

*ProductModal class – Creating a constructor*   
Creating elements responsible for displaying modal  
Setting up styles for each elements  
  
Adding some element to the quantity container- responsible for displaying quantity on the modal  
  
adding function to add quantity and subtract   


Setting up style for add to cart button   
  
adding mouse hover to add to cart button  
  
adding a call back function to add to cart button  
  
Adding style to descripton and close button  
  
Adding mouse leave and mouseover in close button

Adding style to discount text  
  
Adding style to image  
  
Adding elements to detail\_container  
  
Adding all the element to modal\_content  
  
adding on click function to close the modal  


This code is responsible for resizing the model regarding to the size of the window  
  
Then return this class, therefore you can add this modal to any div by using append or appendChild  
  
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  
