Introduction

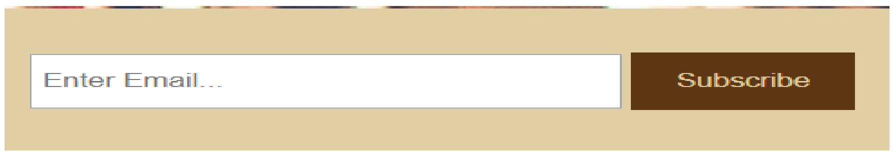
Pelham-Rare Books is book store based in Dublin, Ireland. It has one of the finest collections of Antiquarian books in Dublin. The book store in keeping up with the latest Industry Trends and fellow competitors has decided to take its store Digital. Going digital not only make them in sync with the current trends but also provide a great deal of convenience to its customer in term of enquiring checking out the collection and store details. Having a website will give them the ability to reach out to readers on a global level.

We use **JavaScript** to set the constraints, rules and operations to define and control the **DOM** (Document Object Model) and to achieve the interactivity needed on any website.

1 Inclusion of At least 4 Different GUI Controls

The 4 GUI controls on the website are – textbox, dropdown, list and button.

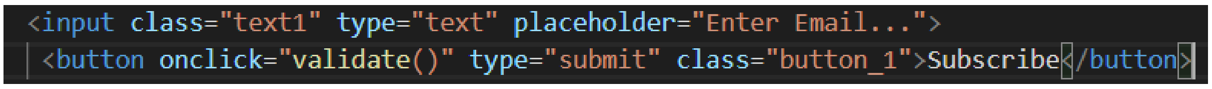
The text box and the button are used on the home page in website under the Subscribe to Newsletter section, below is the screen shot for the same.



1.1: Button and Text input

Textbox is implemented using the input type= “text”.

Button is implemented using the button type = “submit”.

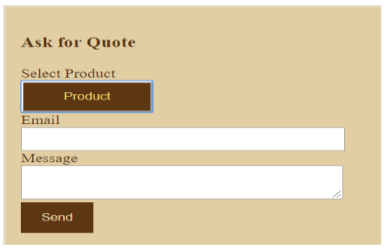


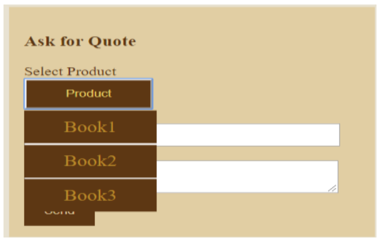
```
<input class="text1" type="text" placeholder="Enter Email...">
 <button onclick="validate()" type="submit" class="button_1">Subscribe</button>
```

1.2: HTML for text and button

Drop-down has been implemented on the contact us page in the get a quote side bar.

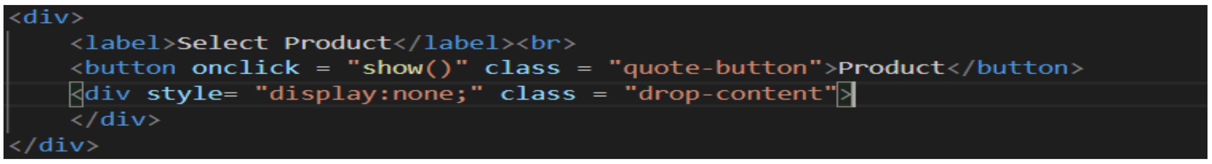
Under the Select product heading when the user clicks on the button Product it gives a **drop-down** list of items for which the customer can send in an inquiry or get a quote.





1. Drop-Down in the Contact us Page

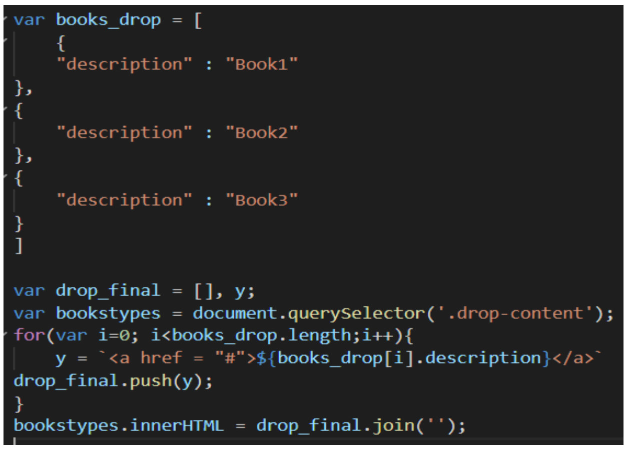
I have used a **button** by the name Product which when pressed displays the book options.



```
<div>
    <label>Select Product</label><br>
    <button onclick = "show()" class = "quote-button">Product</button>
    <div style= "display:none;" class = "drop-content">
    </div>
</div>
```

1.4: HTML code for Drop-Down

I have used an **array** to store the different book options and then using the **innerHTML** I have written the drop-down element to the **DOM**. Below shown is the sample code for the same.

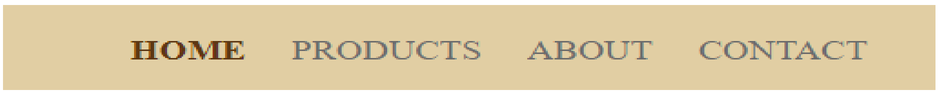


```
var books_drop = [
    {
    "description" : "Book1"
},
{
    "description" : "Book2"
},
{
    "description" : "Book3"
}
]

var drop_final = [], y;
var bookstypes = document.querySelector('.drop-content');
for(var i=0; i<books_drop.length;i++){
    y = `<a href = "#">${books_drop[i].description}</a>`
drop_final.push(y);
}
bookstypes.innerHTML = drop_final.join('');
```

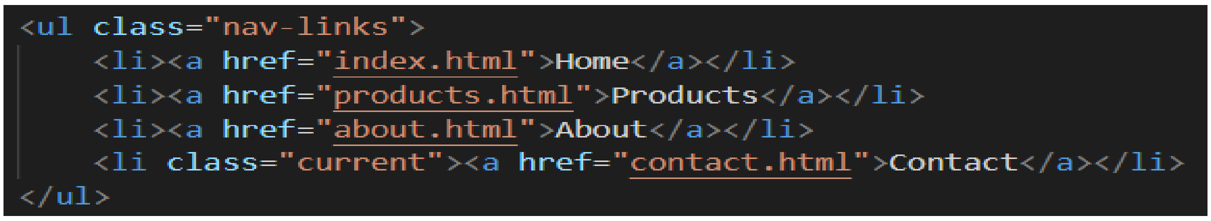
1.5: Array Initialized and values passed using innerHTML to the parent Class

List has been implemented for the navigation menu.



1.6: Navigation Bar for the Website

Below is the sample code for the same.

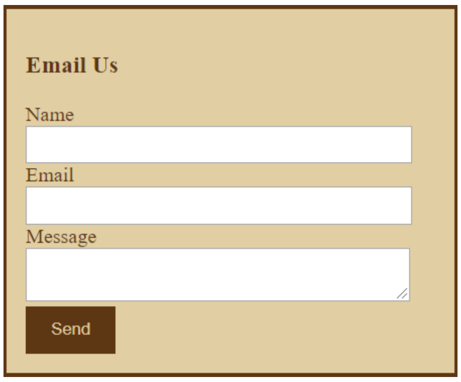


```
<ul class="nav-links">
    <li><a href="index.html">Home</a></li>
    <li><a href="products.html">Products</a></li>
    <li><a href="about.html">About</a></li>
    <li class="current"><a href="contact.html">Contact</a></li>
</ul>
```

1.8: HTML code for Navigation Bar

2 An HTML form containing at least 3 User Input Elements.

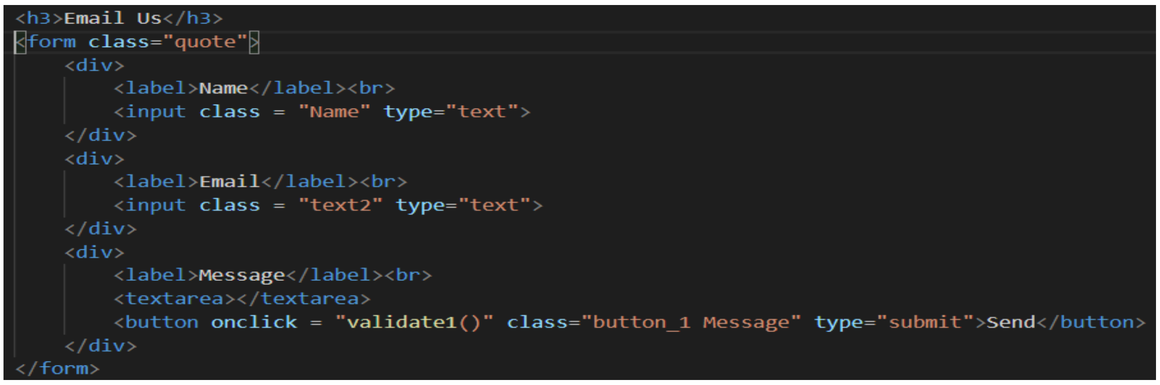
The 3-user input in the **HTML form** are **Name, Email** and **Message**. These are shown in the footer of all the pages for the customers convenience.



2.1: Form Example

Below is the sample code for the form elements.

For all the three 3 user inputs the input type is text. The submit button when clicked run the script for validation which is discussed later in the report.



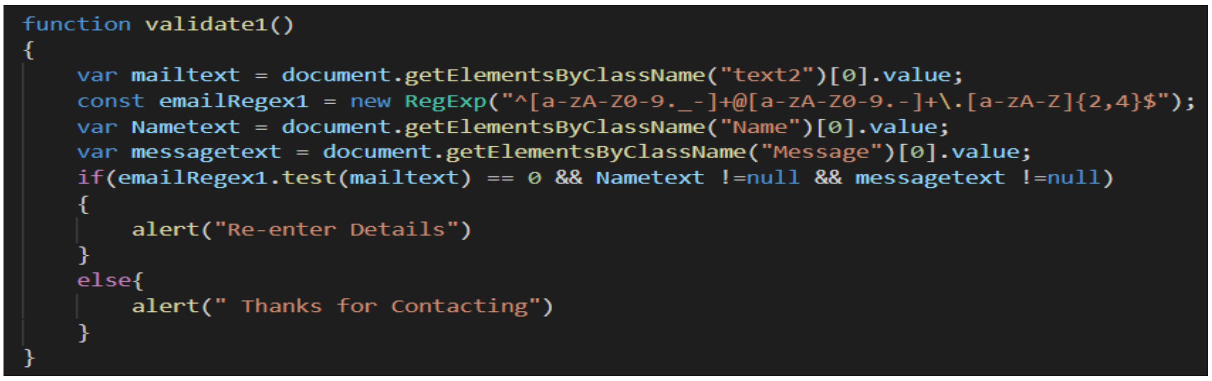
```
<h3>Email Us</h3>
<form class="quote">
    <div>
        <label>Name</label><br>
        <input class = "Name" type="text">
    </div>
    <div>
        <label>Email</label><br>
        <input class = "text2" type="text">
    </div>
    <div>
        <label>Message</label><br>
        <textarea></textarea>
        <button onclick = "validate1()" class="button_1 Message" type="submit">Send</button>
    </div>
</form>
```

2.2: HTML Code for the form

3 Proper validation of these Input elements Using JavaScript

The input element used are **Name, Email** and **Phone Number**. The email has been validated using the **regex**(regular Expression), where the email input has been tested against the regular expression using the test() function in JavaScript.

The name and the message inputs are checked using !=null. Below is the sample code from Java Script for your reference.



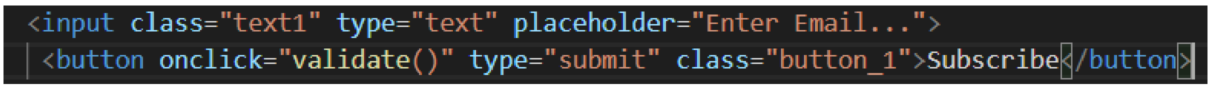
```
function validate1()
{
    var mailtext = document.getElementsByClassName("text2")[0].value;
    const emailRegex1 = new RegExp("^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$");
    var Nametext = document.getElementsByClassName("Name")[0].value;
    var messagetext = document.getElementsByClassName("Message")[0].value;
    if(emailRegex1.test(mailtext) == 0 && Nametext !=null && messagetext !=null)
    {
        alert("Re-enter Details")
    }
    else{
        alert(" Thanks for Contacting")
    }
}
```

3.1: Javascript code for validation of Input Elements in the Form

4 4 Event Handlers for at least four events on the on the website

The 4-events handler used are **onclick** for subscribe on the home page, **onclick** for drop-down on the contact page, **onchange** for change in quantity in the cart items on products page and **onclick** when button to remove the items is pressed under the cart in the products page. There is one more event which is onclick which is triggered when the button ADD to CART is clicked in the products page

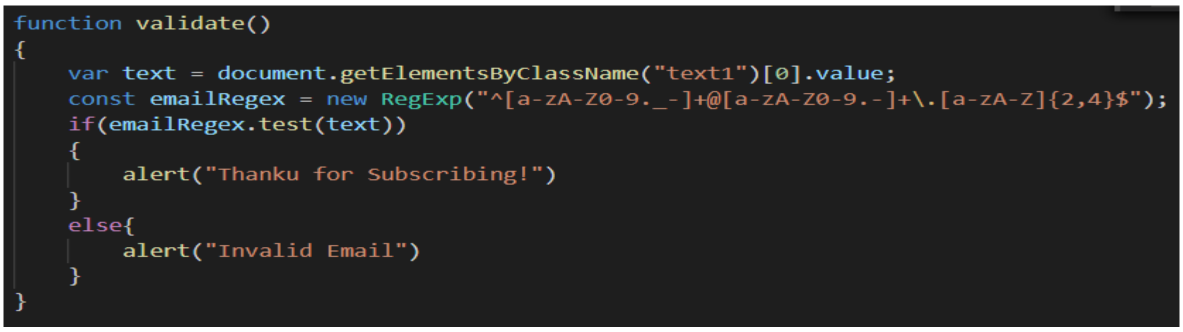
Onclick is used on subscribe button, Product button for drop down and Send email button in the footer. Below is the sample code for the same.



```
<input class="text1" type="text" placeholder="Enter Email...">
  <button onclick="validate()" type="submit" class="button_1">Subscribe</button>
```

4.1: HTML for the onclick event on Newsletter Section

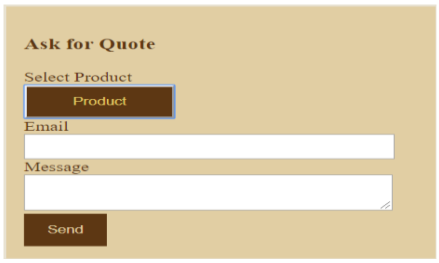
The Subscribe button **onclick** triggers an event which checks weather the input email is correct using the regex expression. If the email entered is correct it throws an alert saying "Thanku for Subscribing!" if not it says "Invalid Email".



```
function validate()
{
    var text = document.getElementsByClassName("text1")[0].value;
    const emailRegex = new RegExp("^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$");
    if(emailRegex.test(text))
    {
        alert("Thanku for Subscribing!")
    }
    else{
        alert("Invalid Email")
    }
}
```

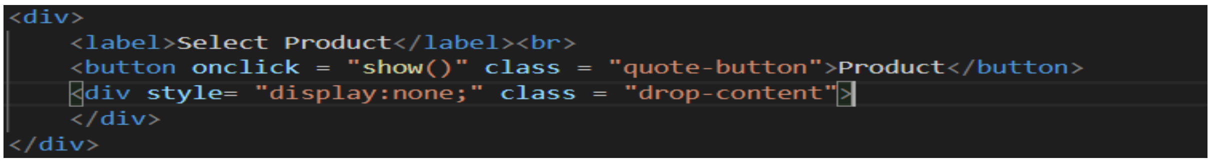
4.2: Java Script code for Email Verification

I have used **onClick** on the Contact Us page as well in the get a Quote section. Below is the image showing the place it is used.



4.3: Drop Down

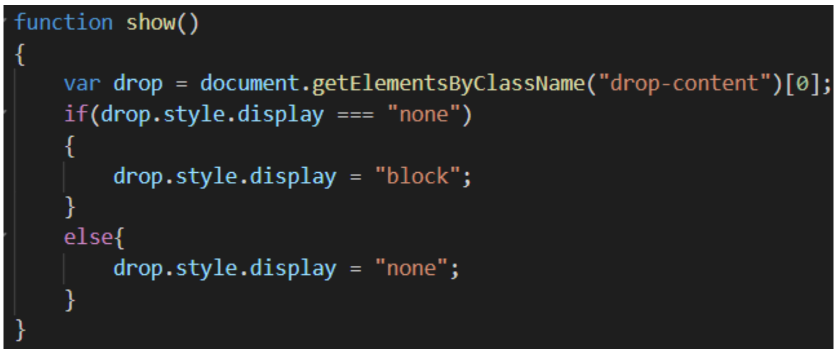
The **onClick** event is triggered when the products button is pressed showing a different product option.



```
<div>
    <label>Select Product</label><br>
    <button onclick = "show()" class = "quote-button">Product</button>
    <div style= "display:none;" class = "drop-content">
    </div>
</div>
```

4.3: HTML for Drop Down

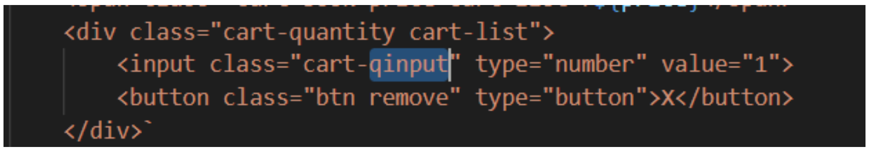
Below shown is the JavaScript code for the same.



```
function show()
{
    var drop = document.getElementsByClassName("drop-content")[0];
    if(drop.style.display === "none")
    {
        drop.style.display = "block";
    }
    else{
        drop.style.display = "none";
    }
}
```

4.4: Event Trigger Function show to display Drop-Down

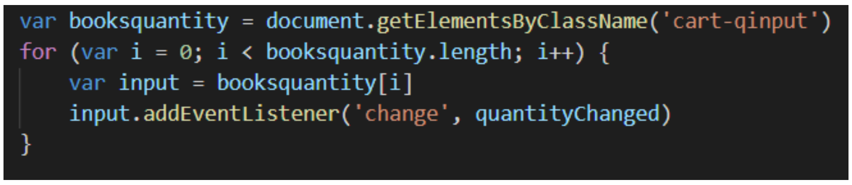
The **onchange** event is triggered when the change in quantity is made for the products in the cart. The input type is number, and the event is activated whenever there is change in the number.



```
    <div class="cart-quantity cart-list">
        <input class="cart-qinput" type="number" value="1">
        <button class="btn remove" type="button">X</button>
    </div>`
```

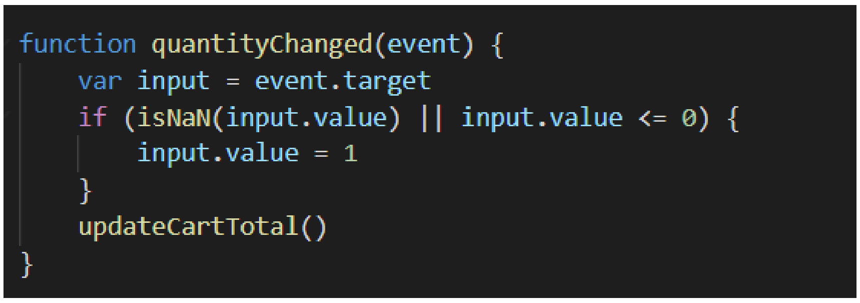
4.5 HTML for Product Quantity(input type=number)

On change the event triggers and updates the cart total. If the number is <1 the input returns to one. So, the input can only be 1 or >1.



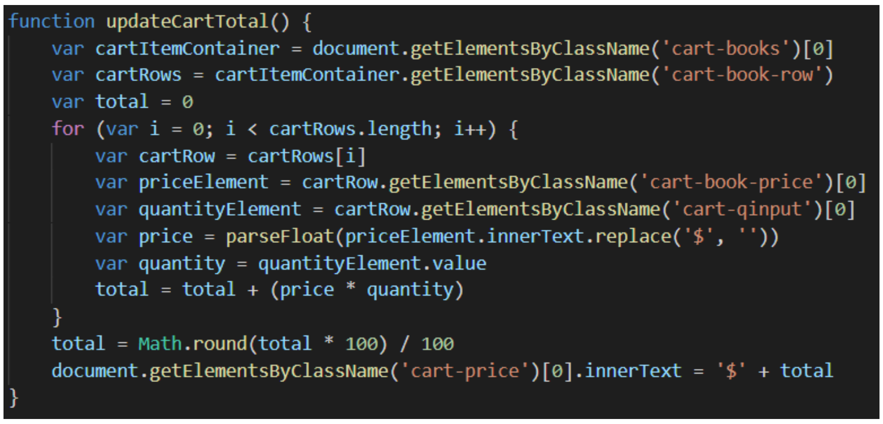
```
var booksquantity = document.getElementsByClassName('cart-qinput')
for (var i = 0; i < booksquantity.length; i++) {
    var input = booksquantity[i]
    input.addEventListener('change', quantityChanged)
}
```

4.6 Change in number triggers the Event



```
function quantityChanged(event) {
    var input = event.target
    if (isNaN(input.value) || input.value <= 0) {
        input.value = 1
    }
    updateCartTotal()
}
```

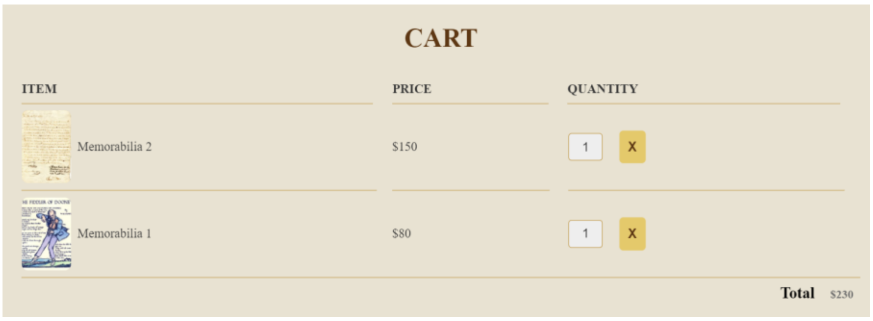
4.7 Function Event checking if the quantity is 1 or greater



```
function updateCartTotal() {
    var cartItemContainer = document.getElementsByClassName('cart-books')[0]
    var cartRows = cartItemContainer.getElementsByClassName('cart-book-row')
    var total = 0
    for (var i = 0; i < cartRows.length; i++) {
        var cartRow = cartRows[i]
        var priceElement = cartRow.getElementsByClassName('cart-book-price')[0]
        var quantityElement = cartRow.getElementsByClassName('cart-qinput')[0]
        var price = parseFloat(priceElement.innerText.replace('$', ''))
        var quantity = quantityElement.value
        total = total + (price * quantity)
    }
    total = Math.round(total * 100) / 100
    document.getElementsByClassName('cart-price')[0].innerText = '$' + total
}
```

4.8 Depending on the quantity the total cart price is computated

Another onclick event is placed on the remove button a cross sign that appears in the products page. This button when pressed remove the corresponding item from the cart and updates the total of the cart. Below is the image of the page where this is used.

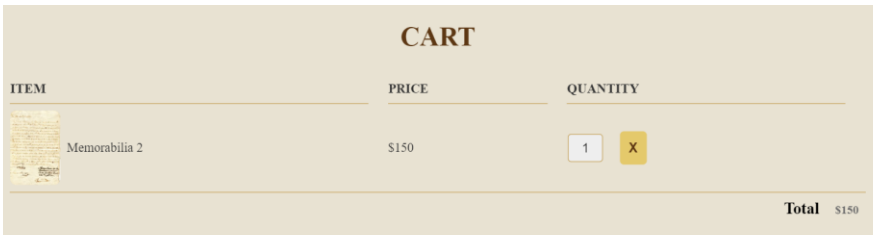


CART

ITEM	PRICE	QUANTITY
Memorabilia 2	$150	1 X
Memorabilia 1	$80	1 X

Total $230

4.9 Image showing the Remove Button in cart

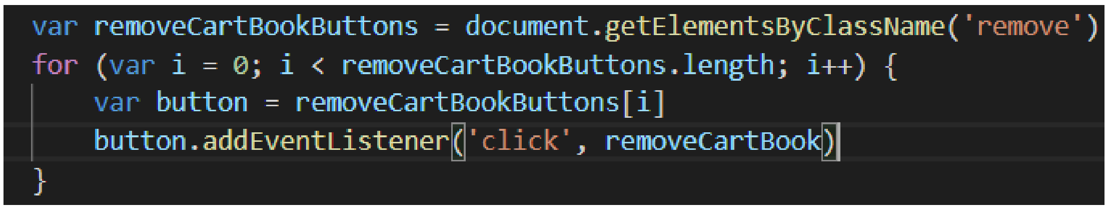

CART

ITEM	PRICE	QUANTITY
Memorabilia 2	$150	1 X

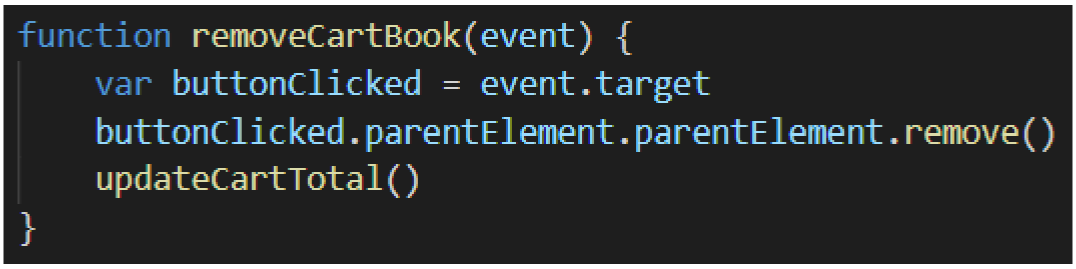
Total $150

4.10 Item Details removed when Remove Button is pressed

When triggering the event calls the function removeCartBook which removes the corresponding element in the cart and updates the cart total. Below is the Java Script code for the same.


```
var removeCartBookButtons = document.getElementsByClassName('remove')
for (var i = 0; i < removeCartBookButtons.length; i++) {
    var button = removeCartBookButtons[i]
    button.addEventListener('click', removeCartBook)
}
```

4.11 Java Script for adding Event to the Buttom


```
function removeCartBook(event) {
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove()
    updateCartTotal()
}
```

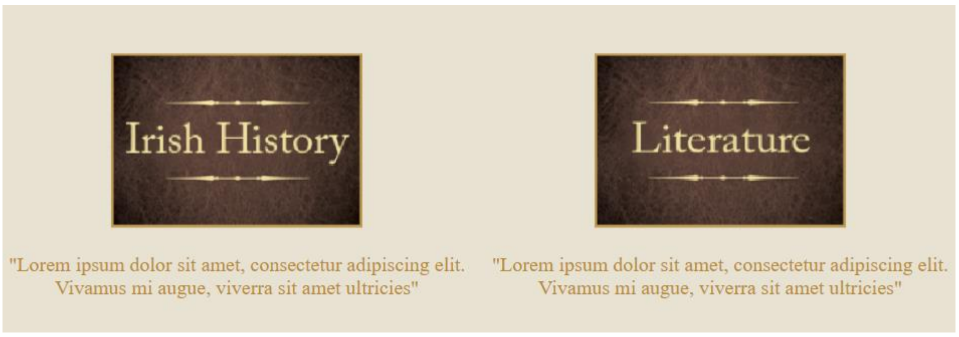
4.12: Function to remove items from the cart when button is pressed

Note: referred to the below lesson and material I found on YouTube:-
https://www.youtube.com/watch?v=YeFzkC2awTM&t=578s

https://github.com/WebDevSimplified/Introduction-to-Web-Development/tree/master/Introduction%20to%20JavaScript/Lesson%201

5 Arrays

Array are used in the home page to update the images for the different categories of book and display the description text as well.

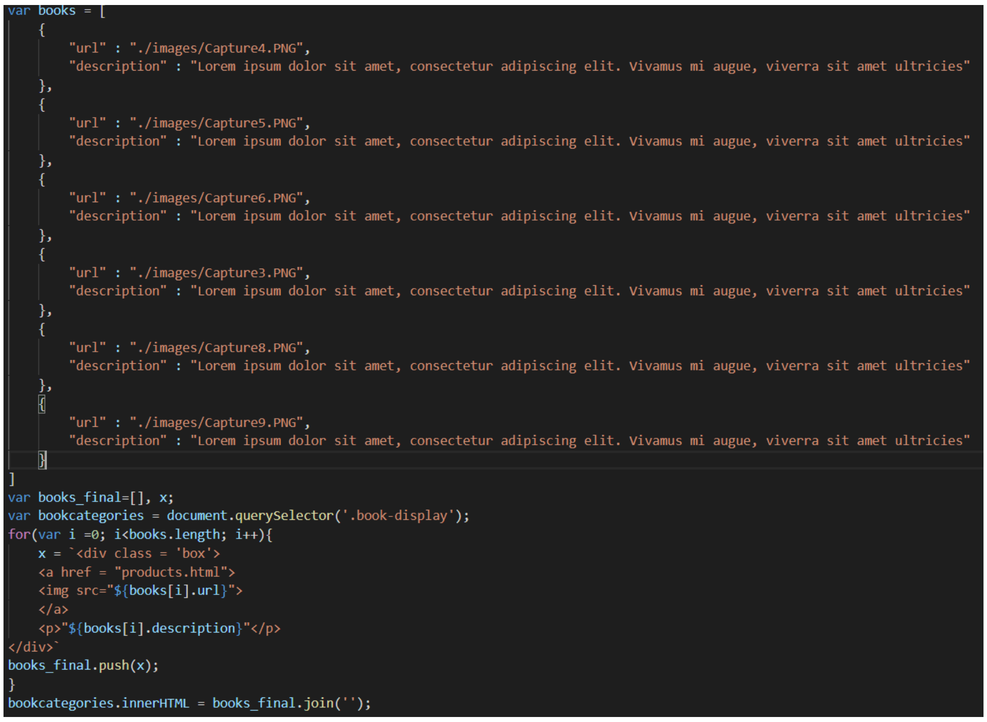


"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"

"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"

5.1: Home Page Arrays used to display image and Text

In the below shown code two arrays have been used one is the books array and the other is books_final array. The books array is initialized with the image url and the description, using for loop each element in the books array is iterated and the value is used to the books_final array. The books_final is assigned to the inner html of the parent element which display the elements.



```
var books = [
    {
        "url" : "./images/Capture4.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    },
    {
        "url" : "./images/Capture5.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    },
    {
        "url" : "./images/Capture6.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    },
    {
        "url" : "./images/Capture3.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    },
    {
        "url" : "./images/Capture8.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    },
    {
        "url" : "./images/Capture9.PNG",
        "description" : "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra sit amet ultricies"
    }
]
var books_final=[], x;
var bookcategories = document.querySelector('.book-display');
for(var i =0; i<books.length; i++){
    x = `<div class = 'box'>
    <a href = "products.html">
    <img src="${books[i].url}">
    </a>
    <p>"${books[i].description}"</p>
</div>`
books_final.push(x);
}
bookcategories.innerHTML = books_final.join('');
```

5.2: Java Script for initializing the array and pushing it to the parent class

6 Functions

The two functions I have used in my code are one is function validate() and the other is function show(). The validate function is used in the newsletter section of the home page. This function is called when the subscribe button is pressed. It check whether the text input in the email field matches with the regex function (Regular Expression).

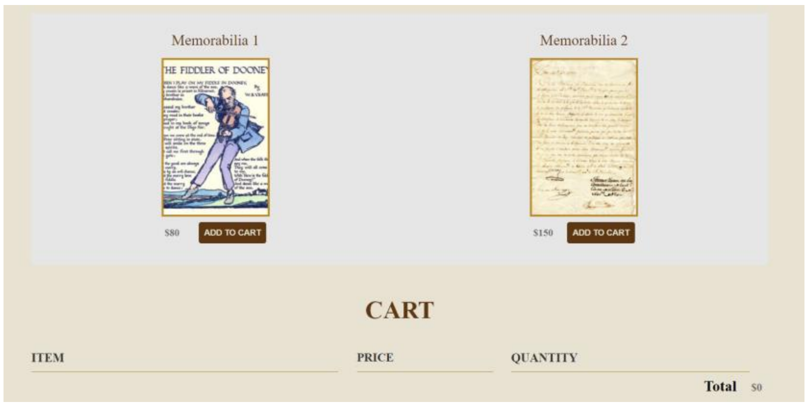
The show() function is used in the get a quote section of the Contact page. The show function is called when the user presses the products button. The function displays the drop down showing the products options for which the customer can get a quote.

7 Dynamic Manipulation

Dynamic manipulation is used in the products Page for the cart section.

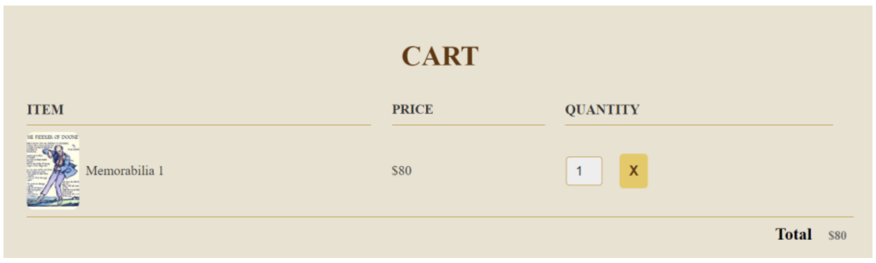
When the user presses the add to cart button book details along with the image title price and quantity are displayed under the cart section.

The first image shown is of empty cart.



7.1: Empty Cart in Products Page

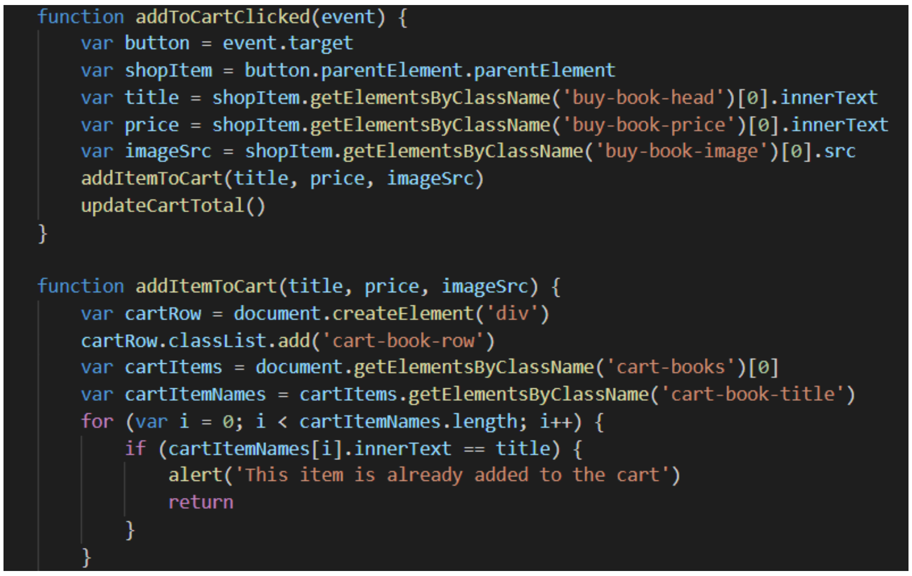
Below image shows details specific to the product for which add to cart is pressed.



7.2: Item Details Added

Below is the java script code for the same. When add to cart is pressed an on click event is triggered. The function addtoCartClicked is called which take the title price and image link for the respective item. These items are then added to the cart section using the additemToCart function. It also checks for the condition if an item is already present in the cart. If yes, the web page gives an alert for the same.

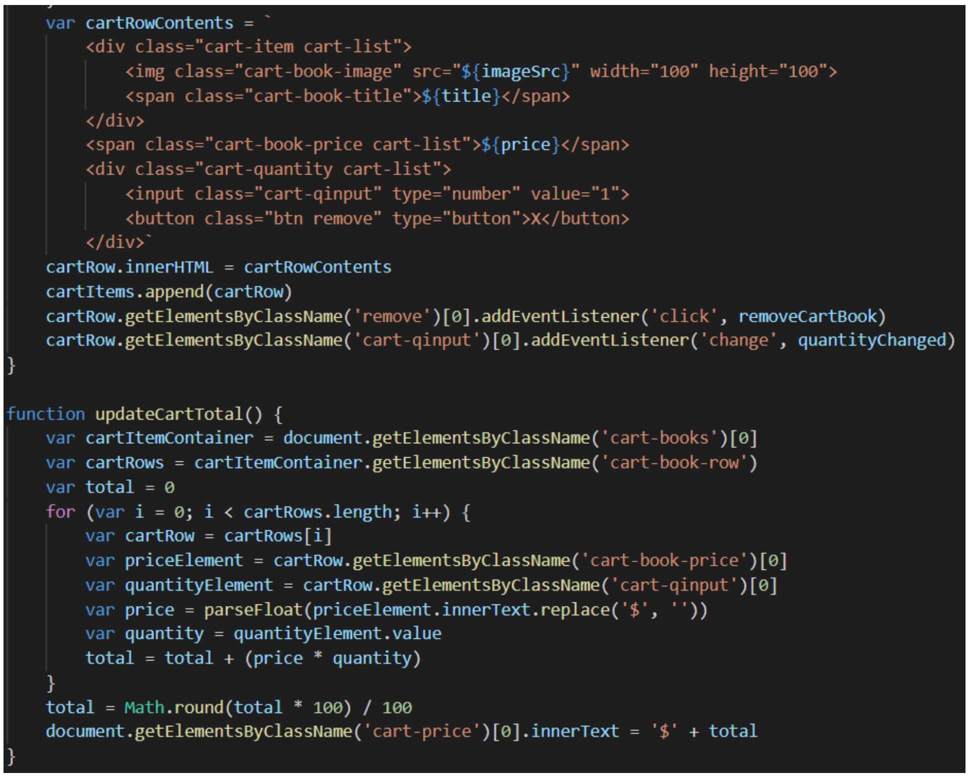
The product details are now appended to the parent element using the innerHTML. The cart total is updated, and the cart looks similar to the image shown in **7.2.**



```
function addToCartClicked(event) {
    var button = event.target
    var shopItem = button.parentElement.parentElement
    var title = shopItem.getElementsByClassName('buy-book-head')[0].innerText
    var price = shopItem.getElementsByClassName('buy-book-price')[0].innerText
    var imageSrc = shopItem.getElementsByClassName('buy-book-image')[0].src
    addItemToCart(title, price, imageSrc)
    updateCartTotal()
}

function addItemToCart(title, price, imageSrc) {
    var cartRow = document.createElement('div')
    cartRow.classList.add('cart-book-row')
    var cartItems = document.getElementsByClassName('cart-books')[0]
    var cartItemNames = cartItems.getElementsByClassName('cart-book-title')
    for (var i = 0; i < cartItemNames.length; i++) {
        if (cartItemNames[i].innerText == title) {
            alert('This item is already added to the cart')
            return
        }
    }
```

7.3: Event on click Add to cart is trigged



```
    var cartRowContents = `
        <div class="cart-item cart-list">
            <img class="cart-book-image" src="${imageSrc}" width="100" height="100">
            <span class="cart-book-title">${title}</span>
        </div>
        <span class="cart-book-price cart-list">${price}</span>
        <div class="cart-quantity cart-list">
            <input class="cart-qinput" type="number" value="1">
            <button class="btn remove" type="button">X</button>
        </div>`
    cartRow.innerHTML = cartRowContents
    cartItems.append(cartRow)
    cartRow.getElementsByClassName('remove')[0].addEventListener('click', removeCartBook)
    cartRow.getElementsByClassName('cart-qinput')[0].addEventListener('change', quantityChanged)
}

function updateCartTotal() {
    var cartItemContainer = document.getElementsByClassName('cart-books')[0]
    var cartRows = cartItemContainer.getElementsByClassName('cart-book-row')
    var total = 0
    for (var i = 0; i < cartRows.length; i++) {
        var cartRow = cartRows[i]
        var priceElement = cartRow.getElementsByClassName('cart-book-price')[0]
        var quantityElement = cartRow.getElementsByClassName('cart-qinput')[0]
        var price = parseFloat(priceElement.innerText.replace('$', ''))
        var quantity = quantityElement.value
        total = total + (price * quantity)
    }
    total = Math.round(total * 100) / 100
    document.getElementsByClassName('cart-price')[0].innerText = '$' + total
}
```

7.4: Cart Details are paused to the parent class using inner html the

File Structure

All the images and icons used in website are stored in the images folder. There are three java script files index.js contains script running in the home page, contact.js contains script for the contacts page and the products.js contains the script running in the product page. All the JS files are stored in a separate JS folder. The CSS file is stored in the folder by the name style.

Note: Apart from the Code in the products page which have been referenced from a YouTube lesson and the regex expression, rest of the code is self-written.

References:

https://stackoverflow.com/questions/46155/how-to-validate-an-email-

https://www.youtube.com/watch?v=YeFzkC2awTM&t=578s

https://github.com/WebDevSimplified/Introduction-to-Web-Development/tree/master/Introduction%20to%20JavaScript/Lesson%201