SOFT20081

Internet Application Programming

User Manual

T0321003

I hereby declare that I am the sole author of this report and that all third-party items including code have been adequately acknowledged and referenced.



Rick Richard Harith

**Table of Content**

1. Introduction

1.1 Success Criteria

2. Design Process

2.1 Initial Research

2.2 Initial Design (Justification)

2.2.1 Site Map

2.2.2 Wireframe

2.2.3 Moodboard

3. Implementation

3.1 Features Developed, Implementation Processes and Challenges Encountered

4. Summary and Conclusion

Sources and Reference

Appendix

Link to Site on the UDON Server:-

**1.0 Introduction**

This is the supporting documentation for the e-commerce website created for SOFT20181 Internet Application Programming.

This e-commerce website is intended to promote art stationeries. It consists of 18 pages built for desktop PCs first, the website has been tested on 7 devices including iPhone SE, iPhone XR, iPhone 12 Pro, Pixel 5, Samsung Galaxy S8+, Samsung Galaxy S20 Ultra and the iPad Air. The website has been written in HTML5 and is rendered using CSS level 3. Client-side interaction is achieved using JavaScript and JQuery.

The website has been designed to be mostly media responsive for displays of different sizes on other devices, there are a few errors that will be mentioned. Furthermore, the images used have been compressed and kept in good quality to ensure faster load times and all of the code has been validated and contains no errors.

The websites has been tested on devices with alternate resolutions and some problems still remain.

**1.1 Success Criteria**

Everything on the website functions as intended with the exception of a couple of parts which will be discussed in the appropriate sections. There are issues with the responsive design of the website currently, which will be addressed in the relevant sections below.

|  |  |  |
| --- | --- | --- |
| **Success Criteria Point** | **Priority** | **Implemented?** |
| Website Navigation | Required | Yes, a navigation bar and footer has been implemented across all pages which links to all internal pages. |
| Customised Presentation | Required | **Yes**, background colours and font sizes of the main body can be changed, they are controlled by Javascript and the user preference is stored using the local storage API. |
| Registration Page | Required | **Yes**, appropriate form controls have been implemented in the registration form to ensure users do not submit malformed email addresses and name fields only contain letters and spaces. |
| Product pages | Required | **Yes**, product pages have been made for each product present in the website, some products also have relevant videos from YouTube. |
| Thumbnail image | Required | **Yes**, the products thumbnail image responds to a click and enlarges for the user. |
| Responsive Design | Required | **Not entirely**, the ‘index.html’ will appear broken under certain device resolutions, but most of the website is responsive. |
| Product Slider | Optional | **Yes**, a product slider is coded in and the user can navigate it with the arrows present in the website, or use the horizontal scroll to do so. |
| Smooth scrolling animation | Optional | **Yes**, this has been done with Javascript, when navigating the product page, links can be pressed to smoothly scroll the corresponding elements into view. |
| Product Image Gallery | Optional | **Yes**, the product pages have simple image viewers and see multiple images of said product, this was achieved with a script present in each of the HTML product pages. |

Sources and references for all the material used can be found on the last page of the report.

**2.0 Design Process**

**2.1 Initial Research**

The design of the website was aimed to be minimalistic and straight-forward, with plain colours and rounded fonts, the colours used are light and palatable while being eye catching enough. Elements within the website follow this principle in terms of design, the customisable colours are generally softer colours as well. After looking at certain e-commerce sites, far too much information is present on screen for some, overwhelming the user. Because of this, the website was designed to be simple, so it is easy on the users eyes and allows to them to quickly scan and navigate through the website with ease.

* The navigation bar should be simple and have small animations to further the interactivity, it must be visually minimalistic while striking so it catches the eyes of users, making it easier to navigate between pages.
* Some e-commerce websites have home pages with an animated video as a background to immediately capture the user’s attention to sign up before browsing. Usually to direct customers to a product they wish to promote or registrations to the website.
* Stationery sites tend to have bland and simplistic designs, while the simplistic nature is appealing, more softer colours and sleeker element designs should help increase interest in the website
* Best selling products are displayed front and centre to maximise customer curiosity, allowing users to browse deeper into the website
* Most colour changing websites have text and background colour pre-sets to avoid user error and making white text appear on white backgrounds, or black text on black backgrounds.
* Sites that allow the user to change font sizes usually preserve the websites layout.
* Overall, the website should aim to grab a user’s attention with simple designs and eye-catching colours, products should also be shown to peak interest in potential customers.

**2.2 Initial Design**

The website would follow a simple layout to navigate, and would generally function as shown below.

Diagram

Description automatically generated

Figure . Site Map

Users would be introduced to the home webpage where all the pages can be accessed at once. However, the home page would be designed to draw users to the most important aspect of the website, which would be the sign up page. Naturally, having more users sign up shows for potential customers, making the site as simple as possible to navigate is a key factor to ensuring this.

Table

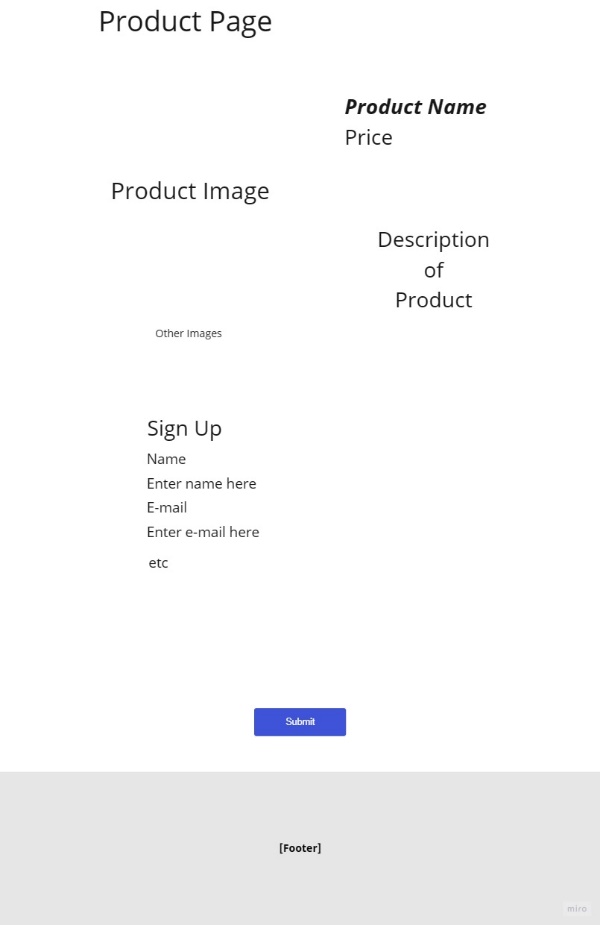
Description automatically generated with low confidence

Figure . and 3. Wireframe

The above wireframe illustrates the layout of the website, the initial idea would be to have a simple product slider in the Home to navigate and see the best-selling products, accompanied by a background video, grabbing the users attention immediately. The product page would be a split where one half of the page showed an image of the product and the other its details. The sign up would be a simple form control in design.

Since colours need to be customisable, elements should be split to give visual variety when the colour of the website is changed, font size should also be considered in the design as it could easily be broken if the font is adjusted by increments of 2 pixels each button press, a reset button would be necessary.

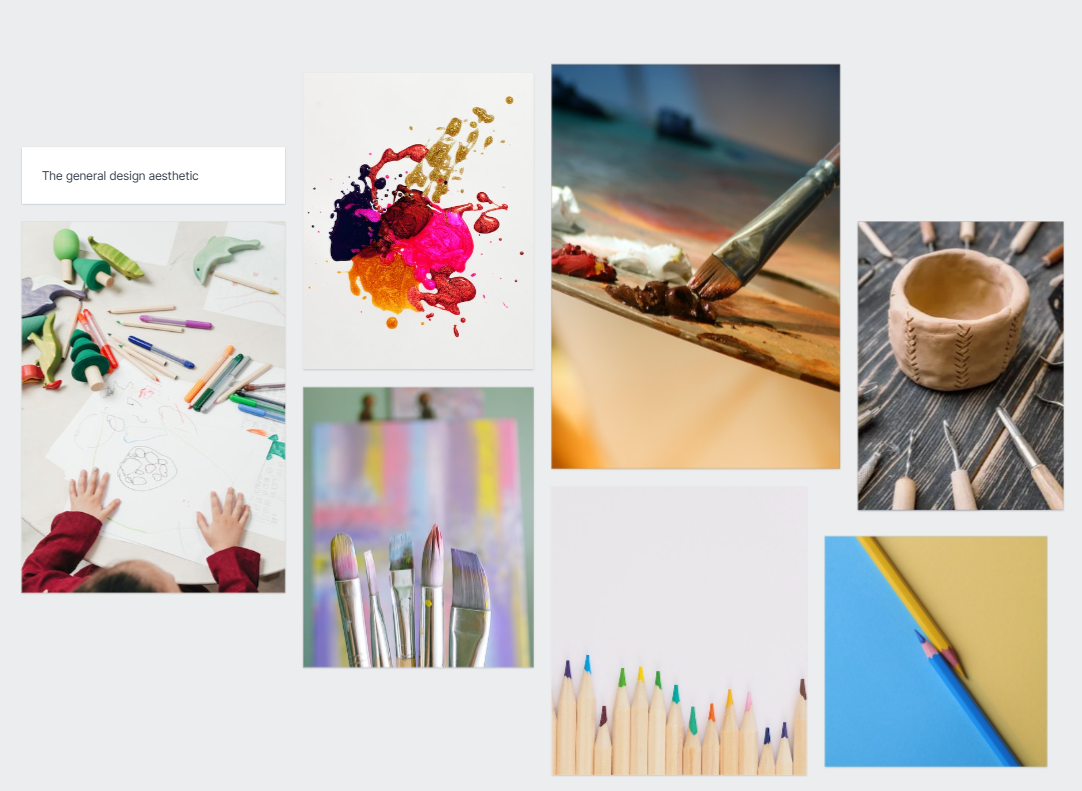


Figure 4. Moodboard

This would be the general design of the website, minimalistic but vibrant in colours. Visually, images like this should give a sense of calmness and elegance in the websites design. Following this general aesthetic should allow the website to have a relaxing nature to it. Art is all about beauty, to show said beauty is appropriate for a professional art stationery website.

**3.0 Implementation**

**3.1 Features Developed, Implementation Processes and Challenges Encountered**

**Section 1: Home Page**

The navigation bar and footer are essential to the navigation of the page, the navigation bar in particular turns into a hamburger menu because of a media query in the CSS to allow for a more functional and appealing navigation bar under smaller screen resolutions.

The home page (index.html) is divided into 3 sections, most of the elements are separated by <div> and larger sections of the html are separated with <sections>. The welcome page has an mp4 video that offers users to sign up immediately. The video covers up the website because of the ‘object-fit-cover’ CSS.

After that follows the product slider, the following HTML code displays a list of products in a slider format. Notably, there are two buttons called “l-button” and “r-button” that are used to navigate through the product slider. The code below is the Javascript code that allows for the buttons to navigate through the product slider:

const productContainer = [...document.querySelectorAll('.product-container')];

const RButton = [...document.querySelectorAll('.r-button')];

const LButton = [...document.querySelectorAll('.l-button')];

productContainer.forEach((item, i) => {

let containerDimensions = item.getBoundingClientRect();

let containerWidth = containerDimensions.width;

RButton[i].addEventListener('click', () => {

item.scrollLeft += containerWidth;

})

LButton[i].addEventListener('click', () => {

item.scrollLeft -= containerWidth;

})

})

By using the “querySelectorAll” method, it selects all the elements of the page that are within the class = “product-container”, specifically “r-button” and “l-button” are selected in this query. Then the script uses the “forEach” line to iterate through elements in the “productContainer” array. The getBoundingCLientRect() method is used to return the size and position of an element relative to the viewport. Finally, the buttons have click event listeners, “r-button” when pressed will scrollLeft and “l-button” will decrease the scrollLeft property increment and make the slider go right instead.

A good variety of HTML tags were utilised though in hindsight, more planned out <sections> would have benefited the websites slightly messy design.

**Section 2: Presentation Changing**One of the requirements of the website is to allow for the user to customise the colours of the website. I have implemented the feature by utilising “.root” in the CSS, it is a pseudo-class selector and is used to select the highest “root” element of a document. This allows for custom variables to be made and store values according to the root.

With this knowledge, the Javascript is used to set colours based on buttons in the HTML, 2 variables are made, “—background-color” and “—text-color”, they are used to set text and background colour as simply changing the background colour may result in harder to read text. These colours are given unique IDs that are selected In the Javascript, they set different combinations of colours when the buttons are clicked as shown here:

const redButton = document.querySelector('#red');

const blueButton = document.querySelector('#blue');

const greenButton = document.querySelector('#green');

const yellowButton = document.querySelector('#yellow');

const whiteButton = document.querySelector('#white');

redButton.addEventListener('click',()=>{

document.documentElement.style.setProperty('--background-color', '#FF999B')

document.documentElement.style.setProperty('--text-color', '#3B0001')

localStorage.setItem('background-color', '#FF999B');

localStorage.setItem('text-color', '#3B0001');

});

blueButton.addEventListener('click',()=>{

document.documentElement.style.setProperty('--background-color', '#96E1FF')

document.documentElement.style.setProperty('--text-color', '#001061')

localStorage.setItem('background-color', '#96E1FF');

localStorage.setItem('text-color', '#001061');

});

greenButton.addEventListener('click',()=>{

document.documentElement.style.setProperty('--background-color', '#93EDA1')

document.documentElement.style.setProperty('--text-color', '#004008')

localStorage.setItem('background-color', '#93EDA1');

localStorage.setItem('text-color', '#004008');

});

yellowButton.addEventListener('click',()=>{

document.documentElement.style.setProperty('--background-color', '#FAFA9D')

document.documentElement.style.setProperty('--text-color', '#BF9D04')

localStorage.setItem('background-color', '#FAFA9D');

localStorage.setItem('text-color', '#BF9D04');

});

whiteButton.addEventListener('click',()=>{

document.documentElement.style.setProperty('--background-color', '#FFFFFF')

document.documentElement.style.setProperty('--text-color', '#000000')

localStorage.setItem('background-color', '#FFFFFF');

localStorage.setItem('text-color', '#000000');

});

window.addEventListener('load', () => {

let bgColor = localStorage.getItem('background-color');

let textColor = localStorage.getItem('text-color');

if (bgColor) {

document.documentElement.style.setProperty('--background-color', bgColor);

}

if (textColor) {

document.documentElement.style.setProperty('--text-color', textColor);

}

});

When these colours are set, it is saved to the local storage API, and fetched on window load, so users will not lose the websites appearance after a refresh.

Besides that, the font size changer is also implemented. The HTML pages are all wrapped in a <main>, all three functions select the <main> tag and the “increaseFontSize()” function adds 1 pixel in font size to all text within <main>, the opposite is done by the “decreaseFontSize” function. Both of these functions save the new font size to the local API storage. An array is used to iterate through the elements within <main> to assign the “storedSize” font size. The problem that comes with approaching font size changing like this is the user error that may occur, the website is not coded to be responsive to these changes and always gets broken afterwards. So a reset font button had to made to minimise the damage that may be caused. This is done by the “resetFontSize()” function that resets all text elements font size within <main> by removing the inline font-size style using “elements[i].style.fontSize = “”;”. The font saved within the local storage is also removed with .removeItem.

**Section 3: Sign Up Form Validation**

The sign up form requires the user to send in valid e-mails and names. This means that the e-mail must be in the correct format and names can only have spaces and letters. The best approach to this is regular expressions, also know as regex. Below is the example code for my e-mail validation function:

function ValidateEmail(){

var email = document.getElementById("EMAIL").value;

var regx = /^([a-zA-Z0-9\.\_]+)@([a-zA-Z0-9])+.([a-z]+)(.[a-z]+)?$/

if(email.trim() === ""){

alert("E-mail field cannot be empty");

return false;

}

else if(regx.test(email)){

alert("E-mail format accepted")

return true

}

else{

alert("Invalid e-mail format")

return false;

}

}

2 variables are defined, “email” is assigned to the id “EMAIL” while “regx” is defined and assigned a regular expression that will match up to the pattern of an email address. “ifs” are used to check certain conditions. Firstly, if the field is empty, an error message will print and no value will be returned, this can also be done on HTML 5 with the required tag. Secondly if the e-mail matches up with the regular expression, it will be accepted and a value will be returned, else and an error message is printed along with a false return.

This concept is used again for the “CheckName()’ function but with a regex for only spaces and letters: /^[a-zA-Z\s]\*$/

**Section 4: Animated Thumbnails and Gallery**

This part of my code is not present in my Javascript external script, it has been implemented within the HTML file itself because of the code being far too specific.

window.onload = function() {};

First of all, the script is wrapped with a window on load event to ensure that the Javascript code is only executed after the page has finished loading. Not allowing elements to be available within the DOM will bring errors. After that, the code utilises some spans created in the HTML to act as buttons for the gallery and zoom in function. Different on click events will lead to loading up different images, and to give the illusion of the button lighting up when selected. Demonstrated with the following HTML, CSS and Javascript:

JS:

for(bt of productImgBtn){

bt.classList.remove("active");

}

this.classList.add("active");

}

CSS:

.product-img-btn.active{

background-color: var(--text-color);

}

HTML:

<span class="product-img-btn active"></span>

<span class="product-img-btn"></span>

<span class="product-img-btn"></span>

Besides that, there is also an image enlarger within the code, the thumbnail will be animated with CSS as it enlarges.  
  
JS:

enlargeImgBtn.addEventListener("click", function() {

productGalleryImg.classList.toggle("enlarged");

});

CSS:

.enlarged {

transition: all 0.5s ease-in-out;

transform: scale(1.5);

}

The same button can be clicked again to close the enlarged image.

**Section 5: Small Features**

The website has smaller features such as smooth scrolling implemented when scolling “#” elements into view, as shown in this code below:

document.querySelectorAll('a[href^="#"]').forEach(anchor => {

anchor.addEventListener('click', function (e) {

e.preventDefault();

document.querySelector(this.getAttribute('href')).scrollIntoView({

behavior: 'smooth'

});

});

});

Besides that, a media query and some Javascript code also allowed for the navigation bar to shrink to a smaller size when the resolution is shrunk below 1200 pixels as seen here:

@media (max-width: 1200px){

.hamburger{

display: block;

}

Javascript is then used to animate the hamburger menu smoothly.

**Challenges Encountered:** The website is simply not as responsive as it can be, many elements will blend in together with font-size changing, and some nested divs do not take up space as their display is inline-block, changing it to block breaks some features.

If better planning and more media queries were used to make a more responsive design, the website would have met the requirements of the assessment.

**Summary and Conclusion:**

The website meets most of the main requirements of the assessment and I am quite satisfied with the outcome. The colours changing make the website more interactive, the image enlargement method is far more appealing then the “window.setInterval()” method to me, regular expressions helped make form validation far more easier, videos were added according to the product and find the website quite neat in design. The products page has empty spaces but centering the content breaks the product slider in the home page, ideally more products filling up the space would make it less jarring to look at.

If I had more time, I would commit to making the website more responsive as well as fixing the many design issues in the Products.html page and index.html page. Adding more products and making an “Add to Cart” function would also help sell the idea of the website being an e-commerce website more.

**Sources and References**

**Annotated Source Code:**

* CodeBrah (2021). *How To Change Font-Size Using JavaScript | Change Font-Size JavaScript*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=T1rLPD0zDB4 [Accessed 8 January 2023].
* CodingGate (2020). *How to change site colors & save in Local Storage Using Pure JS*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=vJ4ndwKlmCE [Accessed 5 January 2023].
* Devression (2021). *How To Make Ecommerce Website Image Slider with HTML CSS Javascript -EASY TUTORIAL*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=OQZNAMjC6Vg [Accessed 1 January 2023].
* Easy Tutorials (2022). *How To Make Product Page Design For Website Using HTML CSS And JS*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=uS032avlZ6I [Accessed 13 January 2023].
* Musanni, F. (2022). *Change Website Colors with CSS and Javascript | Light and Dark Mode Toggle*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=3Kterjsc74s [Accessed 2 January 2023].
* Pop Florin (2019). *JavaScript Client-side Form Validation*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=rsd4FNGTRBw [Accessed 5 January 2023].
* QuickCodingTuts (2022). *Build A Product Slider From Scratch | HTML, CSS, JS | Online Store Animations*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=lV-wGuAN0VY [Accessed 1 January 2023].
* Simplilearn (2020). *Email Validation In JavaScript | JavaScript Email Validation Tutorial For Beginners | Simplilearn*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=nRHCoOVSu5k [Accessed 5 January 2023].
* W3schools.com. (n.d.). *How To Create A Split Screen*. [online] Retrieved from: https://www.w3schools.com/howto/howto\_css\_split\_screen.asp [Accessed 9 January 2023].
* W3schools (n.d.). *How To Create Arrows/Triangles with CSS*. [online] W3schools.com. Retrieved from: https://www.w3schools.com/howto/howto\_css\_arrows.asp [Accessed 27 December 2022].

**Media Used:**

* Caves, J. (2018). *Choosing the Right Canvas for Your Painting - Jackson’s Art Blog*. [online] Jackson’s Art Blog. Retrieved from: https://www.jacksonsart.com/blog/2018/09/13/choosing-the-right-canvas-for-your-painting/ [Accessed 28 December 2022].
* Dreams Time (n.d.). *School stationery and Office supplies*. [online] 170,003 Office Supplies Stock Photos - Free & Royalty-Free Stock Photos from Dreamstime. Retrieved from: https://www.dreamstime.com/photos-images/office-supplies.html [Accessed 29 December 2022].
* Free Range (2023). *Free Stock Photo of Pens and Pencils | Download Free Images and Free Illustrations*. [online] Freerangestock.com. Retrieved from: https://freerangestock.com/photos/131097/pens-and-pencils-.html [Accessed 28 December 2022].
* Infinitely sound (2021). *Close Up of Artist’s Hand While Painting | Free Stock Video Footage HD 4K*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=6Aj\_keAAY-Y [Accessed 28 December 2022].
* Istockphoto.com. (2018). *Sculpting Clay Stock Photos, Pictures & Royalty-Free Images - iStock*. [online] Retrieved from: https://www.istockphoto.com/search/2/image?phrase=sculpting+clay [Accessed 30 December 2022].
* Monocle Films (2014). *Retail special: stationery shops - Prague, Vancouver & London*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=08dL2O8lTqs&t=238s [Accessed 28 December 2022].
* Motion Array (2018). *Fine Artist Drawing New Design Stock Footage*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=Qqgmt4Eq3s8 [Accessed 28 December 2022].
* Royalty (2018). *FREEDIO: "Scribbling school art " royalty free HD stock video footage*. *YouTube*. Retrieved from: https://www.youtube.com/watch?v=bvfG\_LOHkUM [Accessed 28 December 2022].
* Stationery World. (2022). *Paint - Stationery World*. [online] Retrieved from: https://stationeryworldonline.co.uk/paint/ [Accessed 13 January 2023].

‌