

Group Coursework Report

Informatics Institute of Technology

Department of Computing

Bsc(Hons) Artificial Intelligence and Data Science

Module: CM1605 Web Technology

Module Coordinator: Ms. Janani Harischandra

Tutorial Group : B

Coursework Group No : 6

Student Details: 2118518 | 20210822 | Mohamed Mishaal Rifath

2118954 | 20200509 | B. A. D. Yohan Abhishek

2117533 | 20210508 | M. Inuka Seniya Rodrigo

2117539 | 20210591 | Puspasothy Seyon

Table Of Contents

Table Of Contents	2
Introduction	4
Research on Existing Systems	5
1) Amazon.com	5
User-friendly design	5
Multiple payment options	5
Reliable customer service	5
User Reviews	5
Product Discounts	6
2) Daraz.lk	7
Functional and well designed	7
Optimized for mobile devices	7
Clear purpose	7
Easy navigation	7
Fast page loading	7
Technical Discussion	8
Student 1:	8
Presentation Page:	8
Sign up page:	9
Student 2:	15
Main/Navigation	15
Video	16
CSS	16
Form	17
Quiz	18
Student 3:	20
Buy-Products page	20
Change font size	23
About us page	24
Student 4:	25
Creating a well-formed xml file to store information of listing items	25
JavaScript to read and display information from the xml file	26
Creating JavaScript functionality to change the background and text color of the listing page	29
Changing text color:	32
JavaScript validation to check compulsory fields are filled and valid email address in newsletter Form	32
Discussion of UX/UI principles	37

Navigation Techniques	37
Color Balance / Selection / Consistency	38
Color Contrast Test	38
Typography / Consistency	39
Accessibility	39
Accessibility Test	40
Site Diagram	41
Self-Reflection	42
Puspasothy Seyon-Student 1	42
M. Mishaal Rifath - Student 2	42
Yohan Abhishek-Student 3	42
Inuka Rodrigo - Student 4	43

Introduction

Topic Chosen: Ecommerce Site – BuyAnything.com

Features Available:

- Sign up form to learn user preferences.
- Product purchasing page.
- Gallery page, displaying a variety of products.
- Listing page, displaying past user purchases.
- About us page.
- Feedback form for the user.
- Interactive quiz to earn discounts.

Members:

Member 1: 2118518 | 20210822 | M. Mishaal Rifath - Student role 2

Member 2: 2118954 | 20200509 | B. A. D. Yohan - Student role 3

Member 3: 2117533 | 20210508 | M. Inuka Seniya Rodrigo - Student role 4

Member 4: 2117539 | 20210591 | Puspasothy Seyon - Student role 1

Research on Existing Systems

1) Amazon.com

- User-friendly design

Amazon web site has an appealing user experience which facilitates customers to find what they are looking for.

It has prioritize customer experience by,

* Adding a search bar

* High quality home page which is simple and straightforward

* Listing category pages in the navigation bar

- Multiple payment options

Amazon offers multiple payment options such as debit and credit card, PayPal, Stripe to make the purchasing stage easy for the customers. This will improve the chance of fastening the final checkout process.

- Reliable customer service

Amazon provides 24/7 customer service via a chatbot to address the customer needs at any time.

- User Reviews

In an online website, customers won't have a chance to see a product in person. Therefore, they should rely on what previous buyers have said about the products. Including a section to write product reviews helps customers to decide if they want to buy the product or not. Amazon has already included this feature to increase customer satisfaction.

- Product Discounts

Amazon encourages its customers to create Amazon accounts so that they can get loyalty pricing offers. So customers will feel that they are getting a good deal and this could lead to increasement of sales.

2) Daraz.lk

- Functional and well designed
 - The webpage of an ecommerce store must be well designed because it represents the company, its products, and its services. As a result, the page must appear visually appealing and, most importantly, professional. As important as the design is, the site must also respond quickly to the needs of the users.
- Optimized for mobile devices
 - In today's world, there are a variety of devices, so it's critical that your website is mobile-friendly. Daraz has optimized its services for mobile devices, which is a huge benefit given how popular mobile devices are. Daraz has its own mobile app as well.
- Clear purpose
 - Every company has a stated purpose, and the user should be able to see it within the first few seconds of visiting the website.
- Easy navigation
 - The user experience on the web is heavily influenced by navigation. Customers will find it easier to find what they're looking for with good navigation, whereas customers will be frustrated with poor navigation. Daraz has a primary navigation menu that includes links to the site's most important pages. To make it faster and easier to get to any page on the website, daraz uses a dropdown navigation menu.
- Fast page loading
 - We live in a fast-paced world where we expect immediate responses to our questions. In order to provide a positive user experience, daraz loads its website quickly. This could be a result of investing in a good hosting service.

Technical Discussion

Student 1:

I was given the role of Student 1, which is discussed below:

Presentation Page:

This page gives an intro to all the group members and is designed to move to the main page automatically after 5 seconds.

```
< script >

setTimeout(function()

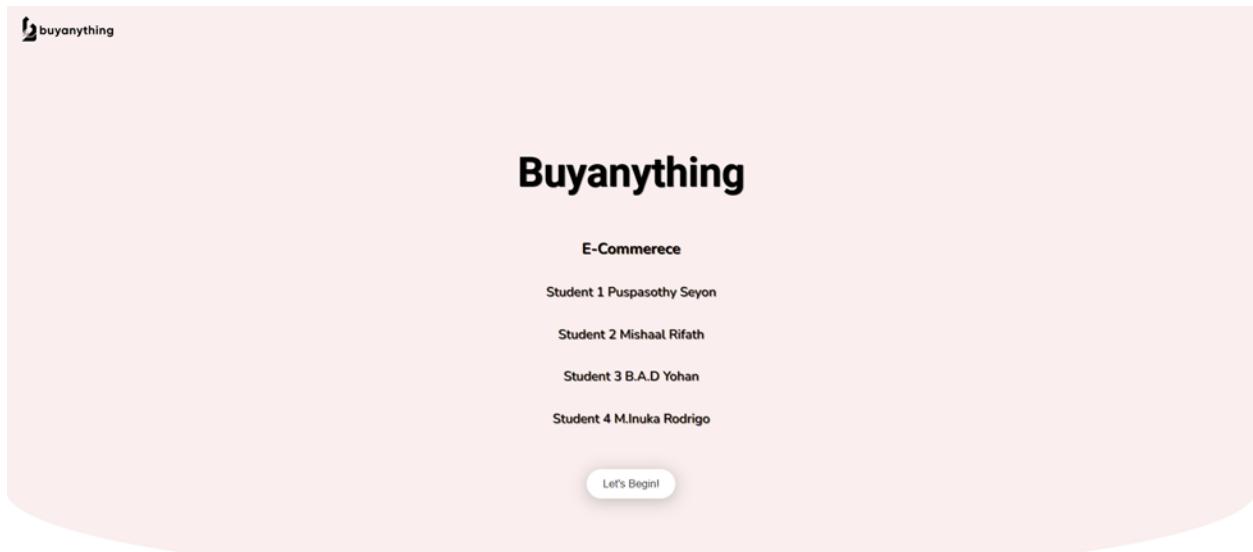
{

    window.open('main.html', '_blank');

}, 5000);

< / script >
```

“setTimeout()” method calls off a function after a certain time.



Sign up page:

Sign up page requires the user to enter 5 compulsory details and choose 3 compulsory preferences in order to sign up.

```
<script>

    function validateForm() {

        let uname = document.Sign_up_Form.uname;

        let age = document.Sign_up_Form.age;

        let mobilenumber = document.Sign_up_Form.mobilenumbers;

        let address = document.Sign_up_Form.address;

        let email = document.Sign_up_Form.email;

        let password = document.Sign_up_Form.password;

        let Preference1 = document.Sign_up_Form.Preference1;

        let Preference2 = document.Sign_up_Form.Preference2;

        let Preference3 = document.Sign_up_Form.Preference3;

        if (uname.value == "") {

            alert("Name is required");

        }

        else if (age.value == "") {

            alert("Age is required");

        }

        else if (mobilenumbers.value == "") {
```

```

        alert("Mobile Number is required");

    }

    else if (address.value == "") {

        alert("Address is required");

    }

    else if (email.value == "") {

        alert("Email is required");

    }

    else if (password.value == "") {

        alert("Password is required");

    }

    else if (Preference1.value.length <= 0) {

        alert("Choose a Preference 1");

    }

    else if (Preference2.value.length <= 0) {

        alert("Choose a Preference 2");

    }

    else if (Preference3.value.length <= 0) {

        alert("Choose a Preference 3");

    }

}

else {

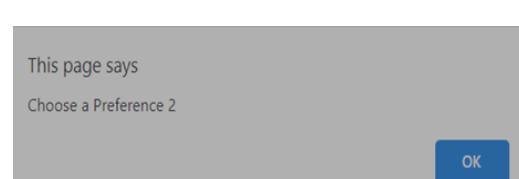
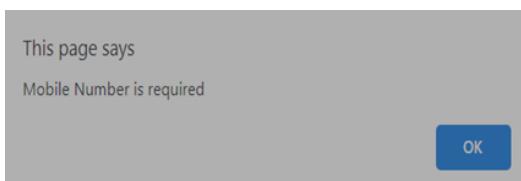
    alert("Dear"+ " " +document.Sign_up_Form.uname.value+ " ,Thank
you for signing up with us, the recommended results will be shown in a
while.");

}

window.location.href = "main.html";

```

```
        }  
  
    }
```



Only if all the fields are the user will get a message

"Dear "username entered", Thank you for signing up with us, the recommended results will be shown in a while.")

This page says

Dear Thanos ,Thank you for signing up with us, the recommended results will be shown in a while.

OK



Username
Enter Username

Age
Enter Age

Mobile Number
Enter Mobile Number

Address
Enter Address

Email
Enter example@mail.com

Password
Confirm Password

Preference 1 Clothing Fashion Electronics
Preference 2 Adidas Nike AirJorden
Preference 3 Used New Both

Login

Gallery Page:

Lastly I created the gallery page to display the different types of collection we got, where the user can see the thumbnail image with description which will enlarge upon clicking the image.

```
<script>  
function myFunction(imgs) {
```

```

let expandImg = document.getElementById("expandedImg");

let imgText = document.getElementById("imgtext");

expandImg.src = imgs.src;

imgText.innerHTML = imgs.alt;

expandImg.parentElement.style.display = "block";

}

</script>

```

Here, “imgtext” is also considered a description of the image.

The collage consists of five images arranged in a grid-like layout:

- Top Left:** A pair of blue jeans and white sneakers on a dark surface.
- Top Middle:** A row of luxury cars (BMW) parked in a parking lot.
- Top Right:** A curved row of bookshelves filled with books in a library.
- Middle Left:** Various electronic devices including a smartphone, a laptop, a VR headset, and a pair of headphones.
- Middle Right:** A collection of makeup brushes in a white holder.

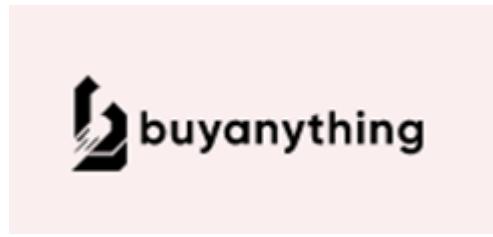
Below each image is a caption:

- Choose From a Large Variety of Brands**
- Buy and Sell At Your Own Time**
- Learn From Anywhere**
- We Prioritize Safety**
- Make Yourself Look Good**

Bottom Row:

- buyanything** logo
- © 2022 Buyanything.
All Rights Reserved.
- Useful links**
[Home](#)
[Gallery](#)
- Contact Us**
[Editors Page](#)
[About Us](#)

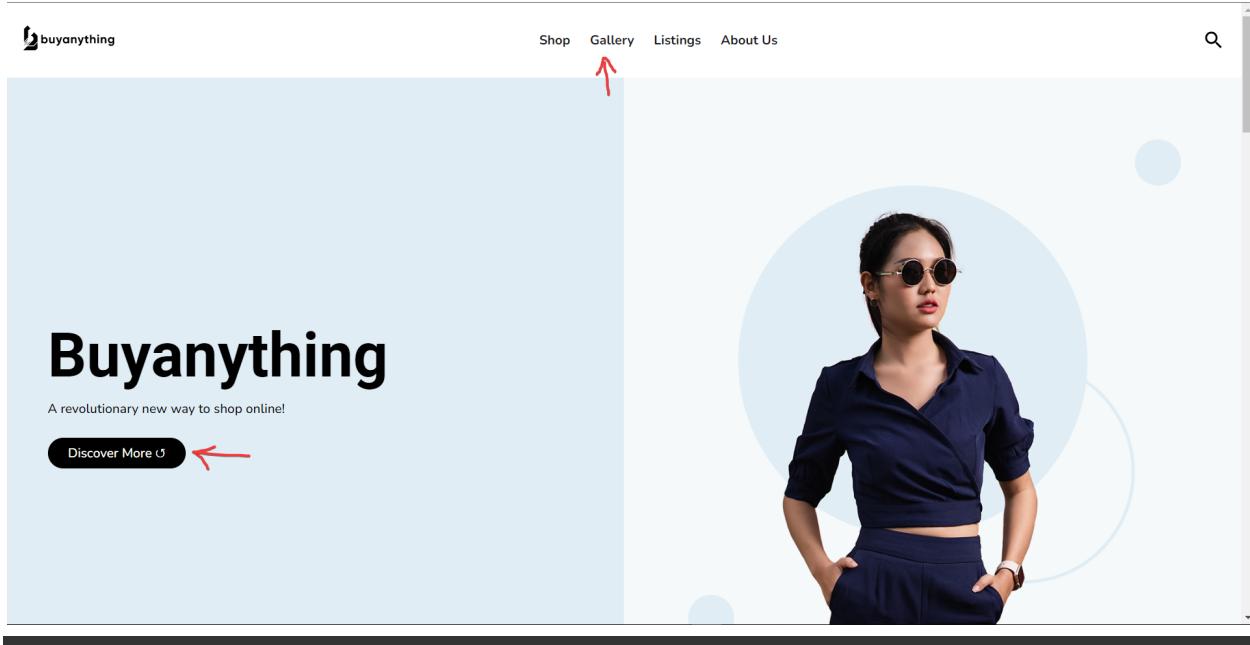
The logo is added to every page:



Student 2:

Main/Navigation

Users can navigate the website from links in the header & footer as well as through buttons found throughout the page.



```
<header>
    <div class="navbar">
        <a href="main.html"></a>
        <nav>
            <ul class="links" id="menuItems">
                <li><a href="productPage.html">Shop</a></li>
                <li><a href="gallery.html">Gallery</a></li>
                <li><a href="listings.html">Listings</a></li>
                <li><a href="aboutUs.html">About Us</a></li>
            </ul>
        </nav>
        
    </div>
</header>
```

Video

What We Do



Buyanything is an interactive AI powered web application which is capable of better understanding our customer search queries and what is the

```
<video controls id="demoReel">
  <source src="assets\teaser.mp4" type="video/mp4"
  Your browser doesn't support the video tag.
</video>
```

CSS

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Navigation styling:

```
nav li, nav a {
  font-family: "Nunito", sans-serif;
  font-weight: 700;
  font-size: 16px;
  color: #1a1a1d;
  text-decoration: none;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 25px 20px;
}

.logo {
  cursor: pointer;
  height: 30px;
}

.links {
  list-style: none;
  padding: 10px;
}

.links li {
  display: inline-block;
  padding: 0 10px;
}

.links li a {
  transition: all .3s ease 0s;
  text-decoration: none;
}

.links li a:hover {
  color: #ff6359;
}
```

Form

The image displays two versions of a feedback form. Both versions have a dark header bar with the title "Feedback Form" and a sub-label "Name". Below the header are four input fields: "Enter Name", "Email Address", "Enter Email Address", and "Comments" (with placeholder "Enter Comments Here"). A "Rating" dropdown menu is set to "Excellent". At the bottom are two red buttons labeled "Reset" and "Submit". The right version includes a validation message "Name is required, Email is required" at the bottom.

Validation:

```
form.addEventListener('submit', (e) => {
  let messages = []
  if (name.value === '' || name.value == null) {
    messages.push('Name is required')
  }

  if (email.value === '' || email.value == null) {
    messages.push('Email is required')
  } else if (ValidateEmail(email.value) == false) {
    messages.push('Enter Valid Email')
  }

  if (messages.length > 0) {
    e.preventDefault()
    errorElement.innerText = messages.join(', ')
  } else if (comment.value === '' || comment.value == null) {
    alert(`Dear ${name.value}, Thank you very much for your feedback.
You have rated our site as ${rating.value}`)
  } else {
    alert(`Dear ${name.value}, Thank you very much for your feedback.
You have rated our site as ${rating.value} and your comment was
${comment.value}`)
  }
});
```

```
// Checks if email address is valid using regular expression
function ValidateEmail(inputText)
{
    let mailformat =
/^(([^\<>()[]\.,;:\s@"]+(\.[^<>()[]\.,;:\s@"]+)*|(.+))@((\[[0-9]{1,3}
\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-\-0-9]+\.)+[a-zA-Z]{2,}))$/
;
    return mailformat.test(inputText.toLowerCase());
}
```

Quiz

```
function startQuiz() {
    let userPoints = 0;

    let question1 = prompt("Is Buyanything an Ecommerce Site? (Yes/No)");
    userPoints = evalYesNo(question1.trim(), userPoints);
    let question2 = prompt("Does our website use AI and machine learning?
(Yes/No)");
    userPoints = evalYesNo(question2.trim(), userPoints);
    let question3 = prompt("Does our website suggest products based on a
customer's past behavior? (Yes/No)");
    userPoints = evalYesNo(question3.trim(), userPoints);
    let question4 = prompt("Can our website predict context from our
customer's search queries? (Yes/No)");
    userPoints = evalYesNo(question4.trim(), userPoints);
    let question5 = prompt("Do we offer a various selection of products?
(Yes/No)")
    userPoints = evalYesNo(question5.trim(), userPoints);

    switch (evalUserscore(userPoints)) {
        case 0:
            alert(`Thanks for taking our quiz! Unfortunately you haven't
earned a badge`);
            break;
        case 1:
            alert(`Congratulation! You have earned ${userPoints} points
with a Bronze badge, please claim the points in your next purchase`);
            break;
        case 2:
            alert(`Congratulation! You have earned ${userPoints} points
```

```
with a Silver badge, please claim the points in your next purchase`);  
        break;  
    case 3:  
        alert(`Congratulation! You have earned ${userPoints} points  
with a Gold badge, please claim the points in your next purchase`);  
        break;  
    }  
    userBadge = evalUserscore(userPoints);  
}
```

Student 3:

Buy-Products page

First task was to create a product page to buy products (with at least 2 different products)
A screenshot of this section can be seen below

The screenshot shows a 'SHOP PRODUCTS' page with three items displayed in a row:

- WATER BOTTLE**
LKR 2000
In stock: 20 items
- WHITE CAP**
LKR 1500
In stock: 20 items
- TRAVEL BAG**
LKR 5500
In stock: 20 items

Each item has a 'Buy' button below it. A shopping cart icon with the number '0' is located in the top right corner.

```
function addToCartSection(imageName, name, price, stock) {
    /*This is the main method of the add to cart methods. Validates the
    quantity of items required by customer and adds to cart */
    if(stock==0){
        alert(name+" is out of stock at the moment!")
    } else{
        let quantity = prompt("Enter quantity: ");
        if(isNaN(quantity) || quantity=="" || quantity<=0 || quantity==null){
            alert("Please select a valid quantity!")
        }
        else {
            if(stock>=quantity){
                cartItems.push(name+","+quantity)
                cartTotalPrice += price*quantity;
                changeStockText(name.split(" ")[1],quantity)
                addToCartBox(imageName, name, price, quantity);
            }
        }
    }
}
```

```

        else{
            alert("There is only "+stock+" in stock.");
        }
    }
}
}

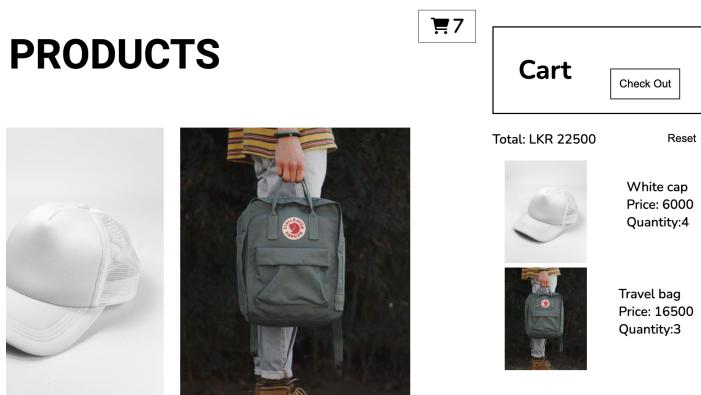
```

With the use of additional javascript and css knowledge, the cart was created as a side view panel, with two states as opened and closed to make the page look more professional.

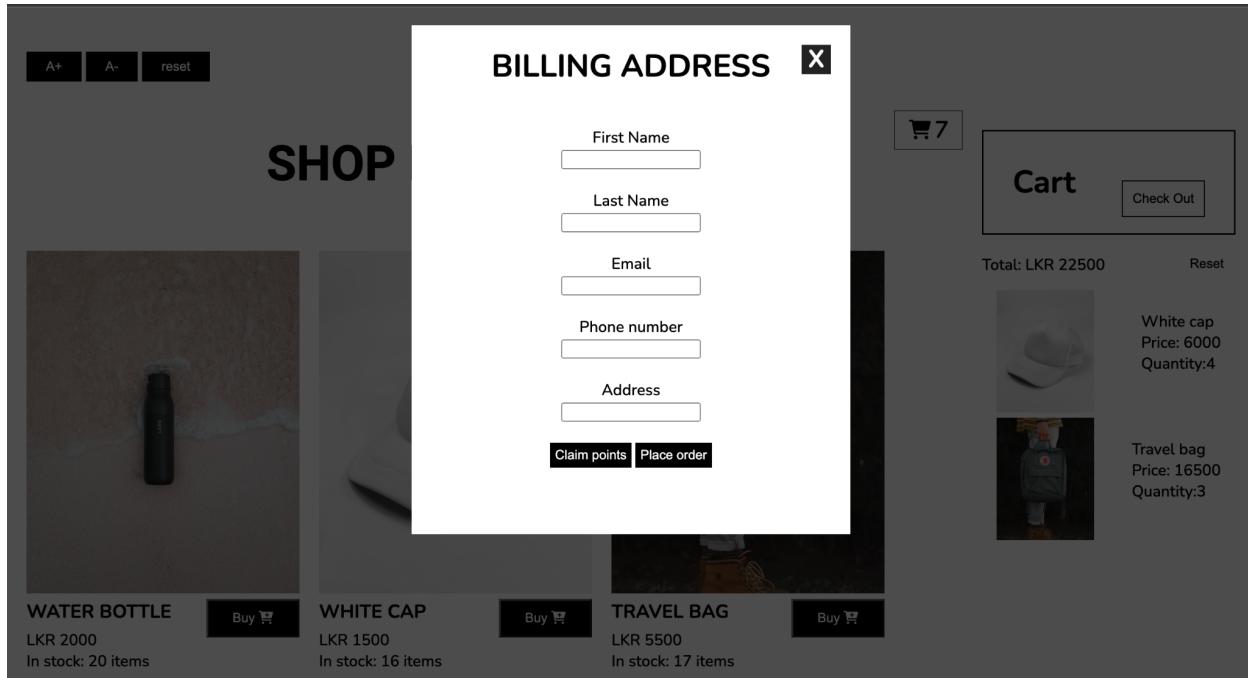
(closed state)



(opened state)



Given below is the form required by the specification. In order to make it look more appealing, additional css and js were used.



Below is the code to validate the form

```

function showAndClearField(formReference) {
    /*This method validates the form and alerts
    the customer of the order been placed */
    customer.fname = formReference.firstName.value;
    customer.lname = formReference.lastName.value;
    customer.mail = formReference.email.value;
    customer.phone = formReference.phoneNumber.value;
    customer.address = formReference.address.value;

    if(customer.fname === "") {
        alert("Please fill your first name!");
    } else if(customer.lname === "") {
        alert("Please fill your last name!");
    } else if(customer.mail === "") {
        alert("Please fill your last name!");
    } else if(customer.address === "") {
        alert("Please fill your last name!");
    } else if(customer.phone === "") {
        alert("Please fill your contact number!");
    } else{
        toPrint = "Order summary\n\n";
        for (let i=0; i<cartItems.length; i++){
            toPrint += "Name: "+cartItems[i].split(",") [0]+ " | ";
    }
}

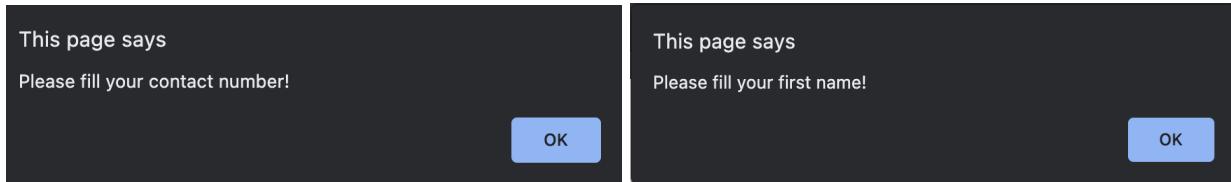
```

```

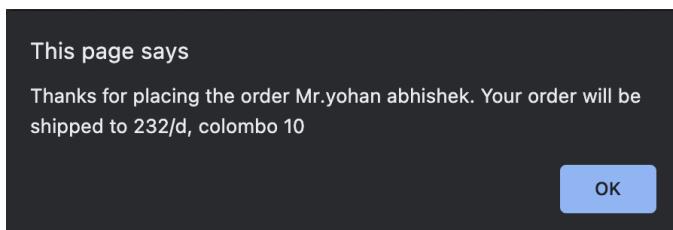
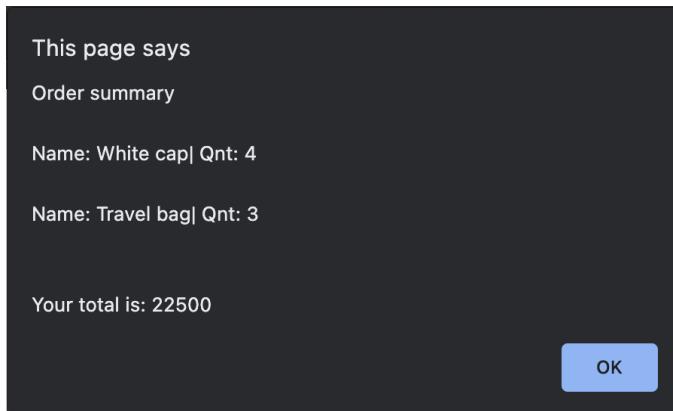
        toPrint += "Qnt: "+cartItems[i].split(",")[1]+"\n\n";
    }
    toPrint+='\nYour total is: '+cartTotalPrice;
    alert(toPrint)
    alert("Thanks for placing the order Mr."+customer.fname+" "+customer.lname+".");
    Your order will be shipped to "+customer.address);
    document.getElementById("popup-1").classList.toggle("active");
    document.getElementById("content").innerHTML = "";
    cartTotalPrice = 0;
    cartItems = [];
    document.getElementById("totalPrice").innerHTML = "Total: LKR "+cartTotalPrice;
    document.getElementById("cart-display-qnt").innerHTML = "";
}
}

```

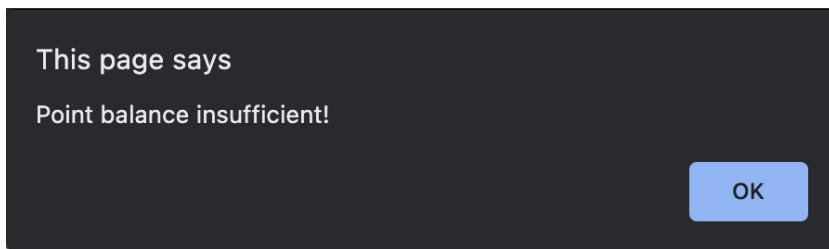
An example of the js validation used for the form is given below



When the user is done with selecting items and adding to cart, the checkout button can be clicked and it will load the above form. After successfully filling the form, the total bill will be displayed as below

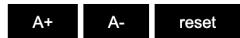


If the user redeems the points from the quiz, an alert will be displayed, with the new cart total. If the user has no points, the below given alert will be displayed.



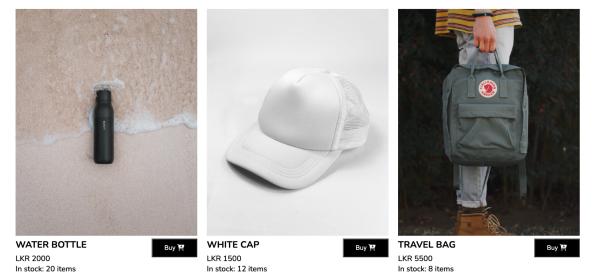
Change font size

In order to change the font size of the product page, a plus, minus and reset buttons were included as below.



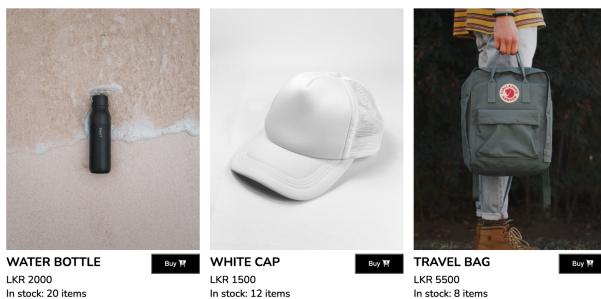
Before changing text size;

SHOP PRODUCTS



After changing text size;

SHOP PRODUCTS



Given below is the function to change font size.

```
function changeFontSize(type){  
    /*This function changes the font size of  
    the product page and about us page*/  
    let id = ["#resetFont"];  
    let element = document.querySelector(id);  
    let fontSize = window.getComputedStyle(element,  
null).getPropertyValue("font-size");  
    fontSize = parseFloat(fontSize);  
  
    if(type === "increase"){  
        element.style.fontSize = (fontSize+1)+"px";  
    }  
    else if(type === "decrease") {  
        element.style.fontSize = (fontSize-1)+"px";  
    }  
    else{  
        element.style.fontSize = (16)+"px";  
    }  
}
```

About us page

Code used to display the student information is given below.

```
function displayMemberInfo(name, studentNumber){  
    document.getElementById("student"+studentNumber).innerHTML =  
name+"<br>"+ Student: "+studentNumber;  
}  
function removeInfo(studentNumber){  
    document.getElementById("student"+studentNumber).innerHTML = null;  
}
```

The about us page can be accessed from the header and the footer. Given below is a screenshot of the page

About us

[Contact](#)[Contact](#)[Contact](#)[Contact](#)

The name and the student roll carried out by each student can be viewed by hovering over each photo as shown below. By clicking on the contact button, the page directs the user to our personal emails.

[Contact](#)

Mishaal Rifath
Student: 2

Student 4:

Creating a well-formed xml file to store information of listing items

Sample code:

```
<product>

    <category>LIFESTYLE</category>

    <name>addidas Men's Grand Court SE Sneaker</name>

    <actualPrice>Rs.45,000</actualPrice>

    <discountedPrice>Rs.42,500</discountedPrice>

    <discountRate>7% off</discountRate>

</product>

<product>

    <category>HEADPHONE</category>

    <name>Sony ZX Series Wired On-Ear Headphones with Mic, Black MDR-ZX110A</name>

    <actualPrice>Rs.75,000</actualPrice>

    <discountedPrice></discountedPrice>

    <discountRate></discountRate>

</product>

</purchasedOrders>
```

Explanation:

Used `<product>` element to store information of each listing item.

In some items, <discountPrice> and <discountRate> element values remain empty because all the products will not receive any discounts.

JavaScript to read and display information from the xml file

Sample code:

```
<script>

let xmlContent = "";

let allTheProducts = document.getElementById('p-items');

fetch('CWListingPage(XML).xml').then((response)=>{

    response.text().then((xml)=>{

        xmlContent = xml;

        let parser = new DOMParser();

        let xmlDOM = parser.parseFromString(xmlContent, 'application/xml');

        let product_info = xmlDOM.querySelectorAll('product');

        let counter = 1;

        let commonNameCategory = 'categoryOfProduct';

        let commonNameProduct = 'nameOfTheProduct';

        let commonNameActualPrice = 'actualPriceOfTheProduct';

        let commonNameDiscountedPrice = 'discountedPriceOfTheProduct';

        let commonNameDiscountRate = 'discountedRateOfTheProduct'

        product_info.forEach(productXmlNode =>{

            let categoryId = commonNameCategory.concat(counter);

            document.getElementById(categoryId).innerHTML =
productXmlNode.children[0].innerHTML;
```

```
let productNameId = commonNameProduct.concat(counter);

document.getElementById(productNameId).innerHTML =
productXmlNode.children[1].innerHTML;

let productPriceId = commonNameActualPrice.concat(counter);

document.getElementById(productPriceId).innerHTML =
productXmlNode.children[2].innerHTML;

let productDiscountedPriceId = commonNameDiscountedPrice.concat(counter);

if(document.getElementById(productDiscountedPriceId)) {

    document.getElementById(productDiscountedPriceId).innerHTML =
productXmlNode.children[3].innerHTML;

}

let productDiscountedPriceRateId = commonNameDiscountRate.concat(counter)

if(document.getElementById(productDiscountedPriceRateId)) {

    document.getElementById(productDiscountedPriceRateId).innerHTML =
productXmlNode.children[4].innerHTML;

}

counter++;

});

});

});

</script>
```

Explanation for the steps taken to retrieve information

1.Read the entire file as a string from the xml file then provide a function which is going to be called whenever the result will be returned, and this function has 'response' as an argument.

Here, 'response' is an object.

2.Extract the text from the 'response' object and then parse it to the variable called 'xmlContent'.

3.Created the XML DOM Parser and also created the XML DOM Object

Name of the XML DOM Parser = 'parser'

Name of the XML DOM Object = 'xmlDOM'

4.Used a variable named 'product_info' to get information of all products with the use of 'xmlDOM' object and the inbuilt function called querySelectorAll().

'product_info' is a list

5.Iterate through each value from the "product_info" list and retrieve and send details to appropriate places to display in the web page, using the following code.

```
document.getElementById(IDName).innerHTML=productXmlNode.children[Number].innerHTML;
```

productXmlNode is an element in 'product_info' list.

Creating JavaScript functionality to change the background and text color of the listing page

Sample code

```
<script>

function displayOptions() {
    document.getElementById("bgColour").classList.toggle("show");
}

function displayOptionsForText() {
    document.getElementById("textColour").classList.toggle("show");
}

function changeBGColour(color){
    document.body.style.background = color;
}

function changeTextColour(color){
    let products = document.getElementById("p-items");
}
```

```

        for(let counter =
0;counter<products.getElementsByClassName("product").length;counter++){

            let oneProduct = products.getElementsByClassName("product")[counter];

            let productInformation =
oneProduct.getElementsByClassName("product-info")[0];

            let categoryOfTheProduct =
productInformation.getElementsByClassName("product-info-top")[0];

            let p_category =
categoryOfTheProduct.getElementsByClassName("category-title")[0];

            p_category.style.color = color;

            let p_name = oneProduct.getElementsByClassName("product-name")[0];

            p_name.style.color = color;

        }

    }

</script>

```

Explanation of each function:

1. function displayOptions():

Take values given in each tag which defines under

tag with the id value “bgColour”,and then display it in the web page whenever user clicks on button with the value of “Change background color”.

2. function displayOptionsForText():

Same logic used in displayOptions() function will be applied here.

3. function changeBGColour(color):

When this function is called, the background color of the web page will change.

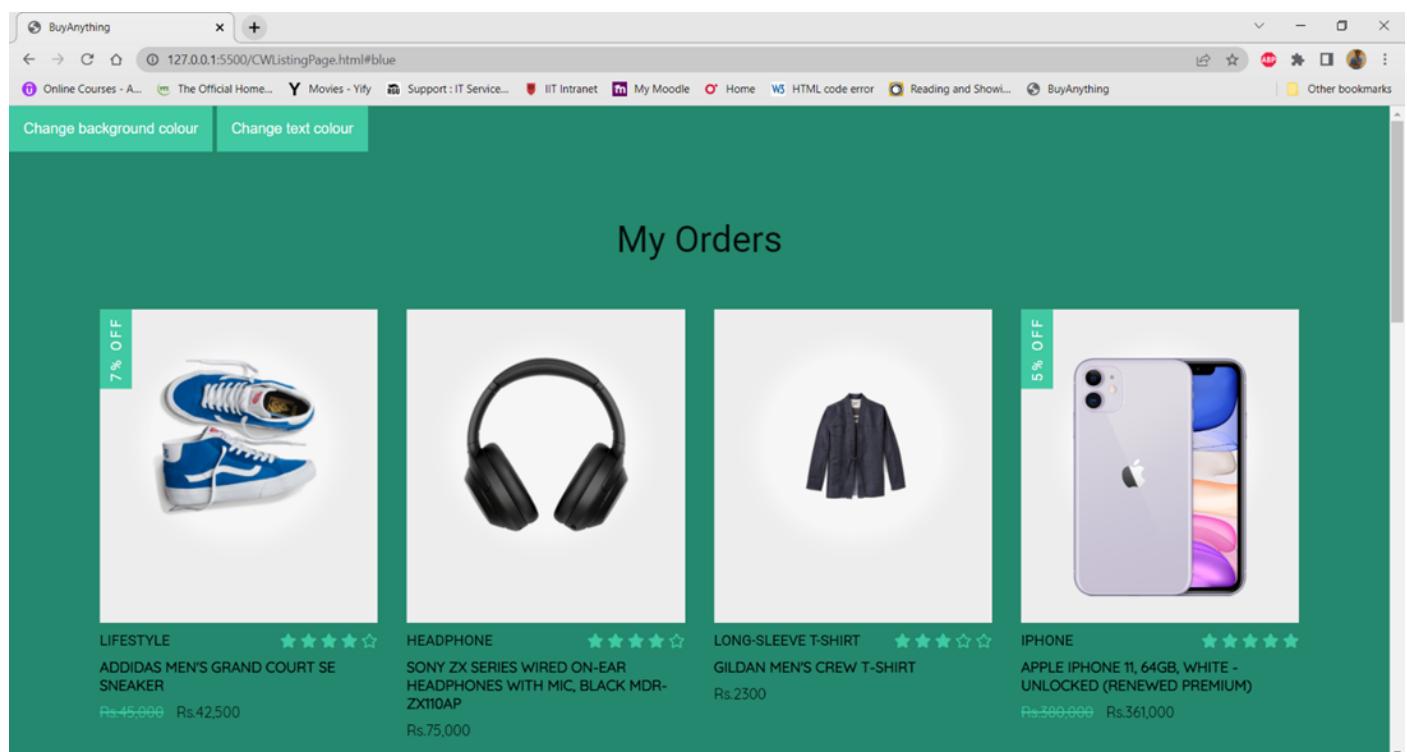
4. function changeTextColour(color):

“products” is a list which consists of all the product details.

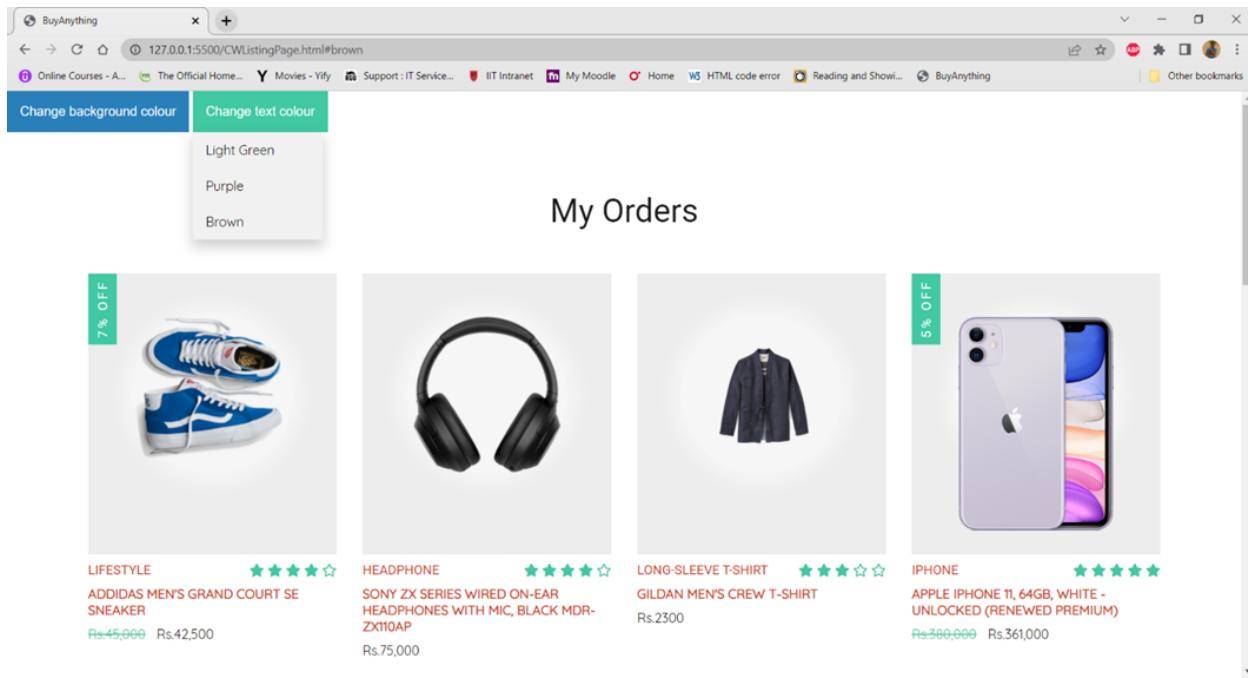
In this function, it will iterate through each element in the ‘products’ list and change the text color of product information and product category.

Output images:

Changing back-ground color:



Changing text color:



JavaScript validation to check compulsory fields are filled and valid email address in newsletter Form

```
<script>

    const name = document.getElementById('form-name');

    const email = document.getElementById('form-email');

    const form = document.getElementById('form');

    const emailStructure =
/^([a-zA-Z0-9\._]+@[a-zA-Z0-9]+\.[a-zA-Z]+(\.[a-zA-Z]+)?$/;

    form.addEventListener('submit',(e)=>{

        if(name.value===' ' || name.value==null){

            if(email.value===' ' || email.value==null){
```

```

        alert('Name and email both must be filled out.');

    }

    else if(emailStructure.test(email.value)!=true){

        alert('Name must be filled out and invalid email ID');

    }else{

        alert('Name must be filled out.');

    }

    e.preventDefault();

}

else{

    if(email.value===' ' || email.value==null){

        alert('Enter an email id.');

        e.preventDefault();

    }else if(emailStructure.test(email.value)!=true){

        alert('Invalid email ID');

        e.preventDefault();

    }else{

        //Success

        alert('Dear '+name.value+', you have successfully subscribed  
for a personalized newsletter.')

    }

}

});

</script>

```

Explanation on the code:

Used inbuilt function “addEventListener” to run when user clicks on subscribe button.

1. Checks if the user has given a name

* If the name was not given,

#Then checks if the email was not given,

If both were not given , 'Name and email both must be filled out.' message will be displayed as a pop up message in the web page.

If an email was given, then checks if the email is a valid email ID.

If the email address is an invalid valid address, 'Name must be filled out and invalid email ID' message will be displayed in the web page.

If the email address is a valid address, 'Name must be filled out.' message will be displayed as a pop up message in the web page.

*If the name was given,

Then checks if the email was not given,

If the email address was not given, 'Enter an email id.' message will be displayed as a pop up message in the web page.

#If the email was given,

Checks if the address given by user is valid or not,

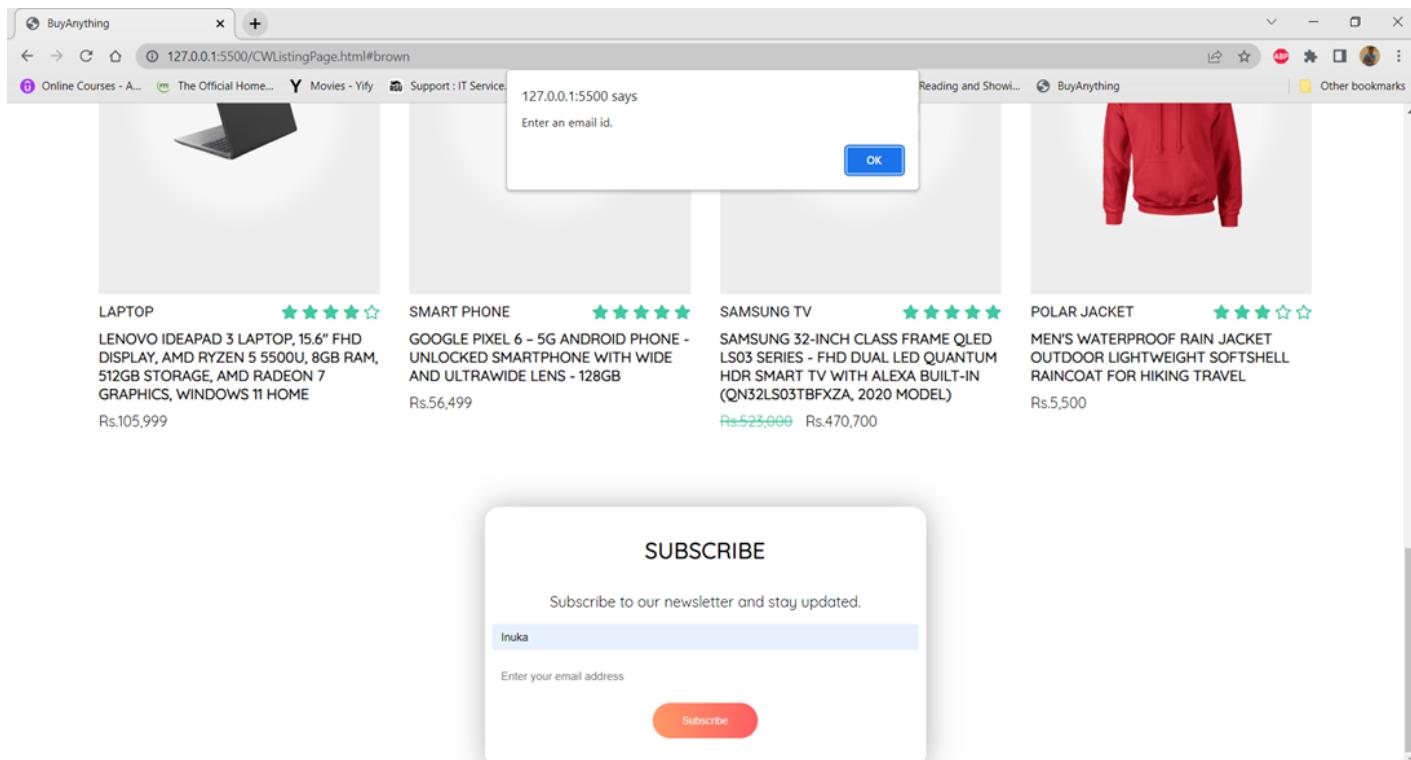
If the email address is an invalid valid address, 'Invalid email ID' message will be displayed on the web page.

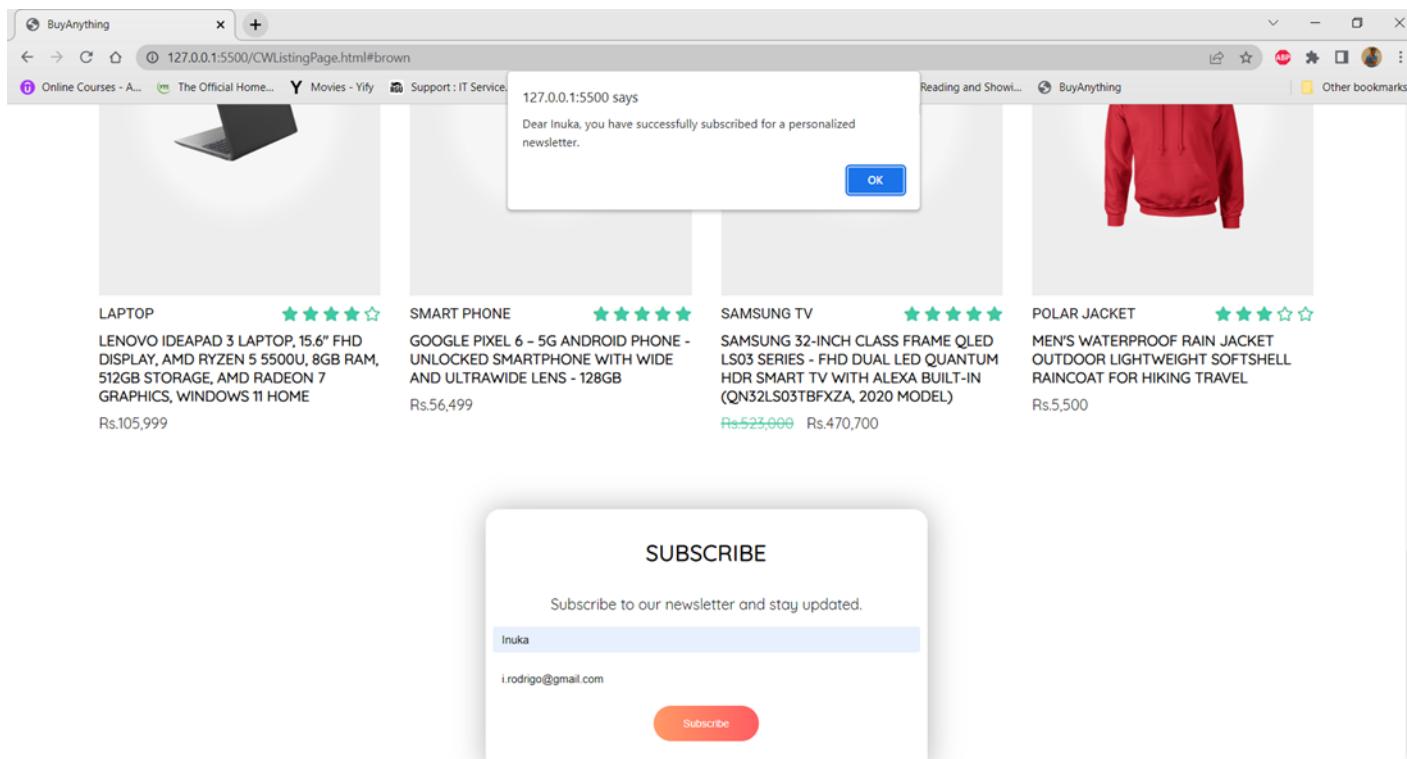
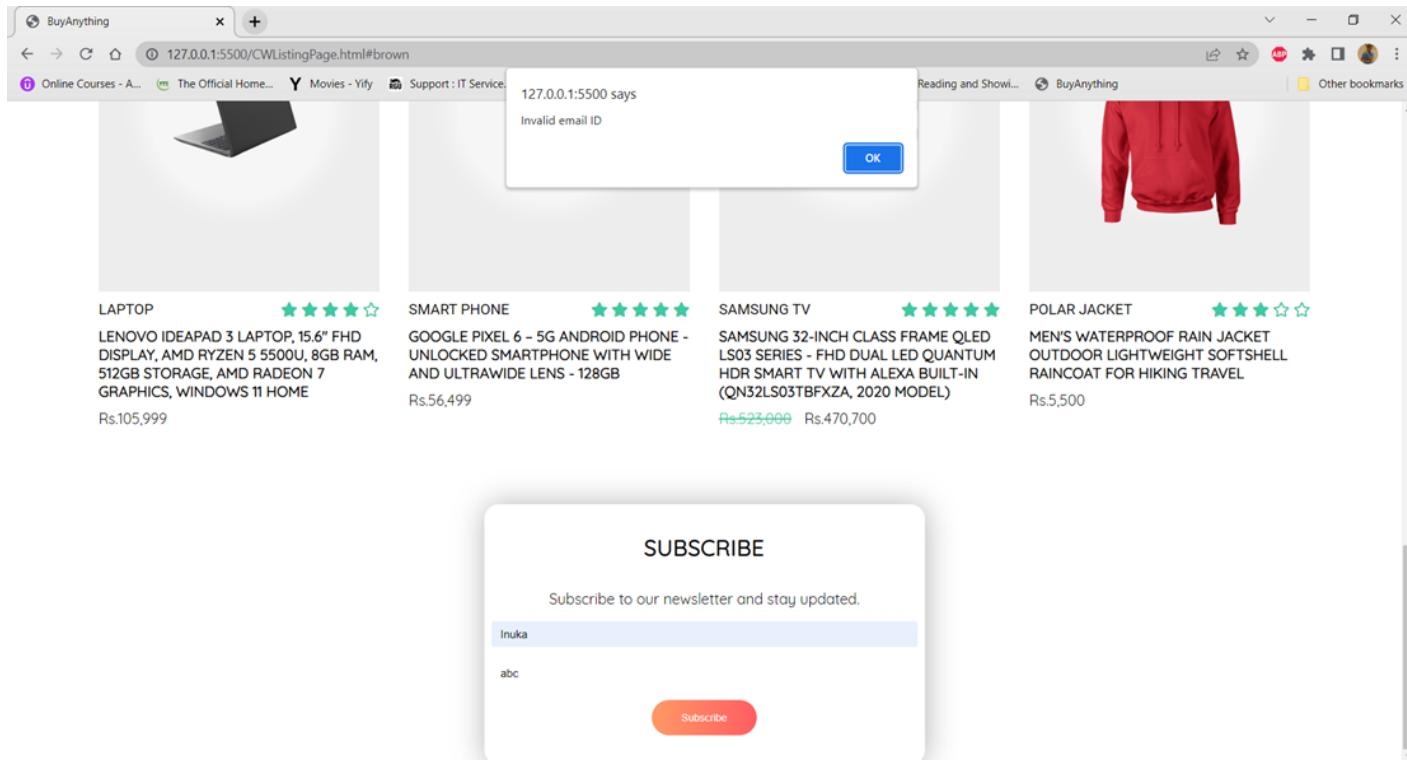
If it is valid, 'Dear '+name.value+', you have successfully subscribed for a personalized newsletter.' Message will be displayed in the web page as a pop up message and the form will be submitted and be sent to the relevant email address.

```
e.preventDefault();
```

```
//This will prevent submitting if the textFields are empty or the email address is an invalid email address.
```

Output images:





Discussion of UX/UI principles

Navigation Techniques

By using a consistent header and footer that contain discoverable navigation links on all pages, the user is able to easily jump to different sections of the website at their convenience.

Implemented as followed:

```
<header>
  <div class="navbar">
    <a href="main.html"></a>
    <nav>
      <ul class="links" id="menuItems">
        <li><a href="productPage.html">Shop</a></li>
        <li><a href="gallery.html">Gallery</a></li>
        <li><a href="listings.html">Listings</a></li>
        <li><a href="aboutUs.html">About Us</a></li>
      </ul>
    </nav>
    
    
  </div>
</header>
```

```
<footer>
  <div class="footerContainer">
    <div class="row">
      <div class="footer1">
        
      </div>
      <div class="footer2">
        <h3>© 2022 Buyanything.</h3>
        <p>All Rights Reserved.</p>
      </div>
      <div class="footer3">
        <h3>Useful links</h3>
        <ul class="footerLinks">
          <li><a href="main.html">Home</a></li>
```

```

        <li><a href="gallery.html">Gallery</a></li>
    </ul>
</div>
<div class="footer4">
    <h3>Contact Us</h3>
    <ul class="footerLinks">
        <li><a href="#">Editors Page</a></li>
        <li><a href="aboutUs.html">About Us</a></li>
    </ul>
</div>
</div>
</div>
</footer>

```

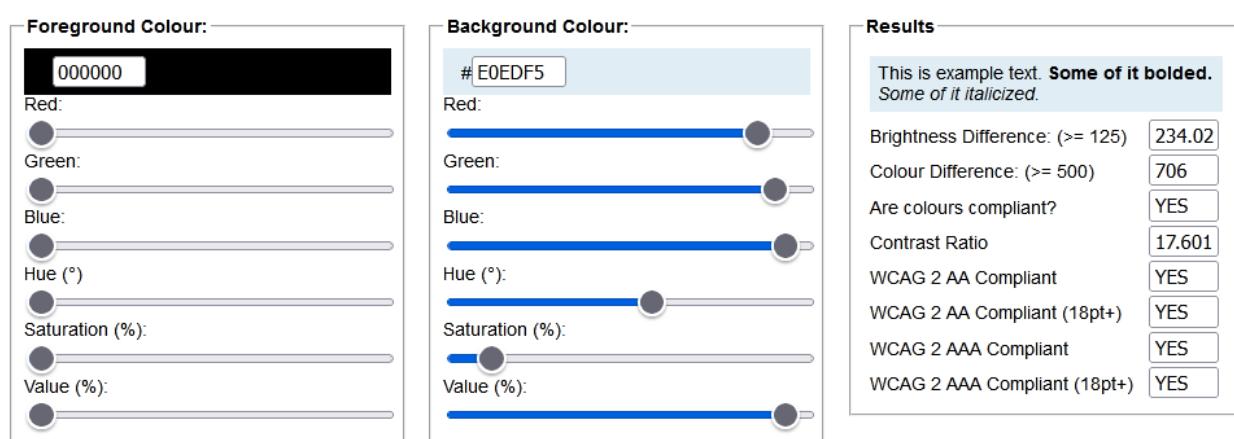
Color Balance / Selection / Consistency

By using a visually appealing and contrasting color palette, high-resolution images, bold icons and logos we were able to ensure the site is both accessible and captivating for our users.

Snippet:

```
.preview {
    margin: auto;
    padding-left: 50px;
    padding-right: 50px;
    background: linear-gradient(90deg, #e0edf5 50%, #f5f9fa 50%);
}
```

Color Contrast Test



Typography / Consistency

Two fonts were used for our website. Roboto, a bold style, was used for our headings, in order to stand out to the user, whilst remaining legible. Nunito, a soft and friendly style, was used for most of our body text.

Appropriate sizes and weights were utilized for different elements (e.g. header) to ensure readability.

Relevant CSS code:

```
body {  
    font-family: "Nunito",sans-serif;  
    font-weight: 500;  
}  
  
#header {  
    font-family: "Roboto",sans-serif;  
    font-weight: 700;  
    font-size: 50px;  
    padding: 25px;  
    margin: 25px 0;  
    text-align: center;  
}
```

Accessibility

To ensure our website remained accessible, we utilized the following

- Headings used appropriately for non-visual users to understand layout of the page.

```
<h1>Buyanything</h1>  
<p>A revolutionary new way to shop online!</p>  
<a href="productPage.html" class="button">Discover More &#8634;</a>
```

- Alternate text was provided for all images, for individuals who can not see them.

```

```

- Label elements added to all forms, for screen-readers to know which prompts are associated with which fields.

```
<label for="name">Name</label>  
<input id="name" name="name" type="text" placeholder="Enter Name">
```

Accessibility Test

The following apply to the entire page:

Powered by WebAIM

Styles: OFF ON

Summary

Category	Count
Errors	0
Contrast Errors	0
Alerts	3
Features	13
Structural Elements	18
ARIA	0

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

View details

buyanything website logo

Shop Gallery Listings About Us

Search button menu button

Buyanything

A revolutionary new way to shop online!

Discover More

</> Code

The following apply to the entire page:

Powered by WebAIM

Styles: OFF ON

Summary

Category	Count
Errors	0
Contrast Errors	0
Alerts	2
Features	13
Structural Elements	15
ARIA	0

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

View details

buyanything website logo

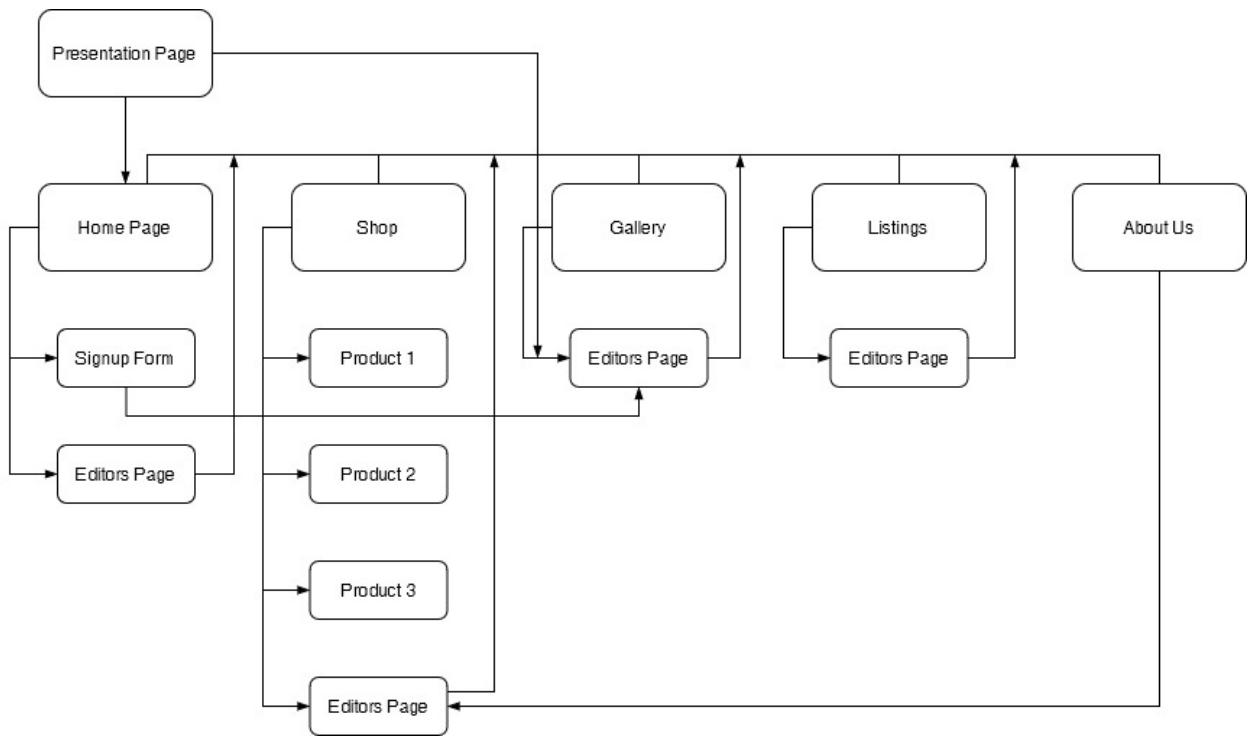
Shop Gallery Listings About Us

A+ A- reset

h2 SHOP PRODUCTS

Cart icon: 0

Site Diagram



Self-Reflection

Puspasothy Seyon-Student 1

Our team did a fantastic job in bringing out our website to life. Even though we stumbled upon several barriers in reaching our goals, including working with 13+ hour nation wide power cut, we fulfilled the requirements of this coursework.

One other problem we faced was how to keep all our work in sync and slowly build up in a very organized manner. We used GitHub to integrate our codes together while having a meeting on google meet. In time the web-site shaped up to its final form.

On a personal note, this project helped us improve our skills. We were able to work as a team and get a small glimpse into the world of real Web developers.

M. Mishaal Rifath - Student 2

This project has been a great learning experience for myself and the team. We were able to complete our assigned tasks while staying coordinated thanks to the use of online tools such as GitHub, to organize our source code and GDocs which made it possible for us to work on the report simultaneously.

Despite all of us facing several difficulties out of our control such as several hour long power cuts, the team was able to complete all our tasks on time. If I had to do this project again there is no other group I'd rather work with.

Personally, working on this has improved my skills and given me a greater understanding of HTML, CSS and JS, giving me confidence for any future projects where I might require the use of any one of the languages.

Yohan Abhishek -Student 3

We were given this assignment about a month ago, and it has been an unfathomable experience, especially the self-learning sections. Because we began working on the coursework as a group as soon as possible, the tension was low, and we were able to complete our tasks and be satisfied with the webpage we produced.

Working as a group was a fantastic experience that gave me a taste of how projects are completed in the industry. We stored and edited our files on GitHub because it is much more convenient. We collaborated on the report using Google Drive as a team.

Because the variables update when the website refreshes, it would have been slightly more convenient if we had stored our products and amounts in databases. To summarize, despite the fact that we had to overcome many obstacles, we were able to complete our work on time and learn a lot regarding web development.

Inuka Rodrigo - Student 4

Working with all my other 3 members helps me to get close to each other and work in an effective manner. Power cut and internet connection issues are the major problems we faced while completing the project but we managed to control such problems. When working on the assignment, I faced problems especially when writing JavaScript codes but after referring to online resources, I was able to solve those issues. This assignment gives me a good understanding of HTML, CSS, JavaScript and XML. With the use of GitHub, we were able to share our codes and other files in an efficient manner.

References

W3Schools (2019). *W3Schools Online Web Tutorials*. [online] W3schools.com. Available at: <https://www.w3schools.com/>.

Youtube (2022). *YouTube*. YouTube. Available at: <https://www.youtube.com/>