Informatics Institute of Technology
Department of Computing

Bsc(Hons) Artificial Intelligence and Data Science

Module: CM1605 Web Technology

Module Coordinator: Ms. Janani Harischandra

Group Coursework Report

Tutorial Group : C

Coursework Group No : Group C1

Student Details :

Mindiya De Zoysa -2330949- Student Number 1

Ethan Perera -2331419- Student Number 2

Chirath Setunge -2330910- Student Number 3

Chanul Vitharana -2330948- Student Number 4

Table of Contents

1.0 Introduction

Our team created the front end of "A Healthy Place.com" an interactive AI-powered web application, in response to the increasing need for affordable and effective healthcare solutions.

A Healthy Place.com provides a number of tools that enable people to take control of their health:

Online Store: Through the application, users can easily buy health-related products.

Interactive Quiz: By answering questions on health, users can win discounts on medical supplies, which lowers the cost of self-care.

Feedback System: Users can offer their thoughts about the application, which helps to make it better every time.

The user interface (UI) of the application consists of the following pages:

- Sign Up Page
- Home Page
- Quiz Page
- Product Page
- Listing Page
- Gallery Page

The team members involved in developing this project are:

- Mindiya De Zoysa -2330949- Student Number 1
- Ethan Perera -2331419- Student Number 2
- Chirath Setunge -2330910- Student Number 3
- Chanul Vitharana -2330948- Student Number 4

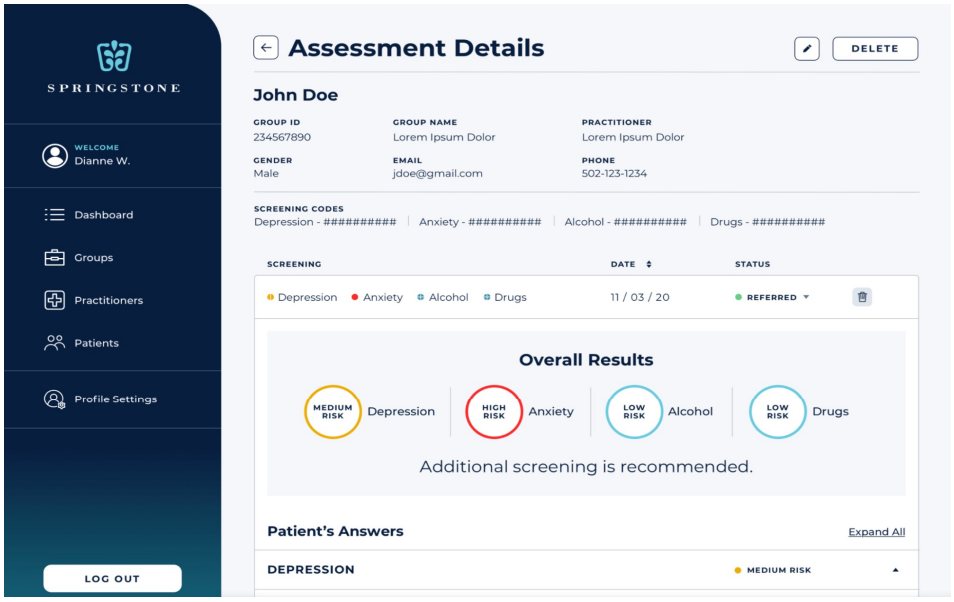
This report describes the front-end design's technical features in detail and offers a critical analysis of the UI's design decisions, including colour and font, and how they affect the user's experience.

2.0 Research On Existing Systems

There are several popular health recommendation web applications available in the market, such as Medisafe Pill Minder and Trackers, Eyecare Live, Teladoc, Springstone, MySugr and MyChart. These applications have been analyzed based on user preferences to identify their limitations and positive points.

Springstone

A leading mental healthcare provider offers a private and secure application for assessing and treating youth mental health. It is designed to improve the mental health screening and referral process for employers, employees, and primary care physicians. A frictionless user experience reduces user stress to improve assessment accuracy. A progressive web app (PWA) developed by DBS Interactive, Springstone's app reaches a broader audience of users because it is device-agnostic, accessible via either mobile or desktop devices with a browser. The application offers customizable administrative permissions for individual users, groups, and care coordinators.



Teladoc

The Teladoc app connects patients to doctors, therapists, dietitians, nurses, and coaches 24/7 through phone or video visits for many urgent needs. It also offers a variety of self-guided programs that can be accessed at any time.

During registration, users can choose their preferred language. The company also offers American Sign Language as an option. When users connect with a healthcare provider, a video service creates a three-way connection with an interpreter without extra charge. The designers built the most important information into one screen. Clicking “Get Care” lets users scroll to see the different care options and links to common questions.

The company offers transparency about the data used and linked to users. Doctors can also prescribe medication through the app.

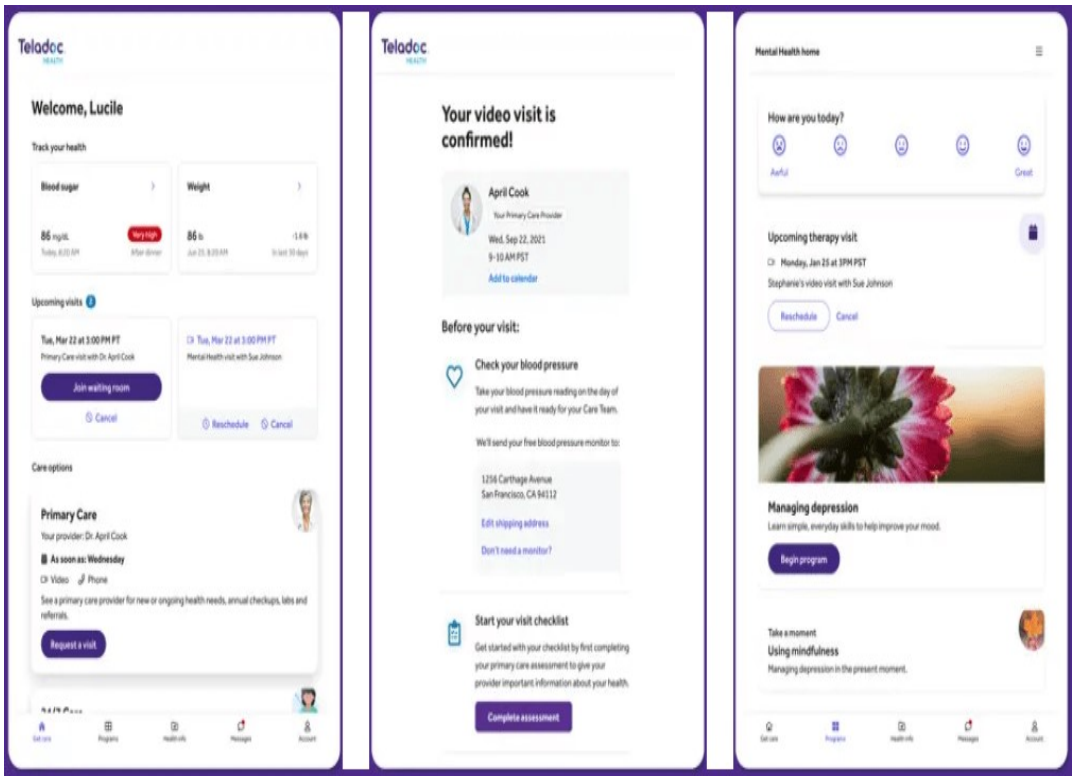


Figure 1: Site similar to our health page

3.0 Technical Discussion

3.1 Student 1 (Mindiya De Zoysa)

3.1.1 Presentation Page

So, the content regarding the presentation page will show the name of the group members for 5 seconds and then it will be redirected to the main page.

HTML Code

```
<body>
    <div class="welcome-container">
        <div class="welcome-message">
            <p>Group-C(1).</p>
            <p>Mindiya De Zoysa</p>
            <p>Ethan Perera</p>
            <p>Chirath Setunge</p>
            <p>Chanul Vitharana</p>
        </div>
    </div>

    <script>
        window.onload = function() {
            setTimeout(function() {
                window.location.href = "../../Student 2/home_page.html";

            }, 5000);
        };
    </script>
</body>
```

This HTML code represents a simple web page with a welcome message displayed inside a container. The welcome message includes a list of names under the "Group-C(1)" heading. After the page loads, a JavaScript function is executed using the `window.onload` event handler. This function waits for 5 seconds (5000 milliseconds) using `setTimeout`, and then it redirects the user to home_page.html.

3.1.2 Sign-Up Form

So the content is regarding the sign-up page in this the user must enter an email address which should be a valid email like "hallo@gmail.com" and then both the passwords should be matched up(the original password and also the reenter password). Then the user should click sign up then he will go into asking personal details like age, telephone number those things but in this the personal details and user information section is not a must to fill only the signup part is a must to fill. so then when the user completes the process he can click on complete then he will be redirected to the main page.

JavaScript Code

```
<br>
    <script>
        var currentTime = new Date().getHours();
        var greeting = "";

        if (currentTime < 12) {
                greeting = "Good Morning";
        } else if (currentTime < 15) {
                greeting = "Good Afternoon";
        } else {
                greeting = "Good Evening";
        }

        document.write("<h2>" + greeting + ", Welcome to our Signup Page.</h2>");
    </script>
```

Over here I am using java script to greet the user who is entering the page. If it is before 12 pm then it will display Good Morning to the user like wise.

```
<script>
    const passwordFields = document.querySelectorAll('input[type="password"]');
    const showPasswordCheckboxes = document.querySelectorAll('.showPassword');

    showPasswordCheckboxes.forEach((checkbox, index) => {
        checkbox.addEventListener("change", function() {
            if (this.checked) {
                passwordFields[index].type = "text";
            } else {
                passwordFields[index].type = "password";
            }
        });
    });
</script>
```

Over here I am using a small check box where the user can check the input password which will be convert to text and then if he unchecked it again it will go to password.

```
<script>
    let currentStep = 1;

    function nextStep(step) {
      document.getElementById(`step${step}`).classList.remove('active');
      currentStep = step + 1;
      document.getElementById(`step${currentStep}`).classList.add('active');
    }

    function previousStep(step) {
      document.getElementById(`step${step}`).classList.remove('active');
      currentStep = step - 1;
      document.getElementById(`step${currentStep}`).classList.add('active');
    }

    function validateRegistration() {
        const username = document.getElementById('username').value.trim();
        const email = document.getElementById('email').value.trim();
        const password = document.getElementById('password').value.trim();
        const confirm_password = document.getElementById('confirm_password').value.trim();

        // Regular expression to match email format
        const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
```

```
    if (!username || !email || !password || !confirm_password) {
        alert('Please fill in all registration fields');
    } else if (!emailPattern.test(email)) {
        alert('Please enter a valid email address');
    } else if (password !== confirm_password){
        alert('Both the passwords should be same!');
    } else if (password.length < 8){
        alert('Password must be at least 8 characters long!');
    } else {
        alert(`Dear ${username}, thank you for signing up with us. The recommended results will be shown in a while.`);
        nextStep(1);
    }
}

function redirectToMainPage() {

    window.location.href = "../../Student 2/home_page.html";
}
</script>
```

Over here as there are 3 different forms created by 3 div ids then the page should be able to move to all the three forms so here I am using the function next step to go back and forth and also if the user hasn't entered the valid details in the sign-up it will say to enter the details and also the passwords should be same and the length of the password should be more than 8. If all the required fields are entered properly and also the passwords are okay then it will show a pop up message saying that, from the entered username if will displaya small thank you message.

3.1.3 Gallery Page

So, the content regarding the gallery page in this the there are 5 images. So as the user clicks it will be thumbnailed and also the image will be large and it will show a message next to the thumbnailing image. However, the 5 images will still be shown in the page under the thumbnail image so when the user clicks on the next image then it will go to the next page.

JavaScript Code

```
<script>
    $(document).ready(function (){
        $('.container.grid-col a').click(function(e){

            let href = e.target.parentElement.getAttribute('href');

            $('.tabcontent').each(function(index,element){
                if(`#${element.id}`!=href){
                    $(element).removeClass('active');
                }
            })

            $('.container.blog-content span').click(function(){
                $(href).removeClass('active');
            })

            $(href).toggleClass('active');
            $('.tabcontent').css({
                height:"30rem"
            });

        })
    })
</script>
```

3.2 Student 2 (Ethan Perera)

Student 2 was tasked with developing a web page for the home page, the quiz and a form to give the user the ability to write their feedback over how their experience was with the quiz page. In terms of approach take to implement a solution to these problems, a prompt box/text area was referred to for the quiz as to where it was validated using java script. It had ensured that the user had entered the exact value required to receive points from the quiz to use on the product page (this would conclusively deduct the overall cost of a purchase the user had made on that page). A popup was referred to output the users results Regard the following code snippet to see how it was implemented:

```
// Quiz Java script

function quiz_submit(){
```

```
    var q1 = document.getElementById("first_q");
    var q2 = document.getElementById("second_q").value;
    var q3 = document.getElementById("third_q").value;
    var q4 = document.getElementById("fourth_q").value;
    var q5 = document.getElementById("fifth_q").value;

    let states = [false,false,false,false,false];

    switch (q1.valueAsNumber){
        case 37:
            score+=2;
            states[0]=true;
            break;

        case "":
            break

        default:
            score--;
    }

    switch (q2.trim().toLowerCase()){
        case "vitamin a":
            score+=2;
            states[1]=true;
            break;

        case "":
            break

        default:
            score--;
    }

    switch (q3.trim().toLowerCase()){
        case "carry oxygen":
            score+=2;
            states[2]=true;
            break;

        case "":
            break

        default:
            score--;
    }

    switch (q4.trim().toLowerCase()){
```

```
        case "skin":
            score+=2;
            states[3]=true;
            break;

        case "":
            break

        default:
            score--;
    }

    switch (q5.trim().toLowerCase()){
        case "filter waste from the blood":
            score+=2;
            states[4]=true;
            break;

        case "":
            break

        default:
            score--;
    }

    sessionStorage.setItem('score',score);

    if (score<=0){
        alert("You have scored: 0 points");
    } else{
        alert("You have scored: "+score);
    }

    var incorrect_string = "";

    for (let i = 0; i <= states.length; i++) {
        switch(i){
            case 0:
                if(states[0]!=true){
                    incorrect_string+="You got question 1 wrong!<br>";
                }
                break

            case 1:
                if(states[1]!=true){
                    incorrect_string+="You got question 2 wrong!<br>";
                }
                break
```

```
            case 2:
                if(states[2]!=true){
                    incorrect_string+="You got question 3 wrong!<br>";
                }
                break

            case 3:
                if(states[3]!=true){
                    incorrect_string+="You got question 4 wrong!<br>";
                }
                break

            case 4:
                if(states[4]!=true){
                    incorrect_string+="You got question 5 wrong!<br>";
                }
                break
        }
    }
    if (incorrect_string!="")
        document.getElementById("popupText").innerHTML = incorrect_string;
    else
        document.getElementById("popupText").innerHTML = "you got all the
questions right!"
    var modal = document.getElementById("myModal");
    modal.style.display = "block";
}

// Show the popup with user details
function showPopup() {
    var name = document.getElementById("name").value;
    var email = document.getElementById("email").value;
    var age = document.getElementById("age").value;
    var contact = document.getElementById("contact").value;

    if (name.trim() == "" || email.trim() == "" || age.trim() == "" ||
contact.trim() == ""){
        var popupText = "Ensure that you fill all the fields"
    }else {
        var popupText = "Name: " + name + "<br>Email: " + email + "<br>Age:
" + age + "<br>Contact: " + contact;
    }
    document.getElementById("popupText").innerHTML = popupText;
    var modal = document.getElementById("myModal");
    modal.style.display = "block";
}
```

```
// Close the popup
function closePopup() {
    var modal = document.getElementById("myModal");
    modal.style.display = "none";

    document.body.style.overflow = ""; // Enable scrolling
}
```

The script is simple in retaining the users answer by equating it to an exact value, however in terms of maximising its simplicity, the user's input was converted to a lower case such that it may be easier to receive the correct answer. In terms of how the points were then stored, the "set item" function was referred to.

3.3 Student 3 (Chirath Sethunga)

3.3.1 Product page

The purpose of the product page is to make it easier for consumers to browse and make purchases. This is accomplished by clearly dividing the parts devoted to product exploration from those that deal with shopping cart handling.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>A Healthy Place/Products</title>
  <link rel="stylesheet" href="style_sheets/Product page.css">
  <nav>
    <h1 class="logoheader"><a href="home_page.html">A Healthy Place</a></h1>
    <ul>
      <li><img alt="Health.com_logo" src="style_sheets/images/Health_Logo.png" alt="logo"
class="logo"></li>
      <li class="dropdown">
        <a href="home_page.html" class="dropbtn">Home</a>
        <div class="dropdown-content">
          <a href="Gallery_pg.html">Gallery</a>
          <a href="Aboutus.html">About us</a>
        </div>
      </li>
      <li><a href="Product page.html">Products</a></li>
      <li><a href="quiz.html">Quiz</a></li>
      <li><a href="Feedbacklisting.html">Feedbacks</a></li>
      <li><a href="Signup_pg.html">Sign in</a></li>
    </ul>
  </nav>
</head>
```

```
<body>
<div class="main-container">
  <div class="left">
    <div class="top">
      <h2 class="title">PRODUCTS</h2>
      <div class="scroll-buttons">
        <button class="scroll-btn left-btn"><</button>
        <button class="scroll-btn right-btn">></button>
      </div>
    </div>
    <div class="middle">
      <div class="item">
        <img src="style_sheets/images/Oximeter.png" alt="product1" class="PImage" id="p1">
        <h2 class="name">O<sub>2</sub> METER</h2>
        <div class="price">£60</div>
        <div class="butns">
          <div class="qty">QTY</div>
          <input type="number" aria-label="Quantity" class="counter" value="1" min="1">
          <button class="addCart">Add to Cart</button>
        </div>
      </div>
      <div class="item">
        <img src="style_sheets/images/thermo.png" alt="product2" class="PImage" id="p2">
        <h2 class="name">THERMOMETER</h2>
        <div class="price">£30</div>
        <div class="butns">
          <div class="qty">QTY</div>
          <input type="number" aria-label="Quantity" class="counter" value="1" min="1">
          <button class="addCart">Add to Cart</button>
        </div>
      </div>
      <div class="item">
        <img src="style_sheets/images/bandage.png" alt="product3" class="PImage" id="p3">
        <h2 class="name">3M BANDAGE</h2>
        <div class="price">£15</div>
        <div class="butns">
          <div class="qty">QTY</div>
          <input type="number" aria-label="Quantity" class="counter" value="1" min="1">
          <button class="addCart">Add to Cart</button>
        </div>
      </div>
      <div id="last" class="item">
        <img src="style_sheets/images/omron- blood pressure monitor.png" alt="product4"
class="PImage">
        <h2 class="name">PRESSURE MONITOR</h2>
        <div class="price">£100</div>
        <div class="butns">
          <div class="qty">QTY</div>
          <input type="number" aria-label="Quantity" class="counter" value="1" min="1">
          <button class="addCart">Add to Cart</button>
        </div>
```

```
        </div>
      </div>
      <div class="bottom">
        <form class="personal-details">
          <fieldset>
            <legend>Personal Details</legend>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" placeholder="Chirath Setunge">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email"
placeholder="chirath.20230374@iit.ac.lk">
            <label for="tel">Tel No:</label>
            <input type="tel" id="tel" name="tel" placeholder="071-123-4567">
            <label for="address">Address:</label>
            <textarea name="address" id="address" rows="3" cols="30" placeholder="No.123,
ABCD Rd, XYZ"></textarea>
          </fieldset>
        </form>
        <form class="card-details">
          <fieldset>
            <legend>Card Details</legend>
            <label for="card-number">Card Number:</label>
            <input type="text" id="card-number" name="card-number" placeholder="1234-5678-
9012-1234">
            <label for="expiry-month">Expiry month:</label>
            <input type="text" id="expiry-month" name="expiry-month" placeholder="12">
            <label for="expiry-year">Expiry year:</label>
            <input type="text" id="expiry-year" name="expiry-year" placeholder="2029">
            <label for="cvv">CVV:</label>
            <input type="text" id="cvv" name="cvv" placeholder="123">
          </fieldset>
        </form>
      </div>
    </div>
    <div class="right">
      <div class="RTop">
        <h2 class="shopping-cart">Shopping Cart</h2>
      </div>
      <div class="RMiddle"></div>
      <div class="discountBox">
        <label for="points">Points:</label>
        <input type="number" id="points" class="points-input" min="0" value="0">
        <p class="point-rule"> 1 point = 1% disocunt</p>
      </div>
      <div class="totalBox">
        <div class="subtotal">Subtotal: &<span class="subtotal-amount">0.00</span></div>
        <div class="discount-price">Discount: &<span class="discount-
amount">0.00</span></div>
        <div class="grand-total">Grand Total: &<span class="grand-total-
amount">0.00</span></div>
      </div>
```

```
        <div class="RBottom">
            <button class="clearOrder">Clear Order</button>
            <button class="placeOrder" onclick="togglePopup()">Place Order</button>
        </div>
    </div>
</div>
<div class="popup" id="orderPopup">
    <div class="popuptext" id="popupContent">
        <h3>Order Details</h3>
        <p id="popupMessage"></p>
        <button class="okayButton" id="okayButton">Okay</button>
    </div>
</div>

<footer class="footer"> <!--Footer has designed by Student 4 (Chanul)-->
    <div class="col-1">
        <h3>Links</h3>
        <a href="Gallery_pg.html">Gallery</a>
        <a href="form.html">Rate Us</a>
        <a href="Product page.html">Products</a>
        <a href="Aboutus.html">About us</a>
    </div>
    <div class="col-2">
        <h3>Newsletter</h3>
        <form class="footer_form" id="newsletterForm">
            <input type="text" aria-label="newsletter-name" placeholder="Enter your name here"
name="name" id="name">
            <br>
            <input type="text" aria-label="newsletter-email" placeholder="Enter your email address
here" name="email" id="email">
            <br>
            <button type="submit">Subscribe now</button>
        </form>
    </div>
    <div class="col-3">
        <h3>Contacts</h3>
        <p>&copy; This page was made by Chirath Setunge, <br> visit his <a href="Student_3.html"
target="_blank"><b>editors page</b></a> to know more</p>
    </div>
</footer>
<script src="java_scripts/Product page.js"></script>
</body>
</html>
```

Product Listings: The space on the left side of the page is where products are shown. Users can get information about the prices and available items here. Other important features in this

section, such as the "Add to Cart" buttons that let customers easily add desired items to their virtual shopping basket, are also essential for starting the purchase process.

Shopping Cart: The user's shopping cart is managed on the right side of the page. This part offers a clear view of the selected products along with the ability to eliminate unneeded items or change quantity. Additionally, it has discount application features that let customers lower their entire order amount before checking out. To keep users updated on the entire cost of their choices, this section also shows the grand total and the current order subtotal.

User Information: A specific part of the product page allows users to add personal information related to their order. to guarantee the efficient delivery of the ordered goods.

3.3.2 Cart Management:

• **Adding Products:** Items are copied and added to the cart area when "Add to Cart" buttons are clicked. Quantity inputs and minus buttons are also included for modifying amounts.
• **Removing Products:** Users can eliminate products when the quantity hits zero by using the minus buttons to reduce the quantity.

• **Emptying Cart:** Pressing the "Clear Cart" button completely empty

```
document.addEventListener('DOMContentLoaded', function() {
    const addToCartButtons = document.querySelectorAll('.addCart');
    // Event listener for 'Add to Cart' buttons
    addToCartButtons.forEach(button => {
        button.addEventListener('click', function() {
            // Clone selected item to create a cart item
            const item = this.closest('.item');
            const cartItem = item.cloneNode(true);
            const quantityInput = cartItem.querySelector('.counter');
            const minusButton = document.createElement('button');
            minusButton.textContent = '-';
            minusButton.classList.add('minus');
            // Event listener for minus button to adjust quantity or remove item from cart
            minusButton.addEventListener('click', function() {
                if (parseInt(quantityInput.value) >= 1) {
                    quantityInput.value = parseInt(quantityInput.value) - 1;
                }
                if (parseInt(quantityInput.value) === 0) {
                    cartItem.remove(); // Remove the item from the cart if quantity is 0
                }
```

```
        });

        cartItem.classList.add('cart-item');

        // Remove product image from the cloned cart item
        cartItem.querySelector('.PImage').remove();

        // Replace the "Add to Cart" button with minus button and quantity input
        cartItem.querySelector('.butns').replaceWith(minusButton, quantityInput);
        // Append the cart item to the cart container
        const cartItemsContainer = document.querySelector('.RMiddle');
        cartItemsContainer.appendChild(cartItem);
    });
  });
});
```

3.3.3 Calculations

- The code is designed to calculate and display subtotal, discount, and grand total on a continuous basis. It does this by utilizing points to account for item pricing, quantities, and any applicable discounts.
- **Point-Based Discounts:** A discount of 1% is represented by one point, and discounts are determined as a proportion of the sum.

```
// Function to calculate subtotal, discount amount, and grand total
document.addEventListener('DOMContentLoaded', function() {
    // Function to calculate subtotal, discount amount, and grand total
    function calculateTotal() {
        // Initialize subtotal
        let subtotal = 0;
        const cartItems = document.querySelectorAll('.cart-item');

        // Calculate subtotal by summing up the prices of all items in the cart
        cartItems.forEach(item => {
            const price = parseFloat(item.querySelector('.price').textContent.replace('₹',
'));
            const quantity = parseInt(item.querySelector('.counter').value);
            subtotal += price * quantity;
        });

        // Display subtotal
        const subtotalAmount = document.querySelector('.subtotal-amount');
        subtotalAmount.textContent = subtotal.toFixed(2);

        // Calculate discount amount (1 point = 1% discount)
        const points = parseInt(document.getElementById('points').value);
```

```
        const discountPercentage = points;
        const discountAmount = (subtotal * discountPercentage) / 100;

        // Display discount amount
        const discountAmountElement = document.querySelector('.discount-amount');
        discountAmountElement.textContent = discountAmount.toFixed(2);

        // Calculate grand total based on points
        let grandTotalAmount;
        if (points === 0) {
            grandTotalAmount = subtotal;
        } else {
            grandTotalAmount = subtotal - discountAmount;
        }

        // Display grand total
        const grandTotalAmountElement = document.querySelector('.grand-total-
amount');
        grandTotalAmountElement.textContent = grandTotalAmount.toFixed(2);
    }

    // Call calculateTotal function when page loads
    calculateTotal();

    // Call calculateTotal function every second to update the grand total
    setInterval(calculateTotal, 100);

    // Call calculateTotal function when add to cart
    const addToCartButtons = document.querySelectorAll('.addCart');
    addToCartButtons.forEach(button => {
        button.addEventListener('click', calculateTotal);
    });

    // Call calculateTotal function when points input changes
    const pointsInput = document.getElementById('points');
    pointsInput.addEventListener('change', calculateTotal);
});
```

3.3.4 Order Placement

- **Order Form Validation:** This code verifies that user-provided information, including name, email, phone number, address, and card information, is formatted correctly.
- **Order Summary Popup**: The order details, including goods, quantities, prices, subtotals, discounts, grand totals, and delivery addresses, are displayed in a popup window upon successful validation.

```
document.addEventListener('DOMContentLoaded', function() {
    // Event listener to handle placing an order
    const placeOrderButton = document.querySelector('.placeOrder');
    const popup = document.getElementById('orderPopup');
    const mainContent = document.querySelector('.main-container');

    placeOrderButton.addEventListener('click', function() {
        // Get user inputs
        const nameInput = document.getElementById('name').value;
        const emailInput = document.getElementById('email').value;
        const telInput = document.getElementById('tel').value;
        const addressInput = document.getElementById('address').value;
        const cardNumberInput = document.getElementById('card-number').value;
        const expiryMonthInput = document.getElementById('expiry-month').value;
        const expiryYearInput = document.getElementById('expiry-year').value;
        const cvvInput = document.getElementById('cvv').value;
        // Regular expressions for input validation
        const yearRegex = /^\d{4}$/; // Matches 4 digitts only
        const monthRegex = /^(0[1-9]|1[0-2])$/; // Matches two digits between 01 and 12
        const telRegex = /^\d{10}$/; // Matches 10 digits only
        const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/; // Matches basic email format

        // Check if all required fields are filled and meet validation criteria
        if (!nameInput || !emailInput || !telInput || !addressInput || !cardNumberInput ||
!expiryMonthInput || !expiryYearInput || !cvvInput) {
            // If any required field is empty, display a prompt box
            alert('Please fill out all required fields.');
        } else if (!telRegex.test(telInput)) {
            // If telephone number format is incorrect, display an alert
            alert('Please enter a valid telephone number.');
        } else if (!emailRegex.test(emailInput)) {
            // If email format is incorrect, display an alert
            alert('Please enter a valid email address.');
        } else if (!yearRegex.test(expiryYearInput)) {
            // If year format is incorrect, display an alert
            alert('Please enter a valid year format.');
        } else if (!monthRegex.test(expiryMonthInput)) {
            // If month format is incorrect, display an alert
            alert('Please enter a valid month.');
        } else {
            // All required fields are filled, proceed with displaying the popup
            mainContent.classList.add('blur');
            const message = generatePopupMessage(nameInput, addressInput);
            document.getElementById('popupMessage').innerHTML = message;
            popup.classList.add('show');
        }
    });

    // Close the popup when the user clicks the okay button
    const okayButton = document.getElementById('okayButton');
    okayButton.addEventListener('click', function() {
```

```
            mainContent.classList.remove('blur');
            popup.classList.remove('show');
            location.reload();
        });
    });
    // Function to generate message for the order popup
    function generatePopupMessage(name, address) {
        let message = `Dear ${name},<br><br>You have ordered:<br>`;

        // Iterate over each item in the cart
        const cartItems = document.querySelectorAll('.cart-item');
        cartItems.forEach(item => {
            const itemName = item.querySelector('.name').textContent;
            const quantity = parseInt(item.querySelector('.counter').value);
            const price = parseFloat(item.querySelector('.price').textContent.replace('£', ''));
            const itemCost = quantity * price;
            message += `${quantity}x  ${itemName} at a cost of £${itemCost.toFixed(2)}<br>`;
        });

        // Calculate total bill
        let totalBill = 0;
        cartItems.forEach(item => {
            const price = parseFloat(item.querySelector('.price').textContent.replace('£', ''));
            const quantity = parseInt(item.querySelector('.counter').value);
            totalBill += price * quantity;
        });

        // Add total bill to the message
        message += `<br>Your Subtotal bill is £${totalBill.toFixed(2)}.<br>`;

        const grandTotal = document.querySelector('.grand-total-amount').textContent;

        message += `Your Grand Total is £${grandTotal}<br>`;

        // Add address to the message
        message += `<br>Your order will be delivered to: ${address}`;

        return message;
    }

    var score = sessionStorage.getItem('score');

        if (score !== null) {
            // Do something with the score, e.g., display it
            console.log('Score:', score);

            // Update the max attribute of the points input field
            var pointsInput = document.getElementById('points');
            pointsInput.setAttribute('max', score);
        }
```

3.3.5 About Us

```
<body>
  <div class="main-container">
    <div class="container">
      <div class="top">
        <h2 class="title">About Us</h2>
        <div class="zoom-buttons">
          <button class="zoom-btn decrease">-</button>
          <button class="zoom-btn increase">+</button>
        </div>
      </div>
      <div class="aboutUsContainer">
        <div class="student" data-id="stu1">
          <img src="style_sheets/images/Student_1.jpeg" alt="stu1" >
          <h2>Student 1</h2>
        </div>
        <div class="student" data-id="stu2">
          <img src="style_sheets/images/Student_2.jpeg" alt="stu2">
          <h2>Student 2</h2>
        </div>
        <div class="student" data-id="stu3">
          <img src="style_sheets/images/Student_3.jpeg" alt="stu3">
          <h2>Student 3</h2>
        </div>
        <div class="student" data-id="stu4">
          <img src="style_sheets/images/Student_4.jpeg" alt="stu4">
          <h2>Student 4</h2>
        </div>
      </div>
    </div>
    <div class="right">
      <div class="student-details">
        <img src="style_sheets/images/AboutUs.png" alt="Student Image" id="student-image">
        <h2 id="student-name">About Us</h2>
        <p id="student-id"></p>
        <p id="student-role"></p>
        <p id="student-email"></p>
      </div>
    </div>
  </div>
  <footer class="footer"> <!--Footer has designed by Student 4 (Chanul)-->
    <div class="col-1">
      <h3>Links</h3>
      <a href="Gallery_pg.html">Gallery</a>
      <a href="form.html">Rate Us</a>
      <a href="Product page.html">Products</a>
      <a href="Aboutus.html">About us</a>
```

```
      </div>
      <div class="col-2">
        <h3>Newsletter</h3>
        <form class="footer_form" id="newsletterForm">
          <input type="text" aria-label="newsletter-name" placeholder="Enter your name here"
name="name" id="name">
          <br>
          <input type="text" aria-label="newsletter-email" placeholder="Enter your email address
here" name="email" id="email">
          <br>
          <button type="submit">Subscribe now</button>
        </form>
      </div>
      <div class="col-3">
        <h3>Contacts</h3>
        <p>&copy; This page was made by Chirath Setunge, <br> visit his <a
href="Student_3.html" target="_blank">editors page</a> to know more</p>
      </div>
    </footer>
<script src="java_scripts/AboutUs.js"></script>
</body>
```

This page presents the team that created the website for "A Healthy Place.com"

Team Member Details: An object named studentDetails contains data about every team member. Information such as name, ID, role, email, and picture path are contained in this object.

```
const studentDetails = {
    stu1: {
        name: "Mindiya De Zoysa",
        id: "S1",
        role: "1",
        email: "m.agampodi@rgu.ac.uk",
        image: "style_sheets/images/Student_1.jpeg"
    },
    stu2: {
        name: "Ethan Christoff Perera ",
        id: "2331419",
        role: "2",
        email: "e.modarage@rgu.ac.uk",
        image: "style_sheets/images/Student_2.jpeg"
    },
    stu3: {
        name: "Chirath Shamika Setunge",
        id: "2330910",
        role: "3",
        email: "c.setunge-mudalige-don@rgu.ac.uk",
        image: "style_sheets/images/Student_3.jpeg"
    },
    stu4: {
```

```
        name: "Chanul Vitharana",
        id: "2330948",
        role: "4",
        email: "c.vitharana@rgu.ac.uk",
        image: "style_sheets/images/Student_4.jpeg"
    },


};
```

- **Hover Effect:** The script collects the relevant details from studentDetails and displays them in the right part of the page when a user hovers over a team member's card . The member's picture, name, ID, role, and email are all included in this.

```
// Attach event listeners to each student element
    const students = document.querySelectorAll('.student');
    students.forEach(student => {
        student.addEventListener('mouseenter', function() {
            const studentId = this.getAttribute('data-id');
            updateStudentDetails(studentId);
        });
        student.addEventListener('mouseleave', function() {
            resetRightContainer();
        });
```

- **Reset Functionality:** The script returns the right container to its initial state when the user takes the mouse away from a team member card. It deletes the specific details and shows a substitute image.

```
function resetRightContainer() {
        document.querySelector('.right').classList.remove('active');
        document.getElementById('student-image').src =
"style_sheets/images/AboutUs.png";
        document.getElementById('student-name').textContent = "";
        document.getElementById('student-id').textContent = "";
        document.getElementById('student-role').textContent = "";
        document.getElementById('student-email').textContent ="";
    }
```

- **Zoom Buttons:** The zoom buttons (increase and decrease) are managed by the script. The team member information section's text size can be increased by clicking the increase button, up to

a certain amount. On the other hand, the decrease button lowers the text size while preventing it from falling below a predetermined minimum.

```
// Zoom-in and zoom-out functionality
    const increaseBtn = document.querySelector('.zoom-btn.increase');
    const decreaseBtn = document.querySelector('.zoom-btn.decrease');
    const aboutUsContainer = document.querySelector('.aboutUsContainer');
    let currentFontSize = 16; // Default font size

    // Function to increase font size
    function increaseFontSize() {
        currentFontSize += 5;
        aboutUsContainer.style.fontSize = currentFontSize + 'px';
    }

    // Function to decrease font size
    function decreaseFontSize() {
        if (currentFontSize > 5) { // Ensure font size doesn't go below 5px
            currentFontSize -= 5;
            aboutUsContainer.style.fontSize = currentFontSize + 'px';
        }
    }

    // Attached event listeners to the plus and minus buttons
    increaseBtn.addEventListener('click', increaseFontSize);
    decreaseBtn.addEventListener('click', decreaseFontSize);
```

3.4 Student 4 (Chanul Vitharana)

Student 4's role was to create the Feedback listing page which contains the feedback for the items from the products page and the sign-up form for the personalized Newsletter page.

3.4.1 Feedback listing page

There are drop down menus in the feedback listing page where user can use to change the background colors and the text colors of the listing page. The listing page has 4 background colors and 4 text colors to choose from.

```
function changeBackgroundColor(color) {
    document.body.style.backgroundColor = color;
}

function changeTextColor(color) {
    document.body.style.color = color;
}
```

Here there are two functions as “changeBackgroundcolor” and “changetextcolor” to change the text and the background colors in the feedback listing page. When this function is called in the html file relevant to the feedback listing page using the click event listeners, when the user clicks from the color in the drop-down menu it called the relevant function and changes the color accordingly. These drop-down menus are added to the navigation button in the listing page.

```
document.addEventListener("DOMContentLoaded", function() {
loadFeedbacks();
});

function loadFeedbacks() {
const xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState === 4 && this.status === 200) {
   displayFeedbacks(this);
}
};
xhttp.open("GET", "feedbacks.xml", true);
xhttp.send();
}

function displayFeedbacks(xml) {
const xmlDoc = xml.responseXML;
const feedbackList = xmlDoc.getElementsByTagName("feedback");
let htmlContent = "";
for (let i = 0; i < feedbackList.length; i++) {
const name = feedbackList[i].getElementsByTagName("name")[0].childNodes[0].nodeValue;
const comment = feedbackList[i].getElementsByTagName("comment")[0].childNodes[0].nodeValue;
const rating = feedbackList[i].getElementsByTagName("rating")[0].childNodes[0].nodeValue;
htmlContent += `<div class="feedback-item">
                <h3>${name}</h3>
```

```
            <p>${comment}</p>
            <p><strong>Rating:</strong> <span class="stars">${"&#9733;".repeat(rating)}</span></p>
        </div>`;
}

document.querySelector(".container").innerHTML = htmlContent;
}
```

This JavaScript code allows feedback from an XML file to be dynamically retrieved and displayed on a webpage. To make sure the HTML document is completely loaded before running, it watches for the "DOMContentLoaded" event. The process of retrieving feedback data is initiated by the loadFeedbacks() method, which is called once the event is triggered.

An XMLHttpRequest object is created by the loadFeedbacks() function in order to send an asynchronous HTTP (AJAX) request to the server for the "feedbacks.xml" file. When the request's state changes and it is successfully completed (ready State is 4 and status is 200), it provides a callback function that performs responses. The feedback items, which consist of the name of the feedback provider, their comment, and a numerical rating, are extracted from the XML response by the callback, displayFeedbacks().

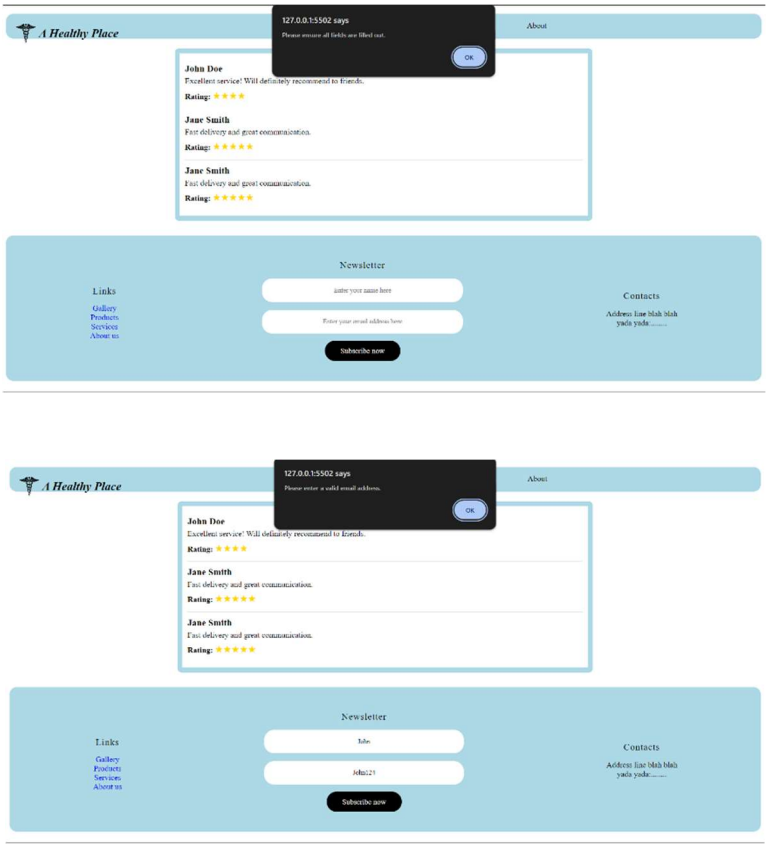
The information is formatted into HTML content, with the rating graphically represented by repeated star symbols (☆), the name in a <h3> tag, and the comment in a <p> tag. After that, this content is added to a webpage container element with the class.container, arranging each feedback item so that consumers may read it easily.

3.4.2 Newsletter form

Moreover, Our website consists of a standard HTML sign up form to register customers for our Hospital webpage. This form will use JavaScript to Validate the user inputs.

```
document.getElementById('newsletterForm').onsubmit = function(event) {
    event.preventDefault(); // Prevent form submission to server for demonstration

    // Get the values from the form
    var name = document.getElementById('name').value;
    var email = document.getElementById('email').value;

    // Check if fields are filled
    if (!name || !email) {
        alert('Please ensure all fields are filled out.');
        return; // Stop further execution
    }


    // Basic email regex for validation
    var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    // Check if email is valid according to regex
    if (!emailRegex.test(email)) {
        alert('Please enter a valid email address.');
        return; // Stop further execution
    }

    // If all checks pass
    alert(`Dear ${name}, you have successfully subscribed to our personalized newsletter.`);
};
```

The above JavaScript Snippet is used to check if the user name and the email filed is filled, if not it will alert the user with a message, Also it will check if the user has entered a proper email using regex, If it's not a valid email website will alert the user to enter the valid email address. Once the user has enters the proper inputs for the required fields, a pop-up message(Success message) will be appeared with the user's name which will be take in as arguments.



The above images will show how the error messages are popped for different invalid user inputs. This will make sure the user enters only the correct inputs so that when regisetering users for the newsletter all the functions can be executed smoothly without getting any errors.

4.0 Discussion of UX/UI principles/Applications/Justifications

4.1 Navigation Techniques

A Healthy Place.com uses a top navigation bar to make key services easily accessible. This navigation bar makes use of the tag for semantic reasons. This element notifies search engines exactly how the webpage is structured and expressly designates the enclosing links as being a part of the main navigation system. Every navigation link is created using the element, which turns into a clickable element when it is presented on the web page. The page the user will be taken to when they click on these components is indicated by the **href** attribute, which gives the URL or path of the destination webpage.

```
<nav>
      <h1 class="logoheader"><a href="home_page.html">A Healthy Place</a></h1>
      <ul>
         <li><img alt="Health.com_logo" src="style_sheets/images/Health_Logo.png" alt="logo"
class="logo"></li>
         <li class="dropdown">
            <a href="home_page.html" class="dropbtn">Home</a>
            <div class="dropdown-content">
               <a href="Gallery_pg.html">Gallery</a>
               <a href="Aboutus.html">About us</a>
            </div>
         </li>
         <li><a href="Product page.html">Products</a></li>
         <li><a href="quiz.html">Quiz</a></li>
         <li><a href="Feedbacklisting.html">Feedbacks</a></li>
         <li><a href="Signup_pg.html">Sign in</a></li>
      </ul>
   </nav>
```

For the A Healthy Place.com, the CSS code creates a visually appealing and intuitive navigation bar. Making use of the tag creates a structure that is obvious to both users and search engines. Bullets in the list format should be eliminated, and uniform padding and margins should be used to create clean lines. Because it doesn't have any unnecessary decoration and uses the parent element's color, the text in the navigation bar is still visible and clear. A solid border that matches the light blue backdrop and is rounded at the corners to provide a unified visual identity. When a link is hovered over, a light blue animation that fills the link's width activates an interactive underlining effect. A navigation bar that combines simple design concepts, a dash of visual flair, and simple user interface is produced.

```
nav ul {
   list-style-type: none;
   margin: 1;
   padding: 1;
   gap: 5rem;
   display: flex;
   justify-content: center;
   background-color: lightblue;
   border: 15px solid lightblue;
   border-radius: 15px;
}
nav a {
   position: relative;
   display: inline-block;
   text-decoration: none;
   color: inherit;
  }
```

```css
nav a::after {
  content: '';
  width: 0%;
  height: 2px;
  background: rgb(33, 159, 201);
  display: block;
  margin: auto;
  transition: .5s;
}

nav a:hover::after{
  width: 100%;
}

@keyframes enlarge {
    0% { transform: scale(1); }
    100% { transform: scale(1.1); }
}/*Keyframe modifer used here to enlarge obj with a .3s delay while having a size increment from 0% to 100%*/

/*Meant for the logo*/
.logo {
    position: absolute;
    top: 0;
    left: 0;
    width: 40px;
    height: 40px;
    margin: 20px;
  }

.logo:hover {
    transform: scale(1.5);
    animation: enlarge 0.5s infinite alternate;/*The animation loop is made endless here*/
}

.logoheader {
    position: absolute;
    margin-left: 65px;
    top: 6px;
    left: 1px;
    font-weight: bold;
    font-style: italic;
}

.rectangle {
    width: calc(100vw - 40px);
    height: calc(100vh - 40px);
    margin: 1;
    border-radius: 15px;
    border: 4px solid lightblue;
    background-color: lightblue;
```

```
  }

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content a:hover {
  background-color: #f1f1f1;
}

.dropdown:hover .dropdown-content {
  display: block;
}

.dropdown-content a::before,
.dropdown-content a::after {
    content: "";
    position: absolute;
    width: 100%;
    height: 100%;
    background-color: transparent;
    transition: background-color .7s ease;
}
```

4.2 Colour balance/selection/consistency

A healthy place website is a website used for a hospital, when considering colors for a website like this, factors like user's perspective and emotions need to be

considered. Healthy place website used a mixture of colors ,Light blue and white. These colors were chosen because these colors can be easy on users eyes and each color represents something unique like,

- Light blue represents warm and secure feeling, indicating all is well. Pale blue is loved for medical websites' backgrounds because it develops a peaceful and reliable atmosphere for the user.
- White is a mark of purity, simplicity and cleanliness, White color used in a website can create an atmosphere of sterility and efficiency. Moreover it will make the site look clean and organized

Hence color palette of light blue and white is used mostly in the healthy place webpage in order to make sure users gets a good impression for a healthcare Website.

Instead of using solid colors for the background, the home page uses an image to create a modern and eye-catching look. This method improves the user interface, giving users a more visually appealing and engaging experience.

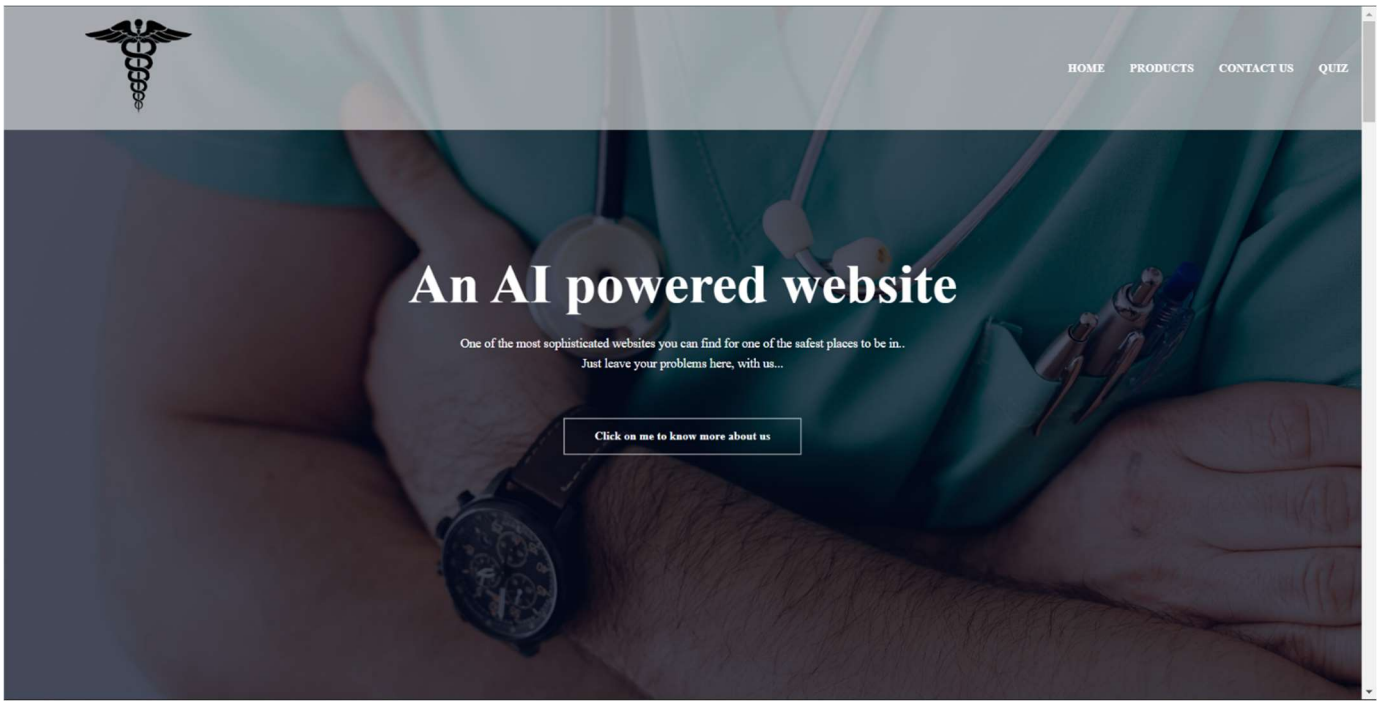


Figure 2 Home page

The site used a rectangular and oval layouts for various sections such as Products page, Quiz, Feedback page and About us page. Body background of all the sections except the Home page utilizes White and Blue to provide contrast and a user

friendly environment. Hovering effects, Shadows and borders are used are also used to increase the contrast of the overall web page. Although some users might find some insufficient contrast, it won't affect the functionality of the webpage since the main functional elements already uses high contrast colors.

```
.registration,
.personal,
.details {
    max-width: 600px;
    margin: 30px auto;
    text-align: left;
    background-color: lightblue;
    padding: 40px;
    border-radius: 20px;
    border: 7px solid rgb(4, 81, 136);
    box-shadow: 4px 6px 2px rgb(0,0);
}
```

Above code snippet is taken from the signup.css file which is linked to the signup_pg.html to add the background color of light blue and box shadow uses the color black to make it more visually appealing to the user.

```
left .middle .item:hover {
    outline: 5px solid rgb(4, 81, 136);
    outline-offset:0px;
}
```

Above code snippet is take from the product page.css file which is linked with the product page.html this css codes will make sure to display and outline of dark blue around the products with a cream color inside the outline to increase the contrast and interactivity in the products page.

Figure 3 Product hovering effect

4.3 Colour Contrast Test

In this section, evidence is provided that each page meets the colour contrast test requirements, except for one section. The body colours on the Feedback page do not pass the colour contrast check. However, due to the need to maintain this consistency, it has been left unchanged.it has been made such that this page is meant to cater to users that wish to give feedback on their experience with the quiz, so the following constraints had to be met.

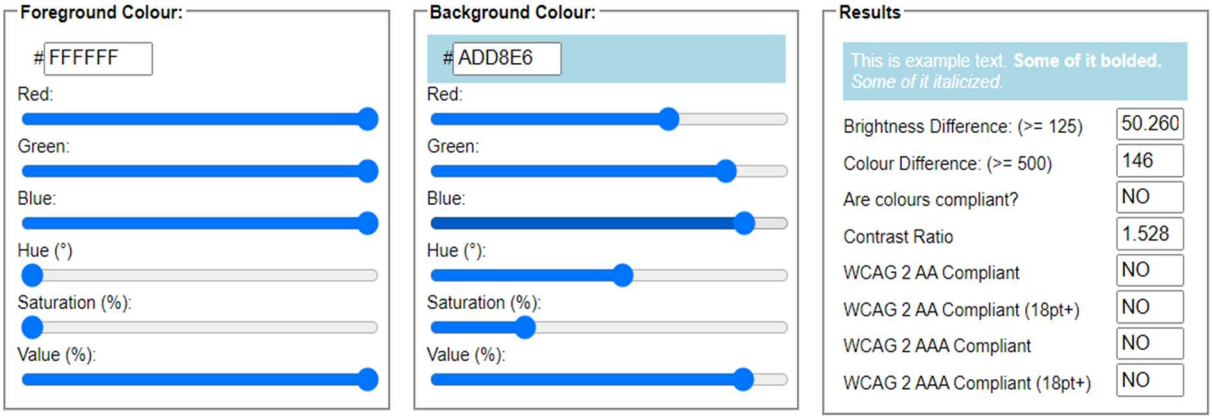


Figure 4 Feedback page body color contrast test

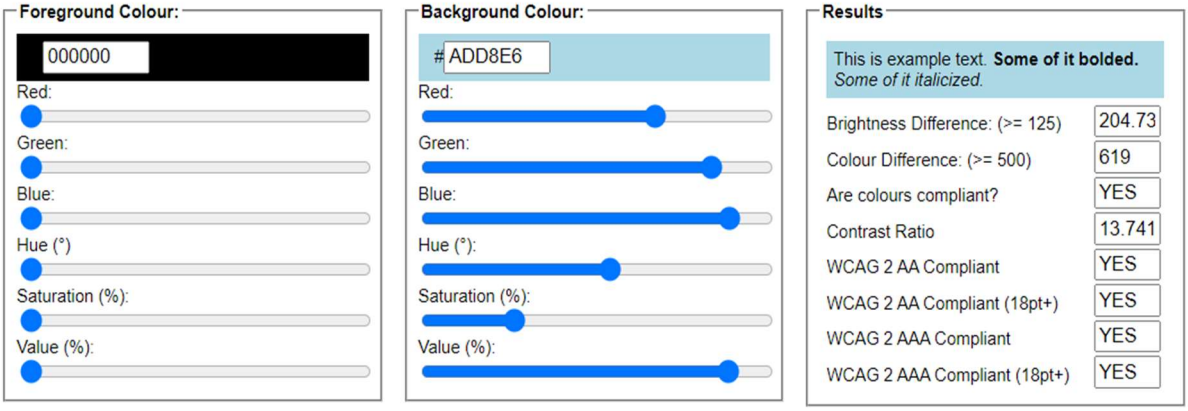


Figure 5 Nav bar and footer color contrast test

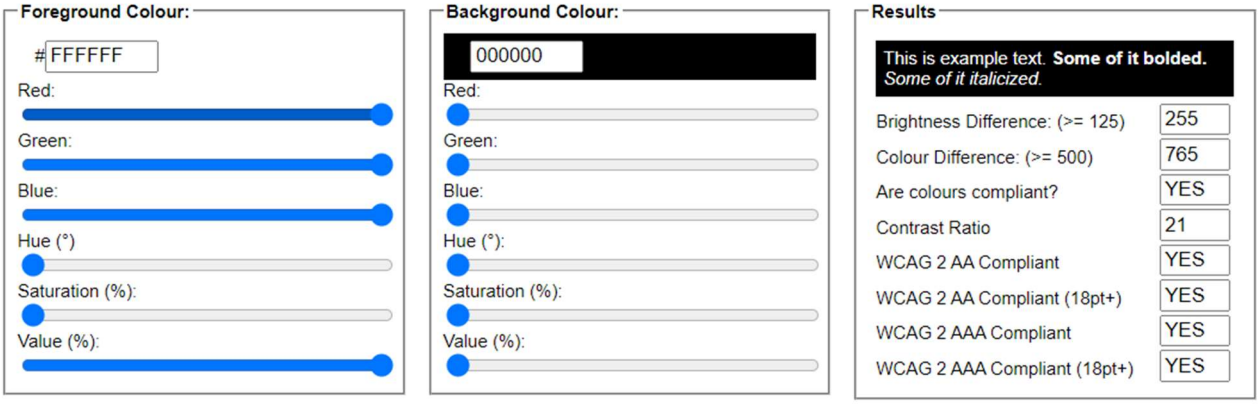


Figure 6 Buttons color contrast test

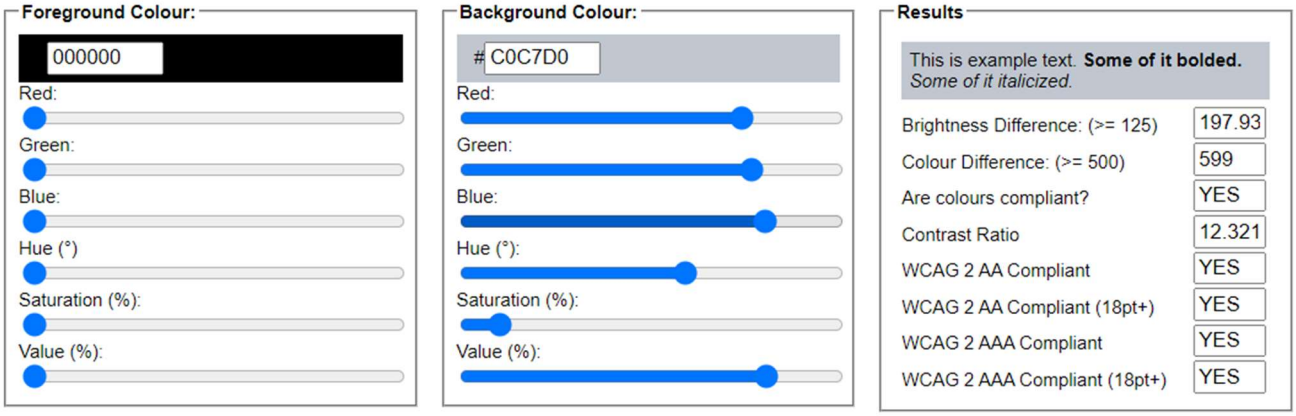


Figure 7 Home page website name color contrast test

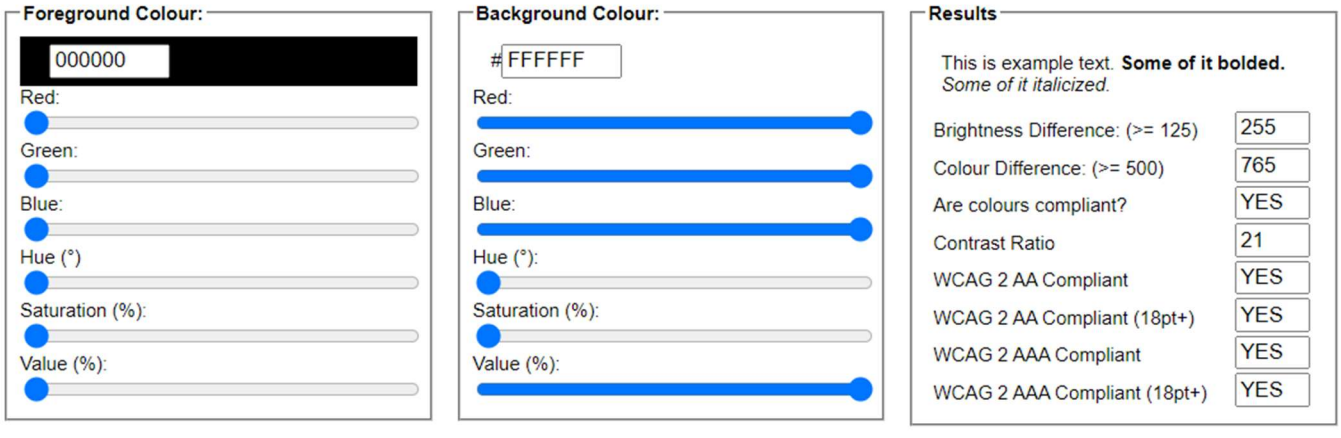


Figure 8 Quiz page color contrast test

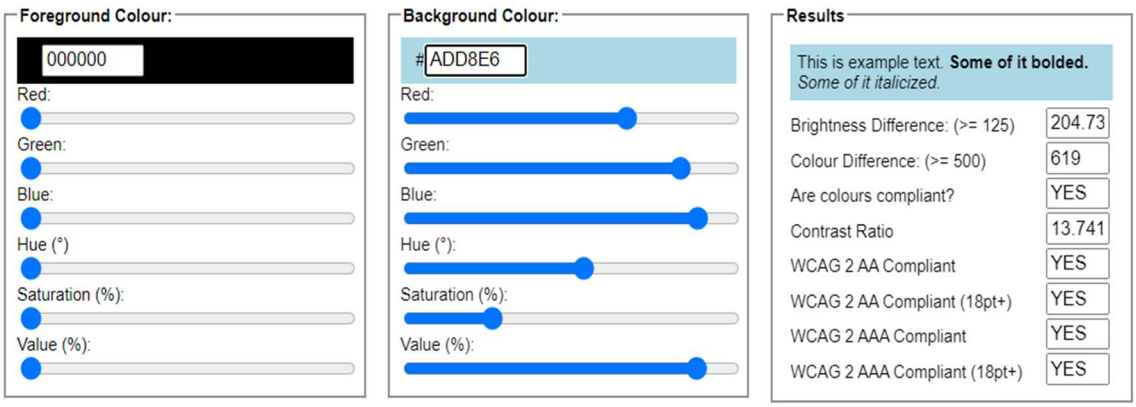


Figure 9 Sign up page color contrast test

4.4 Typography/consistency

In terms of the fonts and sizes referred to, we had used "Times New Roman" which is a font that is relatively easy to read and make out. Besides that, the letters space itself out from each other such that the design of the font makes it such that the page looks simple and uncomplicated, making it an ideal font for the context of a hospital web page. Furthermore, to ensure that the font remained consistent with all the pages developed, it had been made such that on each styling sheet all the elements present in them were to use a single font. In terms of the sizes referred to, to make the text appear relevant and not too large at the same time to the user we had used an average font size of 20px~10px. This range referred to here is both subtle and explicit in implying its presence on a page.

4.5 Accessibility

With regard to the accessibility of our website, we had referred to a series of techniques to ensure that the users has an ideal experience on our web pages such that they can easily navigate through them and reach the point they wish to reach such that they have no issues in the process of doing so. Regard the following subsections to see how each accessibility technique mentioned below was implemented with a snippet from the web pages:

4.5.1 Text Accessibility Techniques

In terms of improving the accessibility of the text on each page we had used a set font size and style such that it would not vary per page and is easy to read and skim through. Regard the following snippet to see how the technique was implemented:

```
* {
    font-family:'Times New Roman', Times, serif;
}
```

The font was applied to the entire page. In the html tag in the html pages the lang attribute was used to specify the language to be used. Furthermore, with regard to the listings page it may be seen that the background colour and text colour may be changed to something that the user would prefer.

4.5.2 Image Accessibility Techniques

With regard to the images present in each page, we had ensured that each images is provided with an alternative text which gives each image a descriptive text of what they are supposed to be. In case a user does not understand the purpose of an image on a page, they may simply refer to the alternate text to understand it. Aside from the use of alternate texts, the images on the home page were made to be responsive such that when the user hovers over them they are given a small animation or hover effect to enhance their experience. Regard the following snippet to see how the logo on the home page was animated:

HTML:

```
<a href="home_page.html"><img src="images/Health_Logo.png" class="nav-logo"
alt="logo"></a>
```

CSS:

```
@keyframes enlarge {
    0% { transform: scale(1); }
    100% { transform: scale(1.1); }
}/*Keyframe modifer used here to enlarge obj with a .3s delay while having a
size increment from 0% to 100%*/

.nav-logo:hover{
    transform: scale(1.5);
    animation: enlarge 0.5s infinite alternate;/*The animation loop is made
endless here*/
}
```

The animation loop here makes it seem as if there is a heartbeat when the logo is hovered over.

4.5.3 Table Accessibility Techniques

Our pages did not regard the use of tables; however it was utilized in the development of the quizzes and forms present in the page to create a structure for them an align them such that they look like they are aligned with each other in an organized manner. To improve its

accessibility, each row and column was defined through the use of an unordered list. Refer to the following code snippet to see how a few accessibility techniques were implemented:

```
<tr>
    <td><label for="contact">Contact<br>Details:</label></td>
    <td><input type="tel" id="contact" name="contact"></td>
</tr>
```

A table was referred to here in order to align each element and question with each other.

4.5.4 Form Accessibility Techniques

To improve the accessibility of our forms and quizzes we had regarded the use of labels to ensure that the user may understand what a question is meant to be or what is being asked of them from that question. Furthermore, the use of placeholder values in certain questions make it such that the user is given a hint of what the answer is supposed to be, conclusively helping the user understand what they are supposed to do. Regard the following snippet to get an idea of how these techniques were implemented:

```
<form class="health_form" id="usr_dets_form" onsubmit="feedback_required()">
    <fieldset>
        <legend>Fill in the details</legend>
        <table class="form_table">
            <tr>
                <td><label for="name">Name</label></td>
                <td><input type="text" id="name" name="name" value="Enter
your name..." onfocus="clearPrompt(this)" onblur="returnPrompt(this)"></td>
            </tr>
            <tr>
                <td><label for="email">Email</label></td>
                <td><input type="email" id="email" name="email" value="Enter
your email here..." onfocus="clearPrompt(this)"
onblur="returnPrompt(this)"></td>
            </tr>
            <tr>
                <td><label for="age">Age</label></td>
                <td><input type="number" id="age" name="age"></td>
            </tr>
            <tr>
                <td><label for="contact">Contact Details:</label></td>
                <td><input type="tel" id="contact" name="contact"
value="Enter your contact details here..." onfocus="clearPrompt(this)"
onblur="returnPrompt(this)"></td>
            </tr>
            <tr>
```

```
                <td><label for="Feedback">Your Feedback:</label></td>
                <td><input type="tel" id="Feedback" name="Feedback"
value="Enter your feedback here... (optional)" class="large_text"
onfocus="clearPrompt(this)" onblur="returnPrompt(this)"></td>
            </tr>
            <tr>
                <td><label for="Rating">Rating:</label></td>
                <td><input type="number" id="rating" name="rating"></td>
            </tr>
            <tr>
                <td><button type="submit" id="submit" name="submit"
value="submit">Submit</button></td>
                <td><button type="reset" id="reset" name="reset"
value="reset">Reset</button></td>
            </tr>
        </table>
    </fieldset>
</form>
```

4.6 Accessibility Test

Regard the following screenshots to see how the accessibility test was for the home and product page:

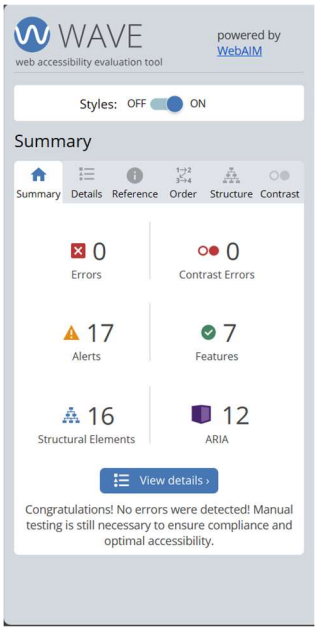


Figure 10: Accessibility test for the home page

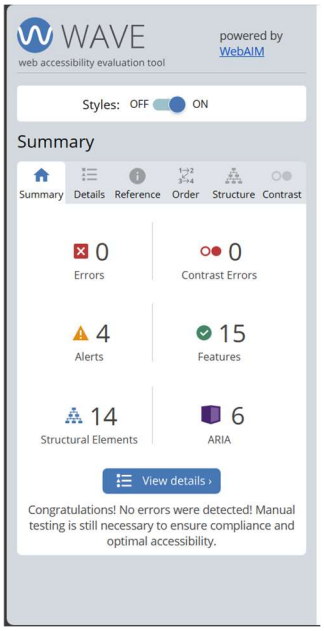


Figure 11: Accessibility test for the product page

4.7 Site Diagram

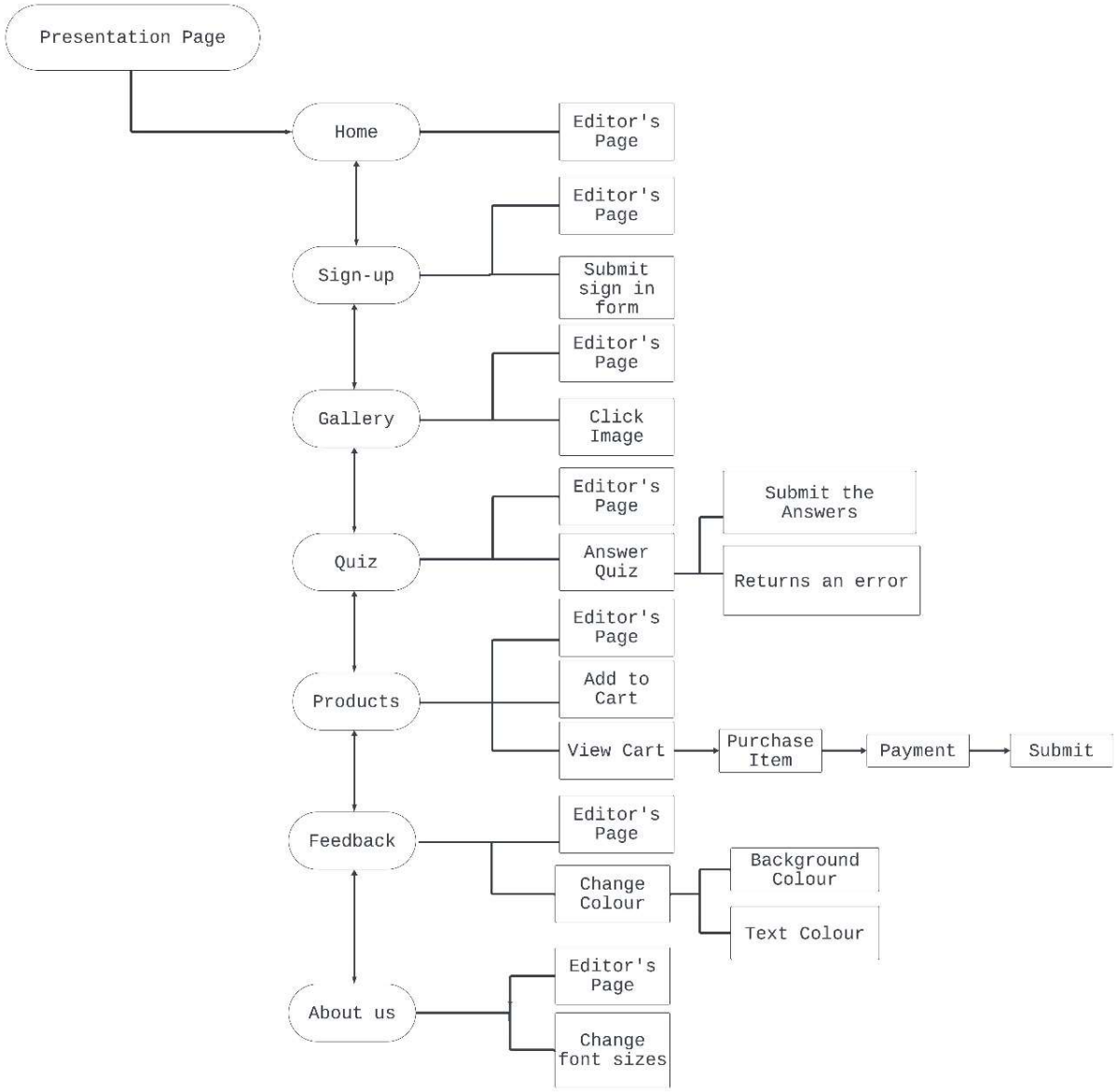


Figure 12: Site DIagram for the website

5.0 Self Reflection

5.1 Student 1 (Mindiya)

Student 1 encountered difficulties when constructing the gallery page because of inconsistent thumbnail dimensions and when developing the sign-up form because of alterations to the fields post-validation. Through exploration of graphic design principles and acquisition of skills in JavaScript DOM manipulation, the tasks were ultimately accomplished.

5.2 Student 2 (Ethan)

Designing a hospital quiz page with JavaScript proved challenging due to its asynchronous nature and limitations in handling dynamic content and user interaction. Overcoming complexities in content rendering, cross-browser compatibility, and validation underscored the importance of resilience and adaptability in navigating JavaScript constraints, fostering valuable learning experiences in web development.

5.3 Student 3 (Chirath)

When creating the product page for "A Healthy Place.com," I put the needs of the customer first and integrated shopping cart and product research features with ease. Clarity in layout and design was ensured by attention to detail, and a responsive design accommodated a wide range of user types. Working together was essential to creating a seamless, user-cantered experience.

5.4 Student 4 (Chanul)

At first, I had trouble getting XML files to load dynamically onto the page. Statically hardcoding the feedback list would have been the obvious solution, but I was eager to investigate more dynamic options. My objective was to create a system that could instantly retrieve and show actual user feedback from an XML file.

6.0 References

w3Schools (2019). JavaScript HTML DOM. [online] W3schools.com. Available at: https://www.w3schools.com/js/js_htmldom.asp.

W3Schools (n.d.). JavaScript DOM EventListener. [online] www.w3schools.com. Available at: https://www.w3schools.com/js/js_htmldom_eventlistener.asp.

www.w3schools.com. (n.d.). onclick Event. [online] Available at: https://www.w3schools.com/jsref/event_onclick.asp.

www.w3schools.com. (n.d.). CSS :hover Selector. [online] Available at: https://www.w3schools.com/CSSref/sel_hover.php.

Web Dev Simplified (2019). Learn HTML Forms In 25 Minutes. YouTube. Available at: https://www.youtube.com/watch?v=fNcJuPIZ2WE.

www.youtube.com. (n.d.). Responsive Product Cards design with HTML and CSS. [online] Available at: https://www.youtube.com/watch?v=kRs3aTi3pzU.

www.youtube.com. (n.d.). Show Thumbnail in Large Size Image When Click On with pure Html CSS without Javascript. [online] Available at: https://www.youtube.com/watch?v=pDd4Wf8Ew-s [Accessed 7 Apr. 2024].

www.youtube.com. (n.d.). The Easiest Way to Parse XML with JavaScript. [online] Available at: https://www.youtube.com/watch?v=lUCQgqc4K2A&t=335s [Accessed 7 Apr. 2024].