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	: B		
Coursework Group No	: 7		
Student Details	:		
	2237925	20221732	Lukmal Ilyas
	2236755	20210902	Minudaka Liyanage
	2237036	20221585	Mandinu Handapangoda
	2237044	20221359	Yenuka Rajapaksha

Contents

1. Introduction

ScheduleIt, is a scheduling assistant application , is more than just a scheduling tool. It seamlessly syncs with your calendar and sends automated reminders, ensuring you never miss an appointment again. Plus, it offers real-time availability and time zone conversion, making it easy to schedule with people across the globe.

- Member 1 2237925 – Student role 1
- Member 2 2236755 – Student role 2
- Member 3 2237036 – Student role 3
- Member 4 2237044 – Student role 4

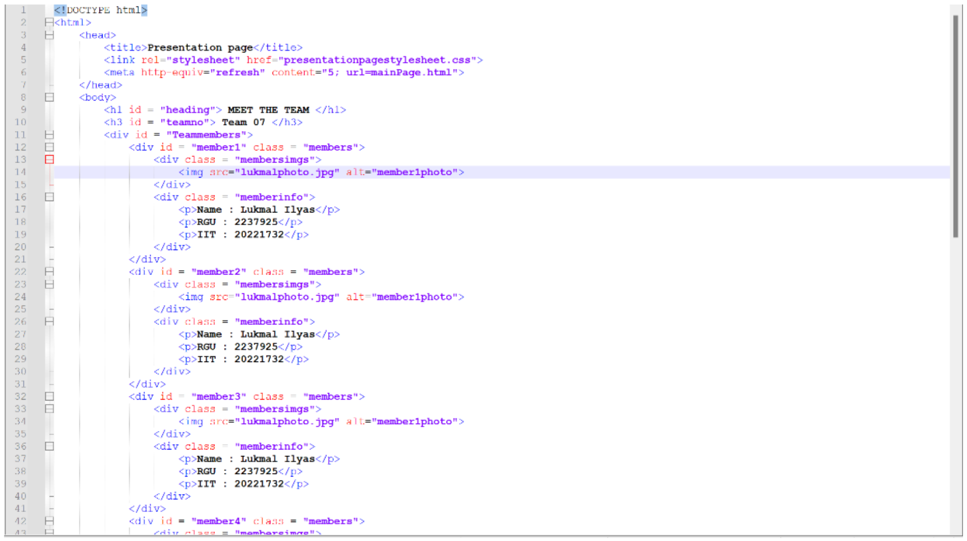
2. Research on existing systems

Calendly and doodle are two similar websites used for scheduling. Both sites are used to schedule meetings and work-related things. They provide paid services for professionals to take care of their meetings. However, they cannot be used to schedule and plan day to day activities. Doodle is used in many areas such as educations, sales, recruiting etc. It provides poll options which increases engagement among users. It also allows to share your availability with others. You can set a time for a particular meeting as well. It is also used to schedule interviews for recruiting by companies. This also provides the opportunity to connect your calendar to the website. On the other hand, Calendly allows its user to schedule meeting through it. It allows users to set automatic emails to remind the other party about the scheduled meetings.
It allows the whole organization to connect through it. It gives the ability to be used on any device. Overall, both this scheduling sites provides a great platform for the users to schedule and manage their meetings with their colleagues and business partners. But they do not provide a solution for the individual person to plan his or her day. They also do not predict the best days to do tasks.

3. Technical Discussion

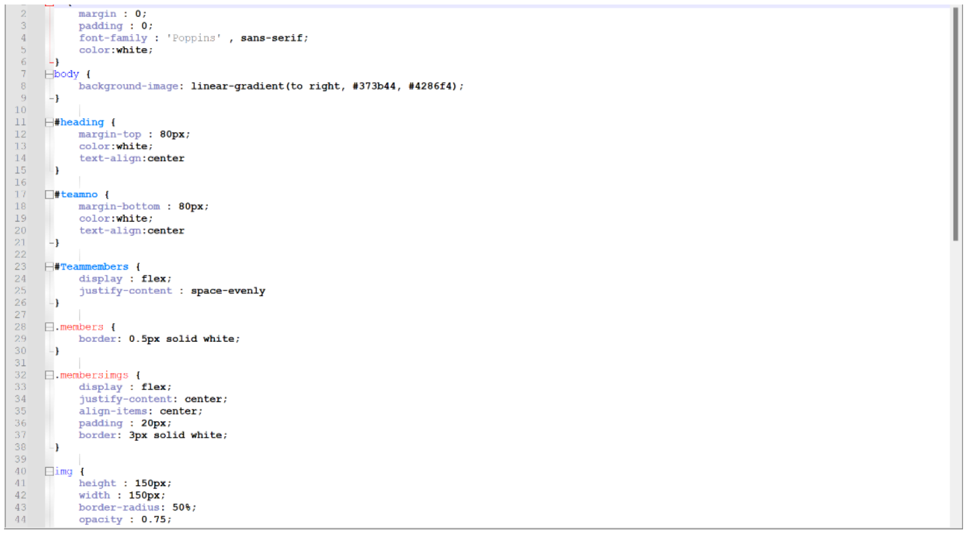
Student 1
Presentation page

In the presentation page images of members and their details are placed in div containers.

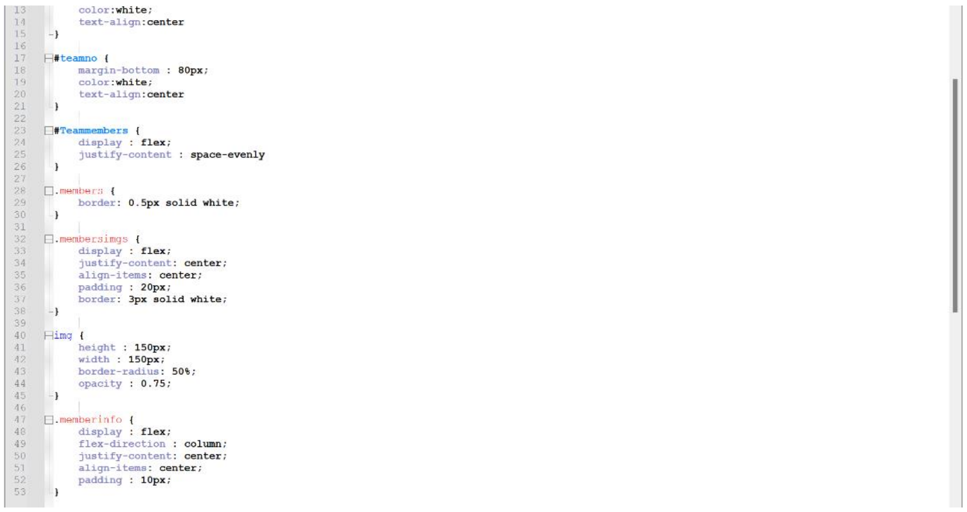


```html
<!DOCTYPE html>
<html>
    <head>
        <title>Presentation page</title>
        <link rel="stylesheet" href="presentationpagestylesheet.css">
        <meta http-equiv="refresh" content="5; url=mainPage.html">
    </head>
    <body>
        <h1 id = "heading"> MEET THE TEAM </h1>
        <h3 id = "teamno"> Team 07 </h3>
        <div id = "Teammembers">
            <div id = "member1" class = "members">
                <div class = "membersimgs">
                    <img src="lukmalphoto.jpg" alt="member1photo">
                </div>
                <div class = "memberinfo">
                    <p>Name : Lukmal Ilyas</p>
                    <p>RGU : 2237925</p>
                    <p>IIT : 20221732</p>
                </div>
            </div>
            <div id = "member2" class = "members">
                <div class = "membersimgs">
                    <img src="lukmalphoto.jpg" alt="member1photo">
                </div>
                <div class = "memberinfo">
                    <p>Name : Lukmal Ilyas</p>
                    <p>RGU : 2237925</p>
                    <p>IIT : 20221732</p>
                </div>
            </div>
            <div id = "member3" class = "members">
                <div class = "membersimgs">
                    <img src="lukmalphoto.jpg" alt="member1photo">
                </div>
                <div class = "memberinfo">
                    <p>Name : Lukmal Ilyas</p>
                    <p>RGU : 2237925</p>
                    <p>IIT : 20221732</p>
                </div>
            </div>
            <div id = "member4" class = "members">
                <div class = "membersimgs">
```

Margin and padding are set 0px initially, font is set to Poppins sans serif color white. The background color is set to linear gradient (to right, #373b44, #4286f4). The margin of the heading is set to 80px, color is set to white, and the text is aligned to center. Member images are flexed and aligned center, height and width is set to 150px and boarder of 0.5px is given. Opacity of member images is set 0.75 and boarder-radius is set to 50%. Member descriptions are column flexed and aligned center.



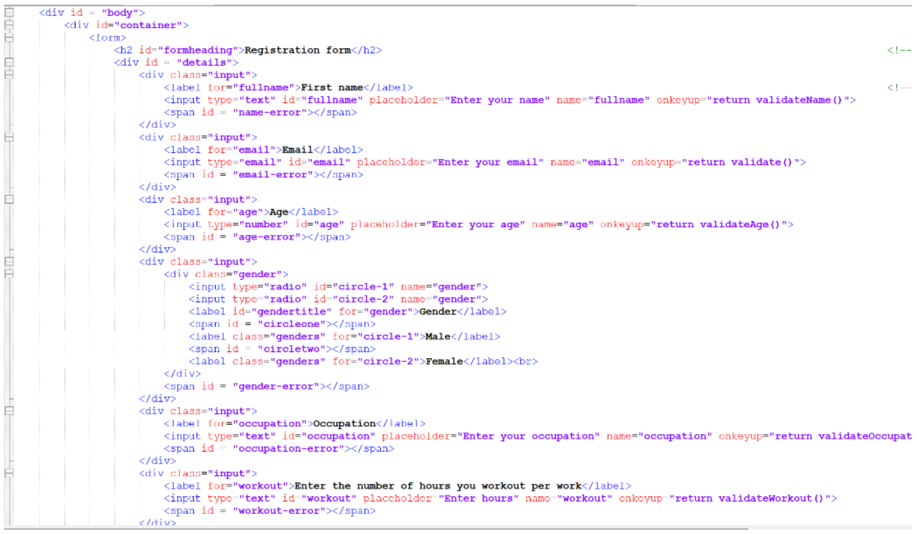
```css
    margin : 0;
    padding : 0;
    font-family : 'Poppins' , sans-serif;
    color:white;
}
body {
    background-image: linear-gradient(to right, #373b44, #4286f4);
}

#heading {
    margin-top : 80px;
    color:white;
    text-align:center
}

#teamno {
    margin-bottom : 80px;
    color:white;
    text-align:center
}

#Teammembers {
    display : flex;
    justify-content : space-evenly
}

.members {
    border: 0.5px solid white;
}

.membersimgs {
    display : flex;
    justify-content: center;
    align-items: center;
    padding : 20px;
    border: 3px solid white;
}

img {
    height : 150px;
    width : 150px;
    border-radius: 50%;
    opacity : 0.75;
```



```css
    color:white;
    text-align:center
}

#teamno {
    margin-bottom : 80px;
    color:white;
    text-align:center
}

#Teammembers {
    display : flex;
    justify-content : space-evenly
}

.members {
    border: 0.5px solid white;
}

.membersimgs {
    display : flex;
    justify-content: center;
    align-items: center;
    padding : 20px;
    border: 3px solid white;
}

img {
    height : 150px;
    width : 150px;
    border-radius: 50%;
    opacity : 0.75;
}

.memberinfo {
    display : flex;
    flex-direction : column;
    justify-content: center;
    align-items: center;
    padding : 10px;
}
```

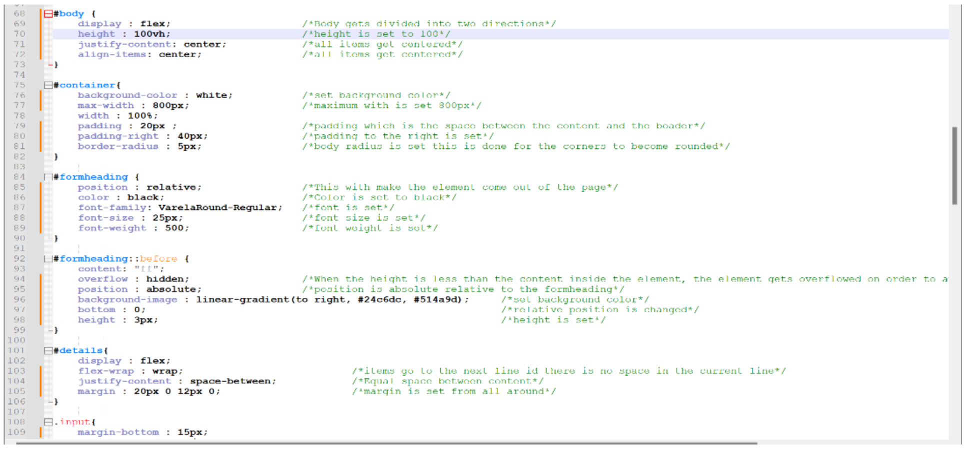
Registration form

A form is made to enter user details and user preferences. Labels and input tags are used for the input in the form. All the inputs are placed in containers. Span tags are used for pop errors when the user enters wrong information.

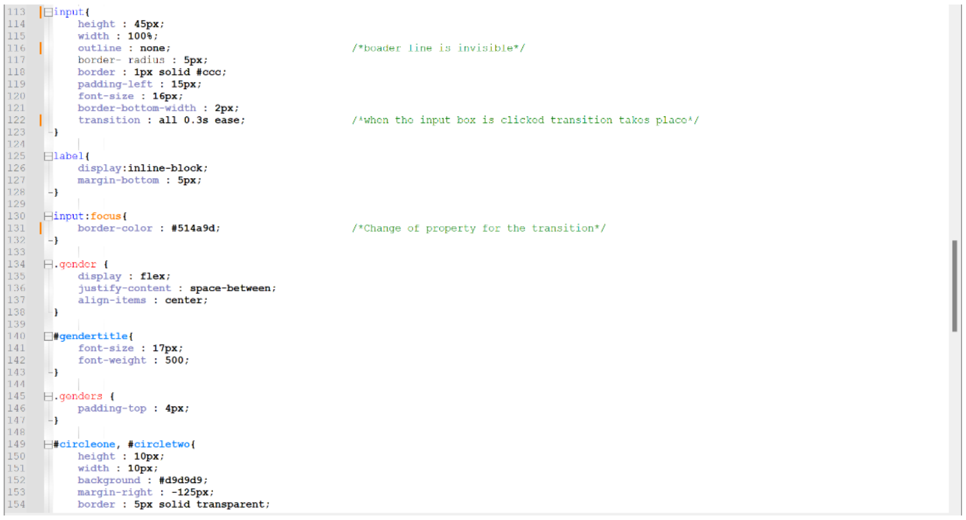


```html
<div id = "body">
    <div id="container">
        <form>
            <h2 id="formheading">Registration form</h2>                                                  <!--
            <div id = "details">
                <div class="input">
                    <label for="fullname">First name</label>                                             <!--
                    <input type="text" id="fullname" placeholder="Enter your name" name="fullname" onkeyup="return validateName()">
                    <span id = "name-error"></span>
                </div>
                <div class="input">
                    <label for="email">Email</label>
                    <input type="email" id="email" placeholder="Enter your email" name="email" onkeyup="return validate()">
                    <span id = "email-error"></span>
                </div>
                <div class="input">
                    <label for="age">Age</label>
                    <input type="number" id="age" placeholder="Enter your age" name="age" onkeyup="return validateAge()">
                    <span id = "age-error"></span>
                </div>
                <div class="input">
                    <div class="gender">
                        <input type="radio" id="circle-1" name="gender">
                        <input type="radio" id="circle-2" name="gender">
                        <label id="gendertitle" for="gender">Gender</label>
                        <span id = "circleone"></span>
                        <label class="genders" for="circle-1">Male</label>
                        <span id = "circletwo"></span>
                        <label class="genders" for="circle-2">Female</label><br>
                    </div>
                    <span id = "gender-error"></span>
                </div>
                <div class="input">
                    <label for="occupation">Occupation</label>
                    <input type="text" id="occupation" placeholder="Enter your occupation" name="occupation" onkeyup="return validateOccupat
                    <span id = "occupation-error"></span>
                </div>
                <div class="input">
                    <label for="workout">Enter the number of hours you workout per work</label>
                    <input type="text" id="workout" placeholder="Enter hours" name="workout" onkeyup="return validateWorkout()">
                    <span id = "workout-error"></span>
                </div>
```

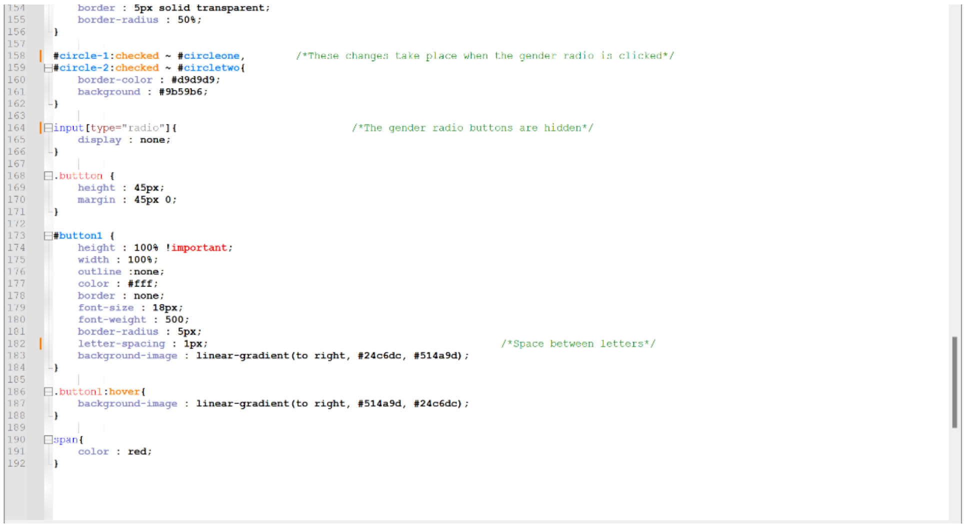
The body is set to display flex to align items to center. Background color of the from set to white and width and padding set accordingly to the page. In the form heading font is changed to VaeralaRound-Regular and position is set to relative. Input boxes are set to flex wrap and space between is set to equal. Input boxes are set to width 50% of the container and margin-bottom is set to 15px. Button color is set to linear-gradient(to right, #24c6dc, #514a9d), width and height is set to 45px, font-size is set to 18px and color to white, border radius is set to 5px and none and transition set to hover and reverse gradient.



```css
#body {
    display : flex;                     /*Body gets divided into two directions*/
    height : 100vh;                     /*height is set to 100*/
    justify-content: center;            /*all items get centered*/
    align-items: center;                /*all items get centered*/
}

#container{
    background-color : white;           /*set background color*/
    max-width : 800px;                  /*maximum with is set 800px*/
    width : 100%;
    padding : 20px ;                    /*padding which is the space between the content and the boader*/
    padding-right : 40px;               /*padding to the right is set*/
    border-radius : 5px;                /*body radius is set this is done for the corners to become rounded*/
}

#formheading {
    position : relative;                /*This with make the element come out of the page*/
    color : black;                      /*Color is set to black*/
    font-family: VarelaRound-Regular;   /*font is set*/
    font-size : 25px;                   /*font size is set*/
    font-weight : 500;                  /*font weight is set*/
}

#formheading::before {
    content: "ff";
    overflow : hidden;                  /*When the height is less than the content inside the element, the element gets overflowed on order to a
    position : absolute;                /*position is absolute relative to the formheading*/
    background-image : linear-gradient(to right, #24c6dc, #514a9d);     /*set background color*/
    bottom : 0;                                                         /*relative position is changed*/
    height : 3px;                                                       /*height is set*/
}

#details{
    display : flex;
    flex-wrap : wrap;                           /*items go to the next line id there is no space in the current line*/
    justify-content : space-between;            /*Equal space between content*/
    margin : 20px 0 12px 0;                     /*margin is set from all around*/
}

.input{
    margin-bottom : 15px;
```

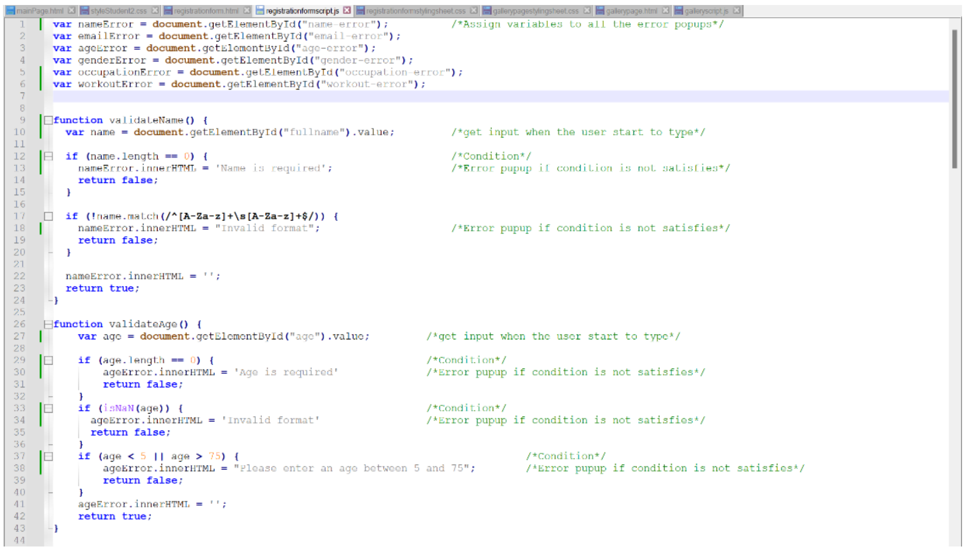


```css
input{
    height : 45px;
    width : 100%;
    outline : none;                             /*boader line is invisible*/
    border- radius : 5px;
    border : 1px solid #ccc;
    padding-left : 15px;
    font-size : 16px;
    border-bottom-width : 2px;
    transition : all 0.3s ease;                 /*when the input box is clicked transition takes place*/
}

label{
    display:inline-block;
    margin-bottom : 5px;
}

input:focus{
    border-color : #514a9d;                     /*Change of property for the transition*/
}

.gender {
    display : flex;
    justify-content : space-between;
    align-items : center;
}

#gendertitle{
    font-size : 17px;
    font-weight : 500;
}

.genders {
    padding-top : 4px;
}

#circleone, #circletwo{
    height : 10px;
    width : 10px;
    background : #d9d9d9;
    margin-right : -125px;
    border : 5px solid transparent;
```



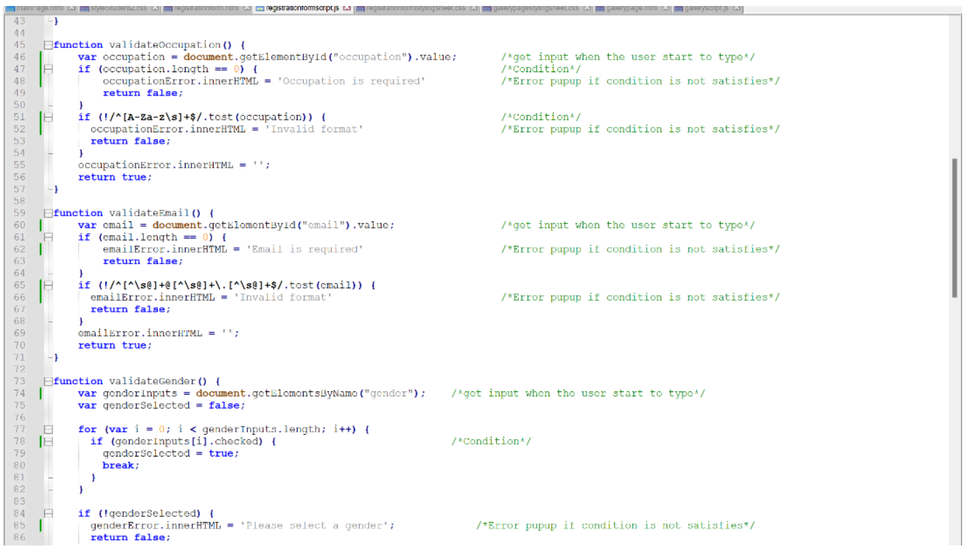
```css
    border : 5px solid transparent;
    border-radius : 50%;
}

#circle-1:checked ~ #circleone,         /*These changes take place when the gender radio is clicked*/
#circle-2:checked ~ #circletwo{
    border-color : #d9d9d9;
    background : #9b59b6;
}

input[type="radio"]{                            /*The gender radio buttons are hidden*/
    display : none;
}

.button {
    height : 45px;
    margin : 45px 0;
}

#button1 {
    height : 100% !important;
    width : 100%;
    outline :none;
    color : #fff;
    border : none;
    font-size : 18px;
    font-weight : 500;
    border-radius : 5px;
    letter-spacing : 1px;                                               /*Space between letters*/
    background-image : linear-gradient(to right, #24c6dc, #514a9d);
}

.button1:hover{
    background-image : linear-gradient(to right, #514a9d, #24c6dc);
}

span{
    color : red;
}
```

Variables are created for span tags to show an error message if the input is not correctly entered. For each input there are set of conditions where when the condition is not satisfied an error message will show and a false Boolean is returned from the function. If the user has entered correct information true Boolean is returned from the function. After pressing submit there is a condition to check which Boolean is returned from each function is true. Thereafter, an alert will pop up with the user's name only if the user has entered the information properly, if the user has entered the information correctly another alert will pop up saying invalid input.



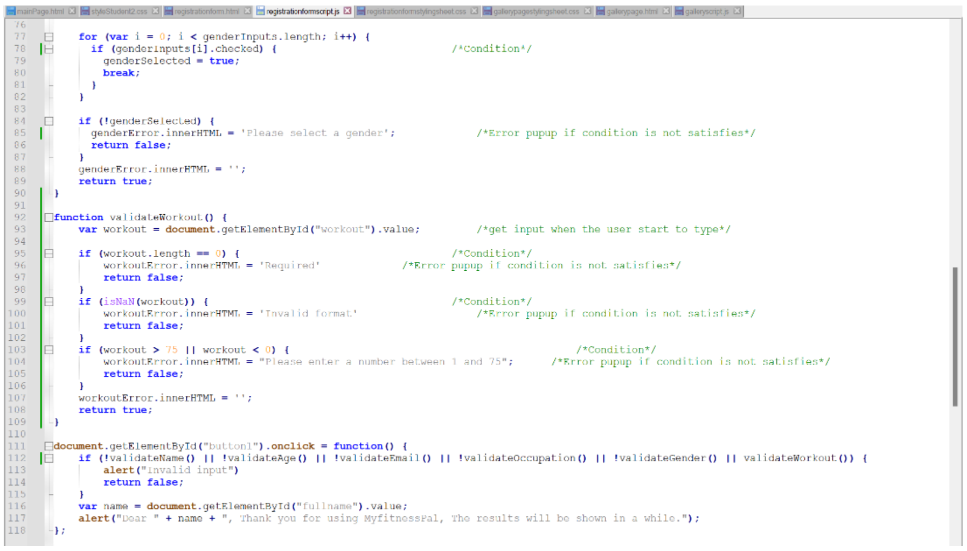
mainPage.html
styleStudent2.css
registrationform.html
registrationformscript.js
registrationformstylingsheet.css
gallerypagestylingsheet.css
gallerypage.html
galleryscript.js

```javascript
var nameError = document.getElementById("name-error");          /*Assign variables to all the error popups*/
var emailError = document.getElementById("email-error");
var ageError = document.getElementById("age-error");
var genderError = document.getElementById("gender-error");
var occupationError = document.getElementById("occupation-error");
var workoutError = document.getElementById("workout-error");


function validateName() {
  var name = document.getElementById("fullname").value;         /*get input when the user start to type*/

  if (name.length == 0) {                                       /*Condition*/
    nameError.innerHTML = 'Name is required';                   /*Error pupup if condition is not satisfies*/
    return false;
  }

  if (!name.match(/^[A-Za-z]+\s[A-Za-z]+$/)) {
    nameError.innerHTML = "Invalid format";                     /*Error pupup if condition is not satisfies*/
    return false;
  }

  nameError.innerHTML = '';
  return true;
}

function validateAge() {
    var age = document.getElementById("age").value;         /*get input when the user start to type*/

    if (age.length == 0) {                                  /*Condition*/
        ageError.innerHTML = 'Age is required'              /*Error pupup if condition is not satisfies*/
        return false;
    }
    if (isNaN(age)) {                                       /*Condition*/
      ageError.innerHTML = 'Invalid format'                 /*Error pupup if condition is not satisfies*/
      return false;
    }
    if (age < 5 || age > 75) {                                              /*Condition*/
        ageError.innerHTML = "Please enter an age between 5 and 75";        /*Error pupup if condition is not satisfies*/
        return false;
    }
    ageError.innerHTML = '';
    return true;
}

```



mainPage.html
styleStudent2.css
registrationform.html
registrationformscript.js
registrationformstylingsheet.css
gallerypagestylingsheet.css
gallerypage.html
galleryscript.js

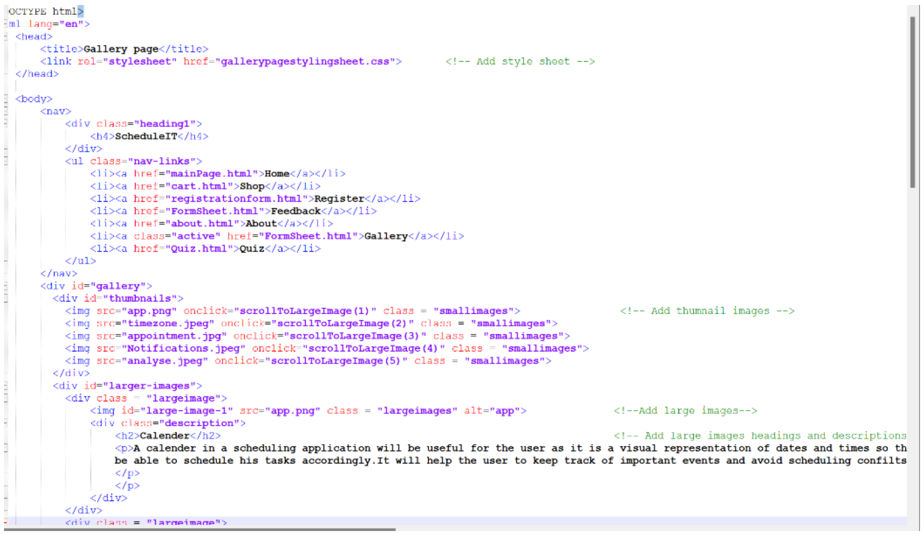
```javascript
}

function validateOccupation() {
    var occupation = document.getElementById("occupation").value;       /*get input when the user start to type*/
    if (occupation.length == 0) {                                       /*Condition*/
        occupationError.innerHTML = 'Occupation is required'            /*Error pupup if condition is not satisfies*/
        return false;
    }
    if (!/^[A-Za-z\s]+$/.test(occupation)) {                            /*Condition*/
      occupationError.innerHTML = 'Invalid format'                      /*Error pupup if condition is not satisfies*/
      return false;
    }
    occupationError.innerHTML = '';
    return true;
}

function validateEmail() {
    var email = document.getElementById("email").value;                 /*get input when the user start to type*/
    if (email.length == 0) {
        emailError.innerHTML = 'Email is required'                      /*Error pupup if condition is not satisfies*/
        return false;
    }
    if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {
      emailError.innerHTML = 'Invalid format'                           /*Error pupup if condition is not satisfies*/
      return false;
    }
    emailError.innerHTML = '';
    return true;
}

function validateGender() {
    var genderInputs = document.getElementsByName("gender");    /*get input when the user start to type*/
    var genderSelected = false;

    for (var i = 0; i < genderInputs.length; i++) {
      if (genderInputs[i].checked) {                            /*Condition*/
        genderSelected = true;
        break;
      }
    }

    if (!genderSelected) {
      genderError.innerHTML = 'Please select a gender';             /*Error pupup if condition is not satisfies*/
      return false;
```


mainPage.html
styleStudent2.css
registrationform.html
registrationformscript.js
registrationformstylingsheet.css
gallerypagestylingsheet.css
gallerypage.html
galleryscript.js

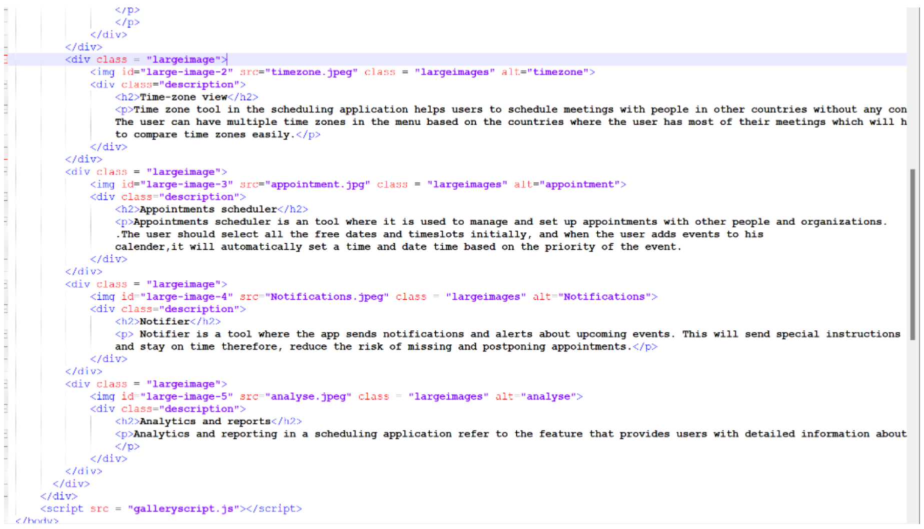
```
    for (var i = 0; i < genderInputs.length; i++) {
      if (genderInputs[i].checked) {                        /*Condition*/
        genderSelected = true;
        break;
      }
    }

    if (!genderSelected) {
      genderError.innerHTML = 'Please select a gender';              /*Error pupup if condition is not satisfies*/
      return false;
    }
    genderError.innerHTML = '';
    return true;
}

function validateWorkout() {
    var workout = document.getElementById("workout").value;          /*get input when the user start to type*/

    if (workout.length == 0) {                              /*Condition*/
        workoutError.innerHTML = 'Required'              /*Error pupup if condition is not satisfies*/
        return false;
    }
    if (isNaN(workout)) {                                   /*Condition*/
        workoutError.innerHTML = 'Invalid format'                     /*Error pupup if condition is not satisfies*/
        return false;
    }
    if (workout > 75 || workout < 0) {                                          /*Condition*/
        workoutError.innerHTML = "Please enter a number between 1 and 75";      /*Error pupup if condition is not satisfies*/
        return false;
    }
    workoutError.innerHTML = '';
    return true;
}

document.getElementById("button1").onclick = function() {
    if (!validateName() || !validateAge() || !validateEmail() || !validateOccupation() || !validateGender() || validateWorkout()) {
        alert("Invalid input")
        return false;
    }
    var name = document.getElementById("fullname").value;
    alert("Dear " + name + ", Thank you for using MyfitnessPal, The results will be shown in a while.");
};
```

Gallery page

A gallery page contains thumbnail pictures of the large images. Small images are added using img tag and they are placed in a div container. Large images and their descriptions are placed in div container. Large images are added using img.

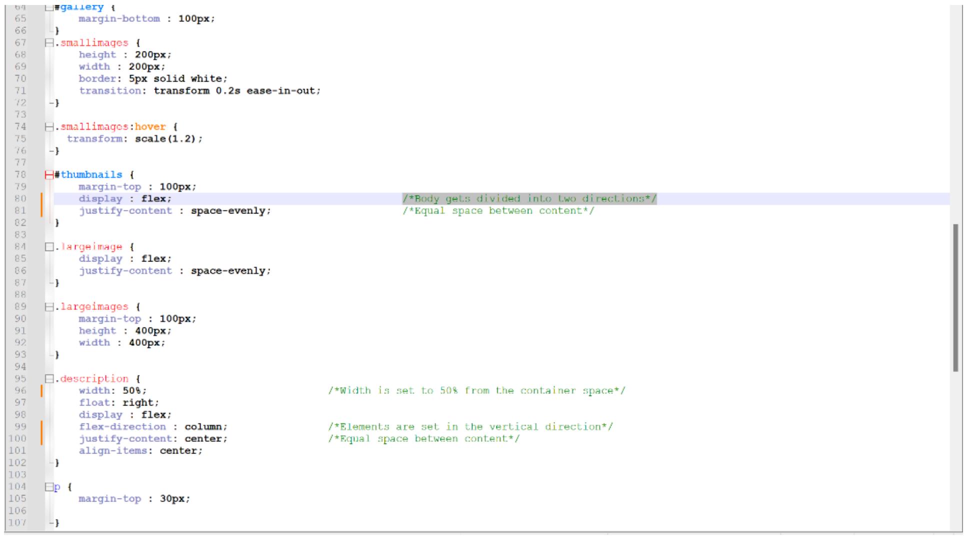


```
OCTYPE html>
ml lang="en">
 <head>
     <title>Gallery page</title>
     <link rel="stylesheet" href="gallerypagestylingsheet.css">       <!-- Add style sheet -->
 </head>

 <body>
     <nav>
         <div class="heading1">
             <h4>ScheduleIT</h4>
         </div>
         <ul class="nav-links">
             <li><a href="mainPage.html">Home</a></li>
             <li><a href="cart.html">Shop</a></li>
             <li><a href="registrationform.html">Register</a></li>
             <li><a href="FormSheet.html">Feedback</a></li>
             <li><a href="about.html">About</a></li>
             <li><a class="active" href="FormSheet.html">Gallery</a></li>
             <li><a href="Quiz.html">Quiz</a></li>
         </ul>
     </nav>
     <div id="gallery">
       <div id="thumbnails">
         <img src="app.png" onclick="scrollToLargeImage(1)" class = "smallimages">               <!-- Add thumnail images -->
         <img src="timezone.jpeg" onclick="scrollToLargeImage(2)" class = "smallimages">
         <img src="appointment.jpg" onclick="scrollToLargeImage(3)" class = "smallimages">
         <img src="Notifications.jpeg" onclick="scrollToLargeImage(4)" class = "smallimages">
         <img src="analyse.jpeg" onclick="scrollToLargeImage(5)" class = "smallimages">
       </div>
       <div id="larger-images">
         <div class = "largeimage">
             <img id="large-image-1" src="app.png" class = "largeimages" alt="app">              <!--Add large images-->
             <div class="description">
                 <h2>Calender</h2>                                                                <!-- Add large images headings and descriptions
                 <p>A calender in a scheduling application will be useful for the user as it is a visual representation of dates and times so th
                 be able to schedule his tasks accordingly.It will help the user to keep track of important events and avoid scheduling confilts
                 </p>
                 </p>
             </div>
         </div>
         <div class = "largeimage">
```



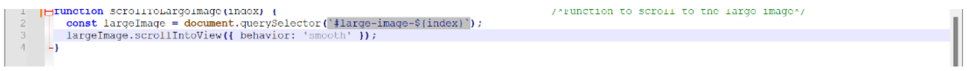
```html
                </p>
                </p>
            </div>
        </div>
        <div class = "largeimage">
            <img id="large-image-2" src="timezone.jpeg" class = "largeimages" alt="timezone">
            <div class="description">
                <h2>Time-zone view</h2>
                <p>Time zone tool in the scheduling application helps users to schedule meetings with people in other countries without any con
                The user can have multiple time zones in the menu based on the countries where the user has most of their meetings which will h
                to compare time zones easily.</p>
            </div>
        </div>
        <div class = "largeimage">
            <img id="large-image-3" src="appointment.jpg" class = "largeimages" alt="appointment">
            <div class="description">
                <h2>Appointments scheduler</h2>
                <p>Appointments scheduler is an tool where it is used to manage and set up appointments with other people and organizations.
                .The user should select all the free dates and timeslots initially, and when the user adds events to his
                calender,it will automatically set a time and date time based on the priority of the event.
            </div>
        </div>
        <div class = "largeimage">
            <img id="large-image-4" src="Notifications.jpeg" class = "largeimages" alt="Notifications">
            <div class="description">
                <h2>Notifier</h2>
                <p> Notifier is a tool where the app sends notifications and alerts about upcoming events. This will send special instructions
                and stay on time therefore, reduce the risk of missing and postponing appointments.</p>
            </div>
        </div>
        <div class = "largeimage">
            <img id="large-image-5" src="analyse.jpeg" class = "largeimages" alt="analyse">
            <div class="description">
                <h2>Analytics and reports</h2>
                <p>Analytics and reporting in a scheduling application refer to the feature that provides users with detailed information about
                </p>
            </div>
        </div>
      </div>
    </div>
    <script src = "galleryscript.js"></script>
</body>
```

Small images are flexed in the horizontal direction and spaced evenly. Height and Width of the small images are set to 200px. The boarder of the image is set to white, and thickness is set to 5px. When the curser is placed on the small image transition CSS property is used to scale the small image to 1.2. The container of the large image is flexed and spaced evenly. The height and width of the large images are set to 400px and a margin of 100px is used for the top in order to have space between large images. Description container is flexed in the vertical direction and it's centered. Its width is set to 50% of the page and it's floated to the right. The margin of the paragraph is set to 100px from the top to have space from the heading tag.



```css
#gallery {
    margin-bottom : 100px;
}
.smallimages {
    height : 200px;
    width : 200px;
    border: 5px solid white;
    transition: transform 0.2s ease-in-out;
}

.smallimages:hover {
  transform: scale(1.2);
}

#thumbnails {
    margin-top : 100px;
    display : flex;                                 /*Body gets divided into two directions*/
    justify-content : space-evenly;                 /*Equal space between content*/
}

.largeimage {
    display : flex;
    justify-content : space-evenly;
}

.largeimages {
    margin-top : 100px;
    height : 400px;
    width : 400px;
}

.description {
    width: 50%;                           /*Width is set to 50% from the container space*/
    float: right;
    display : flex;
    flex-direction : column;              /*Elements are set in the vertical direction*/
    justify-content: center;              /*Equal space between content*/
    align-items: center;
}

p {
    margin-top : 30px;

}
```

A function is created when named scrollToLargeImage and the index of the image is taken as the parameter where a variable is created named large image which selects the large image and scrollIntoView is used to scroll to the large image.



```javascript
function scrollToLargeImage(index) {                                /*Function to scroll to the large image*/
  const largeImage = document.querySelector(`#large-image-${index}`);
  largeImage.scrollIntoView({ behavior: 'smooth' });
}
```

Student 2
In creating the main page, firstly I focused on making the navigation bar. First I defined the header and navigation tags, then add the logo using div element. Then by taking an unordered list i gave each list item an anchor tag with a href attribute that links to the corresponding page. Also an active class to show which page the user was in.

```
<header>
    <nav>

      <div class="heading">
        <h1>ScheduleIt</h1>

      </div>

      <ul class="nav-links">
        <li><a class="active" href="mainPage.html">Home</a></li>
        <li><a href="cart.html">Shop</a></li>
        <li><a href="registrationform.html">Register</a></li>
        <li><a href="newsletter.html">Newsletter</a></li>
        <li><a href="FormSheet.html">Feedback</a></li>
        <li><a href="bestdayspage.html">BestDays</a></li>
        <li><a href="FormSheet.html">Gallery</a></li>
        <li><a href="Quiz.html">Quiz</a></li>
        <li><a href="pages/about.html">About</a></li>
      </ul>
    </nav>
  </header>
```

For the styling of the navigation bar I used an external CSS file which was used by the other members.

```css
* {

    margin: 0px;

    padding: 0px;

    box-sizing: border-box;

    font-family: sans-serif;

  }

  body {
    background: linear-gradient(to right, #0f2027, #203a43, #2c5364);
    font-family: 'Poppins', sans-serif;
}

  .body-text {

    display: flex;

    color: #ccc;

    font-family: "Montserrat", sans-serif;

    align-items: center;

    justify-content: center;

    margin-top: 250px;

  }

  nav {

    display: flex;

    justify-content: space-around;

    align-items: center;

    min-height: 8vh;

    background-color: teal;

    font-family: "Montserrat", sans-serif;

  }

  .heading {

    color: white;

    letter-spacing: 5px;

    font-size: 20px;

  }

  .heading img {
    width: 600px;
max-width: 50%;
   height: 100px;
   font-size: 25px;
 }


 .nav-links {

   display: flex;

          justify-content: space-around;

          width: 30%;

        }

        .nav-links li {

          list-style: none;

        }

        .nav-links a {

          color: white;

          text-decoration: none;

          letter-spacing: 3px;

          font-weight: bold;

          font-size: 14px;

          padding: 14px 16px;

        }

        .nav-links a:hover:not(.active) {

          background-color: lightseagreen;

        }

        .nav-links li a.active {
          background-color: #4caf50;
        }
```

For the form , I made a div class called container and made fields for each necessary input and on submit made the form go thru a javascript validation. If some fields were not filled or incorrect formats were caught and the user is alerted using the alert function.

```html
<div class = "container">
    <!--on clicking the submit button the validationform is called-->
    <form id="myForm" name = "myForm" onsubmit="return validationForm()">
      <h1>Feedback Form</h1><br>
      <!--name field-->
        <div class ="input control">
          <label for="name">Name:</label>
          <input type="text" id="name" name="name"><br>
          <div class="error"></div>
              </div>
              <!--email field-->
              <div class="input-control">
                <label for="email">Email:</label>
                <input type="text" id="email" name="email"><br>
                <div class="error"></div>
              </div>
              <!--Comments-->
              <div class="input-control">
                <label for="comments">Comments:</label><br>
                <textarea id="comments" name="comments" rows="5"
cols="50"></textarea><br><br><br>
                <div class="error"></div>
              </div>
              <!--rating-->
            <label for="rating">Rating:</label>
            <!--allow the user to choose a rating from a dropdown list-->
            <select id="rating" name="rating">
              <option value="">Select a rating</option>
              <option value="1">Poor</option>
              <option value="2">Unsatisfactory</option>
              <option value="3">Satisfactory</option>
              <option value="4">Very Satisfactory</option>
              <option value="5">Outstanding</option>
            </select><br><br>
            <!--submit button that will submit the form data when clicked-->
            <input type="submit" value="Submit">
            <!--button that will reset the form data when clicked-->
            <input type="button" value="Reset" onclick="resetForm()">
            <!--take the user back to the main page when clicked-->
            <input type="button" value="Back to Main Page"
onclick="goToMainPage()">
            </div>
          </form>
```

```
<script>
        function validationForm() {
        //retrieves the input values from the fields
        var name = document.forms["myForm"]["name"].value;
        var email = document.forms["myForm"]["email"].value;
        var rating = document.forms["myForm"]["rating"].value;
        //if name field is empty the user is alerted
        if (name == "") {
          alert("Name must be filled out");
                return false;
              }
              //if email field is empty user is alerted
              if (email == "") {
                alert("Email must be filled out");
                return false;
              } else {
                //else if the email format is wrong the user gets alerted
                var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
                if (!emailRegex.test(email)) {
                  alert("Invalid email format");
                  return false;
                }
              }
              //if rating isnt chosen user gets alerted
              if (rating == "") {
                alert("Rating must be filled out");
                return false;
              }
            }

              function resetForm() {
                document.getElementById("myForm").reset();
              }

              function goToMainPage() {
                window.location.href = "mainPage.html";
              }

          </script>
```

For the quiz, I used JavaScript to create a timer, multiple choice questions and a score with a badge for the user after finishing the quiz. First a div element was made called display-container which carried the questions. Upon clicking the start button the quiz gets started, with a timer counting down which when it is over will move to the next question.

```
let timeLeft = document.querySelector(".time-left");
let quizContainer = document.getElementById("container");
let nextButton = document.getElementById("next-button");
let countOfQuestion = document.querySelector(".number-of-question");
let displayContainer = document.getElementById("display-container");
let scoreContainer = document.querySelector(".score-container");
let restart = document.getElementById("restart");
let userScore = document.getElementById("user-score");
let startScreen = document.querySelector(".start-screen");
let startButton = document.getElementById("start-button");
let navbar = document.getElementsByClassName("nav-links");
let questionCount;
let scoreCount = 0;
let count = 11;
let countdown;

//contains the questions and answers of the quiz which is kept in an array
const quizArray = [
    {
        id: "0",
        question: "What are the key features of a scheduling assistant application, and how do they simplify the scheduling process for users?",
        options: [
        "Automated appointment scheduling, calendar syncing, and reminders",
        "Advanced analytics and reporting, customizable scheduling workflows",
        "Integration with social media platforms, personalized chatbot assistance",
        "Virtual reality meeting rooms, AI-powered scheduling assistants",
        ],
        correct: "Automated appointment scheduling, calendar syncing, and reminders",
    },
    {
        id: "1",
        question: "Can a scheduling assistant application sync with a user's calendar and send reminders for upcoming appointments and meetings?",
        options: [
        "No, scheduling assistant applications can only schedule appointments but cannot sync with calendars or send reminders",
        "Yes, most scheduling assistant applications can sync with popular calendar applications and send reminders via email, SMS, or push notifications",
        "Only some scheduling assistant applications offer calendar syncing and reminders, depending on the pricing plan",
        "Scheduling assistant applications cannot send reminders but can sync with calendars to display upcoming appointments within the application",
        ],
        correct: "Yes, most scheduling assistant applications can sync with popular calendar applications and send reminders via email, SMS, or push notifications"
    },
    {
        id: "2",
        question: "How does a scheduling assistant application handle conflicts or overlaps in scheduling, and can it suggest alternative times or dates?",
        options: [
        "Scheduling assistant applications cannot handle conflicts or overlaps in scheduling, and users must manually resolve any issues",
        "Most scheduling assistant applications offer automated conflict resolution and suggest alternative times or dates based on user preferences",
        "Conflict resolution and alternative suggestions are only available in premium versions of scheduling assistant applications",
        "Scheduling assistant applications can handle conflicts but do not offer suggestions for alternative times or dates",
        ],
correct: "Most scheduling assistant applications offer automated conflict resolution and suggest alternative times or dates based on user preferences",
    },
    {
        id: "3",
        question: "Are there any privacy or security concerns associated with using a scheduling assistant application, and how are these addressed by the developers?",
        options: [
        "There are no privacy or security concerns associated with using a scheduling assistant application",
        "Developers of scheduling assistant applications adhere to strict privacy and security policies and encrypt user data",
        "Privacy and security concerns vary depending on the application and its developers, and users should research each application before use",
        "Scheduling assistant applications have access to sensitive user data but take necessary measures to protect it, such as two-factor authentication and access controls",
        ],
        correct: "Developers of scheduling assistant applications adhere to strict privacy and security policies and encrypt user data",
    },
    {
        id: "4",
        question: "What are some examples of industries or professions that can benefit from using a scheduling assistant application, and how have these applications improved their productivity or efficiency?",
        options: [
        "Healthcare, education, and legal professions have seen significant productivity improvements from using scheduling assistant applications",
        "Only large corporations with complex scheduling needs benefit from using scheduling assistant applications",
        "Scheduling assistant applications have not shown to improve productivity or efficiency in any industry or profession",
        "Scheduling assistant applications are only useful for scheduling personal appointments and not for professional use",
        ],
        correct: "Healthcare, education, and legal professions have seen significant productivity improvements from using scheduling assistant applications",
    },

];
//waits for the restart button is clicked
restart.addEventListener("click", () => {
    initial();
    //shows the quiz again by removing hide
    displayContainer.classList.remove("hide");
    //the restart button is hidden
    scoreContainer.classList.add("hide");
});
//goes to next question
nextButton.addEventListener("click", (displayNext = () => {
    //tracks question count
    questionCount += 1;
    //checks whether all questions are over
    if (questionCount == quizArray.length){
        //quiz gets hidden
        displayContainer.classList.add("hide");
        //pop up window of score is shown with the restart button
        scoreContainer.classList.remove("hide");
        //pop up window properties
        let popupWidth = 600;
        let popupHeight = 400;
        let left = (screen.width / 2) - (popupWidth / 2);
        let top = (screen.height / 2) - (popupHeight / 2);
        let popupWindow = window.open('', 'Quiz Results', 'height=' + popupHeight + ',width=' + popupWidth + ',left=' + left + ',top=' + top);
        let resultImg;
        let resultText;
        //gives the badges and scores accordingly
        if (scoreCount >= 8){
            resultImg = "gold.png";
            resultText = "Congratulations! You scored " + scoreCount + " out of 10, please claim the points in your next purchase";
        }
        if (scoreCount < 8 && scoreCount >=6){
            resultImg = "silver.png";
            resultText = "Good job! You scored " + scoreCount + " out of 10, please claim the points in your next purchase";
        }
        if (scoreCount < 6){
            resultImg = "bronze.png";
            resultText = "Better luck next time! You scored " + scoreCount + " out of 10, please claim the points in your next purchase";
        }
        //set title
        let htmlContent = "<html><head><title>Quiz Results</title></head>";
        //showing the score and badge
        htmlContent += "<body style='text-align: center;'><h1>Quiz Results</h1>";
        htmlContent += "<img src='" + resultImg + "' alt='' style='width: 100px;'><br>";
        htmlContent += "<p>" + resultText + "</p>";
        htmlContent += "</body></html>";

        popupWindow.document.write(htmlContent);
        popupWindow.focus();
    }
    else{
        //if there are still questions remaining
        countOfQuestion.innerHTML = questionCount + 1 + " of " + quizArray.length + " Question";
        //displaying the question and answer options for the current question
        quizDisplay(questionCount);
        count = 11;
        //stops the timer
        clearInterval(countdown);
        timerDisplay();
    }

})
);
//creates the timer
const timerDisplay = () => {
    //sets the setinterval() to execute a function every second
    countdown = setInterval(() => {
        //the function decreases the count by 1
        count--;
        //shows the time left
        timeLeft.innerHTML = `${count}s`;
        //when the count reaches 0 meaning time is over the next question is shown
        if (count == 0) {
```

```javascript
clearInterval(countdown);
            displayNext();
        }
    }, 1000);
};
//questioncount is the index of the current quiz card that shd be displayed
const quizDisplay = (questionCount) => {
    //array of all elements in container-mid
    let quizCards = document.querySelectorAll(".container-mid");
    //a card in quizCards
    quizCards.forEach((card) => {
        //hides the cards
        card.classList.add("hide");
    });
    //hide is removed when the quiz card needs to be shown
    quizCards[questionCount].classList.remove("hide");
};
//
function quizCreater(){
    //questions and options appear in a random order each time the quiz is taken
    quizArray.sort(() => Math.random() - 0.5);

    for(let i of quizArray){
        //shuffles the options for each question in the quizArray so that they appear in a random order each time the quiz is taken
        i.options.sort(() => Math.random() - 0.5);
        //creates a new div element that will hold the question and options
        let div = document.createElement("div");
        //sets the display to none so that the div is initially hidden
        div.classList.add("container-mid", "hide");
        //shows the question user is on
        countOfQuestion.innerHTML = 1 + " of " + quizArray.length + " Question";
        //new p element to hold the question
        let question_DIV = document.createElement("p");
        //question is css class
        question_DIV.classList.add("question");
        //innerHTML of the question_DIV element is set to the question property of the current object in the quizArray
        question_DIV.innerHTML = i.question;
        div.appendChild(question_DIV);
        //the checker function gets called when one of the buttons is clicked
        //innerHTML of each button element is set to one of the options for the current question in the quizArray
        div.innerHTML += `
        <button class="option-div" onclick="checker(this)">
        ${i.options[0]}</button>
        <button class="option-div" onclick="checker(this)">
        ${i.options[1]}</button>
        <button class="option-div" onclick="checker(this)">
        ${i.options[2]}</button>
        <button class="option-div" onclick="checker(this)">
        ${i.options[3]}</button>
        `;
        //current question and options visible on the webpage
        quizContainer.appendChild(div);
    }
}
```

```javascript
//checks user option
function checker(userOption){
    //assigns it the inner text of the userOption button clicked by the user
    let userSolution = userOption.innerText;
    //creates a variable called question and assigns it the HTML element with class container-mid at the index questionCount.
    let question = document.getElementsByClassName("container-mid")[questionCount];
    //assigns it an array of HTML elements with class option-div found within the question element
    let options = question.querySelectorAll(".option-div");
    //checks if userSolution matches the correct answer for the current question in quizArray.
    if(userSolution === quizArray[questionCount].correct){
        //If the user's answer is correct, the correct class is added to the userOption button clicked
        userOption.classList.add("correct");
        scoreCount += 2;

    }else{
        //incorrect class is added to the userOption button clicked by the user
        userOption.classList.add("incorrect");
        scoreCount -=1;
        //stops the score from being negative
        if (scoreCount <= 0){
            scoreCount = 0;
        }

        options.forEach((element) => {
            //If the text of the current option matches the correct answer for the current question in quizArray
            if (element.innerText === quizArray[questionCount].correct){
                //correct class is added to the current option button
                element.classList.add("correct");
            }
        });
    }
    //timer for the current question is stopped
    clearInterval(countdown);
    //All the buttons in options are disabled to prevent the user from changing their answer
    options.forEach((element) => {
        element.disabled = true;
    });
}
//making everything to the initial values
function initial(){
    quizContainer.innerHTML = "";
    questionCount = 0;
    scoreCount = 0;
    count = 11;
    clearInterval(countdown);
    timeLeft.innerHTML = ""
    timerDisplay();
    quizCreater();
    quizDisplay(questionCount);
}
//sets up an event listener for the click event on the start button
startButton.addEventListener("click", () => {
    //adds the hide class to the start screen element, which hides it from view
    startScreen.classList.add("hide");
    //removes the hide class from the display container element,which shows it
displayContainer.classList.remove("hide");
    initial();
});
//sets up an event listener for when the window loads
window.onload = () => {
    //removes the hide class from the start screen element, which shows it.
    startScreen.classList.remove("hide");
    //adds the hide class to the display container element, which hides it from view
    displayContainer.classList.add("hide");
};
```

Student 3

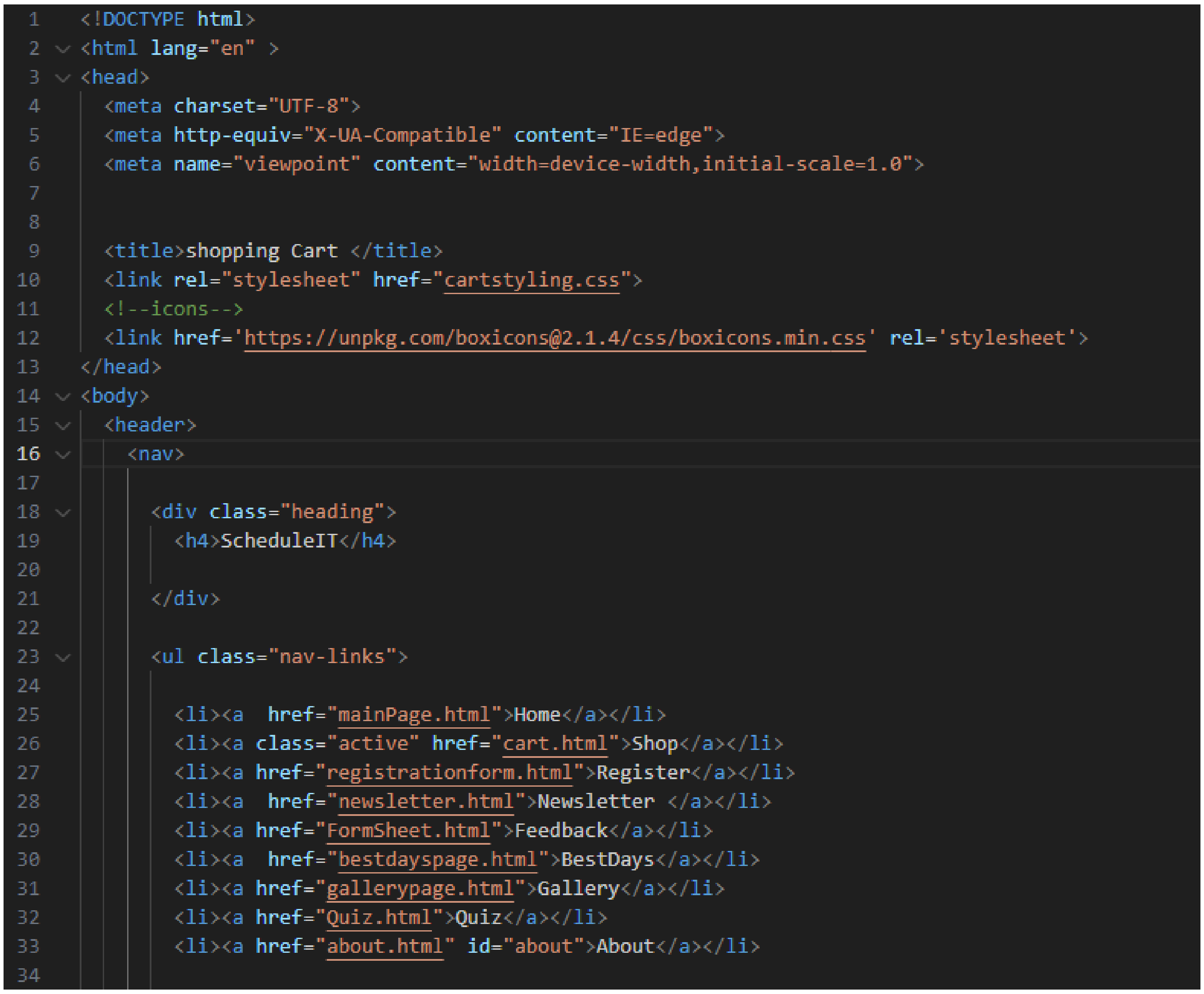
The tasks allocated for student 3 are, to develop the cart with the use of javascript for a fully functional shopping webpage to buy the services provided by the website by the user and to develop the about us page to display information about our team.

[I.] <u>Cart.</u>

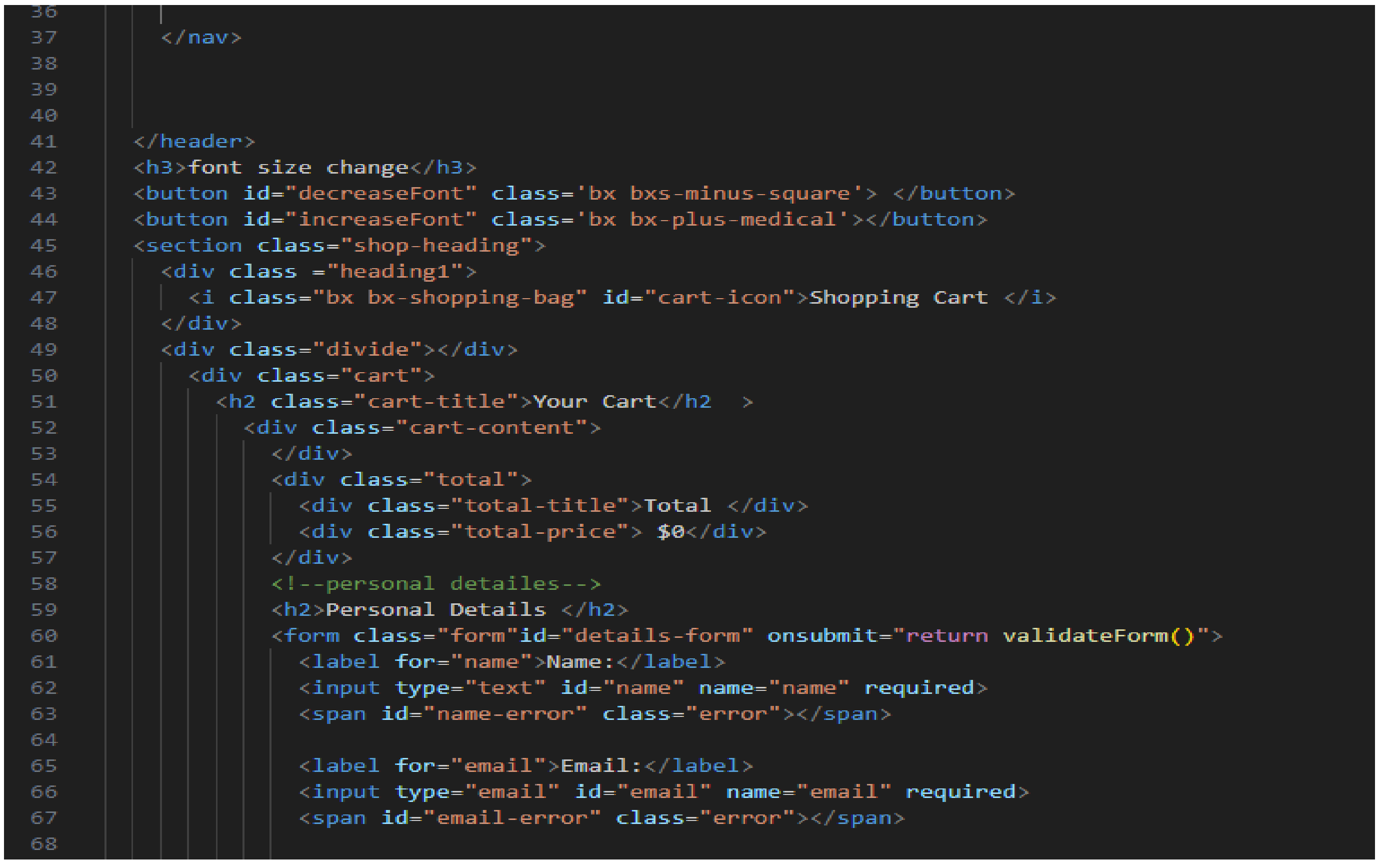
The website we developed was based on an AI scheduling application for users. This diverts under a service-providing website. So literally the shop contains the services to be purchased.

- As per html was coded for the cart site. Included appropriate images for relevant scheduling services that we provide and used css for positioning, padding coloring purposes.
- Each item listing is done within a main div tag. Inside it relevant listings are placed with image tags headings. Same goes for every item.
- For the use of javascript and css each elemt in the html is added with classes and ids.

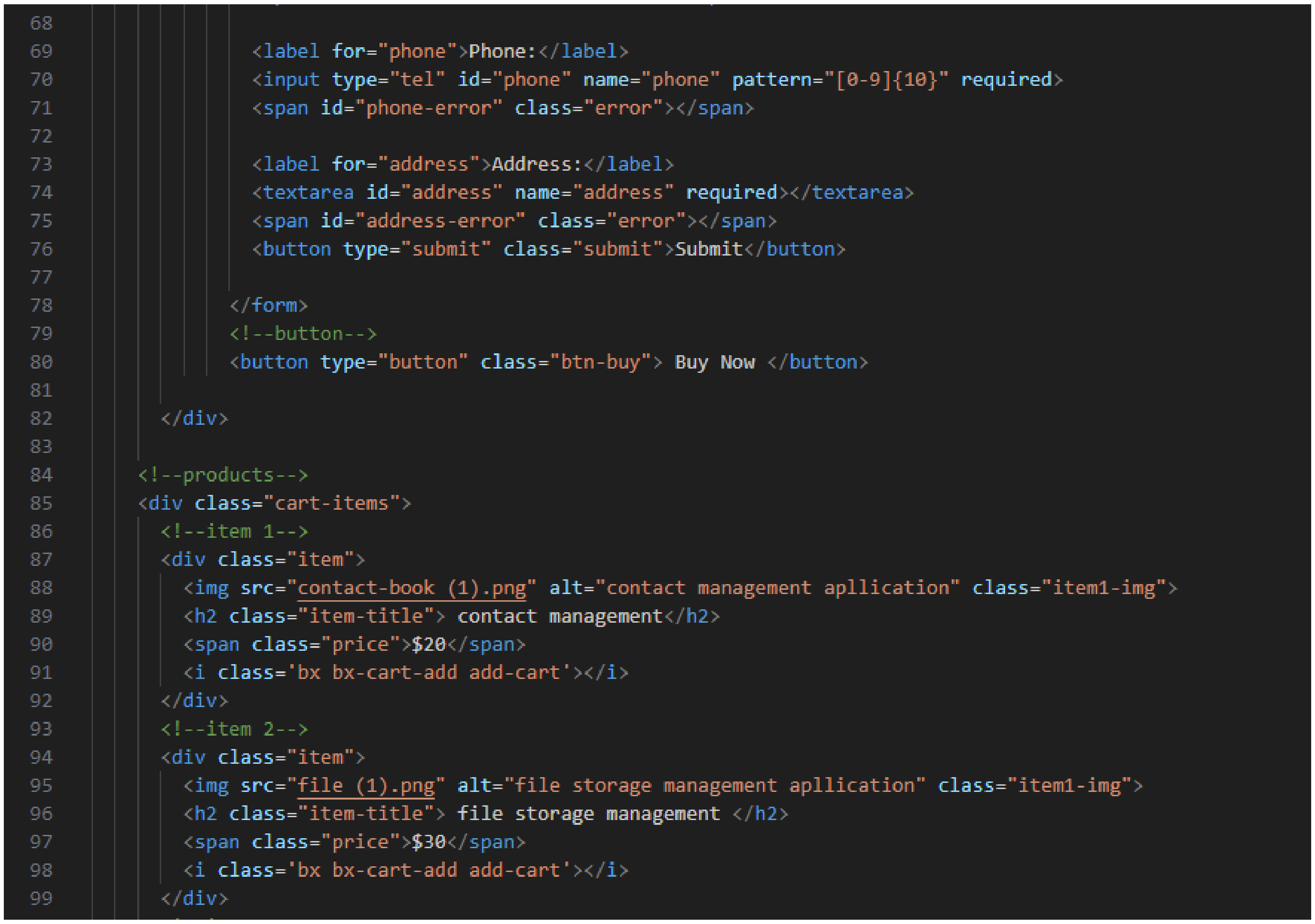
Cart.html



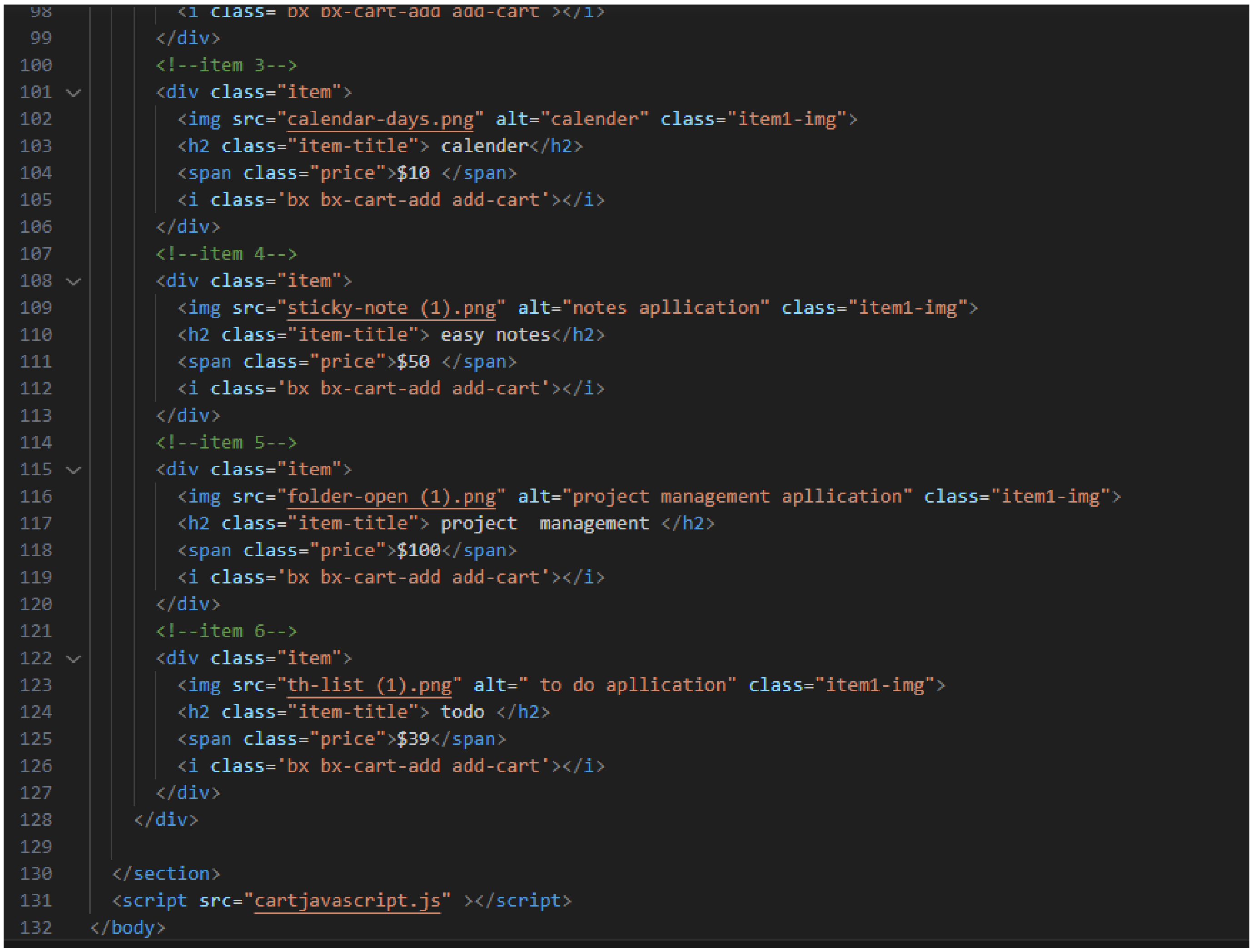
```html
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewpoint" content="width=device-width,initial-scale=1.0">


  <title>shopping Cart </title>
  <link rel="stylesheet" href="cartstyling.css">
  <!--icons-->
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
</head>
<body>
  <header>
    <nav>

      <div class="heading">
        <h4>ScheduleIT</h4>

      </div>

      <ul class="nav-links">

        <li><a  href="mainPage.html">Home</a></li>
        <li><a class="active" href="cart.html">Shop</a></li>
        <li><a href="registrationform.html">Register</a></li>
        <li><a  href="newsletter.html">Newsletter </a></li>
        <li><a href="FormSheet.html">Feedback</a></li>
        <li><a  href="bestdayspage.html">BestDays</a></li>
        <li><a href="gallerypage.html">Gallery</a></li>
        <li><a href="Quiz.html">Quiz</a></li>
        <li><a href="about.html" id="about">About</a></li>

```



```html
      </nav>



    </header>
    <h3>font size change</h3>
    <button id="decreaseFont" class='bx bxs-minus-square'> </button>
    <button id="increaseFont" class='bx bx-plus-medical'></button>
    <section class="shop-heading">
      <div class ="heading1">
        <i class="bx bx-shopping-bag" id="cart-icon">Shopping Cart </i>
      </div>
      <div class="divide"></div>
        <div class="cart">
          <h2 class="cart-title">Your Cart</h2  >
            <div class="cart-content">
              </div>
              <div class="total">
                <div class="total-title">Total </div>
                <div class="total-price"> $0</div>
              </div>
              <!--personal detailes-->
              <h2>Personal Details </h2>
              <form class="form"id="details-form" onsubmit="return validateForm()">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required>
                <span id="name-error" class="error"></span>

                <label for="email">Email:</label>
                <input type="email" id="email" name="email" required>
                <span id="email-error" class="error"></span>

```

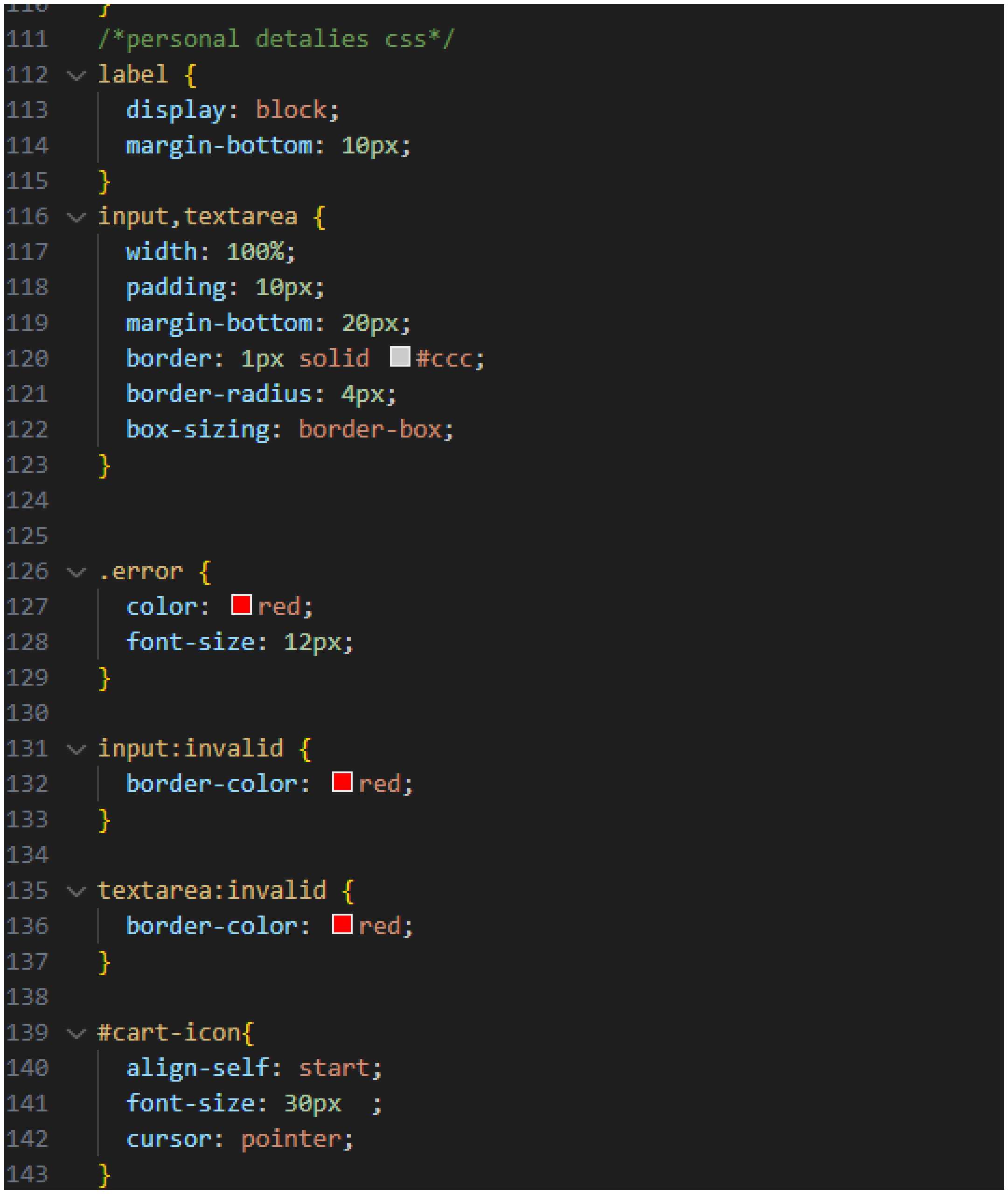


```html

                <label for="phone">Phone:</label>
                <input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required>
                <span id="phone-error" class="error"></span>

                <label for="address">Address:</label>
                <textarea id="address" name="address" required></textarea>
                <span id="address-error" class="error"></span>
                <button type="submit" class="submit">Submit</button>

              </form>
              <!--button-->
              <button type="button" class="btn-buy"> Buy Now </button>

        </div>

      <!--products-->
      <div class="cart-items">
        <!--item 1-->
        <div class="item">
          <img src="contact-book (1).png" alt="contact management apllication" class="item1-img">
          <h2 class="item-title"> contact management</h2>
          <span class="price">$20</span>
          <i class='bx bx-cart-add add-cart'></i>
        </div>
        <!--item 2-->
        <div class="item">
          <img src="file (1).png" alt="file storage management apllication" class="item1-img">
          <h2 class="item-title"> file storage management </h2>
          <span class="price">$30</span>
          <i class='bx bx-cart-add add-cart'></i>
        </div>
```

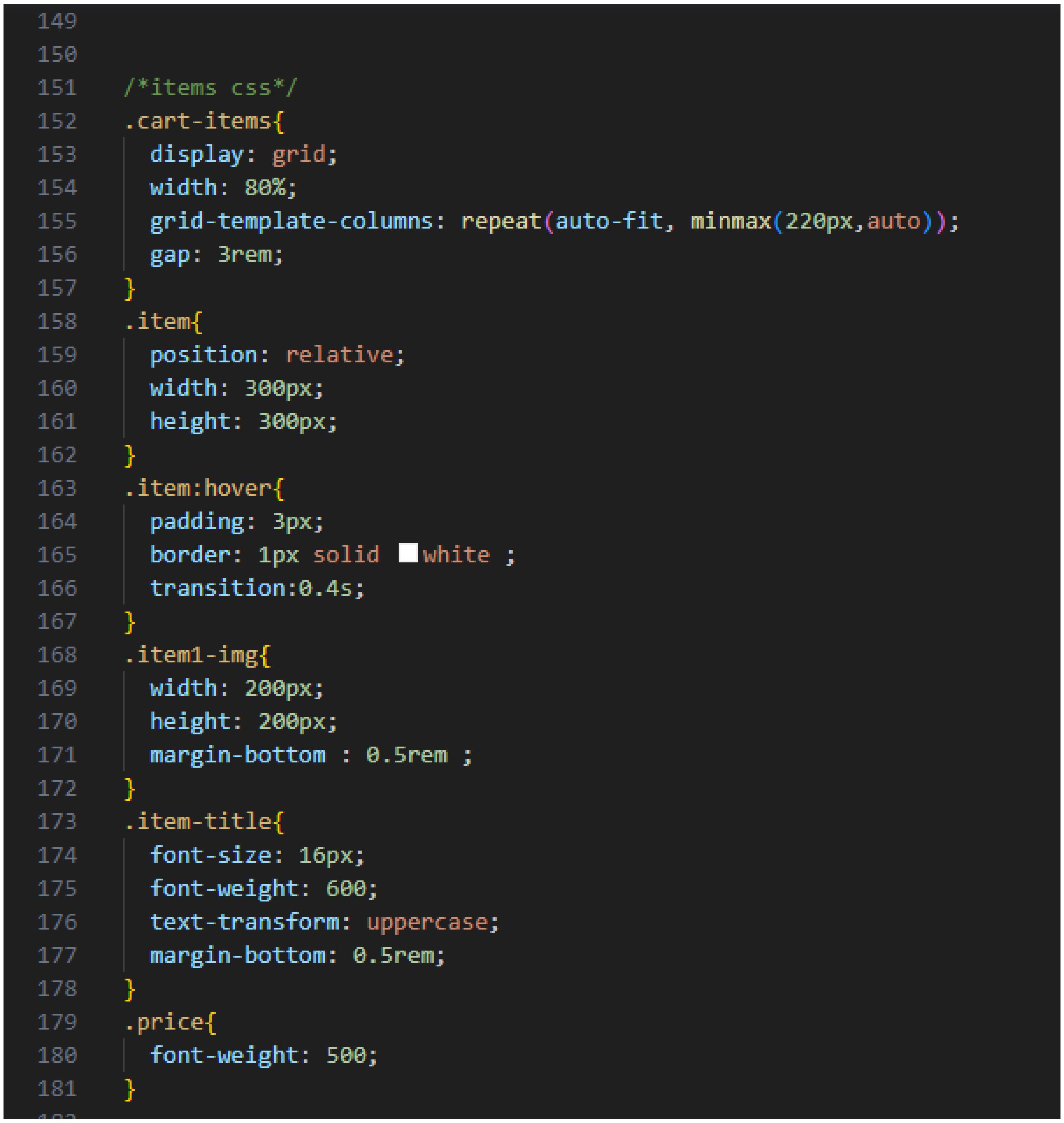


```
        <i class='bx bx-cart-add add-cart'></i>
      </div>
      <!--item 3-->
      <div class="item">
        <img src="calendar-days.png" alt="calender" class="item1-img">
        <h2 class="item-title"> calender</h2>
        <span class="price">$10 </span>
        <i class='bx bx-cart-add add-cart'></i>
      </div>
      <!--item 4-->
      <div class="item">
        <img src="sticky-note (1).png" alt="notes apllication" class="item1-img">
        <h2 class="item-title"> easy notes</h2>
        <span class="price">$50 </span>
        <i class='bx bx-cart-add add-cart'></i>
      </div>
      <!--item 5-->
      <div class="item">
        <img src="folder-open (1).png" alt="project management apllication" class="item1-img">
        <h2 class="item-title"> project  management </h2>
        <span class="price">$100</span>
        <i class='bx bx-cart-add add-cart'></i>
      </div>
      <!--item 6-->
      <div class="item">
        <img src="th-list (1).png" alt=" to do apllication" class="item1-img">
        <h2 class="item-title"> todo </h2>
        <span class="price">$39</span>
        <i class='bx bx-cart-add add-cart'></i>
      </div>
    </div>

  </section>
  <script src="cartjavascript.js" ></script>
</body>
```

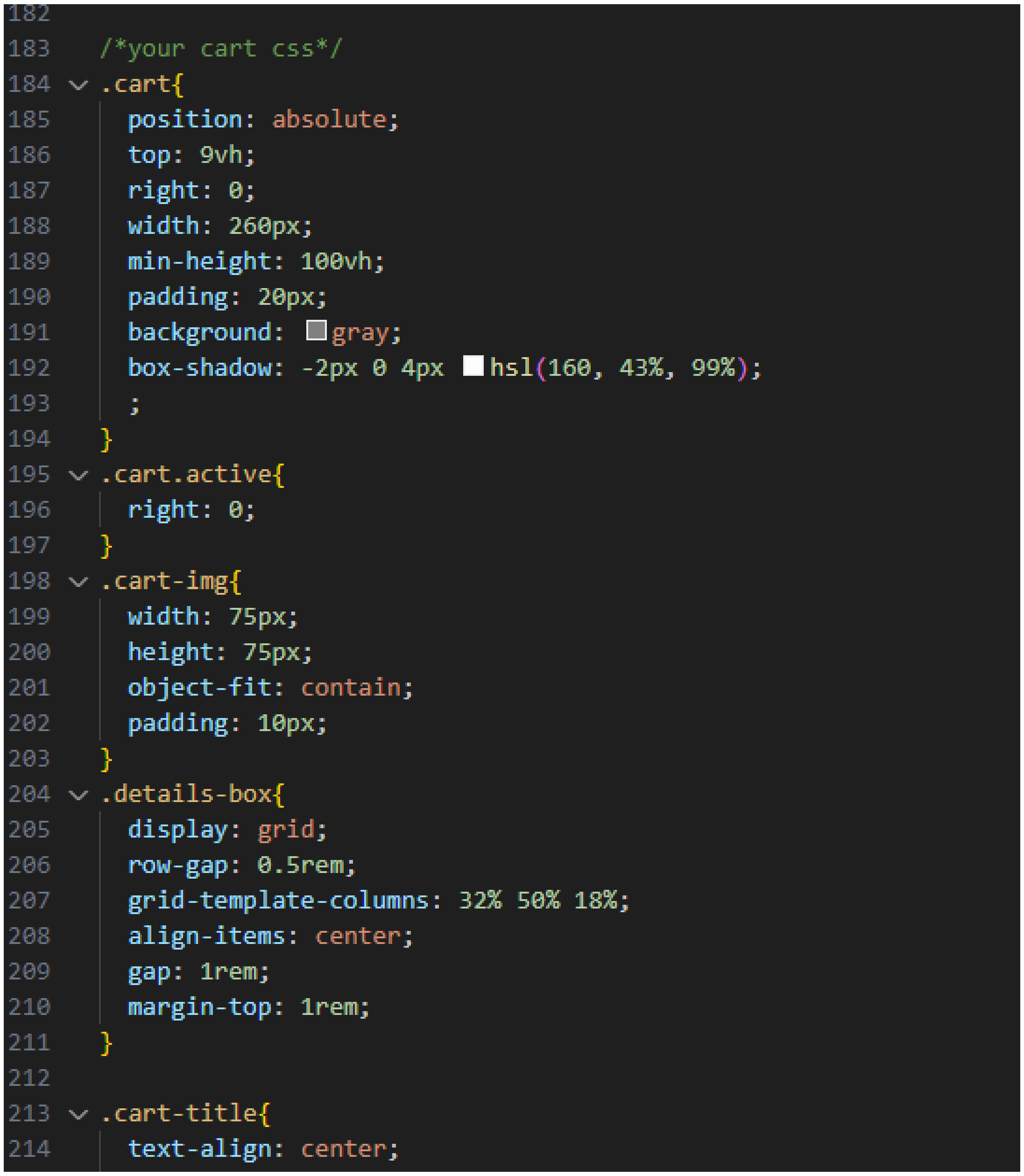
Cart.css



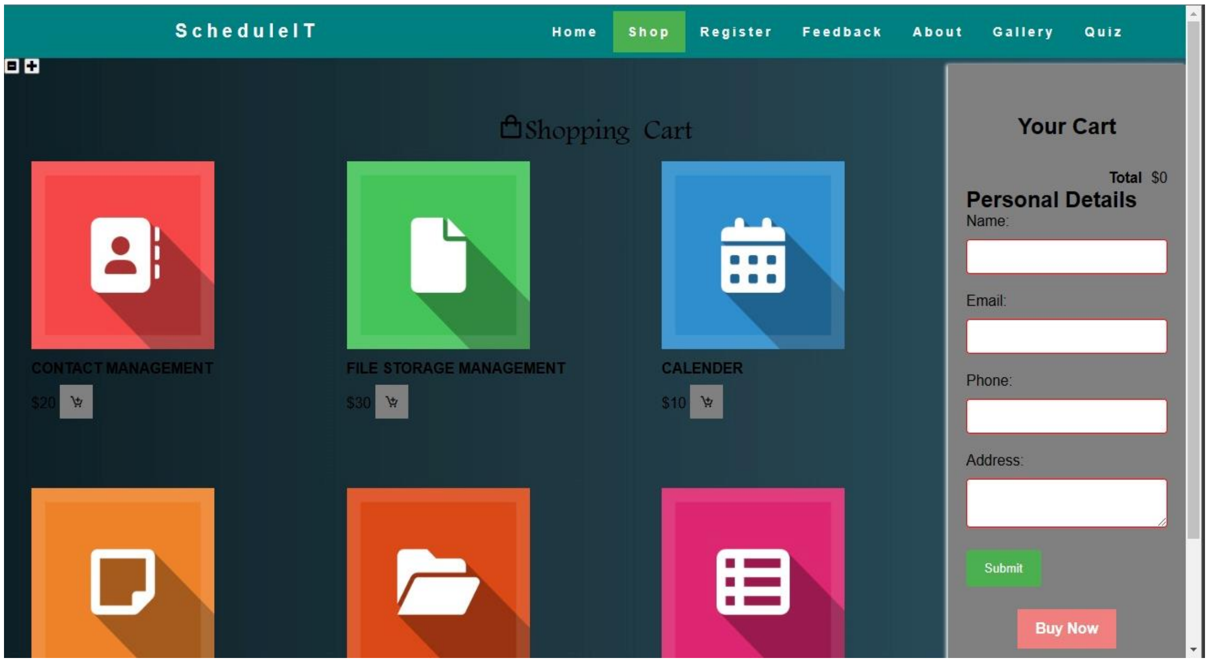
```css
}
/*personal detalies css*/
label {
  display: block;
  margin-bottom: 10px;
}
input,textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}


.error {
  color: red;
  font-size: 12px;
}

input:invalid {
  border-color: red;
}

textarea:invalid {
  border-color: red;
}

#cart-icon{
  align-self: start;
  font-size: 30px  ;
  cursor: pointer;
}
```



```


/*items css*/
.cart-items{
  display: grid;
  width: 80%;
  grid-template-columns: repeat(auto-fit, minmax(220px,auto));
  gap: 3rem;
}
.item{
  position: relative;
  width: 300px;
  height: 300px;
}
.item:hover{
  padding: 3px;
  border: 1px solid white ;
  transition:0.4s;
}
.item1-img{
  width: 200px;
  height: 200px;
  margin-bottom : 0.5rem ;
}
.item-title{
  font-size: 16px;
  font-weight: 600;
  text-transform: uppercase;
  margin-bottom: 0.5rem;
}
.price{
  font-weight: 500;
}
```



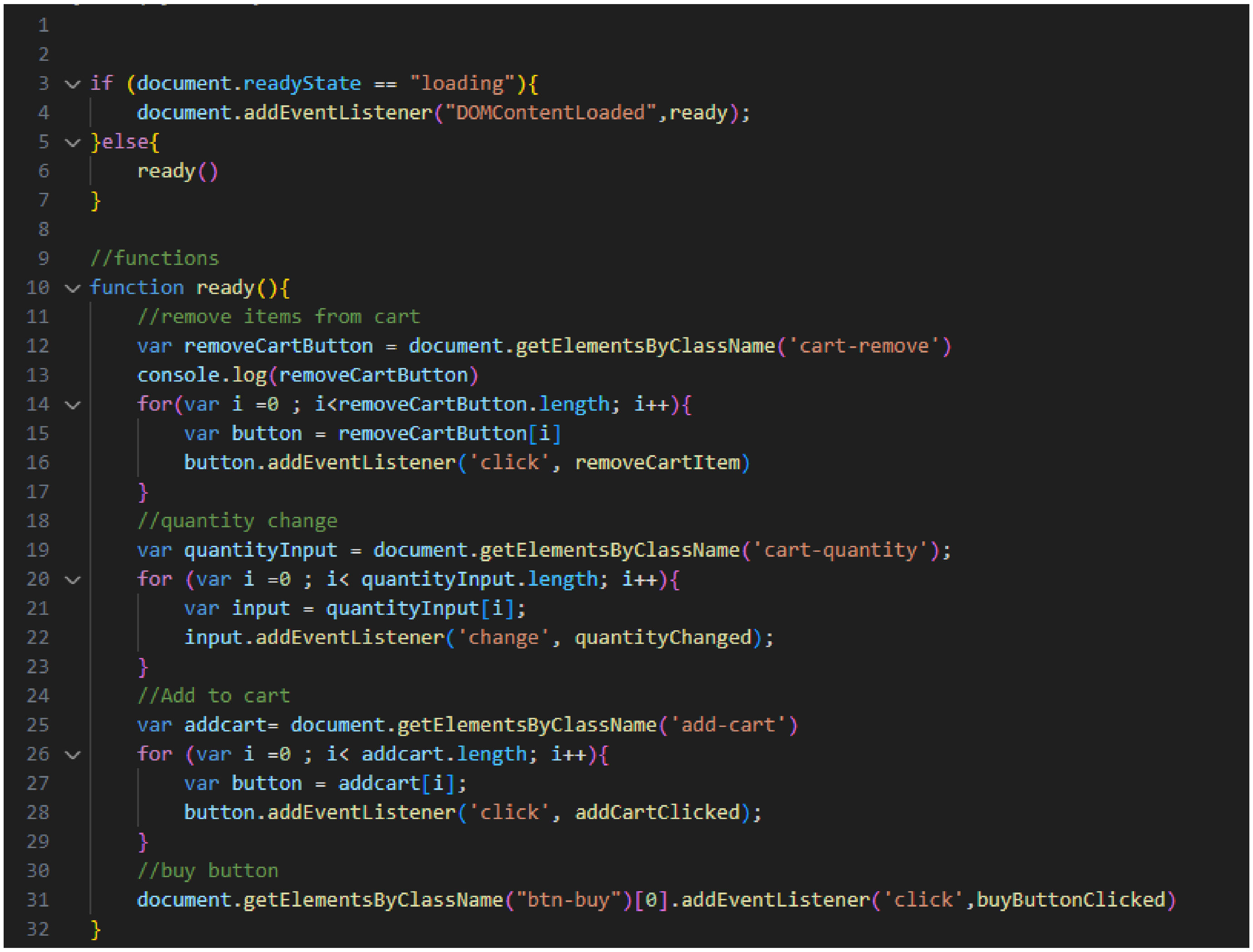
```css

/*your cart css*/
.cart{
  position: absolute;
  top: 9vh;
  right: 0;
  width: 260px;
  min-height: 100vh;
  padding: 20px;
  background: gray;
  box-shadow: -2px 0 4px hsl(160, 43%, 99%);
  ;
}
.cart.active{
  right: 0;
}
.cart-img{
  width: 75px;
  height: 75px;
  object-fit: contain;
  padding: 10px;
}
.details-box{
  display: grid;
  row-gap: 0.5rem;
  grid-template-columns: 32% 50% 18%;
  align-items: center;
  gap: 1rem;
  margin-top: 1rem;
}

.cart-title{
  text-align: center;
```



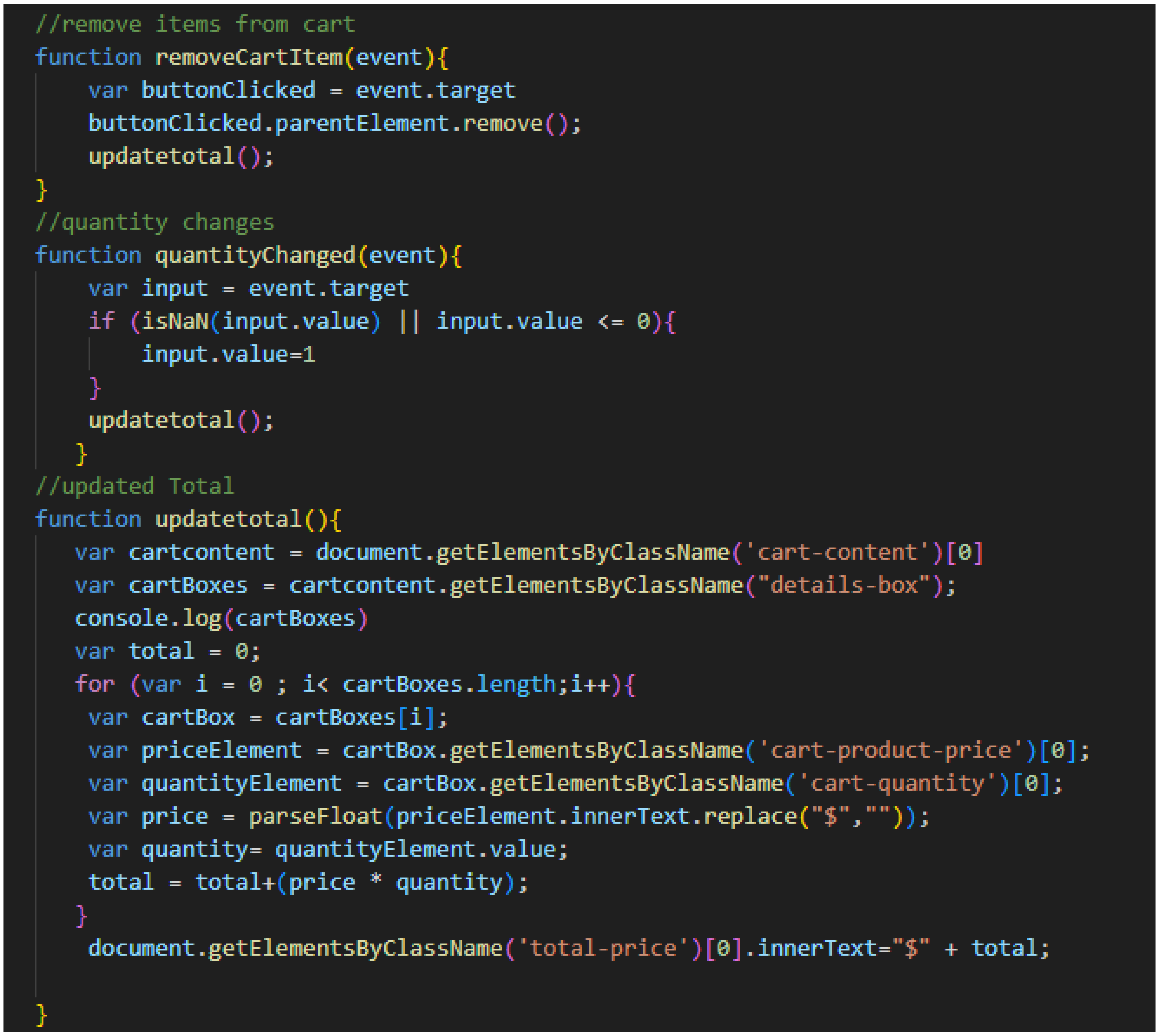
- Included a personal detail form for the user to fill out upon buying the purchases.

Challenge: For this cart site coding the javascript part for user interactivity and functioning parts was challenging. The same was for the personal details part as it needed to be functioning of javascript.

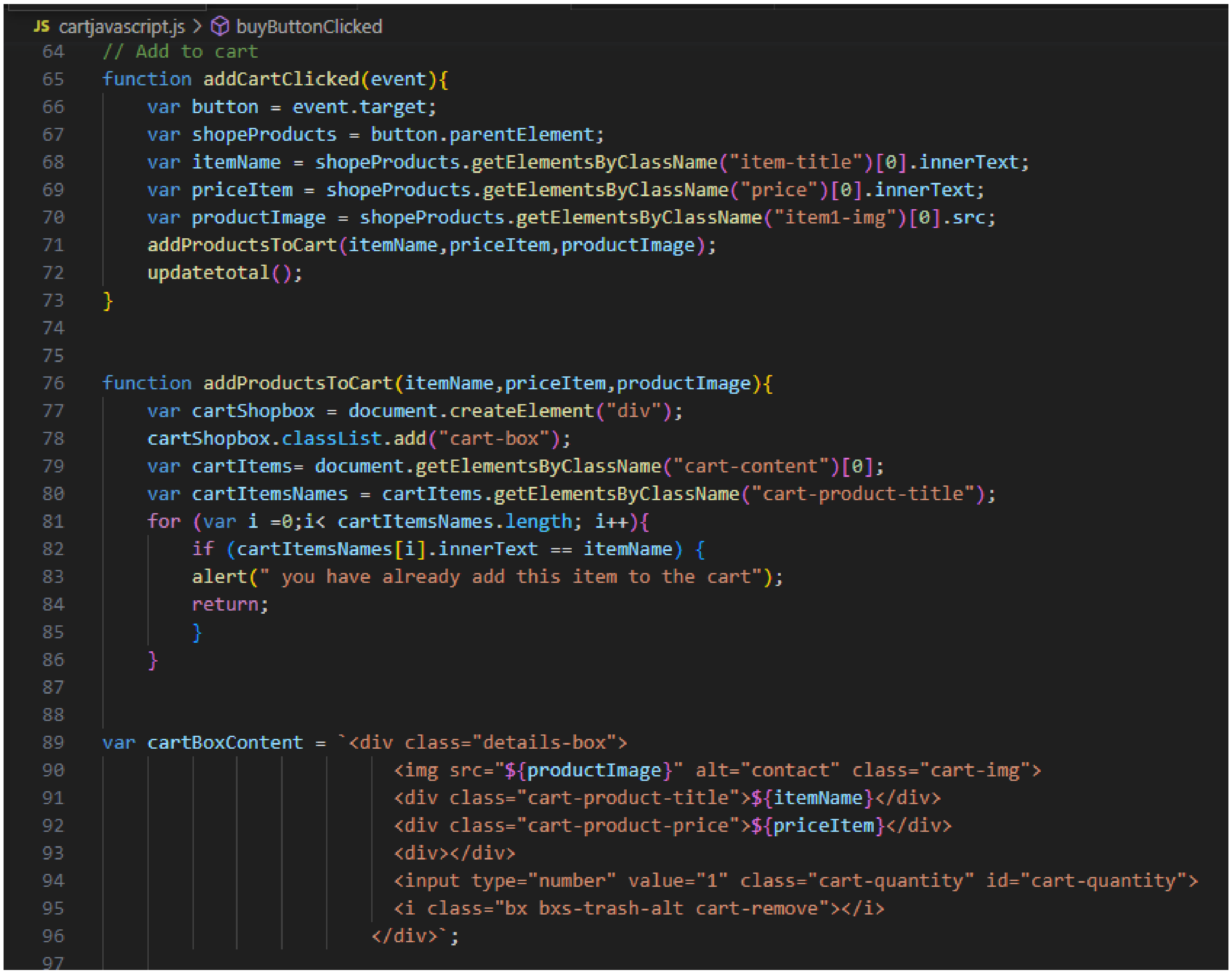
Strategy: Used functions for each part separately and call them on the necessary places. Declared the variable required for varying parts. Conditions like 'if,else'are being used for validity checking points. During looping parts conditions like 'for','while' loops are used inside the functions.



```


if (document.readyState == "loading"){
    document.addEventListener("DOMContentLoaded",ready);
}else{
    ready()
}

//functions
function ready(){
    //remove items from cart
    var removeCartButton = document.getElementsByClassName('cart-remove')
    console.log(removeCartButton)
    for(var i =0 ; i<removeCartButton.length; i++){
        var button = removeCartButton[i]
        button.addEventListener('click', removeCartItem)
    }
    //quantity change
    var quantityInput = document.getElementsByClassName('cart-quantity');
    for (var i =0 ; i< quantityInput.length; i++){
        var input = quantityInput[i];
        input.addEventListener('change', quantityChanged);
    }
    //Add to cart
    var addcart= document.getElementsByClassName('add-cart')
    for (var i =0 ; i< addcart.length; i++){
        var button = addcart[i];
        button.addEventListener('click', addCartClicked);
    }
    //buy button
    document.getElementsByClassName("btn-buy")[0].addEventListener('click',buyButtonClicked)
}
```

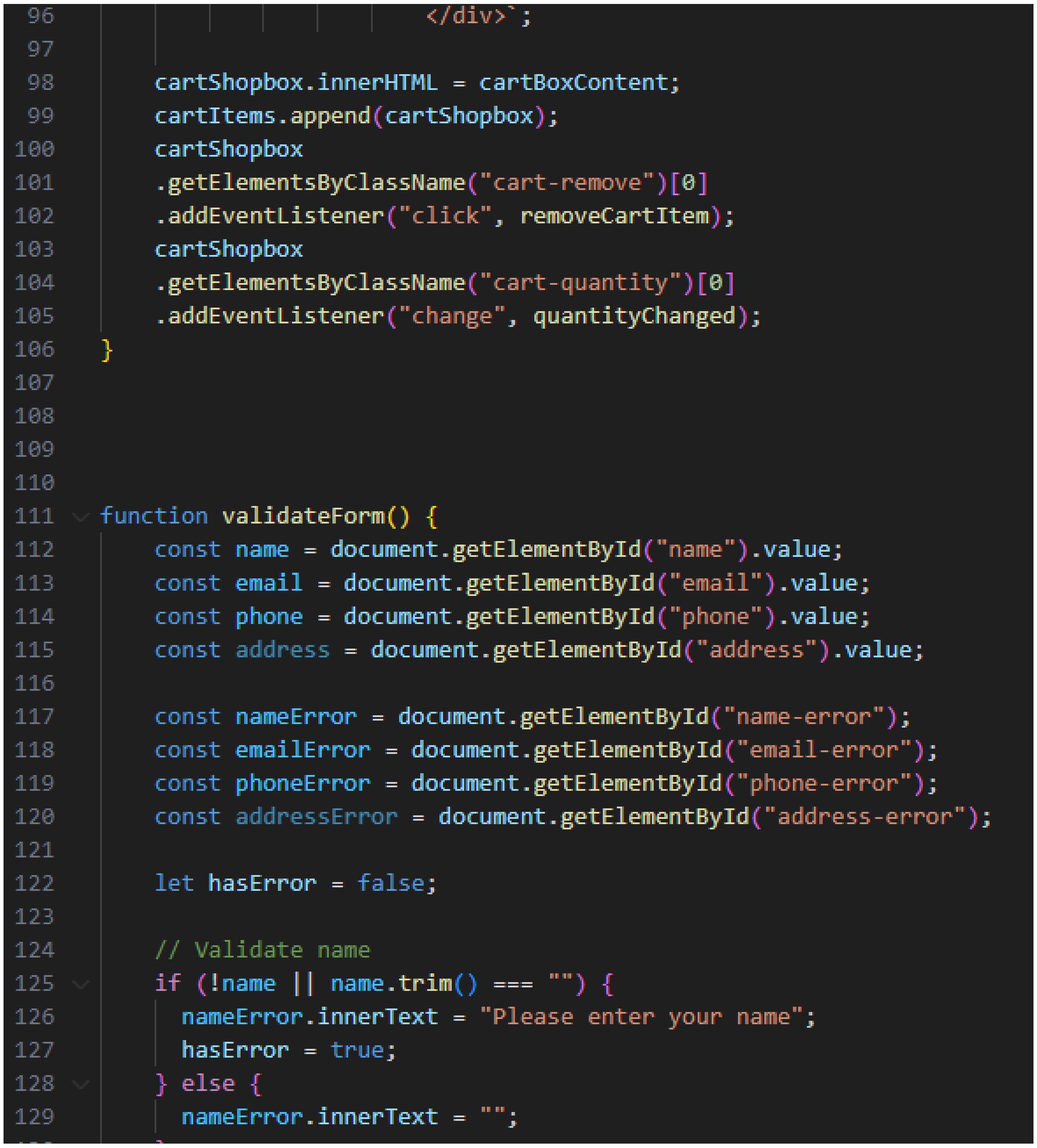


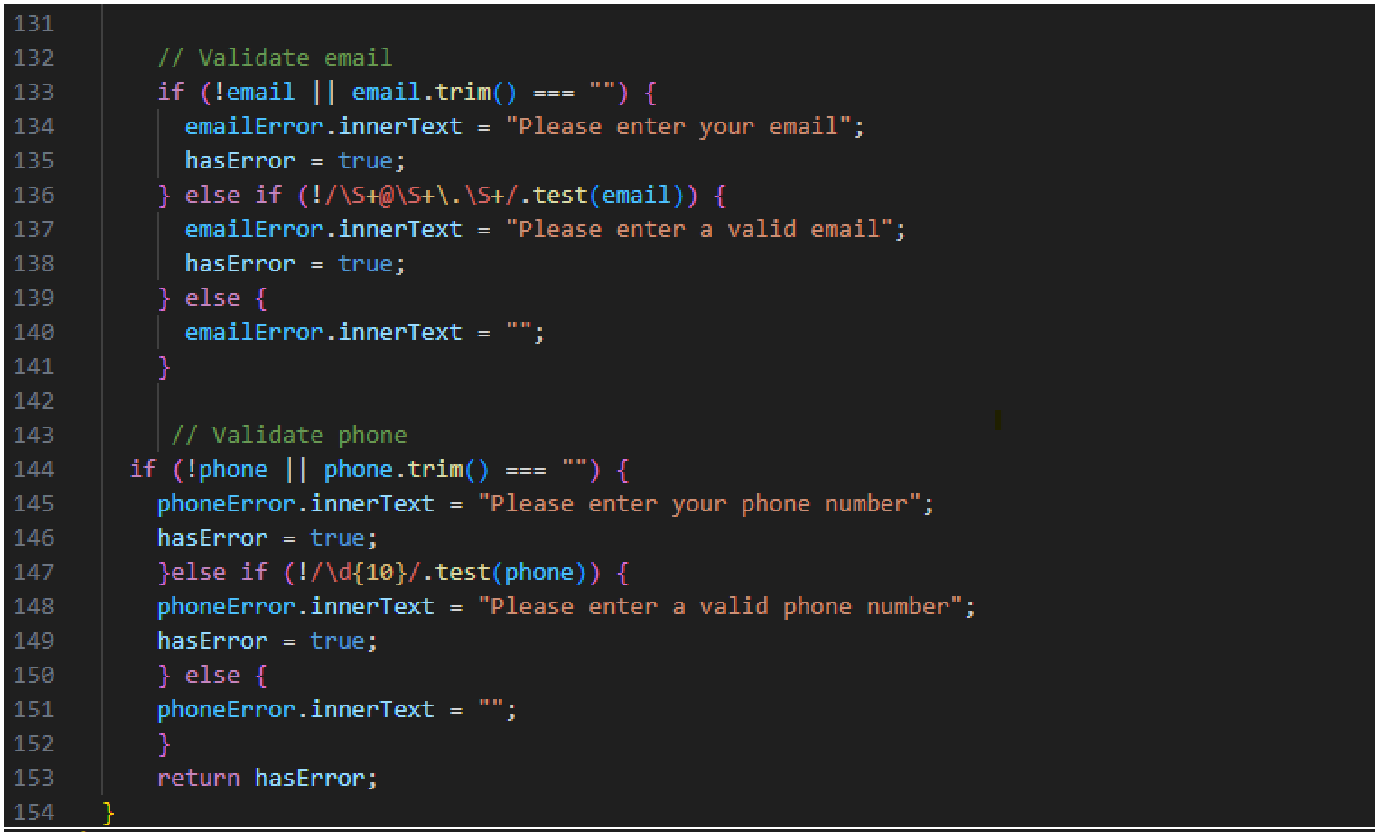
```
//remove items from cart
function removeCartItem(event){
    var buttonClicked = event.target
    buttonClicked.parentElement.remove();
    updatetotal();
}
//quantity changes
function quantityChanged(event){
    var input = event.target
    if (isNaN(input.value) || input.value <= 0){
        input.value=1
    }
    updatetotal();
   }
//updated Total
function updatetotal(){
   var cartcontent = document.getElementsByClassName('cart-content')[0]
   var cartBoxes = cartcontent.getElementsByClassName("details-box");
   console.log(cartBoxes)
   var total = 0;
   for (var i = 0 ; i< cartBoxes.length;i++){
    var cartBox = cartBoxes[i];
    var priceElement = cartBox.getElementsByClassName('cart-product-price')[0];
    var quantityElement = cartBox.getElementsByClassName('cart-quantity')[0];
    var price = parseFloat(priceElement.innerText.replace("$",""));
    var quantity= quantityElement.value;
    total = total+(price * quantity);
   }
    document.getElementsByClassName('total-price')[0].innerText="$" + total;

}
```



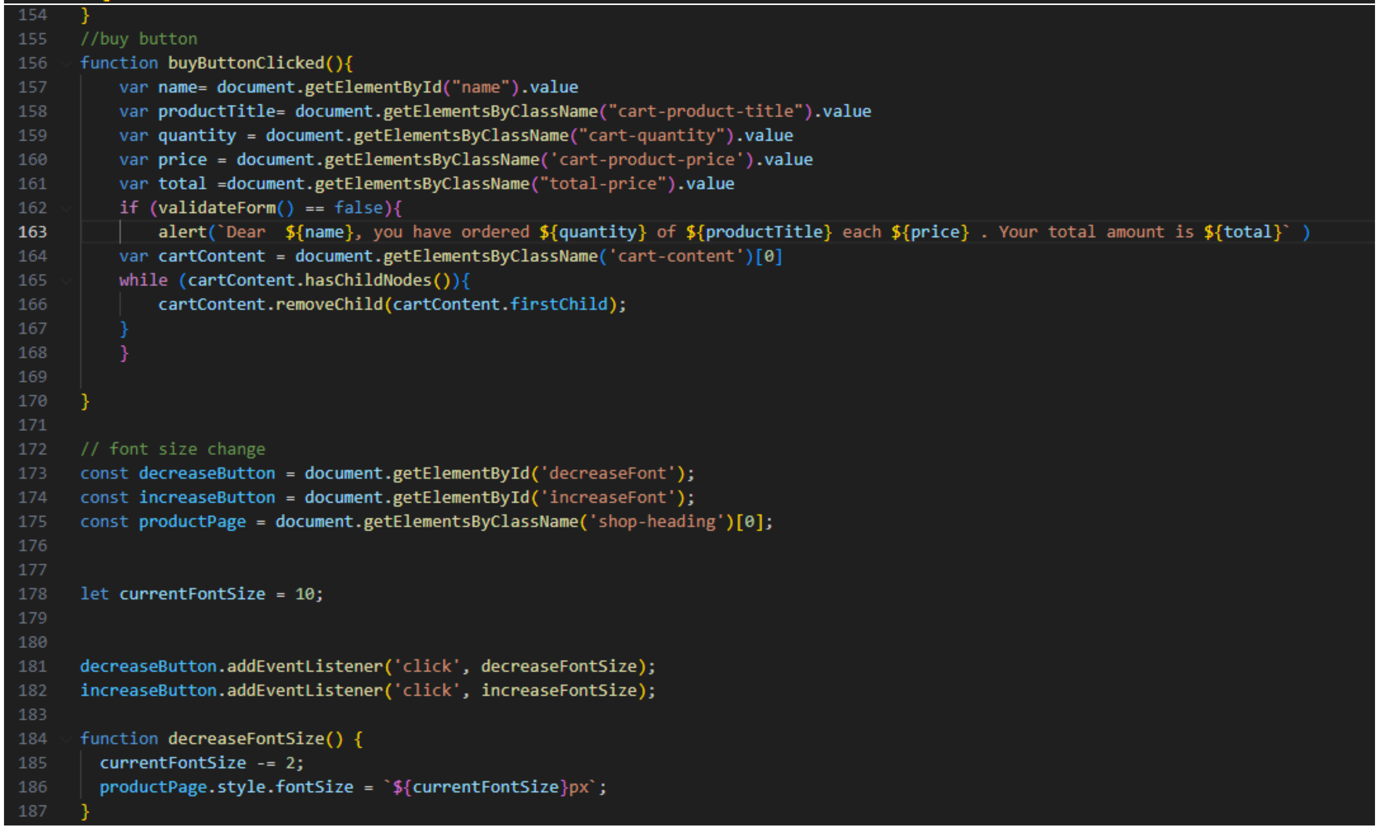
JS cartjavascript.js > buyButtonClicked

```
// Add to cart
function addCartClicked(event){
    var button = event.target;
    var shopeProducts = button.parentElement;
    var itemName = shopeProducts.getElementsByClassName("item-title")[0].innerText;
    var priceItem = shopeProducts.getElementsByClassName("price")[0].innerText;
    var productImage = shopeProducts.getElementsByClassName("item1-img")[0].src;
    addProductsToCart(itemName,priceItem,productImage);
    updatetotal();
}


function addProductsToCart(itemName,priceItem,productImage){
    var cartShopbox = document.createElement("div");
    cartShopbox.classList.add("cart-box");
    var cartItems= document.getElementsByClassName("cart-content")[0];
    var cartItemsNames = cartItems.getElementsByClassName("cart-product-title");
    for (var i =0;i< cartItemsNames.length; i++){
        if (cartItemsNames[i].innerText == itemName) {
        alert(" you have already add this item to the cart");
        return;
        }
    }


var cartBoxContent = `<div class="details-box">
                         <img src="${productImage}" alt="contact" class="cart-img">
                         <div class="cart-product-title">${itemName}</div>
                         <div class="cart-product-price">${priceItem}</div>
                         <div></div>
                         <input type="number" value="1" class="cart-quantity" id="cart-quantity">
                         <i class="bx bxs-trash-alt cart-remove"></i>
                       </div>`;

```

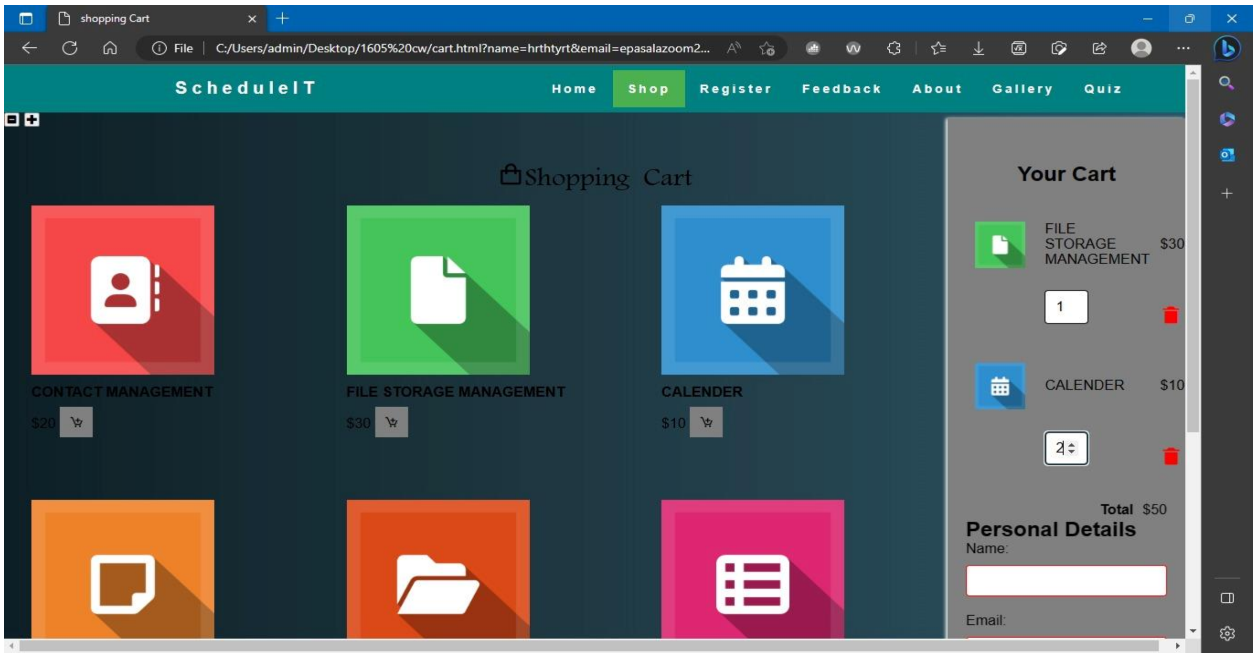

```
                    </div>`;

        cartShopbox.innerHTML = cartBoxContent;
        cartItems.append(cartShopbox);
        cartShopbox
        .getElementsByClassName("cart-remove")[0]
        .addEventListener("click", removeCartItem);
        cartShopbox
        .getElementsByClassName("cart-quantity")[0]
        .addEventListener("change", quantityChanged);
}




function validateForm() {
    const name = document.getElementById("name").value;
    const email = document.getElementById("email").value;
    const phone = document.getElementById("phone").value;
    const address = document.getElementById("address").value;

    const nameError = document.getElementById("name-error");
    const emailError = document.getElementById("email-error");
    const phoneError = document.getElementById("phone-error");
    const addressError = document.getElementById("address-error");

    let hasError = false;

    // Validate name
    if (!name || name.trim() === "") {
      nameError.innerText = "Please enter your name";
      hasError = true;
    } else {
      nameError.innerText = "";
```



```

    // Validate email
    if (!email || email.trim() === "") {
      emailError.innerText = "Please enter your email";
      hasError = true;
    } else if (!/\S+@\S+\.\S+/.test(email)) {
      emailError.innerText = "Please enter a valid email";
      hasError = true;
    } else {
      emailError.innerText = "";
    }

     // Validate phone
  if (!phone || phone.trim() === "") {
    phoneError.innerText = "Please enter your phone number";
    hasError = true;
    }else if (!/\d{10}/.test(phone)) {
    phoneError.innerText = "Please enter a valid phone number";
    hasError = true;
    } else {
    phoneError.innerText = "";
    }
    return hasError;
  }
```



```
}
//buy button
function buyButtonClicked(){
    var name= document.getElementById("name").value
    var productTitle= document.getElementsByClassName("cart-product-title").value
    var quantity = document.getElementsByClassName("cart-quantity").value
    var price = document.getElementsByClassName('cart-product-price').value
    var total =document.getElementsByClassName("total-price").value
    if (validateForm() == false){
        alert(`Dear  ${name}, you have ordered ${quantity} of ${productTitle} each ${price} . Your total amount is ${total}` )
    var cartContent = document.getElementsByClassName('cart-content')[0]
    while (cartContent.hasChildNodes()){
        cartContent.removeChild(cartContent.firstChild);
    }
    }

}

// font size change
const decreaseButton = document.getElementById('decreaseFont');
const increaseButton = document.getElementById('increaseFont');
const productPage = document.getElementsByClassName('shop-heading')[0];


let currentFontSize = 10;


decreaseButton.addEventListener('click', decreaseFontSize);
increaseButton.addEventListener('click', increaseFontSize);

function decreaseFontSize() {
  currentFontSize -= 2;
  productPage.style.fontSize = `${currentFontSize}px`;
}
```



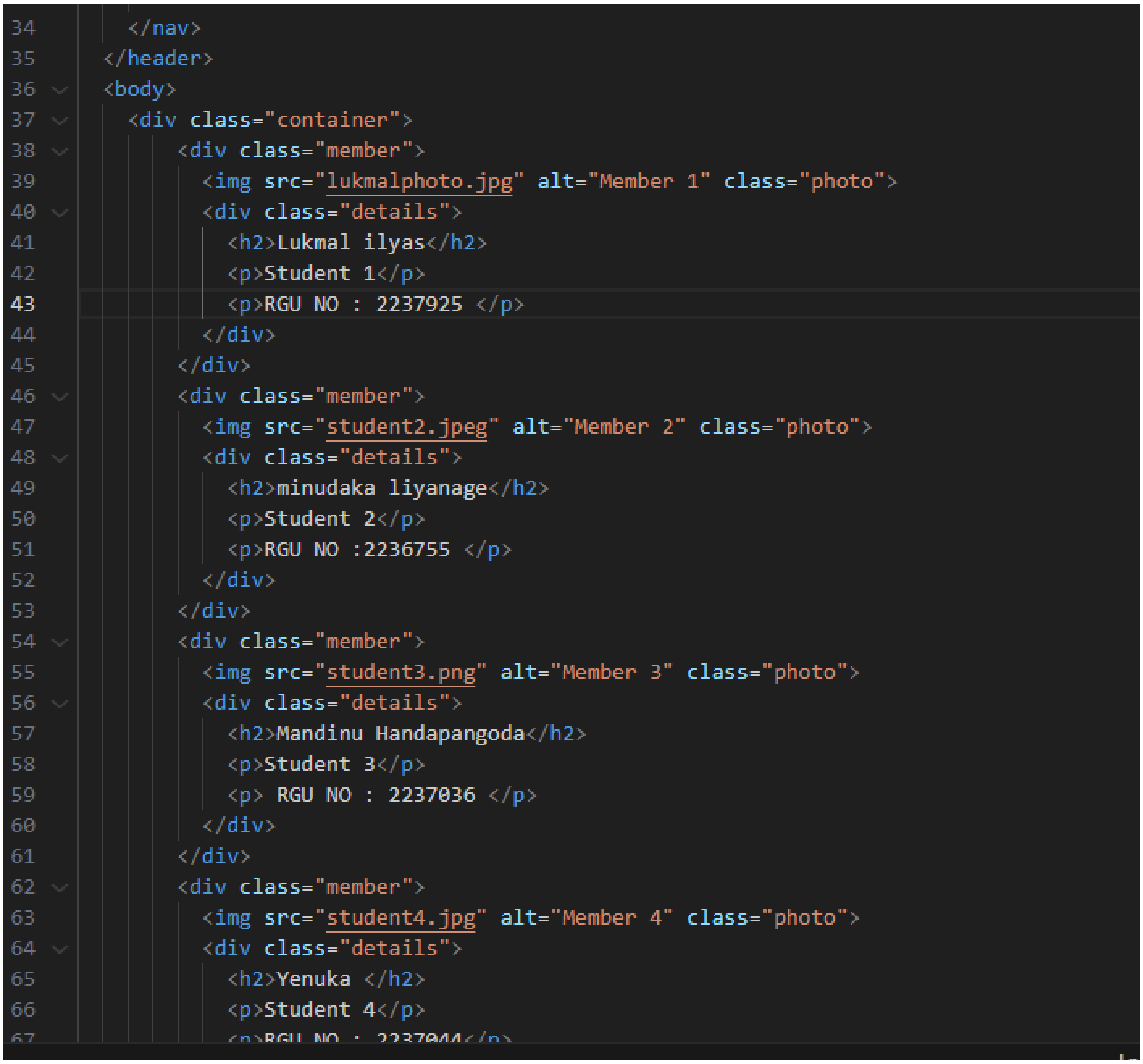
[II.] **<u>About Us</u>**

- Details of team member who developed the web age are noted in this.
- Photo of each member appears on page and upon hover it gives primary information about him.

Challenges: Addition of details to appear upon hover.

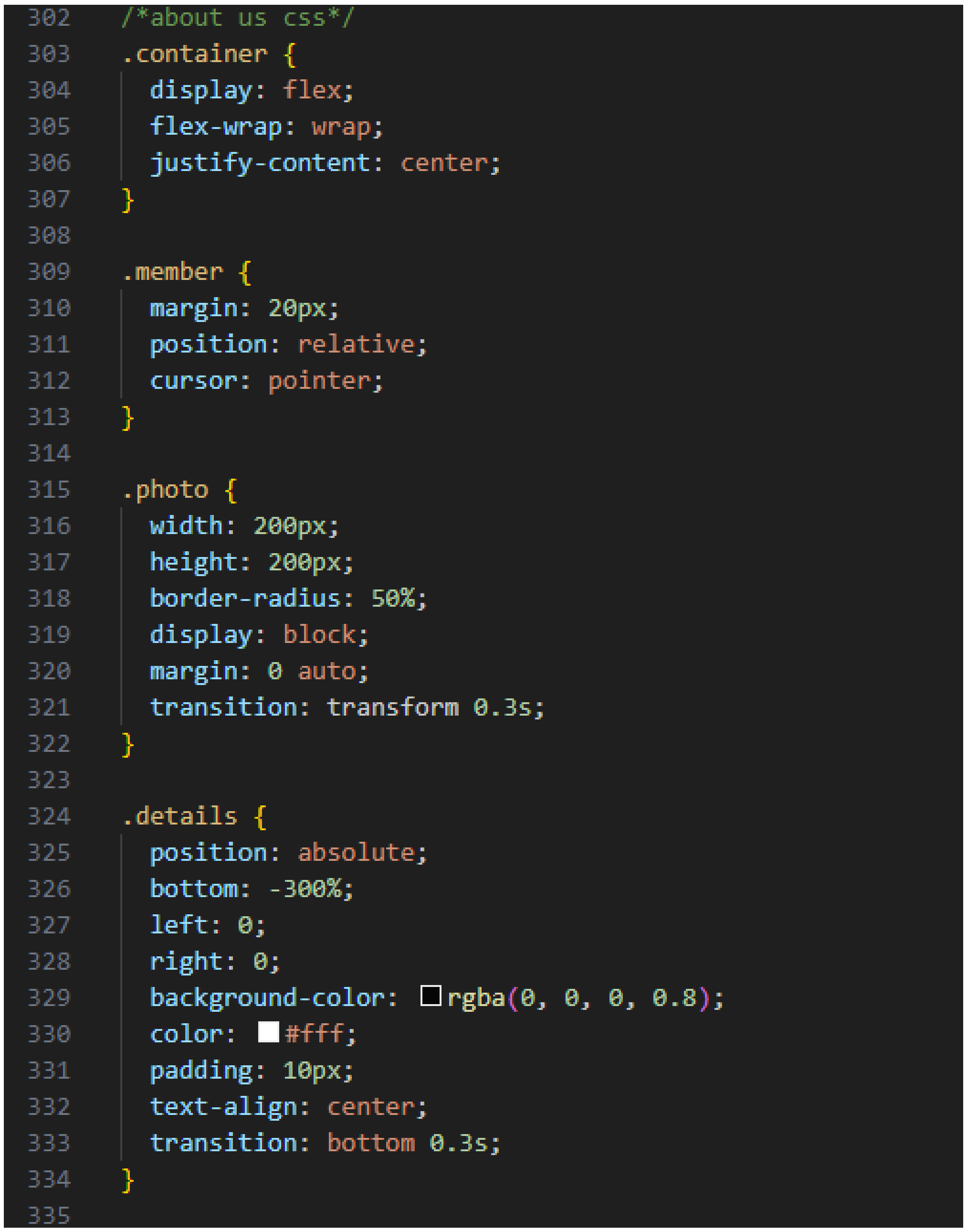
Strategy: Used a class name as ‘details’ for another div separately within the main div tag ‘container’ and in that included the details to be appeared.

About.html

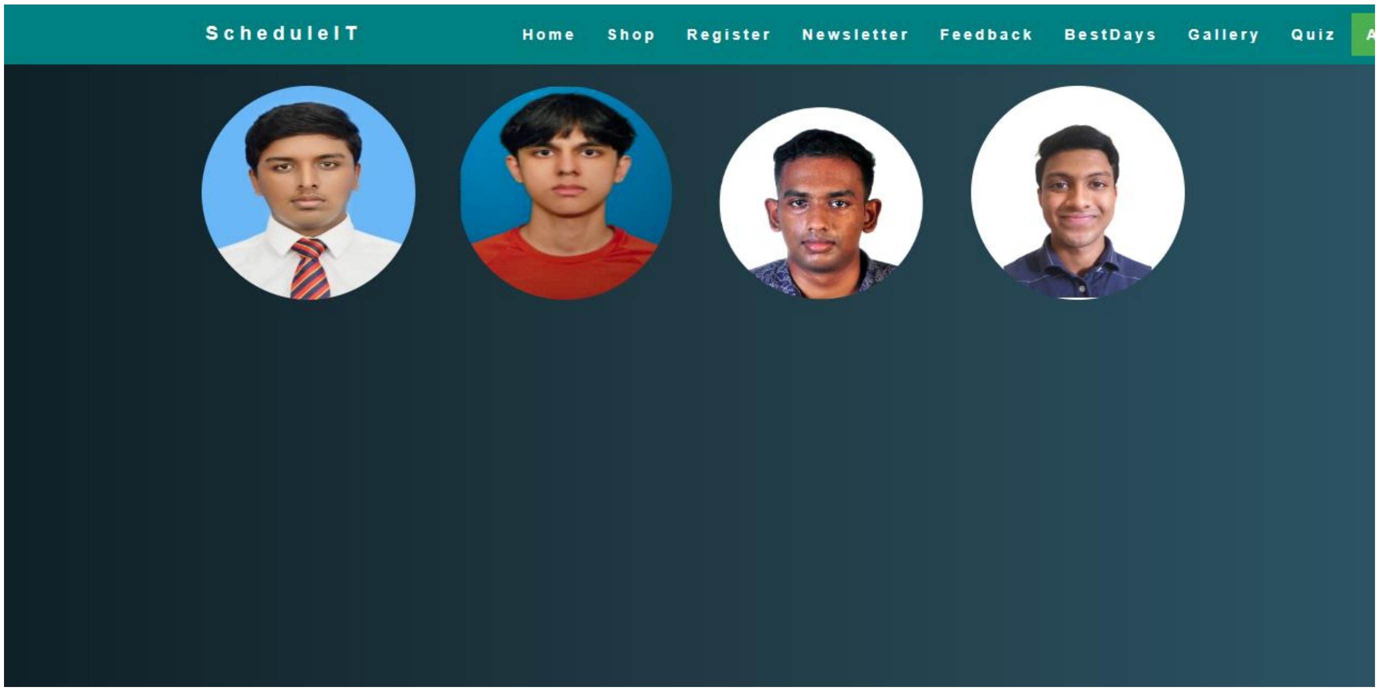


```
  </nav>
</header>
<body>
  <div class="container">
      <div class="member">
        <img src="lukmalphoto.jpg" alt="Member 1" class="photo">
        <div class="details">
          <h2>Lukmal ilyas</h2>
          <p>Student 1</p>
          <p>RGU NO : 2237925 </p>
        </div>
      </div>
      <div class="member">
        <img src="student2.jpeg" alt="Member 2" class="photo">
        <div class="details">
          <h2>minudaka liyanage</h2>
          <p>Student 2</p>
          <p>RGU NO :2236755 </p>
        </div>
      </div>
      <div class="member">
        <img src="student3.png" alt="Member 3" class="photo">
        <div class="details">
          <h2>Mandinu Handapangoda</h2>
          <p>Student 3</p>
          <p> RGU NO : 2237036 </p>
        </div>
      </div>
      <div class="member">
        <img src="student4.jpg" alt="Member 4" class="photo">
        <div class="details">
          <h2>Yenuka </h2>
          <p>Student 4</p>
          <p>RGU NO : 2237044</p>
```

Css for about

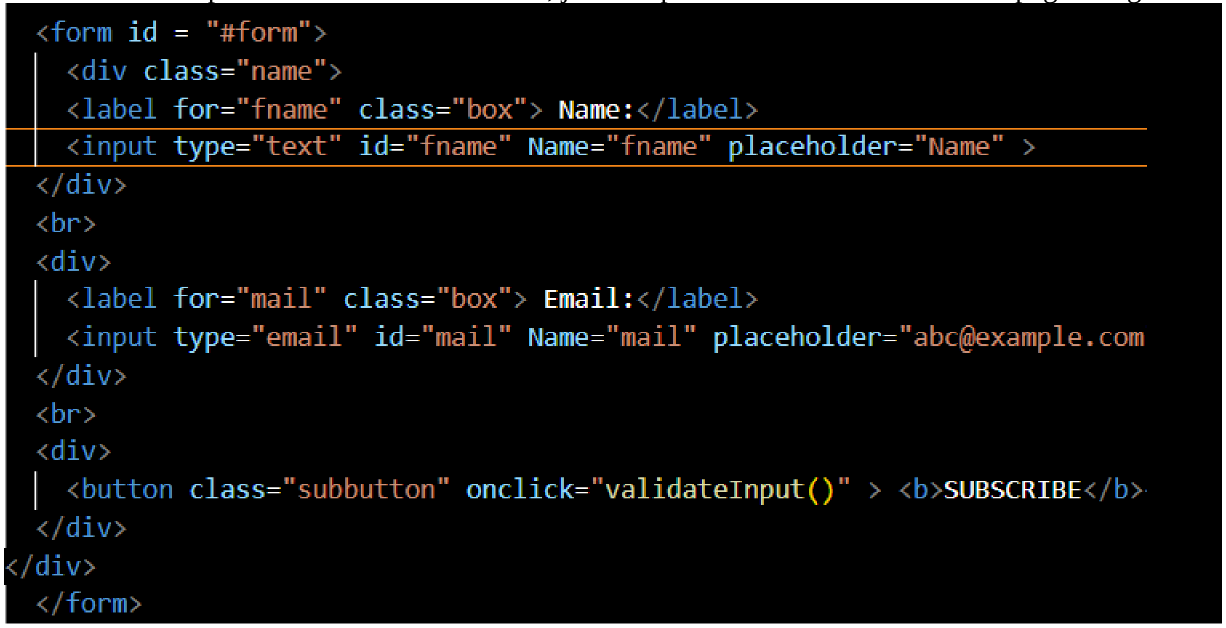


```css
/*about us css*/
.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}

.member {
  margin: 20px;
  position: relative;
  cursor: pointer;
}

.photo {
  width: 200px;
  height: 200px;
  border-radius: 50%;
  display: block;
  margin: 0 auto;
  transition: transform 0.3s;
}

.details {
  position: absolute;
  bottom: -300%;
  left: 0;
  right: 0;
  background-color: rgba(0, 0, 0, 0.8);
  color: #fff;
  padding: 10px;
  text-align: center;
  transition: bottom 0.3s;
}

```



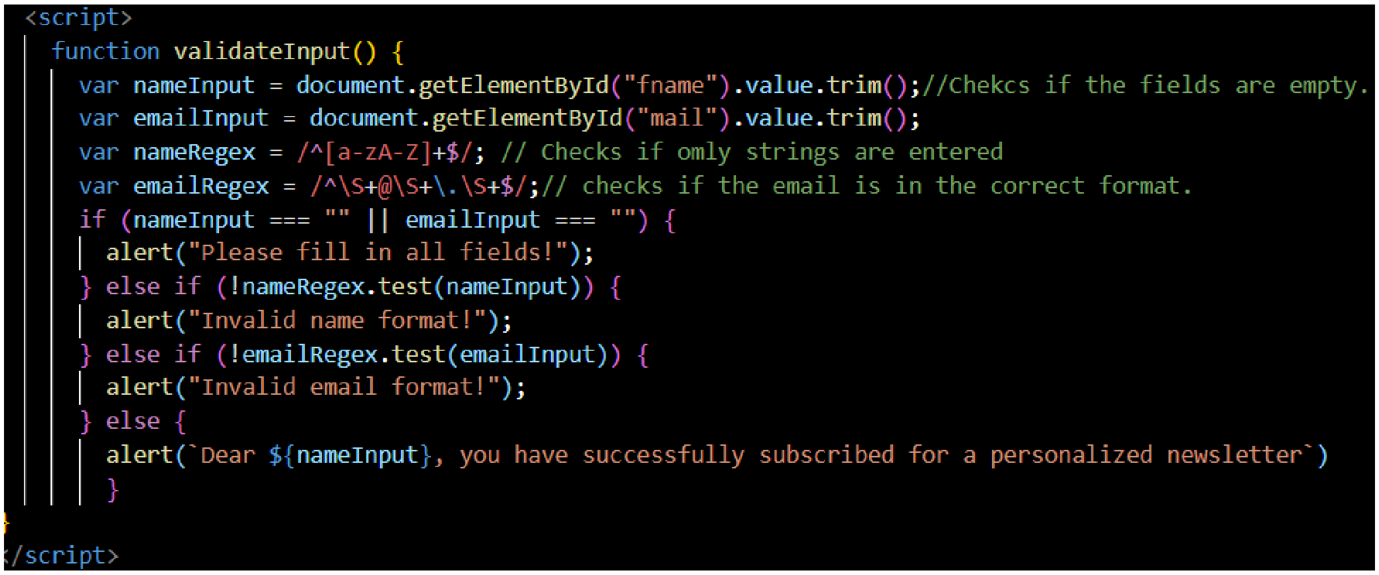
Student 4

As student 4 I had to create a newsletter subscription form , allow the user to change the background and text color of my favorite page and to store data in a xml file and represent them using java script and CSS in a web page. To create the subscription form I used basic html , java script and CSS. I created the web page using the <form> tag.



```
<form id = "#form">
  <div class="name">
  <label for="fname" class="box"> Name:</label>
  <input type="text" id="fname" Name="fname" placeholder="Name" >
</div>
<br>
<div>
  <label for="mail" class="box"> Email:</label>
  <input type="email" id="mail" Name="mail" placeholder="abc@example.com
</div>
<br>
<div>
  <button class="subbutton" onclick="validateInput()" > <b>SUBSCRIBE</b>
</div>
</div>
  </form>
```

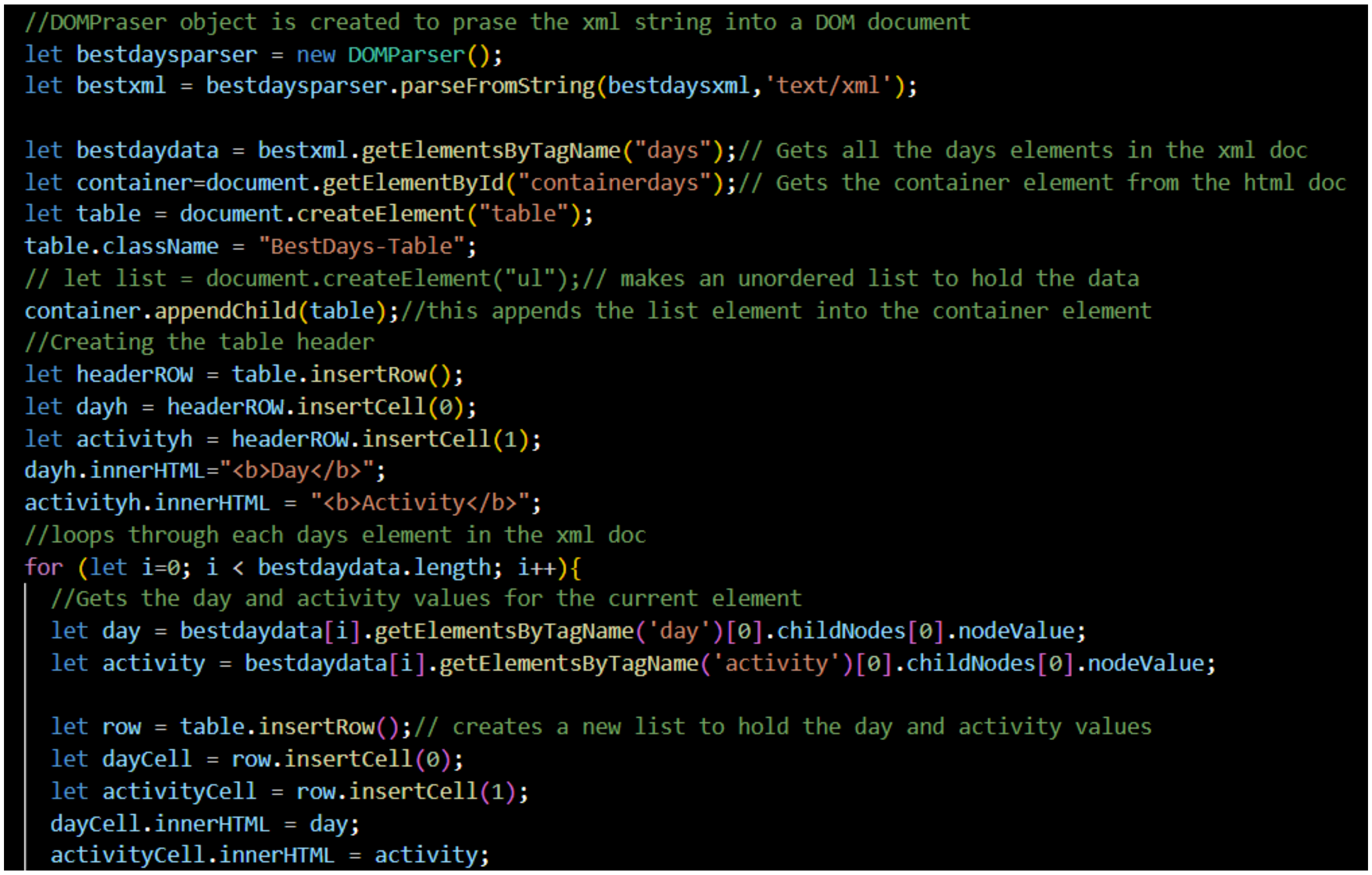
Within the <label> tags I create the labels for the Name and Email. And then I got inputs from the user for the name and email. I also created a button so that the user can click to submit his or her information. Within the button tags I called a java script function known as validateInput(). This function checks if both the fields are filled and in the name field it checks if only strings are entered and in the email field it checks if the proper format of the email is entered. And if wrong inputs were entered an alert message pops up so that the user can correct what he or she entered.



```
<script>
  function validateInput() {
    var nameInput = document.getElementById("fname").value.trim();//Chekcs if the fields are empty.
    var emailInput = document.getElementById("mail").value.trim();
    var nameRegex = /^[a-zA-Z]+$/; // Checks if omly strings are entered
    var emailRegex = /^\S+@\S+\.\S+$/;// checks if the email is in the correct format.
    if (nameInput === "" || emailInput === "") {
      alert("Please fill in all fields!");
    } else if (!nameRegex.test(nameInput)) {
      alert("Invalid name format!");
    } else if (!emailRegex.test(emailInput)) {
      alert("Invalid email format!");
    } else {
      alert(`Dear ${nameInput}, you have successfully subscribed for a personalized newsletter`)
      }
}
</script>
```

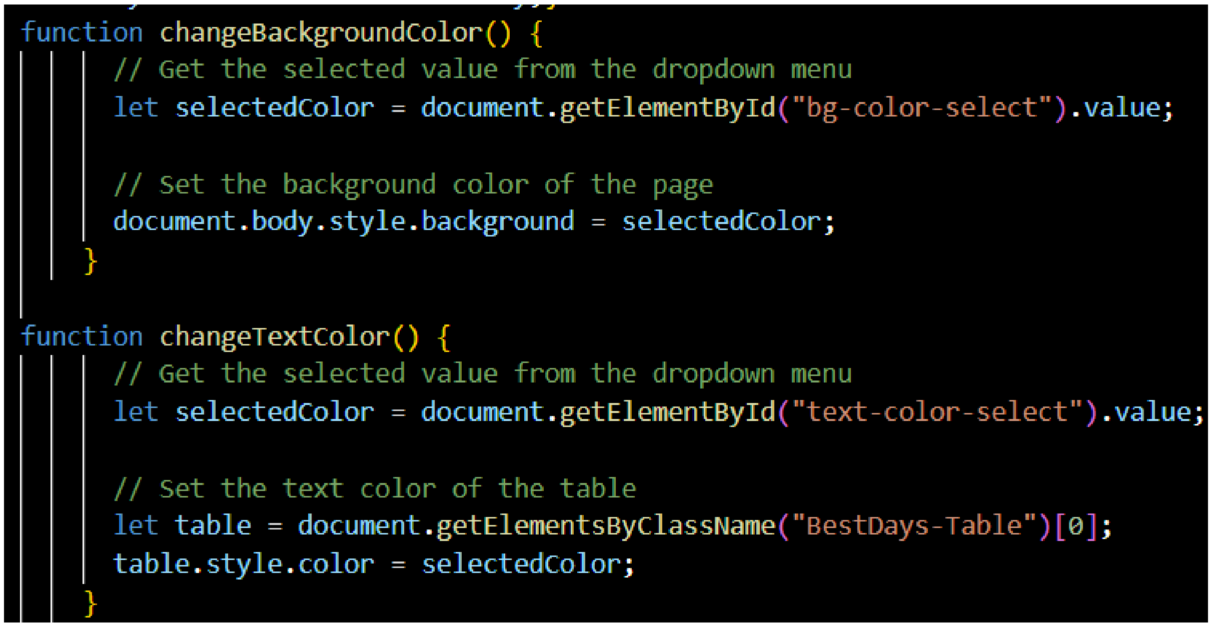
Then I used CSS to style my webpage for the subscription. I used the classes and ids to refer the specific html parts to my CSS file. Used it to make my subscribe button more interactive, have the same text style and size and to make the page look nicer and user friendly.

Then I used an xml file to store data about the best days a user can do certain tasks. Used the proper format required to form a well-organized xml file. To represent the data in the web page I used java script. I used the DOM Parser object to parse xml string to a DOM document. I created a table to show the data in an organized manner using the java script. And used a for loop to pass the xml data into the cells of the table.



```
//DOMPraser object is created to prase the xml string into a DOM document
let bestdaysparser = new DOMParser();
let bestxml = bestdaysparser.parseFromString(bestdaysxml,'text/xml');

let bestdaydata = bestxml.getElementsByTagName("days");// Gets all the days elements in the xml doc
let container=document.getElementById("containerdays");// Gets the container element from the html doc
let table = document.createElement("table");
table.className = "BestDays-Table";
// let list = document.createElement("ul");// makes an unordered list to hold the data
container.appendChild(table);//this appends the list element into the container element
//Creating the table header
let headerROW = table.insertRow();
let dayh = headerROW.insertCell(0);
let activityh = headerROW.insertCell(1);
dayh.innerHTML="<b>Day</b>";
activityh.innerHTML = "<b>Activity</b>";
//loops through each days element in the xml doc
for (let i=0; i < bestdaydata.length; i++){
  //Gets the day and activity values for the current element
  let day = bestdaydata[i].getElementsByTagName('day')[0].childNodes[0].nodeValue;
  let activity = bestdaydata[i].getElementsByTagName('activity')[0].childNodes[0].nodeValue;

  let row = table.insertRow();// creates a new list to hold the day and activity values
  let dayCell = row.insertCell(0);
  let activityCell = row.insertCell(1);
  dayCell.innerHTML = day;
  activityCell.innerHTML = activity;
```

Then with the use of CSS I made the webpage look better. Used java script to allow the user to change the background and text colors using pull down box.



```
function changeBackgroundColor() {
    // Get the selected value from the dropdown menu
    let selectedColor = document.getElementById("bg-color-select").value;

    // Set the background color of the page
    document.body.style.background = selectedColor;
  }

function changeTextColor() {
    // Get the selected value from the dropdown menu
    let selectedColor = document.getElementById("text-color-select").value;

    // Set the text color of the table
    let table = document.getElementsByClassName("BestDays-Table")[0];
    table.style.color = selectedColor;
  }
```

4. Discussion of UX/UI principles/Applications/Justifications

4.1 Navigation Techniques

There is a horizontal navigation bar on the top of the website which contains all the features of our website. These features are accessed using buttons which have a hover effect which shows a different color when hovering. Also the buttons have a green color which shows the active site your on currently. Breadcrumb trail wasn’t used here as it doesn’t really give a classy look to the website but the user is easily able to jump from each section throughout the website.

HTML:

```html
<nav>

      <div class="heading">

        <h1>ScheduleIT</h1>

      </div>

      <ul class="nav-links">

        <li><a class="active" href="mainPage.html">Home</a></li>

        <li><a href="pages/services.html">Shop</a></li>

        <li><a href="pages/contact.html">Register</a></li>

        <li><a href="FormSheet.html">Feedback</a></li>

        <li><a href="pages/about.html">About</a></li>

        <li><a href="FormSheet.html">Gallery</a></li>

        <li><a href="Quiz.html">Quiz</a></li>

      </ul>

    </nav>
```

CSS:

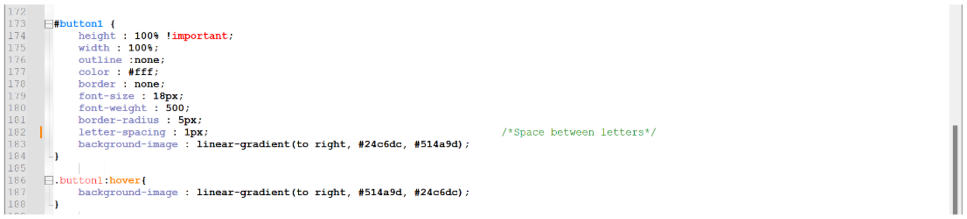
```css
* {

    margin: 0px;

    padding: 0px;

    box-sizing: border-box;

    font-family: sans-serif;

     }

     body {
       background: linear-gradient(to right, #0f2027, #203a43, #2c5364);
       font-family: 'Poppins', sans-serif;
   }

     .body-text {

       display: flex;

       color: #ccc;

       font-family: "Montserrat", sans-serif;

       align-items: center;

       justify-content: center;

       margin-top: 250px;

     }

     nav {

       display: flex;

       justify-content: space-around;

       align-items: center;

       min-height: 8vh;

       background-color: teal;

       font-family: "Montserrat", sans-serif;

```

```
 }

 .heading {

   color: white;

   letter-spacing: 5px;

   font-size: 20px;

 }


.nav-links {

   display: flex;

   justify-content: space-around;

   width: 30%;

 }

.nav-links li {

   list-style: none;

 }

.nav-links a {

   color: white;

   text-decoration: none;

   letter-spacing: 3px;

   font-weight: bold;

   font-size: 14px;

   padding: 14px 16px;

 }

.nav-links a:hover:not(.active) {

   background-color: lightseagreen;

 }

.nav-links li a.active {

    background-color: #4caf50;
    }
```

As for the gallery page, the hover effect is used to expand the thumbnail picture when the curser is placed. When the thumbnail picture is clicked it automatically scrolls to the main picture of the thumbnail picture which also contains a description of the image. There is footer placed at the bottom of the page where the user can click on the page editor's link to navigate to the page editor's page.

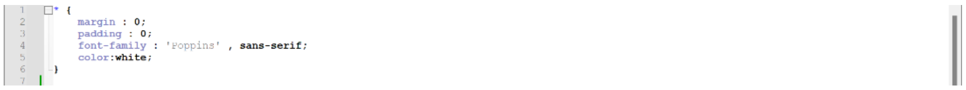
4.2 Color balance/selection/consistency

As for the registration form a linear gradient color linear gradient (to right, #24c6dc, #514a9d is used for the submit button. The text color is set to white so that the submit button is visible which satisfies the principal readability. And the hover effect is used on the button where the linear gradient is changed to linear gradient (to right, #514a9d, #24c6dc) for the user to know that the curser is placed on the submit button which is used to satisfy emphasis principle.



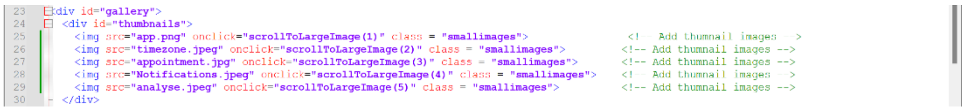
```
#button1 {
    height : 100% !important;
    width : 100%;
    outline :none;
    color : #fff;
    border : none;
    font-size : 18px;
    font-weight : 500;
    border-radius : 5px;
    letter-spacing : 1px;                                   /*Space between letters*/
    background-image : linear-gradient(to right, #24c6dc, #514a9d);
}

.button1:hover{
    background-image : linear-gradient(to right, #514a9d, #24c6dc);
}
```

As for the gallery page the text is given in white because the background of the webpage is green in color. Therefore, for the text to be visible white color is used. Use of contrasting colors helps the user to identify the text instantly and draw the user's attention. This satisfies the ux ui principles readability and emphasis.



```
* {
    margin : 0;
    padding : 0;
    font-family : 'Poppins' , sans-serif;
    color:white;
}
```

The gallery contains a set of images about the application where the user will be able to get a quick summary of the application without having to scroll through each individual image. This satisfies communicating information.



```
<div id="gallery">
 <div id="thumbnails">
   <img src="app.png" onclick="scrollToLargeImage(1)" class = "smallimages">           <!-- Add thumnail images -->
   <img src="timezone.jpeg" onclick="scrollToLargeImage(2)" class = "smallimages">      <!-- Add thumnail images -->
   <img src="appointment.jpg" onclick="scrollToLargeImage(3)" class = "smallimages">    <!-- Add thumnail images -->
   <img src="Notifications.jpeg" onclick="scrollToLargeImage(4)" class = "smallimages">  <!-- Add thumnail images -->
   <img src="analyse.jpeg" onclick="scrollToLargeImage(5)" class = "smallimages">       <!-- Add thumnail images -->
 </div>
```

For the Home page, we decided to go with a green based theme throughout. The shades of green get darker as you go down the page which gives the user a sense of depth. The images used are matching to the green color theme therefore not contrasting. Overall the green color theme is supposed to give a refreshing look towards the user.

```css
.intro {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  width: 100%;
  height: 520px;
  background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0, 0, 0, 0.5) 100%), url("scheduleexamplepic.png");
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
}

.intro h1 {
  font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
  font-size: 60px;
  color: #ffffff;
  font-weight: bold;
  text-transform: uppercase;
  margin: 0;
}

.intro p {
  font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
  font-size: 40px;
  color: #ffffff;
  text-transform: uppercase;
  margin: 20px 0;
}

.intro button {
  background-color: #5bcc97;
  color: #ffffff;
  padding: 10px 25px;
  border: none;
  border-radius: 5px;
  font-size: 20px;
  font-weight: bold;
  cursor: pointer;
  box-shadow: 0px 0px 20px rgba(255, 255, 255, 0.4)
}

.intro button a {
  color: rgb(255, 255, 255);
}

.intro button:hover {
  background-color: #1c7e51;
}

.desc {
  display: flex;
  justify-content: space-around;
  align-items: center;
  padding: 40px 80px;
}

.desc .desc-text {
  font-family: Roboto;
  font-size: 30px;
```

```
 color: #11dfd4;
  margin: 10px 0;
  text-align: center;
}

.desc2 {
  display: flex;
  justify-content: center;
  align-items: center;
        padding: 80px;
        background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0,
0, 0, 0.5) 100%)
        }

      .desc2 img {
        width: 600px;
        max-width: 100%;
        height: auto;
        border-radius: 10px;
        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.3);
        order: 1;
        margin-left: 30px;
      }
      .desc2-text {
        flex: 1;
        margin-right: 30px;
      }

      .desc2-text p {
        font-family: Roboto;
        font-size: 24px;
        color: #0ee4b6;
        margin: 10px 0 30px;
        line-height: 1.5;
        text-align: justify;
      }
```

For the shopping cart and about page gradient colour is used in the background linear-gradient(to right, #0f2027, #203a43, #2c5364); and the text colour of the items are set to black with weight of 600 and the hover effect is also used and for your cart grey colour background is used.

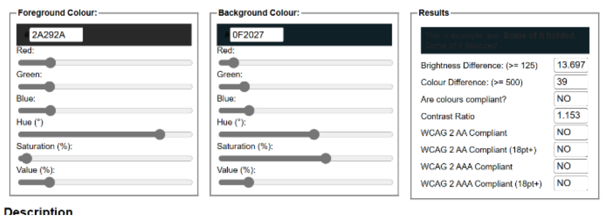
```
background: linear-gradient(to right, #0f2027, #203a43, #2c5364);
```

```
background: gray;
 box-shadow: -2px 0 4px hsl(160, 43%, 99%);
 ;
```

Used white for texts and a linear gradient with blue and green for the background.

```
body {
   background: linear-gradient(to right, #0f2027, #203a43, #2c5364);
   font-family: 'Poppins', sans-serif;
}
```

4.3 Color Contrast Test



Foreground Colour: # 2A292A

Red: · Green: · Blue: · Hue (°) · Saturation (%): · Value (%):

Background Colour: # 0F2027

Red: · Green: · Blue: · Hue (°): · Saturation (%): · Value (%):

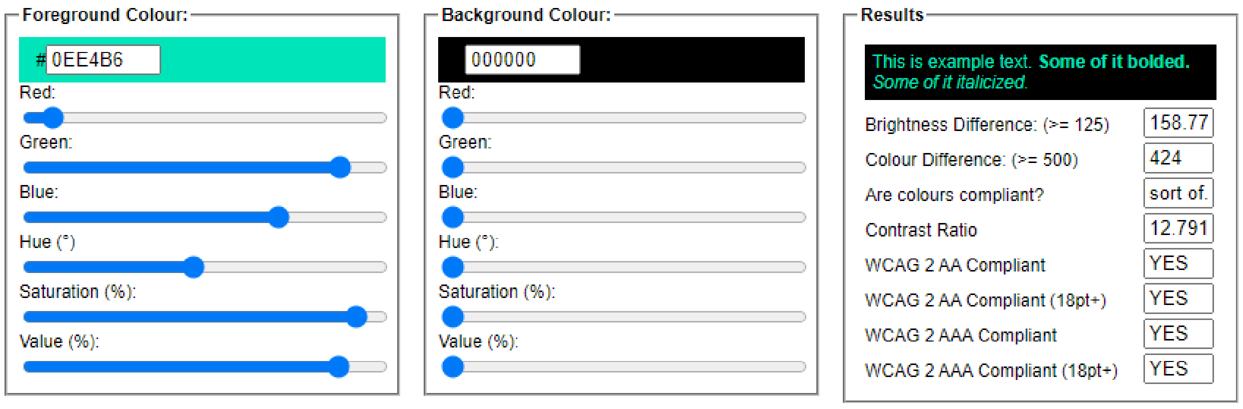
Results

This is example text. **Some of it bolded.** *Some of it italicized.*

Brightness Difference: (>= 125)	13.697
Colour Difference: (>= 500)	39
Are colours compliant?	NO
Contrast Ratio	1.153
WCAG 2 AA Compliant	NO
WCAG 2 AA Compliant (18pt+)	NO
WCAG 2 AAA Compliant	NO
WCAG 2 AAA Compliant (18pt+)	NO

Description

div class = "desc2"



Foreground Colour: # 0EE4B6

Red: · Green: · Blue: · Hue (°) · Saturation (%): · Value (%):

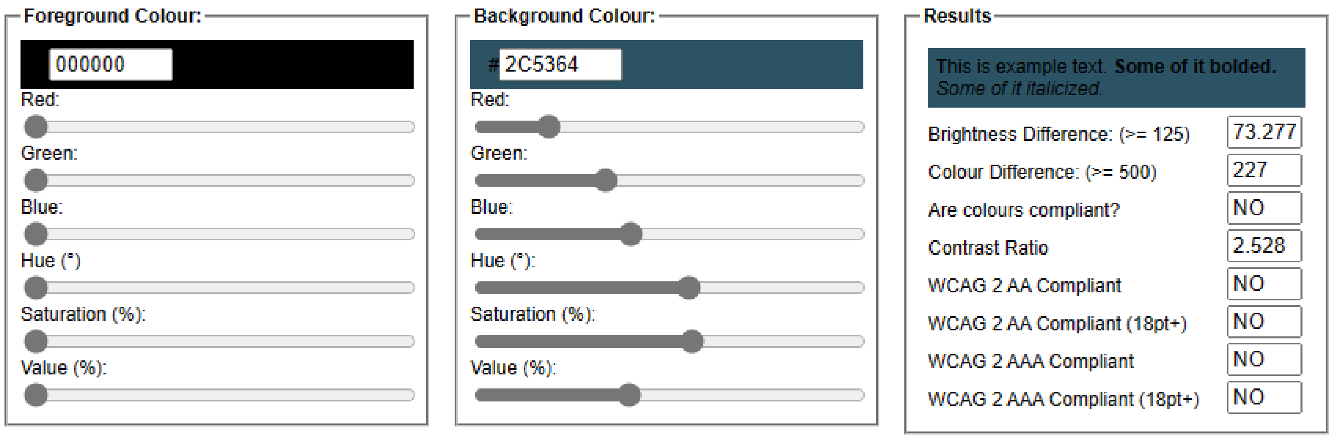
Background Colour: 000000

Red: · Green: · Blue: · Hue (°): · Saturation (%): · Value (%):

Results

This is example text. **Some of it bolded.** *Some of it italicized.*

Brightness Difference: (>= 125)	158.77
Colour Difference: (>= 500)	424
Are colours compliant?	sort of.
Contrast Ratio	12.791
WCAG 2 AA Compliant	YES
WCAG 2 AA Compliant (18pt+)	YES
WCAG 2 AAA Compliant	YES
WCAG 2 AAA Compliant (18pt+)	YES

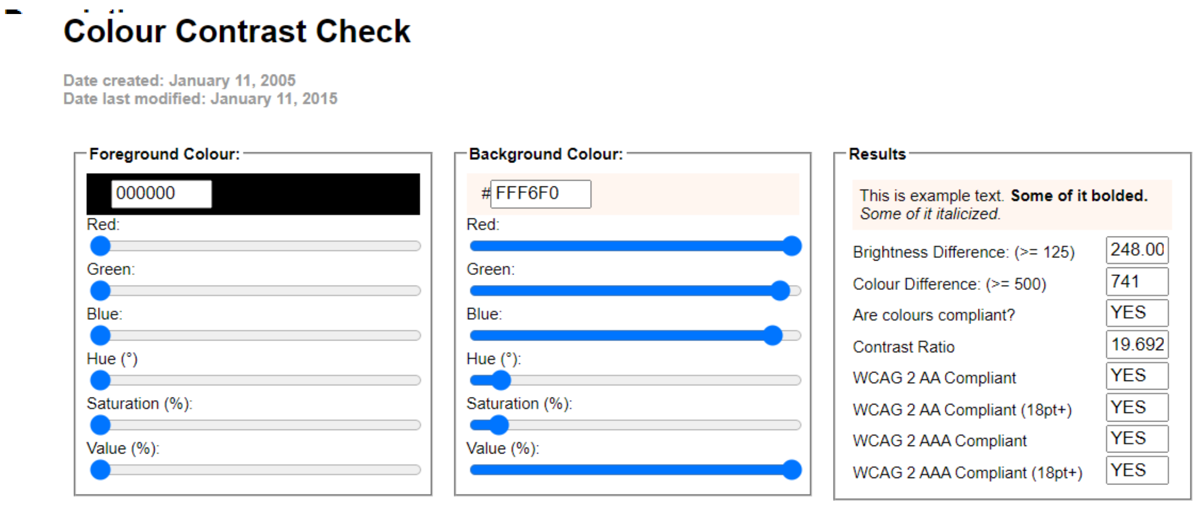

Foreground Colour: 000000
Red:
Green:
Blue:
Hue (°)
Saturation (%):
Value (%):

Background Colour: # 2C5364
Red:
Green:
Blue:
Hue (°):
Saturation (%):
Value (%):

Results

This is example text. **Some of it bolded.**
Some of it italicized.

Brightness Difference: (>= 125)	73.277
Colour Difference: (>= 500)	227
Are colours compliant?	NO
Contrast Ratio	2.528
WCAG 2 AA Compliant	NO
WCAG 2 AA Compliant (18pt+)	NO
WCAG 2 AAA Compliant	NO
WCAG 2 AAA Compliant (18pt+)	NO


Colour Contrast Check

Date created: January 11, 2005
Date last modified: January 11, 2015

Foreground Colour: 000000
Red:
Green:
Blue:
Hue (°)
Saturation (%):
Value (%):

Background Colour: # FFF6F0
Red:
Green:
Blue:
Hue (°):
Saturation (%):
Value (%):

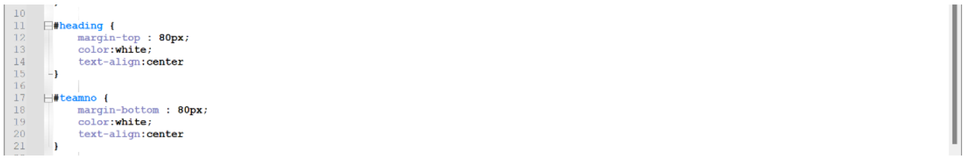
Results

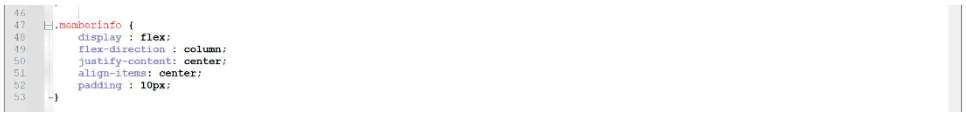
This is example text. **Some of it bolded.**
Some of it italicized.

Brightness Difference: (>= 125)	248.00
Colour Difference: (>= 500)	741
Are colours compliant?	YES
Contrast Ratio	19.692
WCAG 2 AA Compliant	YES
WCAG 2 AA Compliant (18pt+)	YES
WCAG 2 AAA Compliant	YES
WCAG 2 AAA Compliant (18pt+)	YES

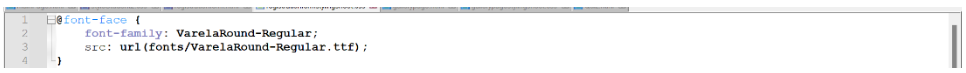
4.4 Typography / consistency

As for the presentation page the headings are shown in bigger font than the member details to show hierarchy for the in the page

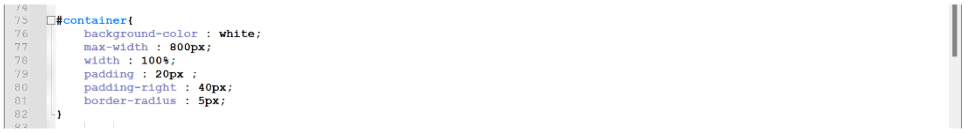

```
#heading {
    margin-top : 80px;
    color:white;
    text-align:center
}

#teamno {
    margin-bottom : 80px;
    color:white;
    text-align:center
}
```


```
.memberinfo {
    display : flex;
    flex-direction : column;
    justify-content: center;
    align-items: center;
    padding : 10px;
}
```

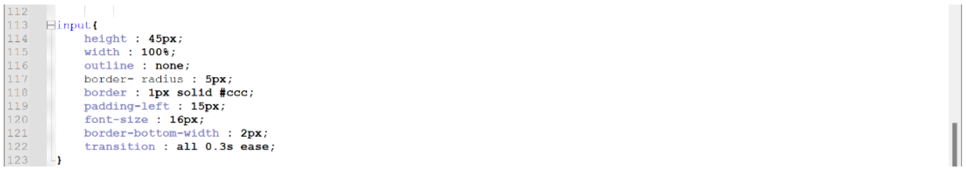
As for the registration form the heading is shown in a bigger font size and text labels of the input boxes are shown in a smaller font size in order hierarchy of the form which will help to show the flow of the form. The same font size and style are used for the text labels to group them together to show that they are text labels. Apart from the heading and text labels a different font is used for the submit button. This shows that it is a standout from the rest of the form, and which will help to draw the user's attention.



```
@font-face {
    font-family: VarelaRound-Regular;
    src: url(fonts/VarelaRound-Regular.ttf);
}
```

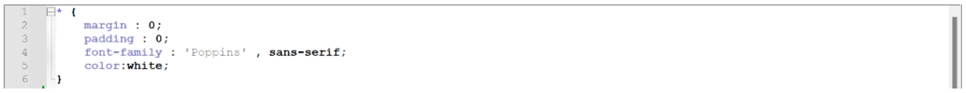


```
#container{
    background-color : white;
    max-width : 800px;
    width : 100%;
    padding : 20px ;
    padding-right : 40px;
    border-radius : 5px;
}
```



```
input{
    height : 45px;
    width : 100%;
    outline : none;
    border- radius : 5px;
    border : 1px solid #ccc;
    padding-left : 15px;
    font-size : 16px;
    border-bottom-width : 2px;
    transition : all 0.3s ease;
}
```

As for the gallery page the heading of the image is represented in a bigger font format and description of the image is represented in a smaller font format to show hierarchy in the gallery page. This will help the user to identify the headings and descriptions of the gallery page instantly. Sans serif font is used here as is clear and easier to read.



```
* {
    margin : 0;
    padding : 0;
    font-family : 'Poppins' , sans-serif;
    color:white;
}
```

For the Home page, it starts with a comparatively larger font size (60px) title, which instantly is the center of attention in the page, which serves the purpose of sending a direct message to the user. A register button right under that title is the second thing the user's eyes go to because of the slightly bright green color on it.
CSS:

```
.intro h1 {
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    font-size: 60px;
    color: #ffffff;
    font-weight: bold;
    text-transform: uppercase;
    margin: 0;
  }

  .intro p {
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
    font-size: 40px;
    color: #ffffff;
    text-transform: uppercase;
    margin: 20px 0;
  }
.intro button {
    background-color: #5bcc97;
    color: #ffffff;
    padding: 10px 25px;
    border: none;
    border-radius: 5px;
    font-size: 20px;
    font-weight: bold;
    cursor: pointer;
    box-shadow: 0px 0px 20px rgba(255, 255, 255, 0.4)
  }

  .intro button a {
    color: rgb(255, 255, 255);
  }


  .intro button:hover {
    background-color: #1c7e51;
  }
```

Then the first description comes next with a different color within a darker background which allows it to really outline the description. Font (Roboto) ,Font size 30px have been used here to make it appealing to the eyes as its not too big or too small to read.

CSS:

```
.desc {
    display: flex;
    justify-content: space-around;
    align-items: center;
    padding: 40px 80px;
  }

  .desc .desc-text {
    font-family: Roboto;
    font-size: 30px;
    color: #11dfd4;
    margin: 10px 0;
    text-align: center;
  }
```

Then the second description comes with a different color as the background color gets darker. The font is the same, but the font size is reduced.

```
.desc2 {
   display: flex;
   justify-content: center;
   align-items: center;
   padding: 80px;
   background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0, 0, 0, 0.5)
100%)
   }

 .desc2 img {
   width: 600px;
   max-width: 100%;
   height: auto;
   border-radius: 10px;
   box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.3);
   order: 1;
   margin-left: 30px;
 }
 .desc2-text {
   flex: 1;
   margin-right: 30px;
 }

 .desc2-text p {
   font-family: Roboto;
   font-size: 24px;
   color: #0ee4b6;
   margin: 10px 0 30px;
   line-height: 1.5;
   text-align: justify;
 }
```

In the shopping cart different font style is used for the item names , and have a small font size compared to the navigation bar . h1 tag is used for headings and h3 for other subtropics

```
font-family: sans-serif;
 font-size: 18px;
```

The headings have a bigger font when compared to the other words. So it makes it easier to identify the difference.

```
.heading {

   color:black;

   text-transform: uppercase;

   letter-spacing: 5px;

   font-size: 20px;

  }
```

4.5 Accessibility

4.6 Accessibility Test

Student 1

Accessibility of presentation page

- h1 tag is used for the heading

```
<h1 id = "heading"> MEET THE TEAM </h1>
```

- h3 tag is used for team number

```
<h3 id = "teamno"> Team 07 </h3>
```

- p tag is used for member details

```
<p>Name : Lukmal Ilyas</p>
<p>RGU : 2237925</p>
<p>IIT : 20221732</p>
```

- alt tag is used for alternative text descriptions if the image is not displayed

```
<img src="lukmalphoto.jpg" alt="memberphoto">
```

Accessibility of registration form

- h2 tag is used for registration form heading

```
<h2 id="formheading">Registration form</h2>
```

- label type text tag is used to label the inputs of the form

```
<label for="occupation">Occupation</label>
```

- Input tag is used to input data into the form

```
<input type="text" id="occupation" placeholder="Enter your occupation" name="occupation" onkeyup="return
validateOccupation()">
```

- Input type radio tag is used for selections

```
<input type="radio" id="circle-1" name="gender">
<input type="radio" id="circle-2" name="gender">
```

Accessibility of gallery page

- P tag is used for description of large images

<p>Time zone tool in the scheduling application helps users to schedule meetings with people in other countries without any conflicts. The user can have multiple time zones in the menu based on the countries where the user has most of their meetings which will help the user to compare time zones easily. </p>

- h2 tag is used for headings of large images

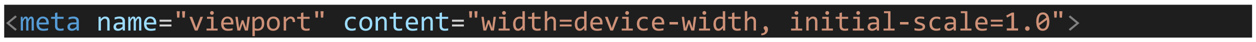
<h2>Time-zone view</h2>

- alt tag is used for alternative text descriptions if the image is not displayed

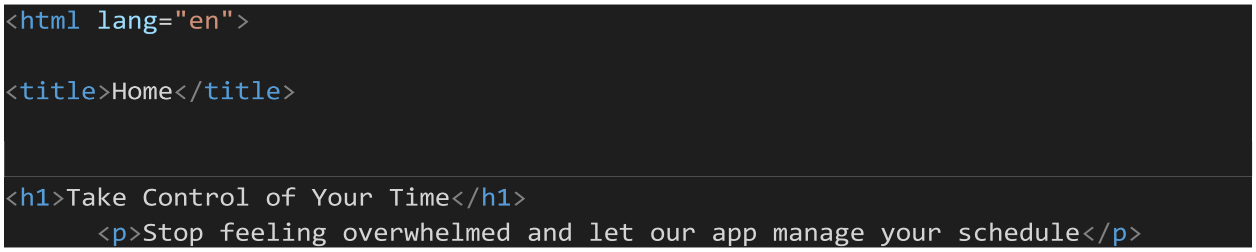
<img id="large-image-2" src="timezone.jpeg" class = "largeimages" alt="timezone">

Student 2

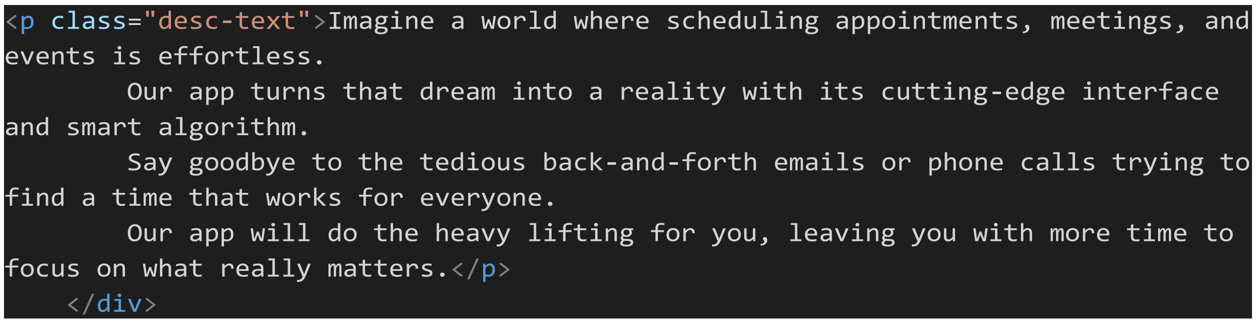
Text accessibility:



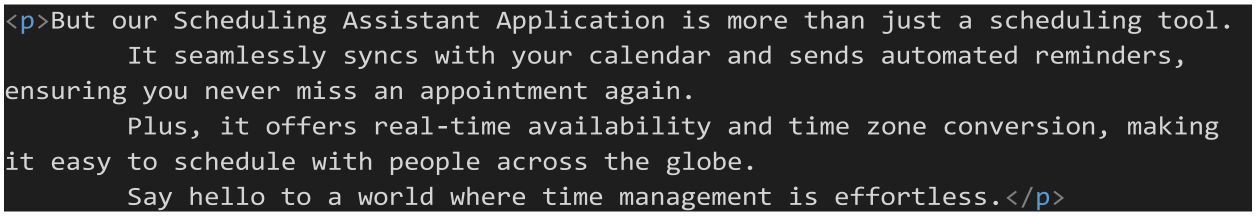
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



```
<html lang="en">

<title>Home</title>


<h1>Take Control of Your Time</h1>
      <p>Stop feeling overwhelmed and let our app manage your schedule</p>
```



```
<p class="desc-text">Imagine a world where scheduling appointments, meetings, and
events is effortless.
        Our app turns that dream into a reality with its cutting-edge interface
and smart algorithm.
        Say goodbye to the tedious back-and-forth emails or phone calls trying to
find a time that works for everyone.
        Our app will do the heavy lifting for you, leaving you with more time to
focus on what really matters.</p>
    </div>
```



```
<p>But our Scheduling Assistant Application is more than just a scheduling tool.
        It seamlessly syncs with your calendar and sends automated reminders,
ensuring you never miss an appointment again.
        Plus, it offers real-time availability and time zone conversion, making
it easy to schedule with people across the globe.
        Say hello to a world where time management is effortless.</p>
```

```
<ul class="nav-links">
            <li><a href="mainPage.html">Home</a></li>
            <li><a href="cart.html">Shop</a></li>
            <li><a href="registrationform.html">Register</a></li>
            <li><a href="newsletter.html">Newsletter</a></li>
            <li><a href="FormSheet.html">Feedback</a></li>
            <li><a href="bestdayspage.html">BestDays</a></li>
            <li><a href="FormSheet.html">Gallery</a></li>
            <li><a class="active" href="Quiz.html">Quiz</a></li>
            <li><a href="about.html">About</a></li>
        </ul>
```

Image accessibility:

```
<img src="schedulepic2.jpg" alt="">
```

```
<img src="timer.jpg" alt="" width="20px">
```

Form accessibility:

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

```
<label for="email">Email:</label>
          <input type="text" id="email" name="email">
```

```
<label for="comments">Comments:</label>
```

```
<label for="rating">Rating:</label>
```

```
<input type="submit" value="Submit">
<input type="button" value="Reset" onclick="resetForm()">
<input type="button" value="Back to Main Page" onclick="goToMainPage()">
```

Student 3

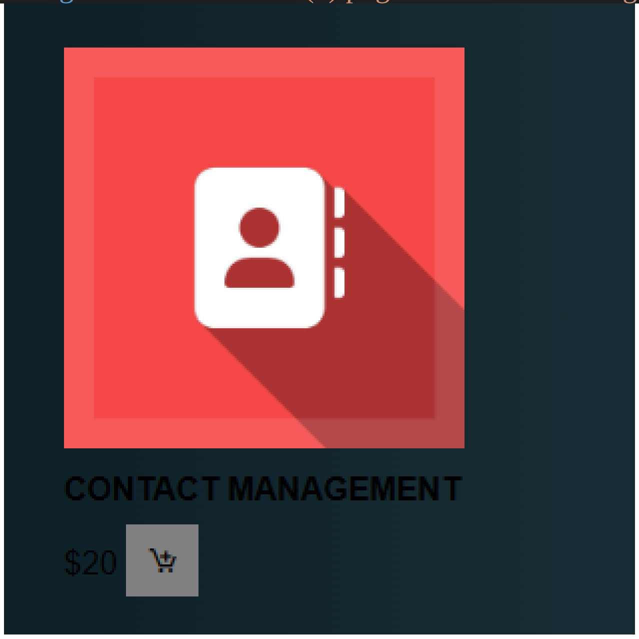
- Text Accessibility Techniques

```
html lang="en" >
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewpoint" content="width=device-width,initial-scale=1.0">
 <title>shopping Cart </title>
```

```
 <h2 class="cart-title">Your Cart</h2  >
<h3>font size change</h3>
<h2>minudaka liyanage</h2>
        <p>Student 2</p>
        <p>RGU NO :2236755 </p>
```

- Image Accessibility Techniques

```
 <img src="contact-book (1).png" alt="contact management apllication" class="item1-img">
```



- Form Accessibility Techniques

```
<form class="form"id="details-form" onsubmit="return validateForm()">
          <label for="name">Name:</label>
          <input type="text" id="name" name="name" required>
         <span id="name-error" class="error"></span>
```

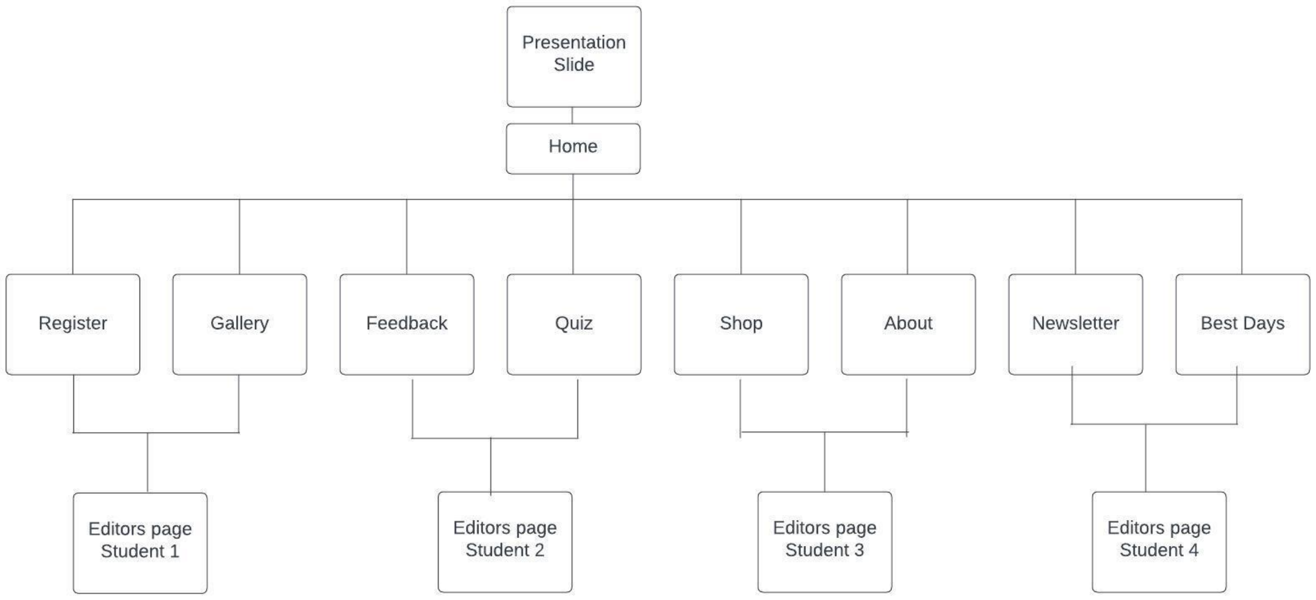
Student 4 -

- Text Accessibility Techniques –
  <h4>ScheduleIT</h4>
- Table Accessibility Techniques –

```
.BestDays-Table{
    border-collapse: collapse;
    width: 500px;
    margin:auto;
    margin-top: 50px;

  }
  .BestDays-Table th, .BestDays-Table td{
    border : 1px solid #ddd;
    padding: 9px;
    text-align:center;
  }
  .BestDays-Table th{
    background: color #f2f2f2;
    color: #333;
    font-weight: bold;
  }
```

- Form Accessibility Techniques –
  ```
  <input type="text" id="fname" Name="fname" placeholder="Name" >
  <input type="email" id="mail" Name="mail" placeholder="abc@example.com" >
  ```

4.7 Site Diagram



5. Self-Reflection

Student 1 –

As this is a group submission, we encountered problems from the initial stages of the coursework from choosing a topic to implementation of the webpage. As with choosing a topic we were given 4 topics to select and work on one. Since we were given 4 topics all the group members had different preferences. After doing research on the current existing systems for each topic we chose scheduling assistant application as our topic because there were already many existing systems for reference. These existing systems helped in our initial part of the website such as with the design part of the webpage.

Student 2 –

At first it was hard to even choose a topic everyone agreed on but now we have built a team connection that we treasure very much. We were able to divide the roles by allocating them to the right people with the right skill set.

The toughest part was using javascript to make a quiz. It took me a while to do it but after continuous research on how to do it i was able to finally get it done. I will make the code abit shorter and less complex next time.

Student 3 –
A four-person team included me in the coursework. In order to do the project on schedule and in accordance with the specifications, we collaborated to achieve this.
The combination of several tasks was one of the biggest problems we ran across during the project. We found compatibility problems that made debugging the code take a lot of time. We overcome this difficulty by honing our teamwork and communication abilities, which allowed us to locate the problem's primary source and come up with workable remedies.
I learned the importance of teamwork, communication, and attention to detail. developing a website requires a systematic approach that involves planning, design, implementation and testing. we could have improved our project by allocating more time and using alternative technologies to overcome some of the challenges.
Overall, the project was a valuable to develop my technical and interpersonal skills.

Student 4 –
We encountered many challenges, but we overcame them as a team. This helped us develop our ability to work in a team and coordinate with each other. We helped each other out at times of need and completed the group work together. During this we learnt a lot about the module as well.

6. References

W3Schools. (n.d.). HTML DOM parser. Retrieved March 31, 2023, from https://www.w3schools.com/xml/dom_parser.asp

W3Schools. (n.d.). HTML tables. Retrieved March 31, 2023, from https://www.w3schools.com/html/html_tables.asp