Website Enhancement Report

Submitted By,

Renu Gopinatha Pillai

Table of Content

Introduction

The goal of this project is to improve the user experience and increase user engagement on the Glass Expressions business website. As part of improving user interaction using JavaScript, I have analysed the website and identified key areas for improvement. This report outlines the website background information, proposed enhancements, implementation details, validation, and browser testing results.

Website Background Information

The website I chose is the one I developed for my HTML project for the business "Glass Expression."

Glass Expression is a glass art studio located in Courtenay, Comox Valley, BC. The owner of the business is Brenda, who has been operating the business for sixteen years. They specialised in creating custom glass art pieces, including stained glass, kiln-fired glass, fused glass, and leaded glass. The studio offers a wide range of services, such as glass repair, restoration, and installation. They also offer classes and workshops to teach glass art techniques to beginners and experienced artists.

The purpose of the website is to provide information about their business, the types of content and services that they offer, and to showcase the glass artwork they have done so far. The website has four pages. The first page is the 'HOME' page, in which they provide the details of the business, the latest news about their business, and the details of the glass art classes they offer. The 'SUPPLIES' page displays the details and images of all the glass art supplies they are selling at their store. On the 'GALLERY' page, their glass artwork images are showing. On the 'CONTACT US' page, the store location details, contact details, working hours, and contact form are given.

The website is built on a foundation of HTML, CSS, and JS. HTML as the primary markup language to create web pages, CSS as the styling language to define the appearance of a web page, and JavaScript as the scripting language to add interactivity and functionality to web pages.

Proposed Enhancement

The following enhancements are proposed for the "Glass Expression" Company website:

1. Implement form validation using JavaScript in the contact form of the website. Form validation is required to validate the user input in forms, ensuring that the data is entered and that it is in the correct format.
2. Display a pop-up message after submitting the contact form. They provide a visually appealing and interactive way to show the message has been successfully sent without taking the user away from their current page.
3. Make the images of the products, like stained glass kits and stained-glass grinders, on the supplies page larger when clicked on without having to leave the page, so the user gets a better viewing experience of the products.

Implementation Details

Feature1: Form validation

HTML code implementation:

To implement form validation, first I added span elements for each form field with id attributes that will be used to display error messages if any of the fields are filled out incorrectly or are empty.

JavaScript code implementation steps:

1. First, get the form element using its ID and add an event listener to it. The event listener is set to trigger the ‘validateForm’ function when the form is submitted.
2. The ‘validateForm’ function is defined to check the form fields for valid values.
3. The function starts by getting the values of the form fields (name, email, phone, and subject) and the error message elements (span field) for each field.

4. Then initializes a Boolean variable isValid to true. This variable is used to check whether all validation checks have passed. If any validation check fails, isValid is set to false.
5. Then checks the name field. If it is empty, an error message is displayed, and isValid is set to false. Otherwise, the error message is cleared.
6. Then checks the email field. If it is empty, an error message is displayed, and isValid is set to false. If the email is not in a valid format, an error message is displayed, and isValid is set to false. Otherwise, the error message is cleared.
7. Then checks the phone field. If it is empty, an error message is displayed, and isValid is set to false. If the phone number is not in a valid format, an error message is displayed, and isValid is set to false. Otherwise, the error message is cleared.
8. Then checks the subject field. If it is empty, an error message is displayed, and isValid is set to false. Otherwise, the error message is cleared.
9. If all the validation checks pass, the form is submitted.

```html
<h3 class="form-heading">Get In Touch</h3>
<!--action="mailto:someone@example.com"-->
<form method="post" action="https://learndigital.dev/" id="myForm">

    <label for="fname"> Name</label> <span id="fname-error"></span>
    <input type="text" id="fname" name="firstname" placeholder="Your name..">

    <label for="email">Email</label> <span id="email-error"></span>
    <input type="text" placeholder="Enter Email" name="email" id="email">


    <label for="phone"> Phone Number</label> <span id="phone-error"></span>
    <input type="text" id="phone" name="phone" placeholder="Enter phone Number">

    <label for="subject">Subject</label> <span id="subject-error"></span>
    <textarea id="subject" name="subject" placeholder="Write something.."></textarea>

  <input type="submit" value="Submit" id="myBtn">

</form>
```

Image1: Added span field for displaying error message.

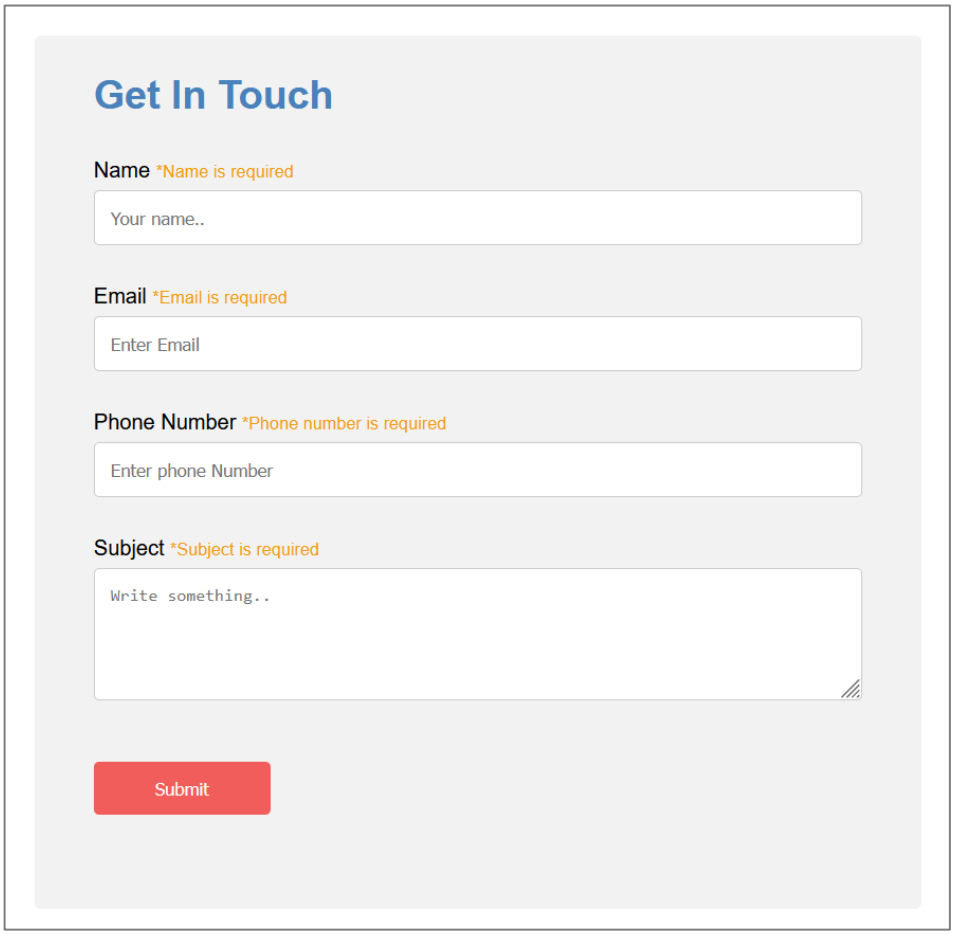


Image2: Form validation error message showing on website.

Feature2: Pop-up message

To implement pop-up success message after submitting the contact form, I used Modal box.

HTML code implementation steps:

1. Created a HTML container element for the modal box, with a unique ID.
2. Inside the modal box container, I added a paragraph element that displays the success message.
3. Then added a span element with a class of "close1" that is used to close the modal box.
4. Then added an ‘Ok’ button to submit the contact form when clicked it.

```html
<!-- HTML code for the modal box -->
<div id="myModal1" class="modal1">
  <div class="modal-content1">
    <span class="close1">&times;</span>
    <p id="modal-message1">Your message has been sent. We will get back to you as soon as possible.</p>
    <button id="okbtn">OK</button>

  </div>
</div>
```

Image3: HTML code snippet of pop-up modal box.

CSS code implementation steps:

1. The modal box is hidden by default, positioned to stay in place, and take up the full screen.
2. The "modal-content1" class sets the appearance of the message and the border of the modal box.
3. The "close1" class sets the appearance of the close button.
4. The “okbtn” id set the appearance of the ok button.

```css
/*-------Success Message------------------*/

/* Modal Box */

.modal1 {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}

/* Modal Content */
.modal-content1 {
  background-color: rgb(86, 167, 182);
  margin: auto;
  padding: 20px;
  border: 1px solid #888;
  width: 50%;
  text-align: center;

}
.modal-content1 p{
  color: white;
  font-size: 1.6rem;
}
```

Image4: CSS code snippet of pop-up modal box.

JS code implementation steps:

1. First, get the form element using its ID and add an event listener to it.
2. Then the event listener is set to trigger the validateForm function when the form is submitted.
3. Then get the modal element and its close and ok button elements using their IDs and classes, respectively.
4. The validateForm function is defined to check the form fields for valid values, it checks each field for valid values according to some regular expressions. If any field has an invalid value or is empty, the corresponding error message is displayed, and the function sets the isValid variable to false. If all fields have valid values, the function sets the isValid variable to true.
5. If all fields have valid values (i.e., isValid is true), the function displays a pop-up message by setting the display style of the modal element to "block."
6. The function also sets up two event handlers for the pop-up message: one for the OK button and one for the close button. If the OK button is clicked, the form is submitted. If the close button is clicked, the pop-up message is closed without submitting the form.

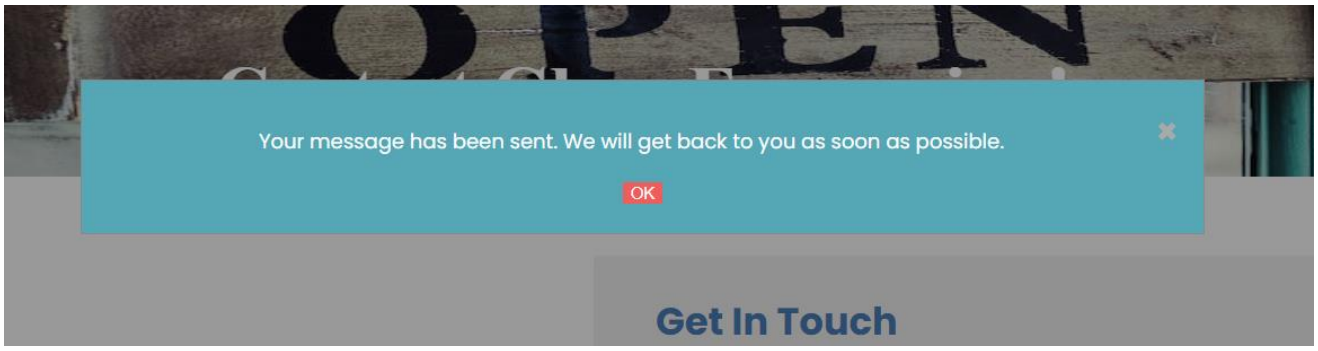


Image5: Popup success message.

Feature3: Modal Images

To make images larger when clicked on without having to leave the page I used modal image.

HTML code implementation steps:

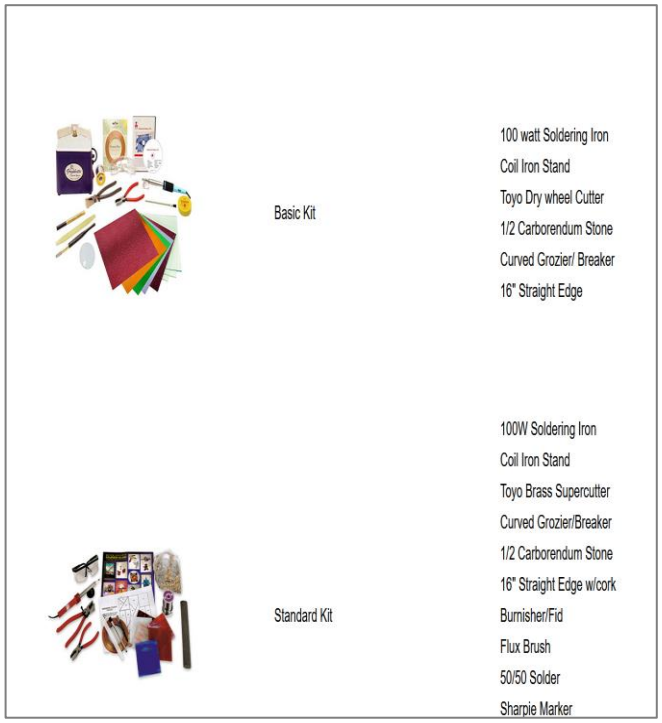
1. Created a HTML container element for the modal box, with a unique ID.
2. Then created a "close" button with the class "close allows the user to close the modal when clicked.
3. An img element with the class "modal-content" and an ID of "myImage". This element displays the image in the modal.
4. A div element with an ID of "caption". This element displays the caption or text related to the image.

```html
<!-- The Modal -->
<div id="myModal" class="modal">

  <!-- The Close Button -->
  <span class="close">&times;</span>

  <!-- Modal Content (The Image) -->
  <img class="modal-content" id="myImage" alt="">

  <!-- Modal Caption (Image Text) -->
  <div id="caption"></div>
</div>
```

Image6: HTML code snippet of Modal Image.

JS code implementation steps:

1. The window.onload function is used to make sure that the JavaScript code is executed only after the webpage has finished loading. The function first gets the span element that is used to close the modal popup and adds a click event listener to it. It then gets each image element in the gallery and adds a click event listener to each of them.
2. Then the imageClick function is called when an image is clicked. It sets the display property of the modal to "block", which makes it visible, and sets the src attribute of the modalImg element to the src attribute of the clicked image. It also sets the innerHTML of the captionText element to the alt attribute of the clicked image, which is used as the caption for the image in the modal.
3. Then the imageClose function is called when the span element is clicked. It sets the display property of the modal to "none", which hides it.



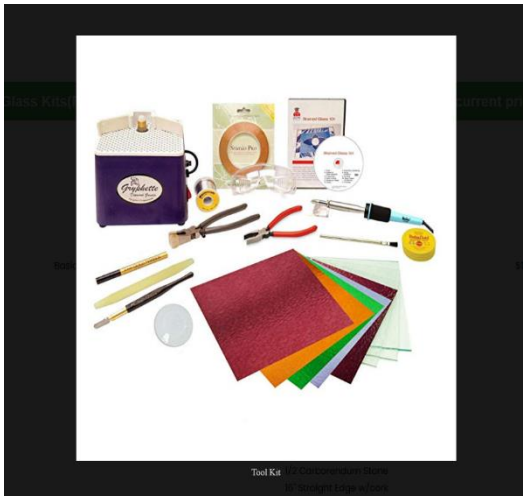


Image7: Modal Image.

HTML and CSS Validation Result

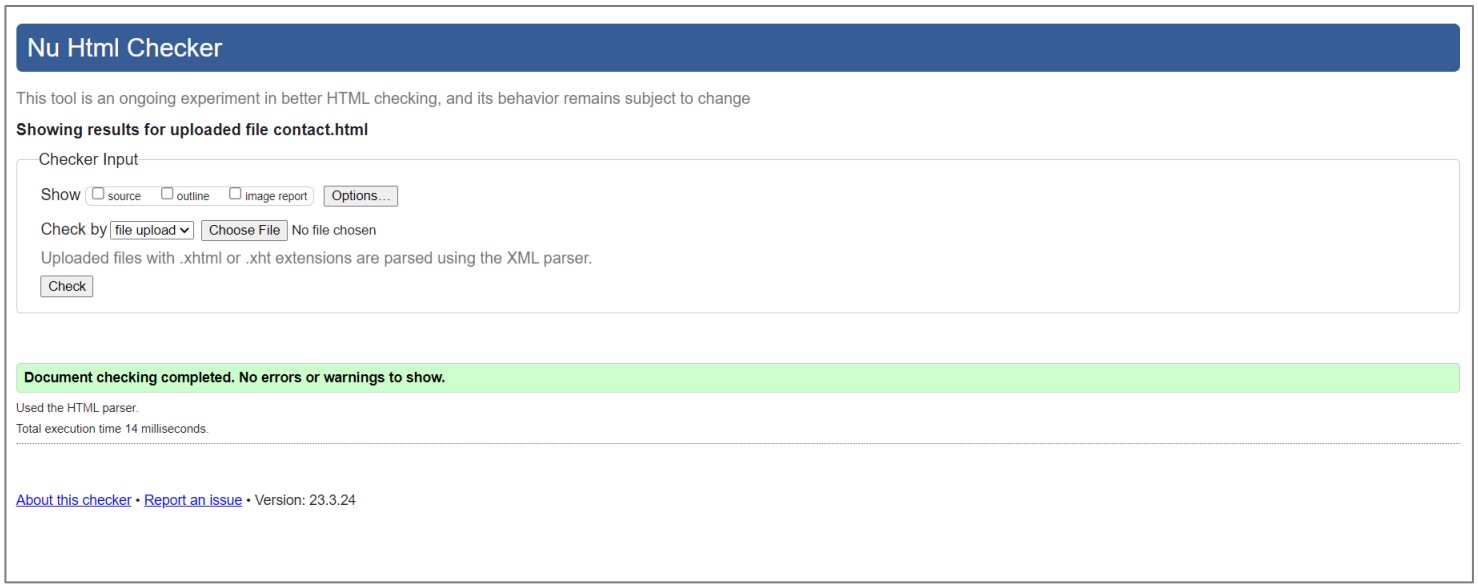


Image8: HTML validation of contact.html page.

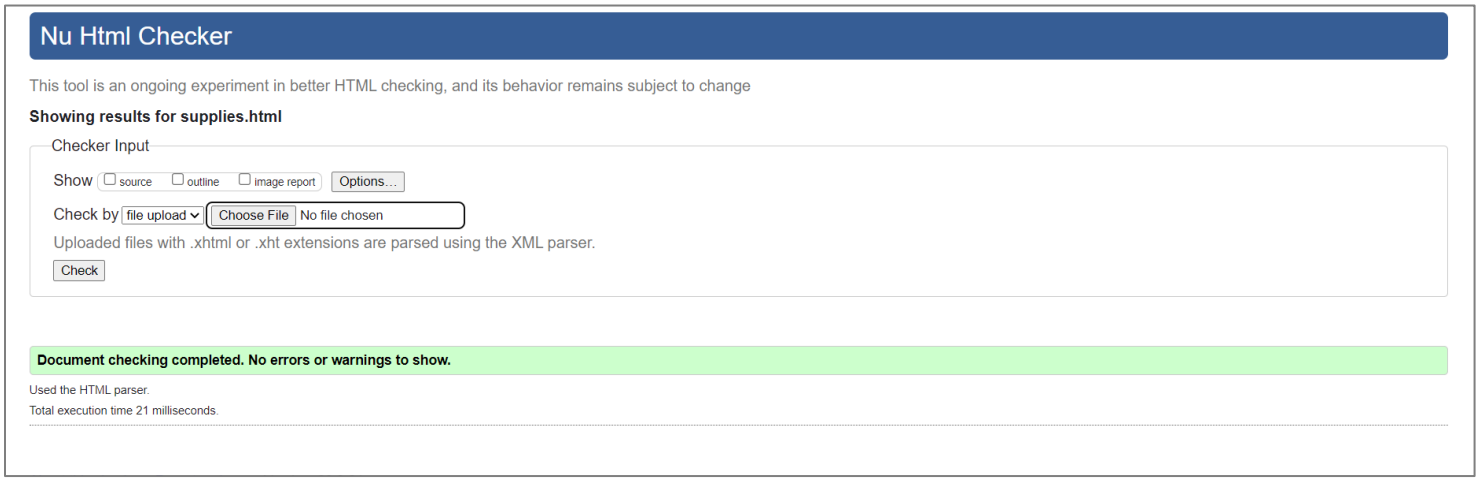
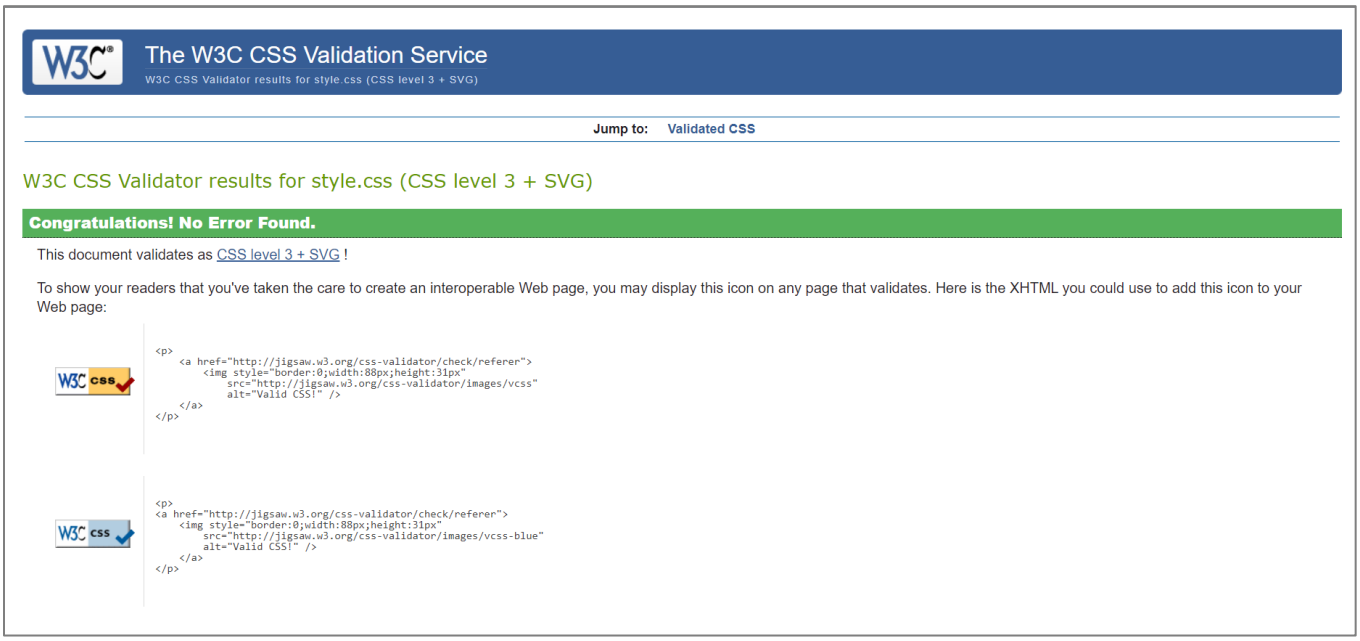


Image9: HTML validation of supplies.html page.



The W3C CSS Validation Service

W3C CSS Validator results for style.css (CSS level 3 + SVG)

Jump to: Validated CSS

W3C CSS Validator results for style.css (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as CSS level 3 + SVG !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<p>
    <a href="http://jigsaw.w3.org/css-validator/check/referer">
        <img style="border:0;width:88px;height:31px"
            src="http://jigsaw.w3.org/css-validator/images/vcss"
            alt="Valid CSS!" />
    </a>
</p>
```

```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">
    <img style="border:0;width:88px;height:31px"
        src="http://jigsaw.w3.org/css-validator/images/vcss-blue"
        alt="Valid CSS!" />
    </a>
</p>
```

Image10: CSS validation of style page.

Browser Testing Result

Tested on:

Windows 11/Microsoft Edge 111.0.1661.62

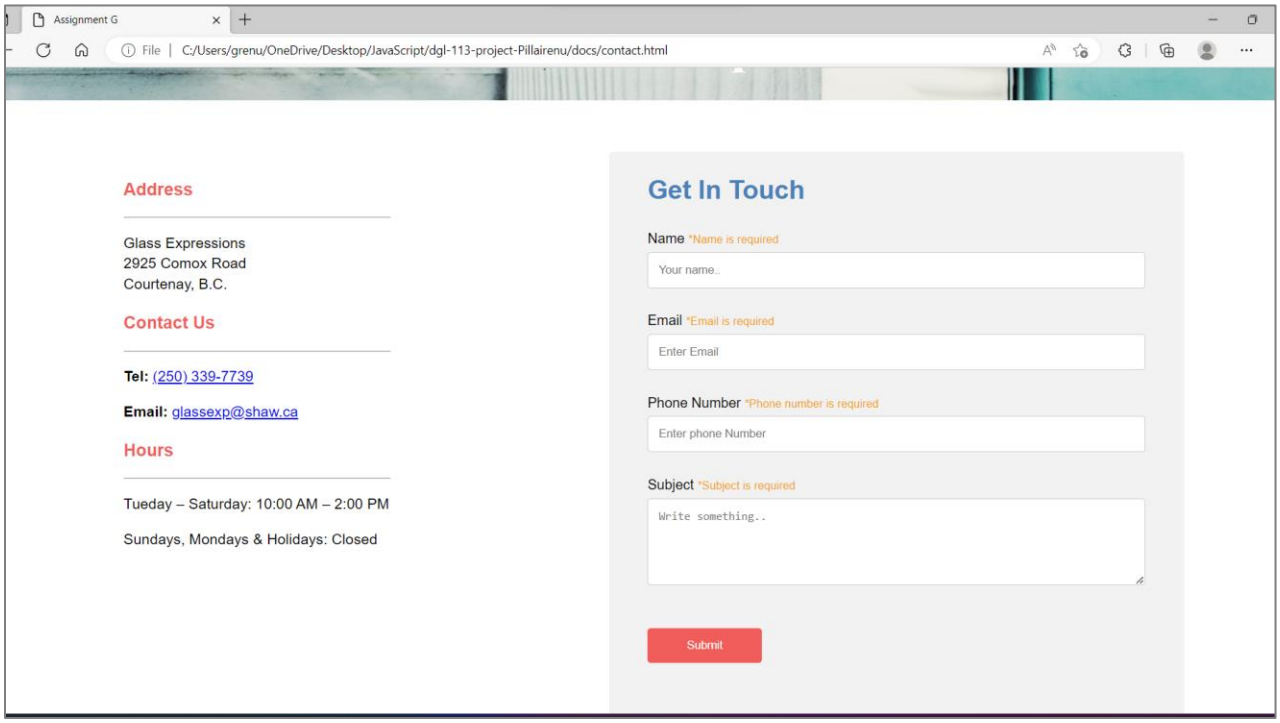


Image11: Feature1 test result on browser edge.

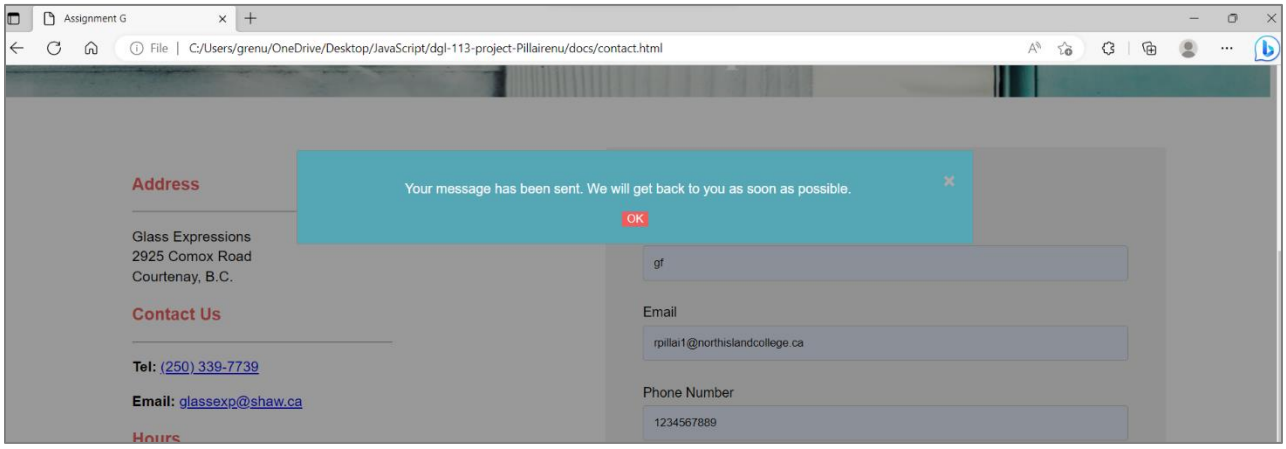


Image12: Feature2 test result on browser edge.

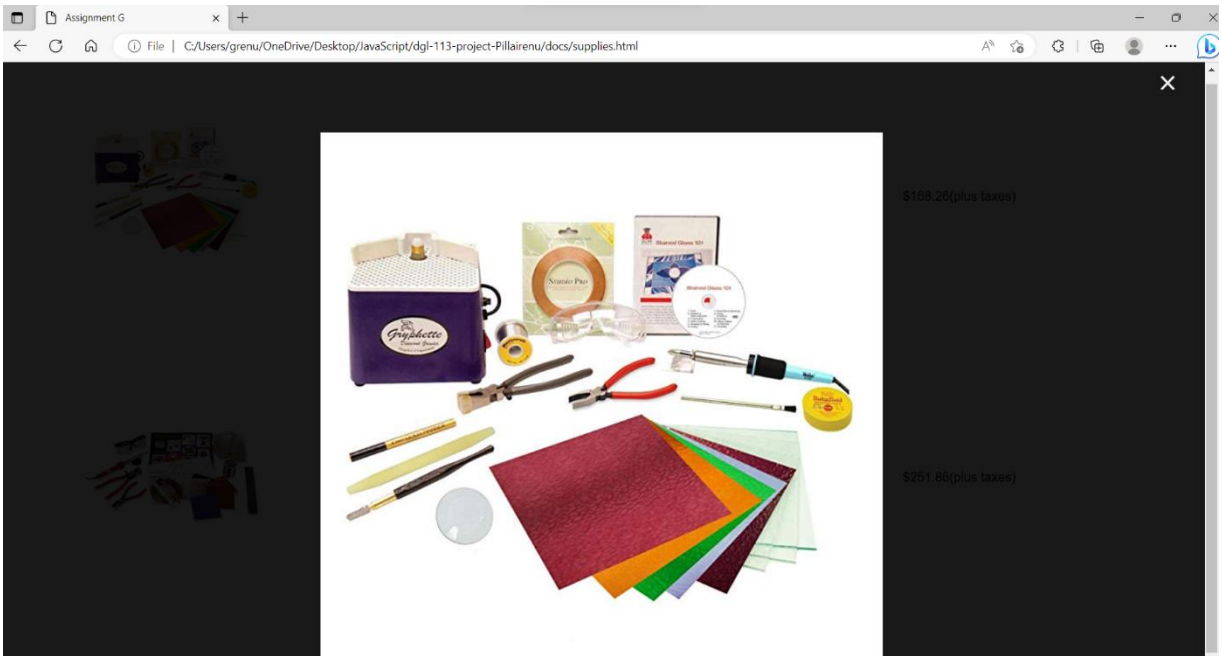
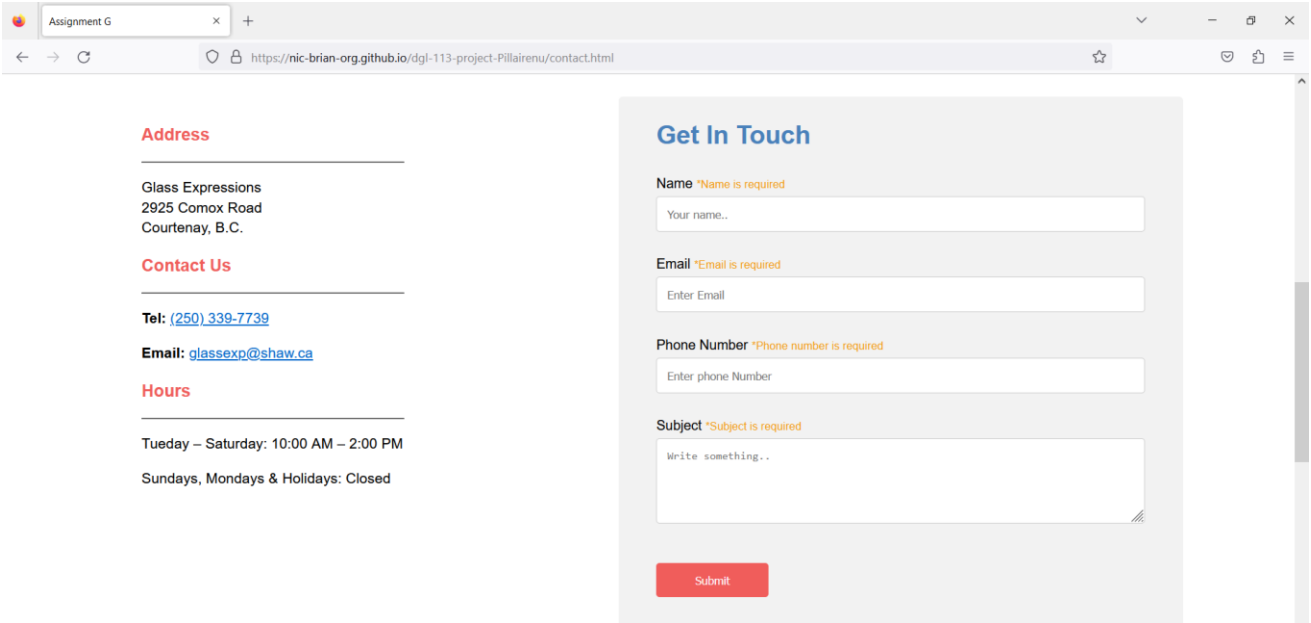
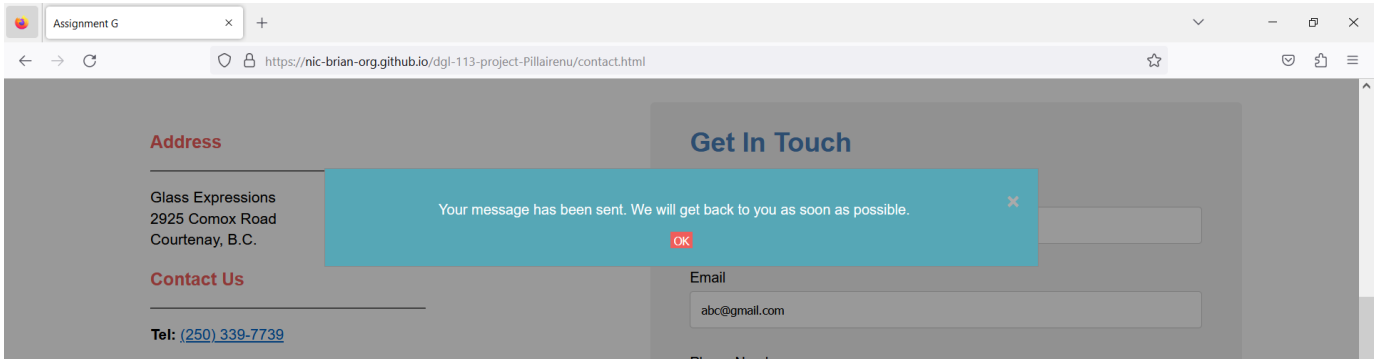


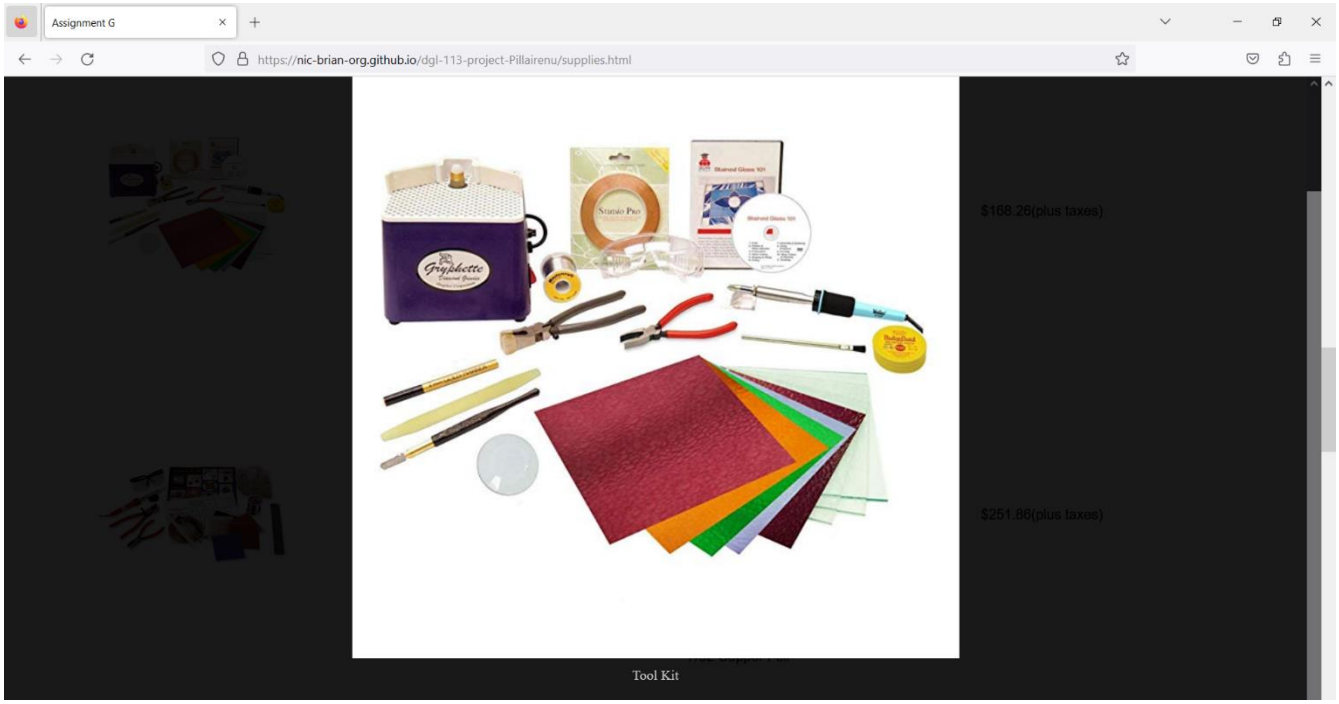
Image13: Feature3 test result on browser edge.

Tested on:

Windows 10/ Firefox111.0.1




Assignment G
https://nic-brian-org.github.io/dgl-113-project-Pillairenu/contact.html
Address
Glass Expressions
2925 Comox Road
Courtenay, B.C.
Contact Us
Tel: (250) 339-7739
Get In Touch
Your message has been sent. We will get back to you as soon as possible.
OK
Email
abc@gmail.com


Assignment G
https://nic-brian-org.github.io/dgl-113-project-Pillairenu/supplies.html
$168.26(plus taxes)
$251.86(plus taxes)
Tool Kit

Project Work Experience

It took me 10 hour(s) to complete the project.

From the project work I learned the following:

1. I learned how to do form validation using JavaScript.
2. I learned hoe implement model dialog box.
3. I learned how to make images larger when clicked on without having to leave the page using modal image.

I found the https://www.w3schools.com/js/ website helpful for completing my project work.

I didn’t encounter with any difficulties.