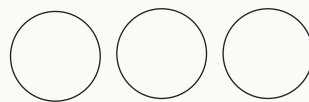


# INTRODUCTION TO WEB DEVELOPMENT



## Elements

The coding and development portions of the Taylor Swift website makeover were particularly difficult, but the finished product was a really good one.

Some of the items on the website were CSS elements that were applied as classes to the html pages. The html pages contained links, hyperlinks, a header, a footer, typography, and excellent web accessibility features like buttons and colors.

The website's final layout is simple and uncomplicated, making it easy for any unique users to figure out and comprehend. It uses excellent alignment, with everything in the right place and at the right distance from one another. Additionally, it utilizes columns and rows to improve alignment and organization. Additionally, top, bottom, left, and right space were used to create contrast between the menu, body, and footer.

## Changes

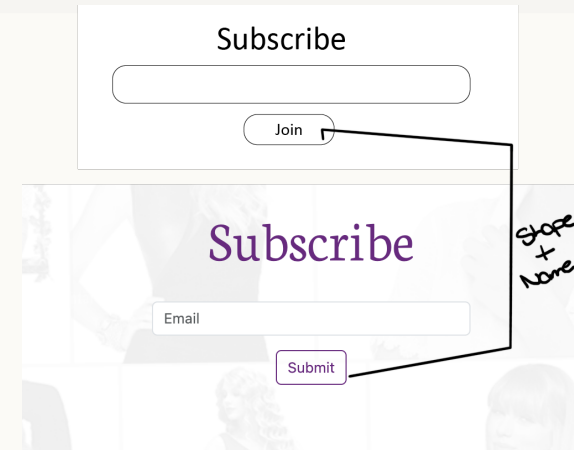
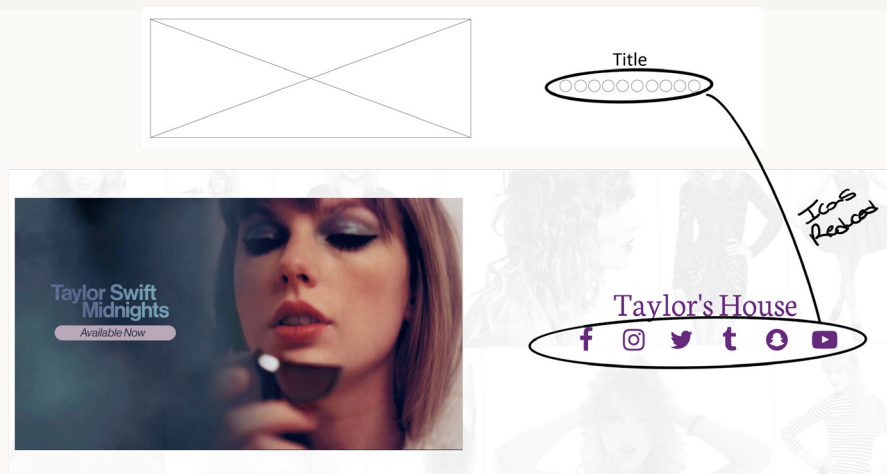
Comparing the website's finalized version to the wireframes created as part of the website planning process at the beginning, some pages had not changed but other pages had been planned differently, such as the "PaymentSubmitted.html" page, which is implemented differently than anticipated.

Changes were made to the menu. The names of the pages, which link to their own pages, have changed from their planned layout on the right side of the menu to the current location on the website's center column between the icons and logo. Another feature that distinguishes the website from the wireframes is the addition of an add-to-cart icon to every page, but when the website was originally designed, add-to-cart was only introduced to the shop pages, such as the products list and product pages.

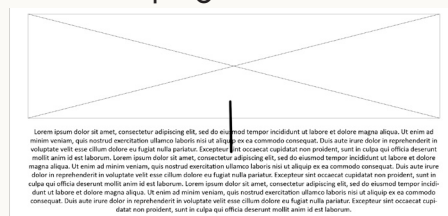
Artist Name



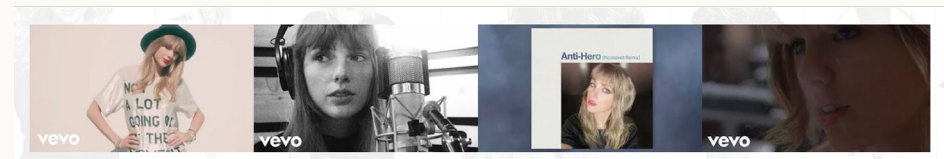
There are a few alterations on the Home page, but not many, particularly when compared to the high-fidelity wireframe of the original website. The icons on the right side of the home page were reduced since they were causing confusion on the website, and the graphics selected for the wireframe are different from those used on the website. Last but not least, join was changed to submit because it is more appropriate for the button in the subscription section.



The layout of the about page was changed because there was too much wasted space if the information had been positioned underneath the image, however it had been placed next to it on the right side of the page due to the image's portrait orientation.



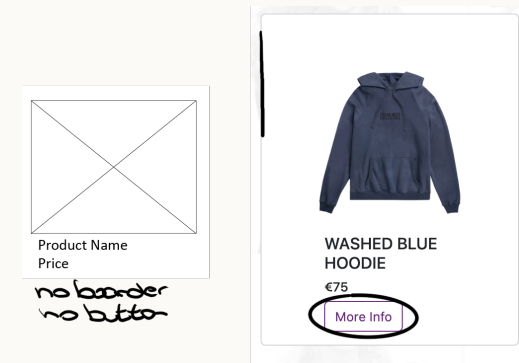
The Tracks page differs in that my intention in the wireframes was to add text, but upon more reflection, I realized that if I were a user, I wouldn't read the song description, so I didn't, included.



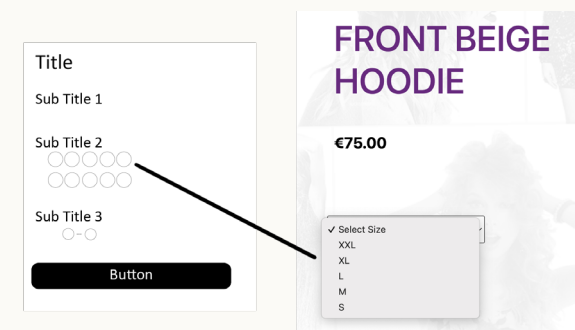
The blog page is one of the few where the change can be seen clearly. When I was creating this page, I wasn't sure how it was coming together and how the wireframe had evolved. I changed it to a course so it would be more presentable as it was developing and simpler to understand while still showing the photo at the same time.



My intention in the merch page was for the image or the product name to link to the product detail page on the Merch page, but I felt it would be more accessible and clearer, especially in terms of web accessibility, for people with special needs to understand it more quickly if there was a button.



The only difference in the product page is that the sizes section has a drop-down menu instead of buttons, which I had predicted would be simpler and more organized in the wireframes, after looking at several various shopping websites the dropdown was the most popular attribute.



The singer's email address was supposed to be entered in the upper portion of the contact page between the 'Get in Touch' and icons, but as I was coding it wasn't clear where it should go. The only other changes to the contact page are the larger email placeholder that was planned. The reset button, which is a new addition to the Java Script control, has been placed next to submit.

The image shows two versions of a contact page. The top version, titled 'Get on Touch Email Address', has fields for Name, Surname, Phone, Email, and a Message box, with a Submit button. The bottom version shows a revised layout with a large 'Email Address' field containing 'mdo@example.com' and separate Submit and Reset buttons. Arrows indicate the movement of elements between the two designs.

The only thing that has changed on the account page is the addition of reset buttons on either side of the sign in and log in buttons.

The image shows two versions of an account page. The left version has Sign In and Log In forms with fields for Name, Surname, Email, Password, and Confirm Password. The right version, labeled 'Revised', adds a Reset button next to the Log In button. A handwritten arrow points from the text 'The only thing that has changed...' to these buttons.

The addition of Billing Address placeholders to the payment details page makes it more aesthetically pleasing and user-friendly.

The image shows two versions of a payment page. The left version is the 'Payment Details' form with fields for Card Number, Cardholder Name, Expiry Date, and Security Code. The right version, labeled 'Billing Address', adds fields for Full Name, Email, Address, City, State, and Zip. A handwritten arrow points from the text 'The addition of Billing Address...' to the new fields.

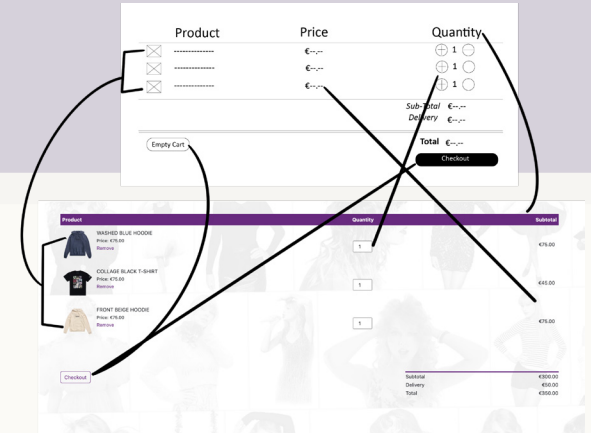
The image shows two versions of a sign-in page. The left version is the 'Sign-In' form with fields for First Name, Last Name, Email, Password, and Confirm Password. The right version, labeled 'Log-In', adds a 'Remember Password' checkbox and a 'Reset' button next to the 'Log In' button. A handwritten arrow points from the text 'The only thing that has changed...' to these buttons.

The last page, the one where payments are submitted, was also completed differently than anticipated, as I previously said. Before starting to develop this page, I realized that using the same layout as the add to cart would be redundant and could possibly confuse users.

The image shows a 'Payment Details' and 'Order Receipt' form. It includes a table for the order items with columns for Product, Price, and Quantity. The table shows three items, each with a price of €1.00 and a quantity of 1. The total is €3.00. Below the table, there are fields for Sub-Total, Delivery, and Total. A 'Home Page' button is at the bottom.

The image shows a 'Thank You' page with a large checkmark icon and the text 'Thank You'. Below the text, it says 'Your payment has been successfully submitted. Thank You.' and a 'Back Home' button.

One of the pages that was designed completely differently than how it was planned was the add to cart page. When I looked at the other website, the layout and elements weren't compatible with the one I planned, and in terms of online accessibility, users would be perplexed by this type of cart. This is why I changed the layout and elements on this page. But in my opinion, the finalized website's implementation is better planned.



## Issues

Although the coding process had originally been meant to be done within the wireframes, there weren't as many complications as there might have been.

The biggest problem I had was when I began incorporating java script into the webpage. In my opinion, Java Script is the most challenging step in the entire development process. Only contact forms, such as the sign in and log in, contact us, and payment details, utilised Java script. The Add to Cart component, the only thing on this website that doesn't function, was where I ran into trouble.

The initial image that appears on the home page, the one with the singer's face, was originally meant to be a video, but I was unable to make it function despite trying several code strategies like autoplay or inside a video property.

Another problem that had bothered me was that I couldn't find out how to make the website both screen and mobile friendly. In my view, there is a code that makes it both screen and mobile friendly, but I haven't worked it out.

Potential Improvements

There are several areas where things may have been done more effectively or added. As I stated in the concerns, my goal was to make the website fully functional. My most desired feature, the add to cart, was something I tried to implement but was unable to do. The logo, which is not the most distinctive and adventurous aspect, might have used some improvement. Spacing might have been used and applied better on pages like contact us, sign in, and log in. The distance between the tracks, which is provided on the track page, is a final item that should have been applied more effectively.

Web Terminologies

Classes were, in my opinion, one of the simplest methods or procedures throughout the entire coding terminology. I was able to organize such items and assign styles thanks to classes. Other terms that were helpful to me included the containers for rows and columns, even if some of those containers were confusing to me when I reloaded them from the web. Nevertheless, these containers were the simplest way for me to organize and keep everything where it was initially planned.

Test Cases

Test Case Type	Description	Test Step	Excepted Results	Status	Comments
Usability	Icons should link to the social media profiles – Should work properly.	User can click on the home icons or footer icons	Links will take the users on the social media page pressed.	Pass	
Functionality	Placeholder should accommodate an @ symbol.	Input the full email address.	@ Should be included.	Fail	Not all were failed. Failed once had been fixed.
Functionality	Arrow should take you to the next blog.	Click on the arrow.	Will take you on the next blog.	Pass	During this test it was figured that the carousel indicators were useless, so they were removed during this process.

Test Case Type	Description	Test Step	Excepted Results	Status	Comments
Usability	Button is linked with the original ticket's website.	Click on the More Info Button.	Links will give the user more information about the tickets.	Pass	
Usability	The image is linked with the YouTube music video.	Click on the Image.	Links will take the users on the music video page.	Pass	
Security	Password placeholder should be replaced by symbols instead of letters and numbers.	Enter your Password.	Content should be replaced.	Pass	
Functionality	The reset button should take off all the content.	Press on the Reset Button.	Content should be removed.	Pass	
Functionality	The sign-in or log-in or submit buttons should have all the placeholders filled up if not a message will appear.	Press the buttons.	A pop-up message will appear.	Pass	
Usability	All buttons are linked to the appropriate page.	Press on the buttons.	Another page should appear.	Pass	



## CSS and HTML Validation

Every validation process for CSS and HTML files is different. The validation website allows developers to enter their coding files, copy and paste their code, or even directly paste the URL of their website. By doing this, the developer can ensure that all CSS and HTML coding is correct and without any warnings, as incorrect coding may result in issues with the website while it is operating.

The procedure is the same for both validations.

The first decision is between Direct Input, File Upload, and URL. When using the HTML or CSS validation, if the URL option is selected, the URL of the corresponding HTML file must be entered. Choosing the html or CSS file will sync to the computer files if the file upload option is selected. Last but not least, if you select the Direct Input method, just copy the code and paste it into the fill-in field. Then click check.


The screenshot shows the 'Validate by URI' form on the W3C Markup Validation Service. It includes a header with the W3C logo and the service name. Below the header, there are three tabs: 'Validate by URI', 'Validate by File Upload', and 'Validate by Direct Input'. The 'Validate by URI' tab is selected. The form has a label 'Validate by URI' and a sub-label 'Validate a document online:'. There is an 'Address:' label followed by a text input field. Below the input field, there is a link 'More Options'. At the bottom right, there is a 'Check' button.

The screenshot shows the 'Validate by File Upload' form on the W3C Markup Validation Service. It includes a header with the W3C logo and the service name. Below the header, there are three tabs: 'Validate by URI', 'Validate by File Upload', and 'Validate by Direct Input'. The 'Validate by File Upload' tab is selected. The form has a label 'Validate by File Upload' and a sub-label 'Upload a document for validation:'. There is a 'File:' label followed by a 'Choose File' button and a text input field. Below the input field, there is a link 'More Options'. At the bottom right, there is a 'Check' button. A note at the bottom states: 'Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our information page on the W3C QA Website.'

The screenshot shows the 'Validate by URI' form on the W3C CSS Validation Service. It includes a header with the W3C logo and the service name. Below the header, there are three tabs: 'By URI', 'By file upload', and 'By direct input'. The 'By URI' tab is selected. The form has a label 'Validate by URI' and a sub-label 'Enter the URI of a document (HTML with CSS or CSS only) you would like validated:'. There is an 'Address:' label followed by a text input field. Below the input field, there is a link 'More Options'. At the bottom right, there is a 'Check' button.

The screenshot shows the 'Validate by file upload' form on the W3C CSS Validation Service. It includes a header with the W3C logo and the service name. Below the header, there are three tabs: 'By URI', 'By file upload', and 'By direct input'. The 'By file upload' tab is selected. The form has a label 'Validate by file upload' and a sub-label 'Choose the document you would like validated:'. There is a 'Local CSS file:' label followed by a 'Choose File' button and a text input field. Below the input field, there is a link 'More Options'. At the bottom right, there is a 'Check' button.

Once the error has been corrected, simply verify the code once again, and if there are no more errors or warnings, continue on to the next file.



The W3C CSS Validation Service

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


Jump to:
Validated CSS

W3C CSS Validator results for `StyleCart.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:



```

<?xml:space="preserve"
xmlns:cc="http://www.w3.org/css-validator/check/referer"
xmlns:img="http://www.w3.org/2002/2204-01/img"
cc:svg="http://www.w3.org/css-validator/images/vcss"
img:width="88px" />

```



```

<?xml:space="preserve"
xmlns:cc="http://www.w3.org/css-validator/check/referer"
xmlns:img="http://www.w3.org/2002/2204-01/img"
cc:svg="http://www.w3.org/css-validator/images/vcss-blue"
img:width="88px" />

```

(close the img tag with > instead of /> if using HTML <= 4.01)

# Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for uploaded file `index.html`**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Choose File](#) No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

**Document checking completed. No errors or warnings to show.**

Used the HTML parser

Total execution time 28 milliseconds.

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for uploaded file About.html**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by   No file chosen

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

**Document checking completed. No errors or warnings to show.**

Used the HTML parser.  
Total execution time 13 milliseconds.

However, since the visual code validator is picky and the web validator is more professional than the visual studio code validator, if there are any problems noted on the coding application [Visual Code] and not on the validator, do not bear them in mind and trust the validator more.