

Final Project – Documentation

Project Overview:

Title:

Final Project h24 – Diavola Website

Introduction:

My final project this semester has been designing and making a website for my band Diavola. The website contains features like a webshop and audio-player. As the band is a metal band, the theme for the colors have been chosen with this in mind, as well as trying to keep readability. The purpose for the website is for fans to be able to go there and shop for merch, have a preview listen to our, check us out on different social medias and see where and when our next gigs are (this is a feature I plan on implementing better later, but for what I had time for now, I added a link to a website where my band is signed up to which is made specifically for this (I have in no way any part of this website)).

Project Requirements:

Features list:

The websites features consist of:

Main page

- A merch shop created from an array, where a user can click on any merch card to get more info like sizes available, the option to add to cart and an option to select how many.
- The merch shop also has a filter function, where the user can filter which products they want to look at
- After the merch shop there is a discography section where the user can check out our latest release and preview our songs from said release. The songs in the audio player are coming from an array, allowing me to add more songs later

Cart:

- The cart features a counter that is visible on both the cart page and the main page. This counter is updated every time the user clicks “add to cart”, and it is also updated when the user removes something from the cart
- The products that are added to the cart are displayed dynamically in the cart products section. Here the user can see the images of the items they

have selected, how many, what size and the total price for that product. Included are also a button to remove the item from the cart.

- There is an input field where the user is urged to input their address details.
- There is a payment option there.
- There is an overview over how much the cart total is, this updated dynamically when the user removes an item from the cart.
- If there is no items in the cart when the clicks on the order button, they get a message telling them that the cart is empty and that there hasn't been any payment deducted. The input form is also reset.
- If there was items in the cart when the user clicked on order, then they get a message thanking them for the order, as well as emptying the cart and the input field.

Technology stack:

For this assignment I have used HTML, CSS and JavaScript. The CSS and JavaScript has been written in a modular style, so that it is easier to maintain later.

Design and planning:

Here is a link to the Figma prototype:

<https://www.figma.com/design/4yFxpjn9FQsjUTGigK0G/FinalProject?node-id=0-1&node-type=canvas&t=pDw9BZTw2YWOb8LI-0>

Development process:

One of my biggest challenges was figuring out how to open a modal filled with the information of the merch card that the user clicked on. I wanted this to be dynamic, but for a long time I couldn't figure out how to dynamically retrieve the merch info from the card the user clicked on.

What I eventually figure out was declaring an empty variable (let selectedCard = null) earlier in the document. I then made a function which passes "card" as an argument and from there I could import the information dynamically from what the user clicked on.

To get the modal to be easy to see, I gave it a classname which has its own styling, and upon closing the modal this class is removed.

I must admit that for coming up with how to open the modal based on what the user clicked on, I had to get help from ChatGPT. What I did was that I asked it how one can create a modal filled with information that is dynamically fetched based on what the user clicks on. Then I did my best to implement its

code into my own. The reason I did it this way was for me to understand what is going on, and why it works. If it worked and I didn't understand why, I asked ChatGPT to explain to me why it was working.

I also struggled with making the counter for the cart. I could get it to update one time, but the first couple of times I didn't manage to have it update often enough. My solution for this was to just add the update cart counter function to every function regarding the cart. So this function is run at the same time as add to cart, remove from cart etc.

I also struggled a bit with the filter-function. This was not understanding how it works, with where it picks up the value from and where to add the eventListener, where I tested out adding the eventListener to both the select element and the options inside it. What worked was adding the eventListener to the select element.

Testing:

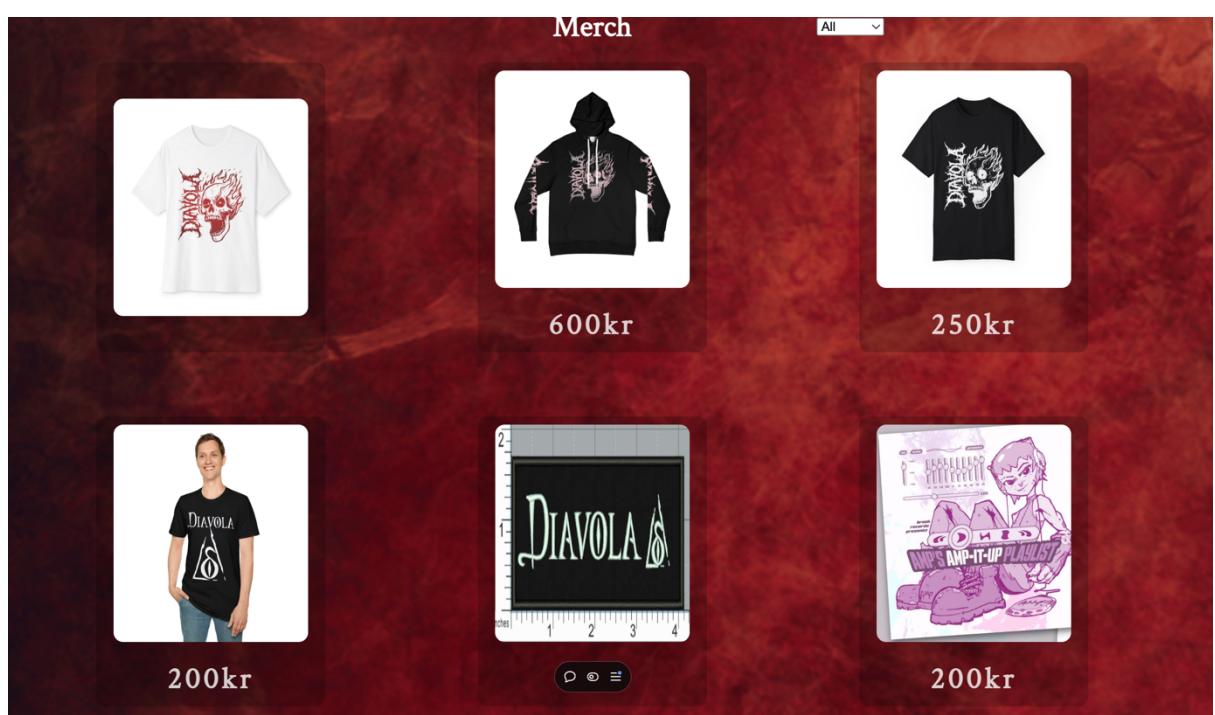
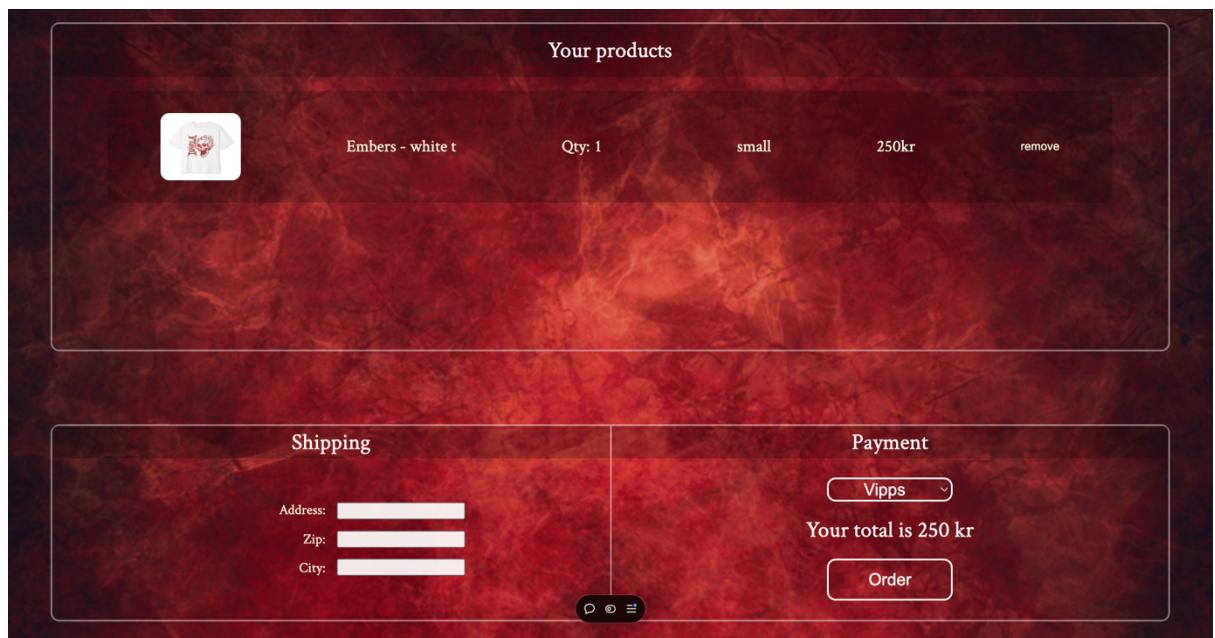
I tested the website for the most part in the browser, utilizing the "console.log" to hunt down bugs and to see what was going on when something didn't work as I intended. I also tested the responsiveness mostly in the browser, but I also tested it briefly on an actual phone, as I think it is a bit difficult to see exactly how big things are when I am using a bigger screen.

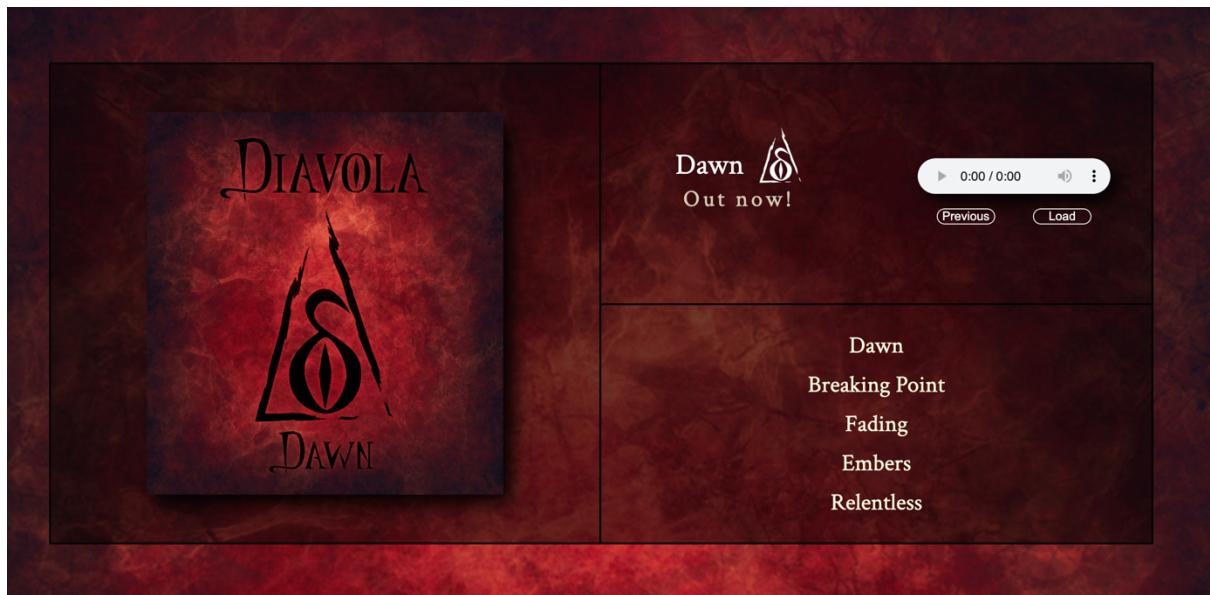
One of the more persisting bugs that I still haven't manage to solve is that for the very first merch card, the price is showing in its normal view, meaning that when the user hasn't clicked on it, they have no way of knowing the price of the product. When the user clicks on it though, then price is visible. I can't figure out why this keeps happening, because the price that is visible in the modal is fetched from the card directly, and all of the merch cards are created from an array using the .forEach method.

I also had a bug where sometimes the text in the cart telling the customer that their cart is empty sometimes wouldn't appear. After logging a whole bunch of things, it came from the cart array loading directly from the localstorage without checking if it was empty or not. This meant that when something had been added to the localstorage and then removed, the feature worked as intended, but when nothing had been added it didn't load an array. This function is based on an if-statement that checks if the length of the cart array is 0 or less.

Final product:

Screen shots:





How to use:

The website is intended to be used as a normal website; the links for the merch and discography takes you to different sections of the main page, whereas the links to concerts and SoMe (social media) takes you to external sites. The cart takes you to the cart page, where one can see the items that has been added to the cart, fill out an order form with address and order.

The audio player is functional add plays some snippets of my bands own songs.

Reflection:

What I learned the most from is probably how much work goes into designing and building just the frontend of a website. It took me longer than I anticipated, especially the merch section.

I also learned how one can fill a variable based on what the user clicks on, as well as how to loop through an array (in this case the audio array) using increments that are updated after a button is clicked.

If I had more time I would have added a picture carousel at under the audio player, and a footer with information about how to contact us. I also would have added a function where a user can updated the quantity of an item directly from the cart instead of having to remove the item and adding it again, which is what the need to do as of now.