Group project ID: 2282.

Username: 404\_Username\_Not\_Found

Online Shop Simulation

With shopping cart and credit card validation using JavaScript, (CSS and HTML for the layout and design component)

\*\*About\*\*

This is a project for the course “Introduction to programming” of the university of St Gallen.

The goal for this project was to create an online shop experience with a home page for item selection, a shopping cart and a payment (Credit Card Verification) option. This project started as JavaScript only but quickly became an HTML and CSS one as well in order to customize the aesthetics and features.

Given the complexity of this project for a beginner, templates, samples and tutorials for the required coding were used as a framework, which were then modified, integrated and expanded upon to fit the needs of my specific code, ideas and project as a whole. (All sources are linked at the bottom)

\*\*Pre-requisites\*\*

* Downloading the main folder (with the files) as it runs on local storage in the web browser.

\*\*Files\*\*

* Images (folder)
* Payment-Confirmation.html
* README.md
* About.html
* Cart.html
* Index.html
* Main.js
* Payment.html
* Style.css
* Online Shop Simulation - jim jacobs.docx

\*\*Instructions to run the code\*\*

1. Go to: <https://github.com/jimbajones/Online-Shop-Simulation>
2. Download the files/folder (click on ‘code’ drop down menu and then download zip)
3. Open the downloaded folder (Online-Shop-Simulation-main)
4. Double Click on the file index.HTML to open it in browser
5. Navigate the online shop (Description/Walkthrough below)

\*\*Description/Walkthrough\*\*

Once you have clicked the file/link mentioned in the previous section, you will find yourself on the home page. Here you will find all the items you can choose from to add to the cart. When the mouse hovers over the images/items, a button to add to cart appears. Each individual click will add an extra item to the cart. To test out all the functionalities of the cart, I recommend that at least two or three products are selected.

Once the selection has been made, you can scroll back up to the top where you will find three text boxes (Home, About, Cart). The Home button is self-explanatory and redundant at this point in time as we are still on the home page. The about button takes you to a separate page, where a duplicate of the “about” section found above has been printed. The third and final button is the cart button which will take you to your cart. Here you will find that all the items you clicked on appear with a number signifying the number of items you have selected. In the cart, for each individual item, you have the option of clicking the “+” button which will add an item, or the “-“ button which will remove one item. the “x” on the left will delete the whole row of items. Furthermore, the items are saved in local storage so won’t disappear (even on refresh) until you manually delete them. Here the “Home” button previously mentioned is useful if you want to go back and add other items to the cart. Once the desired item selection is finalized, you can click on the checkout button. This will take you to the payment page.

On this page you will find a form with inputs to be filled in, namely a name and credit card number.

Here the name is unfortunately there only for the aesthetics (more on this in the limitations). The credit card verification is a simplified mechanism and works according to the following rules:

Firstly, if dashes are inputted, (like this 4444-4444-4444-4444) they get removed from the string. If no value, letters alone or letters and numbers are entered, the prompt reads invalid. The number must be exactly 16 characters. The sum of all numbers must be greater than 16. At least two different numbers must be entered. The last digit cannot be an odd number.

Here are a few examples of invalid inputs:

1111111111111111

1111111111111110

2323232323232323

34536748958739098

234627893787627f

And some valid ones:

2323232323232324

5463782746574836

4444444444444446

If the number is invalid, a red text shows up notifying you of said error. Once a valid credit card number has been entered a green text shows up with the confirmation. once the green light has appeared, click on the “confirm payment” button, which takes you to the final order confirmation page. This last step is more so to have an as close to complete experience rather than a purely functional one (again more on this in the limitations).

\*\*Limitations/Improvements\*\*

The first improvement that could be made would in regard to the adaptive sizing. Although it works, my limited knowledge of how to implement adaptive sizing has its limits. When pushed to extremes (especially super small window size), text and objects start to fall out of alignment, and I did not manage to fully fix that. Also, visually, my method does not always render the prettiest results, as items tend to be oversized on bigger window sizes (especially on large monitors). Furthermore, there is a more accurate/complex way to size which comes up in my code (but was from sample/template codes) and although I was able to integrate it with my parts in a way that works, I did not fully understand it so visually, sizing sometimes looks slightly odd.

The second improvement would be to add the cart total/reminder on the payment page. Although this was not essential, it would be a more complete/real simulation.

Then there is the credit card validation. Having previously done projects on modular arithmetic and the Luhn algorithm, I was familiar with the concept of how credit cards work and read up on how to implement it for a more real validation. Unfortunately, I tried many different methods, but this was above my skill level and I was unable to figure it out in time, so I resorted to a simpler method.

To simplify things, I also used a stripped-down checkout form, so obviously this is an area with possible improvements.

Then there is the payment confirmation button, which in this case is only a link to the order confirmation. Again, here I tried multiple ways of disabling the button until a valid credit card number was entered. I managed to make some sort of disabling action work when text was not entered vs when text was entered but couldn’t figure out the validity part. Hence, I left this out of the code.

I also tried to transfer the inputted name from the form onto the order confirmation page, but considering the input and destination were on two separate html pages, despite having researched solutions, I was unable to make it work.

Finally, I didn’t manage to implement a way to delete the cart items (and thus the number at the top of the page) on the click of payment confirmation. Instead, for aesthetic purposes I found a pseudo loophole (which works visually only) which was to remove the reference to the JavaScript file that codes the cart number from the html file in question. That way the number appears as 0 on that page.

\*\*Sources\*\*

Item images: <https://www.pokemonprice.com/Gallery/98247308-89bc-4ec3-8826-f9d55d1be471>

Icons: <https://ionicons.com/>

Background/wallpaper: <https://wallpapercave.com/pokemon-hd-wallpaper>

Fonts: <https://fonts.google.com/specimen/Press+Start+2P?preview.text_type=custom&query=press&sidebar.open=true&selection.family=Press+Start+2P>

Tutorial : <https://www.udemy.com/course/javascript-shopping-cart-tutorial/learn/lecture/17873220#overview>

Template : <https://www.w3schools.com/howto/howto_css_checkout_form.asp>

Sample : <https://github.com/lezixb/credit-card-validator/blob/master/credit-card_validation_js.html>