I started this project with the idea of creating an ecommerce website to sell custom NFC business cards. The concept was to allow users to choose a card design, customize it with a logo and link, preview their design, and then add it to a shopping cart before checking out. I began by writing the HTML structure for the two preset card options — one matte black and one sleek white — and included "Add To Cart" and "Remove" buttons for each. I gave each product a unique data-id so I could target them later using JavaScript.

Once the layout was in place, I moved on to adding interactivity using JavaScript. I used localStorage to store the cart so that the selected items and their quantities would stay saved even if the user refreshed the page. I created an object called cart in JavaScript, which updated every time the user clicked "Add To Cart". If the item already existed in the cart, its quantity increased; if not, it was added as a new item with its name, price, and quantity. I also used .toFixed(2) to make sure all prices looked clean, showing only two decimal places like you'd expect in an online store.

I added logic for the "Remove" button so users could take items out of their cart. I used getElementsByClassName to loop through all remove buttons and attach event listeners to each one. When clicked, the product was deleted from the cart object and the updated cart was saved back into localStorage.

One of the main features I wanted was a live preview of the uploaded image so users could see how their card would look before submitting. I managed this by using the FileReader() method in JavaScript, which I discovered while reading Mozilla Developer docs and watching YouTube tutorials. I added an event listener to the file input field, and when a file was selected, FileReader converted it to a temporary image URL using readAsDataURL(). I then injected that image into the preview area of the page using innerHTML. This made the user experience feel more interactive and professional.

For form submission, I used a service called FormSubmit, which I found through YouTube. It lets you send form data directly to an email address without needing a server. I connected the final shipping form to my email by setting the form action to FormSubmit's endpoint and adding some hidden fields like _subject, _template, and _captcha to customize the email. To include a summary of the items in the cart, I created a JavaScript function that loops through the cart object, formats the details into a string, and inserts it into a hidden input field before the form is submitted. That way, when the email is sent, it includes the full cart summary along with the shipping information.

I also added a checkout button that, when clicked, sends the cart data to a local backend (localhost:3000) using the fetch() API and tries to create a Stripe checkout session. Even though Stripe wasn't fully set up for testing, I used this as a learning opportunity to understand how front-end code communicates with APIs and handles async operations.

Overall, I used the JavaScript techniques I learned throughout the project — including localStorage, .toFixed(), FileReader(), DOM manipulation, and form handling — to make the page interactive and functional. Most of what I implemented I found by experimenting, searching through Mozilla docs, and watching web development videos online and the module repo. Through trial and error, I was able to connect all the features together and complete the interactive parts of the assignment.