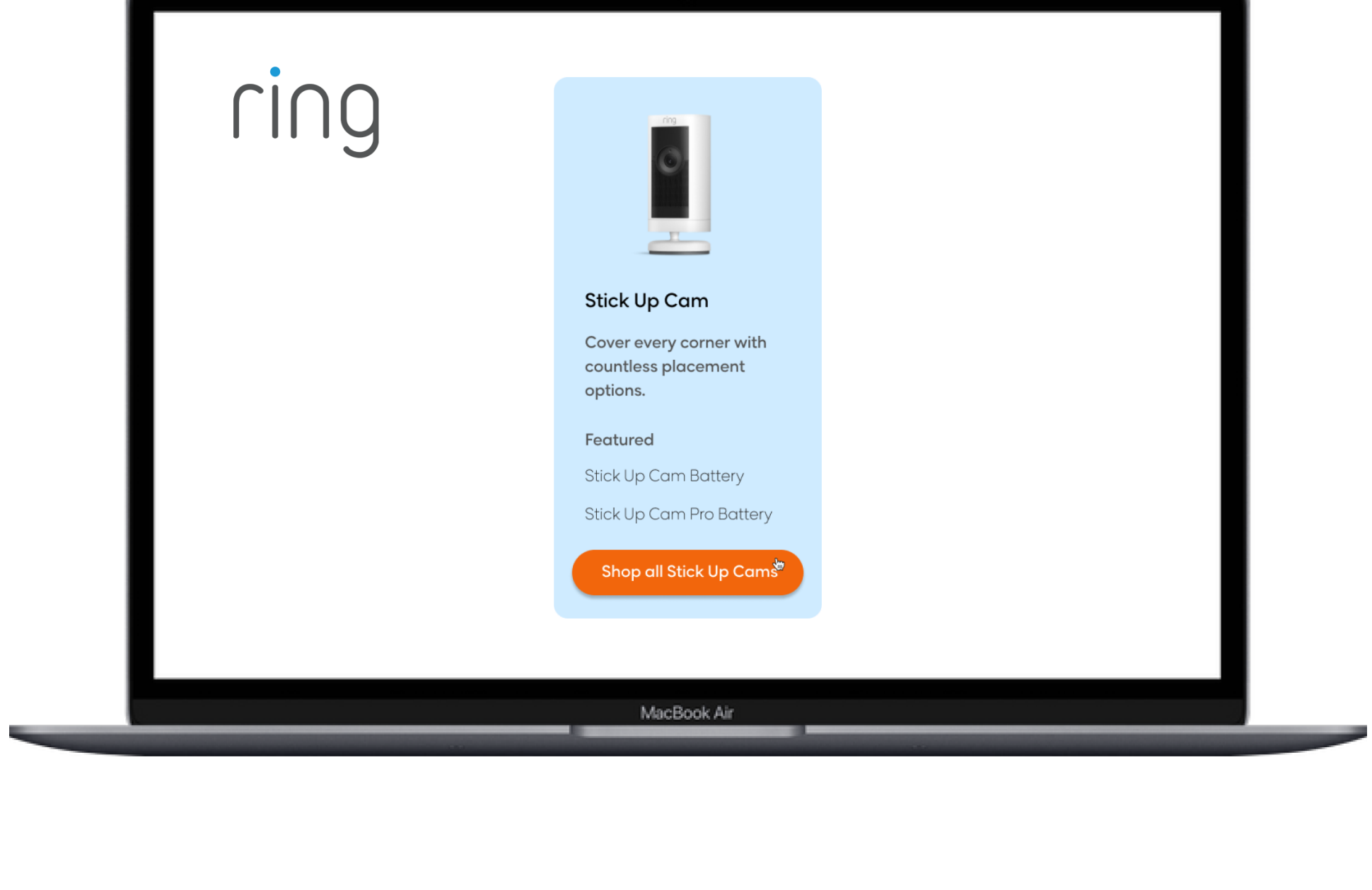


IDM 241 Assignment: Case Study



Project Overview:

The task was to browse the internet and find a site using a microinteraction that was functional but needed improvement. Once we found the chosen microinteraction, we had to code it and plan and deliver our changes. We had a timeline of 10 weeks to develop a meaningfully improved version of the microinteraction.

Context and Challenge:

After looking at various sites, I chose a microinteraction from the Ring website. Ring is a brand that specializes in smart home security products. Ring's smart home product line includes a range of devices designed to enhance home security and monitoring. The specific microinteraction was a product panel for the ring stick-upcam. The microinteraction is accessible by hovering over the products tab on the navigation bar. A list of products drops down with highlighted products shown on clickable panels.

The first challenge was understanding what a microinteraction is and its elements. A microinteraction consists of triggers, rules, feedback and loops and modes. A trigger is an event that initiates the microinteraction, such as click tap, hover, etc.... Rules determine how the interaction works. Feedback refers to the visual or behavioral responses users receive when interacting with a specific element or feature in a digital interface. Loops are repetitive behavior patterns within the microinteraction, and modes refer to new content or states elements can transition to, typically providing user feedback based on their specific interaction.

The Alpha(Original) Modle

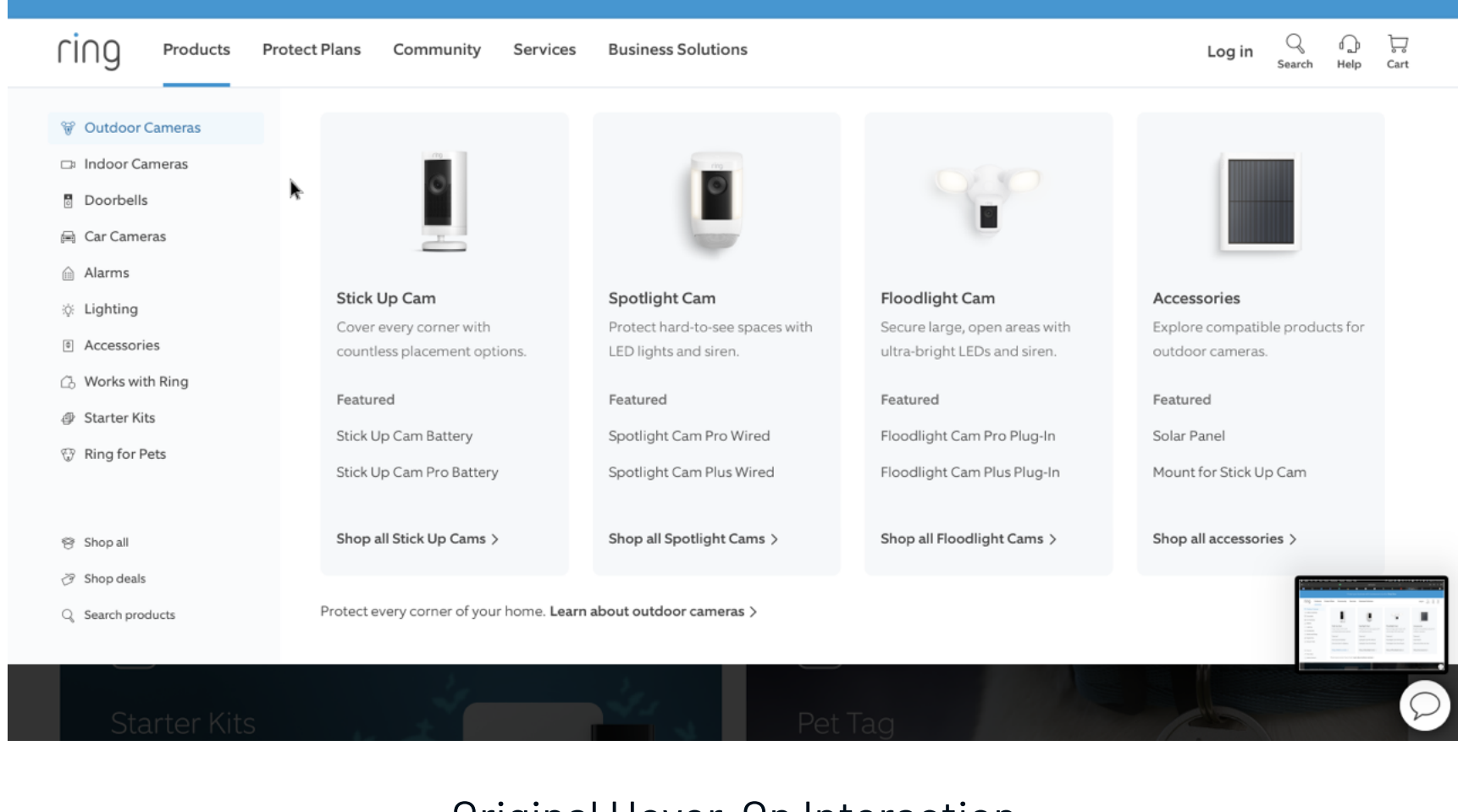
First, I had to code the existing microinteraction and identify its defining features. The original Microinteraction was simple. When a user hovered over a panel, the following events occurred:

- Cursor changes from an arrow to a pointer.
- Image of product grows and expands.
- Bottom call to action statement changes color.

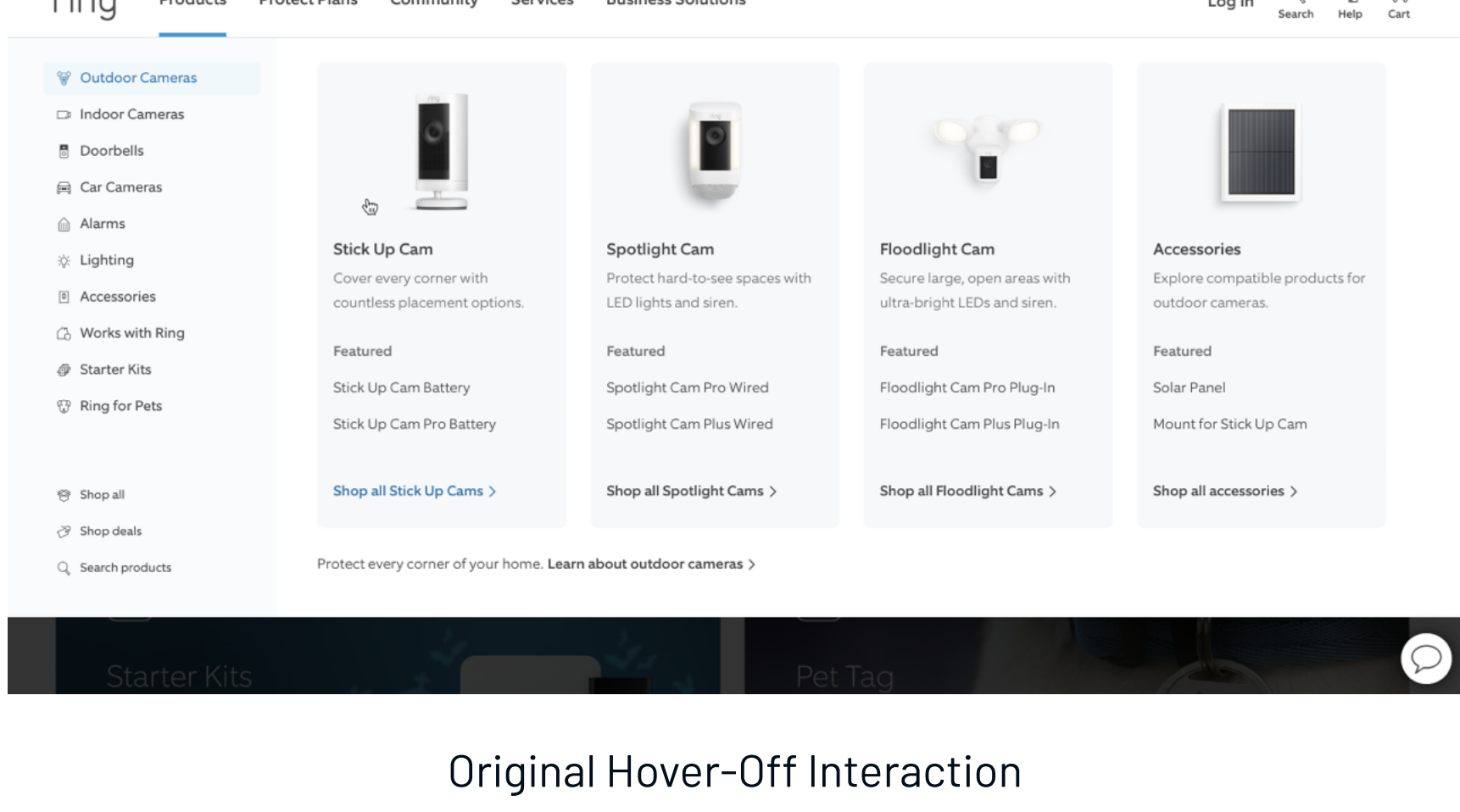
And when a user hovered off the panel:

- Image shrinks back to its original size.
- Call to action statement changes back to its initial color.
- Cursor changes from a pointer to an arrow.

This microinteraction was a simple way to prompt users to click the panel to learn more about the project. It was standard, but I wanted to make it more engaging and reflective of Ring's branding.



Original Hover-On Interaction



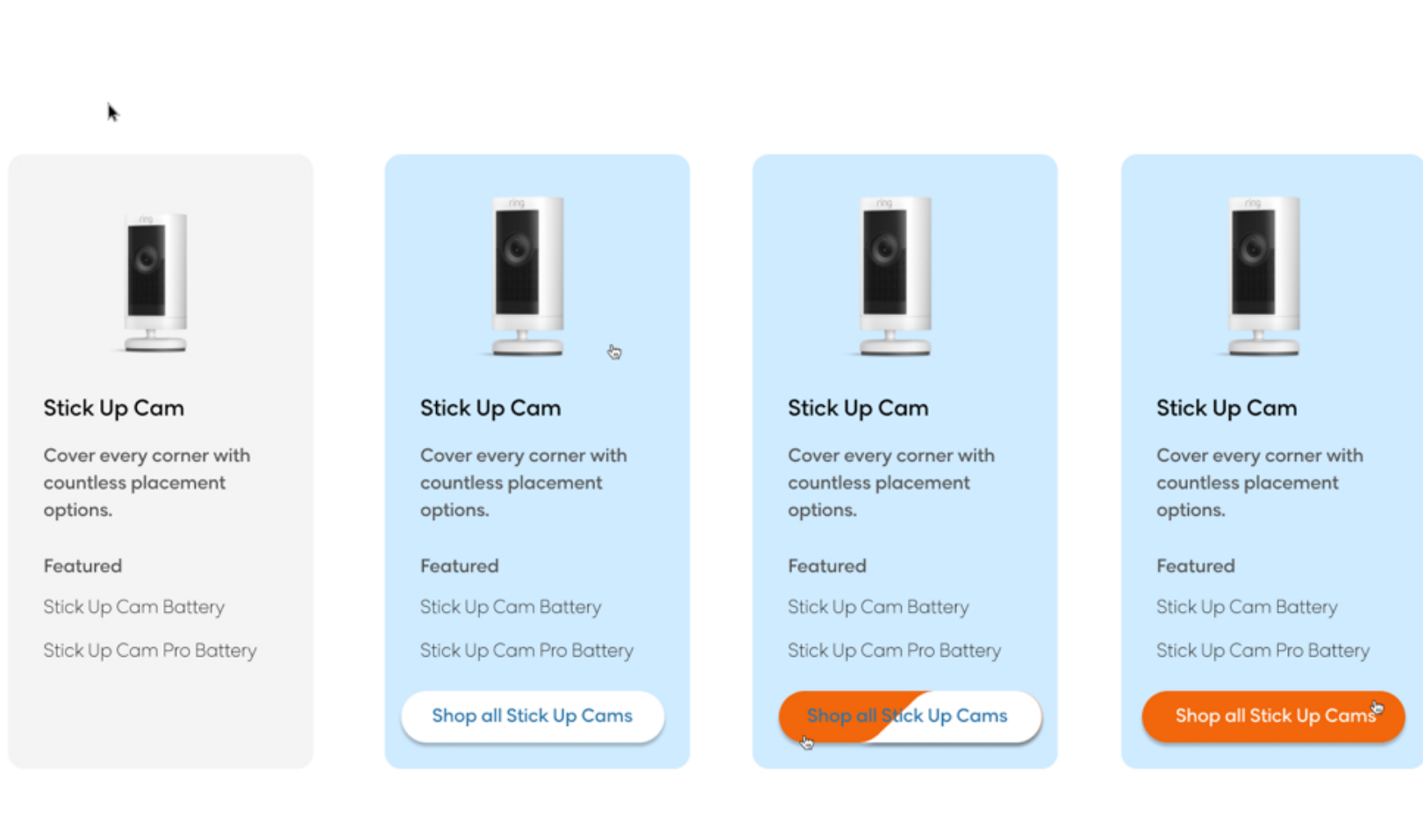
Original Hover-Off Interaction

Utilizing Chat GPT

After identifying the target microinteraction for my project and its elements, the next hurdle was applying my HTML and CSS knowledge from prior terms to code the original microinteraction. It was a definite forced refresher. One of my most frequently used resources was Chat GPT, which I had never used for coding prior to this class. At first, I rejected the noting of using AI for code; I wrongly assumed that using it wouldn't help me learn the content by generating it for me. However, I realized it was time to look for alternatives when I found myself four YouTube videos and six website tutorials deep- all consisting of completely different and convoluted ways to make flexboxes the same length regardless of content. Once I utilized chat GPT, my problem was solved within minutes, and I learned it was a powerful tool that could be used and built upon to suit my needs.

The Beta Build

The next step in my development was transforming the alpha (original) microinteraction into its beta form. I had to improve upon the alpha and add additional elements. I changed the background color on hover to a light version of Ring's signature blue color and transformed the call-to-action text to a clickable button that transitioned to Ring's signature orange with a left swipe transition on hover. Everything else remained the same.



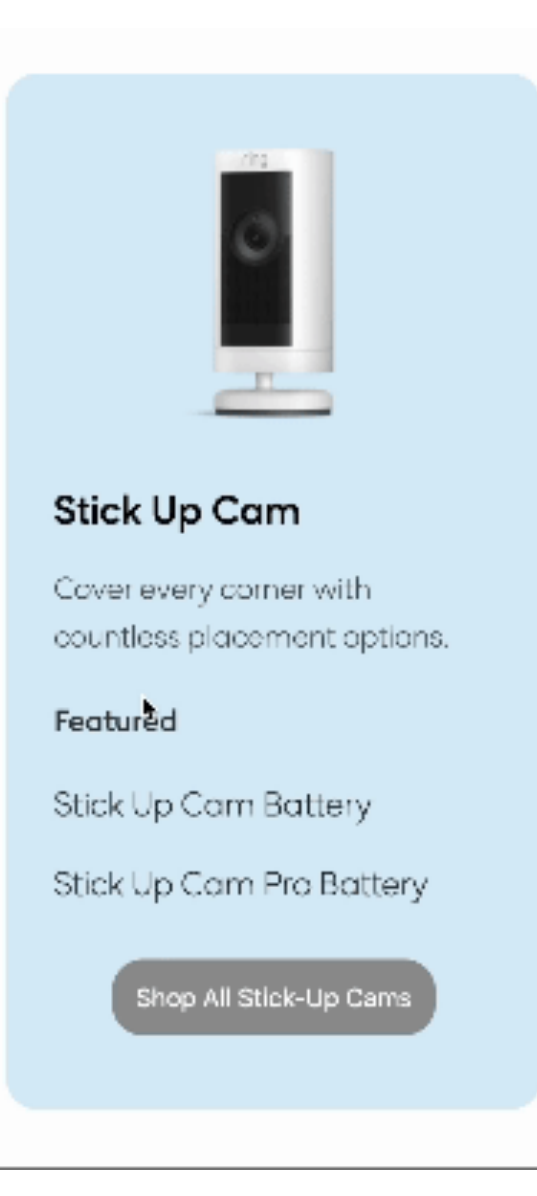
Beta Challenges

When building the beta, I was coding things I had never encountered before. I was learning entirely from scratch. I consistently encountered a problem in which the button text was covered by the orange color when the user hovered over the button. I repeatedly tried to find new ways to ask chat GPT to fix my problem, but nothing worked. I finally realized I had to let go of the notion that ghat GPT could fully solve my problem and began experimenting. I realized that I had not defined the content for the before state of the button. I coded the orange color transition by creating a before state for the button where the color was orange and having that state revealed on hover. Although the button had text, I had to name content with the following command:

.shop-button::before { content: "Shop All Stick-Up Cams";}

Leading Up: The Final Build

Although I had improved and added upon the fundamental aspects of the microinteraction, I still had challenges to overcome before the final build was complete. Another major roadblock I faced was still associated with the color change transition on the button. Although the text had appeared, it was positioned at the top of the button container, and I couldn't get the text to center through traditional methods. I tried to justify and align the content, text, items to thew middle but nothing worked. Chat GPT didn't entirely understand the intricacy of my problem and gave me a default answer that was unresponsive in the code. Finally, after days of scrutinizing, I realized all I needed was top-padding to center. It was the simplest fix, but it reminded me that when problems seem simple, the answer usually isn't very complicated.



The Final Build:

The final build came a considerable way from the alpha model. Now, when the user interacted with the panel, the following occurred:

Hover On Panel

- Cursor changes from an arrow to a pointer.
- Image of product grows and expands.
- "Shop-All" call to action button appears.
- Background quickly fades into a light blue.

Hover Off Panel:

- Panel quickly fades back to its original color.
- Image shrinks back to its original size.
- "Shop-All" Call to action button disappears.
- Cursor changes from a pointer to an arrow.

Hover On Shop-All Button:

- Orange color slides in across button
- Button texts changes color from blue to white.

Hover Off Shop-All Button:

- Orange color slides back off button.
- Button returns to white color.
- Button text changes back from white to blue.

Click the Shop-All Stick-Up-Cam Button

- Button translates down 8 pixels.
- Button subsequently translates back upto original position when click is complete.
- Button remains orange color while cursor stays hovering over it.

Results and Takeaways:

I believe my final build successfully completed my goal for the project. I created an improved and more interactive version of Ring's product panel. I made my additions reflect the brand, I created an overall more enjoyable experience, and I learned a lot throughout this project. I discovered the vast capabilities of CSS and its ability to animate things without the need JavaScript. I also learned how to use AI to my advantage to save time and brain power when de-bugging. Lastly, this project was a reminder of how much details matter. I poured over some issues for days when the problem ended up resolving with one line of code. This wasn't an easy feat, but I feel like a stronger designer and more in-tune with the importance and intricacies behind all the microinteraction we come across every day, and I feel confident that I will be able to effectively implement this knowledge in my future projects.