

# Homework for Customer Success Specialist

Felipe Rosario Dos Santos

## 1. Helping customers with design modifications using CSS

**Note:** All of the responses below are assuming that the CSS will be applied directly to the popup's div. Since I don't have details on where exactly in Shopify it would be applied, it might be necessary to specify this before applying the CSS to the components.

### 1.1 - Can you please change the font color on my pop-up buttons to green (#00ff00)?

**Answer:**

CSS code based on **data-cy** attribute

```
button[data-cy="add"]{  
    color: #00ff00 !important;  
}  
  
button[data-cy="continue-button"]{  
    color: #00ff00 !important;  
}
```

### 1.2 - Can you please make the upsell product name underlined so it looks more like a link?

**Answer:**

CSS code applying text-decoration to the grid items

```
.Product-Title-grid {  
    text-decoration: underline !important;  
}
```

### 1.3 - Can you please hide the price of the parent product?

**Answer:**

```
.Product-Price {  
    display: none !important;  
}
```

#### **1.4 - Can you please increase the font size and use sentence case instead of all caps?**

**Answer:**

```
.Product-Title-grid {  
    font-size: 18px !important;  
    text-transform: none !important;  
}
```

#### **1.5 - Can you please make the upsell product images bigger?**

**Answer:**

This one is tricky because I'm not sure if I should exceed the grid size or if I should adjust the grid itself. So, I decided to apply a zoom effect on the image while keeping it inside the grid. If it was needed to increase the image size without using zoom, I'd need to adjust the grid accordingly

```
.upsell-list-item-left-section-grid img {  
    width: 100% !important;  
    height: 100% !important;  
    object-fit: cover !important;  
    transform: scale(1.2) !important;  
}
```

Basic explanation of my solution: the scale parameter in the transform property increases the image size. The object-fit: cover ensures the image stays fully contained within the grid without distortion, this way, the image appears larger while remaining within the grid's bounds

## **2. Customer support via in-app chat**

### **2.1 General "app is not working" inquiry**

**Merchant:** I am using your app Candy Rack but it's not working on my store.

**Reply:** Sure thing, I will report it to our developer to look into that.

**Merchant:** Thanks but I can't wait. I will look elsewhere.

**Your reply:** ?

**Note:** All of my replies below assume that support is being provided through the chat widget, not via email or ticket.

## 2.1 - My reply:

No problem, could you please provide more details about the issue you're facing? Are there any error messages displayed?

It would be helpful if you could describe the actions you're taking when the problem occurs, for example, during the app installation, when creating your first upsell offer, while customizing, etc. This will give us a clearer idea of how we can assist you in resolving the issue.

For your reference, **Candy Rack** has documentation available that might help guide you through the process. You can find it [here](#).

## 2.2 Upselling hidden products

**Merchant:** I want to upsell an extended warranty with Candy Rack, but I don't want it to be visible in my store like other products.

**Your reply:** ?

## 2.2 - My reply:

Hello, to achieve this, you can use Shopify's "**Unlisted**" product status for your extended warranty. This will hide it from collections and search on your store, but it can still be upsold through **Candy Rack**.

Here's how you can do it:

1. Create your extended warranty as a product in Shopify.
2. Set the product status to **Unlisted**. This will make it invisible on your storefront, but it can still be added to your upsell offers in **Candy Rack**.

That way, the warranty won't be visible to customers browsing your store, but it can still appear as an upsell when appropriate.

For more information regarding this new "unlisted" status on products, please refer to [this link](#).

## 2.3 Product targeting question

**Merchant:** Hi guys, I am using your app **Gift Box** and it's been working great so far. But I would like to give away a free gift only to customers located in the US. How can I do that?

**Your reply:** ?

*Background information: This is not possible at the moment but we are already working on it. It should be ready in around three weeks.*

## 2.3 - My reply:

Hello, unfortunately, it's currently not possible to offer a free gift based on location. We're working on this feature, and the ETA is about **3 weeks from now**.

Please feel free to keep an eye on our release notes for updates: [Gift Box Release Notes](#).

Let me know if you need anything else!

## 2.4 Communication with the merchant

You're chatting with a merchant and then realise that one of your teammates gave them incorrect information. How do you proceed with the merchant and how do you handle the situation with your teammate?

Your reply: ?

### 2.4 - My reply:

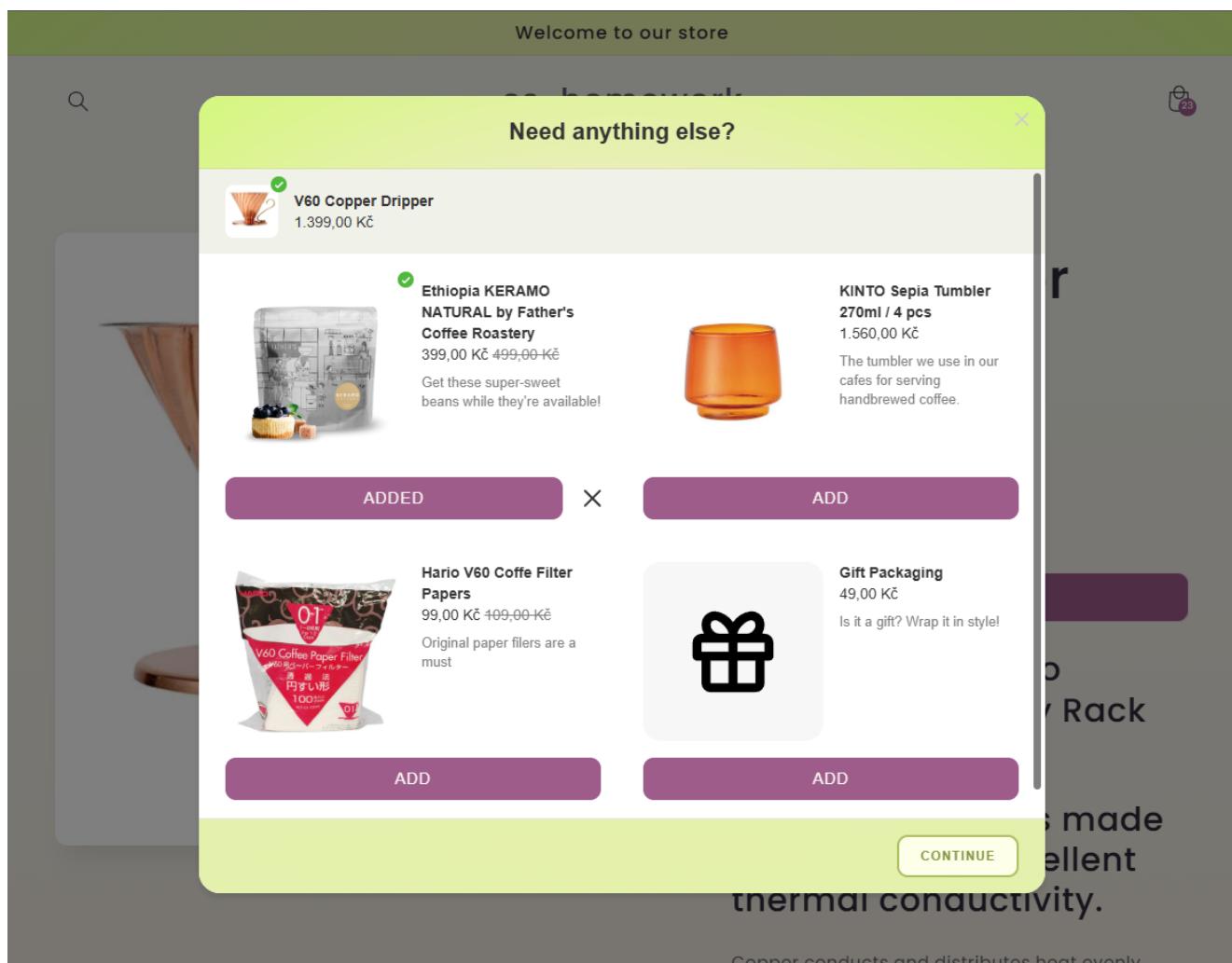
In this situation, I would first apologize to the merchant for the incorrect information provided earlier and for the confusion caused. I would then provide the correct information and ask if there's anything else the merchant needs assistance with.

As for my teammate, I would inform them that they provided incorrect information, and I would share the correct details with them. I would also ensure that they are aware of the correct information moving forward to avoid any similar issues in the future.

## 3. Styling the pop-up for the particular store

For this exercise, I decided that the ideal approach would be to copy the style of each similar property from the main page, for example, the header, footer, buttons, etc. The idea was to make it as similar as possible, but without adding unnecessary properties. The final result is shown in the image below, adding rounded borders, softer fonts, gradients, while maintaining the originality of the layout.

CSS file on GitHub [here](#)



## 4. Frontend - JavaScript/React

[Fix file on GitHub here](#)

Brief explanation of how I arrived at the fix: basically, I looked at the events triggered when clicking "add to cart", navigating through the call stack to see which method was called, until I reached the React component that had the method handling the add to cart event.

From there, I discovered the function that handled the variable storing the selected color variant. The line **const d = o || await B(t) || Bt() || qt(a);** where **B(t)** is the function that implements the exercise function, so I created the function to properly handle the product color selection and return it to this variable.

The screenshot shows a Shopify storefront for 'HIRING TEST STORE'. A product page for a 'Men's Running T-Shirt' (150.00 Kč) is displayed. A 'JUST ADDED TO YOUR CART' message is shown at the top. Below the product, a 'Recommended' sidebar lists the same 'Men's Running T-Shirt' and an 'ADIDAS | CLASSIC BACKPACK'. A red circle highlights the first item in the sidebar. The browser's developer tools are open, showing the call stack and the source code for the 'candyrack-popup-addtocartButtonClick' function. The source code includes imports from 'candyrack-popup-chunk-0.js' and 'candyrack-popup-chunk-1.js', and handles asynchronous operations like promises and async functions to manage the product selection and add-to-cart logic.