# Duplicate a Checkout Button on a Cart Page

To increase chances of conversion on your site's cart page, you may want to increase the visibility of the checkout button by duplicating it elsewhere on the page. You can accomplish this goal by creating a custom Duplicate Element action that also retains its bindings to the cart form submit.

This use case demonstrates exactly how to accomplish this goal with a custom action that duplicates the **Check Out** button below the site's "Shopping Cart" page heading.
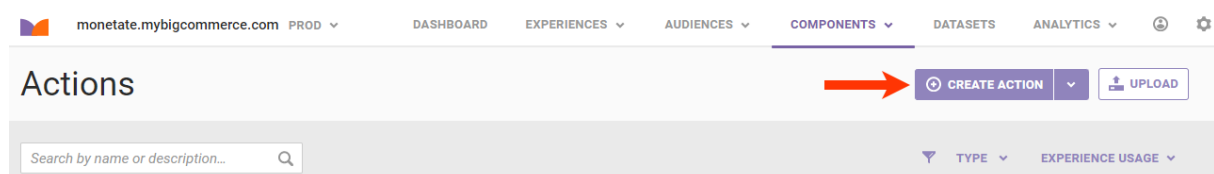
Creating the Duplicate Element Action

Follow these steps to create the custom Duplicate Element action.

1. Click **COMPONENTS** in the top navigation bar, and then select **Actions**.
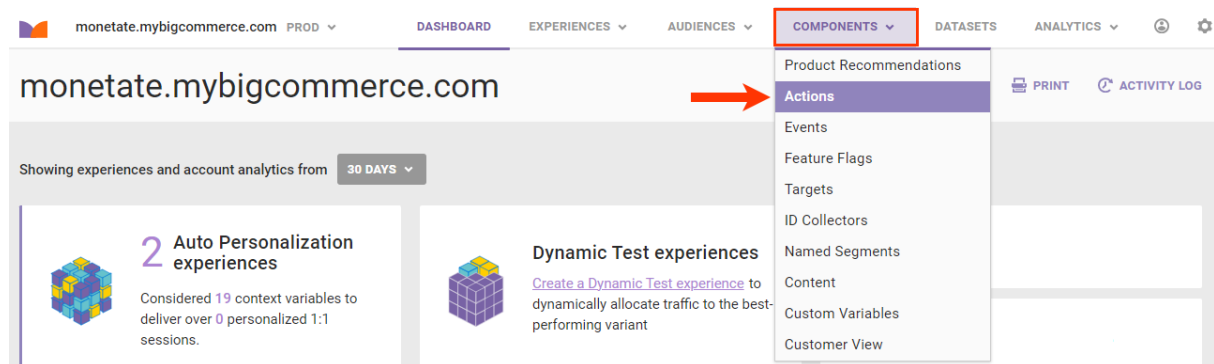
2. Click **CREATE ACTION**.

3. After Action Builder loads on your site, add any product to the cart, and then navigate to the cart page.
4. Use your browser's developer tools to inspect the checkout button's code.

> ℹ️ In this use case, the **Check Out** button to be duplicated is part of a form. That form contains all the product, price, and quantity data that must be submitted to complete the checkout process.
>
> The form action is executed when the form submit event fires, sending the form data to the form handler. The action attribute on the form denotes the URI of the form handler, which in this case is a server-side script at /cart. Since the submit event is triggered by the form and not the button, the submit button must be duplicated within the form element.
>
> To ensure the button remains functional, it must remain within the context of this form

element, therefore it can't be duplicated to any area outside of the `<form>` element if it is to function as the `submit` for that form.

5. Return to Action Builder to click **Duplicate Element** on the **Insert Content** tab.
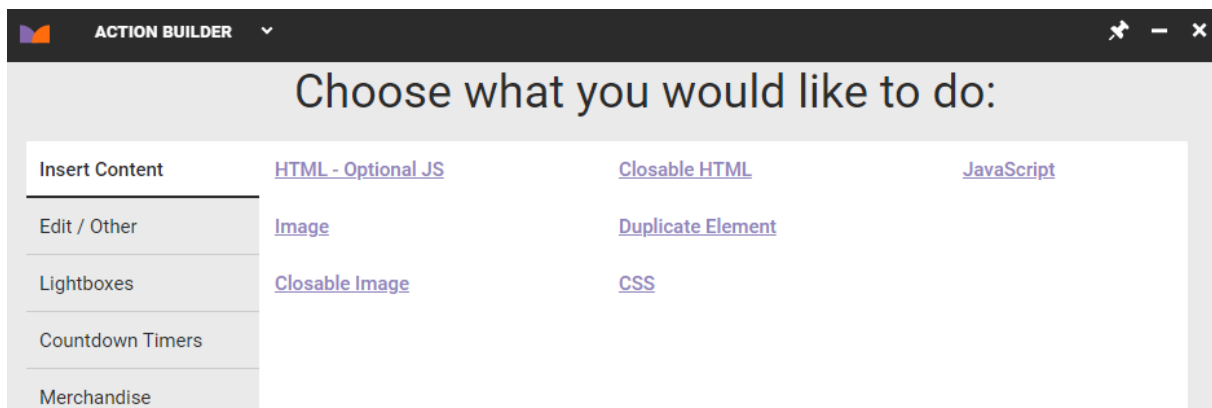
6. Find the selector for the **Check Out** button, and then click the forward arrow.

7. Identify where the cloned button should appear, understanding that you can refine the element selector on the **Details** tab to ensure it falls within the form parent. Click the forward arrow.

8. Ensure **Enable Preview** is set to **ON**, and then confirm that the cloned element appears where you want it and works. If it does, move on to step 9. If the action preview doesn't work as intended, take the following actions.

ACTION BUILDER

Details	**Element To Clone** `#shopify-section-cart-template > div > div.page-width > form.cart > div...`
Creatives	*Element to clone.*
Styles	**Relative Element Selector** `#shopify-section-cart-template > div > div.page-width > h1.small--text-...`
Mask	*Place cloned element in relation to this element.*
Conditions	**Insert Method** FIRST CHILD ∨
Inputs	*Where to insert the cloned element in relation to the Relative Element Selector.*
	Cloned Element ID
	Optional ID to assign to the cloned element.

Enable Preview ON | Target matched (1)

Monetate

Home   Clothing   Shoes   Camping   Gear   Contact   Account   Log out

Check Out **Shopping Cart**

Asolo FSN 95 GTX Hiking Boots
Remove

− 1 + $199.99

SPECIAL INSTRUCTIONS FOR SELLER

$199.99
Shipping & taxes calculated at checkout

Check Out

a. Launch your browser's developer tools to confirm the cloned element is where it needs to be within the page code to work. In this use case, Chrome DevTools reveals that the code for the cloned button is above the `<form>` element when the cloned code needs to be within it.

b. Correct any coding errors that you find. In this use case, the selector for the `<form>` element is the correct one to use in **Relative Element Selector** in Action Builder.


c. Test the cloned button again by clicking it and confirming that the checkout page loads as expected.

9. If necessary, click the **Styles** tab to add any CSS to ensure that the cloned button matches the original one.

> ℹ️ To refer to the container `<div>` or the clone itself if a wrapper isn't being used, you can use the `<CLONE_SELECTOR>` pseudo-selector to refer to it (for example, `<CLONE_SELECTOR> {float: right;}`).

10. Add an action condition to ensure the action only fires on the cart page. Click the **Conditions** tab, click **ADD CONDITION**, expand **Page type**, and then select **Page type =**. Input the value to identify the page type. Click the forward arrow.
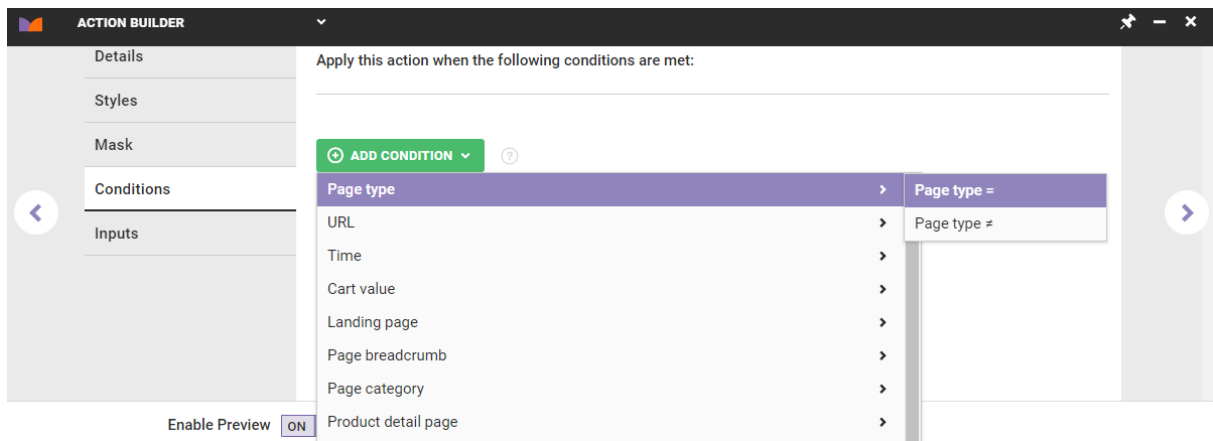
11. Give the action a title, a description, and a subcategory, and then click **CREATE & EXIT**.
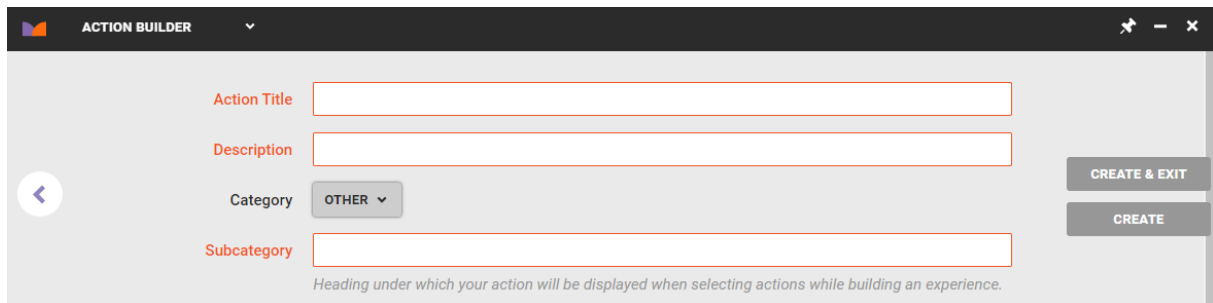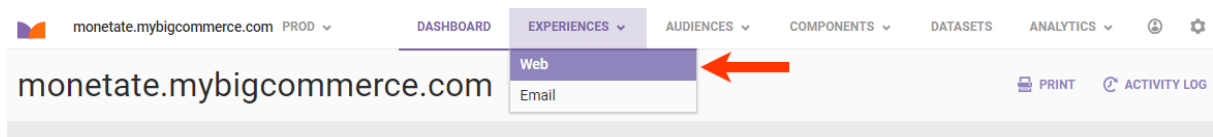

Now that you've created the Duplicate Element action, add it to a Web experience to deploy it on your site.
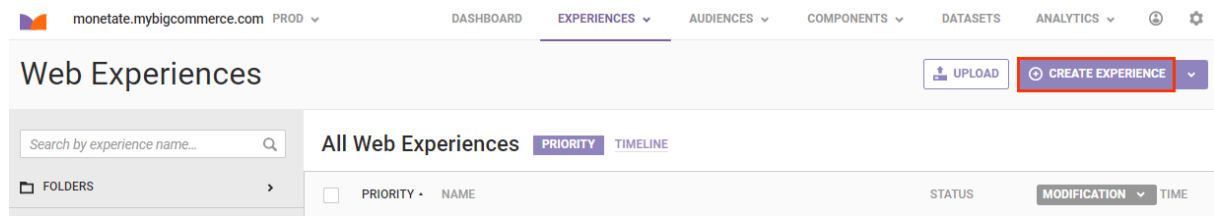
Creating the Experience

Follow these steps to create the experience that uses the custom Duplicate Element action.

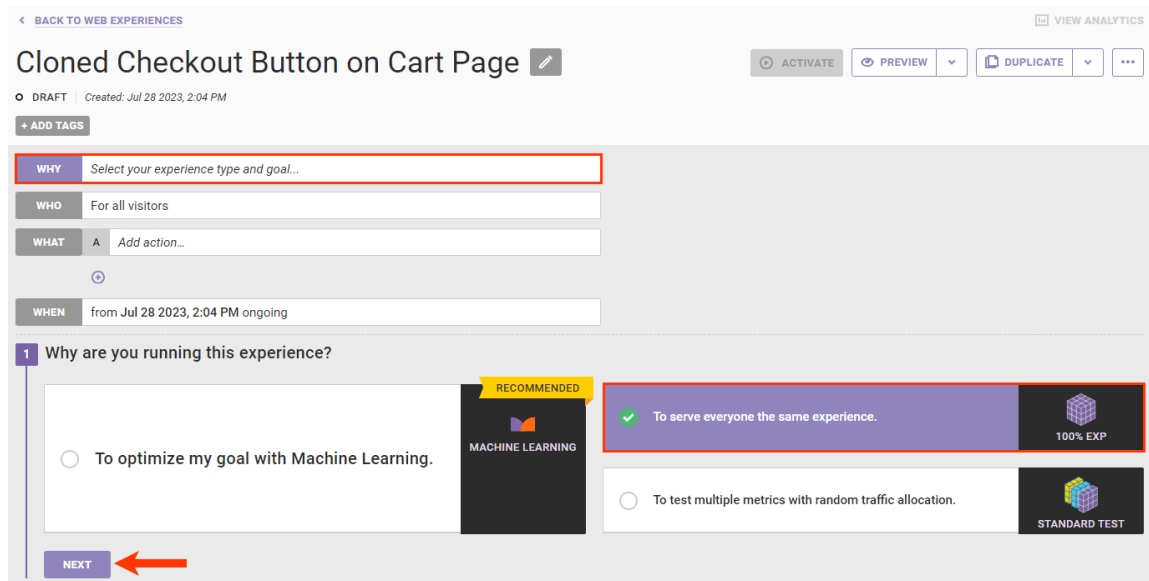1. Click **EXPERIENCES** in the top navigation bar, and then select **Web**.

2. Click **CREATE EXPERIENCE**.

3. Configure the WHY settings.
    a. Click **WHY**, select **To serve everyone the same experience**, and then click **NEXT**.

    b. Select a goal metric, adjust the secondary metrics as necessary, and then click **SAVE**.

4. Configure the WHO settings as necessary. For this use case, the experience should target all site visitors, so there's no need to change the WHO settings.

5. Add the Duplicate Element action to the experience.

   a. Click **WHAT** and then click **ADD ACTION**.
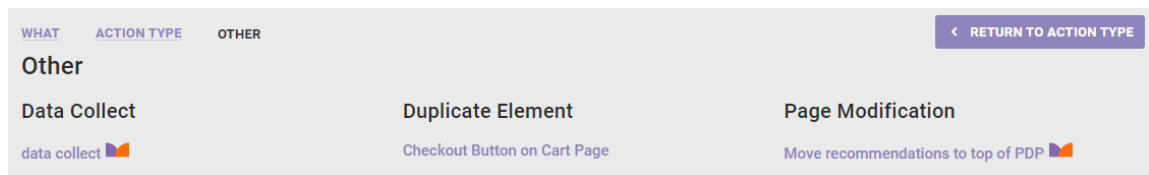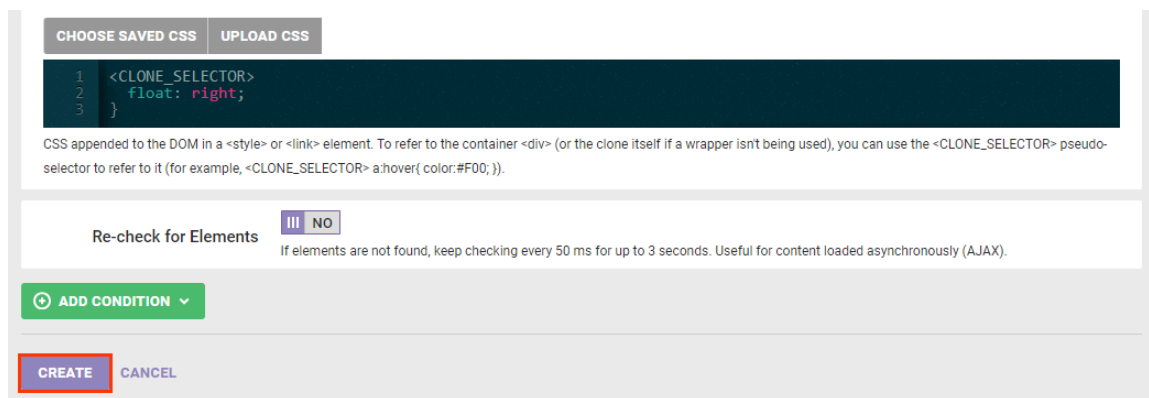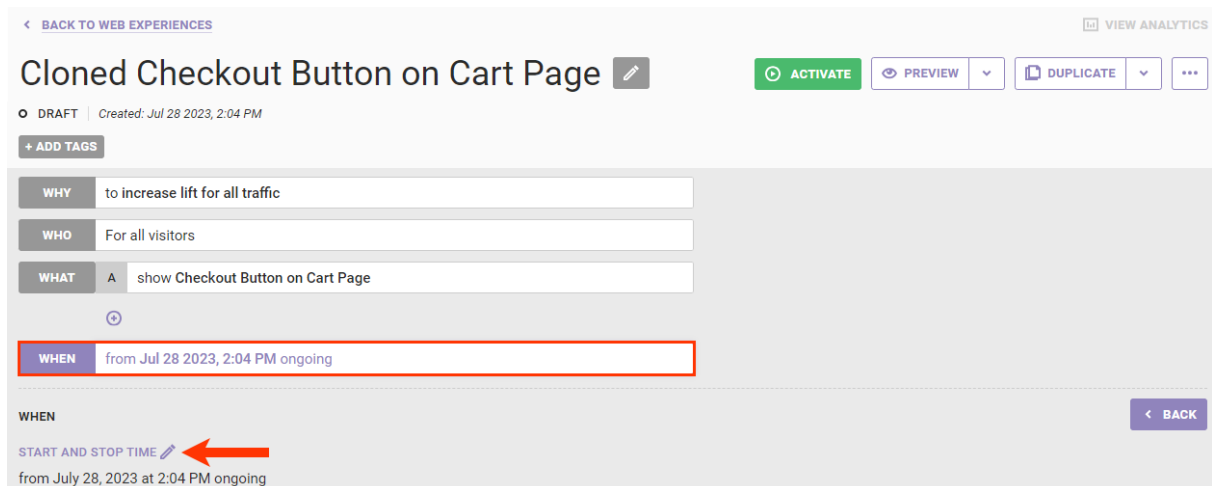

   b. Click **Other**.
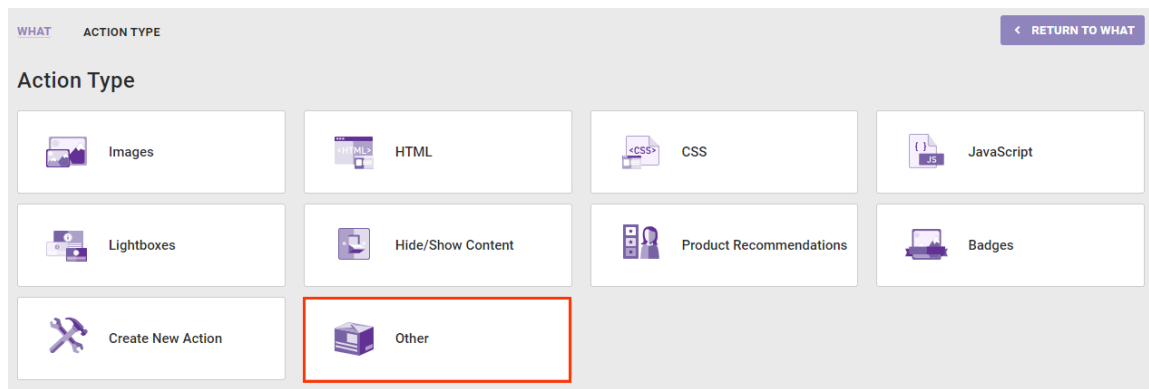
c. Click the Duplicate Element action that you created for this experience.

d. Because the action was so specifically configured in Action Builder, no experience-level changes are required. Click **CREATE**.

6. Click **WHEN** and then click **START AND STOP TIME** to configure the experience timing as necessary.
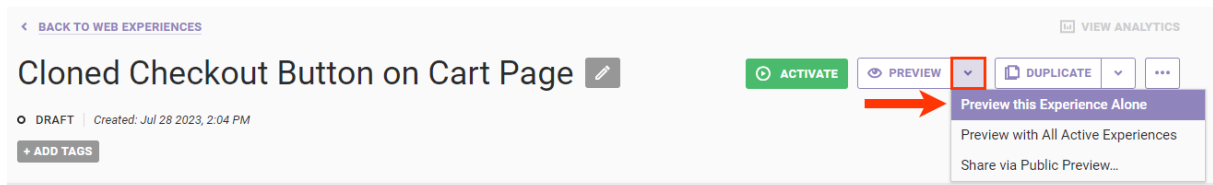
Ensure you test the experience before activating it.

Testing the Experience

Before scheduling when the experience runs by configuring the WHEN settings, you should perform a quality assurance check to ensure the action works as expected.

1. Click the arrow to the right of **PREVIEW**, and then select **Preview this Experience Alone**.

2. Add a product to the cart on your site, navigate to the cart page, and then ensure that the cloned button appears as expected.
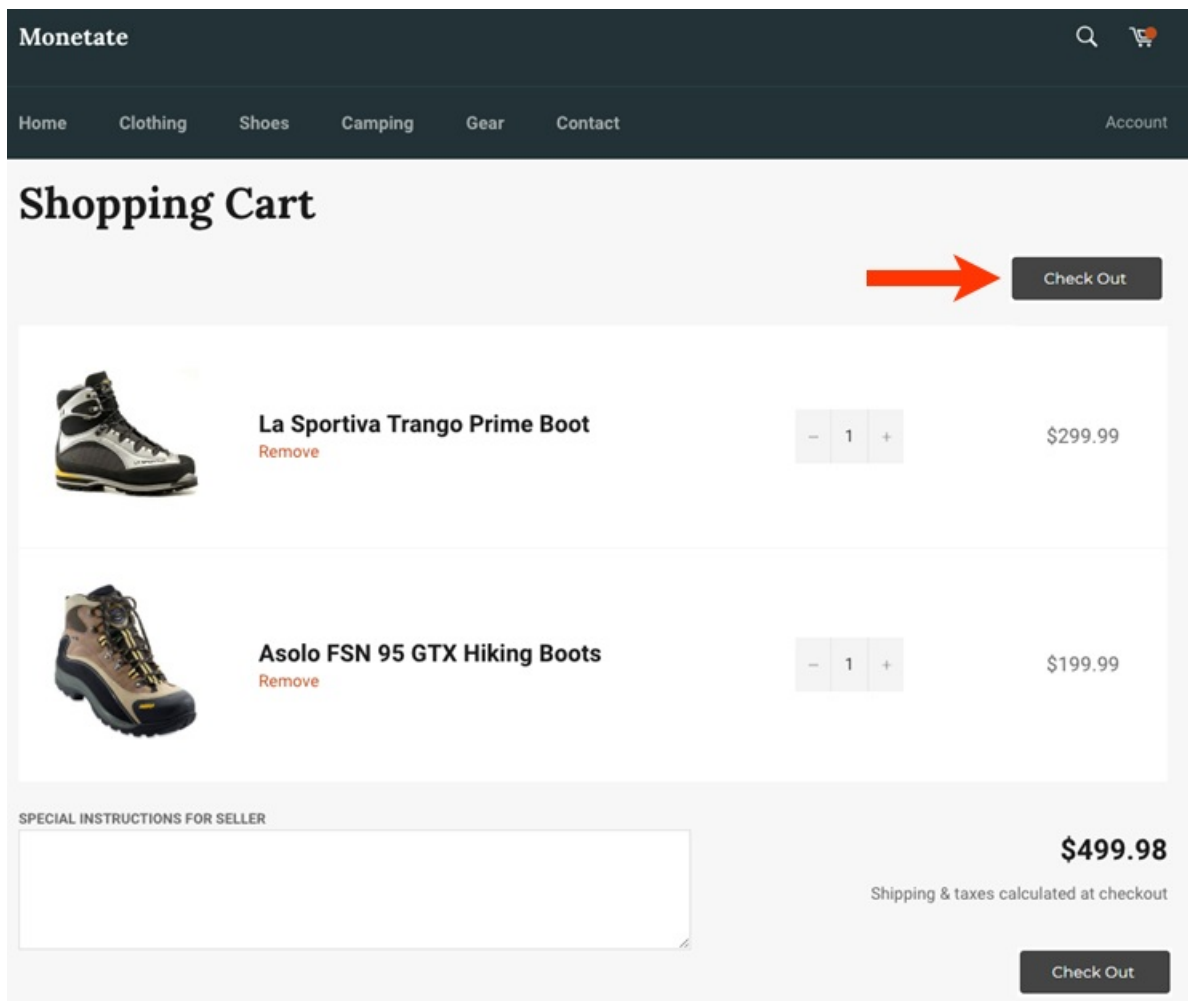
3. Click the cloned button to verify that doing so loads the checkout page.

After you confirm that the action worked as expected, return to the Experience Editor page in the Monetate platform to click **ACTIVATE**.