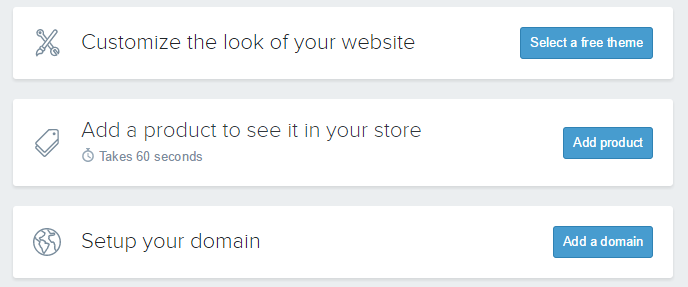
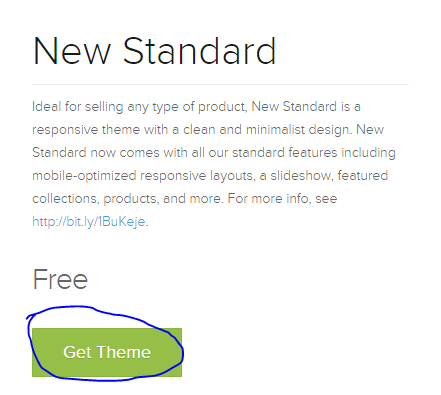
**Creating and Developing a Shopping Cart Within Shopify**

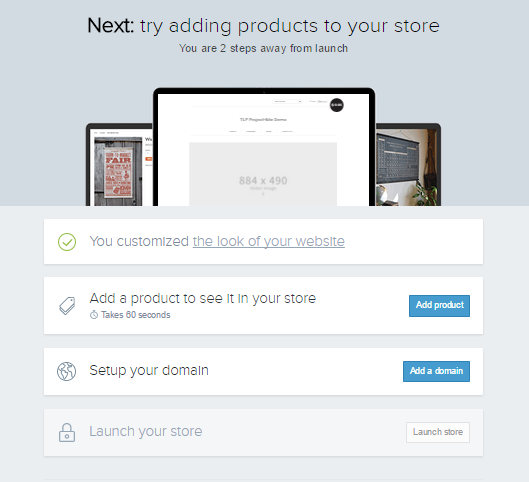
1. The first important step to take to further develop the storefront, as the admin panel prominently displays, is to select a theme. Within Shopify, themes are easy to “test out” due largely to the ease of changing between tasks and the wide array of offerings in the theme store making a decisive pick nearly impossible the first time around.



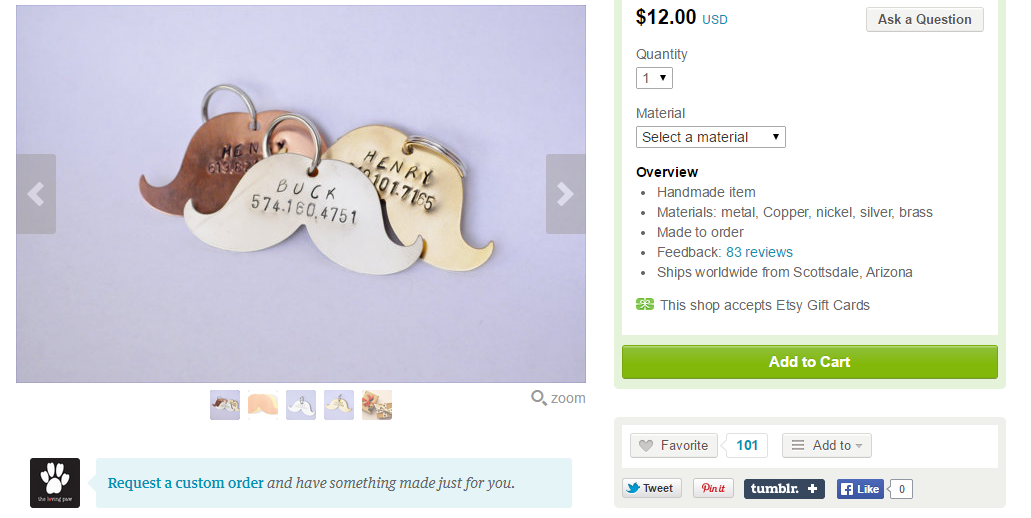
1. For the purpose of testing, our team of developers chose to implement the “New Standard Theme”, available in the “free” section of Shopify’s theme store. From preliminary client feedback, we intend to continue using this theme and through tasteful styling and CSS coding, create the theme our client is wishing for.



1. With the theme installed, the next prompt you will see by visiting the dashboard link, signified by a home icon, is to create a product. We took this step next, choosing a simple product from our clients’ Etsy store and replicating it in Shopify.

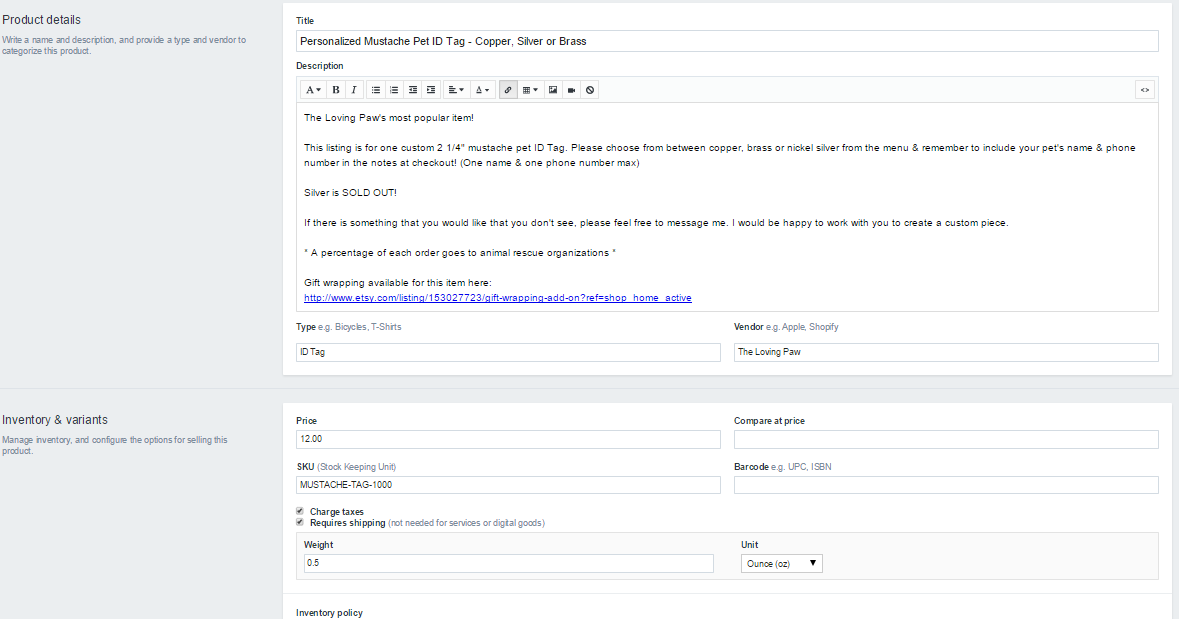


1. With a product page from the existing site to work with, we followed Shopify’s “Add product” page’s order to arrive at a completed/filled out new product page.

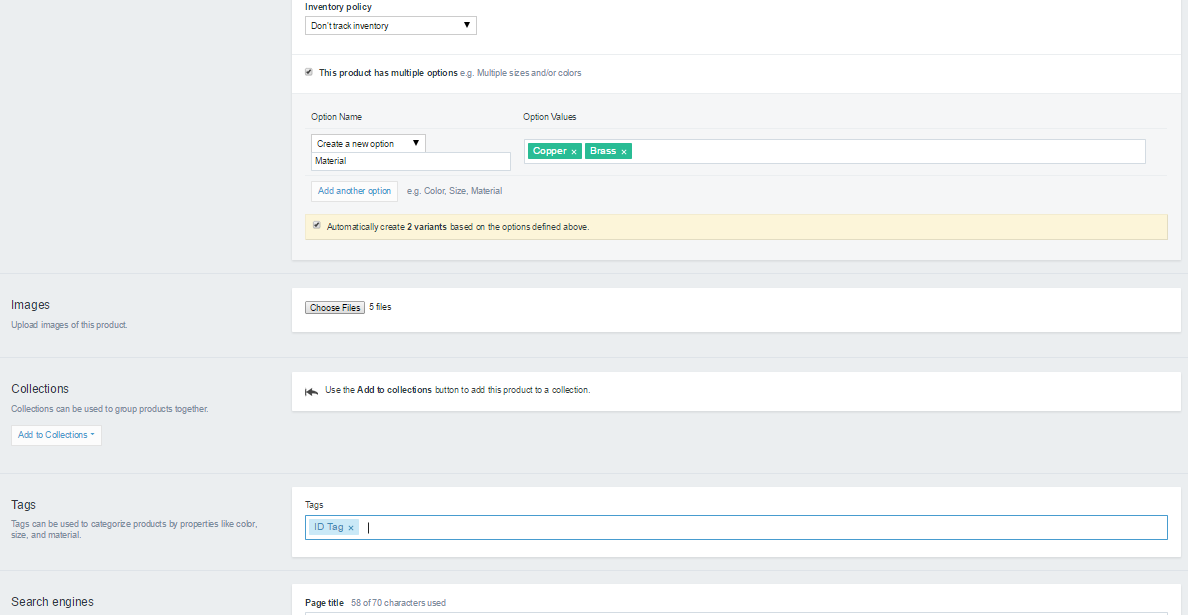


**Product to be brought from OLD Platform**

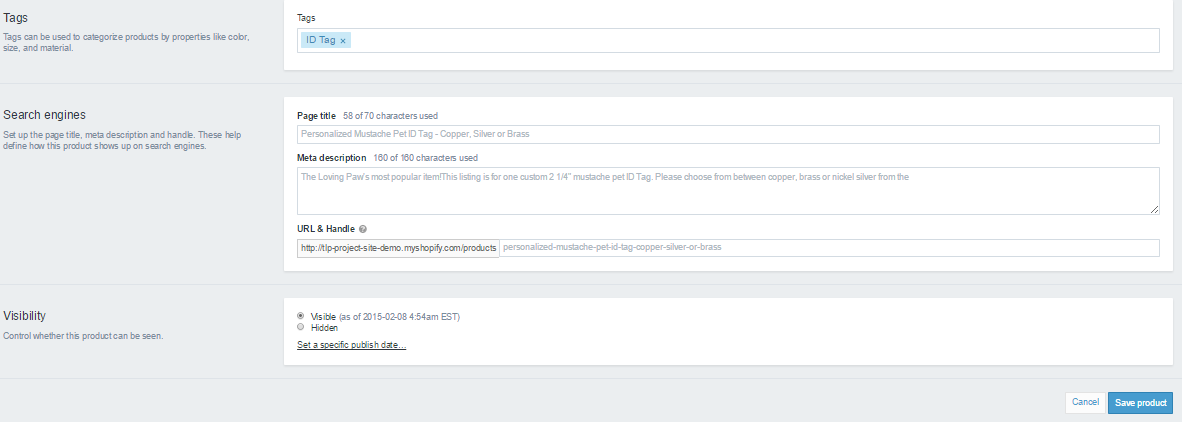
1. A completed product page for the product being brought into Shopify is shown in the following images. Shopify details criteria for each field on a new product page we therefore believe it is outside of the scope of this tutorial to discuss field-level requirements.



Screen 1 above (first screen of the add a product page shown)

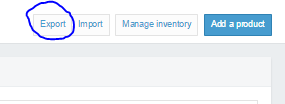


Screen 2 above (middle screen of the add a product page shown)

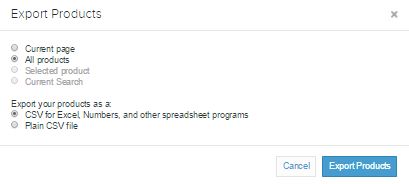


Screen 3 above (end of the add a product page shown)

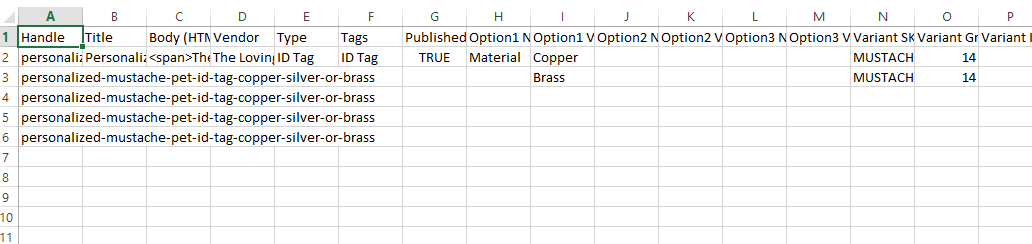
1. With a new product now added, the remaining task indicated on the dashboard is to “Setup your domain” 🡪 “add a domain” which we ignore for the remainder of this tutorial. This is where we will return to when the store is completely built and ready to be put online for customer interaction.
2. Understanding that there are extensive customization offerings beyond our clients’ modest number of products, we postpone the theme customizations, or more specifically the “product page template” customizations until after all products are imported from the old platform, Etsy, to the Shopify platform.
3. To streamline importing new products, we utilize Shopify’s CSV import functionality to allow our development team to manually enter products in database format to a central file and adding them to Shopify with one seamless transaction.
4. To begin this process, we can either utilize a generic CSV template from Shopify, or we can “Export” our previously added product to a CSV and add future products to that file. From the Shopify admin panel, we export our product to a CSV file by clicking “Products” and selecting “Export”, the leftmost selection at the top right of the page. When clicked, you will be presented various export options. Select “All Products” and formatted to: “CSV for Excel, Numbers, and other spreadsheet programs”.



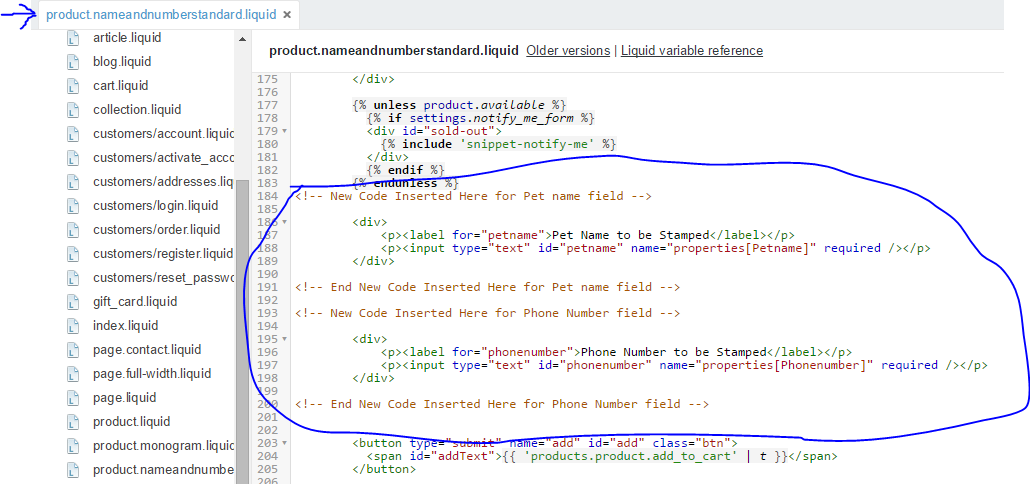
Selecting the export function from the product landing page from admin panel



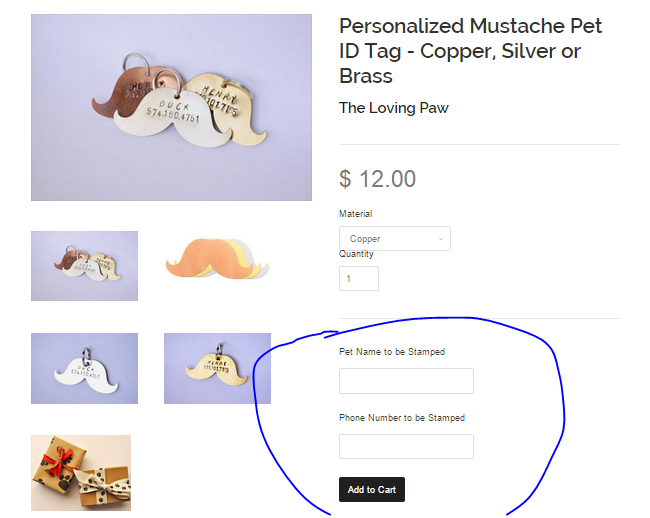
1. The CSV file containing current products will be emailed to the email Shopify has on record, and must be downloaded from there. Our CSV file containing the previously added product, opened in Excel is shown below.



1. The basic rules to follow in Shopify is to only fill-in the fields applicable to the product(s) being entered. Another important rule seen in the above screenshot is the need for 5 rows of product handles, even though only 2 options exist. The reason 5 rows are needed even though variation 1 (Copper) and Variation 2 (Brass) *should* warrant 2 rows, is the existence of 5 images, which are out of scope from the screenshot but exist in column Y. Shopify necessitates as many rows as images, and any rows with images need to have the product handle copied down until covering all images.
2. Through following rules such as those stated above, and those defined and explained further on Shopify’s website (http://docs.shopify.com/manual/your-store/products/product-csv), we now had a CSV file with 274 rows, 28 products, and 100 images, including the product already in the CSV upon its export, ready to import into our site. **A very convenient feature within Shopify’s import function is the default “no overwrite” feature.** This means you can “tack on” a product to a large CSV file with many items already in your store and when you re-upload the CSV, Shopify skips those unchanged and does not duplicate them unnecessarily.
3. Our completed CSV file is available for download/viewing at: (<https://cdn.shopify.com/s/files/1/0760/5531/files/The_Loving_Paw_Backup_Inventory_CSV_With_Images.csv?4241447346786097904>).
4. With the new products imported, our site is taking shape! View/Modify individual products until satisfied with their appearance. This includes, in TLP’s business case, applying “product.liquid” templates for groups of products with similar customization options.
5. Recall from step 9, that various products offer many customizations and therefore warrant the use of custom templates. Having imported the above CSV file, we can view for example the product titled “Personalized Mustache Pet ID Tag - Copper, Silver or Brass” offers “Pet Name” and owner “Phone Number” stamping as standard. Since Shopify only natively supports drop-down-menu based selections/options, we can achieve text-box level themes through utilizing “line item properties”.
6. **Our first business case encompasses products with Pet Name and Owner Phone Number stamping as standard.** In a separate business case this will be a premium option for other categories of products. Utilizing a Shopify tutorial (<http://docs.shopify.com/support/your-store/products/how-do-I-collect-additional-information-on-the-product-page-Like-for-a-monogram-engraving-or-customization>), we implemented line item properties through modifying the code for the theme’s “product template” and adding code to display a text field (note in future applications this can be scaled to fit multiple text fields), for each, a Pet’s name, and the owner’s phone number. We must remember to use HTML5’s “required” designation when declaring the text box field for each entry (name and phone no.). Naming the revised template as “Product.NameAndNumberStandard.Liquid” allows for easy locating for other products with Pet Name and Owner Number as standard. This product’s page now appears in compliance with business rules.



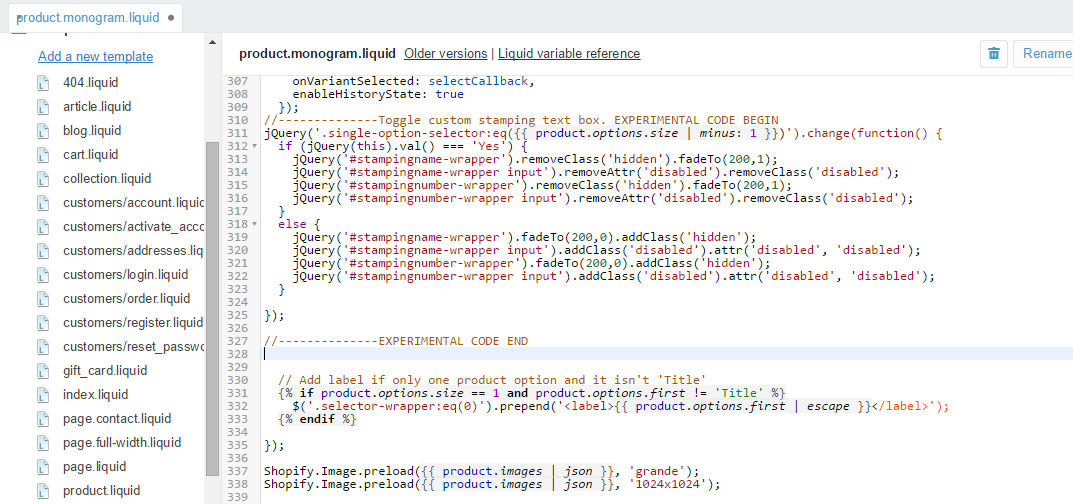
Shown above, we implemented 2 simple text fields one for Pet name, and another for Owner Phone number. Utilizing the required field, the template is now able to validate user entries. This template becomes named “Product.NameAndNumberStandard.Liquid” for future use.



1. The Product.NameAndNumberStandard.Liquid template is available for download at (<https://cdn.shopify.com/s/files/1/0760/5531/files/product.nameandnumberstandard.liquid.txt?17210124432529771243>).
2. The next business case, covering much of The Loving Paw’s Tag-Line ID products is a premium stamping option which will cost an additional $3.00 should customers select it. This option includes similar features to the “Product.NameAndNumberStandard.Liquid” however the need for the additional charge to be calculated **necessitates a separate template for products with this option.**
3. Utilizing yet another Shopify developer-authored tutorial (<http://docs.shopify.com/support/your-store/products/how-do-I-collect-additional-information-on-the-product-page-Like-for-a-monogram-engraving-or-customization>), *for reference,* we made additions to yet another variation/copy of the “product.liquid” (copying product.liquid’s contents into a new blank template created, which we named “product.monogram.liquid” to signify premium personalization.
4. The changes committed to the product.monogram.liquid file are shown below and discussed as well.



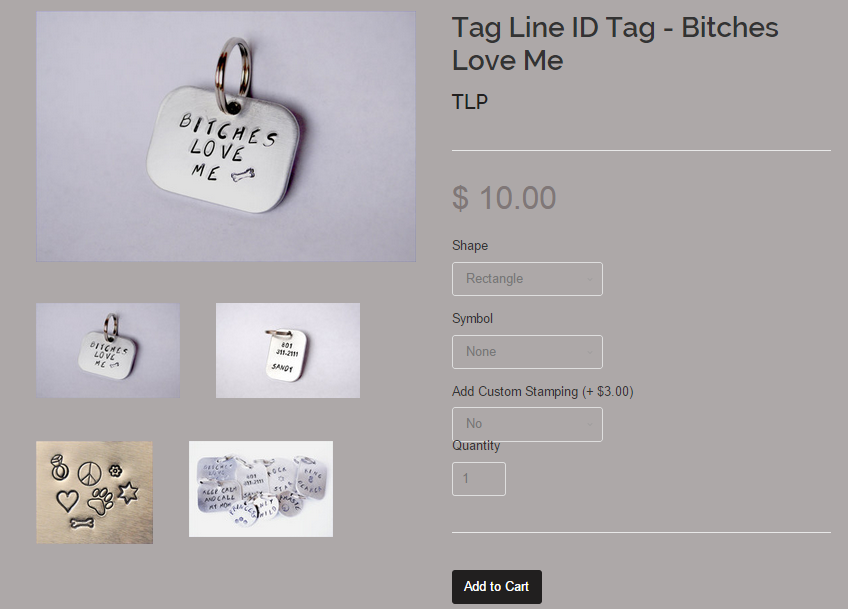
By utilizing text fields with the class “hidden”, each text field (separate block of code shown above) are created upon page load however remain hidden, or invisible to users until a choice on the premium option is taken. At the bottom of this “product.monogram.liquid” page we installed the jQuery code which will determine if the user picked “yes” from the drop-down menu prompting them to select or reject premium stamping options. At this step we made the text fields, triggered them to be hidden by default, and at the bottom of the page, shown below, we will install the logic to determine their future state.



As can be seen, the user will be prompted, via a “single-option-selector” a fancy way of describing a drop-down-menu, to select, *or keep* “No” **since it is the default selection at page load**, or select “Yes” **capitalization is critical. A lower case “yes” on the product page means this code should reflect the “yes” and match all capitalization patterns.**

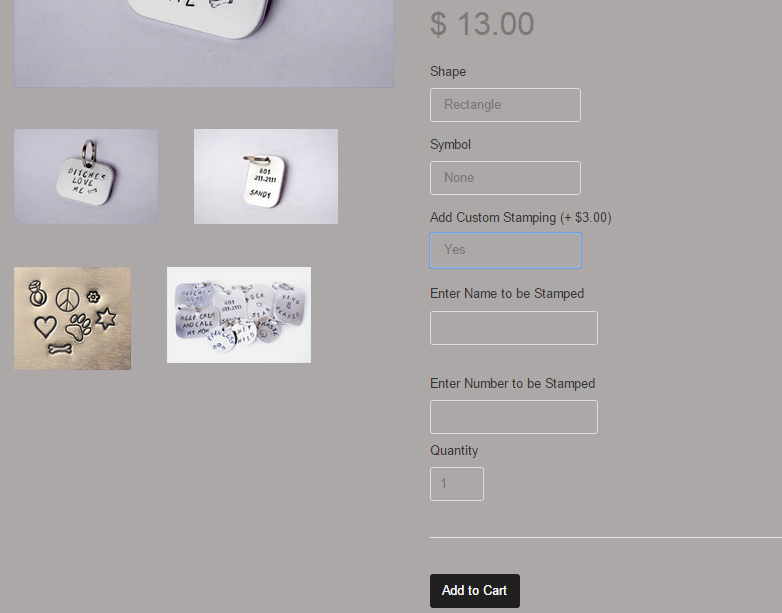
The first “if (jQuery(this).val() === ‘Yes’) { }block of code checks if the user chose “Yes” for the premium stamping option. If this is true, we remove the hidden attribute from the “Stamping Name” text field and then give it a location. Then the disabled feature is removed making the text box available for user entry/interaction. This logic is also duplicated (with 2 more lines) for the “Stamping Number” fields. It is important to note the dependency, The Loving Paw, to standardize processes offers these options together only. Meaning there is no logic needed to activate only one text field and not the other.

Finally, in line 318 shown above, the “else {……. }” block of code is for the case that the user did not pick “Yes”, which since this is a drop-down-menu means they must have chosen the other option “No”. Therefore the 2 text fields are once again given the hidden class and the disabled attribute. A product page utilizing this business logic is shown below for demonstration.



**Customer Turns-Down Custom Stamping Option**

Above, the customer decided not to request custom stamping and the price remained $10.00.

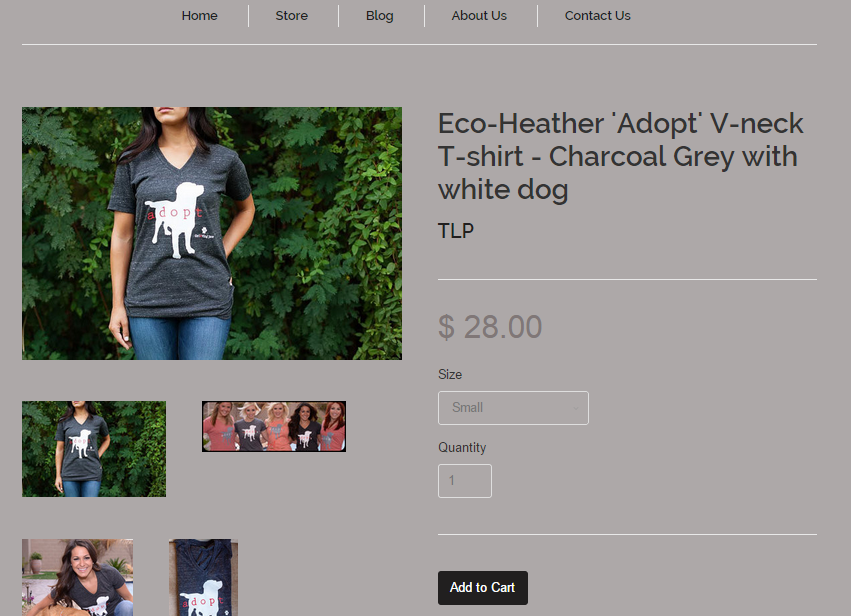


**Customer Selects Custom Stamping Option (+ $3.00)**

Above, the customer chose custom stamping, and the price and text fields to capture their entries were adjusted according to business rules. It is important to note that the price change was not managed in our coding modifications, but rather within Shopify’s variant attribute custom price system. Our variant system setup for the product used in this tutorial is shown below for reference.



1. Having covered the 2 prominent business cases central to The Loving Paw’s business model, our website was now friendly to complex issues which once required duplicate products (in essence trying to cover all possible customizations in numerous variant listings) and even more friendly to routine items such as clothing, duffle bags, jewelry, and standard Tag ID’s without additional customizations. **We preserved the original template from Shopify, the “product.liquid” template to be the ideal template for products with no customizations or special business rules.** A sample of a product page from a clothing item imported in our CSV file from step 12, and available for download in step 13 is shown below for demonstrating the simplicity “product.liquid” offers and compatibility with less complex inventory items.



**Many Sizes Available**

Through very simple variants, the T-shirt shown above is available in sizes ranging from Small to XL, and the inventory on hand for each size is tracked by Shopify. **Each size this shirt is offered in used to be a separate listing on The Loving Paw’s Etsy storefront.** The ease of creating variants within Shopify’s platform makes the occurrence of such scenarios rare, if ever.

1. With the demonstration of creating a storefront, and included material for replication, we arrive at the conclusion of this tutorial. With The Loving Paws’ storefront now ready for launch, we will have numerous “housekeeping” tasks to perform before official launch including SEO, CSS styling, Social media plugins, and construction of common pages and The Loving Paws’ blog page.