# Requirements Document

## Goals and Context

Our goal is to create inventory management system that provides the office supplies for the employee in the company. Currently many employee can’t get their office supplies to support their work. Their office supplies request often being rejected due unavailability of asset. Because, there is still no such a real-time system or application that shows and track the actual asset which are available and unavailable in their company. By integrating all types of the transaction from request until in one integrated system, there is

### What is the vision?

By consolidating the large but fragmented independent organic skincare market, there is the opportunity to create one brand that stands for the entire industry. This brand would represent the honest and wholesome values to all products on the website. Creating this brand will enable the website to better compete against the larger established skincare wholesalers..

## User Personas

*Note: Three example profiles have been included. Depending on the complexity of your product you may need more. Typically we use at least 3 external users and all key internal users.*

The following are the three key user personas that will use the e- commerce website.

### Young Professional

|  |  |
| --- | --- |
| User Type | Website User (External) |
| Occupation | Personal Assistant |
| Name | Sam Smith |
| Gender | Female |
| Age | 34 |
| Education | College |
| Location | New York, NY |
| Objective | Sam works long hours and no longer has the time to research and buy skincare items. She wants to be able to quickly find her favorite brand and product using the website search feature and order for same day delivery to make sure she never runs out. |

|  |  |
| --- | --- |
| User Type | Website User (External) |
| Name | Lucy Stevens |
| **Occupation** | **Magazine Editor** |
| Gender | Female |
| Age | 45 |
| Education | University |
| Location | New York, NY |
| Objective | Sam is often featured in the media and asked about her cosmetics choices. She wants to be seen as caring about the environment while keeping up with the latest trends. Currently she spends a lot of time searching for new products. She would like to be able to view side by side comparisons of the details of each product to be sure it is certified organic for example. |

### 

|  |  |
| --- | --- |
| User Type | Admin (Internal) |
| Name | Dave Mattheson |
| **Occupation** | **Website Content Manager** |
| Gender | Male |
| Age | 27 |
| Education | University |
| Location | San Francisco |
| Objective | Dave is an employee of the e-commerce website. His role is to write content for the website based on the newest trends and products. He wants to be able to see products as they are ordered and before they go live so he can alter copy and imagery and ensure consistent tone through the website. He also needs to know what products have been ordered to write relevant blog copy ahead of products being listed. |

### Site Admin

## User Stories

*Note: This section list high level and key user stories (often called Epics) that the website must support upon completion of the MVP (first) release. The typical format to use for stories is: As a <type of user>, I want <some goal> so that <some reason>.*

### Website Users

As a Website User, I am busy and have little time to buy skincare products. I want to be able to purchase products quickly and painlessly. Ideally with one click purchasing.

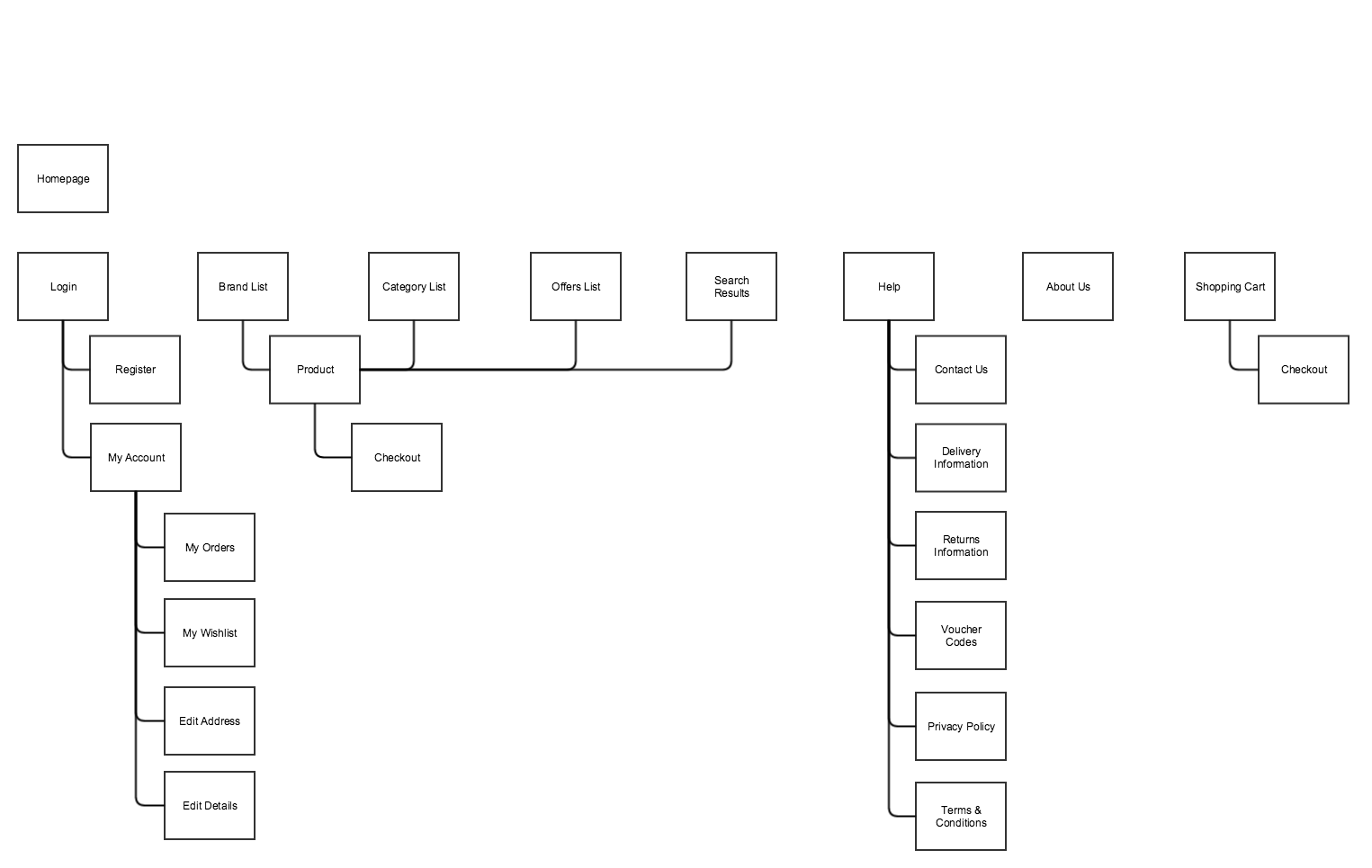
As a Website User, making the correct product choice is vital. I want to see my product options side by side so I can make an educated choice quickly.

### Website Admin

As a Admin, I want to be able to see new products and products categories as they are ordered so I can write content for the website.

As a Admin, I want to be able to track performance of product titles so that I can see what copy best converts to sales.

## Sitemap



## Page Descriptions

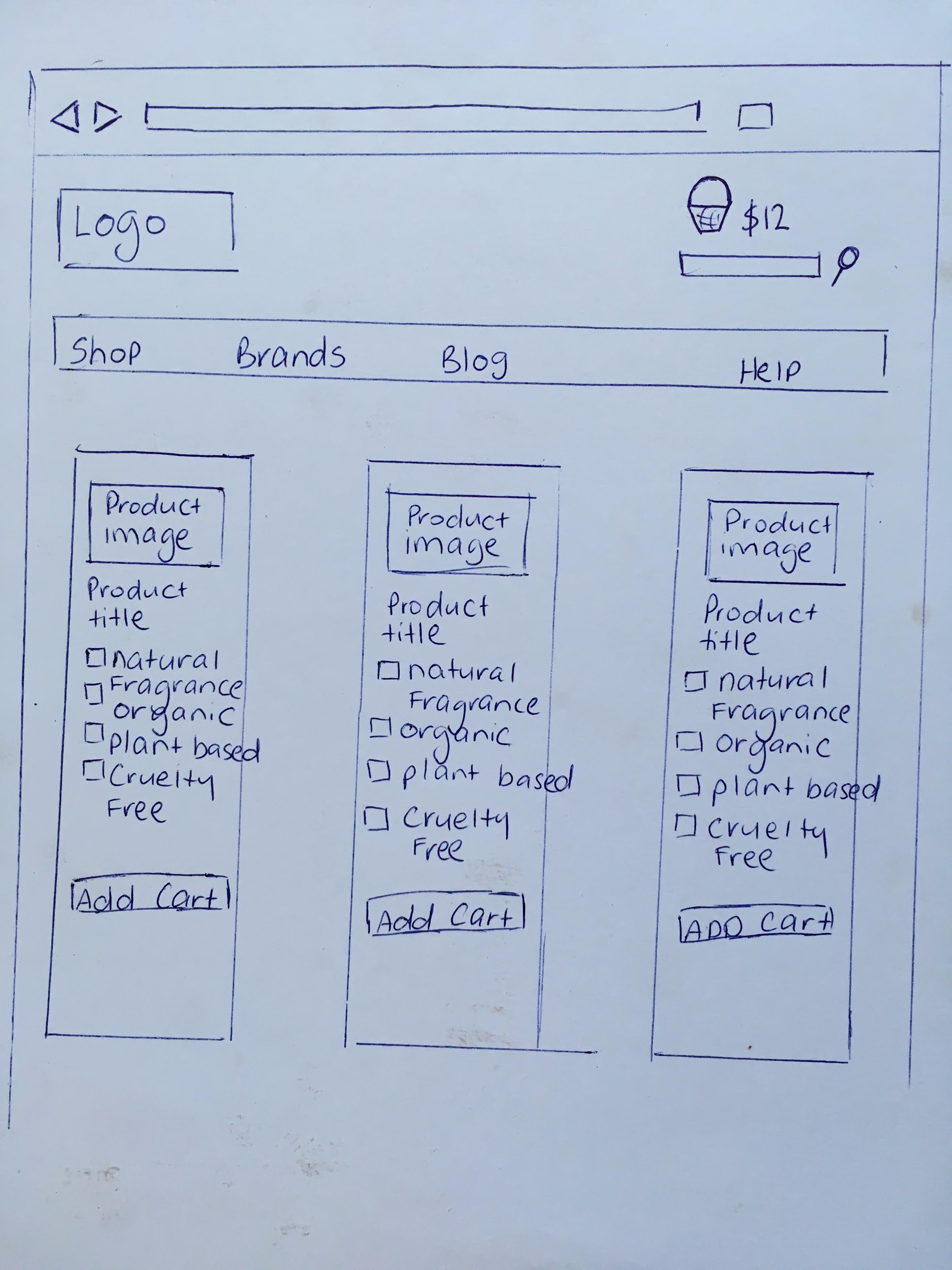
|  |  |
| --- | --- |
| Page | Items |
| Homepage  / | 1. Hero Branding Area 2. Featured Products Section 3. Browse Categories boxes 4. Search form |
| Category Page  /category/:category\_name | 1. Featured Category Products 2. Product filters 3. Product list |
| Product Page  /product/:product\_id | 1. Product Image 2. Product Title 3. Product Description 4. Add to cart button 5. Sizing Chart 6. Product Specifications 7. Product Reviews |
| Checkout Page  /checkout | 1. List of items in cart 2. Proceed to payment button 3. Ability to change quantities in cart |

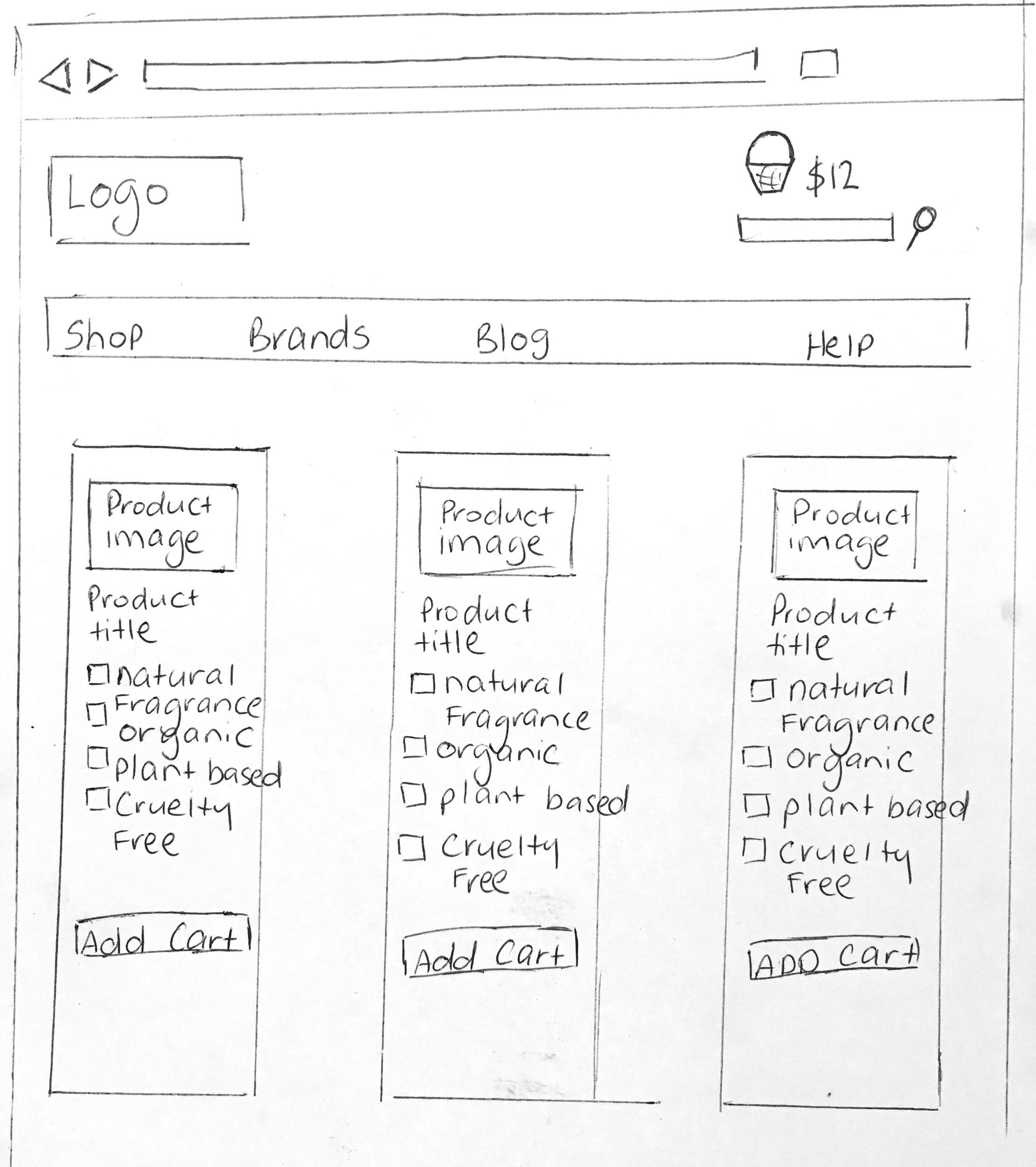
## 

## Wireframes

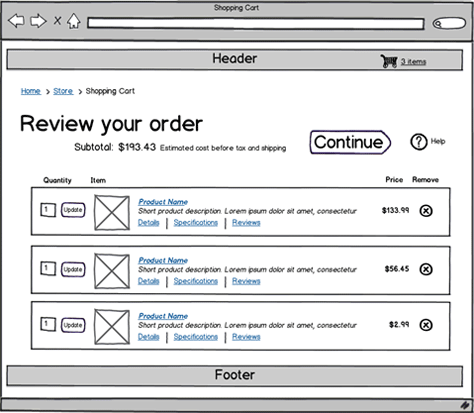
*Note: Below are two examples of wireframes. One is a sketch of the side by side comparison module. Sketches are quick and if done correctly are enough for the development to work with. The other wireframes is of the cart page and was created in Balsamiq, a prototyping tool. While these tools offer more features and consistency they are time consuming and not always required.*

Wireframe 1: Pen & Paper Sketch





Wireframe 2: Prototype Tool



## Non-Functional Requirements

1. The application must be built in Ruby on Rails
2. The application must be hosted on AWS
3. The application must use Stripe for payment processing
4. The application must work in all modern browsers
5. The application must be responsive (work well and look good on all screen sizes)
6. The application must be able to support 1000 simultaneous users

## Risks

1. Our predictive recommendation engine, which is a key differentiator for our startup, may be difficult to code.
2. Our business account may not be approved with Stripe.

## Future Iterations

Below is a list of all future features that will not be featured in the MVP release, but are important to keep in mind.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Shopper Alerts | A feature that will email users when a product that was out of stock that they viewed is back in stock. |
| Wish List | Ability for a user to save an item to their wish list. |
| Mailchimp Integration | When a user creates an account automatically add them to our mailing list using the Mailchimp API. |

# 