# Requirements Document

## Goals and Context

*Note: This section is used to give wider context to the development team and to ensure everyone is working to the same goal.*

### What is the goal?

Our goal is to create mock up to demonstrate distributed, highly scalable, and resiliency ecommerce website that sell all type of product.

### What problem does this project solve?

Currently many web apps that built is not scalable, resiliency.

1. Scalability: Many web apps hardly add new feature since they all tightly coupled together.
2. Resiliency: Request in single area can take a whole app down. Example: User racing to buy promo product. This can cause whole apps down. Even though it’s not necessary for normal user who just browse catalog to experiencing same things.
3. Independent Scalability: If we want to scale our ordering services we don’t have to scale other services. This impossible for non-distributed application.

### What is the vision?

By building distributed system, we highly benefit our product owner, such as

1. Fulfill rapidly changing business requirement.
2. Heterogeneity Programming Language
3. Quickly deliver artifact to PO.
4. Always production quality software.
5. Quickly add new feature.
6. Scale by needs.
7. Trace all system movement.
8. Full System Health Check.
9. And more…

Creating distributed system will enable your organization system to fight better against stale, unmaintained system.

## 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 |
| Objective | Sam likes skincare items. She always visits the application for browsing.  Sometime web can be in high traffic due promo. She still want to able browse the catalog even one of our service down. |

|  |  |
| --- | --- |
| User Type | Operation (Internal) |
| Name | Lucy Stevens |
| **Occupation** | **Operation** |
| Gender | Male |
| Objective | Sam operation in A Bank. Deployment always done in Friday midnight. He and his team responsible for his company deployments. He always does it manually, make sure configuration file place correctly, manage connection to external system, test application works fine or not, this workflow always takes hours for them. He wants to able automate all process deployment so he doesn’t have to figure out why application doesn’t works in production |

### 

|  |  |
| --- | --- |
| User Type | Product Owner (Internal) |
| Name | Dave Mattheson |
| **Occupation** | **Business Analyst** |
| Gender | Male |
| Objective | Dave is an employee of this organization. His role makes sure he can rapidly deliver feature for his organization. He also want to make his software super agile, can adapt to rapidly changing business requirement. |

## 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>.*

### Operations Admin

As an Operation. I want to be able automate our software deployment so I don’t have to work overnight in office.

As an Operation. I want to scale specific services so we can have optimized our resources.

### Website Users

As a Website User. I don’t want to be experiencing any downtime.

As a Website User, I want to see all product by category I’m interested in so I not disrupted by any random product.

As a Website User, I want to see all product by manufacturer I’m interested in so I not disrupted by any random vendor product.

As a Website User, I want to see product detail that I clicked so I know info of that product.

As a Website User, I want to see all pictures in product details so I get the image of the product.

As a Website User, I want to add product that I want to buy to cart so I can make an order.

As a Website User, I want to see all product in cart and total price so I know how many I should spend.

As a Website User, I want to make a purchase so I can buy those products.

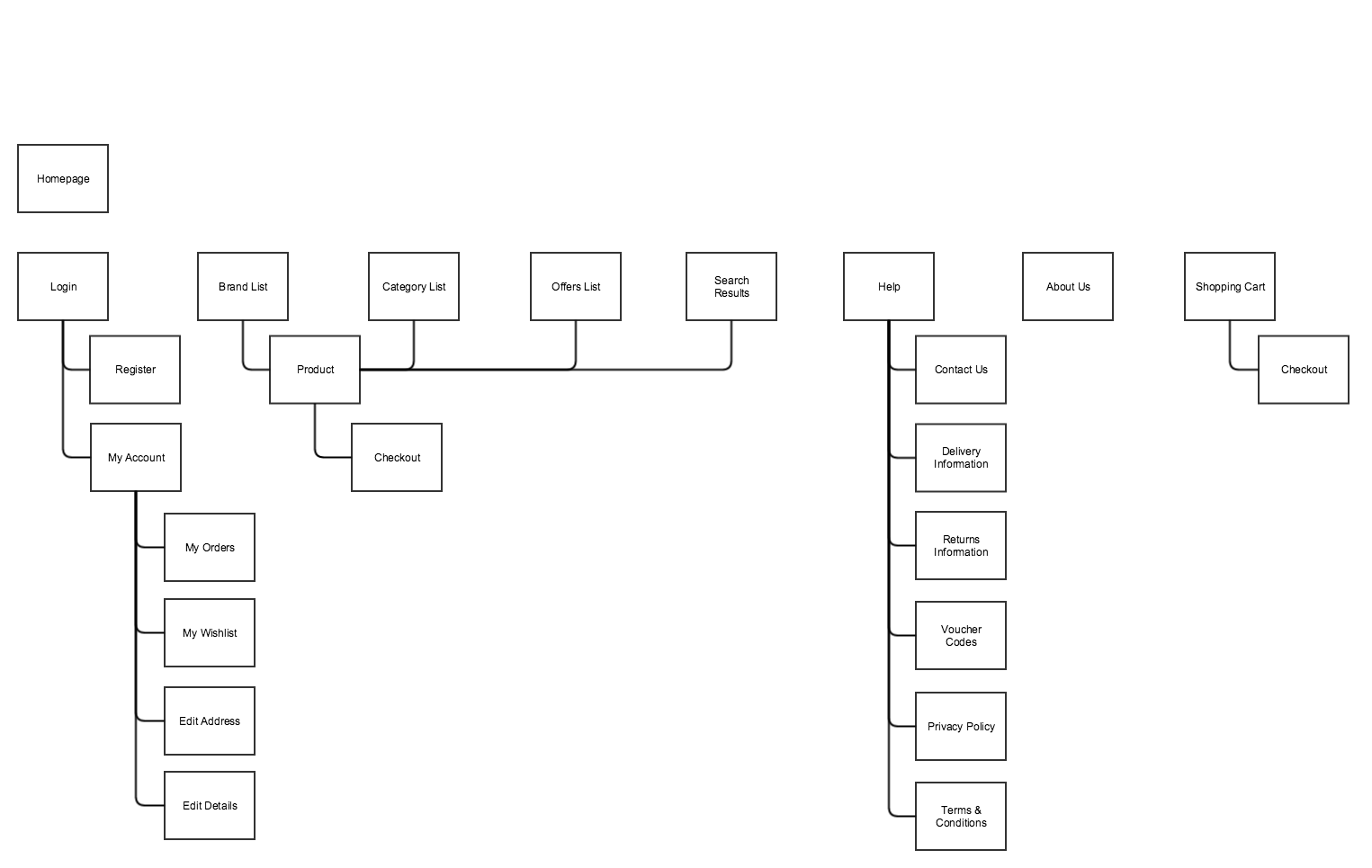
### Website Admin

As a Admin, I want to be able to CRUD manufacturer, products and products categories so I can write content for the website.

As a Admin, I want to be able add product to inventory so I can supply user demand.

As a Admin, I want to be able to track of product movement so that I can see what should I re-supply.

## Sitemap



## Page Descriptions

|  |  |
| --- | --- |
| Page | Items |
| Homepage  / | 1. Carousel Products 2. Category list link 3. Manufacturer list link 4. Categories boxes 5. Manufacturers boxes 6. Product added recently 7. Hottest product recently (by top sold) |
| Category List Page  /category-list | 1. Categories Boxes |
| Manufacturer List Page  /manufacturer-list | 1. Manufacturers Boxes |
| Product List Page  /product-list/?key=category&value=:category\_id  Product List Page  /product-list/?key=manufacturer&value=:manufacturer\_id | 1. Paginated Filter 2. Product list |
| Product Page  /product /:product\_id | 1. Product Image 2. Product Title 3. Product Description 4. Add to cart button 5. Product Rate 6. Product Details 7. Product Color 8. Wishlist button 9. Favorite button 10. Share button 11. Report button |
| Order Tab  /order | 1. Product Cart List 2. Product Cart Delete 3. Ability to change quantities in cart 4. Purchase button 5. Clear Cart button 6. Total Order label |
| Payments Tab  /order | 1. Mock Payment Button |

## 

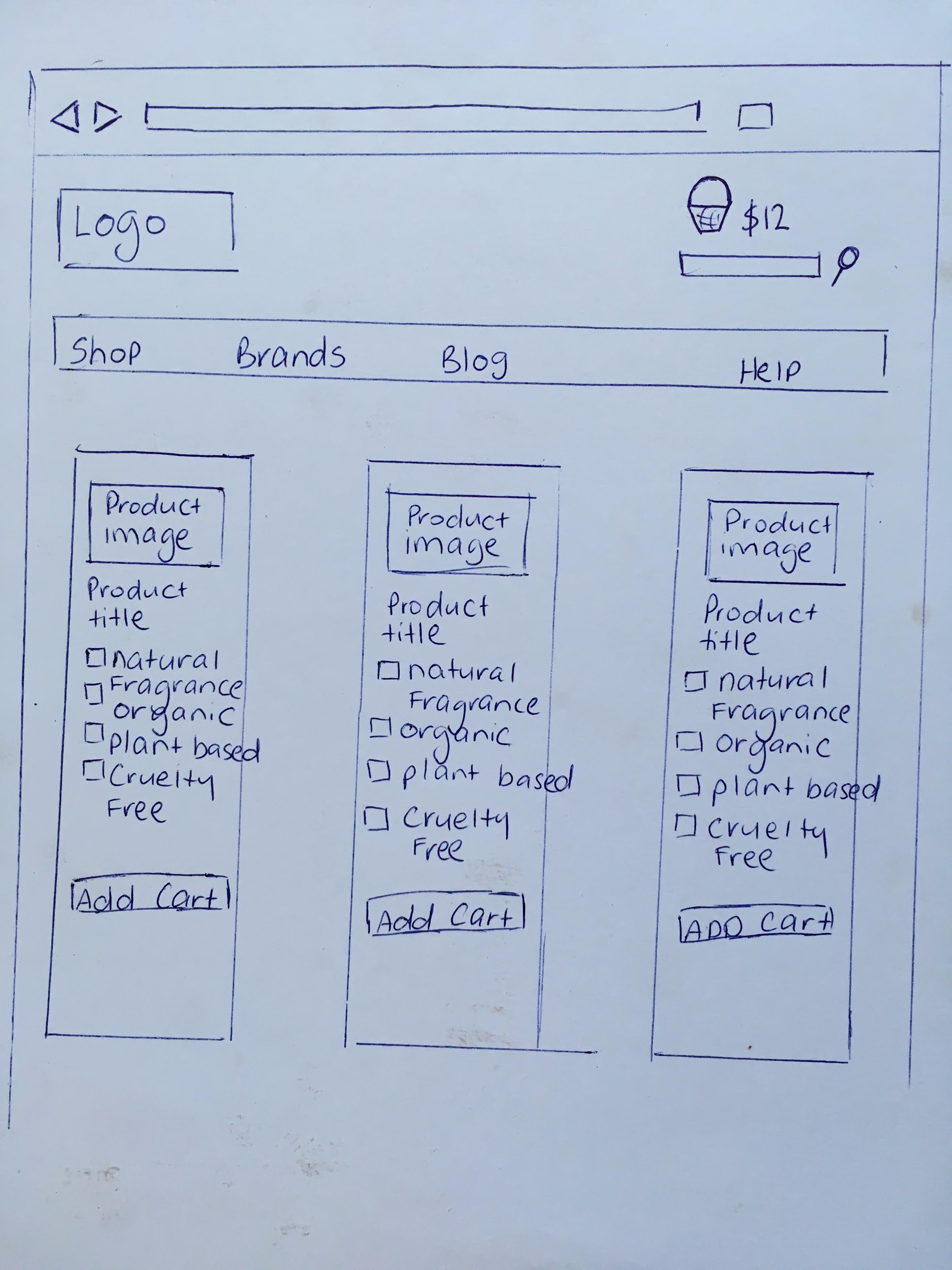
## API Descriptions

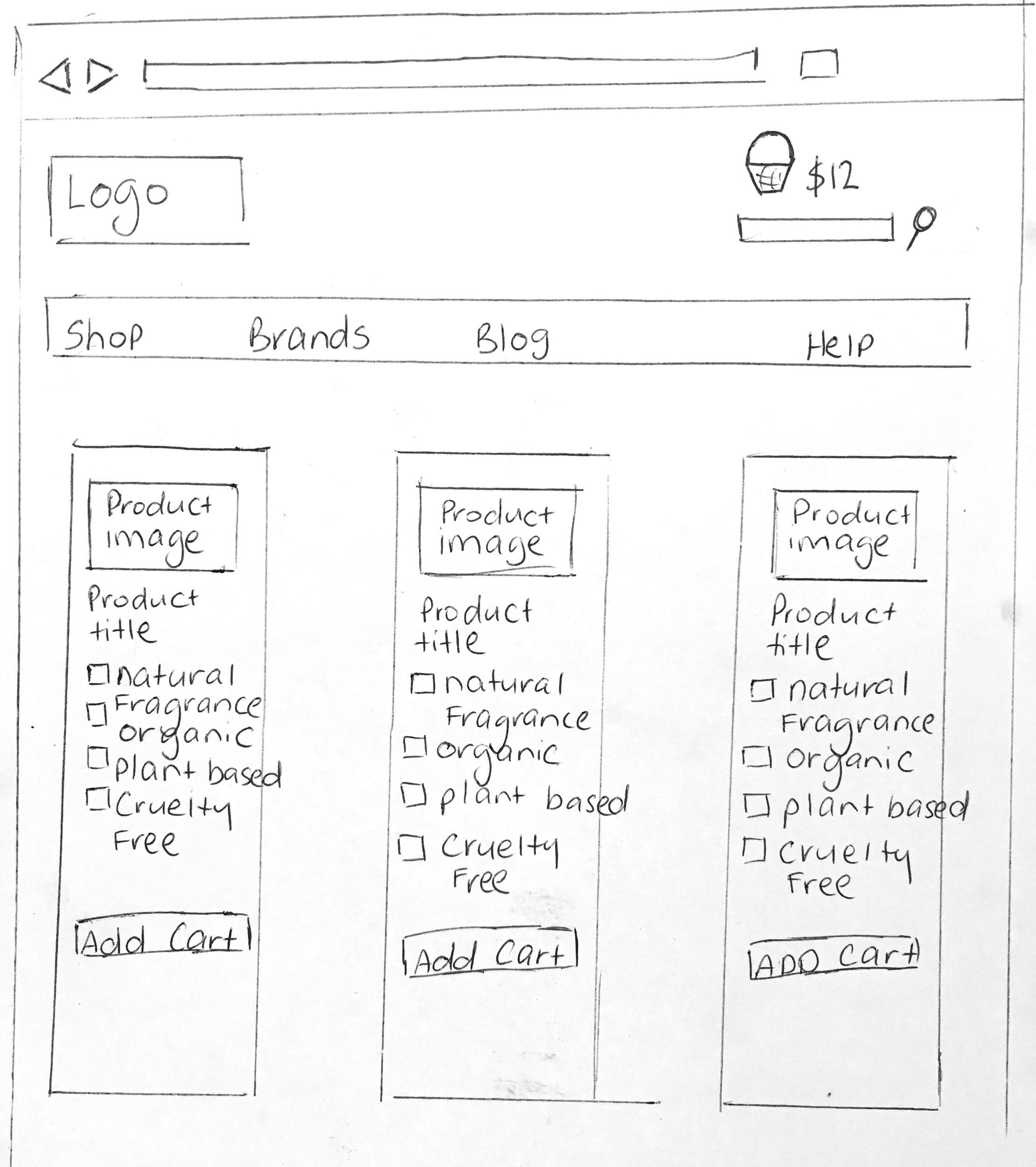
|  |  |
| --- | --- |
| API | Actions |
| Homepage  / | 1. Redirect to Swagger |
| Catalog API – Manufacturer Controller  /api/<version>/manufacturer | 1. Create Manufacturer 2. Update Manufacturer 3. Delete Manufacturer 4. Read Manifacturer 5. Post Manufacturer Image 6. Get Manufacturer Image |
| Catalog API – Product Controller  /api/<version>/product | 1. Create Product 2. Read Product 3. Delete Product 4. Update Product 5. Post Product Images 6. Get Product Images |
| Catalog API – Category Controller  /api/<version>/category | 1. Create Category 2. Read Category 3. Delete Category 4. Update Category 5. Post Category Images 6. Get Category Images |
| Cart API – Basket Controller  /api/<version>/basket | 1. Create Basket 2. Read Basket 3. Delete Basket 4. Update Basket |
| Order API – Order Controller  /api/<version>/order | 1. Create Category 2. Read Category 3. Delete Category 4. Update Category 5. Post Category Images 6. Get Category Images |

## Wireframes

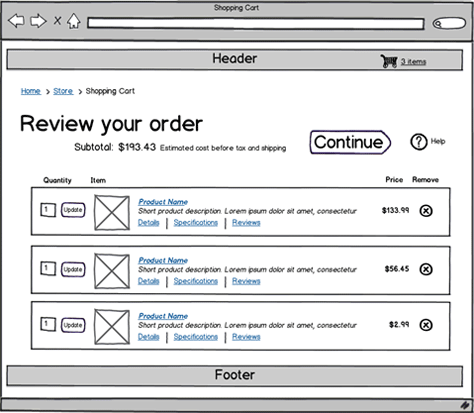
*Note: Below are two examples of wireframes. One is a sketch of the side by side comparison module. Sketches are quick and if done correctly are enough for the development to work with. The other wireframes is of the cart page and was created in Balsamiq, a prototyping tool. While these tools offer more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

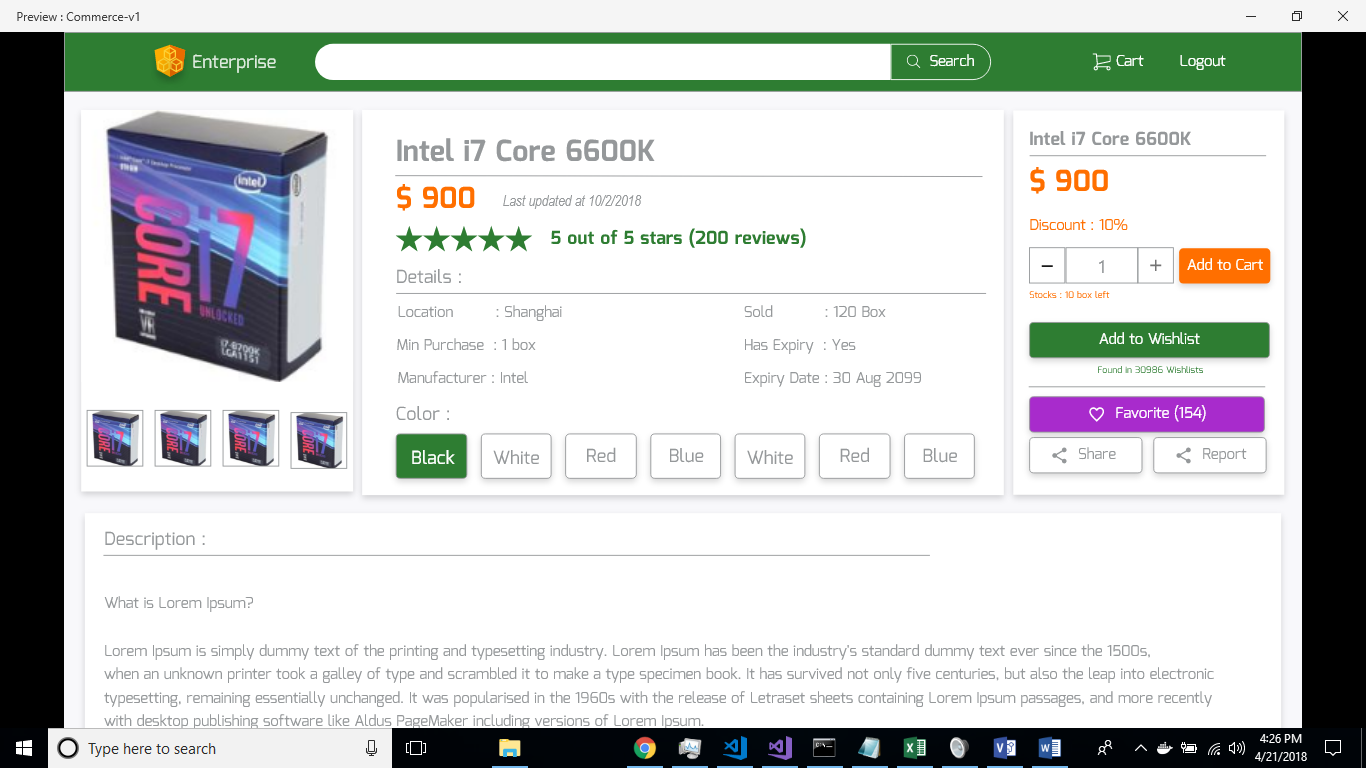
Wireframe 1: Pen & Paper Sketch





Wireframe 2: Prototype Tool

 Wireframe 3: Snapshot



## 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. |

# 