

Software Project

**Alice Corry**

**Your student number is N00211635**

Software Project

Produce a software solution for a business case study or a non-commercial client you identify.

Year 2 2022-23

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

| GitHub | <https://github.com/y2-SW-project/swproject23-ac-png.git> |
| --- | --- |
| Figma (Project) | <https://www.figma.com/files/project/81876427/Software-Project?fuid=1026808025755109748> |
| Video | <https://iadt-my.sharepoint.com/personal/n00211635_iadt_ie/Documents/Recordings/Meeting%20with%20Alice%20Corry%20(Student)-20230421_180620-Meeting%20Recording.mp4?web=1> |

Table of Contents

[1. Introduction](#_eyytwyws84ag)

[2. Business Concept](#_f4128uh71kbr)

[2.1. Business Idea](#_eyyju7tuwhtj)

[2.2. Business Model](#_gythxpj8q0uf)

[2.3. Market Research](#_hb02hqoctdqw)

[2.4. Marketing/Advertising](#_eds91d9h4eaw)

[2.5. Suppliers](#_pd29y8xmv8bo)

[2.6. Competitors](#_ccgindxdxe4e)

[2.7. Employees](#_2411oymypk1r)

[2.8. Environmental Impact](#_jyci1cr2ak24)

[3. Requirements](#_bacgmrwc95wf)

[3.1. Introduction](#_7lyuei1zn3v4)

[3.2. Requirements gathering](#_7fompslurfkl)

[3.2.1. Similar applications](#_gdnpph5umwjz)

[3.2.2. Interviews](#_v9977ytzfrpa)

[3.3. Requirements modelling](#_c48fzqdfgqfi)

[3.3.1. Functional requirements](#_pm6dm1ok1534)

[3.3.2. Non-functional requirements](#_efuzpw59sm44)

[3.3.3. Use Case Diagrams](#_f41fi8lwqta4)

[3.4. Feasibility](#_86iwirzbreae)

[4. Web Application Design](#_a5xjq6xnys8a)

[4.1. Layout](#_4i7ojhp)

4.2. Interaction

[4.3. Colour schemes](#_1ci93xb)

[4.4. Font choices](#_3whwml4)

[4.5. Wireframes](#_jeq7e1lpik5f)

[5. Database Design](#_xymtywxdzuvw)

[5.1. Description](#_3as4poj)

[5.2. Business Reporting Requirements](#_1pxezwc)

[5.3. Textual Representation of Dataset](#_49x2ik5)

[5.4. Business Rules](#_2p2csry)

[5.5. Entity Relationship Diagram](#_147n2zr)

[5.6. Tables](#_3o7alnk)

[5.7. Database Dictionary](#_m0icaj9qycwq)

[6. System Design/ Architecture Overview](#_y8rb1lblyhd8)

[6.1. Introduction](#_32hioqz)

[6.2. Model View Controller](#_1hmsyys)

[6.3. User Authentication](#_41mghml)

[6.4. Routing](#_2grqrue)

[6.5. Templating](#_vx1227)

[7. Testing](#_dmk1ero5r8jt)

[7.2.1. Login/Registration](#_2u6wntf)

[7.2.2. Navigation](#_19c6y18)

[7.2.3. Calculation](#_3tbugp1)

[7.2.4. CRUD](#_28h4qwu)

[7.2.5. Discussion of Functional Testing Results](#_nmf14n)

[8. Project Management](#_iymi419y5tns)

[8.1. Introduction](#_2lwamvv)

[8.2. Project Phases](#_3l18frh)

[8.2.1. Requirements](#_206ipza)

[8.2.2. Design](#_4k668n3)

[8.2.3. Implementation](#_2zbgiuw)

[8.2.4. Testing](#_1egqt2p)

[8.3. SCRUM Methodology (optional)](#_3ygebqi)

[8.4. Project Management Tools](#_2dlolyb)

[8.4.1. GitHub Project](#_sqyw64)

[8.4.2. GitHub](#_bgb7eqy0xmia)

[9. Reflection](#_8jmjd3oe8rfj)

[9.1. Your views on project](#_4bvk7pj)

[9.2. How could the project be developed further?](#_2r0uhxc)

[9.3. Assessment of your learning.](#_1664s55)

[9.4. Completing a large software development project](#_3q5sasy)

[9.5. Technical skills](#_25b2l0r)

[9.6. Further competencies and skills](#_xe5xjfnjoftb)

[10. References](#_4ci8s8e14yxq)

# 

# Introduction

**Overall aim**

Using what we have learnt over the last two years, we must build a Laravel web application with CRUD (Create, Read, Update and Delete) functionality.

**Technologies**

I connected to the server and database using PHP.

I created and organized my database using MySQL.

Bootstrap is the CSS framework I used to style my website.

I styled the website using CSS.

I built the website using HTML.

**Tools**

For my database, I used phpMyAdmin to design a sample structure.

I used this to do research and come up with ideas, Miro.

To wireframe my website, I used Figma.

For my project, I used TablePlus as my database tool.

**Project management**

I frequently saved my code on GitHub and added comments so my lecturers would know what I was working on.

# Business Concept

## Business Idea

DietOnline is an online grocery store that sells specialised food based on different diets. We sell dairy-free, meat-free, gluten-free and other products in ready-to-go packages. Our food will substitute traditional food but still provide the same great taste and nutrition.

We only want to provide nutritional, ready-to-go food, so our food is over half the price of regular shops. We sell everything from bread to sausage and offer services linked to specialised food.

Some of the products and services are;

* Meat-free products (sausages, rashers, burgers, etc.)
* Dairy-free products (milk, butter, cheese, etc.)
* Gluten-free products (bread, biscuits, etc.)
* Products for other diets (low fat, low cholesterol, etc.)
* Recipes for specific diets
* Guides and advice for each diet

## Business Model

Our business model is through service charges for manufacturers, regular customer purchases and any donations we get.

## Market Research

Their customers are individuals and families of all ages on specialised diets and those thinking of going on one.

The customers in this target market are interested in buying healthy, cheap and sustainable for their diets. To appeal to their target market, they sell affordable meals in recyclable packaging with easy-to-read ingredients and instructions.

## Marketing/Advertising

We will market on social media and in newspapers, buses and bus stops.

We will also reach out to various dietary groups and influencers to spread the message of our food.

## Suppliers

Our suppliers would be food manufacturers who share our value in diet-friendly and sustainable food. Before officially contracting, we will do a background check on all suppliers to ensure they use proper practices.

We will also offer a way to sign up as a supplier via our website, although they must come in for a meeting later.

## Competitors

| **Product / Service** | **Strengths** | **Weaknesses** |
| --- | --- | --- |
| Aldi (Plant Menu and Free-from Ranges) | * Cheap * Easy-to-make | * No online shopping option * Scattered around the shop and website |
| The Happy Pear | * All food vegan * Easy-to-make * Website has recipes and guides | * No online shopping option * In very few shops * Only one shop in all of Ireland (Greystones) |

## 

## Employees

Our business operates as a warehouse-like business. When customers order food from our website, our warehouse employees go through our various warehouses with an iPad and pick out all the food.

When the food is picked out at the warehouse, one of our truck drivers will drive to your location and deliver the food to you.

## Environmental Impact

Since most of our target market is vegans, we ensured that our products were packed in all recyclable materials.

We also do our best to ensure that all manufacturers use sustainable means when making and preparing food.

# Requirements

## Introduction

The requirements phase allows developers to determine what the program should be able to do.

Instead of the developer explaining what is necessary, learning what the users would like the application to do is important.

## Requirements gathering

## Similar applications

1. [**Kroger**](https://www.kroger.com/)

**Description**

An American online food store sells organic and plant-based foods that are good for the environment and a healthy lifestyle.

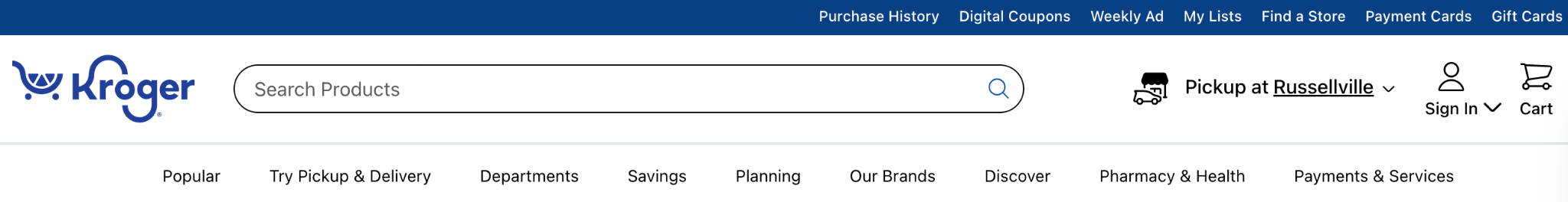
**Advantages**

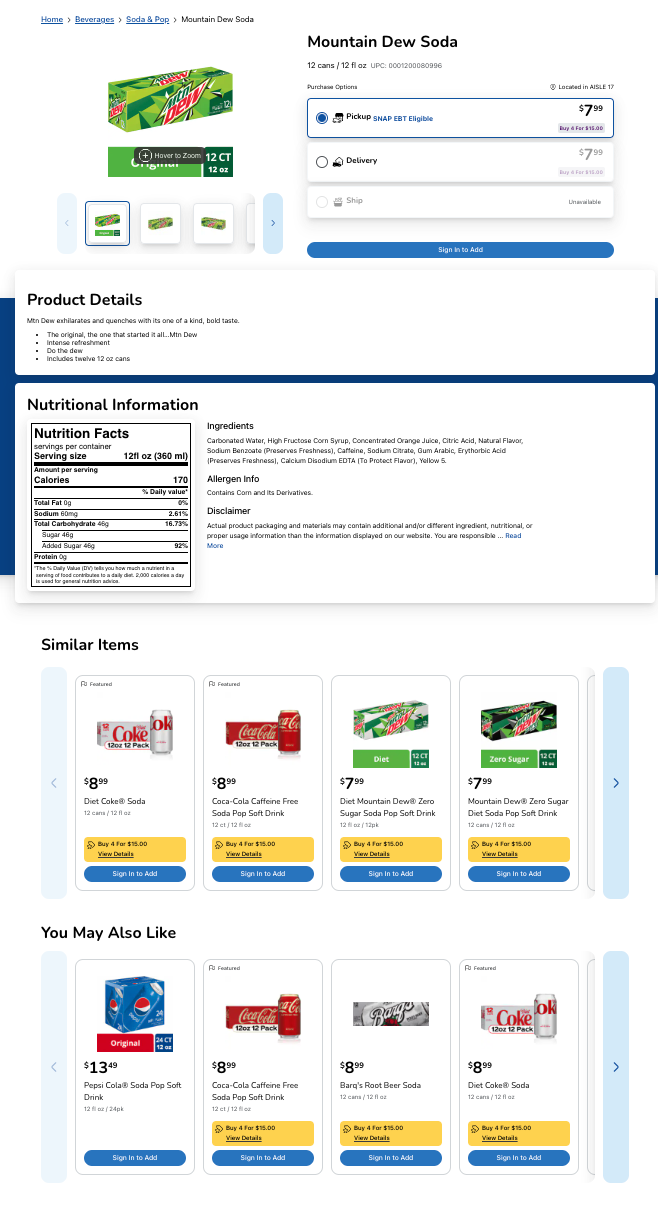
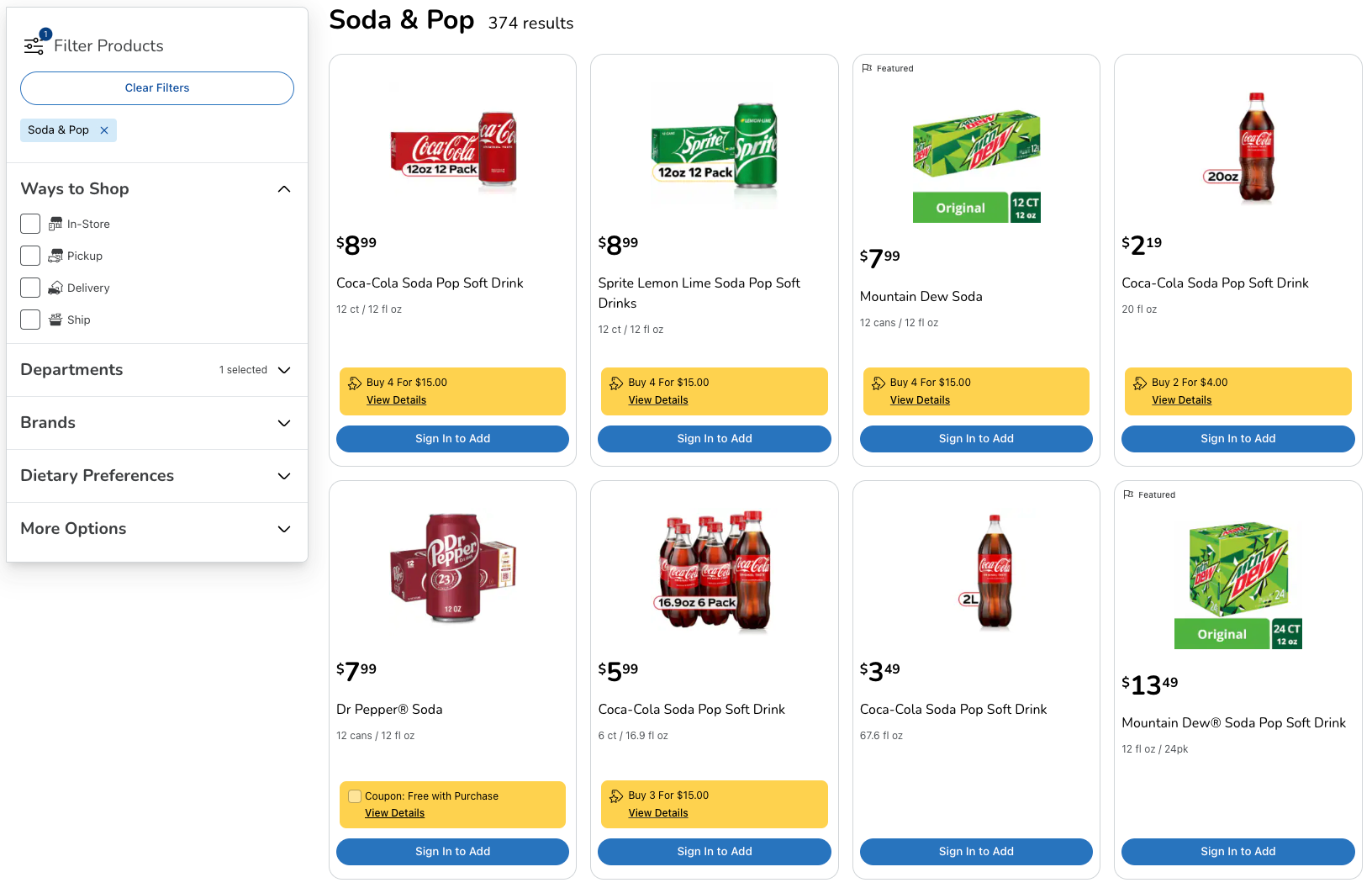
* The website's UI is excellent overall; it is easy to understand and doesn't require a lot of colours.
* The single-product page seems to have a lot of content, including nutritional information, reviews and similar products.

**Disadvantages**

* It's difficult to tell what to click on the navigation bar at the top of the page to access the various parts of the site.
* There seems to be a price variation; while some items are quite cheap and have deals, most appear very pricey.

**Screenshots**





1. [**VeganEssentials**](https://veganessentials.com/)

**Description**

All of their products, free of animal testing and cruelty, are available here. The one-stop shop for all things vegan is called VeganEssentials.

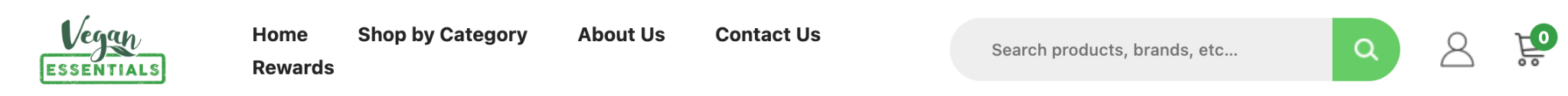
**Advantages**

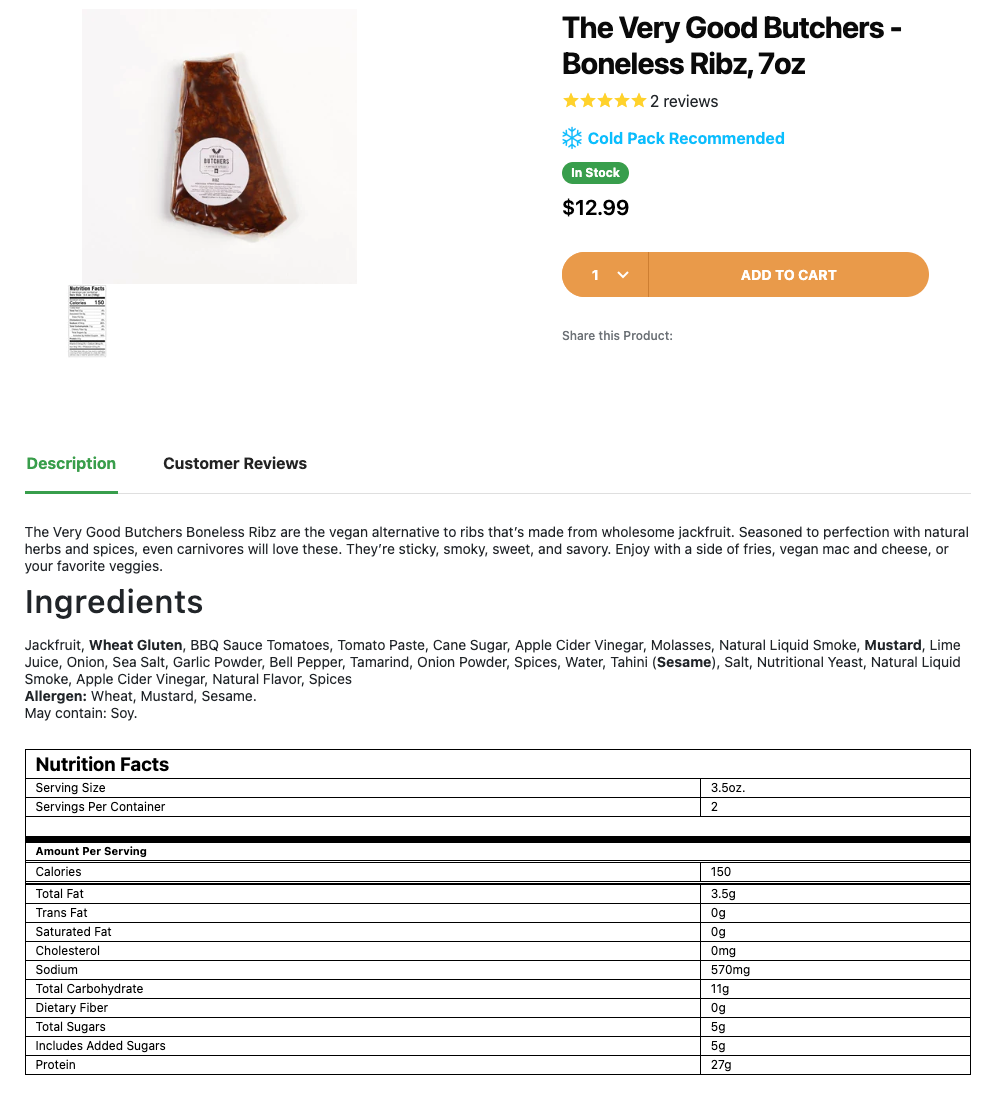
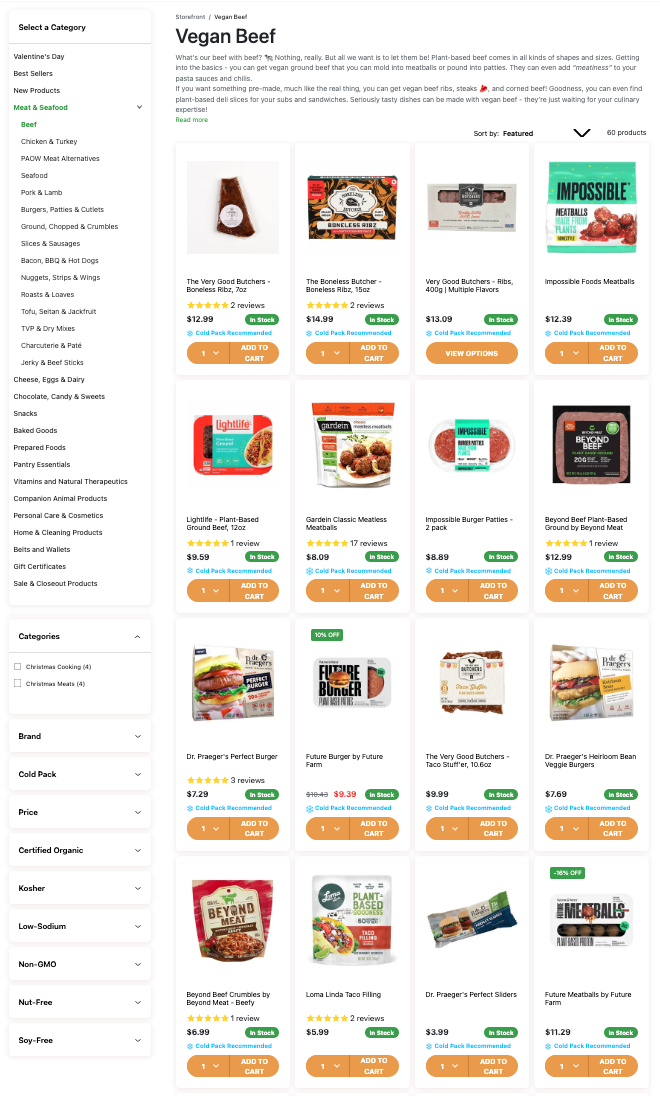
* Excellent website navigation makes it very simple to navigate the site, and the categories are simple to understand.
* The website's UI is fairly simple, but it communicates the message and doesn't look boring.

**Disadvantages**

* The website only sells vegan products; there are none for other diets (e.g. gluten-free).
* The prices are really high.

**Screenshots**

****

****

## Interviews

To determine what users would like the application to perform, two interviews were performed.

1. During the first interview, the user expressed interest in basic functionality, viewing all items as well as a particular item and adding items to their basket.
2. During the second interview, the user asked for basic functionality and more advanced features, such as searching for certain things and having a filter to find items that fit a particular diet.

## Requirements modelling

## Functional requirements

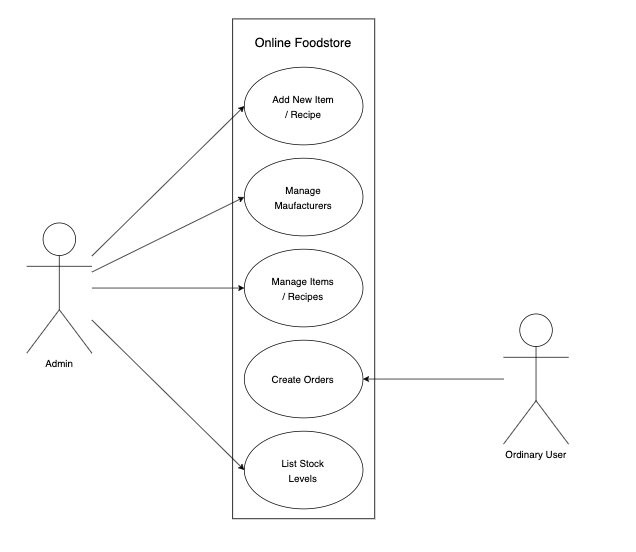
The following is a list of features the application should have:

1. Account creation and login page.
2. Users should be able to view the items and recipes (view all and a single item).
3. Admins can add, remove, and update products and recipes.
4. Depending on their diet, users can filter the items.
5. Items can be added to users' baskets.

## Non-functional requirements

1. To speed up the application, add Javascript validation (as well as PHP).
2. UUIDs are added to the URLs, so the user isn't aware of the database.

## Use Case Diagrams



## Feasibility

The following technologies will be used for this application: Docker, Table Plus (for the database), and Laravel (Framework and Breeze).

There shouldn't be any compatibility issues with these technologies based on my previous experience using them together.

# Web Application Design

## Layout

The website will be built using Laravel and Bootstrap. Laravel will be used to build the framework (or foundation) of the website, and Bootstrap will be used to provide some customised CSS elements.

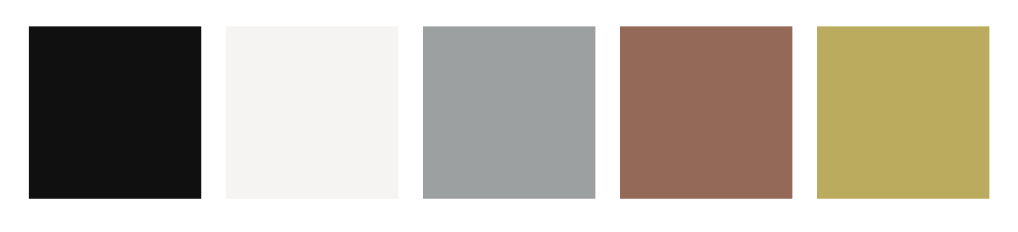
The website will be coded to act responsively so users can use it from any device. If it doesn't work, it won't be a big part of the website, so it should not break.

## Interaction

To navigate the website, the user will use the navigation bar at the top of the page. Also, there will be breadcrumbs that users may use to go to previous pages and see where they are on the website. The pages will also have various links that can be used for easy navigation.

A few forms will be used throughout the website, but only for admin users. These are used to create, read, update, and delete products, categories, manufacturers, diets, and recipes. There will be two forms used by customers, one for creating an account and the other for entering detail when buying products.

## Colour schemes



^ These are the colours I'll use consistently throughout the website; they were inspired by similar colours found on the Shopify website. These colours, in my opinion, look good since they use muted colours around the colours of brown and grey instead of being too boring or bold.

## Font choices

DietOnline

^ The logo in the top left of the page will be in Julius Sans One. Also, just one weight is available, so I will only use that.

DietOnline

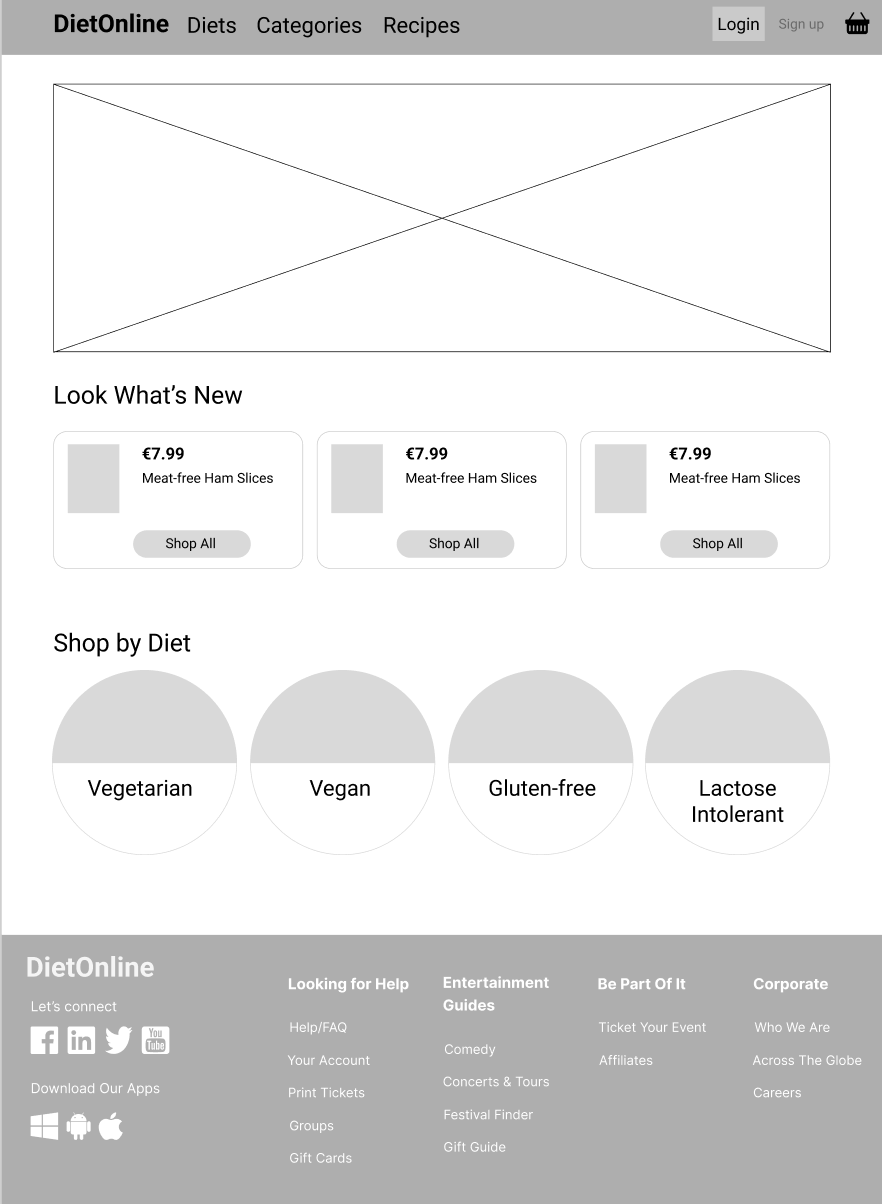
**DietOnline**

DietOnline

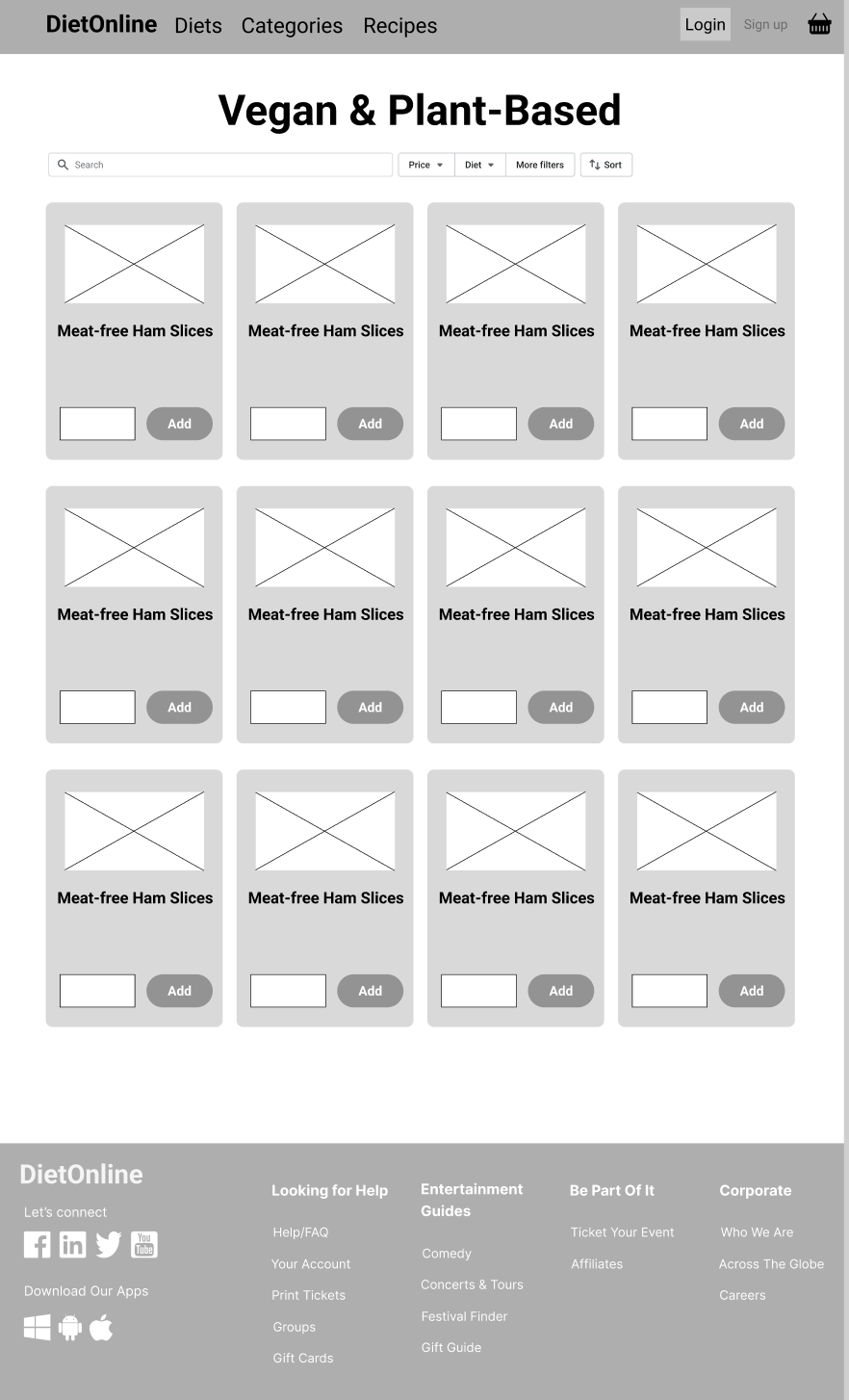
*DietOnline*

^ Roboto will be used for the page's main body. I'll use a variety of weights, sizes, and styles (underline, italic).

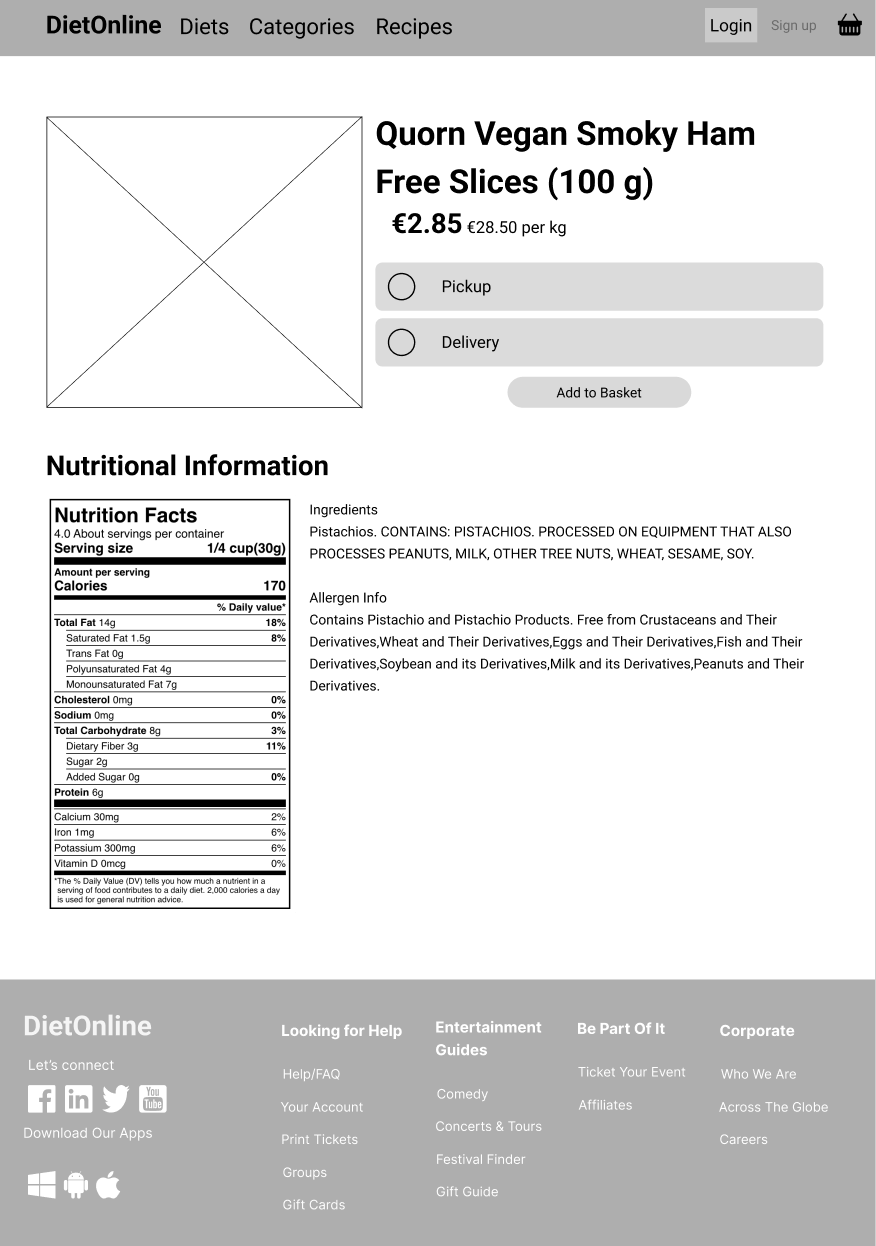
## Wireframes



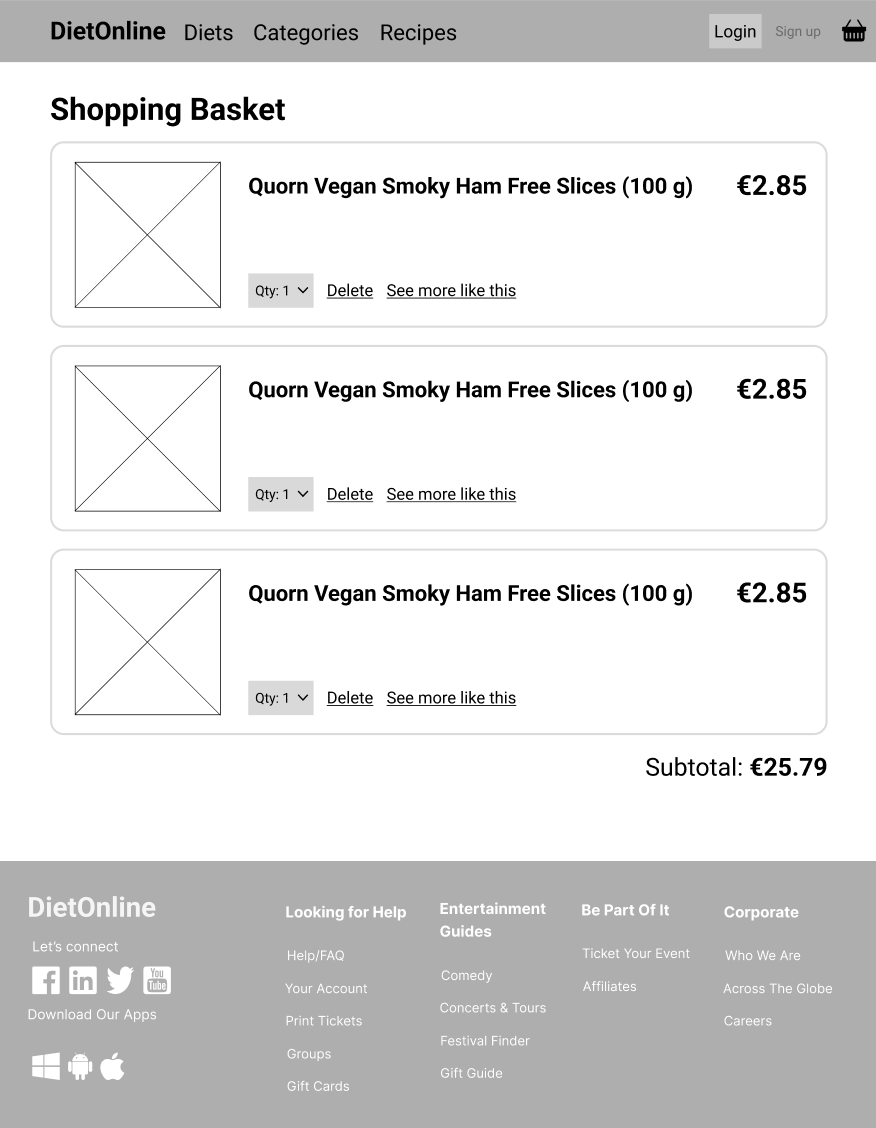
^ The website's home page will be this page. The top of the page will include a navigation bar with login and account creation options using Breeze. There will be helpful links in the footer at the bottom. Also, there will be several categories and cards with basic product information.



^ The navigation bar and footer from the previous page are also used on this page. Although the main function is to display a list of products in a grid layout, there will also be a search bar and some basic filtering options to help find products.



^ This page also uses the previous pages' navigation bar and footer. The main function of this page is to show a single product (the one that was clicked on the previous page), with its price and delivery method option. The allergies are highlighted in **bold** in the nutritional information displayed under the product.



^ The navigation bar and footer from the previous pages are also used on this page. By selecting the basket icon next to the login button, the main function of this page is to display a list of the products in the user's shopping cart. The name and price of the product are shown for each list item. But, there will also be a dropdown to edit the quantity, delete and view related products. Under the list, the basket's subtotal will be shown.

# Database Design

## Description

A company sells food products for various diets on its website. For all of their products and manufacturers, they would need a database. They would need each product's name, price, manufacturer, and stock level. They would also need each manufacturer's name, address and contact information. When creating an account, customers must enter their details. Customers must also enter their credit card information when paying for their order.

## Business Reporting Requirements

Administrators need to be able to create, read, edit and delete manufacturers and products.

Users need to find all food using a list of diets.

Users may need to sort food by price (cheapest at the top).

Users may want to find food from a specific manufacturer.

## Textual Representation of Dataset

PRODUCT (id, name, description, price, manufacturer\_id, image\_id)

MANUFACTURER (id, name, address, email, phone\_number)

IMAGE (id, filename)

USER (id, name email, password)

ROLE (id, name description)

USER\_ROLE (iuser\_id, role\_id)

DIET (id, name, description)

PRODUCT\_DIET (product\_id, diet\_id)

## Business Rules

A Product can have a single Image.

A Diet can have many Products.

A Product can belong to many Diets.

A Manufacturer can have many Products.

A Product can belong to one Manufacturer.

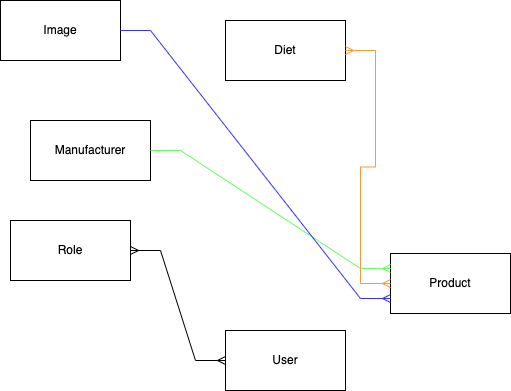
A Role can have many Users.

A User can belong to many Roles.

## Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a graphical way to show the relationship between entities in a database.

An entity is anything that we need to maintain information to carry out transactions.

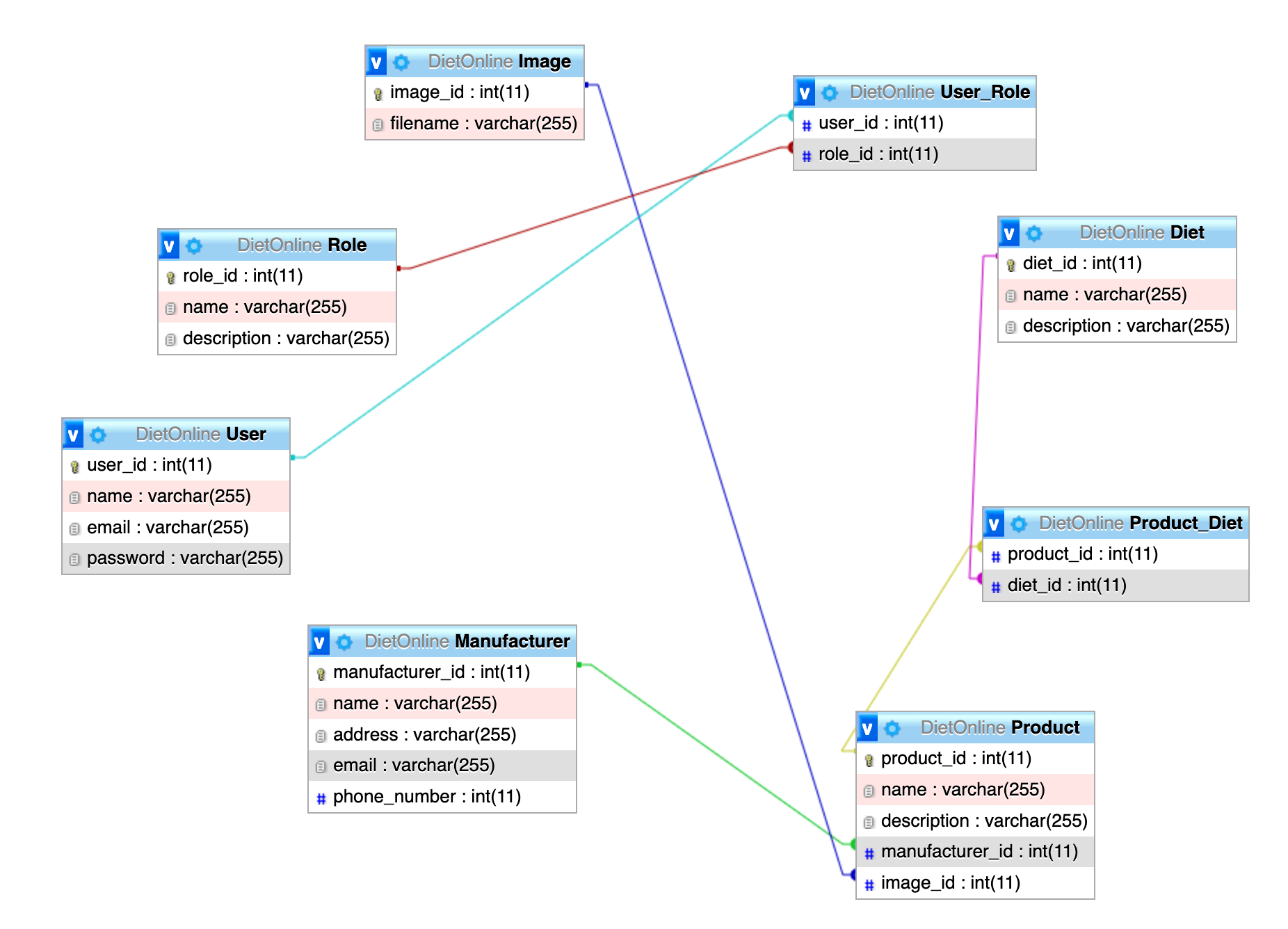


^ The entities and relationships between the entities are illustrated using this ERD.

## Tables

A data scheme is a more detailed ERD with attributes, primary and foreign keys, and linking tables.

The details which describe an entity are known as attributes, an attribute is a description of the entity.



^ The entities, attributes, primary and foreign keys, and relationships between the entities are illustrated using this Data Schema.

## Database Dictionary

**Diet**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| diet\_id | int | N/A | Yes | PK |  |
| name | varchar | N/A | Yes |  |  |
| description | varchar | N/A | No |  |  |

**Image**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| image\_id | int | N/A | Yes | PK |  |
| filename | varchar | N/A | Yes |  |  |

**Manufacturer**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| manufactuer\_id | int | N/A | Yes | PK |  |
| name | varchar | N/A | Yes |  |  |
| description | varchar | N/A | Yes |  |  |

**Product**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| product\_id | int | N/A | Yes | PK |  |
| name | varchar | N/A | Yes |  |  |
| description | varchar | N/A | Yes |  |  |
| manufacturer\_id | int | N/A | Yes | FK | Manufacturer |
| image\_id | int | N/A | No | FK | Image |

**Product\_Diet**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| product\_id | int | N/A | Yes | FK | Product |
| diet\_id | int | N/A | Yes | FK | Diet |

**Role**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| role\_id | int | N/A | Yes | PK |  |
| name | varchar | N/A | Yes |  |  |
| description | varchar | N/A | Yes |  |  |

**User**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| user\_id | int | N/A | Yes | PK |  |
| name | varchar | N/A | Yes |  |  |
| email | varchar | N/A | Yes |  |  |
| password | varchar | N/A | Yes |  |  |

**User\_Role**

| **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Reference Table** |
| --- | --- | --- | --- | --- | --- |
| user\_id | int | N/A | Yes | FK | User |
| role\_id | int | N/A | Yes | FK | Role |

# 

# System Design/ Architecture Overview

## Introduction

Laravel is the web framework I'll be using. Laravel is a web application framework that provides structure and website building.

One of the best options for creating modern websites is Laravel. Laravel can be used with a variety of tools. Laravel is a progressive framework that works for both beginners and experts. It provides basic tools for beginners and advanced ones for more experienced users, like unit testing.

## Model View Controller

The views show the various pages you see. The controllers take the user's action when they click on a certain product to view more information. This can involve using the models to get data from a data source.

The manufacturers and products will use a view, model, and controller. For each, there will be an index view (showing all the items) and a single view (showing a single item).

## User Authentication

Breeze typically needs to be installed separately. But instead, we are using the - -auth command to add user authentication while downloading Bootstrap.

Despite not being Breeze, it makes uses the same concept. This allows user and employee login, signup, password reset, and email verification.

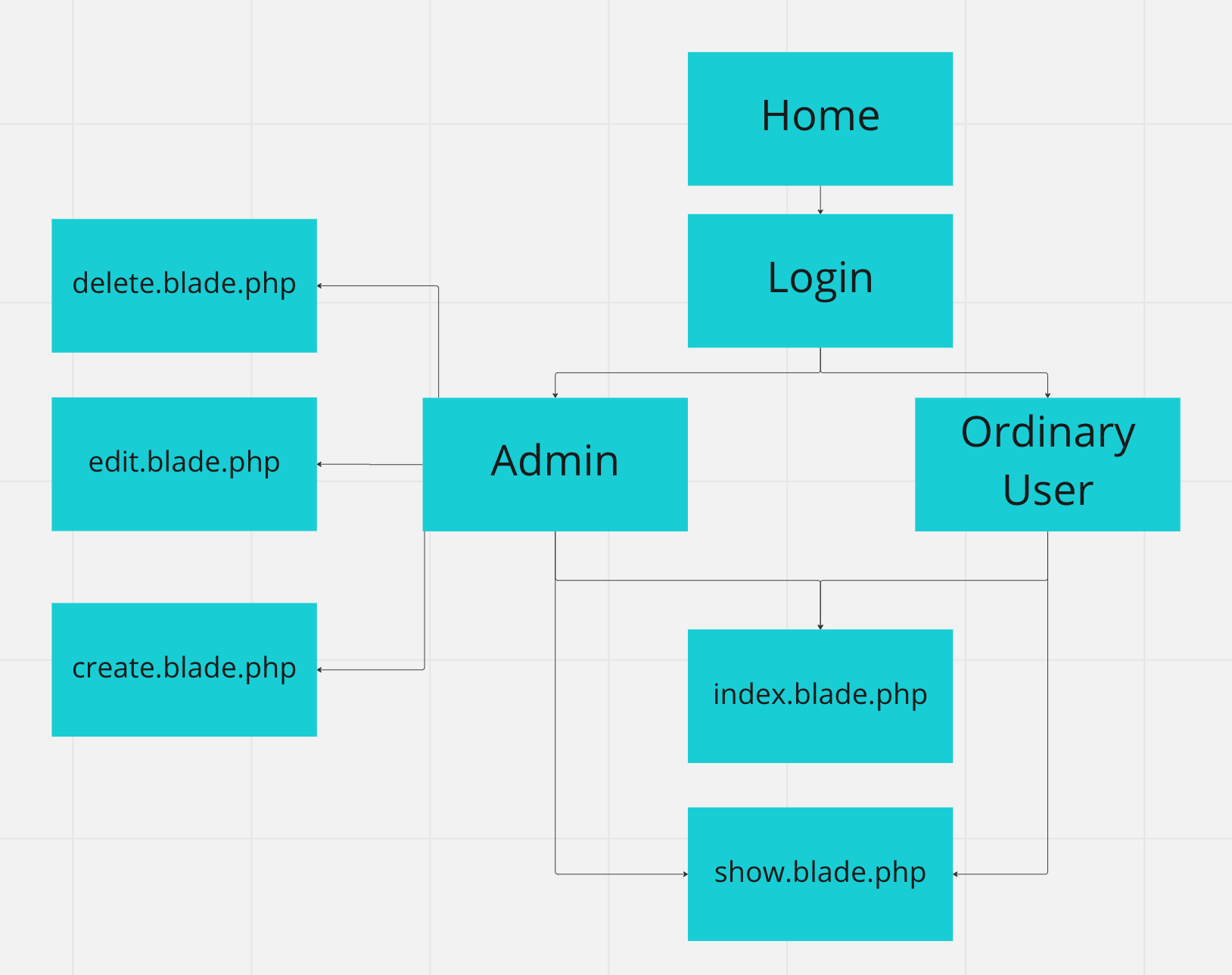
## Routing

The routes can be controlled in web.php. There should be a route for each of the view pages, or .blade.php, in this file.

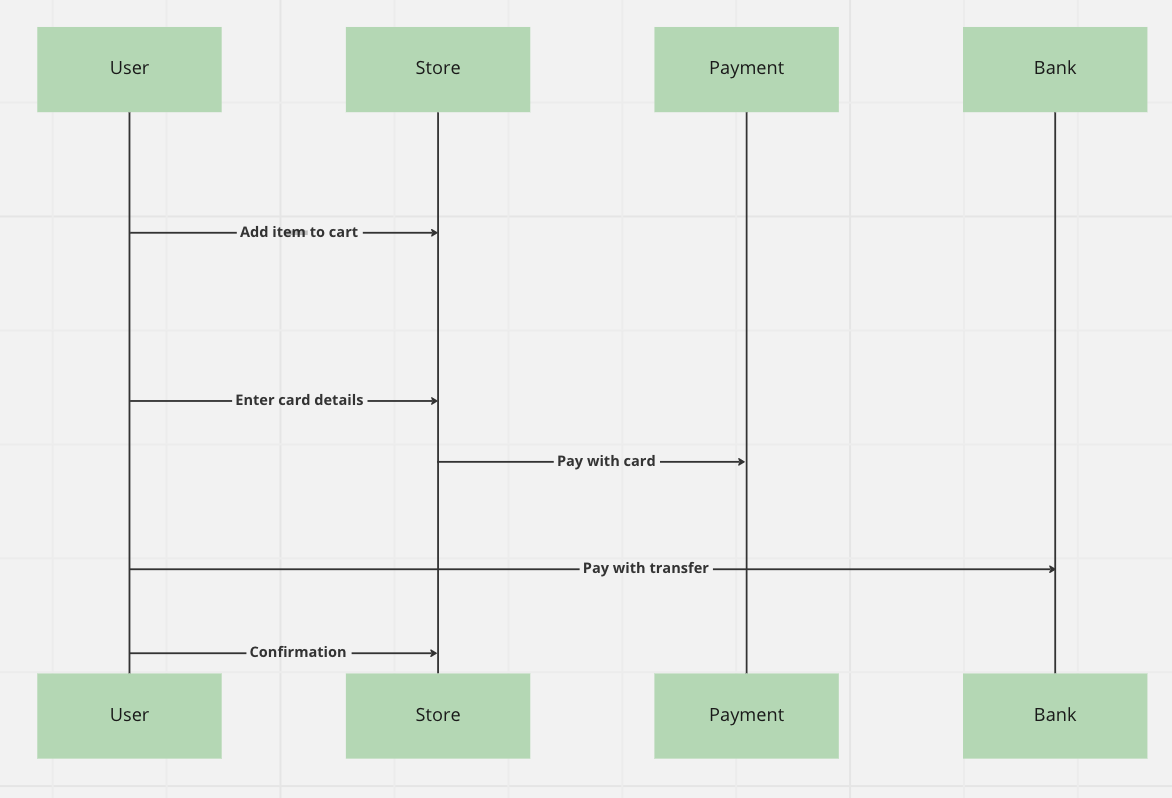
There are two main types of users: admin and ordinary user. The views displayed depend on the role the user has when logging in. While the admin will also have access to the create, update, and delete forms, the ordinary user should only have access to the view all and single view pages.

## Templating

The simple yet useful templating engine with Laravel is called Blade. Unlike several PHP templating engines, the blade does not stop you from using plain PHP code in your templates. Blade template files are normally in the resources/views directory and have the .blade.php file extension.



^ This sitemap shows the pages each type of user can use.



^ This sequence diagram shows the scenario in which a customer buys goods from the store.

# Testing

* 1. Introduction

Even though I only had a short amount of time to test the website, I was still able to do some simple tasks to see whether it worked as intended.

The code was functionally tested to make sure that there were no errors or broken links.

User Testing was used to see whether the website's flow and look made sense and worked well.

* 1. Functional Testing

## Login/Registration

1. Signing in with the account that is generated automatically served as the first test. This turned out to be very simple, but I had expected that. They had no trouble signing in and viewing the various user’s pages.
2. The second test was to create an account. Because there was no method to tell the database the user's role, this was much more difficult than I had expected. As a result, even though the user could create the account, the database was unsure of which views it should be sending out.

## Navigation

Simply making sure that all of the links worked was the only test necessary for the navigation. This was also very easy. There were no broken links or errors; all the links went to the right pages. The only page that they said didn’t work with the links was the home page.

## Calculation

## CRUD

1. All of the creation was tested as the initial test. We checked to see if each form was creating each object. Because I ran out of time to add the picture part, the product create form was the only one that didn't work.
2. The read part was checked in the navigation test.
3. The second test was for the update, this worked the exact same as the create.
4. The last test was to test the delete functionality. This was the only part of a product that I could get working with the image (apart from Read). All the delete buttons successfully delete the correct data.

## Discussion of Functional Testing Results

The Functional Testing was overall successful I was able to tell what was working and what didn’t work. I know that I need to fix the edit and creat forms for the product.

* 1. User Testing

At the end of the Functional testing, I asked the users to rate and comment on the website's appearance and usability. Overall, people said that although the website's somewhat simple appearance, they thought it looked fine. They also thought the website had a very good flow; they could easily locate everything, and everything made sense from a flow perspective.

* 1. Conclusion

I can conclude the tests. I can conclude that the project's guts are present; it simply needs a little extra to stand out more. To make the CRUD and Navigation. a little bit simpler to perform in the future, I can also consider cleaning up some of the techniques used to create some of these tasks.

# Project Management

## Introduction

In this section I will be talking about the project management I used throughout my project. Although I would have loved to use more applications or tools to help me with my project, I just used GitHub.

GitHub is a web-based platform for version control and collaboration. Git is used to store a project's source code and keeps a history of all code changes.

## Project Phases

## Requirements

The requirements phase is important for developers to understand what users want the application to do.

I thought about what I wanted to add to the website and what others wanted me to.

## Design

For this project, I did three different types of design:

* I made an ERD, a database schema, and an example database for Database Design.
* Created a wireframe and a style guide for Web Application Design.
* I looked into possible additions to Laravel's functionality for Software Design.

## Implementation

During this phase I created the project. I created the HTML for the project, I also included the Bootstrap framework to style the website. I then added the data to the web pages and CRUD functionality using PHP.

## Testing

Even though I only had a short amount of time to test the website, I could still do some simple tasks to see whether it worked as intended.

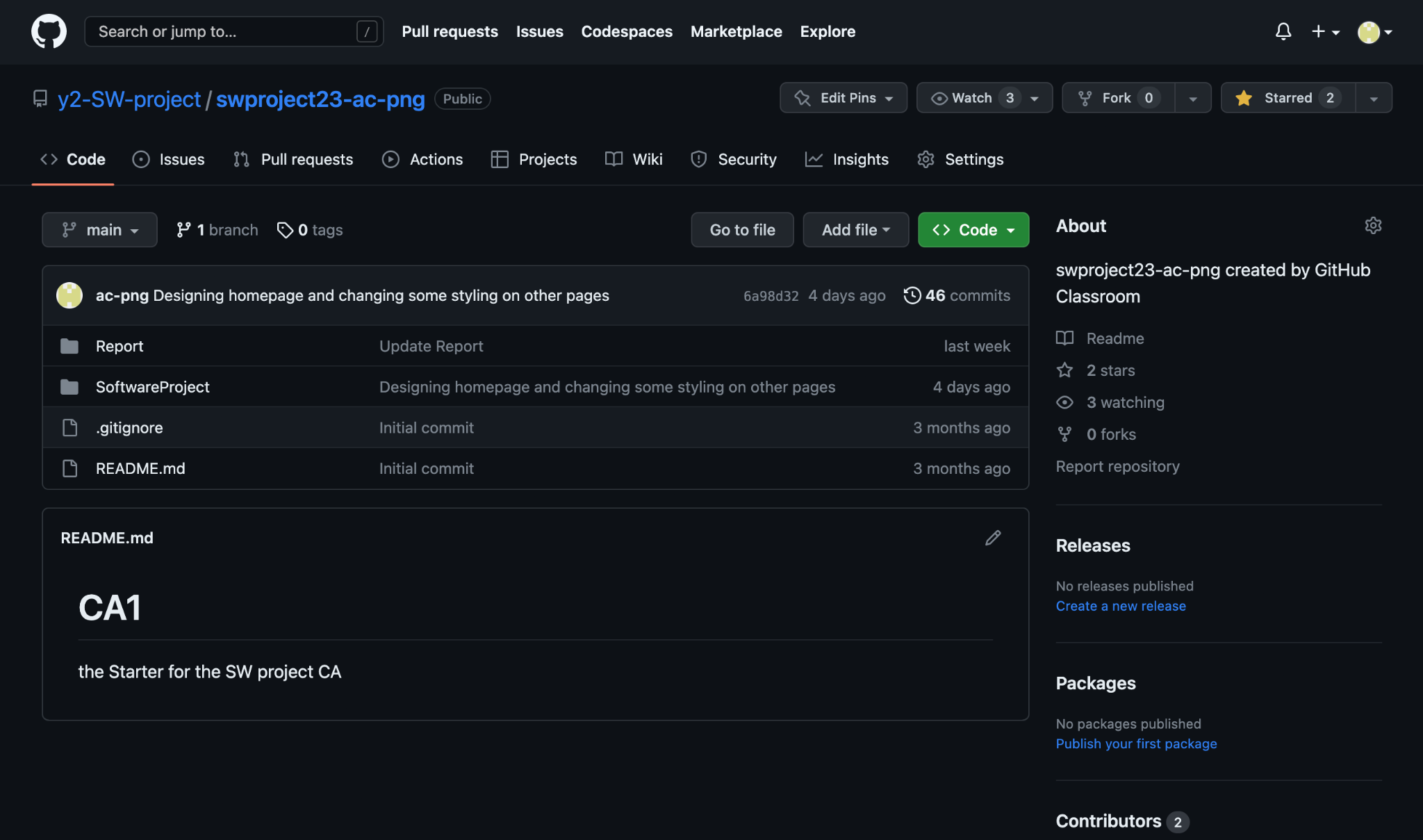
## Project Management Tools

## GitHub Project

## GitHub

For the entire project management, I used GitHub. I took my instructor's assignment and made a cloned repository on my laptop.

Throughout the project, I often used this repository. I would add to the repository by committing the changes whenever I added a new functionality or at the end of the day.



# Reflection

## Your views on the project

I believe that I have generally worked very consistently during the course of the project.

I believe that the process has been helped by my previous experience in making similar projects. I know what worked and what didn't with the prior projects, and I can now improve some of the things I didn't complete.

## How could the project be developed further?

There are a few items that I either didn’t get to do or could not finish:

* I had wanted to include basket functionality, but I didn’t have the time to do so.
* Although I wanted to replace the faker data with genuine data, I wanted to get the CRUD capability up and running first,
* Despite the fact that I did the edit and create for the products before the images were added. Although it didn't work, I attempted to add picture capabilities to the forms.
* I did not add form validation capabilities since I wanted to test the form before including it.
* The home page's style and links are incomplete.

## Assessment of your learning.

The skills and competencies I have learnt are outlined in 9.5. Technical Skills and 9.6. Further Competencies and Skills sections.

Further developments are outlined in the previous section 9.2. How could the project be developed further?

## Completing a large software development project

I used various time and project management techniques to stay on track during the project. To finish this assignment, I had to use my previous modules' knowledge; this simplified things because I knew what areas I needed to focus on the most.

## Technical skills

Through this project, I improved and learned the following technical skills:

* Database: I created an ERD and Data Schema at the beginning to decide the data names and types I needed. To add the database to phpMyAdmin and see how the database should look, I also wrote the SQL code for the database.
* PHP: Although I had used PHP in my projects, I was able to improve my abilities slightly. In my controllers and models, I could do more advanced functions.
* Bootstrap: Although I had used Bootstrap in one of my projects, I was able to improve my abilities slightly. On my web pages, I combined PHP and Bootstrap to add some data to the HTML.

## Further competencies and skills

I would have liked to begin working on the technical parts of the project right away since I would have had more time to include validation and add additional features.

I should have used different apps/programs to keep track of my progress and manage my tasks. Trello might have been useful, but I lacked the confidence to use it and adapt it to my project. Making notes and organising a little bit might have helped by using Notion.

# References

**Researched Websites**

*Kroger : Fresh Food. Low Prices. | Shop Groceries Online*. (n.d.). Kroger : Fresh Food. Low Prices. | Shop Groceries Online. <https://www.kroger.com/>

*VeganEssentials | Best Online Vegan Grocery Store*. (n.d.). Vegan Essentials Online Store. <https://veganessentials.com/>

**Websites to Complete Project**

*Stack Overflow - Where Developers Learn, Share, & Build Careers*. (n.d.). Stack Overflow. <https://stackoverflow.com/>

Jacob Thornton, and Bootstrap contributors, M. O. (n.d.). *Get started with Bootstrap*. Get Started With Bootstrap · Bootstrap v5.3. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

*Laravel - The PHP Framework For Web Artisans*. (n.d.). Installation - Laravel - the PHP Framework for Web Artisans. <https://laravel.com/>