

National College of Ireland

BSC (Honours) in Computing - Evening

Software Development

Academic Year 2021/2022

Gustavo Camello

X18119956

x18119956@student.ncirl.ie

eCamell – eCommerce for Small Business

Technical Report

Contents

Executive Summary.....	3
1.0 Introduction	3
1.1. Background.....	3
1.2. Aims	3
1.3. Technology	3
2.0 System.....	5
2.1. Requirements	5
2.1.1. Functional Requirements to the Admin Portal	5
2.1.1.1. Requirement 1: Log in	5
2.1.1.2. Requirement 2: Manage Products	6
2.1.1.3. Requirement 3: Manage Categories	7
2.1.1.4. Requirement 4: See All Orders.....	9
2.1.1.5. Requirement 5: Customize StoreFront	10
2.1.2. Functional Requirements to the StoreFront	11
2.1.2.1. Requirement 6: View Products	11
2.1.2.2. Requirement 7: Create an Account.....	12
2.1.2.3. Requirement 8: Log in	13
2.1.2.4. Requirement 9: See Orders.....	14
2.1.2.5. Requirement 10: Update Details.....	15
2.1.2.6. Requirement 11: Add Items to Shopping Cart	16
2.1.2.7. Requirement 12: Manage Shopping Cart.....	17
2.1.2.8. Requirement 13: Place an Order.....	18
2.1.3. Data Requirements	20
2.1.4. User Requirements.....	20
2.1.5. Environmental Requirements	20
2.1.6. Usability Requirements	20
2.2. Design & Architecture	20
2.3. Graphical User Interface (GUI)	22
2.4. Testing	38
2.5. Evaluation	42
3.0 Conclusions	43
4.0 Further Development or Research	44
5.0 References	45

6.0	Appendices.....	47
6.1.	Project Proposal	47
1.0	Objectives	49
2.0	Background	49
3.0	State of the Art.....	50
4.0	Technical Approach.....	50
5.0	Technical Details	51
6.0	Special Resources Required	52
7.0	Project Plan	52
8.0	Testing.....	54
6.2.	Reflective Journals.....	55
6.3.	GitHub Repositories	60

Executive Summary

1.0 Introduction

1.1. Background

In today's society, consumers are more likely to shop online rather than physically visit a store. Some members of the country's leadership advise citizens not to venture outside and to keep their critical activities online while at home. As a result, internet buying is becoming more appealing, and the online market is becoming larger than ever.

To help small business to achieve more sales, I am developing the eCamell, a simple and easy to use and maintain eCommerce platform, where business can sell their products online and reach a lot more people.

I choose to make this project because I am really interested in eCommerce platforms, it is something that really gets my attention and I want to learn more about how the functionalities of one works. Also I believed that an fully eCommerce is a good showcase project to have.

1.2. Aims

The aim of this project, is to develop a eCommerce platform to small and medium business, an easy and simple to use admin panel to the business owners manage the commerce and a visual appealing and attractive storefront to the final users.

The application will be a headless commerce, with microservices architecture in the backend. This will provide a more customizable application, easy to maintain and in a business perspective, we will allow the business to use only our storefront or our backend services, making the final application much more flexible.

1.3. Technology

The project will be developed using JavaScript as the main language, because is the programming language that I'm more comfortable in use.

The approach decided to the application is to create a headless commerce platform, where we have the back-end decoupled from the front-end (already mentioned on this document).

The back-end will be created using the microservices architecture with each main module of the application will have its own service.

The front-end (client) will be developed using Next.js, a react framework to create UI components.

The below list shows the languages, libraries, database, etc that the project intend to use:

Backend:

- Node.js: the headless CMS that will be used in the project is built in Node.
- Keystone.js: Keystone is a headless CMS build in Node.js. I decided to use it, because I have an interesting in learn how CMS works and also to speed up the creation of the microservices.
- GraphQL: Keystone offers a GraphQL API or REST API. The project will use the GraphQL API as it is faster and more efficient than REST.

Frontend:

- Next.js: Next.js is a React framework for server side rendering, routing and tooling.
- React.js: the UI of the application will be built in React.js. A JavaScript library.
- Apollo Client: for data management. Connect the frontend to the GraphQL API.
- Tailwind: CSS framework to style the application

Others:

- MongoDB: a non-relational database.
- Jest.js: JavaScript Test Framework, to help test the application.
- Stripe: use of Stripe platform to allow payments.

2.0 System

2.1. Requirements

This section explain the main requirements of the application. The functional requirements are divided in two sections: one for the requirements of the Admin Portal and another for the Storefront (the end user interacts with the application).

2.1.1. Functional Requirements to the Admin Portal

Requirements to the Admin portal (used by the admin of the system). The admin user is the main actor.

2.1.1.1. Requirement 1: Log in

Scope

The scope is to authenticate the user.

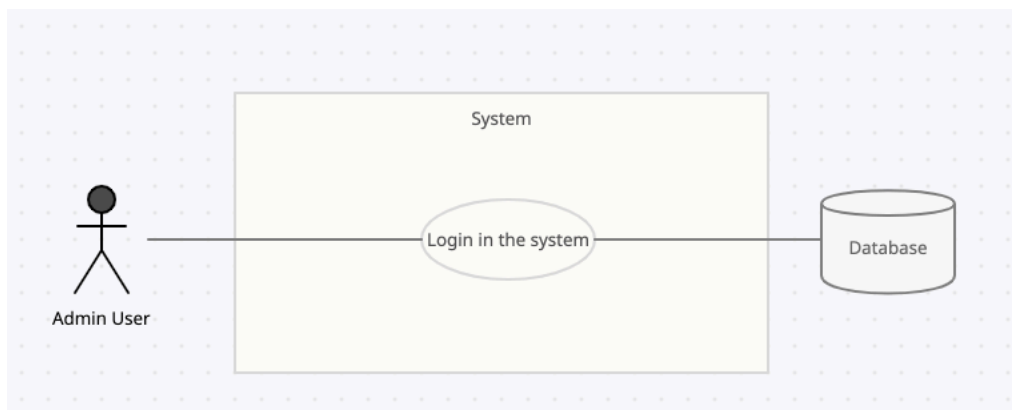
Description

This function ensure that only admins users can access the admin portal and make changes in the application.

Actors

admin user, admin portal, database

Use Case Diagram



Flow Description

Precondition

The system is on and the user is already registered in the system.

Activation

The use case starts when user click on the login button.

Main Flow

1. User access the login page.

2. The user enter the password and email.
3. The user clicks in the Login Button.
4. If credentials are correct, user is redirect to the admin dashboard.

Alternate Flow

A. Wrong credentials

1. User entered the password or email incorrectly, the system asks to try again.
2. Alternate flow A ends.

B. User do not have an account

1. The system asks to the user to create an account.
2. User repeat the main flow.

Termination

The system presents the admin dashboard to the admin user.

Postcondition

The user is logged in.

2.1.1.2. Requirement 2: Manage Products

Scope

This use case allow admin users to manage the products.

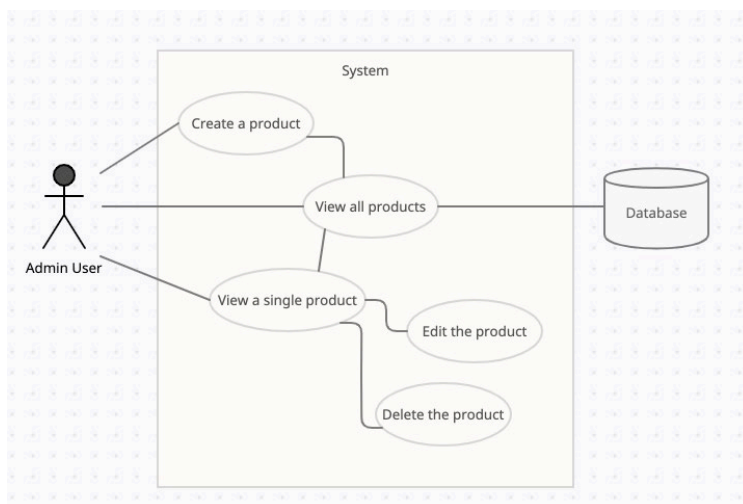
Description

This function allow authorised users to create, delete, update information about the products in the system.

Actors

admin user, admin portal, database

Use Case Diagram



Flow Description

Precondition

System is showing the admin dashboard to the user.

Activation

User clicks in the Products card in the dashboard.

Main Flow

1. System display the list with all products.
2. User clicks in the Create new product button.
3. System shows the form to the user create a new product.
4. User fill-up the form with the data from the new product and submit the form.
5. The system check if the form is correctly filled-up and save the new product.
6. The user can review the newly created product.

Alternate Flow

A. Update product

1. User clicks in a product from the all products list.
2. System shows the info about the product and allow user to change it.
3. User update the products info.
4. End of alternate flow A.

B. Delete product

1. User clicks in a product from the all products list.
2. System display the product page information.
3. User clicks in the Delete product button.
4. End of alternate flow B.

Termination

The system updates the database accordingly and display the all products list to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.1.3. Requirement 3: Manage Categories

Scope

This use case allow admin users to manage the products categories.

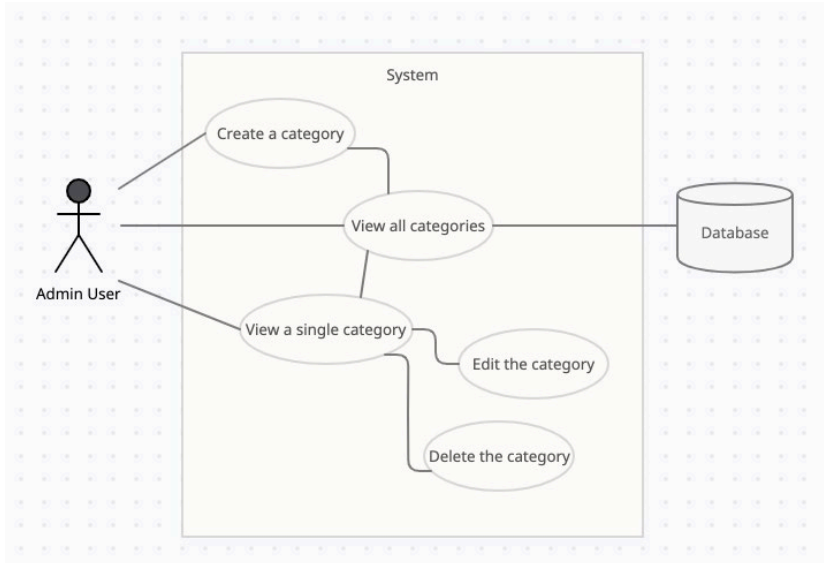
Description

This function allow authorised users to create, delete and update the products categories in the system.

Actors

admin user, admin portal, database

Use Case Diagram



Precondition

System is showing the admin dashboard to the user.

Activation

User clicks in the Categories card in the dashboard.

Main Flow

1. System display the list with all categories.
2. User clicks in the Create new category button.
3. System shows the form to the user create a new category.
4. User fill-up the form with the data from the new category and submit the form.
5. The system check if the form is correctly filled-up and create the new category.
6. The user can see the newly created category and associate products to it.

Alternate Flow

A. Update category

1. User clicks in a category from the all categories list.
2. System shows the info about the category.
3. User update the category info.
4. End of alternate flow A.

B. Delete product

1. User clicks in a category from the all categories list.
2. System display the category page information.

3. User clicks in the Delete product button.

4. End of alternate flow B.

Termination

The system updates the database accordingly and display the all categories list to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.1.4. Requirement 4: See All Orders

Scope

This scope is to give an option to the admin user to view and be aware of all orders made in the system.

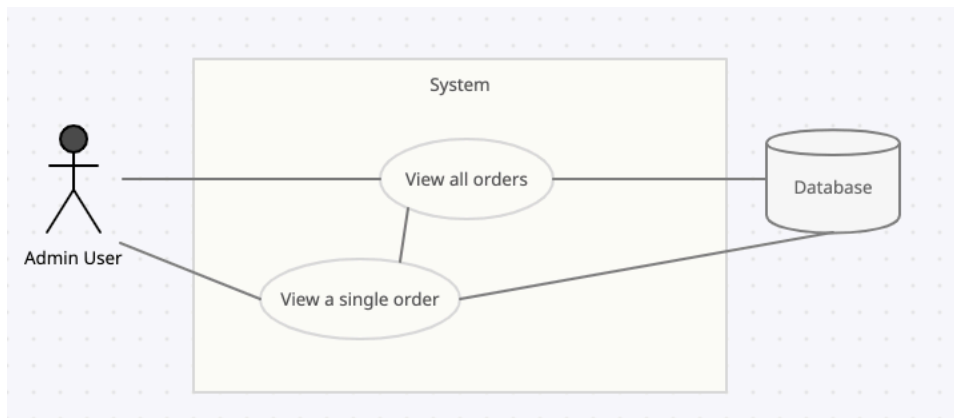
Description

This function allows the admin user to see all orders made in the system through the Storefront. It allow the admin to see the details of the orders.

Actors

admin user, admin portal, database.

Use Case Diagram



Flow Description

Precondition

The user is already logged and the system is showing the admin dashboard to the user.

Activation

The user clicks in the Orders card in the admin dashboard.

Main Flow

1. The user clicks on the Orders card.
2. The system shows the all orders list to the user.
3. The user clicks in an order.
4. The system shows to the user the details of the order.

Termination

The system display the all orders list to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.1.5. Requirement 5: Customize StoreFront

Scope

The scope of this use case is to allow the admin to customize some styles from the storefront.

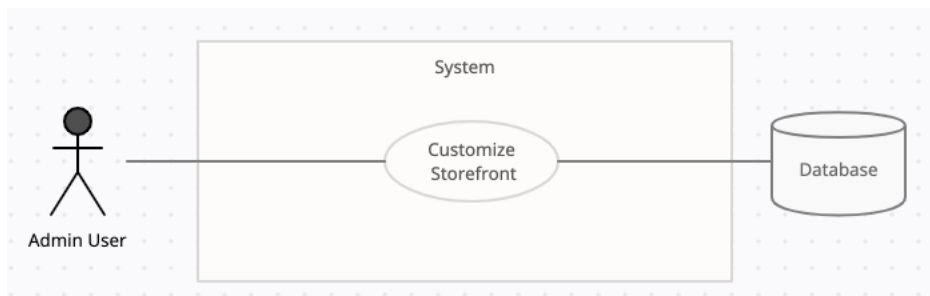
Description

This function allows the admin to define some basic customizations to the Storefront like main and secondary colour for example.

Actors

admin user, admin portal, database

Use Case Diagram



Flow Description

Precondition

The system is on and is showing the admin dashboard to the user.

Activation

The user clicks on the Customize Storefront card in the admin dashboard.

Main Flow

1. User clicks in the card.
2. System present the user with the current values and allow user to update it.
3. User update the information and save the data.
4. System save the into the database and send it to the StoreFront.

Termination

The system presents the admin dashboard to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2. Functional Requirements to the StoreFront

Requirements to the Storefront application (the UI where the final user interacts with the system).
The final user is the actor.

2.1.2.1. Requirement 6: View Products

Scope

The scope of this use case is to show and allow the user to search for products in the system.

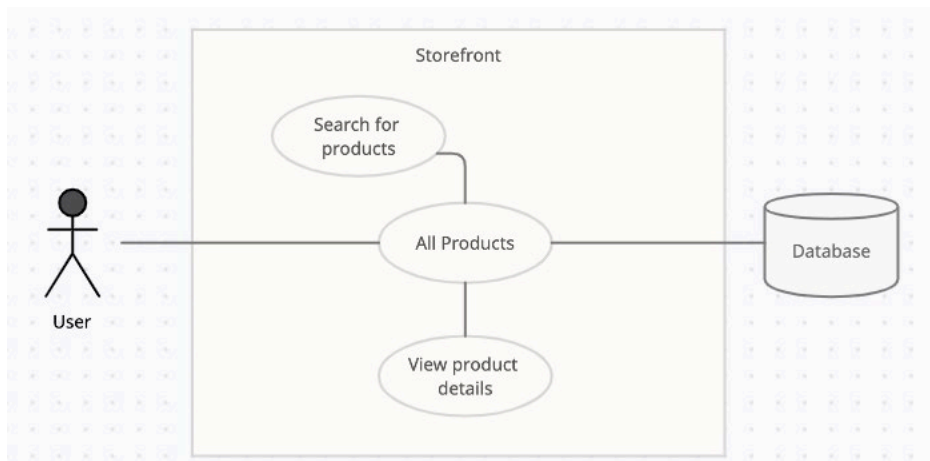
Description

This function allows the user to search and see the products the eCommerce is currently selling. The user can see more details of a product clicking on it to be redirected to the single product page.

Actors

user, storefront, database

Use Case Diagram



Flow Description

Precondition

The system is displaying the homepage of the Storefront.

Activation

The use case starts when the user interacts with the UI, scrolling in the products pages, clicking in the categories and clicking in a single product.

Main Flow

1. The user goes to the Storefront homepage.
2. The system show to the user some products.
3. The user interact with the UI clicking in a single product.
4. The system redirects the user to the product details page.

Alternate Flow

A. Search bar

1. The user type a word in the search bar.
2. The system return all products found with the word entered in the search bar.
3. The user click in the product.
4. The system redirects the user to the product details page.

Termination

The user is on the product details page.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2.2. Requirement 7: Create an Account

Scope

The scope of this use case is allow the user to create an account in the system.

Description

This function ensures that a new user can create an account into the system, making possible to them to place orders, make payments, see past orders, etc.

Actors

user, storefront, database

Use Case Diagram



Flow Description

Precondition

The system is displaying the Register page.

Activation

The use case starts when user click on the Register button.

Main Flow

1. The system show the form to the user enter its credentials.
2. The user provide all necessary credentials to create an account and click on Register.

3. The system checks if the form was correctly filled-up and create the new account.
4. User can access its My Account page and start to shopping.

Alternate Flow

A. User already registered

1. The system show the form to the user enter its credentials.
2. The user provide all necessary credentials to create an account and click on Register.
3. System checks that there is already an account created with its credentials.
4. System lets the user know that an account already exists with the credentials.
5. User can login with the credentials.

Termination

The system redirects the user to the homepage of the Storefront.

Postcondition

The user can access its My Account page and interact with it.

2.1.2.3. Requirement 8: Log in

Scope

The scope is to authenticate the user.

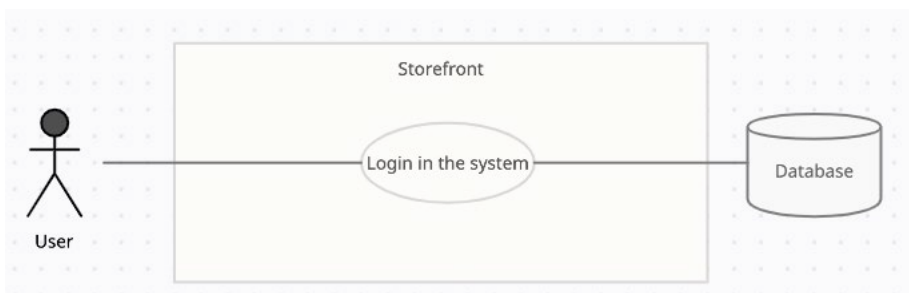
Description

This function ensures that the system can keep a track of the actions of the user, like the orders it has been made, update its details and make payments.

Actors

user, storefront, database

Use Case Diagram



Flow Description

Precondition

The system is on and the user is already registered in the system.

Activation

The use case starts when user click on the login button.

Main Flow

1. User access the login page.
2. The user enter the password and email.
3. The user clicks in the Login Button.
4. If credentials are correct, user is redirect to the homepage of the StoreFront.

Alternate Flow

A. Wrong credentials

1. User entered the password or email incorrectly, the system asks to try again.
2. Alternate flow A ends.

B. User do not have an account

1. The system asks to the user to create an account.
2. User repeat the main flow.

Exceptional Flow

A. User forgot the password

1. User clicks in the button Forgot the password.
2. System sends an email to the user with instructions to reset the password.
3. Exceptional flow A ends.

Termination

The system presents the homepage of the Storefront to the user.

Postcondition

The user is logged in.

[2.1.2.4. Requirement 9: See Orders](#)

Scope

This scope is to give the user the ability to see the orders it has already been made by them.

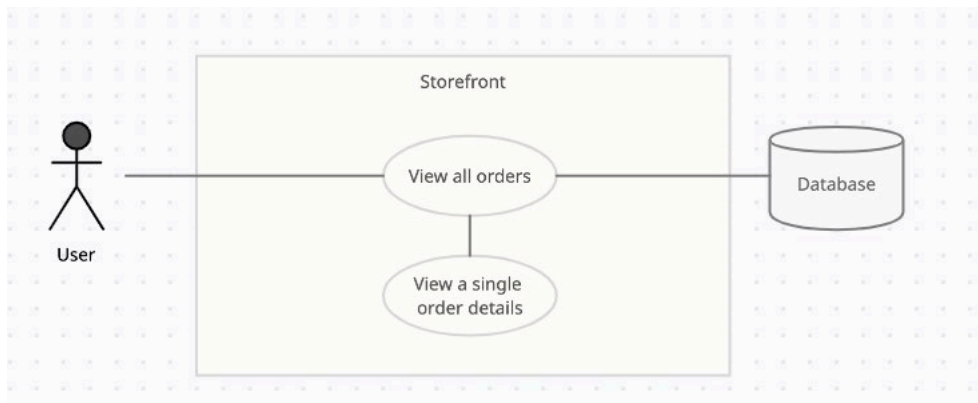
Description

This function allows the user to see all orders and the details of the orders that has already been made by them.

Actors

user, storefront, database.

Use Case Diagram



Flow Description

Precondition

The user is already logged in the system.

Activation

The user clicks in the My Account link in the homepage.

Main Flow

1. The user access the My Account page.
2. The user clicks in Orders in the submenu.
3. The system shows the orders to the user.
3. The user clicks in an order.
4. The system shows to the user the details of the order.

Exceptional Flow

A. User has not made any order

1. The user access the My Account page.
2. The user clicks in Orders in the submenu.
3. The system display a message to warn the user that there is no orders to show.
4. End of the exceptional flow.

Termination

The system display the orders list to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2.5. Requirement 10: Update Details

Scope

This scope gives the user the option to update its details in the system.

Description

This function allows the user to update its details in the system. The user can change address, phone number, name, etc.

Actors

user, storefront, database.

Use Case Diagram



Flow Description

Precondition

The user is already logged in the system.

Activation

The user clicks in the My Account link in the homepage.

Main Flow

1. The user access the My Account page.
2. The user clicks in My Details in the submenu.
3. The system shows the details of the user and allow them to be updated.
4. The user update the details as necessary.
5. The system save the new user details in the database.

Termination

The system display the new user details.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2.6. Requirement 11: Add Items to Shopping Cart

Scope

This scope allows the user to add items to the shopping cart.

Description

This function allow the users to add items that they want to buy in their cart shopping. The system save this products and wait for the user to remove from the cart or place the order.

Actors

user, storefront, database.

Use Case Diagram



Flow Description

Precondition

The user select a product that its desire to buy and the product must be available in stock.

Activation

The use case starts when the user click on the button Add to cart in a product details page.

Main Flow

1. The user selects the item to add to the cart.
2. The system add the item to the cart.
3. The system update the user's cart.

Termination

The system shows the cart page with the item selected.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2.7. Requirement 12: Manage Shopping Cart

Scope

The scope of this use case is to allow the user to manage its cart shopping.

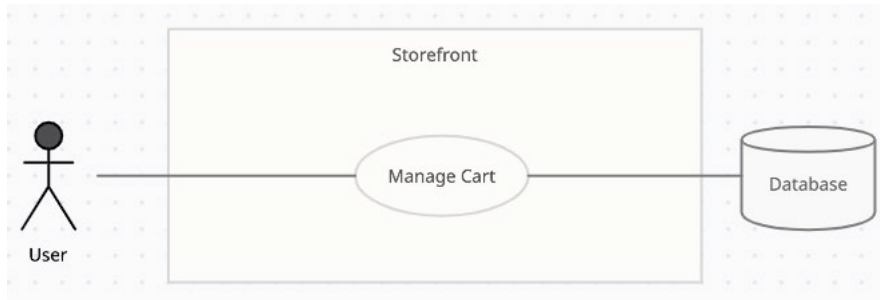
Description

This function allow the user to manage its cart shopping before place an order. The user can remove products, increase or decrease the quantity, select the payment method before place an order.

Actors

user, storefront, database.

Use Case Diagram



Flow Description

Precondition

There must be at least one item in the cart to be able to manage the cart shopping.

Activation

The use case starts when the user navigates to the Cart page.

Main Flow

1. The user increase or decrease a product quantity or remove an item.
2. The system update the cart with the correct quantity and also the total order amount accordingly with the new quantity.
3. The system update the user's cart.
4. The system show to the user the updated cart shopping.

Termination

The system shows the updated cart shopping to the user.

Postcondition

The system goes to a wait state until user interacts with the UI.

2.1.2.8. Requirement 13: Place an Order

Scope

This scope allows the user to place an order.

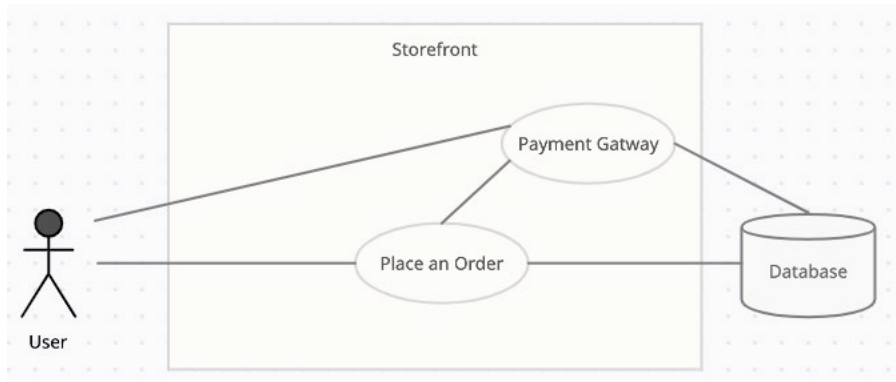
Description

This function allow the user to place an order with the items its desired to buy. After the user check the items in the cart and press the button place the order, its redirect to make the payment.

Actors

user, storefront, payment gateway, database.

Use Case Diagram



Flow Description

Precondition

There is must be at least one item in the cart to be able to place an order.

Activation

The use case starts when the user click in the Place an Order button.

Main Flow

1. The user clicks in the Place an Order button.
2. The system check if the user is logged in.
3. The system redirect the user to the payment gateway.
4. The user pays through the payment gateway.
5. The system receives the response that the user has made the payment.
6. The system display the Successful order page to the user.

Alternate Flow

A. User is not logged in

1. The user clicks in the Place an Order button.
2. The system detects that the user is not logged in.
3. The system redirect the user to the login page.
4. After logged in, the user can continue with the main flow.

Exceptional Flow

A. The system cannot connect to the payment service

1. The user clicks in the Place an Order button.
2. The system try to redirect the user to the payment gateway.
3. The payment gateway responds with an error.
4. The system let the user know about the problem.
5. The user is unable to place the order at that time.
5. End of exceptional flow.

Termination

The user successfully made an order

Postcondition

The system display the Successful order page to the user.

2.1.3. Data Requirements

Different from the project proposal, the database used in the application will be Postgres instead of the MongoDB. The only reason for that change is that the Keystone.js (the technology I am using in the backend) is not supporting Mongo in the recent update, and I always want to use the last version of the technologies I'm working with. The PostgreSQL is a free, open-source relational database.

In this stage, the tables expected in the application are Products, Users, Orders and Shopping Cart.

2.1.4. User Requirements

Admin user requirements:

- Admins should be able to manage the application through an admin dashboard.
- Admins should be able to manage products use all CRUD (create, read, update and delete) operations.
- Admins should be able to see all orders.

Final user requirements:

- Customers should be able to access the storefront from any web browser and screen.
- Users must be able to search and see products in detail.
- Users must be able to select items that they desire to buy and add them into the their shopping cart.
- Users should be able to manage items in the shopping cart.
- Payments must be secure.

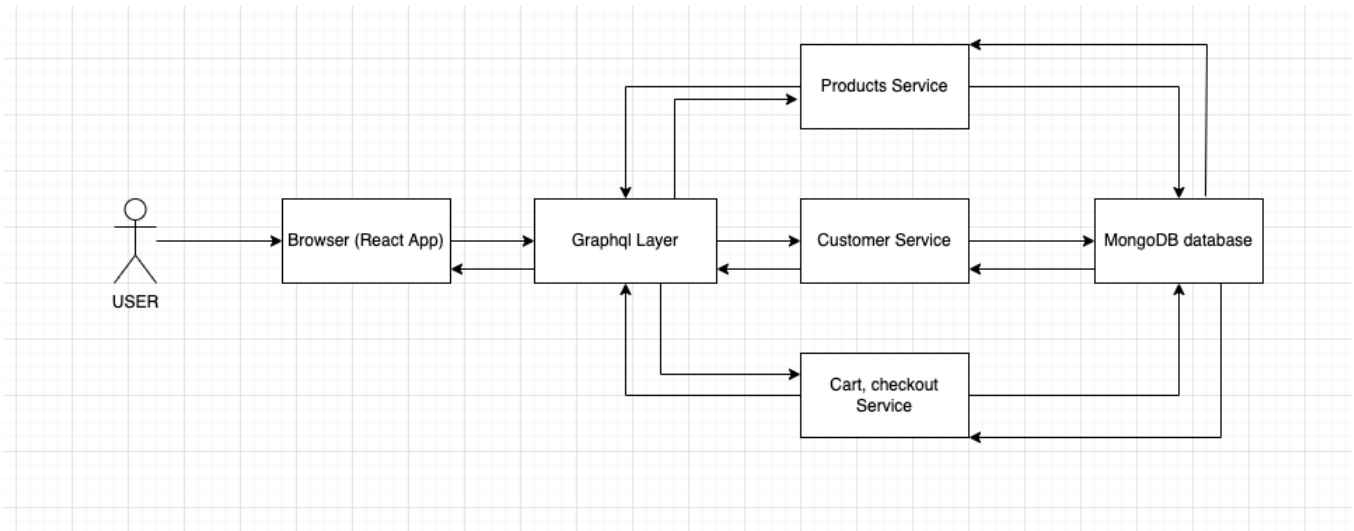
2.1.5. Environmental Requirements

To access the application, users must be connected to the internet and the application can be accessed from computes, smartphones, tables, etc.

2.1.6. Usability Requirements

The application must be user and mobile friendly, for that reason the application was developed using the Tailwind, a library that help us to develop responsive applications, making the application easy to use and accessible in different screens sizes.

2.2. Design & Architecture



eCamell is constructed on a MERN Full-Stack Framework. The application is made of 4 individually parts that together make the application works.

The front end of the storefront and the admin dashboard are developed using React.js and the backend of the application are built with Keystone(Node.js) and MongoDB as a database.

The application works on a request/response cycle. The user request a webpage and depending of the page, the GraphQL layer send the request to the corresponding API. For example, in the homepage, to display the products, a call is made to the products API and return all the products in the database.

Front end

React.js

The interfaces for interacting with the eCamell are designed using the React.JS framework. This approach allows the creation of components for web pages in the application.

ApolloClient

It is the client responsible to make the requests to the GraphQL API.

Back end

Each service of the application are developed using the Keystone CMS. The services (APIs) share the same database.

Keystone.js

Keystone is the CMS that help us to develop our backend and GraphQL APIs. It is developed in Node.js.

MongoDB

A non-relational database was used to develop the application.

2.3. Graphical User Interface (GUI)

There is two UI for the application: the storefront and the admin dashboard to manage the eCommerce. Below there are the screenshots of the key screens of each UI.

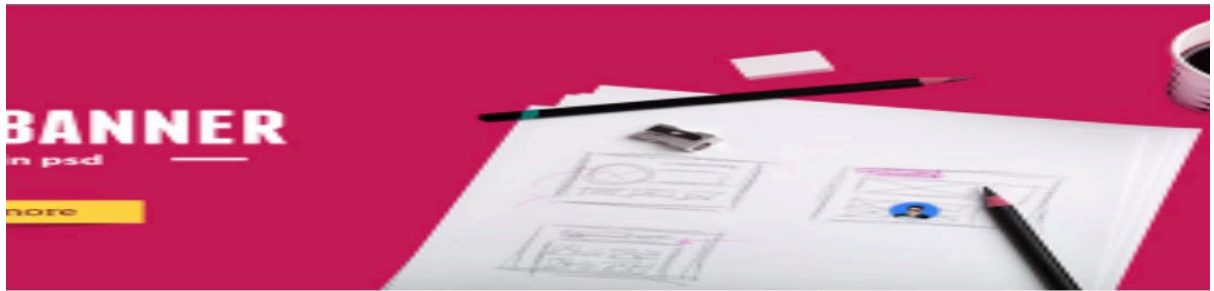
Store Front

Home Page

The home page is the main page of the eCommerce. The user can see the featured products, a banner with some promotional content and also the navigation bar (which is present in all pages) with a search bar, links to different products categories, link to the cart and links to login in the eCommerce.

Search for an item

KEYBOARDS MOUSES MEMORY



! New Products and Releases



Keyboards
The Quick Brown Fox Jumps Over The Lazy Dog
€10.00



Keyboards
Bathroom Wall And Cabinet Organiser In Satin
€20.00



Keyboards
In 1 Fast Wireless Charging Station Dock For iPhone 13/13 Pro Max/iPhone 12/12 Pro Max/11 Pro Max/SE/XR/XS Max/8 Plus/Apple Watch
€50.00



Mouses
Doorway Pull Up Bar, Chin Up Bar Upper Body Workout Bar For Home Gym Exercise Equipment Strength Training
€50.00

Featured Products



Keyboards
The Quick Brown Fox Jumps Over The Lazy Dog



Keyboards
Bathroom Wall And Cabinet Organiser In Satin



Mouses
Doorway Pull Up Bar, Chin Up Bar Upper Body Workout Bar For Home Gym Exercise Equipment Strength Training

Signin/Signup page

The sign in and sign up page are the pages where users can create their account or login in to their own account to see orders already made, change their details, Wishlist and the current cart.

KEYBOARDS MOUSES MEMORY

Sign in to your account

Create an Account

Creating an account is easy! Enter your details and a password to enjoy all the benefits of having an account.

About us About us Directions	Featured Categories Keyboards Mouses Memory	Help Payment Methods Contact us	Newsletter Receive exclusive offers <input type="text" value="Email"/> <input type="button" value="Sign up"/>
--	--	---	---

KEYBOARDS MOUSES MEMORY

Create your account

My Account page

When the user is logged in the application, it is possible to change their details and see the past orders. This can be done through the My Account page. The page shows two tabs: My details, to change the user details and My Orders, to see the past orders.

The screenshot displays the eCamell website's 'My Account' page. At the top, the eCamell logo is on the left, and navigation links for 'My Account', 'WishList', 'Dark', 'Cart', and 'Signout' are on the right. A search bar is positioned below the navigation. A horizontal menu lists 'KEYBOARDS', 'MOUSES', and 'MEMORY'. The 'MY ACCOUNT' section features two tabs: 'My Details' (active) and 'My Orders'. The 'Update your account' form includes fields for First Name (Customer), Last Name (Gustavo), Email (gucamello.dev@gmail.com), and Complete Address (My new address to delivery 4564). A green 'Update Values' button is at the bottom of the form.

eCamell

My Account WishList Dark Cart Signout

Search for an item

KEYBOARDS MOUSES MEMORY

MY ACCOUNT

My Details My Orders

Update your account

First Name
Customer

Last Name
Gustavo

Email
gucamello.dev@gmail.com

Complete Address
My new address to delivery 4564

Update Values

Search for an item

KEYBOARDS MOUSES MEMORY

MY ACCOUNT

My Details

My Orders

Order Number	Payment Method	Total	Status
# 6266bee86d3c3edae302eed	Credit Card	€100.00	Sent See Order

Order Number: # 6266bee86d3c3edae302eed



The Quick brown fox Jumps Over The Lazy Dog
10 x €10.00 - Total: €100.00

Delivery Expected by:
12/05/2022

Delivery Address: My new address to delivery 4564	Payment method Credit Card	Total Price: €100.00
---	--------------------------------------	--------------------------------

# 6266bf186d3c3edae302eef5	Credit Card	€100.00	Sent See Order
# 6266c0706d3c3edae302eefb	Credit Card	€100.00	Sent See Order

Single Product Page

This screen shows a single product. There is the name of the product, an image, buttons where the user can share the product on their social medias, a description of the products, the price and buttons to add the product to the cart and to the Wishlist. Basically, all the information about the product itself. Also, there is the reviews section of the product.

The screenshot shows the eCamell website's single product page for a mouse. The page layout includes a header with the eCamell logo, navigation links for Signin, WishList, Dark, and Cart, and a search bar. Below the header, there are category links for KEYBOARDS, MOUSES, and MEMORY. The main content area features a large image of a black and red gaming mouse. To the right of the image, the product title is "Bathroom Wall and Cabinet Organiser in Satin" (though the image is a mouse), with a 5-star rating and 7 reviews. The price is listed as €20.00. Below the price, there are social sharing buttons for Facebook, WhatsApp, Twitter, and Telegram, along with an "Add To Cart" button and a wishlist icon. At the bottom of the page, there is a "Product Reviews" section with three reviews from "Customer Gustavo": a 5-star review titled "My First Review", a 4-star review titled "Second Review", and another 5-star review.

eCamell

Signin WishList Dark Cart

Search for an item

KEYBOARDS MOUSES MEMORY

KEYBOARDS

Keyboards

Bathroom Wall and Cabinet Organiser in Satin

★★★★★ 7 Reviews

lém de uma construção interna incrível, O P852 também te garante a melhor experiência durante todo o tempo necessário. A almofada é feita em malha esportiva, que evita aquecimento em dias mais quentes, além de impedir o aumento de temperatura após horas de uso.

Share Share Tweet Share

€20.00 Add To Cart

Product Reviews

Customer Gustavo
★★★★★
My First Review

Customer Gustavo
★★★★☆
Second Review

Customer Gustavo
★★★★★

Cart Page

This is the Cart page, maybe the most important page of an eCommerce. User can see what is in their cart, change quantities or remove items from the cart and also they can enter the card details to place an order.







The screenshot shows the 'eCamell' cart page. At the top, there is a navigation bar with the brand name 'eCamell', a search bar, and links for 'My Account', 'WishList', 'Dark', 'Cart', and 'Signout'. Below the navigation bar, there are category links: 'KEYBOARDS', 'MOUSES', and 'MEMORY'. The main content area is titled 'CART' and contains three items in the cart, each with a product image, name, price, and quantity controls. To the right of the items is a 'Summary' section showing the total price and a form to enter card details for placing an order.

eCamell My Account WishList Dark **Cart** Signout

Search for an item

KEYBOARDS MOUSES MEMORY

CART

	Bathroom Wall and Cabinet Organiser in Satin 1 x €20.00 - Total: €20.00	- 1 + 
	in 1 Fast Wireless Charging Station Dock for iPhone 13/13 Pro Max/iPhone 12/12 Pro Max/11 Pro Max/SE/XR/XS Max/8 Plus/Apple Watch 1 x €50.00 - Total: €50.00	- 1 + 
	Doorway Pull Up Bar, Chin Up Bar Upper Body Workout Bar for Home Gym Exercise Equipment Strength Training 1 x €50.00 - Total: €50.00	- 1 + 

Summary

Total €120.00

Enter your card details to place an order

Card Number
1234 1234 1234 1234

Exp Date CVC
MM / AA CVC

Place Order

About us Featured Categories Help Newsletter
About us Keyboards Payment Methods Receive exclusive offers

Wishlist Page

The Wishlist page show all the products the user saved with the intention to buy. It is possible to see the products name and price, click on them to go to the page of the product or add the product directly to the cart.

eCamell My Account WishList Dark Cart Signout

Search for an item

KEYBOARDS MOUSES MEMORY

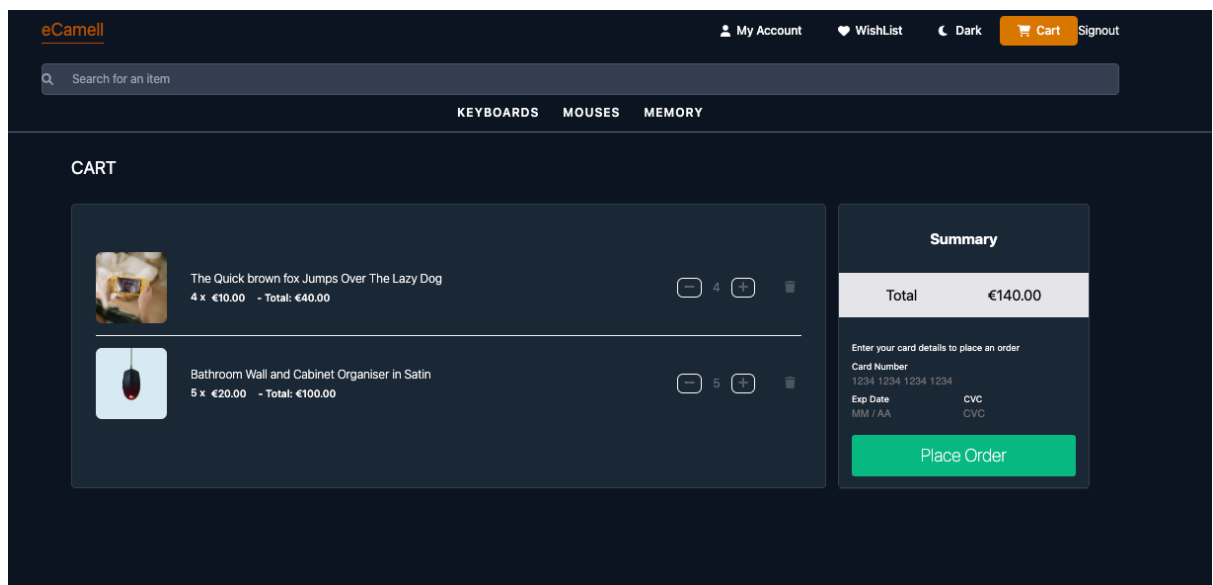
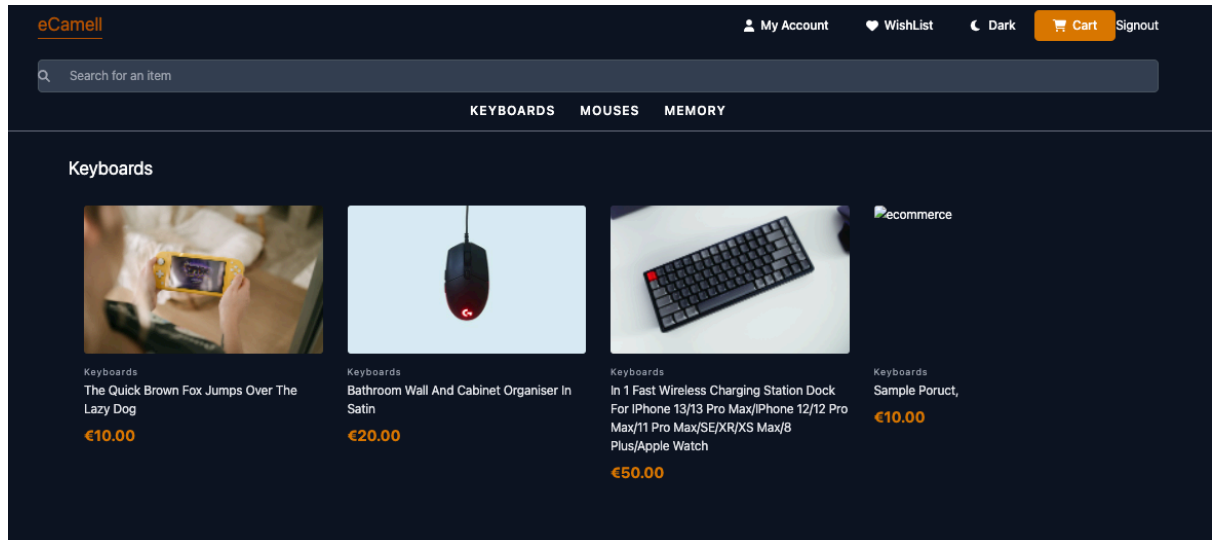
WISHLIST

	The Quick brown fox Jumps Over The Lazy Dog €10.00	Add To Cart
	Bathroom Wall and Cabinet Organiser in Satin €20.00	Add To Cart
	in 1 Fast Wireless Charging Station Dock for iPhone 13/13 Pro Max/iPhone 12/12 Pro Max/11 Pro Max/SE/XR/XS Max/8 Plus/Apple Watch €50.00	Add To Cart

About us About us Featured Categories Keyboards Help Payment Methods Newsletter Receive exclusive offers

Dark Mode

The dark mode is available in all the pages of the application. When the user click on the Dark button in the navigation menu, all the screens change to dark mode. Some examples are the products page and cart.



Mobile Screens

The eCommerce is fully responsive, with all the pages also working well on small screens.


Cart page in small screens.

eCamell 👤 🌙 🛒 \$

🔍 Search for an item

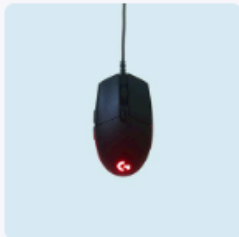
KEYBOARDS MOUSES MEMORY

CART



The Quick brown fox Jumps Over The Lazy Dog
4 x €10.00 - Total: €40.00

− 4 + 🗑️



Bathroom Wall and Cabinet Organiser in Satin
5 x €20.00 - Total: €100.00

− 5 + 🗑️

Summary

Total	€140.00
-------	---------


Single Product page in small screens.

eCamell 👤 🌙 🛒 Sign

🔍 Search for an item

KEYBOARDS MOUSES MEMORY

KEYBOARDS



Keyboards

The Quick brown fox Jumps Over The Lazy Dog

★★★★★ 7 Reviews

Desenvolvido com materiais de alta qualidade e um design requintado, o P852 é o headset perfeito para qualquer ambiente. Com uma beleza protagonizada pela discrição e simplicidade, pode ser usado em todos os setups, dos gamers aos profissionais.

[f Share](#) [📧 Share](#) [🐦 Tweet](#) [💬 Share](#) [📌](#)

€10.00 [Add To Cart](#) [❤️](#)

Product Reviews

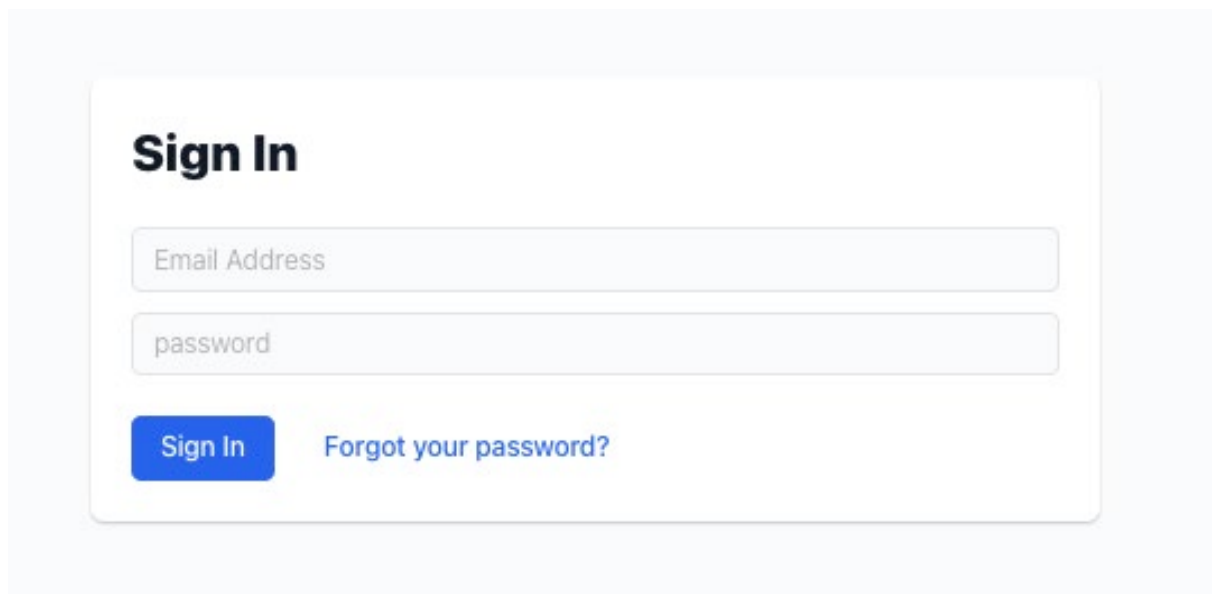
[Leave a Review](#)

Customer Gustavo

Admin Panel

Sign in page

The first step when someone visit the admin panel, is to sign in, as only admin users can access this page. It is a simple login with email and password.



Sign In

Email Address

password

[Sign In](#) [Forgot your password?](#)

Admin Panel Dashboard Page

After the admin user sign in, it will redirect to the main dashboard page. In this page, the user can manage all the eCommerce store, creating new products, new categories, see all orders placed, reviews. It is possible to manage any information from the eCommerce like see users reviews, users Wishlist, orders and also manage some basic styling to the store front. For each card, all the CRUD operations are possible, to simplify this report, I will show the examples with the products module.

KeystoneJS **Dashboard**

Signed in as Admin ...

- Dashboard
- Users
- Products
- Product Images
- Reviews
- Product Categories
- Wish Lists
- Wish List Items
- Carts
- Cart Items
- Order Items
- Orders
- Customers

Users 1 item	Products 5 items	Product Images 6 items
Reviews 14 items	Product Categories 3 items	Wish Lists 7 items
Wish List Items 10 items	Carts 7 items	Cart Items 14 items
Order Items 47 items	Orders 38 items	Customers 29 items
Storefronts 1 item		

Products Page

The products screen display a list with all the products in the eCommerce. The user can filter and sort the list the easily manage all the products.

Products

[Create Product](#) [Filter List](#) ▼

Showing **5 Products**, sorted by **No field** ▼ with **4 columns** ▼

<input type="checkbox"/> Name	Category	Photo	Price
<input type="checkbox"/> The Quick brown fox Jumps Over The Lazy Dog	Keyboards, Mouses	6210efdb223aa53b08223247	10
<input type="checkbox"/> Bathroom Wall and Cabinet Organiser in Satin	Keyboards	6210efe8223aa53b08223263	20
<input type="checkbox"/> in 1 Fast Wireless Charging Station Dock for iPhone 13/13 Pro Max/iPhone 12/12 Pro Max/11 Pro Max/SE/XR/XS Max/8 Plus/Apple Watch	Keyboards	6210eff4223aa53b08223280	50
<input type="checkbox"/> Doorway Pull Up Bar, Chin Up Bar Upper Body Workout Bar for Home Gym Exercise Equipment Strength Training	Mouses	6210f002223aa53b0822329e	50
<input type="checkbox"/> sample Poruct,	Keyboards		10

Create new product page

To add new products, the user can click on the Create Product button, and a modal will show up to the creation of the product.

The image shows a 'Create Product' modal form. On the left, a 'Products' list is visible, showing a table with columns for Name, Description, Reviews, Category, Price, and Photo. The modal form on the right has the following fields:

- Name:** A text input field.
- Description:** A large text area.
- Reviews:** A dropdown menu with 'Select...' and a 'Create related Review' button.
- Featured:** A checkbox.
- Category:** A dropdown menu with 'Select...' and a 'Create related Product Category' button.
- Price:** A text input field.
- Photo:** A dropdown menu with 'Select...' and a 'Create related Product Image' button.

At the bottom of the modal are two buttons: 'Create Product' (in blue) and 'Cancel'.

Edit Product Page

All the information about the product can be updated through the edit product page. The price can be updated, bad reviews can be removed, change the image or delete the product.

Products > Bathroom Wall and Cabinet Organiser in Satin

Name

Bathroom Wall and Cabinet Organiser in Satin

Item ID

6210ef9ada4304d6b4d29704



Description

lém de uma construção interna incrível, O P852 também te garante a melhor experiência durante todo o tempo necessário. A almofada é feita em malha esportiva, que evita aquecimento em dias mais quentes, além de impedir o aumento de temperatura após horas de uso.

Reviews

626e92da6aa860658338c9df x 626e92ed6aa860658338c9f7 x
626e93106aa860658338ca0f x 626e93e06aa860658338ca27 x
626e9da66aa860658338ca78 x 6270e13d6aa860658338cd31 x
6273f32ccbab2b10e0737666 x

Create related Review

View related Reviews

Featured

Category

Keyboards x

Create related Product Category

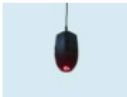
View related Product Categories

Price

20

PHOTO

Source



Alt text

image

Edit

Remove

Create Product Image

Save changes

No changes


Delete


Customize Storefront Page

From the Storefront page, the admin can customize the storefront. For now, two options are available: Change the store name and change the main promotional banner in the home page. More options will be added in the future, for example: change colours of the store or decided payment methods.

Storefronts > **62741a401ba83b4c5a0e198f**

Store name

Item ID
 

Source


Alt text

2.4. Testing

Due to time and resources limitation, the testing was made by myself using the Manual Testing methodology and some unit tests has been written using the Jest library.

Manual Tests

The manual tests were conducted based on the requirements established in the section 2 with the present document.

Admin Portal Tests

Requirement Number	Action	Description	Result	Passed/Failed
1	Log in	Ensure the admin user can login with an email and password	As expected	Passed
2	Manage Products	Ensure the user can manage a product. Create, Update, See and Delete a product.	As expected	Passed
3	Manage Categories	Ensure the user can manage a category. Create, Update, See, Delete a category and associated products to categories.	As expected	Passed
4	See Orders	Ensure the admin user can see all the orders placed in the eCommerce.	As expected	Passed
5	Customize Storefront	Ensure the admin user can customize the storefront through the admin panel.	As expected	Passed

Storefront Tests

Requirement Number	Action	Description	Result	Passed/Failed
6	View Products	Ensure the user can see products pages.	As expected	Passed
7	Create an Account	Ensure the user can create an account with name, password, email and address.	As expected	Passed
8	Log In	Ensure the user can login in the eCommerce.	As expected	Passed
9	See Orders	Ensure the user can see past orders.	As expected	Passed
10	Update Details	Ensure the user can update their details like first name, last name, email and address.	As expected	Passed
11	Add Items to Cart	Ensure the user can add items to their cart.	As expected	Passed
12	Manage Cart	Ensure the user can manage the cart items. Update quantity or remove items from the shopping cart.	As expected	Passed
13	Place an Order	Ensure users can place an order using their credit card details.	As expected	Passed

Unit Tests

The unit tests have been written using the Jest library with React Testing Library. Jest is a simple JavaScript Testing Framework used to testing JS applications.

Product Component

```
import { render, screen } from "@testing-library/react";
import { MockedProvider } from "@apollo/react-testing";
import "@testing-library/jest-dom";

import Product from "../src/components/products/Product";
import { fakeItem } from "../src/lib/testUtils";

const product = fakeItem();

describe("<Product/>", () => {
  it("renders out the price tag and title", () => {
    const { container, debug } = render(
      <MockedProvider>
      <Product product={product} />
    </MockedProvider>
    );

    expect(screen.getByText("5000")).toBeInTheDocument();

    const link = container.querySelector("a");
    debug(link);
    expect(container).toHaveTextContent(product?.name);
  });

  it("renders and match the snapshot", () => {
    const { container, debug } = render(
      <MockedProvider>
      <Product product={product} />
    </MockedProvider>
    );

    expect(container).toMatchSnapshot();
  });

  it("renders the image properly", () => {
    const { container, debug } = render(
      <MockedProvider>
      <Product product={product} />
    </MockedProvider>
    );

    const img = screen.getByAltText("ecommerce");
    expect(img).toBeInTheDocument();
  });
});
```

Result: The Product component works as expected and passed all the tests.

```
Test Suites: 2 passed, 2 total
Tests:       5 passed, 5 total
Snapshots:  1 obsolete, 1 passed, 1 total
Time:       3.872 s
```


Format Money function

```
import { formatMoney } from "../src/components/DisplayMoney";

describe("display money component", () => {
  it("works with whole euros", () => {
    expect(formatMoney(1)).toEqual("€1.00");
    expect(formatMoney(1)).toEqual("€1.00");
    expect(formatMoney(140)).toEqual("€140.00");
  });

  it("works with fractional euros", () => {
    expect(formatMoney(140.21)).toEqual("€140.21");
    expect(formatMoney(525.25)).toEqual("€525.25");
  });
});
```

Result: The display money function is an important function of the whole system, as it is responsible to calculate all the money displayed in the storefront. The function works as expected and passed all the tests.

```
Test Suites: 2 passed, 2 total
Tests:       3 passed, 3 total
Snapshots:   2 obsolete, 0 total
Time:        4.378 s
Run all test suites related to changed files
```

Single Product Component

```
describe("<SingleProduct/>", () => {
  it("renders with proper data", async () => {
    const { container, debug } = render(
      <MockedProvider mocks={mocks}>
        <SingleProduct id="123" />
      </MockedProvider>
    );

    await screen.findByTestId("singleProduct");
    debug();
  });

  it("Errors out when an item is not found", async () => {
    const errorMock = [
      {
        request: { query: SINGLE_ITEM_QUERY },
        variables: {
          id: "123",
        },
        result: {
          errors: [{ message: "Item not found!" }],
        },
      },
    ];

    const { container, debug } = render(
      <MockedProvider mocks={errorMock}>
        <SingleProduct id="123" />
      </MockedProvider>
    );

    await screen.findByTestId("graphql-error");
    debug();
  });
});
```

Result: The component works as expected and passed all the tests.

```
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  1 obsolete, 1 passed, 1 total
Time:        3.7 s
```

2.5. Evaluation

The evaluation of the website was made using the application for some days and some of the points observed can be seen below:

- Overall the application is very clean and well structured. The main navigation of the website is easily found and is always present in all the screens, making possible for the user to access each main functionality of the website from any page.
- The application is easy and simple to use and even non tech skilled people can use and place an order without any guidance.
- The use of GraphQL with Next.js make the application really fast to navigate and during the evaluation process, it did not show any delay or breaking pages.
- The admin panel to manage the application works really well, making possible to manage each part of the system easily and in a well presented UI.

After using the application for some days, we can consider the application as a moderate application. It is simple, with everything an eCommerce needs to offer, but it also offers some cool features like dark mode, share products on social media and from a business perspective, it is very simple to manage the application through the admin panel.

3.0 Conclusions

To develop this application was a quite challenge for me as I did not have experience developing systems with a microservice architecture, so I could learn a lot about it. Overall, I am really happy with the final application as all that I have planned is done in the best way possible.

The advantages of the eCamell offers are that the application is really easy to use, for the end user and to the business to maintain the application. The screens are clear and well structured, without overwhelming the user with unnecessary information. The application works well on mobile screens, what is a must nowadays and offers a easy checkout process where users can place an order in a few clicks.

The disadvantages and limitations of the project are that it could offer more control to the businesses: for example, stock control, a way to promote an item for a specific period and also the system should send emails when an order is placed.

Developing this application was very enjoyable since I could use a lot of what I learned during the 4 years in the National College of Ireland and to be able to successfully develop a complete eCommerce platform makes me impressive with what I can do now!

4.0 Further Development or Research

The application can be considered as a version 1 and I have plans to continue to develop and improve it. With additional time and resources, the project can be improved a lot with new features for the end user and also to the business. Some features that can be added to the application are:

- A last visited products section.
- Recommend products to the user based on their navigation in the application.
- Send an confirmation email after the user place an order.
- Offer more payment methods.
- A newsletter system.
- A promotion system where admin users can manage promotional items easily.
- From a business perspective, offer more customization to the store, like creating bundles of products, add more photos in the product pages, stock control and a possibility to change colours of the storefront.

Those are some of the possible new features to add in the application, but a lot more can be done with more time and resources.

5.0 References

purchasecommerce.com. 2022. *10 Essential Ecommerce Website Requirements & Best Features*. [online] Available at: <<https://www.purchasecommerce.com/blog/10-essential-ecommerce-requirement-best-in-class-feature>> [Accessed 10 November 2021].

The 360 Blog from Salesforce. 2022. *Everything You Need To Know About Headless Commerce*. [online] Available at: <<https://www.salesforce.com/blog/define-headless-commerce/>> [Accessed 10 November 2021].

En.wikipedia.org. 2022. *Headless commerce - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Headless_commerce> [Accessed 10 November 2021].

BigCommerce. 2022. *Headless Commerce: The Evolution of Selling Online (2022)*. [online] Available at: <<https://www.bigcommerce.com/articles/headless-commerce/>> [Accessed 10 November 2021].

Docs.microsoft.com. 2022. *Microservice architecture style - Azure Architecture Center*. [online] Available at: <<https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>> [Accessed 16 November 2021].

Digitalocean.com. 2022. *Monolithic vs microservice architecture: Which is best?* / *DigitalOcean*. [online] Available at: <<https://www.digitalocean.com/blog/monolithic-vs-microservice-architecture>> [Accessed 16 November 2021].

GeeksforGeeks. 2022. *Monolithic vs Microservices architecture - GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/monolithic-vs-microservices-architecture/>> [Accessed 10 November 2021].

Tran, C., 2022. *15 Key eCommerce Website Requirements to Sell Online in 2022*. [online] SimiCart Blog | eCommerce Insights & Magento Tutorials. Available at: <<https://www.simicart.com/blog/ecommerce-website-requirements/>> [Accessed 16 November 2021].

Google Cloud. 2022. *What Is Microservices Architecture?* / *Google Cloud*. [online] Available at: <[https://cloud.google.com/learn/what-is-microservices-architecture#:~:text=Microservices%20architecture%20\(often%20shortened%20to,its%20own%20realm%20of%20responsibility.](https://cloud.google.com/learn/what-is-microservices-architecture#:~:text=Microservices%20architecture%20(often%20shortened%20to,its%20own%20realm%20of%20responsibility.)> [Accessed 16 November 2021]

Apollo GraphQL. 2022. *Apollo GraphQL/ Supergraph: unify APIs, microservices, & databases in a composable graph*. [online] Available at: <<https://www.apollographql.com/>> [Accessed 10 Feb 2022].

GraphQL.org. 2022. *GraphQL/ A query language for your API*. [online] Available at: <<https://graphql.org/>> [Accessed 10 Feb 2022].

KeystoneJS: The superpowered Node.js Headless CMS for developers - Keystone 6. 2022. *KeystoneJS: The superpowered Node.js Headless CMS for developers - Keystone 6*. [online] Available at: <<https://keystonejs.com/>> [Accessed 10 Feb 2022].

Typescriptlang.org. 2022. [online] Available at: <<https://www.typescriptlang.org/>> [Accessed 10 Feb 2022].


MongoDB. 2022. *MongoDB Atlas Database / Multi-Cloud Database Service*. [online] Available at: <<https://www.mongodb.com/atlas/database>> [Accessed 10 Feb 2022].

Nextjs.org. 2022. *Next.js by Vercel - The React Framework*. [online] Available at: <<https://nextjs.org/>> [Accessed 10 Feb 2022].

Node.js. 2022. *Node.js*. [online] Available at: <<https://nodejs.org/en/>> [Accessed 10 Feb 2022].

Reactjs.org. 2022. *React – A JavaScript library for building user interfaces*. [online] Available at: <<https://reactjs.org/>> [Accessed 10 Feb 2022].

Tailwindcss.com. 2022. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. [online] Available at: <<https://tailwindcss.com/>> [Accessed 10 Feb 2022].

Jestjs.io. 2022. *Jest ·  Delightful JavaScript Testing*. [online] Available at: <<https://jestjs.io/>> [Accessed 10 May 2022].

Testing-library.com. 2022. *React Testing Library / Testing Library*. [online] Available at: <<https://testing-library.com/docs/react-testing-library/intro/>> [Accessed 10 May 2022].

6.0 Appendices

6.1. Project Proposal



National College of Ireland

Project Proposal

eCamell – eCommerce for small business

07th November, 2021

BSc (Honours) in Computing

Software Development

Academic Year i.e. 2021/2022

Gustavo Camello

X18119956

X18119956@student.ncirl.ie

Contents

1.0	Objectives	49
2.0	Background	49
3.0	State of the Art.....	50
4.0	Technical Approach.....	50
5.0	Technical Details	51
6.0	Special Resources Required	52
7.0	Project Plan	52
8.0	Testing.....	54

1.0 Objectives

The main objective of the project is to create a fully responsible eCommerce platform with focus on small and medium size business to help them to have a better presence online. The business are the admin of the application, they can create products, upload images, check orders, create promotions, etc and the final user will navigate through an easy and beautiful UI that exceed clients expectations, search for products, buy products, left reviews, etc.

The application will be created as a “headless commerce” where we have the back-end and front-end of the application decoupled. A headless commerce, different from traditional applications, can offer more flexibility, personalization, efficiency to the application. Also the back-end of the application will be constructed using microservices architecture, each main functionality of the application will have its own server.

The real challenge of the project is to make everything work successfully together, as we will have different applications to each concern of the system, and integrate all the services will be a challenge.

Some of the features that the project aim to have:

- Fully responsive: the application should work on desktop and mobile screens.
- A beautiful UI: the application should be built in a manner that is attractive to users to see and use.
- Payments: users should be able to pay using credit cards.
- Admin management: the application should allow the admin to perform admin tasks like create new product, edit a product, check orders, etc.
- Share Products: the user should be able to share a product on social media
- Dark Mode: offers a dark mode to the users
- Microservices architecture: microservices powering up the client side

2.0 Background

In today's society, consumers are more likely to shop online rather than physically visit a store. Some members of the country's leadership advise citizens not to venture outside and to keep their critical activities online while at home. As a result, internet buying is becoming more appealing, and the online market is becoming larger than ever.

To help small business to achieve more sales, I am developing the eCamell, a simple and easy to use and maintain eCommerce platform, where business can sell their products online and reach a lot more people.

I choose to make this project because I am really interested in eCommerce platforms, it is something that really gets my attention and I want to learn more about how the functionalities of one works. Also I believed that an fully eCommerce is a good showcase project to have.

The technologies and the architecture I decided to use in the project, most of them, are new technologies or emergent technologies with high demand on the market. Also due to these technologies and architecture will be a good challenge for me to develop and learn a lot with this project, delivering a great application both for business and for the final client.

To meet the application objectives I will follow my project plan to meet the deadlines, make a good use of the official resources (languages, libraries...) documentation and research on articles, videos, blogs the best way to meet the requirements.

3.0 State of the Art

An eCommerce is not something really innovative as there are thousands of eCommerce's out there, just to mention some there are amazon.ie, etsy.com, Walmart.com, shopcarnaval.com, etc.

My project will be different from the others because I will create an application with an architecture different from the "traditional way" to do it. Most of the eCommerce's on the web are created with the front-end and back-end of the application on the same environment (for example using Magento, Wordpress, Ruby on Rails...), the front-end and back-end are coupled on the same application. The eCamell will use the "headless commerce" concept, where we have a separation of concerns between front-end and back-end.

A headless commerce can offer some benefits that a traditional application can't.

- More flexibility: we can work each part of the system separately
- Efficiency: separation of these concerns makes the application much faster
- Personalization: the UI is no longer tied to a standard out-of-the-box solution interface
- One API: with the decoupling of the parts, we can use anything to show to the user, a website or an Android or iOS app.

Also, it is important to mention that the backend of the application, will be constructed as microservices. Each main functionality of the application will have its own service. Microservices offer us:

- Independence: each module of the system is separated, meaning if a part of the system fails the other parts will continue to work
- Scalability: since the services are separated, we can improve/scale our application much faster and with a low cost
- More flexibility: we can use different technologies for different parts of our UI, without having to be worried with the rest
- Less dependency and easy to test

4.0 Technical Approach

The approach to develop the project will be using Agile methodology. A Scrum board with Sprints (duration 2 weeks) will help me to develop each part of the application and keep on track to deliver the final version at the deadline.

To identify and gather the requirements, some techniques that I intend to use:

- **Brainstorming** with some people from the field, to help me understand what can be made different from the others
- **Interviews** a small amount of people to get a more specific feedback

The requirements already gathered and the basics ones that any eCommerce must have:

- Simple Navigation: it should be easy to the user to find what they are looking for
- Admin Panel: the admin (business) can manage the application
- Easy and Simple Checkout Process: if the process to checkout is too complicated, users will give up
- Mobile-Friendly: nowadays mobile commerce sales comprised 50% of eCommerce sales
- Social Media Share: users love to share on Social Media platforms
- Handle payments: an eCommerce must have a way for users to pay
- Easy to interact and Friendly Interface
- Dark Mode: more and more people are joining the Dark mode on their screens

To break down the requirements into tasks and milestones, I will use the Trello boards to help me to organize the work. Each requirement will be broken into small tasks to be done during the Sprints and the milestones will be considered each main functionality/requirement of the project: Shopping Cart, Stripe Integration, etc.

For example Handle Payments will be broken on:

- Integrate Stripe API
- Develop the frontend page for payment
- Connect the application backend to Stripe API
- Integrate the application backend with frontend

5.0 Technical Details

The project will be developed using JavaScript as the main language, because it is the programming language that I'm more comfortable in use.

The approach decided for the application is to create a headless commerce platform, where we have the back-end decoupled from the front-end (already mentioned on this document).

The back-end will be created using the microservices architecture with each main module of the application will have its own service.

The front-end (client) will be developed using Next.js, a React framework to create UI components.

The below list shows the languages, libraries, database, etc that the project intends to use:

Backend:

- Node.js: the headless CMS that will be used in the project is built in Node.
- Keystone.js: Keystone is a headless CMS built in Node.js. I decided to use it, because I have an interesting in learn how CMS works and also to speed up the creation of the microservices.
- GraphQL: Keystone offers a GraphQL API or REST API. The project will use the GraphQL API as it is faster and more efficient than REST.

Frontend:

- Next.js: Next.js is a React framework for server side rendering, routing and tooling.
- React.js: the UI of the application will be built in React.js. A JavaScript library.
- Apollo Client: for data management. Connect the frontend to the GraphQL API.
- Tailwind: CSS framework to style the application

Others:

- PostgreSQL: a relational database.
- Jest.js: JavaScript Test Framework, to help test the application.
- Stripe: use of Stripe platform to allow payments.

6.0 Special Resources Required

- Payment gateway (Stripe)
- A domain name to deploy the application
- Email host to handle emails
- AWS to deploy the application

7.0 Project Plan

The project plan will be based on a Agile methodology, dividing the plan in Sprints with a 2 weeks duration.

Sprint 1 – 15/11/2021 to 28/11/2021

- Research and Design: Define the site design, colours, fonts, etc
- Project setup: Install the libraries, packages, dependencies to start the project
- Site Layout:
 - Create Header and Nav components
 - Create Page Layout
- Define Database Schema

Sprint 2 – 29/11/2021 to 12/12/2021

- MongoDB: Set up the database
- Keystone.js: Setting up the Headless CMS
- Admin Dashboard: Configure Admin dashboard from Keystone
- Define User Schema
- Add Authentication
- Define Product Schema
- Upload Product Image: allow user to upload image functionality
- Set up Dark Mode

Sprint 3 – 13/12/2021 to 26/12/2021

- Add Apollo Client: configure Apollo to data management
- Fetch data: fetch data from backend to display on frontend
- Create Product functionality

- Products Page design
- Single Product Page design

Sprint 4 – 27/12/2021 to 09/01/2022

- Product Update Page design
 - Frontend Page Layout
 - Backend Process
- Delete Products functionality
 - Backend Process

Sprint 5 – 10/01/2022 to 23/01/2022

- Pagination: implement pagination functionality
- Sign in functionality
 - Frontend Page layout
 - Backend Process
- Sign out functionality
 - Frontend Page layout
 - Backend Process
- Password Reset functionality
 - Send Email to Reset Password

Sprint 6 – 24/01/2022 to 06/02/2022

- Shopping Cart
 - Add Items to Cart
 - Display Items in to Cart
 - Remove/Update Items from the Cart

Sprint 7 – 07/02/2022 to 20/02/2022

- Checkout Process
 - Integrate Stripe API
 - Create Checkout Frontend Pages Layout
 - Backend Process

Sprint 8 – 21/02/2021 to 06/03/2022

- Display Orders
 - Single Order
 - Display All Orders
- Authorization
 - Admin Permissions
 - Users Permissions

Sprint 9 – 07/03/2022 to 20/03/2022

- Product Share: share product on Social Media functionality
- Deployment

Sprint 10 – 21/03/2022 to 17/04/2022

- Application Tests
- Fix errors and work on improvements
- Finish Documentation

8.0 Testing

The application will be tested with different approaches to make sure it is delivered in the best way possible, with minimal errors or bad performance.

The use of an testing library (Jest.js) will help to write automated tests and test the main parts of the system like:

- Pagination
- Password reset
- Data Requests
- User Interaction

In the final stage, with the application deployed, some people will be invited to access the application and use it as a real customer to provide feedback and allow that any user case that may have been forgotten during the development phase to be worked and fixed.

6.2. Reflective Journals

October 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month:

What?

Reflect on what has happened in your project this month?

In this first month, I researched ideas about what I wanted to develop. I decided to develop an eCommerce to the Brazilian community in Ireland. Basically the application is an eCommerce to sell Brazilian products.

After decided what I want to build, I start research about the technologies to use in the project and I chose some new technologies to learn more about them and also have a more challenge project. I posted my Project Pitch and I'm waiting for my supervisor to get back to me to start the Project Proposal.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

This is the first month only, and what I have for it was to make my Project Pitch and decide what and how I will build the application. I consider this as a success because I could finish the Project Pitch on time and also decided what I want to build and how.

As I didn't start to develop the application yet, I can't say that I have any big challenge or that has anything remaining.

Now What?

What can you do to address outstanding challenges?

For the next month, I will delivery my Project Proposal and then start to develop the application. I believe it will be really challenge as I will use technologies that I'm not comfortable to use but I already got some good resources on the internet to help me to overcome any challenge that I may face.

I'm also preparing a Project Planning to help me to keep on track to delivery everything I intend to develop for the next month.

Student Signature

Gustavo Camello

November 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month: NOVEMBER

<p>What? Reflect on what has happened in your project this month? In this month, after discussing the project with my supervisor, I decided to change the Project a bit to have better marks on Innovation and Difficult. The project is still an eCommerce, but instead of a eCommerce to sell Brazilian products it will be a eCommerce platform, that small/medium business can buy and have a presence online. The technologies will be the same but the architecture of the software will be a headless commerce with microservices on the backend. I improve I tweaked the Project Proposal and sent the new one to the supervisor.</p>	
<p>So What? Consider what that meant for your project progress. What were your successes? What challenges still remain? As I needed to change the project, it takes me something to develop the idea and present the new project proposal. I can consider a success, the new project proposal decided after the meeting with the supervisor and with the points he mention, I could improve the project. As I didn't start to develop the application yet, I can't say that I have any big challenge or that has anything remaining.</p>	
<p>Now What? What can you do to address outstanding challenges? In December I will start to develop the demo of the application to show on the mid-presentation and also start work in gathering the requirements for the project, documenting it with the presentation slides and video. It will be a challenging develop the application, as I decided to go with an architecture different from the "traditional" eCommerce applications, but I already research a lot on the internet, how to develop the project and I'm confident that I can deliver a good final application.</p>	
Student Signature	Gustavo Camello

December 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month: DECEMBER

What?

Reflect on what has happened in your project this month?

In this month I started work in the demo of the application. I could make the initial setup work, and have the demo application ready to show at the deadline.

I got the requirements to the application and completed the section in the Document to be uploaded together with the slides and the presentation. Also, I recorded my mid-presentation with the slides of it.

I uploaded the Technical Report, the slides and the video recording of my presentation on Moodle.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

Following my project plan, I'm up to date with it, I could make everything that was meant to be done until the end of this month. The only planned task was to implement the authentication of the application, but as this is a main part of the system, I decided to postpone a little and focus in others functionalities for now.

A lot remain to be done, as I planned to focus in this project next year, but just to mention some challenging's that are still to be done are: start creating the microservices of the application, integrate and make the services communicate among them, deploy the application.

Now What?

What can you do to address outstanding challenges?

I have now a clear path to follow with my application, and also the initial set up with all the libs, technologies I will need already in the project.

To be able to deliver the application in time, I will continue to follow my project plan and research on the web, asking for help when I needed to deliver a complete application as my final year project.

Student Signature	Gustavo Camello
--------------------------	-----------------

January 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	Computing – Software Dev

Month:

What?

Reflect on what has happened in your project this month?

In this month I was focused on the TABAs we had and after it I decided to rest a bit before starting the new semester.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

As I mentioned in the first part, I did not work on the project this month, so the successes and challenges remain the ones from December.

Now What?

What can you do to address outstanding challenges?

For the following months I will focus on the project, to deliver the application on the deadline with the best way possible.

Student Signature	
--------------------------	--

February 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month: FEBRUARY**What?**

Reflect on what has happened in your project this month?

In this month I improved the front end of the application. Improving the codebase in general to be easier for the next steps in the project. Also, I did a good research on how to implement the microservices architecture using the stack I choose for the project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

I considered as a success this month as I improve and almost finish all the front end of the application. All expected pages and components were done. The challenges still remaining are to develop the services (APIs) to connect the front of the application to. And the deployment process.

Now What?

What can you do to address outstanding challenges?

Now the things are starting to get ready, and with all I already done, I'm ready to outstanding the remaining challenges in the project.

To be able to deliver the application in time, I will continue to follow my project plan and research on the web, asking for help when I needed to deliver a complete application as my final year project.

Student Signature

Gustavo Camello

March 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month: MARCH

What?

Reflect on what has happened in your project this month?

This month I was not able to work on the project due other assignments from other lectures and I was focused on those assignments.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

The challenges still remaining are to develop the services (APIs) to connect the front of the application to. And the deployment process.

Now What?

What can you do to address outstanding challenges?

To be able to deliver the application in time, I will continue to follow my project plan and research on the web, asking for help when I needed to deliver a complete application as my final year project.

Student Signature

Gustavo Camello

April 2021

Supervision & Reflection Template

Student Name	Gustavo Camello
Student Number	X18119956
Course	BSc (Honours) in Computing – Evening (Software Development Stream)

Month: April

What?

Reflect on what has happened in your project this month?

On this month I was totally focused in the project and I could make a lot of what was planned. All the main features of the application are done now.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

As mention, all the main functionality of the application are done and now I'm working on improve the overall code of the application and add more functionalities.

Now What?

What can you do to address outstanding challenges?

As I'm arriving at the end of the project, not to many challenges remain, but to deliver a complete application and with everything work and documentation, video, poster... I need to remain focus until the deadline.

Student Signature

Gustavo Camello

6.3. GitHub Repositories

Cart API

<https://github.com/gustavo-camello/eCamell-cartApi>

Customer API

<https://github.com/gustavo-camello/eCamell-customersApi>

Products API

<https://github.com/gustavo-camello/eCamell-productsApi>

StoreFront

<https://github.com/gustavo-camello/eCamell-storeFront>