

Abstract

This report details the logic and process of designing, developing, testing and deploying the Gamers' Den e-commerce website selling gaming products such as laptops, desktops, mice, keyboards and monitors. It also entails the reflections of the team members after completing the website and database systems. The thorough analysis of designing, development and testing of all features identified to achieve the identified goal, objectives and scope of the project are covered in this report. Moreover, the research and reasoning for each feature's functionality and design are covered as well to help readers better understand the process and steps taken to successfully develop and deploy the Gamers' Den e-commerce website. The report also covers the process flow between the website and database as well as the navigation flow of the website. These items will help the reader understand the structure and functions of the website and database.

Table of Contents

List of Figures	4
1.0 Introduction	4
1.1 Process Flow & Navigation Flow Diagrams	4
2.0 Problem Statement and Literature Review	4
3.0 Goal	5
4.0 Objectives	5
5.0 Scope	6
5.1 In-scope elements	6
5.2 Out-of-scope elements	6
6.0 Methodology	6
6.1 Software Functional Requirements	7
6.2 Software Non-Functional Requirements	8
6.3 Layout & Mockups	8
6.4 Database Design	15
7.0 Testing Results & Discussion	16
8.0 Conclusion & Future Works	19
References	20

List of Figures & Tables

1. Figure 1 - Process Flow.....	6
2. Figure 2 - Navigation Flow.....	7
3. Figure 3.1 - Navigation bar and footer mockup.....	12
4. Figure 3.2 - Home/landing page mockup.....	13
5. Figure 3.3 - Products page mockup.....	13
6. Figure 3.4 - Product Details page mockup.....	14
7. Figure 3.5 - Account page mockup.....	15
8. Figure 3.6 - Sign In page mockup.....	15
9. Figure 3.7 - Checkout page mockup.....	16
10. Table 3.8 - Error messages mockup.....	16-17
11. Figure 4 - Database RDM.....	18
12. Table 5.1 - Navigation bar test results.....	19
13. Table 5.2 - Footer test results.....	19-20
14. Table 5.3 - Image hyperlinks test results.....	20
15. Table 5.4 - Button test results.....	20-21

1.0 Introduction

Gamer's Den, hereby referred to as "the Company", is a brand new company looking to sell various electronic goods such as laptops, desktops, mice, keyboards and monitors to customers via an online platform. The time frame given to complete the e-commerce website is 4 months; from 28 April 2021 to 16 July 2021.

1.1 Process Flow & Navigation Flow Diagrams

The process flow is a flowchart detailing how the website and database work to achieve the goal and objectives of the project. The navigation flow is a visual representation of the functions of the website and which pages they bring the user to.

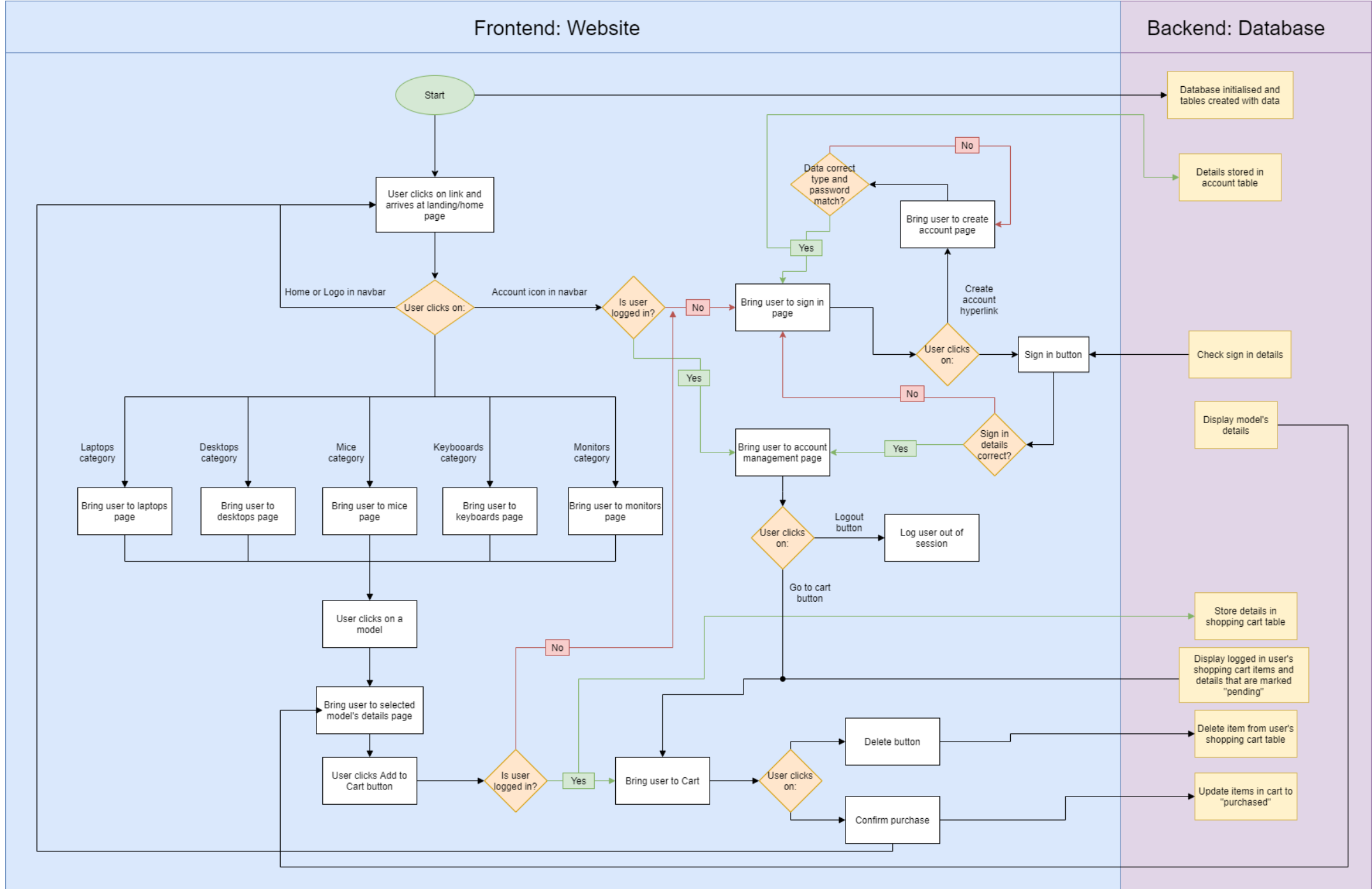


Figure 1 - Process flowchart for the website and database of the project.

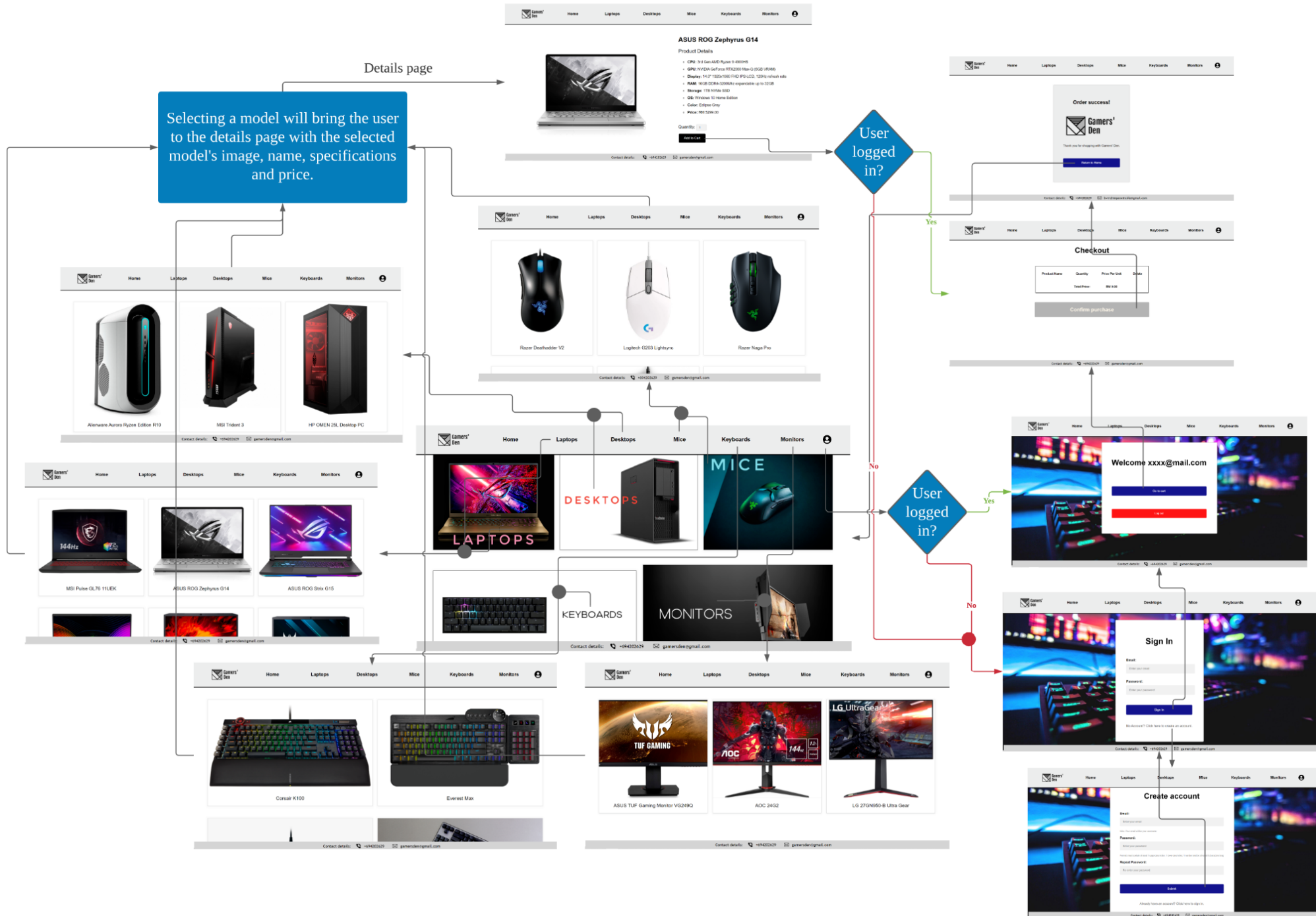


Figure 2 - Navigation flow for the website.

2.0 Problem Statement and Literature Review

As e-commerce picks up more and more traction, Gamer's Den plans to capitalize on this opportunity to target the gaming community, who presumably are more affluent with online shopping.

The initial consideration of having physical stores is omitted due to 2 main reasons. Firstly, the cost of running a physical store plays a major factor. The operation and maintenance cost of running a physical store is high and including, but not limited to, the cost of hiring multiple staff, rental fee of a desirable store location, transportation cost from warehouse to store, and the electricity bill [1]. Thus, a physical store has a very high sunk cost and can lead to bankruptcy of the Company if revenue does not cover costs in the long term.

Moreover, the opening of a physical store requires substantial marketing and advertising strategies in order to draw the attention of the target demographic. However, this traditional way of renting billboards and handing out brochures do not reach the typical gamer who are zoomers and millennials. The customers the Company hopes to attract are spending most of their time online and hence social media advertising is preferred since the cost is lower. Gamers' Den aims to begin advertising through popular online platforms such as Instagram, Facebook, and Twitter.

With these 2 concerns in mind, Gamers' Den decided to sell its products completely online with its own unique e-commerce website.

3.0 Goal

The goal of the project is to develop and successfully deploy a basic but fully functional e-commerce website to enable customers to make online purchases of the Company's goods. The project must be completed a week before the deadline, on 9 July 2021.

4.0 Objectives

Two main objectives for the frontend structure of the project are: a website that must be able to allow users to view all available products and relevant details (product name, price, image

and specifications) and allow users to successfully submit their order details to the Company. The main objective for the backend structure is to create a database that stores the following data for the website to use: user account email and password, user shopping cart details for users' orders and all products and their details. These objectives are identified to achieve the main function of an e-commerce website; to enable online selling and purchasing of goods.

Using the basics of HTML5, CSS, JS and PHP, the project must be achieved a week before 9 July 2021.

5.0 Scope

5.1 In-scope elements

The in-scope elements for this project is to develop a navigation bar and footer, home/landing page, products page, product details page, checkout page, user account feature and page, thank you page and website database. These items are necessary to achieve the project goal and objectives and hence, are given priority in this project.

5.2 Out-of-scope elements

These are items that were considered during the planning stages but were deemed unrelated to the project goal and objectives: FAQ section to help users quickly obtains with frequently asked questions, About Us page to display the information about the Company (not the website), a search bar to reduce the number of clicks required for users to reach their intended destination.

6.0 Methodology

The agile methodology was used to execute the project with 7 defined cycles for each of the scope elements. As the navigation bar and footer are easy to complete, it was combined with the home/landing page requirement in one cycle. Each cycle lasted one to two weeks depending on the difficulty of the feature. Using the agile methodology over the waterfall methodology allows the team to start designing, development, testing and deployment of each feature as soon as the feature's requirement was fully defined [2]. This reduces testing and bug fixing times as each feature was fully tested and fixed within the defined cycle. By 6 July, the website was fully developed and functioning as intended with no major bugs.

The team used Visual Studio Code's Live Share feature to code simultaneously. After each coding session, the host of the Live Share session uploaded all the website files to a Google Drive folder to allow other team members to download the latest code. This allows the team members to work on the website on their own when not all members are free. When edits were made to the code, the files were uploaded to the Drive folder to ensure that all team members will have the latest version of the website.

6.1 Software Functional Requirements

The functional requirements follow the in-scope elements of the project to achieve the goal and objectives of the project. Firstly, the navigation bar was developed to allow users to access the different product categories and the user's account, whereas the footer was developed to display contact information for users to contact the Company. The navigation bar is permanently displayed to always allow users to navigate to different product category pages, whereas the footer is always displayed to provide users with easy-to-access information on how to contact the Company.

Secondly, the development of the home/landing page is necessary because a landing page provides users with information about the website and the product categories available when they click on the website's link. Without adequate information about the website and available product categories, users would not know what the website is about and what they can do with the website.

Moving on, the products page is needed to display all available models for purchase from each product category. As such, each of the 5 categories - laptops, desktops, mice, keyboards and monitors - have a unique products page to display the available models.

The fourth requirement - a product details page - is required to display the selected model's image, name, specifications, price and provide users with the ability to add the selected model to their cart for purchase. With this, users will be able to know more about their selected model and also submit their desired order details for the selected model.

In addition, the requirement of a checkout page is to allow users to purchase their cart items. This page also serves as a final page before submitting orders for users to check their order details before submitting the details to the Company.

The sixth requirement for a user account feature and page is to prevent users from accessing the cart and checkout feature without logging in to their account. This is because the website and database tracks users' order details using their account's email address. In addition, the account page allows users to log out of their account. This prevents unwanted access to a user's account when they have completed shopping on the website.

The reason for the development of the thank you page is to inform users that their purchase has been successfully completed and return users to the home/landing page. Without this page, users would not know that their order has been successfully submitted which would result in confusion.

Finally, the development of the website's database is required to store and retrieve all the details of all product specifications, name and price as well as user account and shopping cart details. Utilising a database allows for better maintainability and manageability of the Company's product catalogue as new products and details only need to be uploaded to the database without editing the HTML source code in order for the website to display the updated catalogue. This reduces the time and effort necessary to complete updates to the Company's product catalogue. In addition, users are able to keep their shopping cart details when they do not confirm their purchase and log out of their account.

6.2 Software Non-Functional Requirements

The first non-functional requirement of the website is that all pages must completely load contents within 1 second. This is to provide quick and reliable access to all website features. Another requirement is that all HTML, CSS, PHP and JS source codes must be designed in a manner that allows for maximum maintainability and manageability. This means that when edits are required for either bug fixes or updates, only the very bare minimum number of edits are required for full update and bug fixes. The number of edits is defined as how many files need to be edited in order for the update to be completed. There is no specification for

the number of edits because different features of the website may have one or more than one edits required.

6.3 Layout & Mockups

A total of 7 total pages are utilised in the layout of the website: home/landing page, products page, product details page, checkout page, account page, sign in page and create account page. The flow of the website can be seen in the navigation flow diagram in Figure 2. The reasoning for each page has been laid out in [6.1 Software Functional Requirements](#).

Some colour schemes considered during the design process were monochromatic, analogous, complementary and split-complementary [3]. The final colour scheme is a monochromatic colour scheme with a mixture white, light gray, gray and black. This is because a minimalistic and simple design prevents the user's attention from straying away from the main content of the website [4]. Throughout the website, a white background was used because it allows for other more important elements to stand out as any colour contrasts well against white. This contrast is important to maintain users' attention on the main content of the website and also help users easily identify the functions that they want to use [5].

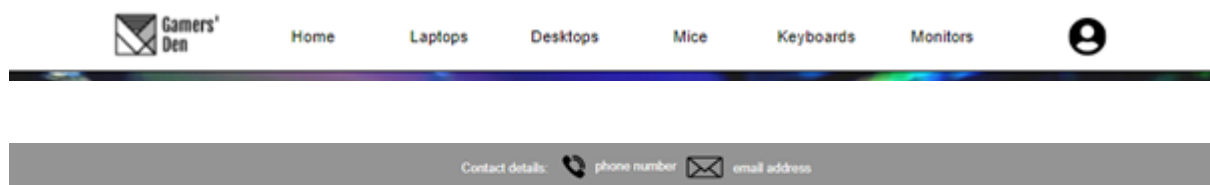


Figure 3.1 - Mockup of the navigation bar and footer. Displays information about the website.

The table, grid and flexbox display types were considered for both the navigation bar and footer. In the final design, the navigation bar utilises flexboxes because it needs to be automatically resized when the browser window is resized, whereas the footer utilises a table because only the contact details - phone number and email address - need permanent display without the need for automatic resizing.

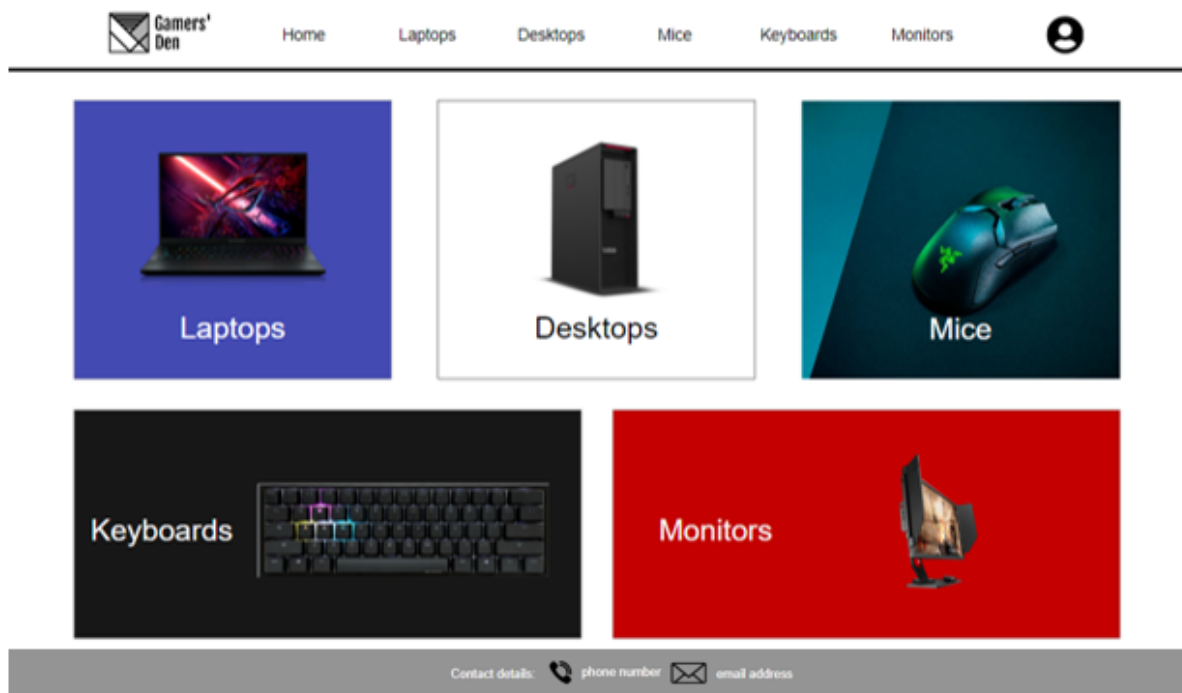


Figure 3.2 - Mockup of the home/landing page after scrolling to the bottom.

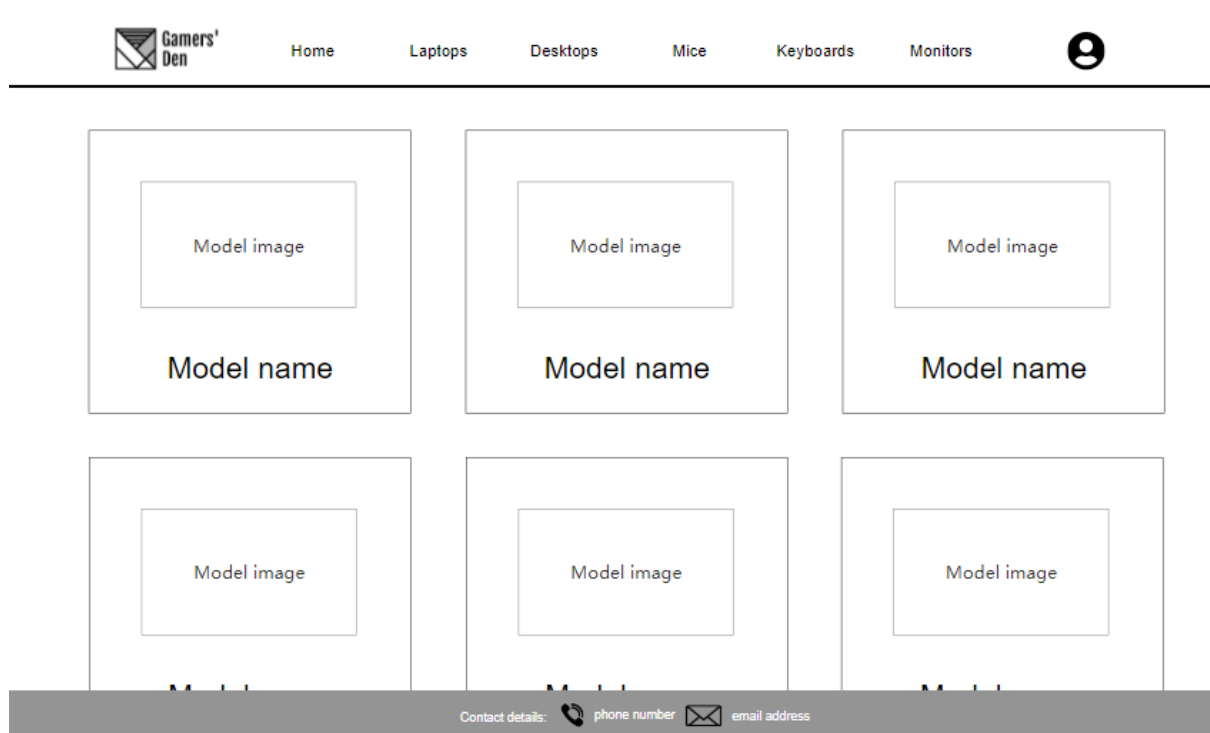


Figure 3.3 - Mockup of the products page.

The home/landing page (Figure 3.2) and products page (Figure 3.3) uses clickable “cards” and images to direct users to the intended pages - products page of the selected category from the home/landing page and selected model’s details page from the products page. The reason for this design choice is because images are easily recognised compared to text [6]. In

addition, images also invoke emotions, making the website more engaging and easier for users to arrive at their desired destination with minimal effort [6]. The cards make it easier for users to scan the whole website quickly and identify their desired model [7]. When designing the products page to display all available models from each product category, the flexbox and grid display types were considered. The final design uses flexboxes because the images need to automatically resize when the browser window is resized.

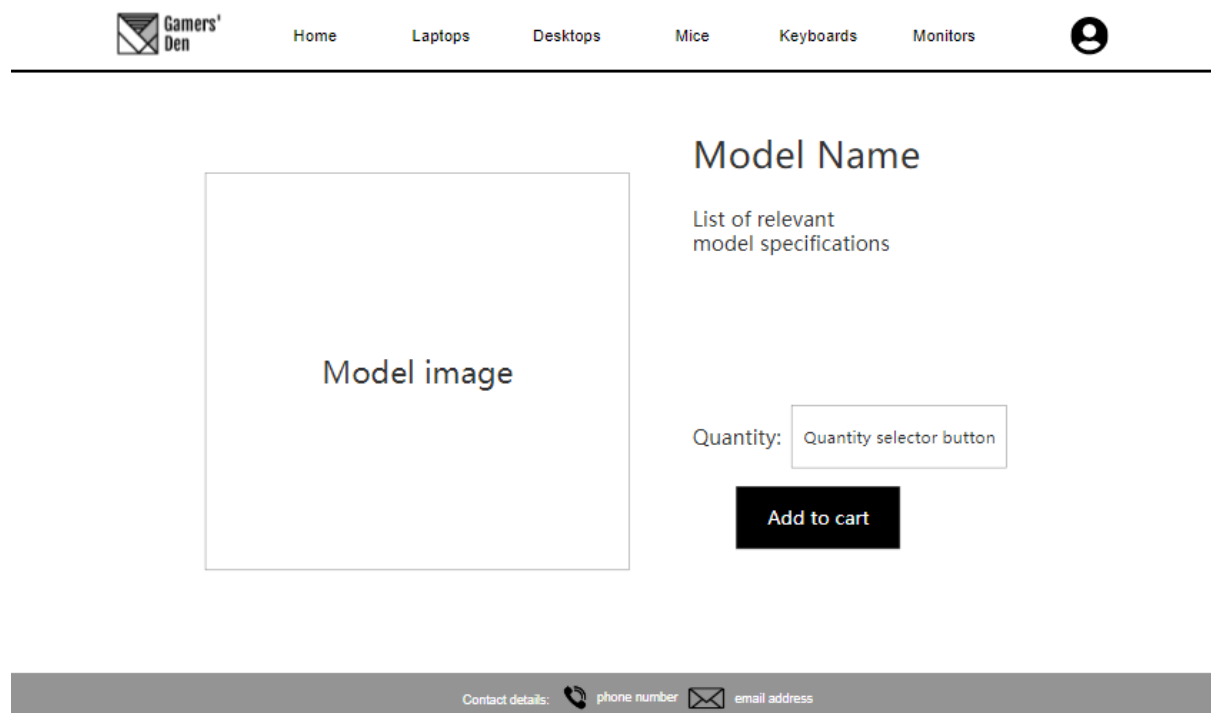


Figure 3.4 - Mockup of the product details page.

The first layout that came to mind when designing the product details page was a 2 column layout. This is because the one column is needed to display a large image of the selected model for users to get an idea of how the model looks in person. On the other hand, another column is needed to display the model name, model specifications and price for users to know specific details about the model. This column also contains a form with a Quantity selector and Add to Cart button. The Quantity selector allows users to select the quantity they wish to purchase, whereas the button submits the following details to the database and stores it in the user's shopping cart record: model name, model price and quantity.

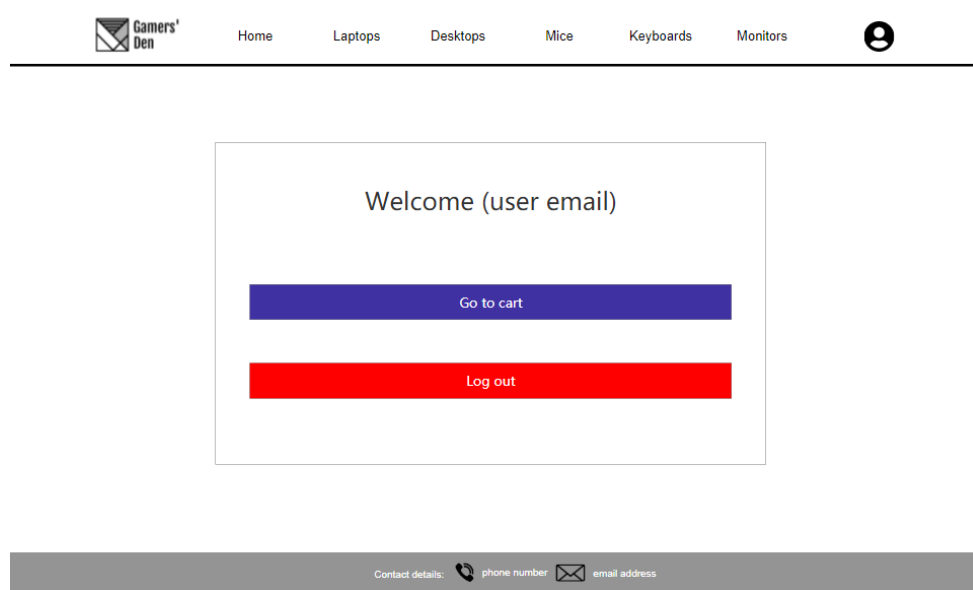


Figure 3.5 - Mockup of the account page.

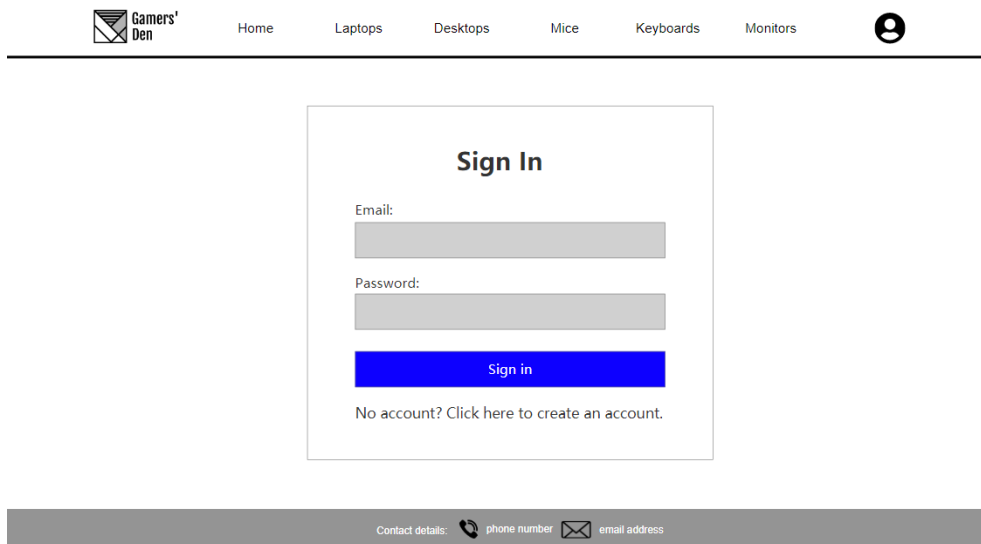


Figure 3.6 - Mockup of the Sign In page.

The account page (Figure 3.5) is only accessible after a user logs into their account. It utilises a grid to display the “card” that contains 2 buttons: “Go to cart” and “Log out”. The grid display type was used because it is easier than flexboxes to manage the “card’s” position on the page. Additionally, there is no need for the card to automatically resize as it is always centered to the page. The buttons’ functions are self-explanatory to make it easy for users to know what they do; “Go to cart” brings users to their shopping cart, whereas “Log out” ends the user’s session by logging them out of their account. Moreover, the Sign In (Figure 3.6) and Create Account pages both use the same card-grid design for the same reasons. However, the card contains the form for the user to enter their account details. This makes it easy for user’s

to identify where they need to input their account details. Both pages display a button that submits the details to the database for account verification.

Figure 3.7 - Mockup of the checkout page.

The checkout page is only accessible from the account page. This flow prevents users from accessing the checkout page containing shopping cart details without first logging in. The checkout page utilises a table to display following details of the items added by users: Product Name, Quantity, Price Per Unit and Total Price. The “Delete” section displays a button to allow users to delete the item specified by the row of the button. The table display type was chosen because a user’s shopping cart can have none, one or as many items as they desire. Tables are suited for displaying numerous texts in an easy-to-read manner. As such, the table display type was chosen in the final design.

Error Message	Purpose
	Appears when the user attempts to add an item to the shopping cart without logging in to their account. Informs the user to create an account or login to their account to continue.
	Appears when the user enters incorrect details in the Sign In page. Informs the user of the possible errors that prevented them from logging in.

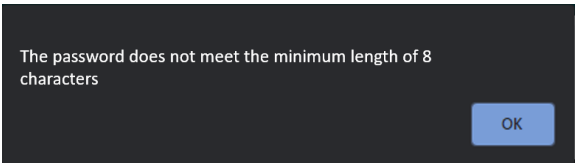
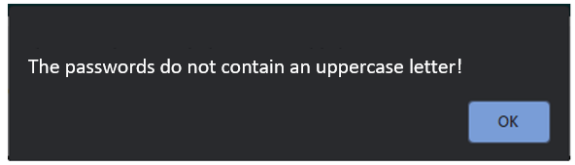
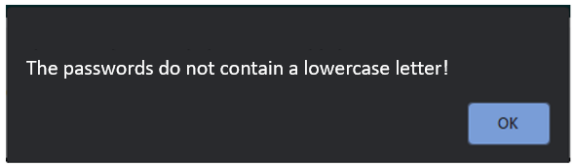
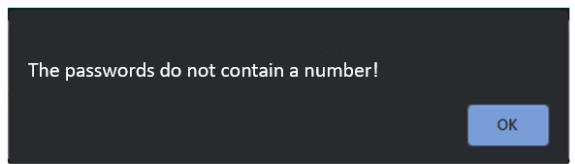
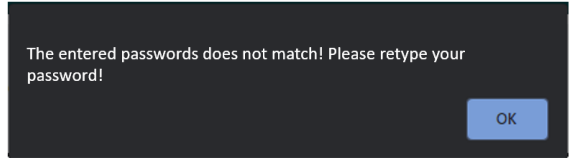
 <p>The password does not meet the minimum length of 8 characters</p> <p>OK</p>	<p>Appears when the user enters a password that does not meet the minimum length during account creation at the Create Account page. Informs the user of the minimum password length.</p>
 <p>The passwords do not contain an uppercase letter!</p> <p>OK</p>	<p>Appears when the user enters a password without an uppercase letter during account creation at the Create Account page. Informs the user to enter a password with an uppercase letter.</p>
 <p>The passwords do not contain a lowercase letter!</p> <p>OK</p>	<p>Appears when the user enters a password without a lowercase letter during account creation at the Create Account page. Informs the user to enter a password with a lowercase letter.</p>
 <p>The passwords do not contain a number!</p> <p>OK</p>	<p>Appears when the user enters a password without a number during account creation at the Create Account page. Informs the user to enter a password with a number.</p>
 <p>The entered passwords does not match! Please retype your password!</p> <p>OK</p>	<p>Appears when a user enters unmatching passwords during account creation at the Create Account page. Informs the user to retype their password.</p>

Table 3.8 - Mockup of the required error messages and their purpose.

6.4 Database Design

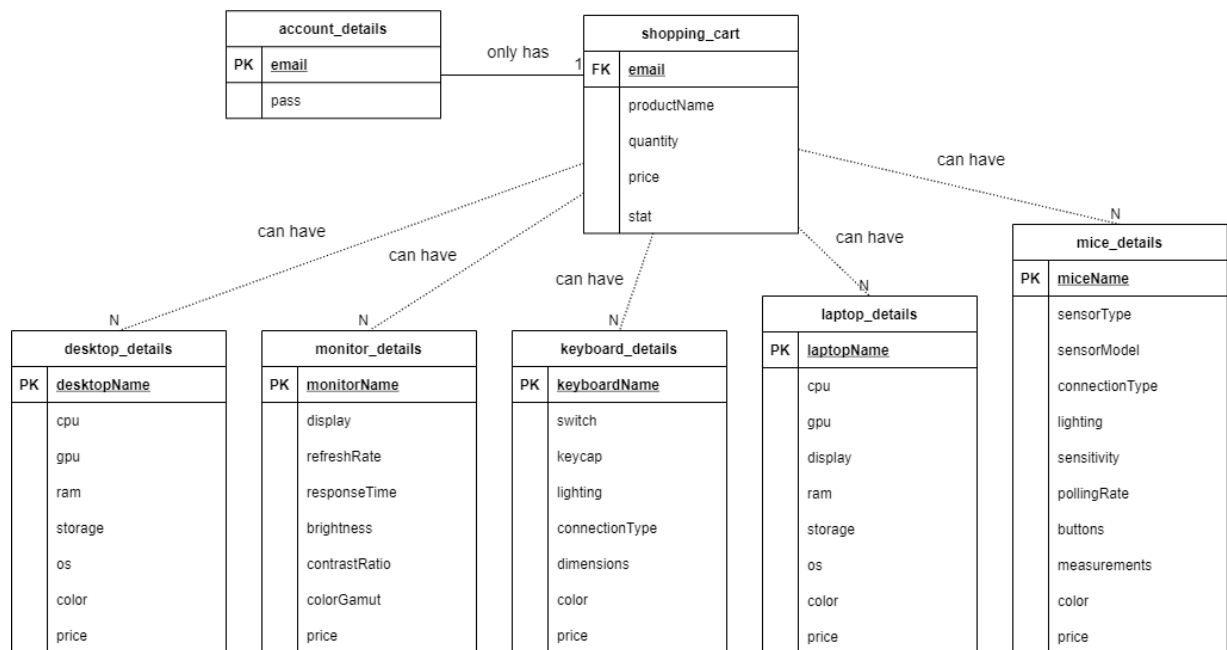


Figure 4 - RDM for the database.

As stated in [6.1 Software Functional Requirements](#), the database is used to store all product details - name, price and relevant specifications - and user account and shopping cart details. When required, the website will extract the details of a product from the database and display it in the product details page. The database stores the product's details according to the product's category: the `desktop_details` table stores the details of all desktops, `laptop_details` stores the details of all laptops and so on.

The `account_details` table stores users' email and password. Each user only has one shopping cart as it is an inefficient use of storage to create multiple carts for the same user. When verifying users accounts during Sign In, the website will submit the Sign In details to the database to check if the email and password are correct.

The `shopping_cart` table contains the details of the items added by the user to their cart. As such, the `shopping_cart` table can contain none, one or many products of the same or different types. The checkout page uses the data from the `shopping_cart` table to display the logged in user's item. To identify the logged in user's items, the shopping cart's foreign key email is used to identify the user.

7.0 Testing Results & Discussion

Navbar

Test Case	Browser	Results	Observation
1. Navbar	Google Chrome	Working	The navbar displays as a row with specified width and height. The other styles declared such as color, background color, and font weight are also displayed.
2. Navbar	Mozilla Firefox	Working	
3. Shrinking Navbar	Google Chrome	Working	The texts do not shrink, but the images shrink when the width of the window reaches approximately 1380px.
4. Shrinking Navbar	Mozilla Firefox	Working	

Table 5.1 - Navigation bar testing for possible events in Google Chrome and Firefox

Based on the results shown in Table 5.1, the navbar works as intended for both test browsers. The navigation bar or navbar uses the flex display type and flex-wrap CSS property. When testing the Shrinking Navbar case, the navbar looks different/smaller for different screen sizes, hence it is not used for user satisfaction. When the navbar is minimized to the minimum browser window size, the image hyperlinks for the logo and account page cannot be seen. This can be fixed by setting a minimum width threshold so that when the threshold is met, the images will stop shrinking.

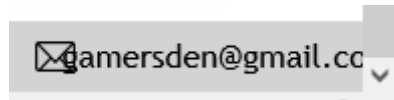
Footer

Test Case	Browser	Results	Observations
1 Footer	Google Chrome	Working	Works as intended. The footer spans the width of the browser, but the content is centered horizontally. All styling is displayed as intended.
2 Footer	Mozilla Firefox	Working	
3. Footer Shrinking	Google Chrome	Working	The words and images overlap at

4. Footer Shrinking	Mozilla Firefox	Working	approximately 500px as the window shrinks.
---------------------	-----------------	---------	--

Table 5.2 - Footer testing for possible events in Google Chrome and Firefox

Based on the results obtained in Table 5.2, the footer functions as intended. When the window is minimized to the minimum browser window size, the words and the images in the footer overlap.



Screenshot displaying the result from Footer Shrinking

Image Hyperlinks

Test Case	Browser	Results	Observations
1. Images	Google Chrome	Working	When images are clicked, users are directed to the specified page.
2. Images	Mozilla Firefox	Working	
3. Images Shrinking	Google Chrome	Working	When the browser window is resized the hyperlinked images are resized due to the usage of flex boxes.
4. Images Shrinking	Mozilla Firefox	Working	

Table 5.3 - Image hyperlinks test for home/landing and product pages in Chrome and Firefox

Based on the results shown in Table 5.3, all the image hyperlinks work as intended when testing with the two test browsers. Each row of products is a flexbox containing flex items. Each flex item contains the product's image displayed above its name. The entire flex item is a hyperlink to the intended details page. When clicked, the flex item is also a form which submits the name of the product to a PHP file for processing the correct details in the product details page. A border appears around the flex item when the mouse hovers over it to indicate to users that the image is clickable and will bring the user to the product's details page.

Buttons

Text Case	Browser	Results	Observations
Add to Cart	Google Chrome	Working	Add the selected item into the database

	Mozilla Firefox	Working	and bring the user to the Checkout page as intended.
Confirm Purchase	Google Chrome	Working	Change the status of cart items from “pending” to “purchased” in the database so users’ carts will not display purchased items. Also brings users to the ThankYou page.
	Mozilla Firefox	Working	
Delete	Google Chrome	Working	Delete the selected item from the database.
	Mozilla Firefox	Working	
Quantity	Google Chrome	Working	Increase/Decrease the quantity of the item users desire to purchase.
	Mozilla Firefox	Working	
Create Account	Google Chrome	Working	Insert account data into the database and brings users to the Sign In page.
	Mozilla Firefox	Working	
Sign In	Google Chrome	Working	Initializes the session and brings users to the home/landing page.
	Mozilla Firefox	Working	
Log out	Google Chrome	Working	Destroys the session and brings users back to the Sign In page.
	Mozilla Firefox	Working	
Go to cart	Google Chrome	Working	Brings users to the checkout page containing their cart details.
	Mozilla Firefox	Working	

Table 5.4 - Button testing for buttons used in Chrome and Firefox

From the results in Table 5.4, all the buttons in the website function as intended when testing in both test browsers. The “Confirm Purchase” button changes the status from “pending” to “purchased” in the database and directs the user to the Thank You page. The button “Quantity” allows the user to choose how many quantities they wish to purchase the item in. When the user clicks the “Add to Cart” button, the website will check if the user is logged into their account or not. If the user is logged in, the details will be submitted to the database

and the user is redirected to the Checkout page. If not, the user is prompted to login to their account. Moving on, the “Sign In” button checks if the exact details of the user is in the database, and will display a message if their details are not correct. The messages displayed can be found in Table 3.7. The “Log out” button ends the session for the users and redirects them to the Sign In page whereas the “Go to cart” button redirects the user to the Checkout page.

8.0 Conclusion & Future Works

In conclusion, the team successfully achieved the goal and objectives of the project within the specified timeline. For the frontend website, the objective achieved is to develop and deploy an e-commerce website that allows customers/users to view all available products and product details and allows users to submit their order details to the Company. For the backend structure, the objective achieved is to develop and deploy a database used by the website to store the products and product details as well as users’ account and shopping cart details.

As for future works, there are some enhancements that are identified for the website. The first is to add images of products in the checkout page for users to easily recognise if the product is what they desire. Another enhancement is to have an auto-logout feature to log users out when they close the website or their browser. This will ensure that users’ accounts will not be accessed by unwanted parties who may have gained access to users’ devices. Thirdly, a useful enhancement is to allow users to track their purchase items’ status. This will ease users’ worry if their product will get delivered or not. Next, instead of directing users to the home page when they sign in, it would be more convenient to direct them to the previous page they were on, i.e. the product details page. Last but not least, more specific error messages in the sign in page rather than combining the possible errors into one message will aid users in quickly identifying what they need to do to solve the error, reducing frustration and time-wasting.

There are two enhancements identified for the source codes. Firstly, it would be more efficient to combine all database PHP scripts into one file and use a sentinel variable to control the execution of the correct script in order to edit the database records. The current design of using multiple files per script will become inefficient for bug fixing when new features are added because there would be too many script files to identify the bug, hence

spending unnecessary additional effort and time. The other would be to improve source code execution efficiency. This is to solve a rare error that appeared during testing whereby if done fast enough, the page will display PHP error messages instead of the intended product details in the product details page. Clicking the back and forward buttons in the browser solves the error but it would be best to iron out these minor bugs to provide a more polished experience.

References

- [1] A. Gaffar Khan, “Electronic Commerce: A Study on Benefits and Challenges in an Emerging Economy”, *Global Journal of Management and Business Research: BEconomics and Commerce*, vol. 16, no. 1, 2016. [Online]. Available: <https://www.journalofbusiness.org/index.php/GJMBR/article/view/1918/1820>. [Accessed July 9, 2021]
- [2] A. Pawlicka, *Agile Software Development Process - Everything You Need to Know*, selleo, Feb. 23, 2021. Accessed on: July 6, 2021. [Online]. Available: <https://selleo.com/blog/agile-software-development-process-everything-you-need-to-know>
- [3] D. Alscher, *The 6 Color Schemes to Keep Everything Picture Perfect*, G2, June 6, 2019. Accessed on: July 6, 2021. [Online]. Available: <https://www.g2.com/articles/color-schemes>
- [4] Digitech, *13 Amazing Benefits of Minimalist Web Design*, Digitech, June 20, 2014. Accessed on: July 7, 2021. [Online]. Available: <https://www.digitechwebdesignaustin.com/web-design-austin/13-amazing-benefits-minimalist-web-design/>
- [5] J. Kliever, *Designing with contrast: 20 tips from a designer*, Canva, n.d. Accessed on: July 7 2021. [Online]. Available: <https://www.canva.com/learn/contrasting-colors/>
- [6] GetFeedback, Inc., *10 Things To Know About Images In Web Design*, GetFeedback, Inc., n.d. Accessed on: July 7, 2021. [Online]. Available: <https://www.getfeedback.com/resources/ux/10-things-to-know-about-images-in-web-design/>
- [7] A. F, *Card-based Design: the Pros & Cons and Best Practices*, UX Planet, July 12, 2018. Accessed on: July 7 2021. [Online]. Available: <https://uxplanet.org/card-based-design-the-pros-cons-and-the-best-practice-2e5ccd6a606d>