LINCIA WALTERS

2136919

Leveraging enhanced User Experience (UX), Predictive Analytics & Artificial Intelligence (AI)

to augment an eCommerce application to boost sales and customer retention

BSc Information Technology
Undergraduate Thesis Report

Department of Computer Science & Technology
University of Bedfordshire

AY21/22

Supervisor: Ravi Ragoonath

ABSTRACT

The prevalence of online shopping for computer hardware in Trinidad and Tobago had been steadily increasing prior to the Covid-19 pandemic but it quickly became one of the few options available. Because of health protocols, everyone was forced online: consumers and businesses alike. Online shopping internationally is the de facto method while online shopping locally is still trying to take root in a sustainable way.

This study will outline the development of an artefact which utilizes good UX design, AI & predictive analytics to alleviate these issues while increasing sales and making shopping user-friendly and as easy as shopping internationally.

It will detail the background research, requirements analysis, design, implementation and testing phases the artefact underwent to achieve its goals. Finally, a summary of the functionalities implemented, a look at how well the artefact achieved its goals, and a reflection on the experiences gained will ultimately be done.

ACKNOWLEDGEMENT

Firstly, I would like to acknowledge the University for accepting me into the final year of this BSc but also for providing me with access to the learning resources necessary for completion. Finally, to Ravi Ragoonath and Marcus Alexander for their constant support and commitment to their students which invariably helped support my studies.

DEDICATION

This report is dedicated to God who continue to guide and strengthen me daily to complete feats like this one. Also, to my understanding friends and family who ceaselessly continue to believe in me.

KEYWORDS

ARTIFICIAL INTELLIGENCE

E-COMMERCE

CHATBOT

ENCRYPTION

ANALYTICS

USER EXPERIENCE

ONLINE SHOPPING

RECOMMENDER SYSTEMS

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Introduction

The project's goals are explained in the Introduction Chapter, together with background information about the industry, the issue, and the project's approach to resolving it. The Introduction Chapter serves as a guide for the remaining chapters of the project by giving a brief overview of each chapter that will be included in the final report.

1.2 The Subject

In the current climate, online computer hardware applications exist but not in a way that utilizes the upsides of shopping online to make online shopping locally for computer hardware worth it more than online shopping internationally. Whether it's because of websites not being 100% user-friendly or not enough variety to help a buyer make good buying decisions, there is hardly any incentive to purchase locally. The project seeks to fill this gap by developing an online computer hardware application that utilizes good UX design, AI & predictive analytics to alleviate these issues while increasing sales and making shopping user-friendly and as easy as shopping internationally.

To get a full picture of the online shopping climate in Trinidad and Tobago as it relates to computer hardware e-commerce applications, gathering requirements need to be done. The below will be used to do this:

1) Researching current local online computer hardware stores and reviewing gaps in their build.

2) Secondary information gathered from websites, journals, books, newspaper articles.

The research avenues mentioned above along with overwhelming evidence that the use of AI, predictive analytics and good UX design dramatically improves customer service, increases sales, and enhances the customer experience helped with the decision to move forward with this project. According to a ChatBot (n.d.) survey for example, 64% of customers said 24/7 availability of a chatbot is the top benefit of AI. Further, Lowden (2014) explains how good UX have a direct relationship with good customer experience.

There are a large amount of brick-and-mortar computer hardware stores in Trinidad and Tobago but not nearly enough options online. PRWE, loprofilepc and njcomputerstt are 3 local examples of online computer hardware applications. Amazon, Newegg and eBay are 3 international examples. Both local and international online stores each have their own unique challenges for consumers detailed below.

Local stores:

1) The sites are poorly designed.

2) There is a lack of engaging UX on the sites while being generally outdated.

3) A lack of variety to help buyers make better buying decisions.

4) No implementation of AI technology to assist customers.

International stores:

1) There are a lot of additional charges like customs charges, online purchase tax and courier shipping charges.

2) A longer wait time.

3) Limits on foreign exchange.

My artefact will solve these problems by:

1) Being designed with good UX in mind.

2) Having enough variety.

3) Computer parts will be indexed in a way that is easy and intuitive for the user to browse.

1.3 Project Aim

The aim of this project is to use the agile methodology, Rapid Application Development (RAD), to develop a computer hardware e-commerce application that utilizes sound UX design, predictive analytics and AI to enhance the customer experience, increase sales and encourage repeat buying respectively.

1.4 Project Objectives

Pineda (2017) defines objectives as “concrete actions needed to achieve the aspirations or steps required to reach predetermined destinations”. The below is a list of objectives for this project:

1) To complete background research with the goal of completing a literature review. This will be done using secondary data pulled from authoritative sources by week 1 of semester 2.
2) To complete requirements analysis by doing a list of all the functional and non-functional requirements to avoid any misalignment about what the system is supposed to do between stakeholders by week 2 of semester 2.
3) To design a minimum viable front-end and an entity-relationship diagram for a bird's eye view of the database back-end by week 3 of semester 2.
4) To implement the functions of the application using the relevant programming language to ensure the artefact is viable enough for its intended purpose by week 10 of semester 2.
5) To test the artefact using a varied amount of test data to ensure the artefact functions in the ways it should by week 11 of semester 2.
6) To evaluate the application’s usability by testing it in a live environment to determine the artefact's level of functionality by week 12 of semester 2.
7) To compile a final Microsoft Word report which will include previously mentioned objectives as individual chapters to document the project's progress by week 12 of semester 2.

1.5 Summary of Chapters

The Final Report is a key deliverable for the Undergraduate Project module. The purpose of this report is for students to plan, create, and polish a major technical report that reflects a course of work intended to develop and broaden their grasp of subject areas related to their studies.

Further, this technical report will examine each stage of the creation of the software artefact while giving the student the chance to analyse and review their work. All supplementary materials, including Project Progress Reports, evaluations, and suggestions from my supervisor, will also be supplied.

The suggested format for the final project report is outlined as follows:

- Chapter 2: Project Management
- Chapter 3: Background Research
- Chapter 4: Requirements Analysis
- Chapter 5: Design
- Chapter 6: Software Implementation
- Chapter 6: Testing
- Chapter 8: Software Evaluation
- Chapter 9: Conclusion

Table 1 below gives a brief summary of each chapter:

Chapter 2- Project Management
This chapter describes the project management methodology employed during this Project.
Chapter 3- Background Research
The Literature Review, which is found in the Background Research chapter, provides the theoretical framework for the project by drawing on relevant critical literature and comparable work in the field to aid the reader in comprehending the problem domain and assisting with the design and implementation phases of the creation of the artefact.
Chapter 4- Requirements Analysis
The Requirements Analysis chapter provides a thorough summary of both the Functional and Non-Functional system requirements as well as a report on the set of requirements that served as the project's starting point and an explanation of the methods used to acquire the requirements.
Chapter 5- Design
The requirements specified in the Requirements Analysis chapter are implemented in the Design chapter, which also discusses the design choices that were used to do so. It also examines the Human-Computer Interaction (HCI) factors and how they affected the Artefact's design.
Chapter 6- Implementation
The Implementation chapter gives an overview of the project's practical work and contains comments on the development environment selected, the databases and web applications chosen, the method for version control, and the specific software tools used to develop the final artefact.
Chapter 7- Testing

The techniques used to test the artefact's functional and non-functional requirements, as well as the results of such testing, are covered in the Testing chapter.
Chapter 8- Evaluation
In Evaluation chapter, the usability of the artefact is the main focus, and heuristic testing is used with a checklist derived from Nielsen's 10 heuristics.
Chapter 9 -Conclusion
The Conclusion chapter includes a self-critical evaluation of how well the project's goals were met or came up short. It also includes a critical analysis of the finished piece of software that was created.

Table 1: Brief summary of final report chapters

CHAPTER 2: PROJECT MANAGEMENT

2.1 Introduction

The Project Management Institute (2018) states that an average of 50% more projects fail in organizations that undervalue project management as a strategic capability for achieving desired results. This chapter examines the role of project management. It evaluates the importance and effect of good project and time management by coupling the Work Breakdown Structure (WBS) and the GANTT chart to successfully manage the project's deliverables.

2.2 Importance of Project Management

According to PMI (2018), adhering to project management methods and strategies reduces risks, cut costs and improves success rates. Therefore, in order to skilfully manage the various parts of completing this project, project management is crucial. This project has many deliverables along with a variety of new skills and technologies to learn. Moreover, project management is crucial to effectively complete this project on time because of the short deadline and penalties for late submission.

The project plan is essential because it acts as a road map for the project and ensures that activities are completed on schedule. Therefore, it must be closely followed in order to ensure that the timeframes and deadlines are met in order to finish the submission on time.

The Project Handbook has offered specified dates for the assignments as well as suggested times for meetings with the project supervisor since the Undergraduate Project unit has begun. To provide a visual depiction of each project stage, a WBS and a Gantt chart was created.

2.3 Work Breakdown Structure

A Work Breakdown Structure (WBS) is a hierarchical outline of the tasks required to complete a project. The WBS “breaks down” the structure of a project into manageable deliverables. Each deliverable is assigned a task, or series of tasks that can be further broken down into subtasks to meet the needs of the project (Mong, 2020).

A WBS should display every deliverable for the project in a single, hierarchical graphic. That graphic can be a flowchart, a table, an outline, or anything else that visually represents everything within a hierarchy (Lynn, 2018).

The first stage in creating the WBS was to determine the high-level tasks, which corresponded to the remaining project deliverables as determined by the Project Handbook.

For the unit, the following six (6) primary deliverables were identified:

1. Reflective Report
2. Software Artefact
3. Final Report
4. Poster
5. Project Viva
6. Project Supervisor Meetings

The second stage involves breaking 5 of these high-level tasks into lower-level sub-tasks:

1. Reflective Report
    a. Self-Reflection
    b. List of Chapters of Final Report

2. Software Artefact
    a. Develop Core Features
    b. Develop Advanced Features

3. Final Report
    a. Complete Writeup of Each Chapter

4. Poster
    a. Create Poster

5. Viva
    a. Prepare Presentation
    b. Schedule Date
    c. Attend Viva

The WBS for this project is provided in Figure 1 below:

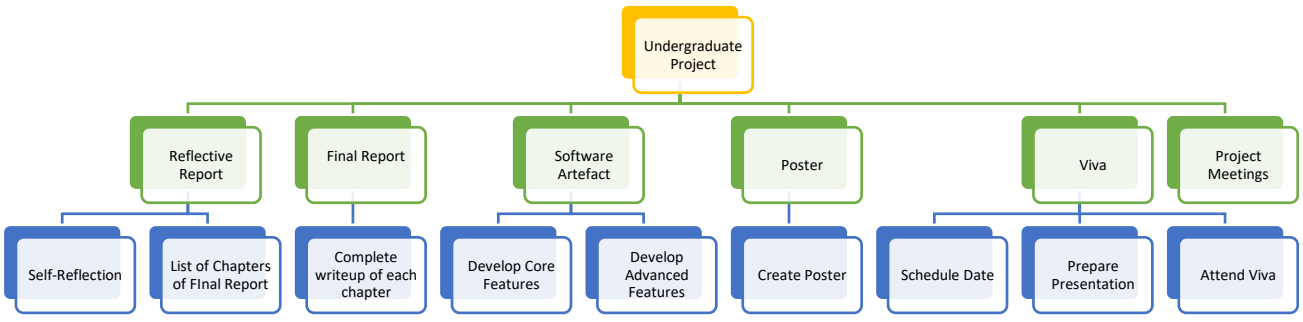


Figure 1: Work breakdown structure

2.4 Gantt Chart

Gantt charts are visual representations of tasks plotted against time (Projectsly, n.d.). Gantt charts are useful for planning and scheduling projects. They help you assess how long a project should take, determine the resources needed, and plan the order in which you'll complete tasks. They are also helpful for managing the dependencies between tasks (MindTools, 2021).

Gantt charts allow project managers to:

- Track projects better
- Boost productivity
- Manage complexity within the project
- Set realistic deadlines and expectations
- Improves clarity
- Keep the team up to date on progress and changes

The Gantt chart in Figure 2 below was constructed by first identifying the tasks using the WBS as well as all the subtasks. The GANTT chart initially assisted but it had to be redone because of overly ambitious deadlines and underestimating the time learning curves span. As a result, the dates on the GANTT Chart had to be changed to reflect the updated schedule as well as the tasks that still needed to be completed along with any progress that had been made.

Figure 2 shows the original GANNT chart while Figure 2a shows the updated one:

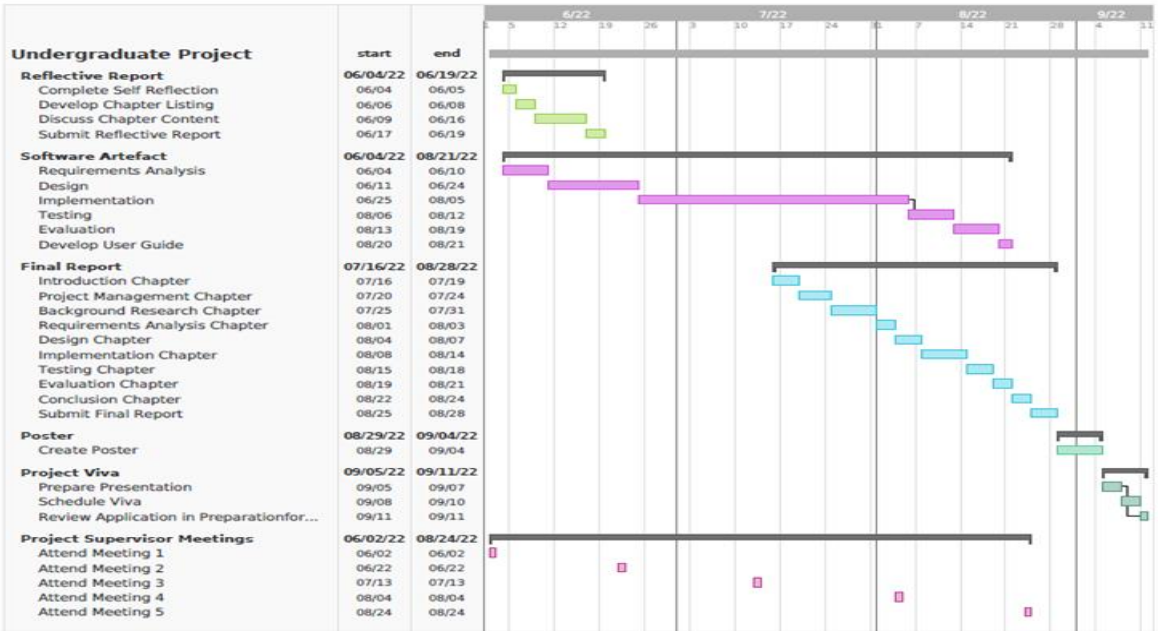


Figure 2: Original GANTT Chart for the Undergraduate Project

PROJECT PLAN
UoB BSc IT Level 6 - Undergrad Project

TaskID	Task	Est. Start Date	Est. End Date	Duration	Predecessor Task	Success Criteria	Actual Start Date	Actual End Date	% Complete
1	Reflective Report	Week 2	Week 6	3 weeks		Submission of Reflective Report			100%
1.1	Complete Self Reflection	Week 2	Week 4	2 weeks		Critical review of time management			100%
1.2	Develop Listing of Chapters	Week 4	Week 5	1 week	1.1	All chapters included based on SDLC			100%
1.3	Discuss content of Chapter	Week 5	Week 6	1 week	1.2	Description provided for each chapter of report			100%
2	Software Artefact	Week 1	Week 12	12 weeks		Submission of Software Artefact			100%
2.1	Requirements Analysis	Week 1	Week 3	2 weeks		Detailed List of Functional and Non-Functional Requirements			100%
2.2	Design	Week 3	Week 5	2 weeks	2.1	A Complete list of normalized tables for Backend; Low Fidelity Prototype for Front End			100%
2.3	Implementation	Week 5	Week 10	5 weeks	2.2	A High Fidelity Prototype of the Ecommerce System			100%
2.4	Testing	Week 10	Week 11	1 week	2.3	A complete set of test plan and test results for the entire artefact			100%
2.5	Evaluation	Week 11	Week 12	1 week	2.4	A heuristic evaluation of the software artefact			100%
2.6	User Guide	Week 12	Week 12	2 days	2.5	Develop a user guide to submit with the artefact			
3	Final Report	Week 7	Week 12	5 weeks		A completed technical report			100%
3.1	Introduction Chapter	Week 7	Week 7	0.5 week		A chapter outlining the Aims and Objectives and Project Background			100%
3.2	Project Management Chapter	Week 7	Week 8	0.5 week	3.1	Chapter outling the Project Management work conducted			100%
3.3	Background Research Chapter	Week 8	Week 8	0.5 week	3.2	Chapter outlining the Literature Review into eCommerce Application			100%
3.4	Requirements Analysis Chapter	Week 8	Week 9	0.5 week	3.3	A complete list of functional and non functional requirements and use case diagrams			100%
3.5	Design Chapter	Week 9	Week 10	1 week	3.4	A chapter outlining the work done to develop the Database back end and the front end.			100%
3.6	Implementation Chapter	Week 10	Week 11	1 week	3.5	A chapter outlining the work done in implementing the application			100%
3.7	Testing Chapter	Week 11	Week 11	0.5 week	3.6	A chapter outlining the test scripts and the test results showing evidence of testing along with defects log for all program defects			100%
3.8	Evaluation Chapter	Week 11	Week 12	0.5 week	3.7	A chapter discussing the work done in completing heuristic evaluation of the system			100%
3.9	Conclusion Chapter	Week 12	Week 12	0.5 week	3.8	A refection on the Project as a whole.			100%
4	Poster	Week 11	Week 12	1 week		Submission of Poster			100%
5	Project Viva	Week 13	Week 14	1 week		Completion of Project Viva with Examiners			
5.1	Schedule Date for Viva	Week 13	Week 13	1 day		Have a date and time for Viva			
5.2	Prepare Q&A for Verbal Presentation	Week 13	Week 13	3 days	5.1	List of Q&As related to the system			
5.3	Review Application in Preparation for VIVA	Week 13	Week 14	3 days	5.2	Completion of walkthrough of the user guide.			
6	Project Meetings	Week 1	Week 13	13 weeks		Submission of 5 Project Progress Reports			60%
6.1	Project Meeting 1	Week 1	Week 1	1 day		Submission of Progress Report 1			100%
6.2	Project Meeting 2	Week 4	Week 4	1 day		Submission of Progress Report 2			100%
6.3	Project Meeting 3	Week 7	Week 7	1 day		Submission of Progress Report 3			100%
6.4	Project Meeting 4	Week 10	Week 10	1 day		Submission of Progress Report 4			
6.5	Project Meeting 5	Week 13	Week 13	1 day		Submission of Progress Report 5			

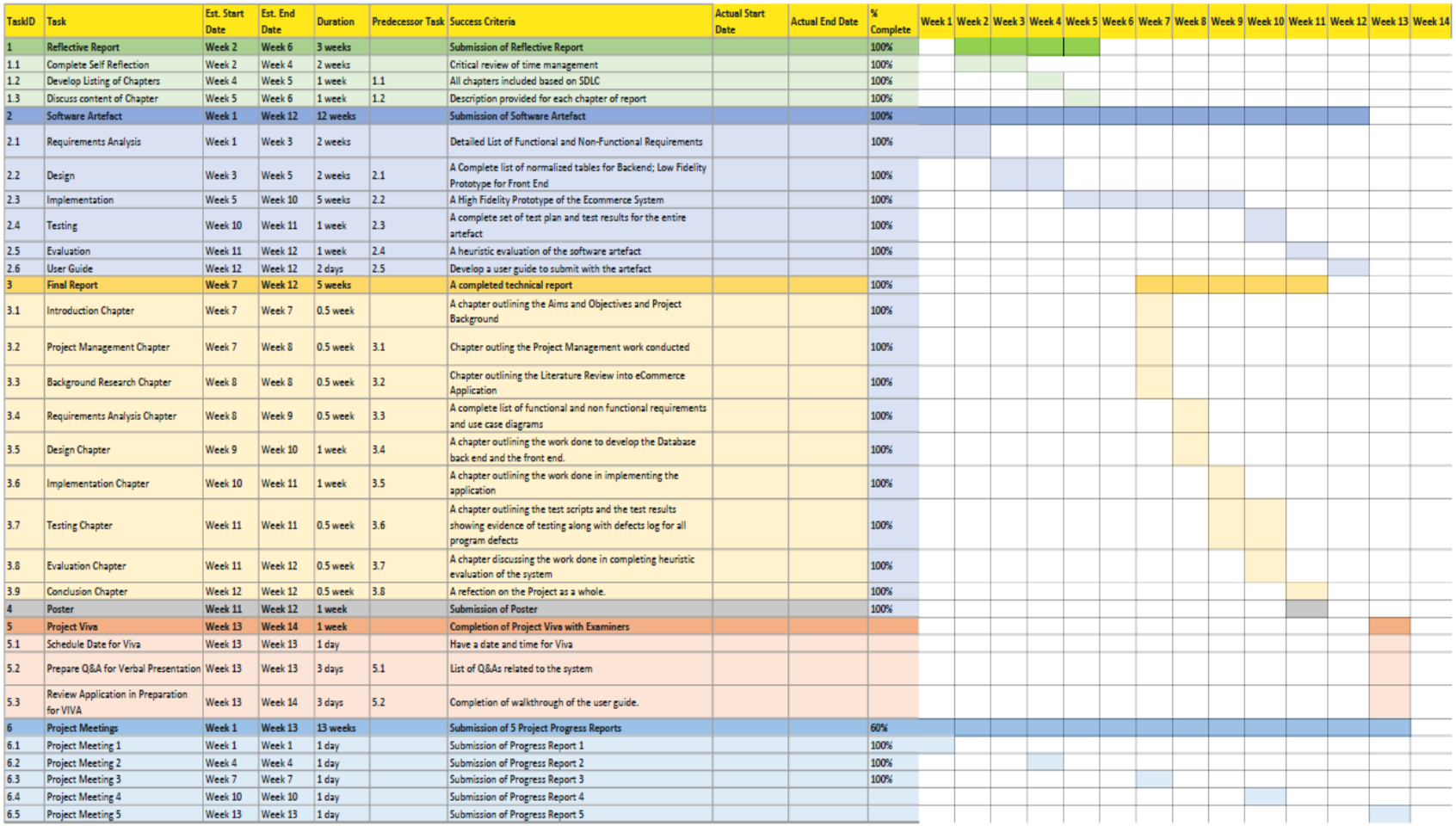


Figure 2a: Updated GANTT Chart for the Undergraduate Project

CHAPTER 3: BACKGROUND RESEARCH

3.1 Introduction

The literature review is a piece of writing that will include summarizing, synthesizing, and/or critiquing the scholarly literature identified through a literature search. It acts as an overview of the scholarly literature relevant to the project. Not only will it be utilized to inform the project's design but also its implementation. As a result, the literature review is critical in assisting the author in understanding the problem domain.

3.2 Ecommerce Systems

Generally, eCommerce (also known as e-commerce, electronic commerce, internet commerce and online commerce) is the subset of e-business that is specifically concerned with all the sales and purchases made on the internet. eCommerce has become almost ubiquitous. According to Oberlo (2021), a little more than a quarter – (27.4%) - of the world's population shop online. The ecommerce industry is rapidly developing, with new eCommerce businesses popping up all the time. The digital economy is also booming, reaching every part of the globe with an internet connection.

eCommerce in its most rudimentary form made its appearance 40 years ago when Michael Aldrich connected a customized TV to a real-time transaction processing computer via a home telephone line. This paved the way for eCommerce giants like Amazon and Alibaba to succeed in the mid-1990s and ultimately become the face of the eCommerce industry. Today, improvements in the quality and reach of internet connectivity globally, the growth of Google traffic and SEO, the rise of social media,

payment gateways, advancements in online security and growing consumer and business adoption have enabled eCommerce to grow exponentially.

There are many reasons eCommerce has experienced growing consumer and business adoption. Some are outlined below:

1. Convenience for consumers. Free shipping, free delivery, 24/7 availability and being able to transact in the comfort of one's home has added convenience to business that could not be realized before the growth of eCommerce. Not only customers enjoy these benefits, but business owners can also leverage this convenience and flexibility to build their revenue.
2. Reduced operating costs for business owners. With no need for a physical location for the actual business, no need for physical storage space, a streamlined delivery system and less employees, business owners can enjoy large costs savings and enjoy a greater profit margin.

Even with all the benefits eCommerce has afforded us, it is not without some drawbacks:

1. Less human interaction. Typically, with a physical location, a customer can walk in and be greeted with a warm welcoming smile from a sales representative whose primary job would be to assist them in their search for an item and get customer feedback, as opposed to an e-commerce store which does not possess this in-store customer engagement. This inability to get immediate customer feedback from the business' customer base can potentially dampen growth and affect sales.
2. Increased competition and its downsides. The ever-evolving industry of e-commerce has since become a hit after making its world-renowned debut back in the eighties (80’s) with all its upsides. Consequently, this naturally means an increase in online competitors, forcing eCommerce stores to expend resources to ensure survival.

The eCommerce market is always evolving with merchants building and/or enhancing their eCommerce stores more than ever before to meet customers where they are. While everything in eCommerce appears to be changing, highlighted below are the most pertinent trends that is anticipated to affect eCommerce in the coming years:

1. Augmented Reality (AR). For eCommerce, augmented reality has been a disruptive force. This type of technology allows shoppers to see the product they're looking for up close, which enhances their buying decisions. In certain areas, such as fashion and home decor, AR transforms the purchasing experience by allowing customers to get a better feel for the item without having to see it in person.
2. Artificial Intelligence (AI). Artificial intelligence and machine learning have automated and personalized shopping experiences. AI is always collecting information on how customers shop, when they make purchases, and what they want from a product or service. AI will be a game-change for eCommerce as it is a piece of technology that can't be recreated in a physical location.
3. Big Data. Big Data is expected to impact the role of eCommerce in our lives by introducing a fascinating contrast - consumers with personalized shopping experiences and consumers without. Big Data will do this for example, by constant ad suggestions on our electronic devices like our doorbells, cameras, and thermostats.

These emerging trends, while exciting, do pose some challenges. Firstly, these technologies are typically more expensive to implement. A large upfront cost to implement can affect the business' bottom line and a break-even point not eventually realized. Secondly, the real fear of data breaches is a constant threat in consumers' minds and will become more of an issue when technologies utilize more consumers' data. Although these challenges exist, the integration of emerging technologies is still being embraced globally.

3.3 Websites Reviewed

Amazon, Ebay, and Alibaba, to name a few, are amongst the most successful eCommerce websites. The employment of online payment methods, predictive analytics, recommender systems, encryption, AI and a properly designed checkout flow are just a few of the aspects that set these websites apart. This literature review aims to obtain a better understanding of some of the essential aspects that can be found on many successful eCommerce websites, with the goal of determining how they can be included into this project to assure its success. A comparative analysis of the

related software tools and development methods will also be included. The principles described above will be elaborated on below.

3.4 Ecommerce Functionalities

This section will cover the many principles that will be addressed while creating the e-commerce application. Namely, online payment methods, predictive analytics, recommender systems, encryption, AI, checkout flow.

3.4.1 Online Payment Methods

Mineraltree (2020) defines online payment methods as a means of conducting business or paying bills online or via an electronic media rather than using physical cheques or cash. Credit cards, automated clearing house, bitcoin, direct debits, and digital wallets are all examples of online payment methods. Over the years, a number of online payment options have become available but as indicated by Georgescu and Georgescu (2006), credit cards are the most used. The image shown in Figure 3 below gives a general idea of how credit card transacting is done between a merchant and a customer.

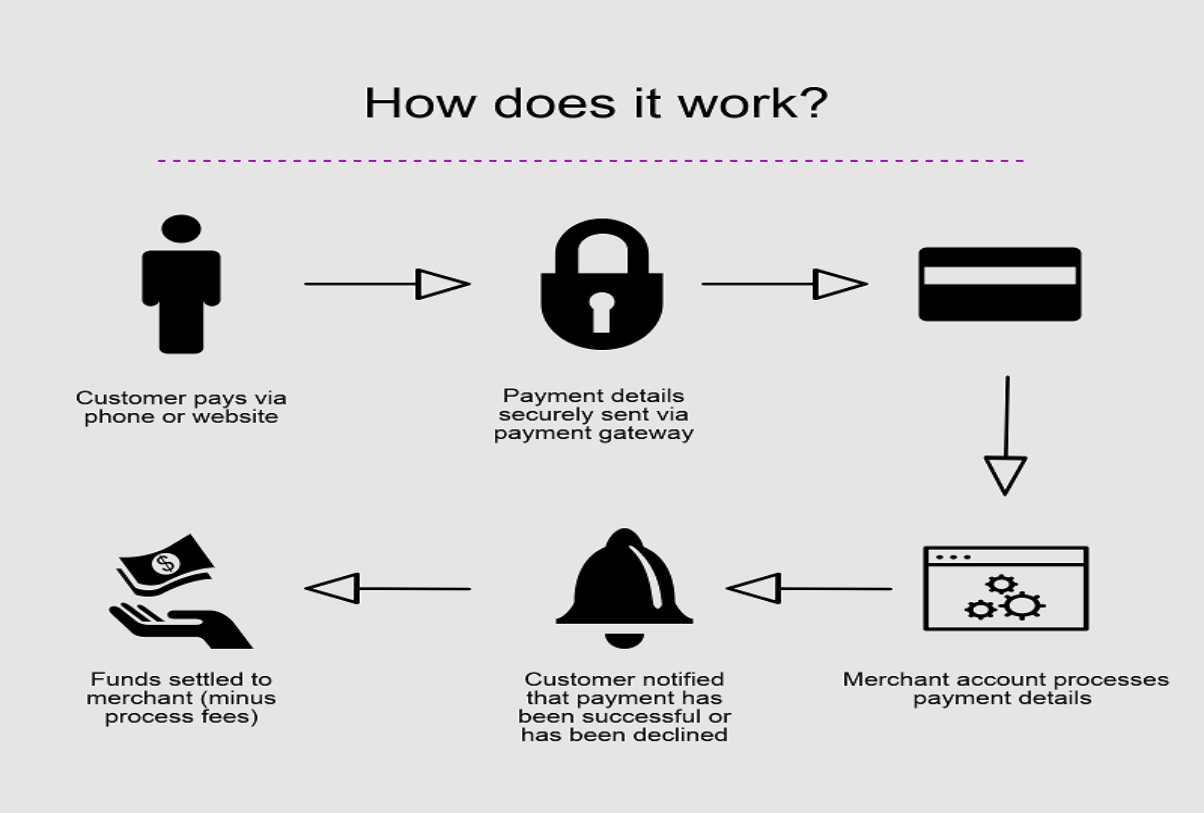


Figure 3: How online credit card payments work

Credit card numbers can be transferred encrypted or unencrypted over the Internet. Some degree of data security is provided by all internet browsers. Most browsers use a 40-bit Secure Sockets Layer (SSL), which is sufficient for most data transfer scenarios. Financial institutions and internet-capable software providers employ 128-bit SSL. It is required that websites warn customers that their credit card information is encrypted. Credit card transactions that are not encrypted are equivalent to handing your credit card number over the phone.

Online payment options may seem like a basic customer touchpoint for credibility and an optimized customer experience, but it also inherently contains a few additional benefits.

1. Global reach - Since online payment options are globally accepted, an eCommerce store will thus be accessible to a global marketplace.
2. Efficiency - Traditional solutions, like cash or cheques, are far less efficient than these computerized gateways. As Pew Research Center (2021) states, 80% of Americans have smartphones. Payments can thus be made from anywhere at any time. Customers want this convenience as opposed to less efficient methods.

As people's thirst for internet shopping has grown, so have fraud attempts. According to Review42 (2020) in 2018, nearly US$24 billion were lost globally to fraudulent credit card transactions. As a result, one of the major disadvantages of online transactions is the businesses' and customers' vulnerability to various malicious cyber assaults. Criminals are becoming more and more versatile in their attempts to disrupt online transactions, from identity theft to database breaches and phishing attacks.

Despite this cause for concern, eCommerce sales are expected to exceed US$1.3 trillion in 2025 (Coppola, 2016) with online payment options being used by all. Figure 4 below have all the major eCommerce apps using online payment options sorted by market share.

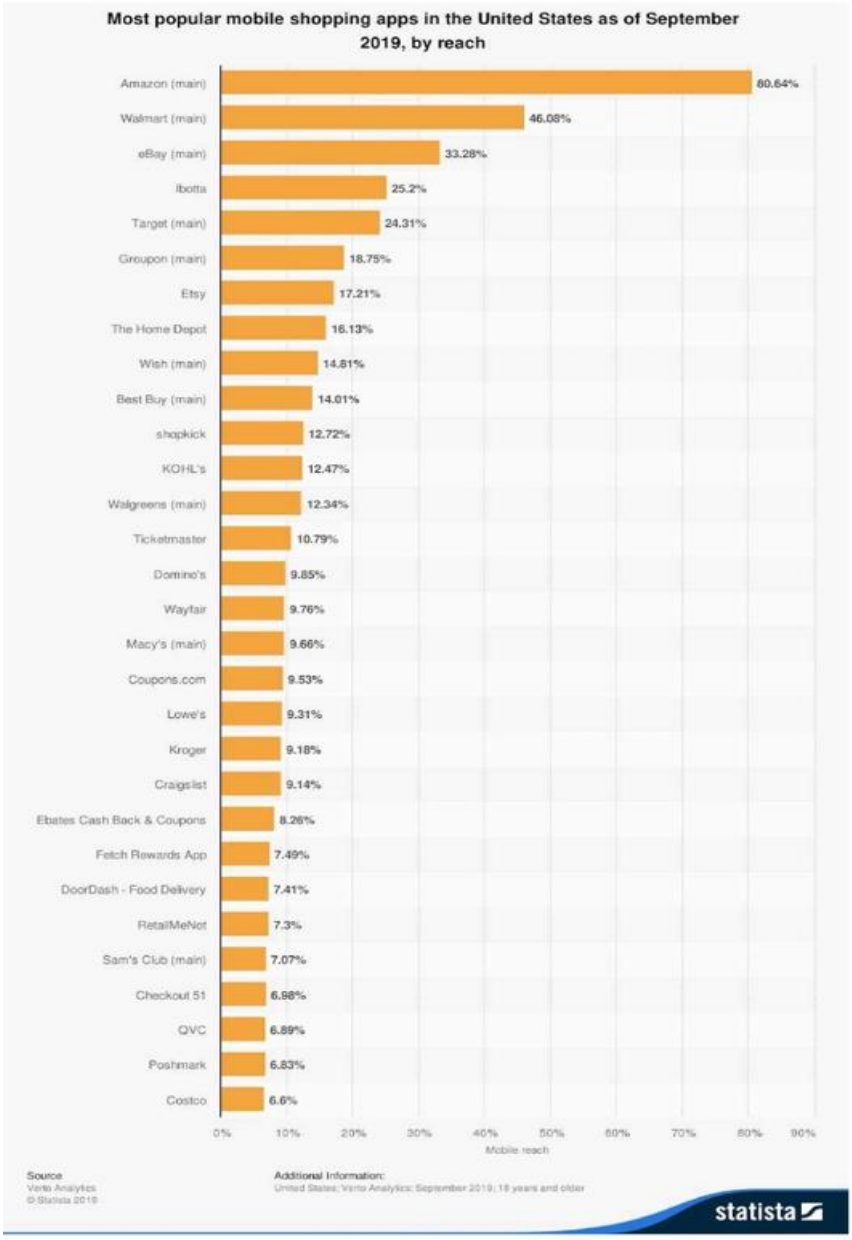


Figure 4: Major ecommerce apps

An online payment gateway will be utilized to integrate online payment options into the artefact. Dinarys (n.d) defines a payment gateway as a supplier of payment authorisation services. It is essentially a software application that connects a website to a bank, allowing customers to pay immediately without having to leave their apartment.

Anyone can utilize this type of service, including eCommerce websites, other service providers who want their customers to pay online, and even non-profit organizations who wish to collect money online.

Payment gateways in Trinidad and Tobago include but are not limited to: Wipay, Stripe, First Atlantic Commerce (FAC) and Paypal. Stripe will be used for its simplicity, ease of use and comprehensive documentation.

3.4.2 Predictive Analytics

Considered a way to tell the future, predictive analytics uses a combination of statistical modelling, machine learning, mathematical processes, and data mining techniques to make predictions about future outcomes. Predicting the future using historical data have proven to be lucrative as 97 percent of companies with revenues exceeding 100 million were found to use some form of business analytics, proffered Chen et al. (2012). Research (n.d) also states that global market value will be an estimated US$39.1 billion in 2028, up from US$8.12 billion in 2020. This growth represents a 20.7% increase in global value in a space of 8 years.

The basic steps that predictive analytics software commonly follow in the forecasting analytics process are listed below and are shown in Figure 5 below.:

1. Data collection: Import data from appropriate sources such spreadsheets and databases, and make sure these data sets are customized to the goal's requirements.
2. Data cleaning and combination: Remove any irrelevant information, check for missing data, and look for any irregularities that could cause the data to be erroneous. Depending on the desired outcome, combine appropriate data sources that would need to be gathered.
3. Developing the model: Establish the hypothesis and build the model using statistical approaches based on gathered data. Consider all variables and components, then test the

model with historical data to see which one produces the most accurate results and proves the premise.

4. Integration of analytics with systems: Implement the model in the operational system by incorporating it into software or devices.
5. Validation of the results: Monitor the model to assess if it is effective in producing the desired outcome. Various factors can cause changes over time, so it is ideal to monitor the model's performance and update them as needed.

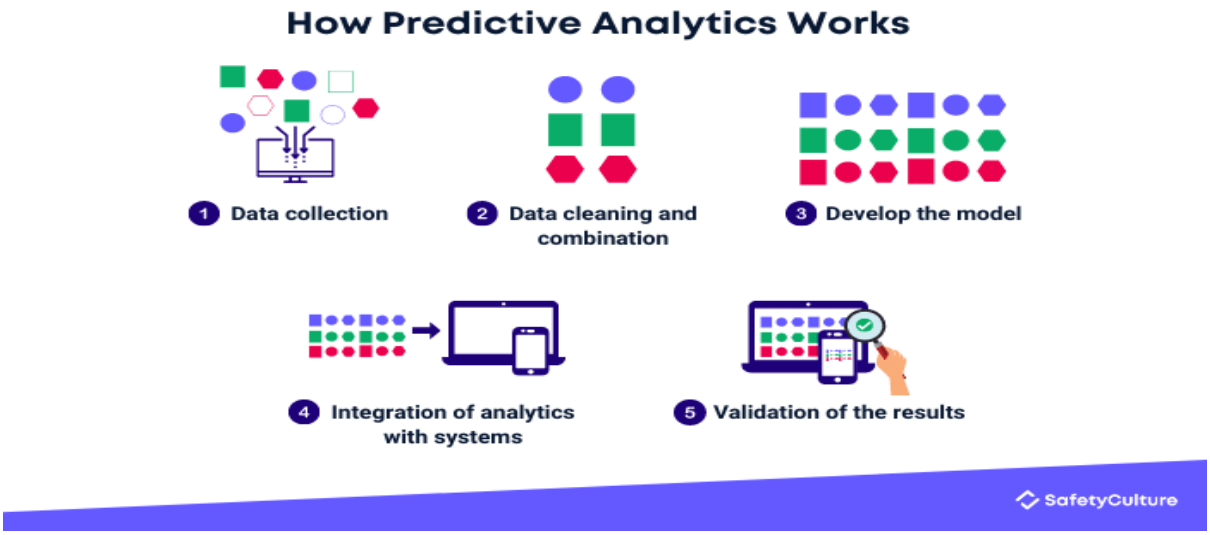


Figure 5: How predictive analytics work

Each search request, client response (or lack thereof), product complaint, buying decision, act of fraud and credit default is mined to generate predictive models. This data analytic tool help enterprises detect fraud, improve customer service, and enhance enterprise decision making.

1. Detect fraud: By recognizing trends and patterns in your data, predictive analytics can forecast possible areas of risk and provide predictions about how these risks may affect your business. Companies may identify and prioritize the most essential risks, analyse the possible impact, and decide on a course of action based on their severity by combining these analytics with a defined risk management methodology.
2. Improve customer service: Modern customers prioritize having a personalized experience and have no issue relinquishing their data. Predictive modelling helps to ensure customers

get the personalized experience they want by these models directly targeting them using data provided.

3. Enhance enterprise decision making: According to Siegel, E. (2015), standard business intelligence insights still require human intervention to complete decision making but predictive analytics is specifically designed to generate conclusive action imperatives. This results in faster, sometimes more accurate decision making.

Even with these significant advantages, predictive analytics still has a couple disadvantages. Some are itemized below:

1. Inherent inaccuracy in predictive modelling: Anyone who relies on predictive modelling should consider a model's potential for error. Even a well-designed model cannot always be 100 percent accurate, which can lead to considerable differences in forecasts and actual outcomes.
2. Costs: Implementing predictive analytical techniques require a significant investment in software and hardware.
3. The need for clean, accurate data: Poor record-keeping and a large number of legacy systems might affect the quality of the data input. Without clean data, the quality of predictions made would not be conducive to good decision making.

A good use-case to examine is Netflix. How does Netflix use predictive analytics? By collecting data from their 151 million members and using data analytics models to learn about customer behaviour and purchasing habits. Then, based on the preferences of their users, they recommend movies and TV series. Engati (n.d.) states that they are right 80% of the time.

In this project, the developer intends to bundle packages (up-selling) and provide suggested purchases using predictive analytics in conjunction with the recommender system (cross-selling).

3.4.3 Recommender Systems

As reported by Melville and Sindhwani (n.d.), the goal of a recommender system is to make meaningful suggestions for items or products that would be of interest to a group of users. According to Sarwar et al. (2000), millions of products are available for purchase on the top E-commerce sites. Consumers find it difficult to choose from so many possibilities. As a result of this issue, recommender systems have emerged to alleviate this problem.

Getting recommendations from reliable sources is an important part of the natural decision-making process, especially when faced with a seemingly endless list of options. Recommender systems help this process by predicting ratings a particular user might rate a specific item, creating a rank and finally returning that back to the end user.

It does this by modelling centric techniques like collaborative filtering and content-based filtering. Both are discussed below and are the difference then shown in Figure 6 below.:

1. Collaborative Filtering: Schafer et al. (2007) defines collaborative filtering as the practice of evaluating or filtering objects based on the opinions of others. The idea behind collaborative filtering is that if user A and B have similar tastes in one product, they are likely to have similar tastes in other products.
2. Content-Based Filtering: Content-based filtering uses historical data. Vatsal (2021) states that content-based filtering generates suggestions based on the user's profile and interests. Unlike the collaborative filtering model, which rely on ratings between the target user and other users, content-based models rely solely on ratings provided by the target user.

The difference between the two can be seen in Figure 6 below:

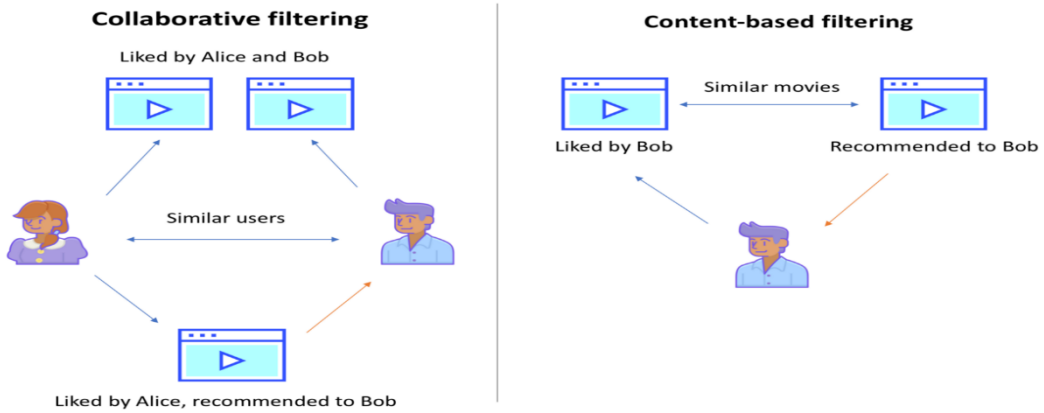


Figure 6: Collaborative versus Content-based filtering

Recommender systems not only assist in determining which items to provide to a specific customer, but they also enhance cross-sell by suggesting other products to customers, as well as improve customer loyalty by encouraging customers to return to the sites that best meet their needs. In terms of drawbacks, some of the most common roadblocks associated with recommender systems according to Melville and Sindhwani (n.d.), are sparsity and the cold-start problem.

1. Sparsity: Simply said, most users do not rate the majority of things, resulting in a sparse user ratings matrix. This is a concern for Collaborative Filtering systems since it reduces the chances of discovering a group of users who have comparable ratings.
2. Cold-start problem: Recommender systems face a huge issue when it comes to new items and consumers. To elaborate, a new item cannot be suggested unless it has been previously rated by another user. This problem affects not only new products, but also obscure items, which is especially problematic for individuals with diverse tastes.
3. Spotify will suggest tracks that are similar to those you have listened to or enjoyed before so you may keep visiting their platform to listen to music. Amazon uses recommendations to propose products to different consumers based on the information they have.

For this artefact, the developer intends to employ a recommender system to give product suggestions to users using collaborative filtering. Using their purchase history, recommendations will be given based on orders from other users who purchased the same item.

3.4.4 Encryption

A fundamental purpose of information technology is to secure data while it is in storage or in transit to prevent unwanted access by bad actors. If eCommerce activities are compromised, businesses lose billions of dollars in income and customer faith in eCommerce is eroded. Statista (n.d.), estimates that in 2021, global e-commerce losses due to online payment fraud were a little over US$20 billion.

Cloudfare UK (n.d.) defines encryption as a technique for encrypting data so that only authorized parties may decipher it. It is the process of transforming human-readable plaintext to incomprehensible text, also known as ciphertext, in technical terms. To put it another way, encryption modifies readable data to make it appear random.

A symmetric cryptosystem is the most fundamental method of encrypting data. This is where data is encrypted and decrypted using the same key. For eCommerce, something more complex is needed since sophisticated computers can break an encryption that has only one key. This is where Public Key Encryption (PKE), or asymmetric encryption, comes in handy. The inclusion of a second key, which makes a huge impact in terms of data integrity protection, was the main enhancement brought about by Public Key Encryption. Two keys are used in public key encryption: one is public, and the other is private. You cannot deduce the other key if you just have one. An outline of this in play can be seen in figure 7 below:

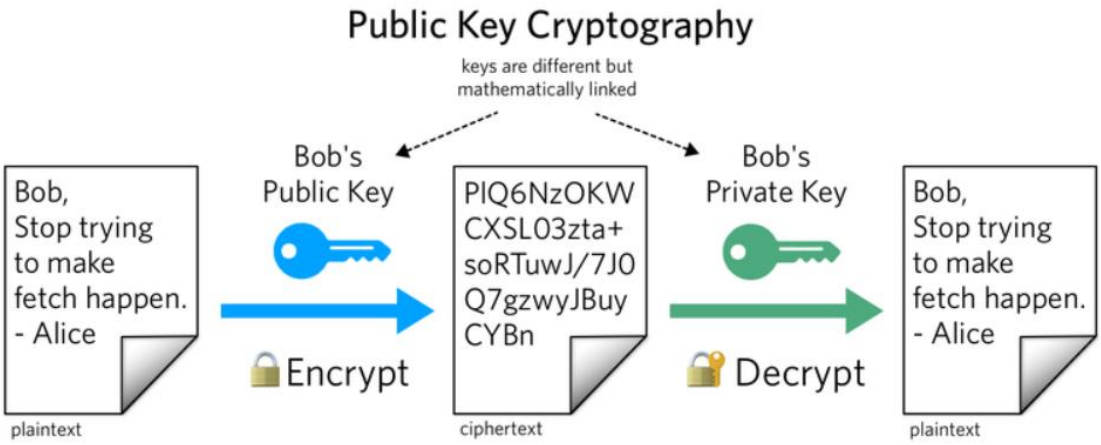


Figure 7: Public key cryptography

Data encryption is not only beneficial but also necessary for a number of reasons, some of which are outlined below:

1. Privacy: Encryption ensures that only the intended recipient or the rightful data owner may read messages or data in transit. This protects sensitive data from being intercepted and read by attackers, ad networks, Internet service providers, and, in some situations, governments.
2. Security: Whether the data is in transit or at rest, encryption helps prevent data breaches. If a business device is lost or stolen, the data on its hard disk will remain protected if its hard drive is securely encrypted. Encrypted communications, on the other hand, allow communicating parties to share sensitive material without it being leaked.
3. Data integrity: On-path attacks, can also be prevented by encryption. When data is sent over the Internet, encryption (together with other integrity safeguards) ensures that the information received by the recipient has not been tampered with along the way.

To limit the risk of exposure, realize these benefits and increase the website's reputation and security, this eCommerce application will offer password and credit card encryption.

3.4.5 Artificial Intelligence (AI)

Artificial Intelligence (AI) has become a ubiquitous technology. From grocery self-checkouts, security systems at airports and as it relates to this artefact, chatbots - AI, once a science-fiction thought experiment has become a reality. Because of AI's potential, it is regarded as the fourth industrial revolution (Vanneschi et al., 2018). Builtin (2022) defines AI as a broad field of computer science concerned with developing intelligent machines that can accomplish activities that would normally need human intelligence.

As indicated, the AI application of chatbots will be used in this artefact. Simply put, a chatbot is a computer application that simulates human-to-human conversation. Tuomas (2021) explains that a few years ago, chatbots were simply built to answer to customers with normal and expected responses. However, with the advancement of AI, chatbots can now provide potential clients with responses and support tailored to their specific wants and expectations.

It works using a simple mechanism as seen in figure 8 below.

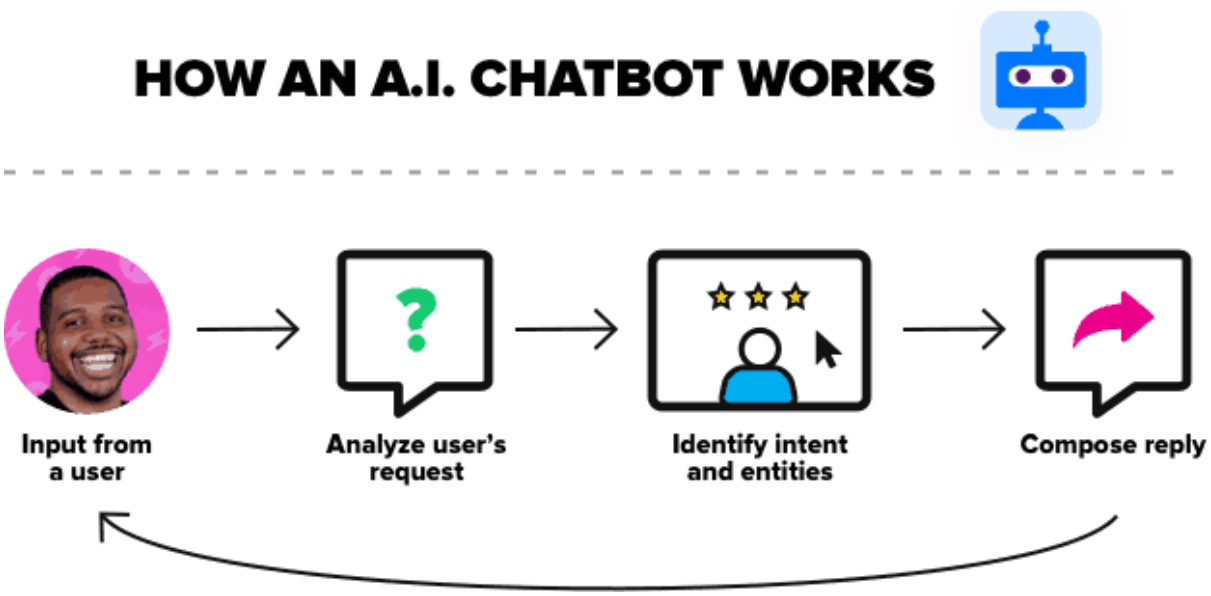


Figure 8: How an AI chatbot works

Chatbots replaces the need for a human to interact with customers. By taking this away, some benefits are realized:

1. 24/7 support: Customers nowadays expect businesses to remain open 24 hours a day, seven days a week. While having a 24-hour customer support personnel is a (expensive) option, with chatbots you can reduce that cost while still ensuring that your clients are served instantly — regardless of the time of day. Customer satisfaction can be improved by providing 24-hour support for queries or general issues.
2. Time and cost savings: Using chatbots to handle the majority (if not all) of your customer support operations will save you a lot of money on your customer service crew. Chatbots provide efficient customer service with less human intervention, allowing you to focus on more important areas of your e-commerce site, such as page layout and checkout. With little resource expenses, you may also substantially reduce human error and enable effective customer support.

Some disadvantages that cause concern with the AI application of chatbots are:

1. Lack of empathy: Bots will not respond to clients with personality or emotion, which is a major turnoff for many customers. Customers want to be treated with empathy, especially when they are having problems with a product or company. If a conversation deviates from a predetermined path, the bot is unable to improvise and lacks human touch, resulting in a poor customer experience.

2. One of the major drawbacks of chatbots is that they can't always help when a customer's question isn't clear or too precise. Chatbots are programmed to respond to general questions with answers that can be found in the bot's database, so if a customer asks a question that isn't on that list, the bot will likely be confused and will either go around in circles trying to understand the question or will be left without an answer. In either instance, this is a poor client experience that can harm your company's reputation.

Even with a couple disadvantages, the feasibility is still enough to justify the need for a chatbot. As evidenced by Amazon's application of AI via Alexa - Amazon's AI-centric, cloud-based chatbot, AI can be lucrative. Kumar and Trakru (n.d.) proffers that Amazon enjoyed a 35% increase in sales by using Alexa's data sets for their targeted marketing algorithms.

For this project, a chatbot will be implemented in an attempt to realize as much benefits of AI as possible.

3.4.6 Checkout Flow

Figure 9 below identifies top reasons carts are abandoned while customers are online shopping from WP Swings (2022).

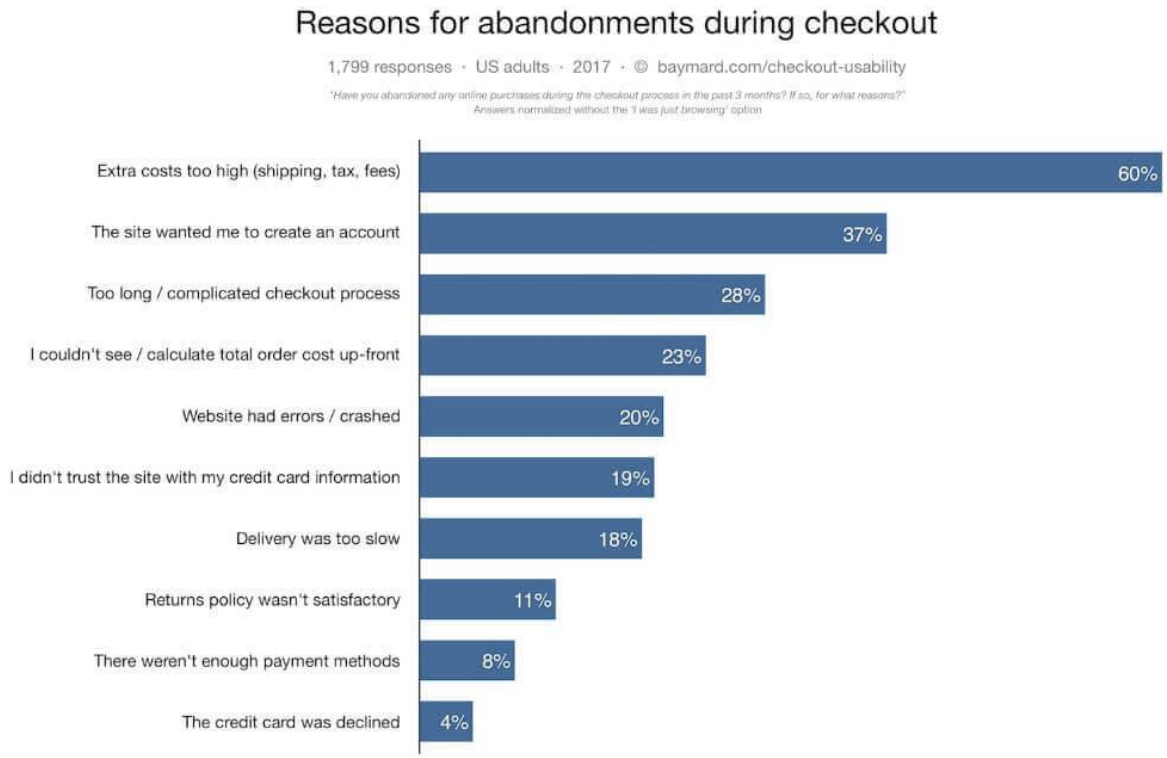


Figure 9: Reasons for cart abandonment

A lengthy, confusing, or difficult checkout process during purchase accounts for about 28% of cart abandonment, according to the above graphic from WP Swings (2022). Checkout plays a significant role in the overall shopping experience on an eCommerce site and needs to be designed in a way to encourage conversions as opposed to cart abandonment as it is the last step in the eCommerce experience.

Belk et al. (2015) categorizes checkout designs into 2 broad categories: (i) a single one-page checkout procedure that contains all the information needed to complete the transaction on one page and (ii) a guided step-by-step checkout process in which customers must fill out their information on various pages in multiple steps.

Good checkout methods can result in improved conversion rates, increased trust, and lower cart abandonment, all of which can contribute to increased revenue while a bad checkout methods will have the opposite effect. Optimizing the checkout process is a simple method to ensure that site visitors become happy, paying clients.

In this artefact, the checkout page will be designed to reduce the number of hoops customers will have to jump through to order their products while ensuring that all required elements for purchase are there. In reviewing some of the most popular eCommerce checkout pages, they all have one thing in common: simplicity. Therefore, this artefact's checkout page will be designed with this, User Experience Design elements and transparency in mind to ensure conversion.

It is at this point that the project-related decisions are discussed. These are the other project-related topics, such as the software used to build the application, the development methodology chosen, user experience design, and referencing, to mention a few. The following project-related options were found to be the most useful for this project.

3.5 Project Related Choices

3.5.1 Software Development Tools

The tools used in the software development process have the potential to make or break a project. After deciding on the target environment and programming language(s), as well as the requirements and end goals, the next step in beginning a software development project is to select the tools that will be utilized throughout the process. It is critical to understand the different sorts of tools available, the benefits they can give, and the consequences of using them. A number of tools are available to help in the software development process.

Allaboutcircuits (n.d.) defines an Integrated Development Environment (IDE) as a piece of software that allows you to develop and debug software in a streamlined manner. Rather than executing all of the procedures required to create an executable program as separate activities, it integrates all of the necessary tools into a single application and workspace. Each tool is aware of its surroundings, and they work together to provide a seamless development environment for the developer.

WAMP and XAMPP were considered to develop the artefact but ultimately, XAMPP was chosen because of its ease of use, user-friendly interface and cross-functionality.

3.5.2 Version Control

A version control system (VCS) is software that helps track and control changes to a software artefact's source code. It maintains a development history, allowing for code review and even rollback to a previous version. As new features are added to your program, using a version control solution will help keep the project organized and create backups. Git, Mercurial, TFS, and SVN are all popular VCSs. Many IDEs and code editors, including Visual Studio, support Git.

A version control/code hosting server will be necessary for the Undergraduate Project to push changes to the code to or download the most recent version of the code from. Github and Bitbucket was considered and although Bitbucket is more flexible than Github, Github was decided upon because it is open-sourced.

3.5.3 User Experience Design (UXD)

User Experience Design (UXD or UED) is a design process whose main goal is to create a system that provides a positive user experience. As a result, UXD encompasses theories from a variety of disciplines, including user interface design, usability, accessibility, information architecture, and Human Computer Interaction (HCI). Good UXD will attract new visitors to the eCommerce store, boost conversion rates, ensure customer retention and increase revenue.

These reasons make good UXD an essential element for this project. It will help to show site visitors that customer care is a big priority. UXD best practices will thus be used when designing this artefact such as:

1. Responsive and mobile-friendly design
2. Simplicity
3. Use of white space
4. Colour psychology

3.6 Summary

90% of eCommerce artefacts fail within the first four months of operation. This is according to ICSID (2021). Therefore, the goal is to give this artefact a chance to be in the 10% that succeed by studying the ones that did and following their example. It has been observed that successful eCommerce businesses implement one or more of the below:

1. Online payment methods: by adding the ability for visitors to complete the purchase on the artefact itself, conversion rates, efficiency and having a global reach would be ensured.
2. Predictive analytics: having a tool that helps with forecasting future events will assist with fraud detection, improve customer service and enhance enterprise decision-making.
3. Recommender systems: a recommendation tool for visiting customers will help to not only improve customer loyalty but also will enhance cross-selling.
4. Encryption: a secure artefact will serve to protect customers' data, ensure privacy and keep the integrity of all data intact.
5. Checkout flow: a good checkout cart design will lower cart abandonment, improve sales and conversion rates, and thereby contribute to increased revenue.
6. Ai/chatbot: a chatbot will provide 24/7 customer support, 24/7 customer service and significant time & cost savings.

3.7 Recommendations

The project's main goal is to create an eCommerce application that improves the user experience. The following recommendations will be implemented in the project based on the literature review:

1. Online payment methods: The payment method used will be Stripe because of its simplicity, ease of use and comprehensive documentation.
2. Predictive analytics: Bundles (up-selling) and suggested purchases (cross-selling) would be developed using predictive analytics in conjunction with the recommender system, with the goal of increasing sales.
3. Recommender systems: The developer intends to employ a recommender system to give product suggestions to users using collaborative filtering. Using their purchase history, recommendations will be given based on orders from other users who purchased the same item.

4. Encryption: This eCommerce artefact will provide password and credit card encryption to reduce the danger of exposure and improve the website's reputation and security.
5. Checkout flow: To ensure conversion, the checkout page for this item will be developed with a focus on simplicity, User Experience Design (UXD) aspects, and transparency in mind.
6. AI/chatbot: Frequently asked questions (FAQs) will be used to develop the chatbot feature.

CHAPTER 4: REQUIREMENTS ANALYSIS

4.1 Introduction

Chakraborty (2012) describes the requirements analysis stage as the most vital part of the development process as it is usually the foundation upon which the rest of the development stages are built on. The importance of gathering requirements cannot be overstated since an inability to identify problems in this stage can be costly later on, both in time and energy expended. Therefore, successful completion of this key early milestone helps to ensure the project has a good chance of success.

One of the main goals of the project is to deliver a responsive, intuitive, user-friendly artefact that has enough variety for the user to make an informed buying decision. With Rapid Application Development (RAD), requirements were initially broad in scope for development to start early. These requirements have since been fleshed out and refined for the creation of a detailed and justified list of requirements.

This chapter will specify the system's functional and non-functional requirements and illustrate the documentation that define and qualify the project's scope such as personas, a use case diagram, site map and activity diagram.

4.2 The Process of Producing the Requirements

In small agile-based projects like this one, Kavitha and Thomas (2011) outline numerous techniques for capturing system requirements that entail ongoing communication via methods including surveys, interviews, and observation, to mention a few. Regrettably, these techniques need a separate university ethical approval process and in the interest of saving time, other techniques

needed to be used. Techniques used were analysing similar websites, conducting in-depth background research and reviewing unit and assignment brief documentation.

Firstly, the analysis of a number of websites, some of which the author is aware of having dealt with and are well-known in the related sector of the sale of computer hardware and accessories, were done. Amazon, Ebay, Alibaba, Newegg, and Best Buy were amongst them. This analysis was the cornerstone for some core and advanced requirements such as account management and the chatbot.

Secondly, background research done using peer-reviewed sources into emerging technologies employed in the e-commerce industry played a significant role in the requirements gathering process. Research into features such as encryption, recommender systems, Artificial Intelligence (AI) and checkout flow not only gave a deeper understanding of the role they will play in the artefact but also emphasized how integral they are as requirements.

Lastly, reviewing the unit literature provided guidelines which assisted the requirement gathering process since the assignment briefs contain the scope and broad expectations of the artefact and is thus a good initial springboard to get started. Reviewing these greatly increased understanding of what is expected from the university for a successful outcome. The initial core requirements were gathered from this such as log in/log out, registration, product detail and product listing.

4.3 Personas

Matthews and Whittaker (2012) define personas as a fictitious archetype of a legitimate user that describes the traits, interests, goals and frustrations of that individual.

In order to justify some of the functionality choices made during the above requirement gathering techniques, personas were established as a way to fully understand the goals and frustrations that

affect potential users that will interact with the system. Doing this will ensure that customer frustrations will not get neglected and thereby encourage customer retention.

In order to be ready for the various needs of prospective customers, a couple of these established personas are presented below in Figure 10 and Figure 11:

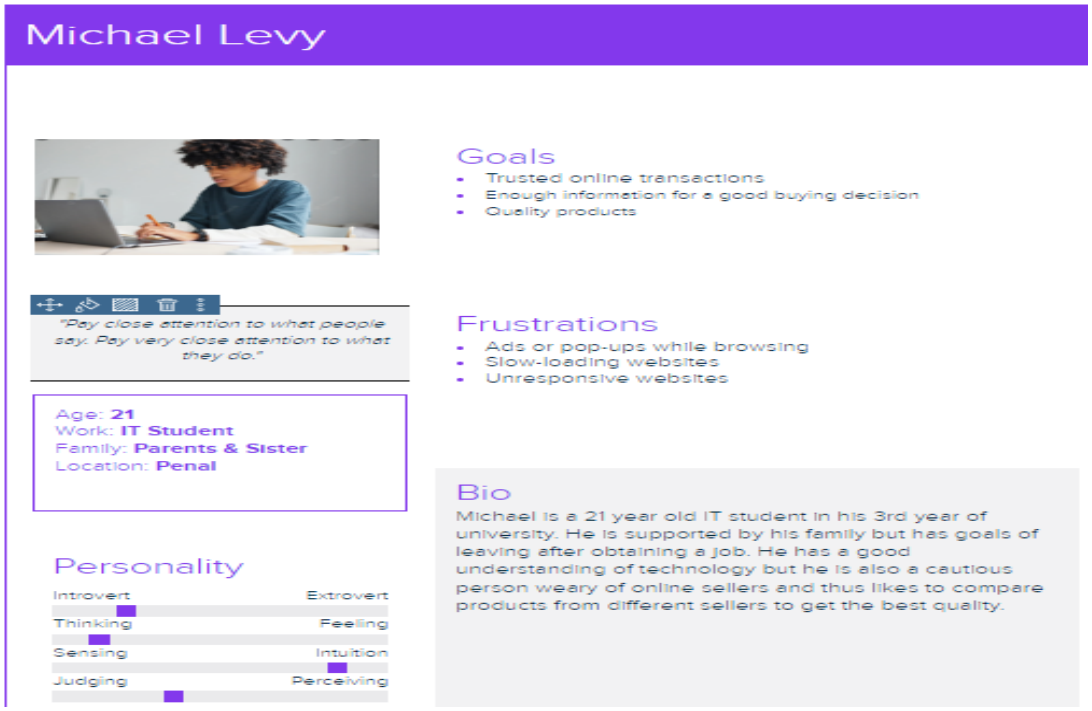


Figure 10: Persona 1 – Michael Levy

Creating this persona justified the use of the product details, password encryption and a secure online payment method requirements. The product details page for each product will assist Michael in making the best buying decision while having a secure online payment method will appease his trust concerns with conducting online transanctions.

Michael Levy's persona also justifies the need for responsive deisgn since he is frustrated with unresponsive websites. Because he has a good grasp of technology, he will appreciate the care and effort taken to build a responsive site which will further endear him to conduct purchases.

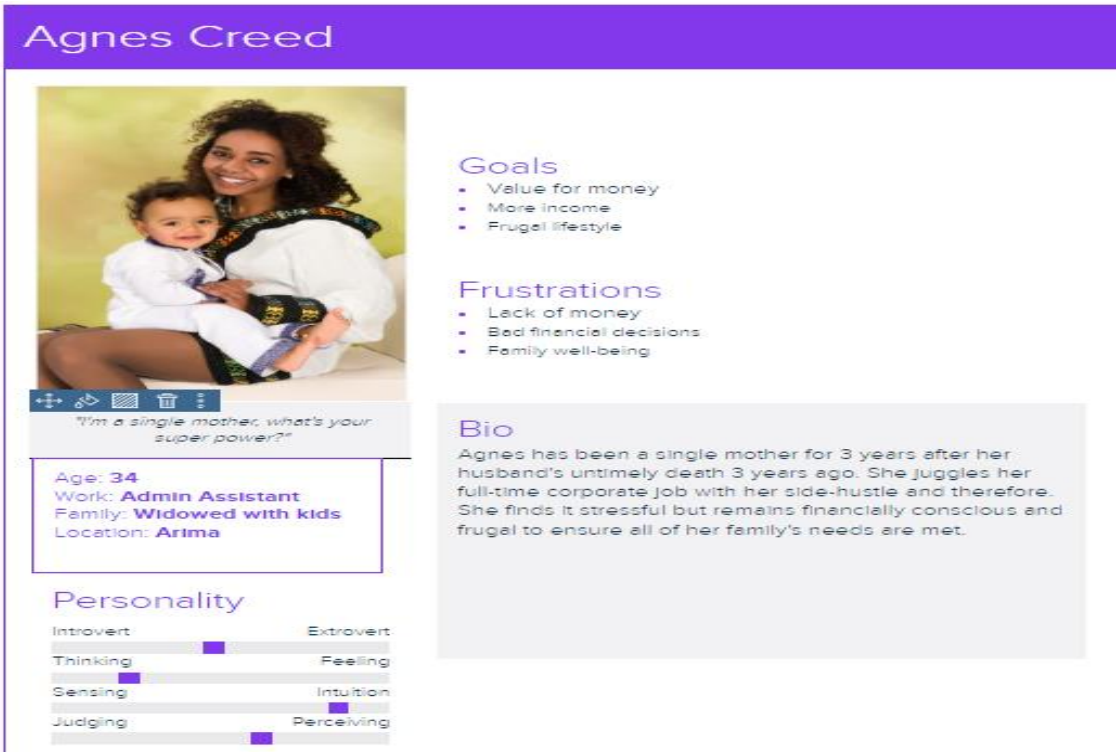


Figure 11: Persona 2 – Agnes Creed

Agnes' persona will appreciate the search, sort, filter feature which will allow her to filter by price since value for her money is a major concern. Agnes will be shopping online mainly in an effort to save time being a single parent with little time to spare, therefore having a further time-saving element such as the search, sort, filter element will benefit her a great deal. Further to this, there is a probability Agnes might experience technical overwhelm not having the disposable time to research computer hardware products so the informational pages will be of use here.

4.4 Functional and Non-Functional Requirements

The requirements analysis stage is the first major initial stage of the project since it establishes what users want and anticipate from a new product. These requirements were obtained during background research done in the literature review, reviewing relevant websites and the unit literature as stated above. They were then refined to determine the most essential requirements for producing a suitable eCommerce application.

The requirements for the artefact were split into functional and non-functional categories. These are explained below.

4.4.1 Functional Requirements:

These are the features the user may interact with and are further divided into Core and Advanced Requirements.

a. Core Requirements:

These are essential and offer the very minimal capabilities needed for a website to be usable, such as a product display page, search capability, and an add-to-cart button.

b. Advanced Requirements

These are the requirements which are typically 'nice to haves' and tend to be more complex features. Features such as recommender systems, a chatbot and encryption. These features increase the functionality of the solution and boost user engagement and enjoyment.

4.4.2 Non-functional Requirements:

These features should satisfy the urgent requirements and demands specified by the user during the requirements gathering process and have an impact on the application's usability, design, security, and browser compatibility.

Table 2 below shows a list of the Functional and Non-Functional Requirements:

Name	Requirement	Core/Advanced	Who Can Access?
FR1	Register/create an account	Core	Guest
FR2	Log in/Log out	Core	Registered user
FR3	Order history/order details	Core	Registered user
FR4	Cart management – Add to cart, edit cart, remove from cart	Core	Registered user
FR5	Product detail	Core	Guest
FR6	Product listing/store catalogue	Core	Guest
FR7	Checkout flow	Core	Registered user
FR8	Search, sort, filter	Core	Public
FR9	Informational Pages	Core	Public
FR10	Online payment method	Advanced	Registered user
FR11	Recommender systems/Predictive Analytics	Advanced	Registered user
FR12	Chatbot Integration	Advanced	Public
FR13	Product Inventory Management	Advanced	Administrator
NFR1	Security – Password Encryption	Advanced	Registered user
NFR2	Usability – Responsive design	Advanced	Public
NFR3	Data Integrity – Database security	Advanced	Registered user
NFR4	Scalability	Advanced	Public
NFR5	Maintainability	Advanced	Public

Table 2: Functional and non-functional requirements

Each functional requirement was further refined to provide both a detailed explanation of how it functions and a rationale or justification for the requirement. Both the database design and the user interface design would benefit from this.

Table 3 below shows an example of the 'FR1 – Register/Create an Account' requirement and its rationale and Appendix B shows the complete list, both functional and non-functional.

Requirement	Register/Create an account
Number	FR1
Description	The user must register on the website before they can make a transaction. The user must input their first and last names, email address, and password to proceed. A record will be saved in the database in the "users" table once the data has been verified. The user is subsequently informed via a notification that his registration was successful.
Rationale	A user who creates an account on the application gets access to the following features: a user may make secure purchases and examine their order history and respective order details.

Table 3: Sample requirements detail of Core Functional Requirement – Register/Create an Account

4.5 Use Case

Another form of documentation used to help with requirements gathering is a use case diagram, otherwise known as a UML diagram. According to Edrawsoft (2021), Use Case Diagrams are a part of the Unified Modeling Language (UML) family format, which is currently widely used in the business field to help display illustrated visualizations of a piece of a system's design. These diagrams are usually developed during the early stages of a project and are represented by four entities (Visual-Paradigm, 2021):

Actor: This person, represented by a stick figure, interacts with the system function and typically has (use case) obligations in the form of inputs and expectations in the form of outputs.

Use Case: Each actor must be connected to a case for this to operate, which is shown as an oval shape.

Link/Association/Relationship: This component, which is a plain line, demonstrates the connections between Actors and Use Cases to demonstrate communication routes.

System: This is shown as a rectangle, and it refers to the complete system as described in the requirements document, acting as the perimeter for all use cases particular to the business functions.

Figure 12 below shows the major elements of a typical Use Case diagram:

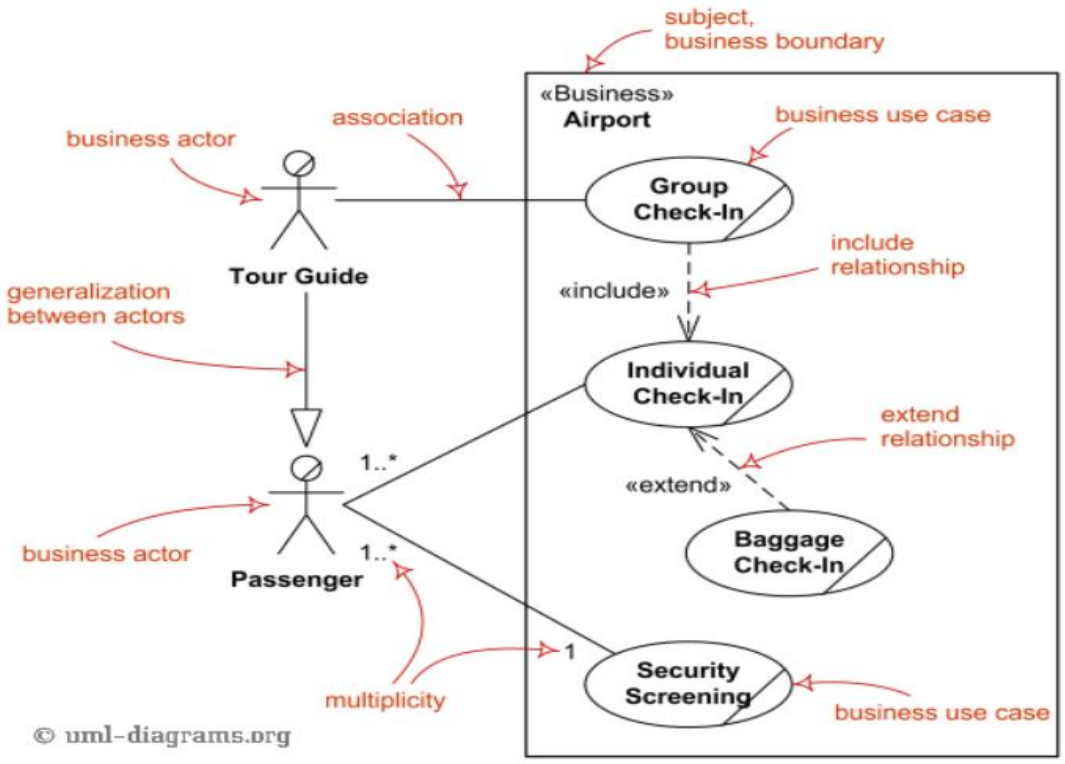


Figure 12: Major elements of a Use Case Diagram

The UML diagram below in Figure 13 illustrates two types of use cases used for the artefact, that of admin and user. First off, it shows a couple different ways a new and registered user can interact with the products in the system.

A new user can only register and view products while a registered user can view products and make purchases. The administrator, on the other hand, can view products and also add products.

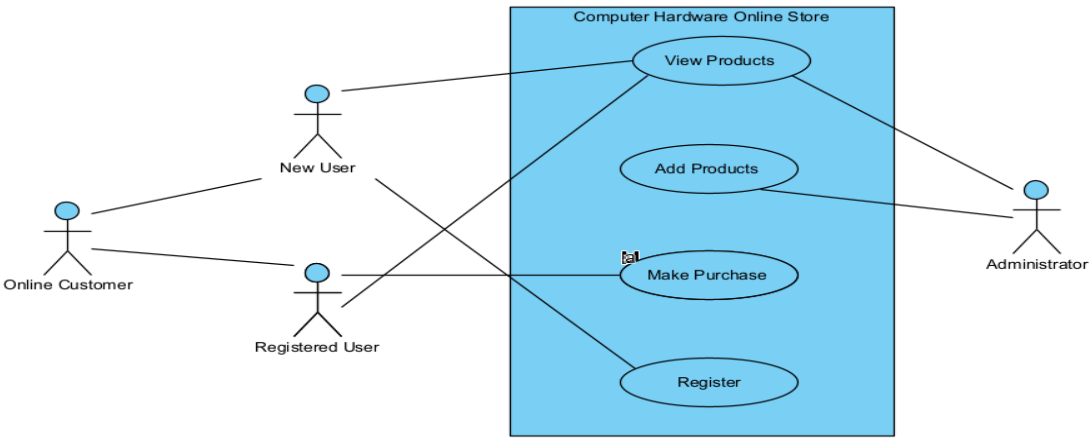


Figure 13: Use Case Diagram for artefact

4.6 Information Architecture

To make the artefact as intuitive as possible, all content need to be mapped out, structured and organized to make it as easy as possible for new and past users to navigate and avoid any confusion. Users need to be able to get around - whether they are aware of where they are going or not - as quickly and efficiently as possible. This is where information architecture (IA) or the use of a site map comes in.

As stated by Ruzza et al (2017), the IA of a website significantly influences its usability and the user experience during navigation. The IA structure that was created for this web application is depicted in Figure 14 below:

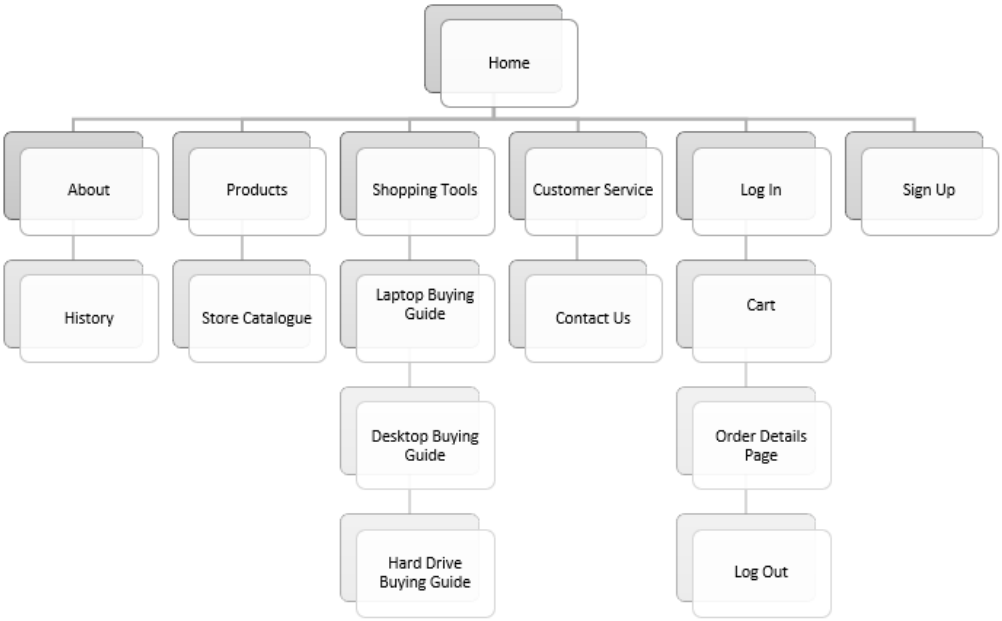


Figure 14: Information Architecture for artefact

4.7 Activity Diagram

Activity Diagrams take the Use Case diagram created above a step further by illuminating the workflow that visitors will encounter on the site as they move from one activity to the next. An activity diagram, according to Walker (2019), is a flowchart that shows the user's progression from one activity (a system action) to the next until the task is finished.

An example of an activity diagram for the artefact can be seen in Figure 15 below. It details the “make purchase” workflow the registered use can do as illustrated in the Use Case diagram. Firstly, the customer adds their product to the cart, proceeds to checkout where the payment is accepted or rejected. If accepted, the payment completes, the order is delivered and the order closes. If rejected, the order will close.

Doing activity diagrams like this helps with identifying weak areas in the artefact that may have been overlooked previously and ensures remedial measures are put in place when implementing.

For example, the “order rejected” decision might have been an overlooked possibility in the checkout process especially using agile as a methodology since agile prioritizes quick development. Therefore, it might be overlooked because of haste. Activity diagrams such as this assists with development by ensuring a method is placed to handle an event such as that when implementing the cart and checkout functionality as opposed to speedily implementing and thereby wasting time revisiting already completed functionalities.

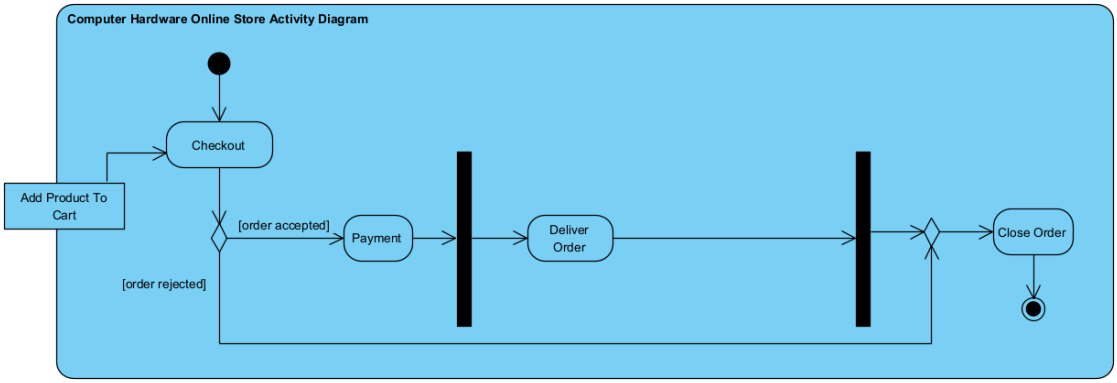


Figure 15: Activity Diagram for artefact

CHAPTER 5: DESIGN

5.1 Introduction

According to Nielsen (2008), having a well-thought design makes your software easier to implement, reduces a need for major changes later and saves you from headaches down the line. Therefore, in an effort to realize these benefits while attempting to encourage customer retention and increased sales, particular attention was placed on the design of the application.

This chapter outlines the approach taken to design both the back-end and front-end of the artefact separately as they each have different needs for the required level of development satisfaction. For the back-end, we will look at the steps taken to create the database which will store the necessary data. On the other hand, for the front-end, we will look at not only the elements of design for the user interface but also HCI concerns and how they may have impacted the design.

5.2 Database Design (Back End Design)

Designing the database for the artefact correctly is of tantamount importance since database design mistakes in an operational database is often impossible to correct (Peinl, 2020). To ensure proper design, the database would have to be normalized to at least the third normal form to avoid duplications of the data (redundancy) which wastes storage space, ensure efficient data retrieval and maintain the integrity of the data stored.

The top down approach as opposed to the bottom up approach was used to design the database because of the level of complexity. This approach would break down the project's goals into smaller problems that are more easily solved as advised by Maxey (2012). The bottom up approach takes

the opposite angle and focuses on working backwards through the system to determine what data should be stored in the database (Burleson, nd). Burleson goes on to state that for the bottom up approach, the designer will need to assess every one of the interfaces that the database has, checking tables, relationships and views.

The top-down approach for data modeling is a three-part process seen below in Figure 16 below:

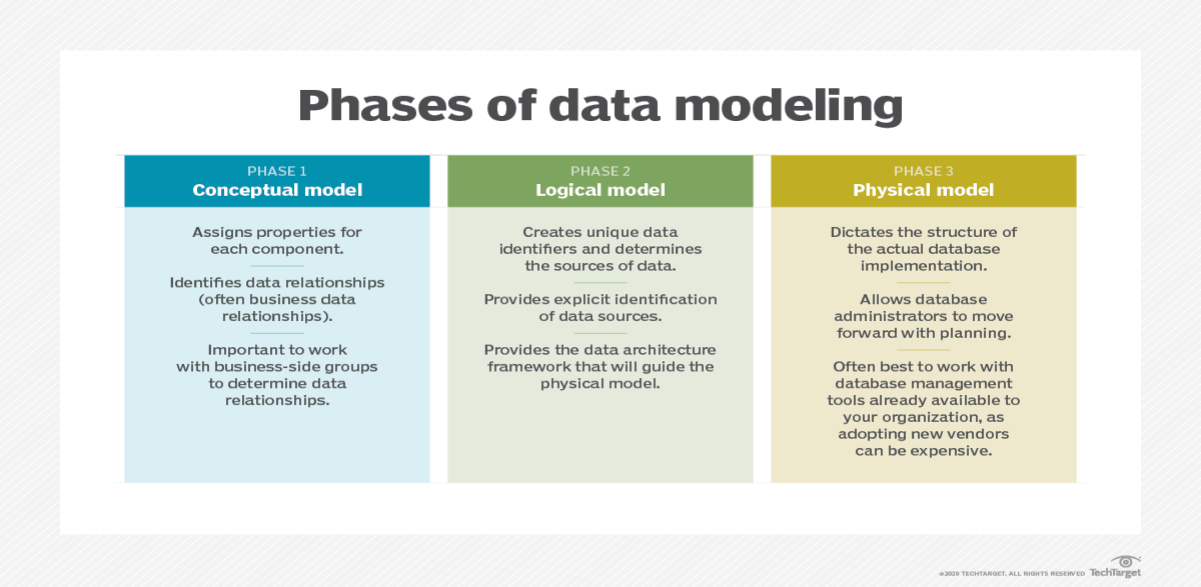


Figure 16: Data modeling phases in the top-down approach

Each step involves progressive refinement of the data for artefact and will be described in the sections below. The Creation of a conceptual Entity Relationship Diagram (ERD) is the first step, and is covered in the section that follows.

5.2.1 ERD

Using entities derived from the requirements analysis stage, a conceptual Entity Relationship Diagram (ERD) was created. Users, Orders, Products are all examples of these entities.

The relationships and cardinalities between the entities were determined after listing these entities and the basic attributes that would be kept in each. The relationships were all binary relationships (between two entities). An example of a relationship between entities would be:

An ORDER can contain many PRODUCTS; A PRODUCT can be included in many ORDERS

When there existed a many-to-many link between the entities, a new entity was made to break the many-to-many connection. The inclusion of an entity for ORDER DETAILS is one illustration of this. a M:M relationship was observed between the ORDER and PRODUCT entities so to remove this, an ORDER DETAILS entity had to be included to establish the necessary 1:M relationships.

The ERD below in figure 17 displays the remaining entities.

Figure 17: ERD for artefact

5.2.2 Normalization Process

The removal of the many-to-many linkage between the ORDER and PRODUCT entity is part of the process known as "normalization," which reduces redundancy and ensures that the data in data dependencies make logical sense.

Microsoft (nd) describes normalization as the process of organizing data in a database. They go on to state that this includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency. This is where primary and foreign keys are determined and unnecessary attributes removed.

In the top down approach, this is necessary to convert the above conceptual model of the ER diagram to a logical model. Before doing this however, the logical schema described below will act as a point of reference for the entities, attributes and their respective keys.

5.2.3 Logical Model

After creating the conceptual ERD, the entities and cardinalities were the used to refine all of the logical limitations that must be applied to the stored data. Entities, attributes and their respective keys were defined here and finally stored in a logical schema to be used to complete the logical model of the database. Five (5) tables were built in the logical schema since there were five (5) entities in the ERD. The tables' attributes were created from the attributes of the entities.

The logical schema table can be seen below in Table 3:

Entity	Attributes
Users	user_id*, first_name, last_name, email, password, user_created, user_type
Orders	order_id*, user_id*, subtotal, total, total_discount_amount, order_created, payment, payment_id
Order Details	order_id*, user_id*, subtotal, total, total_discount_amount, order_created, payment, payment_id
Cart	cart_id*, user_id*, product_id*, cart_created, cart_quantity, cart_status
Products	product_id*, product_title, product_description, product_price, product_discount_amount, product_quantity, product_image, product_created, product_status, product_category

Table 3: Logical schema

5.2.4 Data Dictionary

The last step before the creation of the physical database in the database management program is the creation of the data dictionary. The logical schema goes on to be refined even more into a data dictionary. While the logical schema acts as a reference point for the breakdown of the conceptual model of the database to a logical model, the data dictionary acts as a reference point for the breakdown of the logical model into the physical model. The physical model is the working, functional database.

Data dictionaries are used to provide detailed information about the contents of a dataset or database, such as the names of measured variables, their data types or formats, and text descriptions (USDA, nd).

An example of the data dictionary for the User entity can be seen below in Table 4 while the entire data dictionary can be seen in Appendix C:

Attribute	Data Type	Field Length	Constraint	Description
user_id	integer	11	Primary key	User ID of the user; auto increment
first_name	varchar	100	Not null	First name of user
last_name	varchar	100	Not null	Last name of user
email	varchar	100	Unique and must contain the '@' symbol	Email address of user
password	varchar	300	Not null	Password of the user for logging in
user_created	date	19	Current_timestamp()	Date the user created account
user_type	varchar	20	Either contain 'user' or 'admin'	Type of user account

Table 4: Data dictionary for User entity

5.3 Application Design (Front End Design)

The role of application design cannot be ignored as one of the goals of the artefact is to ensure customer retention and encourage repeat buying. To achieve this, a methodical approach to the artefact's design was absolutely necessary.

By utilizing 'user centred design' (UCD) approaches to design the artefact, the author hopes to make the artefact as user-friendly and intuitive as possible to achieve the above goal. The Interaction Design Foundation (2019) defines UCD as a broad term to describe design processes in which end-users influence how a design take place. To avoid the process of attaining ethics approval from the university, the author will assume the role of the user. This chapter outlines and details the UCD approach undertaken to design the front-end. Namely, this was done using sketches, wireframes, interactive prototypes and verifying all pages comply with Nielsen's 10 heuristic design principles.

5.3.1 Nielsen's Heuristics

Generally speaking, heuristics are a practical approach to a complex, mentally intensive problem. Design heuristics, specifically, is a process where experts use rules of thumb to measure the usability of user interfaces in independent walkthroughs and report issues as described by the Interaction Design Foundation (2019). Heuristic evaluations are meant to be cheaper and faster than usability studies.

Jakob Nielsen's 10 heuristic design principles are the most-used principles. He started development of these principles alongside Rolf Molich in 1990 and released the final set used today in 1994 (uxness, nd) seen below in Table 5. The design of the artefact was done in accordance with Nielsen's ten principles. These will also serve as the foundation for evaluating the software artefact in the future.

PRINCIPLE	DESCRIPTION
Visibility of system status	Users should constantly be kept up to date on what is happening by the system through suitable feedback sent in a timely manner.
Match between system and the real world	Instead of using system-oriented jargon, the system should employ words, phrases, and concepts that are recognizable to the consumers. Ensure that information appears in a logical and natural arrangement by adhering to real-world norms.
User control and freedom	Users frequently choose system options by accident, and they will require an obvious "emergency exit" to get out of the undesirable condition without having to engage in a lengthy conversation. Support undo and redo.
Consistency and standards	Users should not have to speculate if certain expressions, circumstances, or actions are equivalent. Observe platform customs.
Error prevention	Good error messages are great, but smart design that avoids an issue from happening in the first place is much better. Eliminate error-prone situations or check for them and provide consumers the chance to affirm before committing to an activity.
Recognition rather than recall	Reduce the amount of memory required from the user by making objects, actions, and options visible. It shouldn't be necessary for the user to recall details from one section of the dialogue to the next. When necessary, instructions for using the system should be readily visible or accessible.
Flexibility and efficiency of use	In order for the system to serve both beginner and experienced users, accelerators—unseen by the novice user—may often speed up the interaction for the expert user. Permit users to customize routine tasks.
Aesthetic and minimalist design	Information that is unnecessary or infrequently used should not be included in dialogues. Each additional piece of information in a conversation competes with the pertinent pieces and reduces their relative exposure.
Help users recognize, diagnose, and recover from errors	Error messages ought to be written in simple English (without the use of codes), identify the issue clearly, and offer a suitable alternative.
Help and documentation	Although it is preferable if users can operate the system without assistance, assistance and documentation may still be required. Any such material ought to be concise, focused on the user's goal, easy to search, and describe specific actions that need to be taken.

Table 5: Nielsen’s Heuristics

In Table 6 below is an example of how the first principle will be applied in the artefact while

Appendix D contains the full list of principles.

Principle 1: Visibility of systems status	
Description	Users should constantly be kept up to date on what is happening by the system through suitable feedback sent in a timely manner.
Implementation	All screens will include a name in the site header. Design will be consistent throughout the entire artefact.

Table 6: Implementation of Heuristic Principle# 1 – Visibility of system status

5.3.2 Storyboard

While we think of sketches as rough drawings, Design Thinking @ Haas (2011) describes sketching as a process that progresses from crude sketches (the ideation stage) through prototypes along a continuum he refers to as the "design funnel" (usability stage). Essentially, sketches are meant to help one first explore design ideas and save time that might be wasted by rushing too quickly into the more difficult design phases such as color and typography decisions.

For the artefact, a number of rough sketches were done to refine initial design ideas. An example of a sketch can be seen below in Figure 18 below.

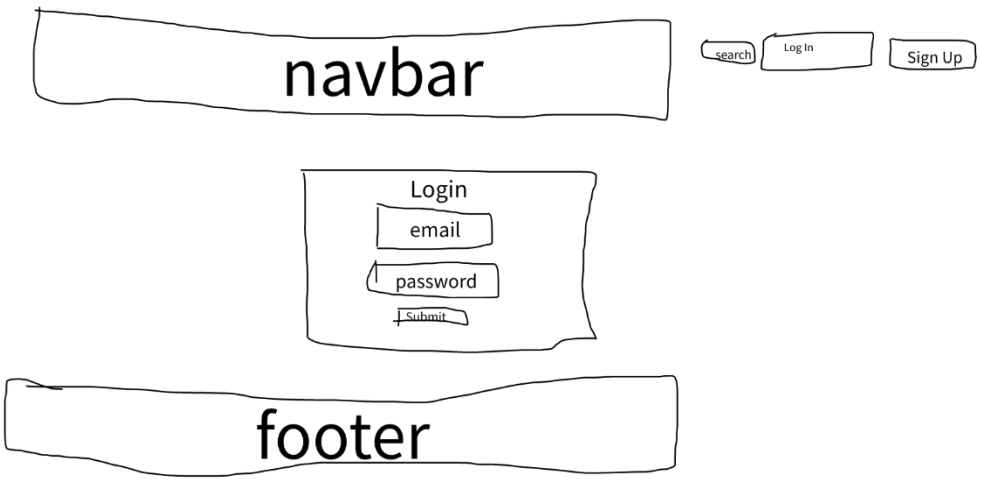


Figure18: Sketch of Log In page

5.3.3 Wireframes

After initial design ideas were done refining the design of most of the pages, wireframes were created to refine this even more. The focus is still on the layout, general user-experience, white space utilization, balance and basic functionality in this stage deliberately ignoring styling decisions. This will be done below, in the interactive prototype stage.

Assistant Secretary for Public Affairs (2019) states that a wireframe is a two-dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of content, functionalities available, and intended behaviours. It's a quick and efficient method to spot usability problems early in the design process.

For each page of the artefact, the navigation bar will be placed horizontally on top of the page. The home button will be to the far left while the about, products or store page, shopping tools and customer service will be centred. Finally, the search bar, log in and sign up will be to the far right. When a user logs in, this will turn into a dropdown menu containing their order and cart page as well as the log out option. This placement is to ensure visitors quickly and easily find what they are looking for on the artefact.

In the body of the page would be the main content. For the log in and registration page, the boxes will be here. On the other hand, for the product details page (seen below in Figure 19), the image of the product will be seen on the left while the product details will be seen on the right.

Below in figure 19 is one example of the product details page wireframe and the rest can be seen in Appendix E:

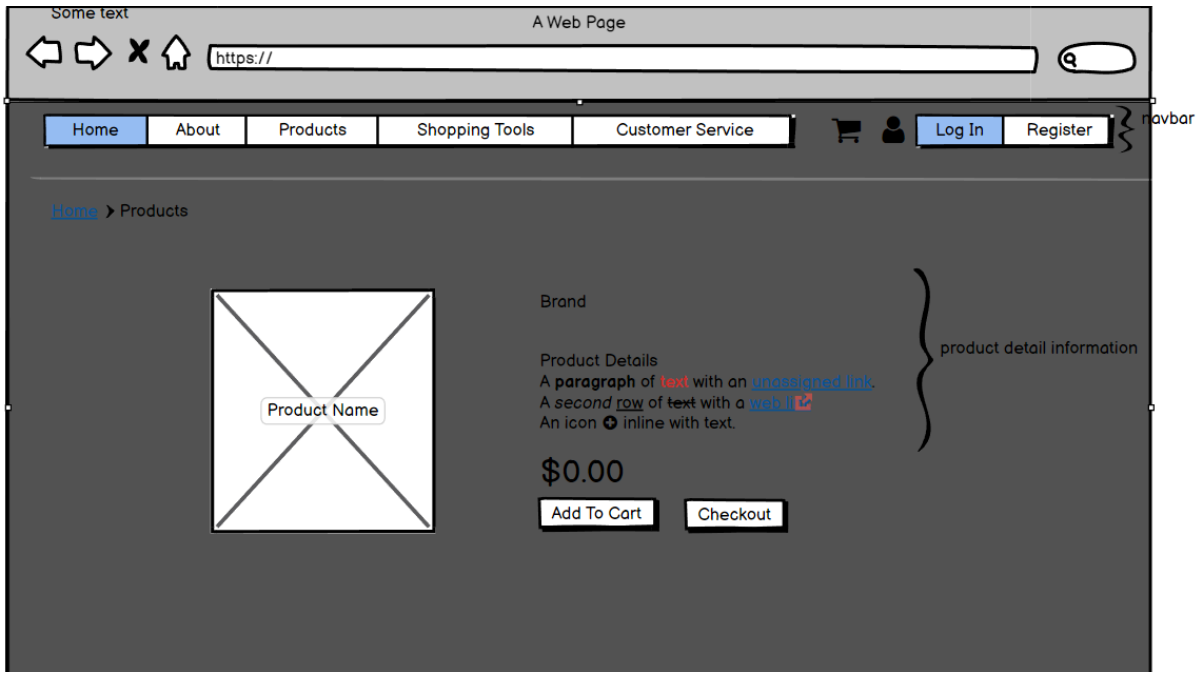


Figure 19: Wireframe for product details page

5.3.4 Interactive Prototype

This is the last stage in the design phase of the artefact and generally help to illustrate the final product, allowing the design team to comprehend the product's function, and target audience (The NYU Dispatch, 2019). This step is crucial since it will very closely resemble the artefact and therefore informs the development stage.

As stated earlier, design decisions are considered in this stage. Design decisions like typography, color and interactivity. The static wireframes created earlier acts as the springboard to explore these ideas and assist with the creation of high-fidelity prototypes.

Since prototyping is typically time-consuming and the development timeline was limited, prototyping was done only on certain elements like the header and footer which is considered crucial. Prototyping was done using Axure.

The font family chosen was Monsterrat, a legible, sans serif font which would help site visitors read and understand content.

The artefact's layout or structure was designed with simplicity in mind to avoid the digital clutter users sometime experience when attempting to purchase products online. This minimalistic approach meant the artefact had to be stripped of all unnecessary elements such as gradients, highlights and irrelevant textures leaving only the quintessential. Each element on the page was carefully considered and only placed if they were necessary to support the functionalities of the artefact.

White space was also used to complete the minimalistic approach of designing the artefact. Empty spaces interspersed on the different pages will allow for visitors to not experience overwhelm and be able to digest the content more easily. This was deemed necessary as specifications for computer hardware can be inherently overwhelming.

Finally, a monochromatic color scheme was used to further convey this minimalist approach while using the color hex# #3598dc blue color on the buttons as this evokes feelings of reliability and security (Anon, nd).

The interactive prototype for the login page can be seen below in Figure 20:

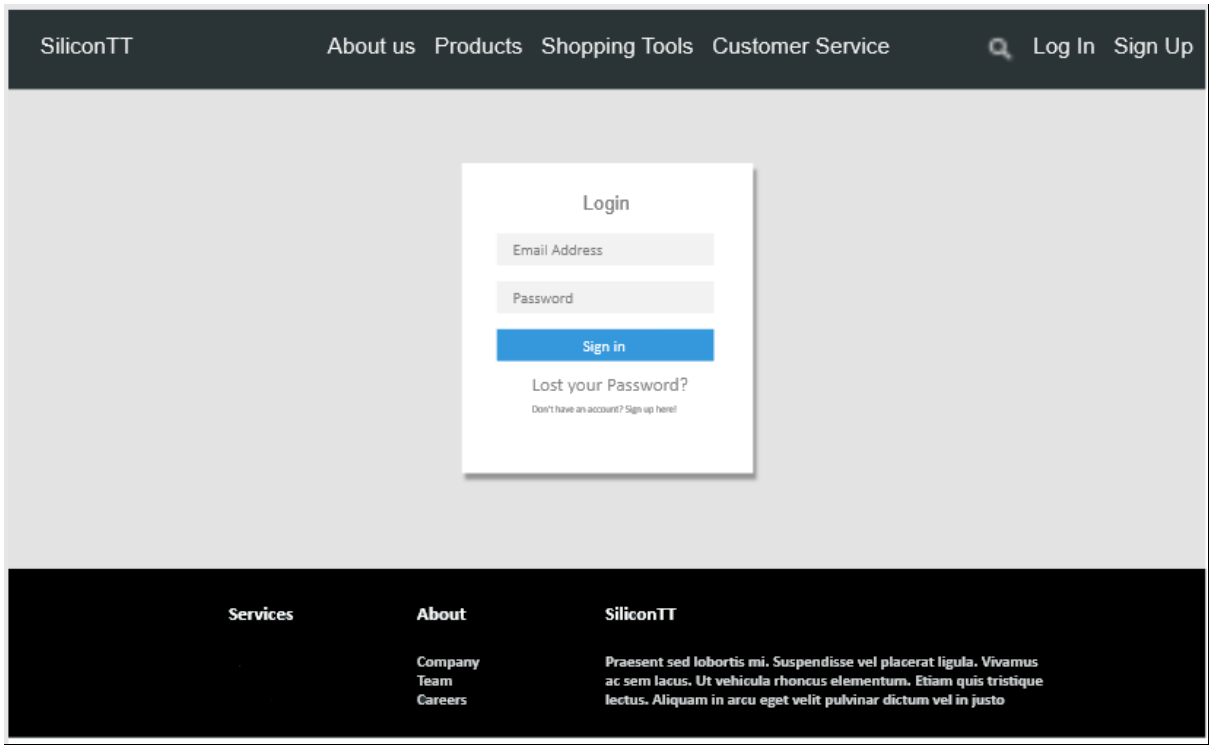


Figure 20: Wireframe for login page

CHAPTER 6: IMPLEMENTATION

6.1 Introduction

The implementation phase is the stage where all the research done in the requirements and design phases is used to develop the practical, physical application. This practical work includes getting all the software tools installed, configured and ready to use and getting the database and application to "speak" to one another. It also requires writing the necessary code and creating the relevant database tables.

A synopsis of all the above stated practical work will be detailed including the process of implementation for all the functional and non-functional requirements. It will also cover any design modifications that occurred, handling of error messages, security considerations, how version control was managed, any difficulties or issues that emerged during development and what was done to rectify them.

6.2 The Development Environment

A development environment provides developers with an interface and a pragmatic overview of the development process, which includes writing code, testing it, fixing bugs, and maintaining it without making any changes to the live environment. After consideration, the tools in Table 7 below was used. Installation and configuration was done without major issue.

SOFTWARE	FUNCTION
Microsoft Windows 10	Operating system used on laptop
Microsoft Office 365	The creation of technical reports was done using Microsoft Word, while the creation of the Gannt chart was done using Excel.
XAMMP	Offline on a local device, the program and its database were housed, managed, built, and tested using cross-platform, Apache, MySQL, PHP, and Perl (Laptop)

GitHub	Used to control versioning
Visual Studio Code	Used as a simplified code editor that supports version control and development tasks – Integrated Development Environment (IDE)
PHP Intelephense (VS Code extension)	To aid with PHP development
Highlight Matching Tag (VS Code extension)	Highlights matching open and close tags
Bracket Pair Colorizer 2 (VS Code extension)	Tailored colouring matching brackets
Auto Complete Tag (VS Code extension)	Automatically adds the close code and renames paired tags
Auto Close Tag (VS Code extension)	Automatically adds the close code
Error Lens (VS Code extension)	Detects and emphasizes coding errors
PHP, HTML, CSS jQuery	Chosen programming languages to build the application
MySQL	Used with PhpMyAdmin, for the database tables creation and management
Google Chrome, Microsoft Edge, Opera, Mozilla Firefox	Various browsers used to test the artefact

Table 7: Development tools used for artefact

6.3 Creation of Database

A project folder named 'web_app_database' was created in PhpMyAdmin to manage the database, and a project folder named 'web_app' was established locally in the XAMPP/htdocs folder to hold all project-related application files and images. After manually creating the database tables using MySQL, all primary and foreign keys were created in accordance with the data dictionary done in the project's design phase. Figure 21 below shows the database created:

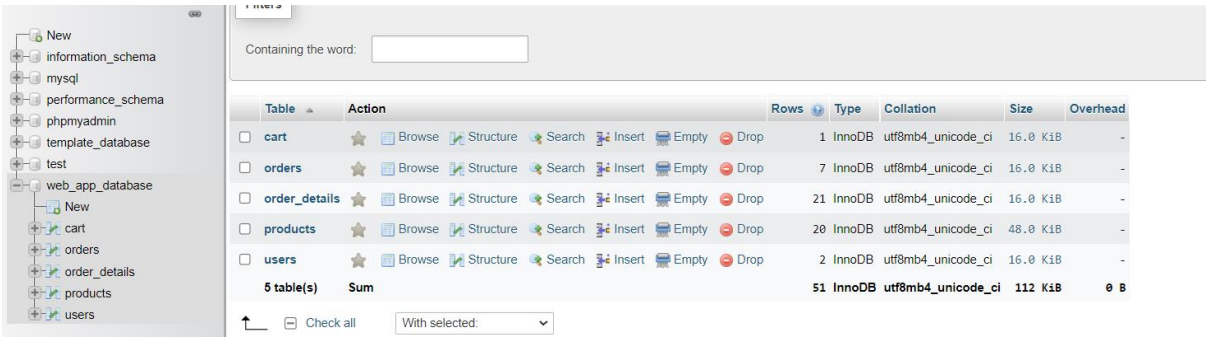


Figure 21: Database created

6.4 Connection between Database and Front End

A connection string was created to establish communication between the application (front-end) and the database (back-end), enabling the pages of the artefact to send and receive data from the MySQL database. This was done by placing the name of the database to be communicate with in the project's env file. This can be seen below in Figure 22 and Figure 23:

```php
class Database
{

    //PDO Connection
    public $pdo = null;

    public function __construct()
    {

        try {

            $servername = $_ENV["DB_SERVERNAME"];
            $username = $_ENV["DB_USERNAME"];
            $password = $_ENV["DB_PASSWORD"];
            $dbname = $_ENV["DB_NAME"];


            $this->pdo = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
            // set the PDO error mode to exception
            $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

            return $this->pdo;
        } catch (PDOException $e) {
            echo "Connection failed: " . $e->getMessage();
        }
```

Figure 22: Connection between the database and front-end

```
# DO NOT USE /
PROJECT_NAME=web_app


# -------------------------------------
# Router configurations
PROJECT_PATH="/${PROJECT_NAME}/"
DIR_COUNT=1


# -------------------------------------
# Database connection
DB_SERVERNAME=localhost
DB_USERNAME=root
DB_PASSWORD=
DB_NAME=web_app_database
```

Figure 23: Connection between the database and front-end in env file

6.5 User Interface Implementation

To date 12 out of 13 of the artefact's functional requirements have been 100% implemented as evidenced by the progress table seen in Table 8 below. To provide further in-depth information on the implemented requirements, a more thorough analysis of each requirement will be conducted in the sections that follow.

Requirement ID	Requirement	Requirement Type	Status Completed
FR1	Register/create an account	Core	100%
FR2	Log in/Log out	Core	100%
FR3	Order history/order details	Core	100%
FR4	Cart management – Add to cart, edit cart, remove from cart	Core	100%
FR5	Product detail	Core	100%
FR6	Product listing/store catalogue	Core	100%
FR7	Checkout flow	Core	100%
FR8	Search, sort, filter	Core	100%
FR9	Informational Pages	Core	100%
FR10	Online payment method	Advanced	100%
FR11	Recommender systems/ Predictive Analytics	Advanced	100%
FR12	Chatbot Integration	Advanced	100%
FR13	Product Inventory Management	Advanced	30%
NFR1	Security – Password Encryption	Advanced	100%
NFR2	Usability – Responsive design	Advanced	100%
NFR3	Data Integrity – Database security	Advanced	100%
NFR4	Scalability	Advanced	100%
NFR5	Maintainability	Advanced	100%

Table 8: Requirements implemented

6.5.1 Functional Requirements Implemented

In addition to samples of code used in the creation process and the front-end display, this section will go over the core requirements that have been implemented so far in the artefact.

The development of the application was done using the model-view-controller (MVC) design pattern coupled with basic object-oriented-programming (OOP) concepts. The MVC design pattern uses modularization of the program's input (model), processing (controller) and output (view) to manage

an artefact's source code. This sort of modularization prevented bugs on one page from affecting another page which could potentially significantly delay development.

In an application using the MVC design pattern, the controller takes in all requests and then directs the model to prepare any data that the view needs. The view makes use of the information prepared by the controller to produce the result.

FR1 Register/Create an account

The option to register/create an account will empower the user, maintain their privacy, secure their data, provide insights into consumer behaviour that can help focus inbound marketing efforts and create an enduring, strong, and permanent customer relationship. During the requirements analysis phase, this was a standard feature in all eCommerce sites viewed and was therefore mandatory.

The register/create an account feature gives users access to features like purchases that are only available to registered users. It was designed to be as familiar as possible without sacrificing needed functionality.

To register, a first name, last name, email address and password must be provided. An HTML form, seen below in Figure 24, is used to capture this information.

```
<div class="registration-form">
    <form action="registration" method="post">
        <h2 class="text-center">Registration</h2>
        <div class="form-group has-error">
            <input type="text" class="form-control" name="first_name" placeholder="First Name" required="required">
        </div>
        <div class="form-group has-error">
            <input type="text" class="form-control" name="last_name" placeholder="Last Name" required="required">
        </div>
        <div class="form-group">
            <input type="text" class="form-control" name="email" placeholder="Email Address" required="required">
        </div>
        <div class="form-group">
            <input type="password" class="form-control" name="password" placeholder="Password" required="required">
        </div>
        <div class="form-group">
            <button type="submit" name="registration" class="btn btn-primary btn-lg btn-block">Register</name=button>
        </div>


    </form>
```

Figure 24: HTML form for user registration

Using the HTML input type as “password” ensured that the password remained truncated to ensure user privacy as seen below in Figure 25.

Figure 25: Truncation of password seen on front-end

If a required field is left out, the user will get an alert prompting for the missing information as seen below in Figure 26.

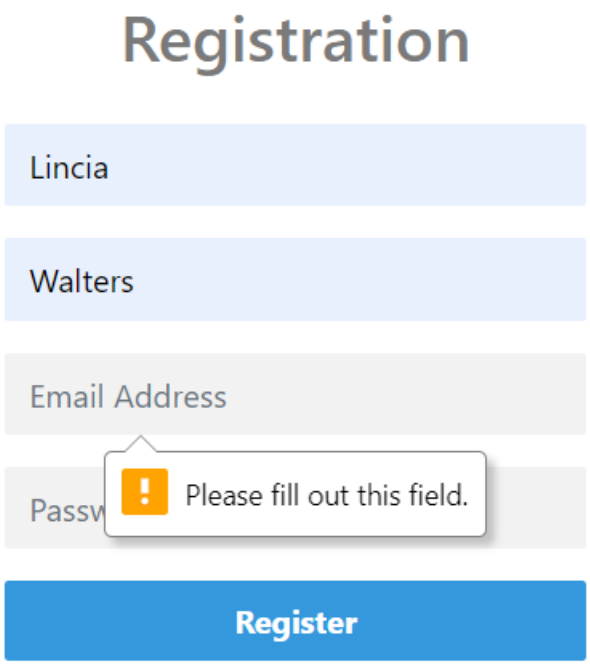


Figure 26: Alert seen if required field is left blank

Once all required fields are filled out successfully and the Register button on the site's page is clicked, the controller will be called by the connection between the HTML form's registration button name and will then proceed to call the "register" function in the Model using an if statement.

The controller's if statement and the model's register function can be seen below in Figure 27 and Figure 28.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["registration"])) {
        $user_object->register($_POST);
    }
}
```

Figure 27: Controller if statement

```php
public function register($inputs)
{
 $data = [
        "first_name" => $inputs["first_name"],
        "last_name" => $inputs["last_name"],
        "email" => $inputs["email"],
        "password" => password_hash($inputs["password"], PASSWORD_DEFAULT)
 ];

    $sql = "INSERT INTO `web_app_database`.`users`
    (`user_id`,
    `first_name`,
    `last_name`,
    `email`,
    `password`,
    `user_created`,
    `user_type`)
    VALUES
    (
    NULL,
    :first_name,
    :last_name,
    :email,
    :password,
    CURRENT_TIMESTAMP(),
    'user'
    );
    ";

    $stmt = $this->pdo->prepare($sql);
    $stmt->execute($data);
}
```

Figure 28: Model register function

All verified information from the Model will then be passed from the “register” function in the User model, the user will be alerted of successful registration and the record inserted into the database to be stored for future log in. Figure 29 below shows the MySQL database record for a new registered user.

user_id	first_name	last_name	email	password	user_created	user_type
3	Lincia	Walters	lincia.walters@hotmail.com	$2y$10$oAOOGO1EyMlz0qUuCbB5w..RXGoo.e...	2022-08-18 19:40:36	user
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 29: MySQL database record for registered user

FR2 Login/Logout

After successfully registering, users can use the “log in page” to access features like checkout, view cart, add to cart, order history and order details as well as view recommended products. A user can only log in if the credentials they provide (email address and password) match those stored in the database's "users" table.

The “login page” was designed to match the “registration page” and the artefacts theme to ensure uniformity. The fact that this page was a clone of the “registration page”, with only two less HTML divs and an input element, as well as minor PHP code additions, made it easy to develop.

To login, the user needs to enter the email address and password they used to register. Similarly to the registration page, the login page used the HTML input type as “password” to ensure that the password remained truncated. This can be seen below in Figure 30.

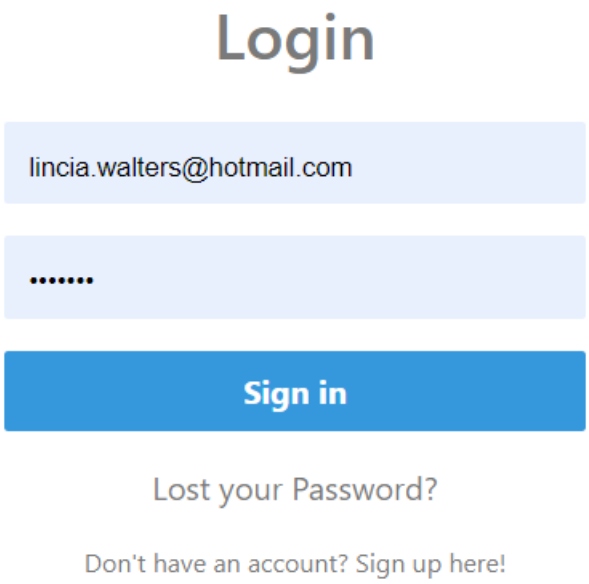


Figure 30: Password truncation seen on front-end for login page

After the user enters this information, the Login controller will be called by the connection between the HTML form’s login button name and will then proceed to call the “login” function in the User model using an if statement to complete login.

The HTML login form displaying the login button name, the controller's if statement and the "login" function in the Model can be seen below in Figures 31, Figure 32 and Figure 33.

```
<div class="login-form">
    <form action="login" method="post">
        <h2 class="text-center">Login</h2>
        <div class="form-group has-error">
            <input type="text" class="form-control" name="email" placeholder="Email Address" required="required">
        </div>
        <div class="form-group">
            <input type="password" class="form-control" name="password" placeholder="Password" required="required">
        </div>
        <div class="form-group">
            <button type="submit" name ="login" class="btn btn-primary btn-lg btn-block">Sign in</button>
        </div>
        <p><a href="#">Lost your Password?</a></p>
        <p class="text-center small">Don't have an account? <a href="http://localhost/web_app/registration">Sign up here!</a></p>
    </form>

</div>
```

Figure 31: HTML login form

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["login"])) {
        if ($user_object->login($_POST)) {
            echo "Login was successful";
            header("location: store");
        }else {
            echo "Incorrect details";
        }
    }
}
```

Figure 32: Controller if statement for login

```
public function login($inputs)
{
    $sql = "SELECT * FROM users WHERE email = ?";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$inputs["email"]]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user && password_verify($inputs["password"], $user["password"])) {
        echo "Found the user";
        $user["password"] = null;
        $_SESSION["current_user"] = $user;
        return true;
    } else {
        echo "No user found";
    }


    return false;

}
```

Figure 33: Login function in the User model

On the user's end, an alert will be displayed confirming successful login and the artefact's navigation bar will not display the "Login" and "Sign Up" options anymore but an account menu displaying the user's name which will contain "Log out", "Cart" and "Orders" along with all other navigational elements and a successful login alert. This can be seen in Figure 34 below. The user is then free to add items to their the cart and checkout when ready.

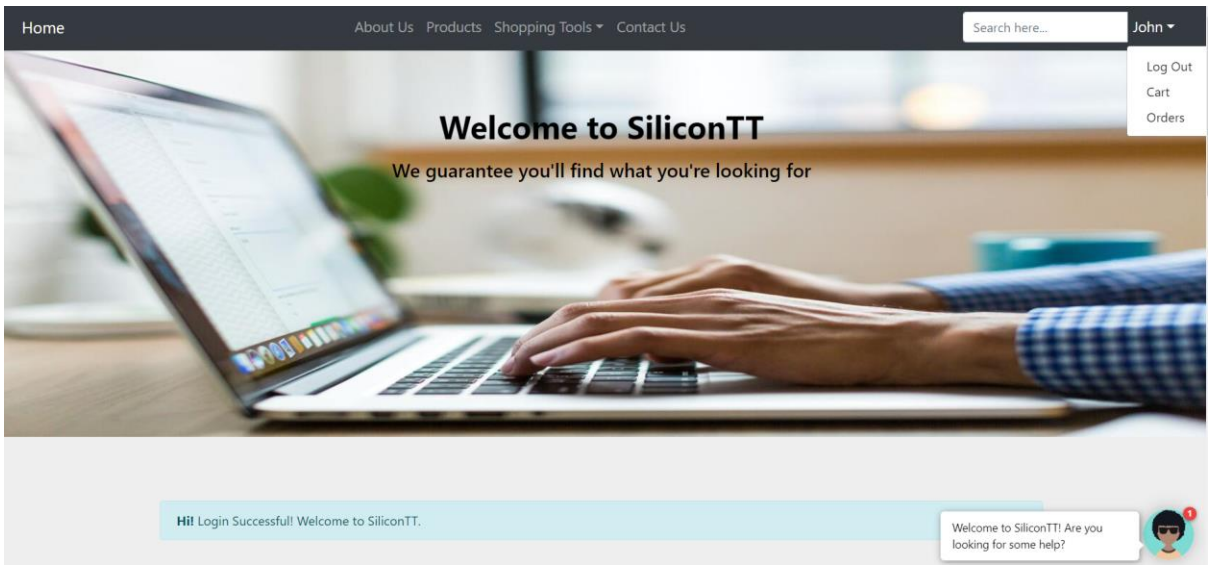


Figure 34: Account Menu and successful alert seen after login

FR3 Order History/Order Details

A customer can easily maintain track of all recent orders by using the “order details/order history” feature. It enables the buyer to access up-to-date details on each online purchase, including payment information, products ordered and transaction dates.

Having a method of record-keeping benefits the buyer in a number of ways. For this artefact, going to the “Orders” page on the user’s account menu enables a customer to view their order history and track the status of any purchase by clicking on the “Details” link. Further to this, order histories like this one serve as a useful guide for those who want to reorder the same item.

Therefore, customers are not required to speak with customer service for anything relating to their order details or history. By keeping the customers informed and communicating effectively, the purchasing experience is enhanced. The customer’s expectations are therefore managed and the possibility of complaints are managed.

This was implemented by creating a getUserOrders and a getUserOrderDetails function in the Order model which would then be called in the Orders and Order-details controllers respectively whenever the user views their orders or order details. These can be seen in Figures 35, 36 and 37 below.

```php
public function getUserOrders($user_id) {
    $sql = "SELECT * FROM orders WHERE user_id = ?";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$user_id]);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}

public function getUserOrderDetails($user_id, $order_id) {
    $sql = "SELECT *
    FROM orders, order_details, products
    WHERE orders.order_id = order_details.order_id
    AND order_details.product_id = products.product_id
    AND orders.user_id = ?
    AND orders.order_id = ?
    ";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$user_id, $order_id]);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}
```

Figure 35: getUserOrders and getUserOrderDetails functions in User model

```
$orders = $order_object->getUserOrders($user_id);
```

Figure 36: getUserOrders function in User model

```
$orders = $order_object->getUserOrderDetails($user_id, $id);
```

Figure 37: getUserOrderDetails function in User model

A snippet of the front-end where the customer can see their orders and respective link to the order details is provided in Figure 38.

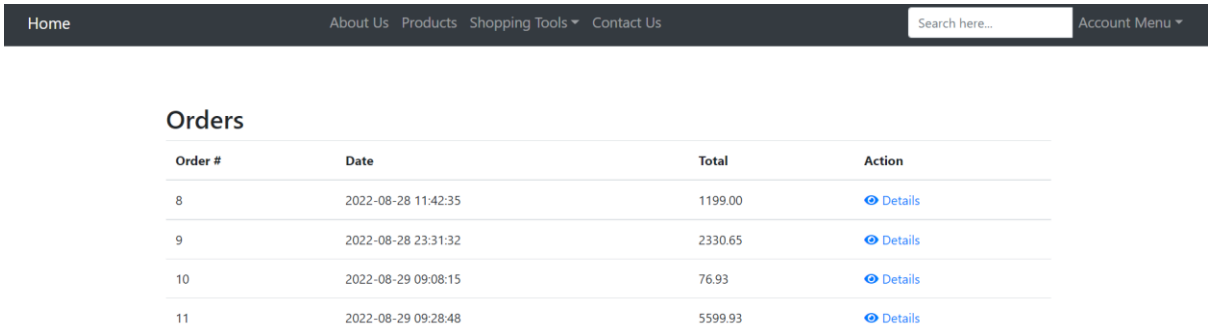


Figure 38: Front-end display of orders and link to order details

FR4 Cart Management

To achieve a completed buying experience, cart management was another mandatory feature.

A shopping cart acts as the bridge between browsing and shopping and therefore must be designed with elements to allow the customer freedom, give agency for as many conversions from browser to shopper as possible and prevent any unforeseen mishaps that may occur resulting in cart abandonment. Cart abandonment is therefore a risk factor that was considered in the design considerations.

In light of the above, design and development of the cart management feature made certain that registered users can add and remove items from their carts as well as change the quantity of each item. A registered user can add products to their cart by clicking the "Add to Cart" button on the product details page. The application will send the user back to their cart so they may either remove the goods, change the quantity, or purchase the items there.

A logged in customer can use the “Add To Cart” button on the specific product detail page to add something to their cart for purchase. When this button is clicked, the product chosen will get added to the cart where it can be removed or purchased.

Add to cart

When the add to cart button is clicked on the user's end, a function in the cart model called addToCart stores the required information that the cart entity needs in the database like the cart_id, the user_id of the customer that added it to the cart, the product_id of the product they added, the quantity of the products they added and so on. and. the front-end view can be seen in Figure 40

```
<div class="col-md-10">
    <button name="add_to_cart" class="btn btn-primary btn-lg">Add to cart</button>


</div>
```

The Add to Cart button in HTML can be seen below in Figure 39.

Figure 39: Add to cart button in HTML

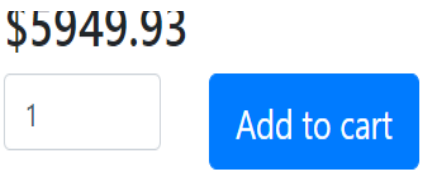


Figure 40: Add to cart button on front-end

The addToCart function in the cart Model can be seen in Figure 41:

```
public function addToCart($user_id, $product_id, $cart_quantity) {
    $data = [
        "user_id" => $user_id,
        "product_id" => $product_id,
        "cart_quantity" => $cart_quantity
    ];

    $sql = "INSERT INTO `web_app_database`.`cart`
    (`cart_id`,
    `user_id`,
    `product_id`,
    `cart_created`,
    `cart_quantity`,
    `cart_status`)
    VALUES
    (
    NULL,
    :user_id,
    :product_id,
    current_timestamp,
    :cart_quantity,
    'cart'
    );
    ";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute($data);
}
```

Figure 41: addToCart function in cart Model

The cart controller will then create a cart_details object which calls another function in the cart model called getCartDetails using an SQL query to allow the customer's cart information to be displayed on the cart page later on.

These can be seen in Figures 42 and 43 below.

```php
$cart_details = $cart_object->getCartDetails($user_id);
```

Figure 42: cart_details object in Cart controller

```php
public function getCartDetails($user_id) {
    $sql = "
    SELECT * FROM cart, products
    WHERE cart.product_id = products.product_id
    AND cart.user_id = ?
    AND cart.cart_status = 'cart'
    ";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$user_id]);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);


    foreach ($result as $product) {
        $this->subtotal += $product["product_price"] * $product["cart_quantity"];
    }

    $this->cart_details = $result;


    return $result;
}
```

Figure 43: getCartDetails in cart Model

Finally, the cart page then uses the cart_details object and the data array wrapped in a PHP loop to display all products in the users cart. This can be seen in Figure 43 below:

```php
<?php foreach ($cart_details as $data):?>
```

Figure 44: cart_details object wrapped in PHP foreach loop with data array

Remove from cart

The remove from cart function is a lot simpler than the Add to cart functionality. After the Remove button is clicked by the user, an “if statement” in the Cart controller calls the removeFromCart function in the Cart Model after testing if the condition is true. The removeFromCart uses a simple

SQL deletion statement using positional parameters. The “if statement” in the Cart controller and the Model’s function can seen in Figure 45, 46 and 47 below.

```php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    if (isset($_POST["cart_id"])) {
        $cart_object->removeFromCart($_POST["cart_id"], $user_id);
    }
}
```

Fiure 45: Cart controller remove from cart

```php
public function removeFromCart($cart_id, $user_id) {
    $sql = "DELETE FROM cart WHERE cart_id = ? AND user_id = ?";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$cart_id, $user_id]);

}
```

Figure 46: removeFromCart function in Cart Model

An entire view of the front-end of the cart can be seen below in Figure 47:

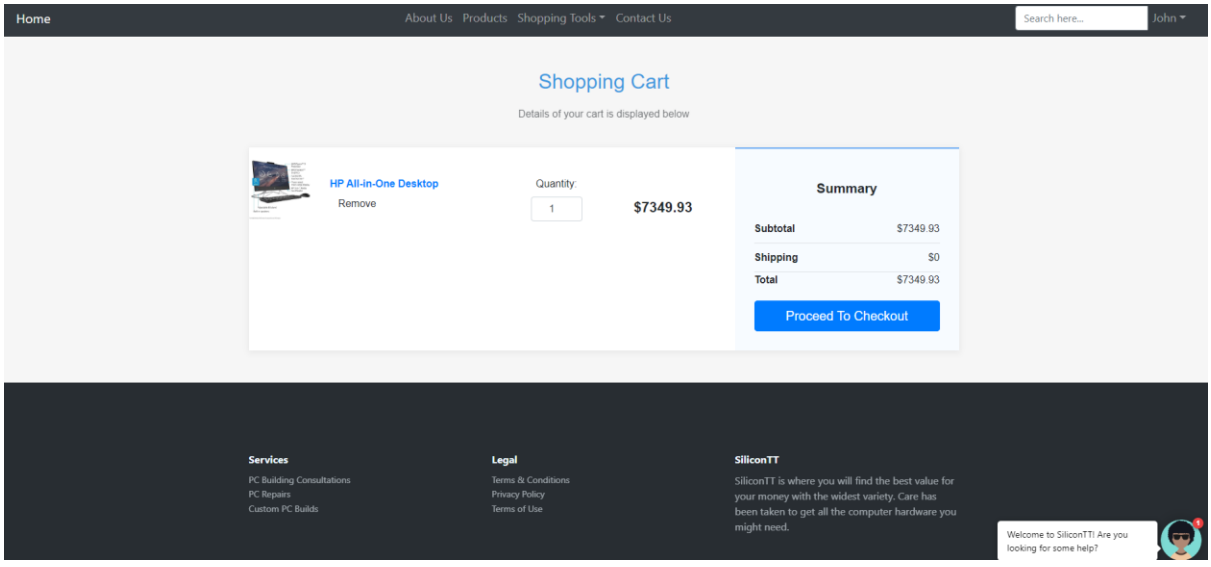


Figure 47: Front-end view of the cart

FR5 Product Detail

Product details act as first-level customer service as they answer a lot of the questions browsing customers would have about a product. Further to this, product descriptions improves credibility, appears professional and entices the customer to complete a purchase.

Care was taken to provide accurate, detailed and relevant descriptions for each product by endlessly researching and attempting to pre-empt any questions the customer might have. For computer hardware, this is absolutely necessary as potential customers might have a list of technical requirements that they need to meet before purchasing a product.

The product details feature includes a details page that will show detailed information about the product such as technical specifications, a description, included components and more. It was implemented by manually updating the database product table (the back-end) with a product title, description, price and relevant images.

To allow the product information in the products table to be displayed on each product's details page, the getProductDetails in the Product Model used a simple SQL query with the product_id as a positional parameter to display the particular product's description on a new web page. This was tied into the details page on the user's end by tying in PHP statements withing the HTML referencing the database's columns.

The getProductDetails function, a front-end example of a product and its details and the HTML page with the PHP connections to the database columns are shown below in Figures 48, 49 and 50.

```php
public function getProductDetails($product_id) {
    $sql = "SELECT * FROM products WHERE product_id = ?";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$product_id]);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}
```

Figure 48: getProductDetails function in Product Model

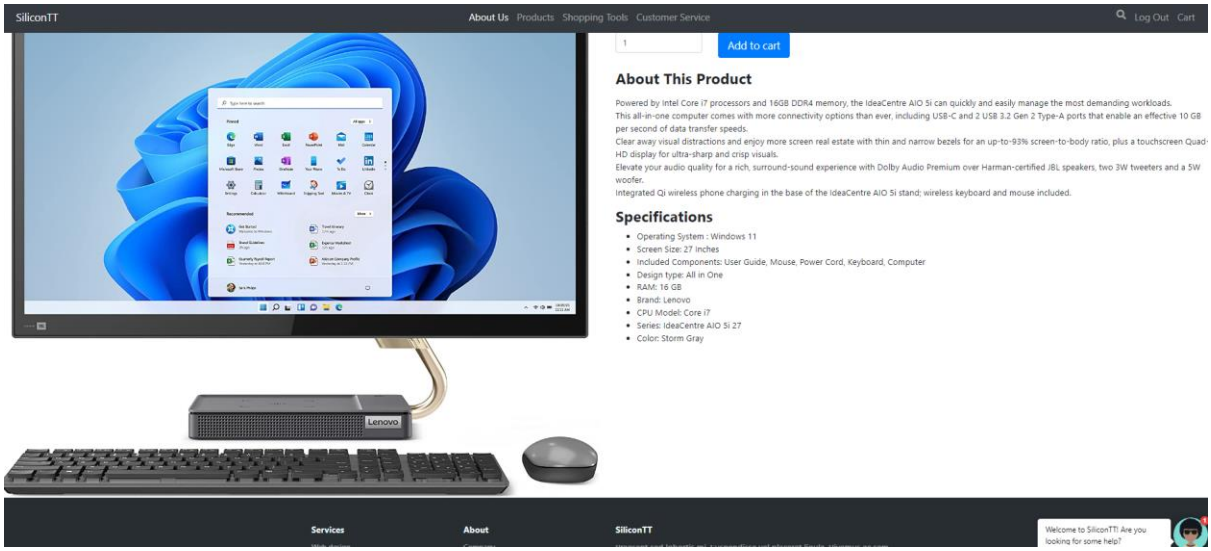


Figure 49: Product detail front-end

```php
<div class="col-md-6">
    <h3><?php echo $data["product_title"]; ?></h3>
    <h3>$<?php echo $data["product_price"]; ?></h3>
    <form action="<?php echo BASE_URL . "details/$id"; ?>" method="post">
        <div class="row">
            <div class="col-md-2">
                <div class="form-group">
                    <input name="cart_quantity" value="1" type="number" class="form-control" id="usr">
                </div>
            </div>
            <div class="col-md-10">
                <button name="add_to_cart" class="btn btn-primary btn-lg">Add to cart</button>
            </div>
        </div>
    </form>
        <div class="row">
            <div class="col-md-12">
               <h3><b>About This Product</b></h3>
                <p><?php echo $data["product_description"]; ?></p>
            </div>
        </div>
</div>
```

Figure 50: PHP in details view front-end page

FR6 Product Listing/Catalog

The product listing catalog page is also crucial for conversions as it is the user's first impression of the product. This page allows the customer to view all the products the store currently has with enough basic information to spark the curiosity of an uncertain customer or assist a customer that is already aware of what they are searching for.

There are standard information that needs to be displayed in this catalog. Standard information like product name, image, price and relevant CTAs. A bootstrap template which reflects these necessary elements was used.

This bootstrap template was edited to be able to interact with the database similarly to the product details page, using PHP within the HTML to reference the respective database columns needed to be displayed. A PHP "foreach loop" was used to loop through all product IDs in the product_id column of the products entity in the database to enable the images, titles and prices to be shown. This can be seen implemented in Figures 51, 52 and 53 below:

```
<div class="row">

<?php foreach ($product_details as $data) :
 $link = BASE_URL . "details/{$data["product_id"]}";
 ?>

 <div class="col-md-4">
     <figure class="card card-product-grid card-lg">
         <a href="<?php echo $link; ?>" class="img-wrap" data-abc="true">
             <img src="<?php echo BASE_URL . $data["product_image1"]; ?>"></a>
         <figcaption class="info-wrap">

                 <div class="row">

                     <div class="col-md-8">

                         <a href="<?php echo $link; ?>" class="title" data-abc="true"><?php echo $data["product_title"]; ?></a>

                     </div>

                     <div class="col-md-4">

                         <div class="rating text-right">
                             <i class="fa fa-star"></i>
                             <i class="fa fa-star"></i>
                             <i class="fa fa-star"></i>

                 </div>


         </figcaption>
         <div class="bottom-wrap">

             <div class="price-wrap">
                 <span class="price h5"></span>  $<?php echo $data["product_price"]; ?><br> <small class="text-success">Free shipping</small>
             </div>
         </div>
     </figure>
 </div>

 <?php endforeach; ?>

                     </div>

                 </div>
```

Figure 51: PHP and HTML used to reference database columns

After this was done, the store controller called a function called getAllProducts which was defined in the Product model as seen below.

```php
$product_details = $product_object->getAllProducts();
```

Figure 52: product object calling function in store controller

```php
public function getAllProducts() {
    $sql = "SELECT * FROM products";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute();
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}
```

Figure 53: getAllProducts function in Product Model

FR7 Checkout Flow

Reaching the checkout stage might be a crucial turning point for potential customers in the buying process. If the checkout process is not convenient and simple enough, a potential customer might abandon this final stage of purchase. This can take various forms but for the artefact, checkout was simplified by firstly choosing a bootstrap template that ensures these features. Secondly, by displaying all relevant information the customer might want on the page such as products to checkout, a clear subtotal amount and all products to checkout. And finally by utilizing familiar elements to avoid confusion.

Figure 54 below shows the checkout front-end page:

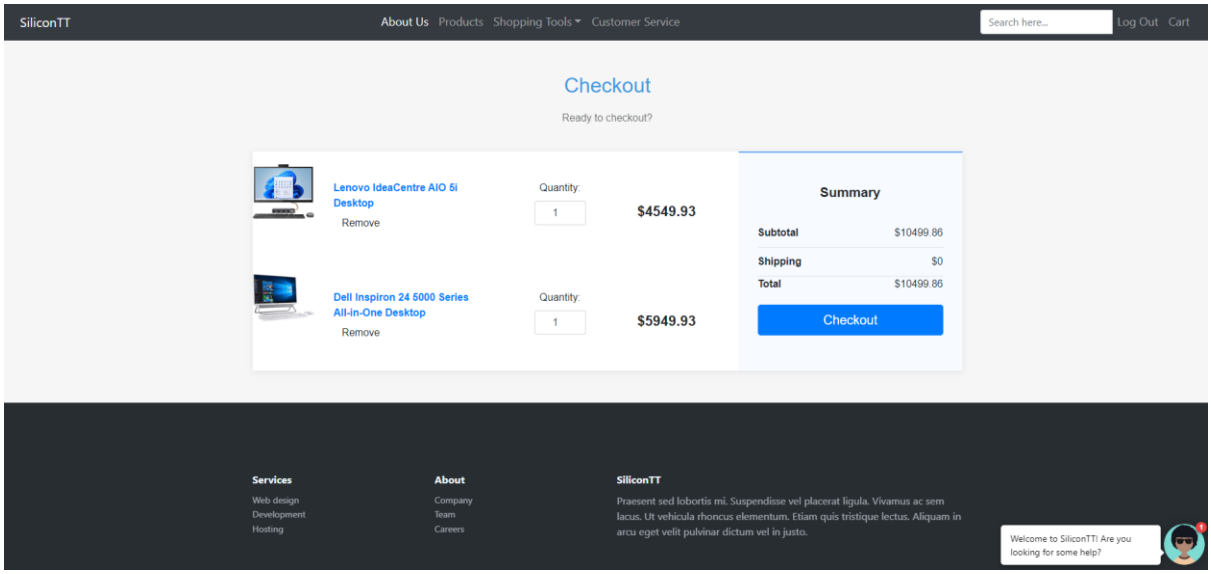


Figure 54: Front-end view of checkout page

As can be seen, it was designed to be as simple and effective as possible with relavant CTAs.

On the back-end, the checkout controller will pull the cart details using the getCartDetails function already created for the cart management requirement (seen in Figure 55) and place it in a cart object to display everything in the customer's cart on the checkout page.

```php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    if (isset($_POST["cart_id"])) {
        $cart_object->removeFromCart($_POST["cart_id"], $user_id);
    }
}


$cart_details = $cart_object->getCartDetails($user_id);


$cart_object->calculateTotal();
```

Figure 55: Checkout controller with cart object and removeFromCart function being called

FR8 Search, sort and filter

As GCFglobal (nd) states, sorting and filtering are tools that let you organize your data. When you sort data, you are putting it in order. Filtering data lets you hide unimportant data and focus only on the data you're interested in.

Searching is a quick, handy and necessary tool used to find specific products.

Proper implementation of the search, sort and filter requirement will enable users to find what they are looking for faster and more effectively. If users cannot find what they are looking for easily, they will go elsewhere. Therefore, this feature is imperative for conversions and basic customer care.

From the user's vantage, these 3 options will be either inputting the specific keywords into the search bar to use the search feature and selecting from a list of items to sort or filter accordingly. The front-end can be seen in Figure 56 and 57 below:

Figure 56: Search bar on front-end

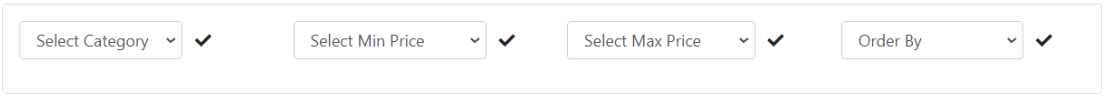


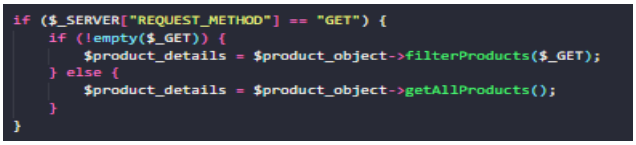
Figure 56: Sort and filter bar on store page

On the back-end side, a simple switch case was used to implement this. The switch case was used for the search, sort and filter function while a separate orderBy function was created as a complement to sort by varying criteria in the Product model. This can be seen below in Figure 57:

```php
//search, sort and filter
public function filterProducts($inputs) {

    $sql = "SELECT * FROM products WHERE product_id > 0";


    foreach ($inputs as $key => $value) {

        if(empty($value)) {
            continue;
        }

        switch ($key) {
            case 'category':
                $sql .= " AND product_category = '$value'";
                break;
            case 'search':
                $sql .= " AND product_title LIKE '%$value%'";
                break;
            case 'min_price':
                $sql .= " AND product_price >= '$value'";
                break;
            case 'max_price':
                $sql .= " AND product_price <= '$value'";
                break;
            case 'order':
                $sql .= $this->orderBy($value);
                break;

            default:
                # code...
                break;
        }


    }

    $stmt = $this->pdo->prepare($sql);
    $stmt->execute();
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}


public function orderBy($value){
    switch ($value) {
        case 'order-title':
            return " ORDER BY product_title ASC";
            break;
        case 'order-title-desc':
            return " ORDER BY product_title DESC";
            break;
        case 'order-price':
            return " ORDER BY product_price ASC";
            break;
        case 'order-price-desc':
            return " ORDER BY product_price ASC";
            break;
        default:
            # code...
            break;
    }
}
```

Figure 57: Switch case used for search, sort and filter

In the Store controller, the function is then called when the feature is used by a customer as can be seen below in Figure 58.



```php
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    if (!empty($_GET)) {
        $product_details = $product_object->filterProducts($_GET);
    } else {
        $product_details = $product_object->getAllProducts();
    }
}
```

Figure 58: Functions called when feature is used

FR9 Informational Pages

Informational pages are eternally useful on an e-commerce site. They answer typical questions visitors would have while establishing authority and expertise.

Using personas developed in the design phase, consideration was given to what visitors might want to learn more about to create these pages. Buying hardware can sometimes be technically overwhelming for the average person so the aim in getting the content for these pages was to alleviate some of this overwhelm. “Desktop buying guide” for example will help an uncertain customer make a decision or teach a certain customer something new.

Below is an example of one of these pages in Figure 59:

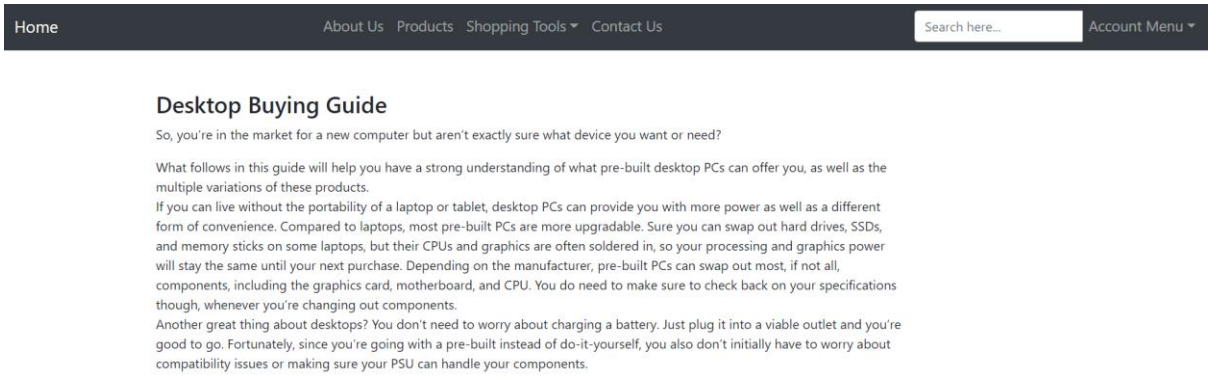


Figure 59: Front-end view of informational page

The implementation of these pages were simple HTML divs and informational content.

FR10 Online payment method

An online payment functionality is necessary for the online hardware store to function as an e-commerce application. If a transaction does not include a payment method for the hardware, it cannot be completed.

Stripe, a supplier of payment services accepting credit cards, mobile wallets, buy-now pay-later services, and more than 135 different currencies was used to act as a payment service provider (PSP). A payment service provider does most of the same things that a merchant account does, to the point where the typical user may not notice the difference at all (Mottola, 2022).

Stripe was integrated allowing Stripe's credit card information fields to be displayed on the cart. The total cost of the cart is passed to the Stripe code pages so that Stripe may be contacted for payment.

Stripe payments made can be seen on Stripe's dashboard below in Figure 60:

Figure 60: Payments on Stripe dashboard

FR11 Recommender systems/Predictive analytics

Product recommendations are ideas for products based on what other customers who have bought the same item the customer is currently purchasing have also bought. As discussed in the literature review, this is called collaborative filtering and predictive analytics will be applied here to do this.

Product recommendations influence how customers connect with the artefact, can encourage lasting loyalty, and give each shopper the impression that they are using a site that is specifically designed for them. This will then translate into increased conversion rates, increase the time spent on the artefact and potentially increase order value.

For the artefact, this was implemented by creating a function in the Product model (seen below in Figure 61) using an SQL query which displays products bought by other customers who bought the same product currently being purchased from the database.

```
public function getRecommendedProducts($product_id) {
    $sql = "SELECT *
    FROM products
    WHERE product_id in
    (select distinct product_id from order_details where order_id in
    (select distinct order_id from order_details where product_id = ?)
    and product_id != ?)"
    ;
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute([$product_id, $product_id]);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}
```

Figure 61: getRecommendedProducts function

A recommended-products.php page was then created to let this function call these recommended products. This will be displayed at the bottom of the product that is being viewed. An example of what this page would look like can be seen in Figure 62 below.

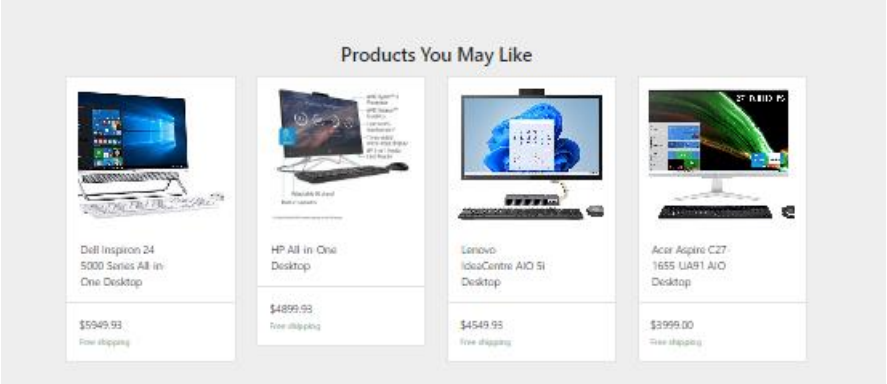


Figure 62: Recommended products displayed on the front-end

FR12 Chatbot integration

A chatbot is a computer program that, at its most basic level, simulates and processes human conversation (spoken or written), enabling people to communicate with digital devices as if they were speaking with a real person.

It is a great tool which will enhance customer service, help ease customer frustration, be a 24/7 point of contact and deliver a personalized customer service experience.

Integrating the chatbot was done using IntelliTicks due to how elaborate and time-consuming development of a bespoke chatbot would have been. IntelliTicks is one of the best CRM software that offers interactive, real time chatbot integration. An account was created and the necessary code placed in the footer.php file to show on every page the user might navigate to. The code can be seen below in Figure 63 along with a screen of interaction with the bot in Figure 64 below.

```
<!-- chatbot code -->
<script type='text/javascript'>
  (function(I, L, T, i, c, k, s) {
    if (I.iticks) return;
    I.iticks = {
      host: c,
      settings: s,
      clientId: k,
      cdn: L,
      queue: []
    };
    var h = T.head || T.documentElement;
    var e = T.createElement(i);
    var l = I.location;
    e.async = true;
    e.src = (L || c) + '/client/inject-v2.min.js';
    h.insertBefore(e, h.firstChild);
    I.iticks.call = function(a, b) {
      I.iticks.queue.push([a, b]);
    };
  })(window, 'https://cdn-v1.intelliticks.com/prod/common', document, 'script', 'https://app.intelliticks.com', 'zjqeBijCKA82qfmfx_c', {});
</script>
```

Figure 63: Chatbot code

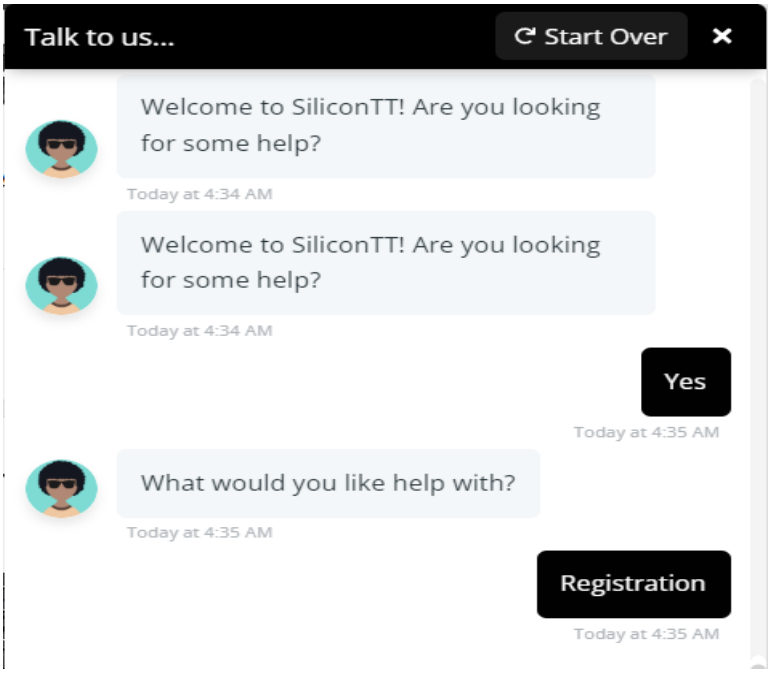


Figure 64: Chatbot interaction

6.5.2 Non-functional Requirements Implemented

NFR1 Security – Password encryption

Before saving the password used to register by customers, encryption scrambles it. Therefore, if someone hacks the server, they find a random string of letters and numbers instead of the registered user's password.

To safeguard the passwords of registered users in the database, password encryption was implemented by using PHP's password_hash() method in the registration's function. Using this powerful one-way hashing technique, the password hash() method generates a fresh password hash. The only way to verify a hashed output, then, is to submit the original value to the hashing algorithm and contrast the outputs. Hashing is hence ideal for preserving user credentials.

The physical implementation of this can be seen below in Figure 64.

```
public function register($inputs)
{
 $data = [
        "first_name" => $inputs["first_name"],
        "last_name" => $inputs["last_name"],
        "email" => $inputs["email"],
        "password" => password_hash($inputs["password"], PASSWORD_DEFAULT)
 ];
```

Figure 65: password_hash used to encrypt passwords in database

NFR2 Usability – Responsive design

A major one of the goals of the artefact is to make it as user-friendly and intuitive as possible for the customer. To this end, responsive design was absolutely necessary.

A responsive website has a dynamic and adaptable layout which automatically adjusts the size of the page contents and elements to fit the screen being utilized for viewing, whether it is a mobile phone, tablet or a different browser. Without responsive design, the user will be forced to scroll, pan, resize or zoom depending on which device is being used. This can result in decreased bounce rates and decrease user frustration.

Developing separate for all screen types is virtually impossible given the large amount of devices that currently exist as opposed to designing one artefact that has the capability to be responsive.

This was implemented by using Bootstrap 4 templates. Bootstrap is an inbuilt User Interaction (UI) or front-end design technique which uses a div grid system of containers, rows, and columns to lay out and align web content with the help of HTML and CSS. This div based architecture has built-in pre-defined classes and elements saving the developer time that would have otherwise been spent writing these from scratch.

An example of how this div architecture is applied can be seen in Figure 66 below which was taken from the cart page.

```
<div class="container">
    <div class="block-heading">
      <h2>Shopping Cart</h2>
      <p>Details of your cart is displayed below</p>
    </div>
    <div class="content">
        <div class="row">

            <div class="col-md-12 col-lg-8">
                <div class="items">
```

Figure 66: Bootstrap's div architecture

Because of the tight deadline, Bootstrap was the best way to implement responsive design since this freed up time to focus on the more difficult tasks of developing the back-end and implementing the functionalities without being derailed by designing.

The “toggle device” toolbar on Google Chrome was used to test responsiveness. It simulates responsiveness of different devices and displays the responding view. This can be seen for a user’s cart page in Figures 67 and 68 below.

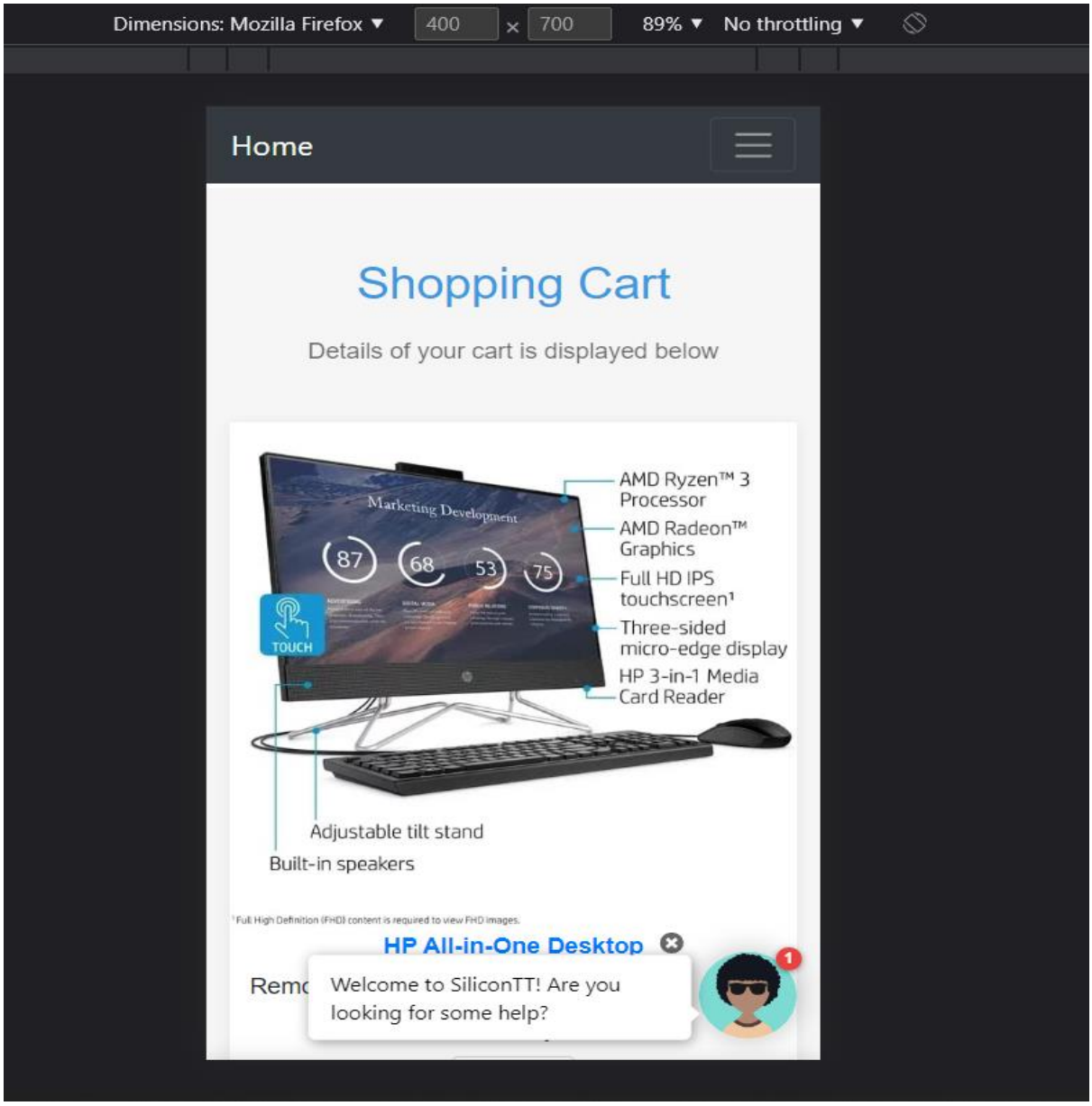


Figure 67: Toggle device for Mozilla Firefox

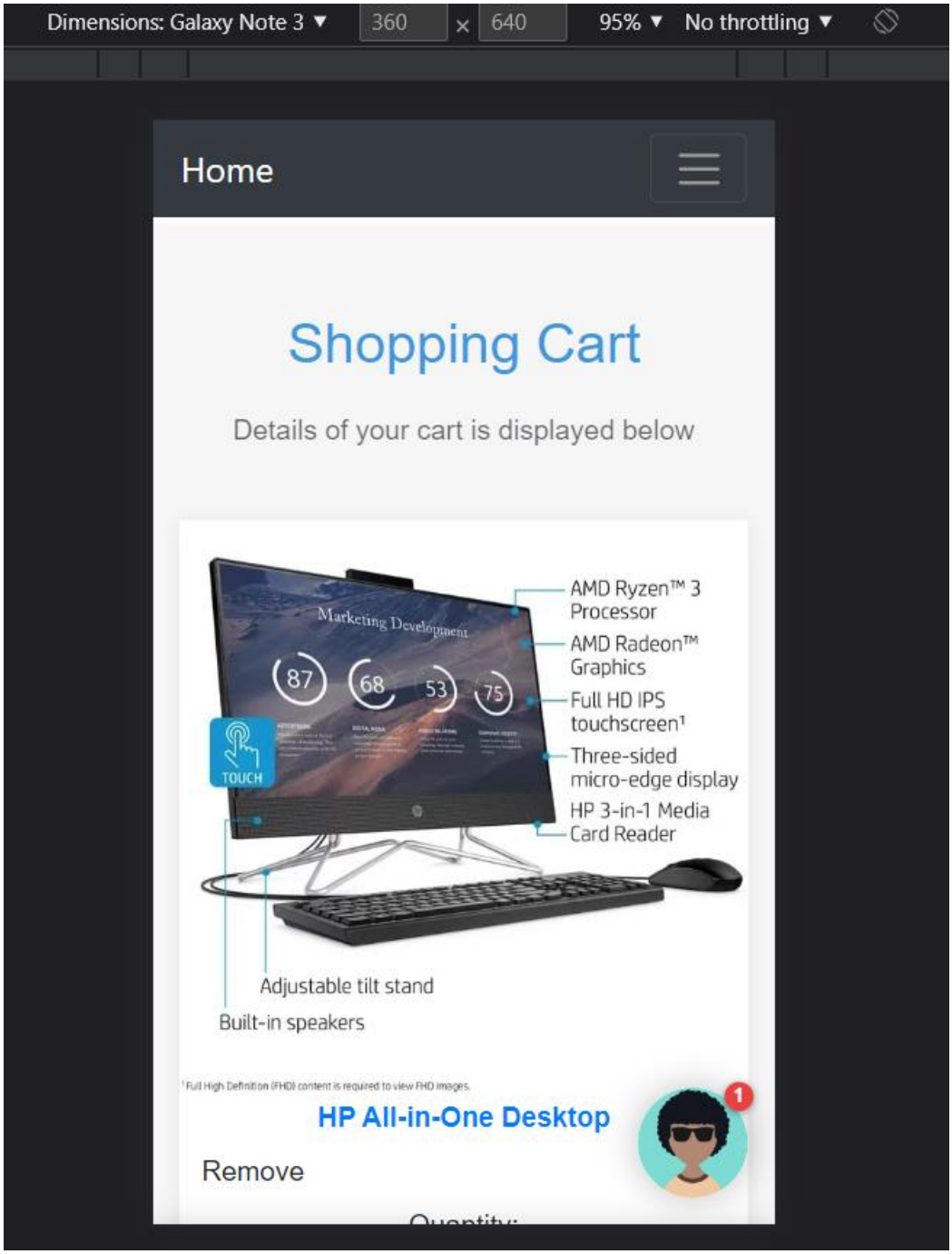


Figure 68: Toggle device for Galaxy Note 3

6.6 Changes to Design

Screen uniformity was a crucial part of the design execution, so a lot of preparatory work was done to ensure that every requirement was met for development to match as closely as possible to the artefact design in the earlier chapters. Therefore, no major adjustments were made to the artefact.

6.7 Error Messages

The user experience will inevitably have errors. When an action fails inadvertently, error messages emerge to reveal the state of the system and to help the user identify the problem and resolve these errors. These will typically occur when:

- The user input is not understood by the artefact.
- The action a user desires to perform is incompatible with the functionality of the artefact.
- The information a form field require is missing

To assist users in smoothly recovering from problems, useful error messages were incorporated throughout the program in accordance with Nielsen's Heuristics, rule 9 (help users recognize, diagnose, and recover from errors). In order for users to understand where the error occurred, the error messages were placed both where they are immediately visible to the user and close to the source of the error. All entered data were validated whenever it was possible.

Below are a few examples of the various error messages that appear when users take prohibited actions.

- When a user tries to log in without a password (Figure 69 below):

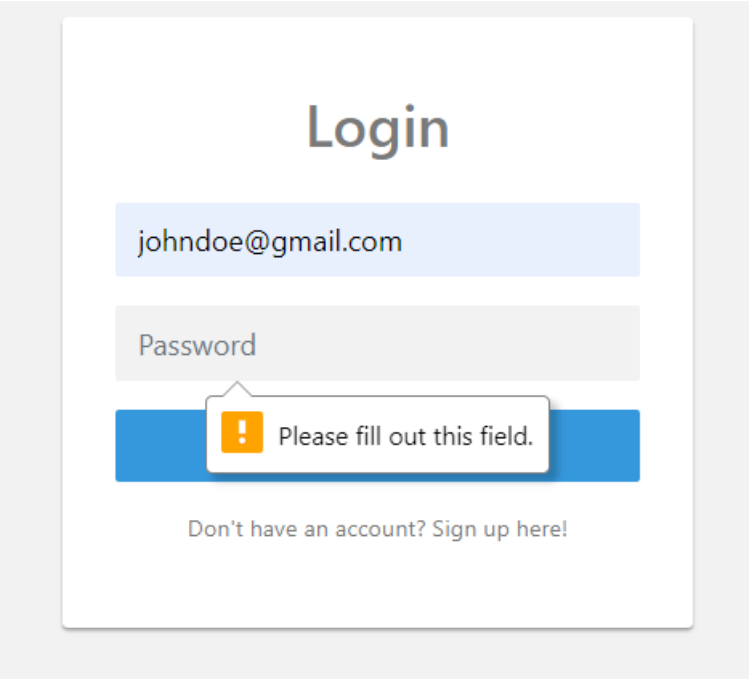


Figure 69: Error seen when user tries to log in without a password

- When a customer tries to access an empty cart page (Figure 70 below):

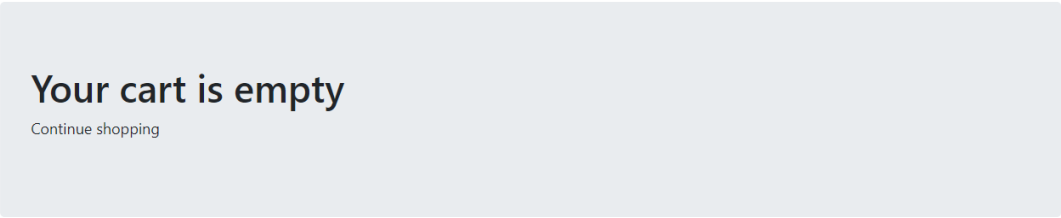


Figure 70: When the user tries to go onto an empty cart page

- When a registered customer tries to log in using an incorrect email or password (Figure 71 below):

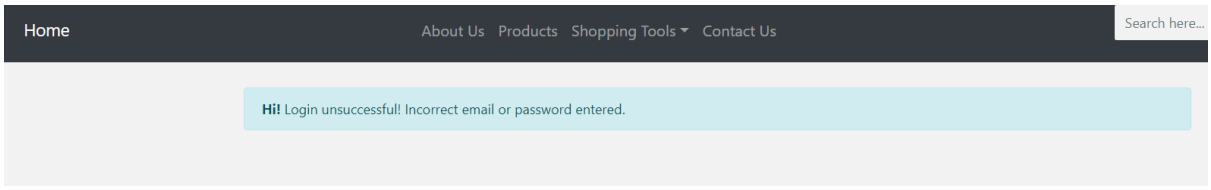


Figure 71: When a user tries to log in with an incorrect email or password

6.8 Security Consideration

Due to the wide variety of cybercrimes and the importance of privacy and security, this application employs a number of security implementations. Authentication measures were therefore put in place to make sure the artefact is as secure as possible. Only users with legitimate email addresses and passwords could log in and access their personal data (such as their order history), for example.

The implementation chapter already covered a few security issues implemented in some of the functional requirements such as the password encryption explained above. Another one is using truncated password inputs.

6.9 Version Control

In software development, version control is a method for managing and tracking source code changes. It can handle merge issues and other irregularities because it maintains code updates and

versions; this protects the source code from irreparable harm and permits experimenting without worrying about making mistakes or introducing code conflicts.

Version control helps with time savings, clutter reductions from having many of the same files with a similar name and add-ons to identify version changes, as well as noting a specific area of modification can be used in subsequent work.

The built-in GIT version control in Visual Studio Code made it easy to regularly create backup sources of the code and upload them to GitHub's web repository. Figure 72 below displays the most recent section of the GitHub online repository's logs.

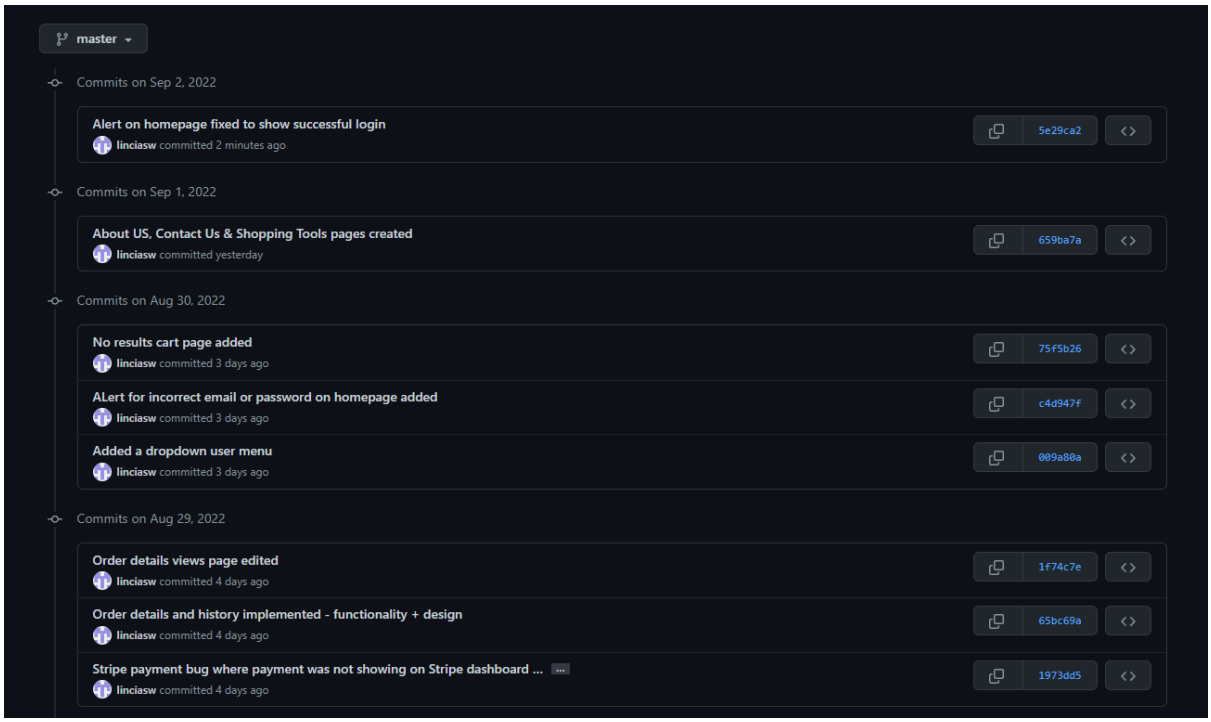


Figure 72: Github commits

6.10 Features Not Implemented

Despite having, in my opinion, a well-thought-out strategy, the application nevertheless missed one of the proposed features because of time-constraints. Nevertheless, I was still able to make the most of the basic and essential components of 12 of the 13 functional requirements 100% functional, leaving only one entirely missing, namely the product inventory management.

The purpose of this feature was to provide an easy, user-friendly and streamlined way for an administrator to manage the products on the artefact as opposed to manually updating the physical database. This is a feature that was considered “nice-to-have” and was thus sacrificed so the approximately 8 hours (6 hours for research and 2 hours for developing) needed to implement this feature could be spent elsewhere implementing and improving on requirements that were considered necessary.

CHAPTER 7: TESTING

7.1 Introduction

Tuteja and Dubey (2012) state that the objective of software testing is to find problems and fix them to improve quality. They go on to break down the primary objectives of testing into four (4) main categories:

- Demonstration: It shows that the system can be used for integration with acceptable risk. It demonstrates functions under special conditions and shows that products are ready for integration or use.
- Detection: It discovers defects, errors and deficiencies. Determines system capabilities and limitations quality of components, work products and the system.
- Prevention: It provides information to prevent or reduce the number of errors clarify system specifications and performance. Identify ways to avoid risk and problems in the future.
- Improving Quality: By doing effective testing, we can minimize errors and hence improve the quality of software.

The rigorous software testing phase of the Agile Lifecycle is critical to ensure the artefact behaves the way it ought to functionally, discover any bugs or defects that may be overseen which may inevitably affect the artefact's performance, check for security holes, styling issues and coding logic errors.

This chapter will discuss the testing that was done on each functional and non-functional requirement before, during, and after the creation of the artefact to minimize any fallout of these issues.

Each test was listed on a test plan, along with the data that would be utilized in each case. The test document contains a record of the outcomes from these tests. Finally, a defects log was then used as the source for bug fixes after testing was completed to document faults that were discovered.

7.2 Test Plan

The project's testing strategy includes creating a set of tests to be performed on all the application's functional and non-functional requirements. It will be tested using a combination of unit and integration tests. Unit testing will concentrate only on one application component, called a unit, ensuring the component can handle known input using valid, invalid and data that will exceed known boundaries. On the other hand, integration testing tests grouped units or the artefact as a whole to check data flow and to verify that all of the units are communicating properly.

Each of these tests will contain test data, anticipated outcomes and actual outcomes, as well as results. All tests will be supported by evidence. A failed test is a defect, and will be recorded in the Defects Log. An attempt will be made to fix all defects after testing is completed.

7.3 Functional Requirements Testing

FR-1 Register / Create an Account

Table 9 below shows the testing done on FR1 - register/create an account with the respective test data, expected outcome, actual outcome and the result. Appendix F will contain all other functional requirement tests and the results.

Test#	Test with data	Expected Outcome	Actual Outcome	Result
FR 1.1	Register with no information inserted in fields	A message asking to please fill out require fields	A message asking to please fill out require fields	Successful
FR 1.2	Register customer Lincia Walters with email address lincia.walters@hotmail.com and password: password	Customer successfully registered, alert for successful registration displayed and customer rerouted to log in page. In the users database, a record for customer was inserted.	Customer successfully registered, alert for successful registration displayed and customer rerouted to log in page. In the users database, a record for customer was inserted.	Successful
FR 1.3	Register another user with email address lincia.walters@hotmail.com and password: password	Error message indicting email address already exists. Please try again.	User was registered erroneously	Failed
FR1.4	Register customer Blake Lively with email address blake.lively@gmail.com and password: password123	Customer successfully registered, alert for successful registration displayed and customer rerouted to log in page. In the users database, a record for customer was inserted.	Customer successfully registered, alert for successful registration displayed and customer rerouted to log in page. In the users database, a record for customer was inserted.	Successful

Table 9: Testing done on FR1 – Register/Create an account

7.4 Non-Functional Requirements Testing

Each non-functional requirement that could be tested had a set of tests created, and the outcomes were recorded just like they were for the functional requirements. Any failed tests were placed in the defects log. Table 10 below shows testing done for NFR -1 Security - Encryption. Appendix G will contain all other non-functional requirement test and the results.

NFR1 – Security – Password encryption

Test #	Test Description	Expected Outcome	Actual Outcome	Result
1.1	User registers with truncated password on registration page	Password truncated while user registers	Password truncated while user registers	Successful
1.2	HASHED password stored in database's users tabel	New record created in users table with HASH encrypted password	New record created in users table with HASH encrypted password	Successful
1.3	User login with truncated password	Password truncated while user logins	Password truncated while user logins	Successful

Table 10: Testing done on NFR1 – Password Encryption

7.5 Defects Log

Defect logging is merely recording defects and bugs in the artefact observed while testing. It acts as a point of reference for items that need to be repaired or refactored in the artefact. Only tests that failed in the functional requirements and non-functional requirements testing done were entered here.

Severity levels were used to indicate priority.

The severity levels are as follows:

- 0 - Not a problem at all.
- 1 - Cosmetic issue: Only requires attention if time permits.

- 2 - Minor problem: Low priority repair.
- 3 - Major problem: High priority resolution

The defects log for the artefact can be seen in Table 11 below:

Entry	Test #	Requirement	Problem	Severity	Comment
1	FR 1.3	Registering user with same email and password	This feature was not fully configured	2	Not fixed because of time constraints
2	10.1	Online payment with credit card	Stripe page not loading for credit card details to be keyed in	3	Fixed

Table 11: Defects log

7.6 Analysis of Test Results

After testing was completed on all functional and non-functional requirements, two defects were identified. One of these defects were not fixed due to time constraints and complexity of implementation. The other defect was at severity level 3 and hence fixed. No major problems were left outstanding.

CHAPTER 8: EVALUATION

8.1 Introduction

Evaluating or assessing the artefact is a key step in the final stages of the development process along with the testing done in the previous chapter. Broadly speaking, it helps identify any room left for improvement, gauge usability for end-users and verifies all goals laid out at the beginning of the project planning phase were met.

The goals at the beginning of the project were to enhance the customer experience, increase sales and encourage repeat buying. While testing answers the question of “does the artefact work?”, evaluation answers the question of “how well does it work?” thus putting the focus on the usability of the artefact.

This chapter seeks to assess if the goals stated above and the usability of the artefact were completed successfully by the application of a heuristic evaluation.

8.2 Method of Evaluation

Although there are a plethora of software evaluation methods such as external user testing and focus groups, in an effort to conserve time that would have otherwise been spent attaining ethics approval from the university, the evaluation method decided upon is a heuristic evaluation.

A heuristic evaluation is a thorough, technically sound process that evaluates the artefact against clear guidelines. Heuristic evaluations are produced from a wide range of projects, observation, study, and experience over a prolonged period of time, so they have some plausibility and authority in the subject to which they are applied.

There is a concern that doing the evaluation using heuristics will produce a personal and prejudiced evaluation instead of concrete, empirical data but using heuristics is still considered an efficient, cost-effective way of identifying usability issues.

8.3 The Evaluation

The 10 usability criteria from Nielsen and Molich, which were outlined in the report's design chapter, served as the basis for the heuristic evaluation used for this project.

The evaluation will identify if the heuristic was not implemented, somewhat implemented, or completely implemented throughout the whole application for each item on the checklist. As a consequence, the outcome was represented by a rating, as follows:

- 0 - indicates the heuristic was not implemented
- 1 - indicates the heuristic was partially implemented
- 2 - indicates the heuristic was fully implemented

Table 12 below show all Nielsen heuristics being evaluated against the artefact:

1	**Visibility of system status**	
#	Review Checklist	Rating
1.1	Does every display begin with a title or header that describes screen contents?	2
1.2	Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?	2
1.3	Is there a consistent icon design scheme and stylistic treatment across the system?	2
1.4	Is there visual feedback in menus or dialog boxes about which choices are selectable?	2
1.5	Is there visual feedback in menus or dialog boxes about which choice the cursor is on now?	2
2	**Match between system and the real world**	
#	Review Checklist	Rating
2.1	If there is a natural sequence to menu choices, has it been used?	2
2.2	Do the selected colors correspond to common expectations about color codes?	2
2.3	When prompts imply a necessary action, are the words in the message consistent with that action?	2
2.4	On data entry screens, are tasks described in terminology familiar to users?	2
2.5	Do related and interdependent fields appear on the same screen?	2
3	**User control and freedom**	
#	Review Checklist	Rating
3.1	Are users prompted to confirm commands that have drastic, destructive consequences?	2
3.2	Can users reduce data entry time by copying and modifying existing data?	2
3.3	Can users move forward and backward between fields or dialog box options?	2
3.4	Can users cancel out of operations in progress?	2
3.5	If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut?	2
4	**Consistency and standards**	
#	Review Checklist	Rating
4.1	Has a heavy use of all uppercase letters on a screen been avoided?	2
4.2	Are icons labelled?	2
4.3	Are vertical and horizontal scrolling possible in each window?	2

4.4	Are there no more than four to seven colours, and are they far apart along the visible spectrum?	2
4.5	Does each window have a title?	2
5	**Error prevention**	
#	Review Checklist	Rating
5.1	Are menu choices logical, distinctive, and mutually exclusive?	2
5.2	Does the system prevent users from making errors whenever possible?	2
5.3	Does the system warn users if they are about to make a potentially serious error?	2
6	**Recognition rather than recall**	
#	Review Checklist	Rating
6.1	Do text areas have "breathing space" around them?	2
6.2	Are field labels close to fields but separated by at least one space?	2
6.3	Is colour coding consistent throughout the system?	2
7	**Flexibility and efficiency of use**	
#	Review Checklist	Rating
7.1	If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut?	2
7.2	On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut?	2
7.3	Does each data entry screen have a short, simple, straightforward, distinctive title?	2
8	**Aesthetic and minimalist design**	
#	Review Checklist	Rating
8.1	Is only (and all) information essential to decision making displayed on the screen?	2
8.2	Have large objects, bold lines, and simple areas been used to distinguish icons?	2
8.3	Does each icon stand out from its background?	2
8.4	Do error messages suggest the cause of the problem?	2
9	**Help users recognize, diagnose, and recover from errors**	
#	Review Checklist	Rating
9.1	Are prompts stated constructively, without overt or implied criticism of the user?	2
9.2	Are prompts brief and unambiguous?	2
9.3	Are error messages grammatically correct?	2
9.4	Do error messages avoid the use of exclamation points?	2
9.5	Do error messages avoid the use of violent or hostile words?	2
10	**Help and documentation**	
#	Review Checklist	Rating
10.1	Navigation: Is information easy to find?	2
10.2	Is the information relevant?	2
10.3	Are on-line instructions visually distinct?	2
10.4	Do the instructions follow the sequence of user actions?	2

8.4 Heuristics Report

A second report was generated for the evaluations that received a score of either 0 (not implemented) or 1 (partially implemented). This report, which functions similarly to the Defects log, is used to track components that did not entirely adhere to the criteria in order to help with solving the identified issues.

Within this table, each item was given a severity rating based on the how much of an impact on the artefact it will have and will subsequently be used as a priority list for correction. This severity rating is on a scale of 0 to 4 and is outlined below:

- 0 - I don’t consider this as a usability problem
- 1 - Cosmetic problem: only fix if time permits
- 2 - Mid Low Priority: low priority, but may be a quick and easily fixable error
- 3 - Mid High Priority: high priority but will take longer to fix
- 4 - High Priority: high priority, may be a quick and easy fix, fix immediately

However, there was no need for this report because every evaluation criterion that the Nielsen's Heuristics were intended to target was fully implemented.

8.5 Analysis of Results

Of the 43 things on the check list, all were fully implemented, according to the Heuristic Report's conclusions.

CHAPTER 9: CONCLUSION

9.1 Introduction

The undergrad project focused on the creation of an elaborate and functional artefact alongside a comprehensive, detailed technical dissertation outlining the steps taken to complete the artefact.

This project's final chapter provides a summary of the work performed and the experiences gained throughout project development. It highlights my successes, failures, experiences gained, skills acquired, critical analysis and reflects on anything I would have done differently if I had the opportunity to do it again.

9.2 Tasks Completed

The requirements chapter outlined thirteen functional requirements and 5 non-functional requirements to be implemented in the artefact after deep research and discovery. The project is currently 98% completed with 12 functional requirements out of the 13 non-functional requirements completed.

Due to the amount of time needed for development juxtaposed against the time I had, the Product Inventory management requirement was started but not completed. This feature was originally considered a nice-to-have on the administration end and therefore does not affect the overall quality of the artefact as a whole.

The complete list of requirements implemented can be seen in the implementation chapter in Table 8. For ease of reference, it is included below as well in Table 13.

Requirement ID	Requirement	Requirement Type	Status Completed
FR1	Register/create an account	Core	100%
FR2	Log in/Log out	Core	100%
FR3	Order history/order details	Core	100%
FR4	Cart management – Add to cart, edit cart, remove from cart	Core	100%
FR5	Product detail	Core	100%
FR6	Product listing/store catalogue	Core	100%
FR7	Checkout flow	Core	100%
FR8	Search, sort, filter	Core	100%
FR9	Informational Pages	Core	100%
FR10	Online payment method	Advanced	100%
%FR11	Recommender systems/ Predictive Analytics	Advanced	100%
FR12	Chatbot Integration	Advanced	100%
FR13	Product Inventory Management	Advanced	30%
NFR1	Security – Password Encryption	Advanced	100%
NFR2	Usability – Responsive design	Advanced	100%
NFR3	Data Integrity – Database security	Advanced	100%
NFR4	Scalability	Advanced	100%
NFR5	Maintainability	Advanced	100%

Table 13: Requirements implemented

9.3 Product Built

Given the 12-week time period for completing development of an artefact, the fact that I was the sole software developer, database designer, front-end designer, testing and implementation team, flaws are expected in the final product. Although not having the full range of capabilities and functionalities of well-known websites like Amazon.com and Newegg.com, the artefact still contain features worth mentioning for a student who was not 100% familiar with some of the technologies. Through grit, honing a passion to learn quickly, focus and determination, the artefact still stands as a fully functioning eCommerce application meeting all of its goals.

Some strengths of the application such as the development of a search, sort and filter bar, the integration of a chatbot, responsive design, the integration of an online payment method and the

application of predictive analytics to recommend products to customers all enable the artefact to be able to one-day scale to compete with the above mentioned mammoths in the industry.

Another major strength is the database design. Designing the database using the top-down approach and normalizing the tables into third normal form will ensure database integrity and by extension, scalability.

Despite these major strengths, some limitations exist however. Because functionality was the first priority and constant, time-consuming bug fixes during development occurred, design of the artefact is an inevitable flaw as less time was spent there. More focus on the artefact's theme, color psychology, typography and user interaction or user experience (UI/UX) considerations would have made the artefact ideal.

Another limitation is the lack of multiple payment options such as Paypal. This would help to increase usability and not isolate customer segments that may not want to use Stripe as a payment option.

Finally, having an option for a guest user to add items to their cart and implementing a functionality which allows seamless registration for checkout to take place would increase sales and help with usability.

9.4 Experienced Gained

I have a newfound respect for the skillset necessary to complete a project of this size. I gained quite a bit of knowledge that I will use in the future.

For instance, I now see the value of project management. It is challenging to plan and manage a project of this magnitude successfully, and in trying to do so, I have learned the subtleties of what is

necessary. In the future, I will find it handy to apply planning tools like GANTT Charts and Work Breakdown Structures. I developed an intimate understanding of how to effectively do a GANTT chart because of errors I made in this project.

Secondly, in terms of time-management, I have learnt about time-blocking, 90-minute sprint study sessions and how important a simple study time-table helps with consistent study and progression. I discovered if I took more than two days off of these daily tasks, momentum will become harder to get back when I was ready to resume. For me, it required more energy to stop for a period of more than two days and resume than it did to do work on the reports or artefacts daily using 90-minute sprint sessions regardless of mood, energy levels or lack of time. Realizing this was a game-changer for me. I decided to time-block two 90-minute sprint study sessions every day but will allow myself to settle for one if personal responsibilities had to be prioritized.

Lastly, I had to adapt my academic writing abilities as well because I had become accustomed to writing in a specific manner for both work and recreation. My work had to follow certain guidelines and conventions in order to satisfy the requirements of the university's assignments, which helped to increase their present capacity. I had been working on this since the initial Project Proposal and continued to this point. Further to this, I had to learn how to write reflectively and this became such a useful practice that I have found myself using it elsewhere.

I had some understanding of web programming with HTML, CSS and SQL before enrolling in the BSc program in my final year, but no experience with PHP. My enthusiasm for web development has increased as a result of my learning to program in PHP with the aid of programs like Visual Studio Code, MySQL and Xampp/phpMyAdmin for SQL databases. I rapidly realized that learning the language, its code, and its operations would require more time spent conducting background study. Utilizing online resources like w3schools, stack overflow, and YouTube rapidly helped with bug fixes and adding functionality exponentially.

10 REFERENCES

Anon, (n.d.). *10 Heuristic Principles – Jakob Nielsen's (Usability Heuristics)*. [online] Available at: https://www.uxness.in/2015/02/10-heuristic-principles-jakob-nielsens.html.

Anon, (n.d.). *Advantages of using blue in your web design – Pumpkin Web Design Manchester*. [online] Available at: https://www.pumpkinwebdesign.com/web-design-manchester/advantages-of-using-blue-in-your-web-design/.

Assistant Secretary for Public Affairs (2019). *Wireframing | Usability.gov*. [online] Usability.gov. Available at: https://www.usability.gov/how-to-and-tools/methods/wireframing.html.

Baymard Institute (2018). *41 Cart Abandonment Rate Statistics - Cart & Checkout - Baymard Institute*. [online] Baymard.com. Available at: https://baymard.com/lists/cart-abandonment-rate.

Belk, M., Germanakos, P., Constantinides, A. and Samaras, G. (2015). A Human Cognitive Processing Perspective in Designing E-Commerce Checkout Processes. Human-Computer Interaction – INTERACT 2015, pp.523–530. doi: 10.1007/978-3-319-22701-6_38.

bbbootstrap.com. (n.d.). *Bootstrap 4 navbar with three sections Example*. [online] Available at: https://bbbootstrap.com/snippets/bootstrap-navbar-three-sections-95989228 [Accessed 6 Sep. 2022].

BBBootstrap (n.d.). *Bootstrap 4 simple ecommerce product list with buy now button Example*. [online] BBBootstrap.com. Available at: https://bbbootstrap.com/snippets/simple-ecommerce-product-list-buy-now-button-86523960.

bbbootstrap.com. (n.d.). *Bootstrap 4 dark website footer with social media icons Example*. [online] Available at: https://bbbootstrap.com/snippets/bootstrap-dark-website-footer-social-media-icons-46403047 [Accessed 6 Sep. 2022].

Bootsnipp.com. (n.d.). *HTML Snippets for Twitter Boostrap framework*. [online] Available at: https://bootsnipp.com/snippets/xr3GB [Accessed 6 Sep. 2022].

BuiltIn (2017). What is Artificial Intelligence? How Does AI Work? | Built In. [online] Builtin. Available at: https://builtin.com/artificial-intelligence.

Chakraborty, A., Baowaly, M.K., Arefin, A. and Bahar, A.N. (2012) The role of requirement engineering in software development life cycle, Journal of emerging trends in computing and information sciences, 3(5), pp.723-729.

ChatBot. (n.d.). *Need-to-Know Chatbot Statistics in 2021*. [online] Available at: https://www.chatbot.com/blog/chatbot-statistics/.

Chen, H., Chiang, R.H.L. and Storey, V.C. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly, 36(4), p.1165. doi: 10.2307/41703503.

Coppola, D. (2016). U.S. e-commerce market size 2016-2022 | Statista. [online] Statista. Available at: https://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/.

data.nal.usda.gov. (n.d.). *Data Dictionary - Purpose | Ag Data Commons*. [online] Available at: https://data.nal.usda.gov/data-dictionary-purpose#:~:text=Data dictionaries are used to.

Design Thinking @ Haas. (2011). *Sketching User Experiences by Bill Buxton*. [online] Available at: https://divergentmba.wordpress.com/2011/03/14/sketching-user-experiences-by-bill-buxton/ [Accessed 2 Sep. 2022].

Dinarys. (n.d.). How to Integrate a Payment Gateway in Your eCommerce Website. [online] Available at: https://dinarys.com/blog/integrate-payment-gatewa-in-your-ecommerce-website.

Edrawsoft (2019) 'Edraw Max' (Version 10.5. 0) [Computer Program]. Available at: https://www.edrawsoft.com/ad/edraw-max/visio.html?gclid=Cj0KCQjw1a6EBhC0ARIsAOiTkrGCxbEj-odEeTeuPMqqIgaAvMQ2- yoHKfM69mcoti9GKw3hL9igEUaAorcEALw_wcB (Accessed: 25 November 2020)

Engati. (n.d.). Predictive analytics and algorithm: How Netflix & Spotify use it! [online] Available at: https://www.engati.com/blog/predictive-analytics#:~:text=Netflix predictive analytics&text=Netflix uses AI-powered algorithms

Epic Bootstrap. (n.d.). *Epic Bootstrap*. [online] Available at: https://epicbootstrap.com/snippets/shopping-cart [Accessed 6 Sep. 2022].

Georgescu, M. and Georgescu, I.E. (2006). The Emergence of Electronic Payment Systems for the Growth of E-Business. [online] papers.ssrn.com. Available at: https://ssrn.com/abstract=903622

icsid.org. (2021). What percentage of ecommerce businesses fail? [online] Available at: https://www.icsid.org/business/what-percentage-of-ecommerce-businesses-fail/

Interaction Design Foundation (2019). *What is Heuristic Evaluation?* [online] The Interaction Design Foundation. Available at: https://www.interaction-design.org/literature/topics/heuristic-evaluation.

Interaction Design Foundation (2019). *What is User Centered Design?* [online] The Interaction Design Foundation. Available at: https://www.interaction-design.org/literature/topics/user-centered-design.

Kavitha, C. . and Thomas, S. M. (2011) 'Requirement Gathering for small Projects using Agile Methods', IJCA Special Issue on Computational Science - New Dimensions & Perspectives, (Number 3-Article 7).

Kumar, T. and Trakru, M. (n.d.). THE COLOSSAL IMPACT OF ARTIFICIAL INTELLIGENCE IN E-COMMERCE : STATISTICS AND FACTS. [online] International Research Journal of Engineering and Technology. Available at: https://www.irjet.net/archives/V6/i5/IRJET-V6I5116.pdf.

Lynn, R. (2018). *What Is Work Breakdown Structure?* [online] Planview. Available at: https://www.planview.com/resources/articles/work-breakdown-structure/.

Matthews, T., Judge, T., & Whittaker, S. (2012). *How do designers and user experience professionals actually perceive and use personas? Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12.* doi:10.1145/2207676.2208573

Melville, P. and Sindhwani, V. (n.d.). Recommender Systems. [online] Available at: https://www.ime.usp.br/~jstern/miscellanea/seminario/Melville1.pdf.

Microsoft (2020). *Database normalization description - Office*. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description.

MindTools (2021). *Gantt Charts: Planning and Scheduling Team Projects*. [online] www.mindtools.com. Available at: https://www.mindtools.com/pages/article/newPPM_03.htm#:~:text=Gantt charts are useful for.

MineralTree. (2020). Electronic Payments: What Your Business Needs to Know. [online] Available at: https://www.mineraltree.com/blog/electronic-payments-what-your-business-needs-to-know/#:~:text=Electronic payments%2C or e-payments

Mong, M. (2020). *What is a Work Breakdown Structure?* [online] Project Business Technology Resources. Available at: https://www.adeaca.com/blog/faq-items/what-is-a-work-breakdown-structure/.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.

Peinl, P.,
2021. *https://www.researchgate.net/publication/348418140_Teaching_Database_Design_and_the_Importance_of_Integrity* . [online] Research Gate. Available at: https://www.researchgate.net/publication/348418140_Teaching_Database_Design_and_the_Importance_of_Integrity [Accessed 24 August 2022].

Pew Research Center (2021). Mobile Fact Sheet. [online] Pew Research Center: Internet, Science & Tech. Available at: https://www.pewresearch.org/internet/fact-sheet/mobile/. Pineda, M.E. (2017). The difference between purpose, goals, and objectives. [online] Profolus. Available at: https://www.profolus.com/topics/difference-between-purpose-goals-objectives/

Pineda, M.E. (2017). The difference between purpose, goals, and objectives. [online] Profolus. Available at: https://www.profolus.com/topics/difference-between-purpose-goals-objectives/

PMI (2010). *The Value of Project Management*. [online] Available at: https://www.pmi.org/-/media/pmi/documents/public/pdf/white-papers/value-of-project-management.pdf.

Project Management Institute (2018). *Success in Disruptive Times*. [online] Available at: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2018.pdf.

Projectsly. (n.d.). *Importance of Gantt Charts in Project Management*. [online] Available at: https://projectsly.com/importance-of-gantt-charts.

Research, Z.M. (n.d.). Predictive Analytics Market - by Software Solutions (Data Mining & Management, Decision Support Systems, Fraud & Security Intelligence, Financial Intelligence, Customer Intelligence, and Others), By Delivery Mode (Cloud-Based Technology and On-Premise 44 Deployment), By End-User (BFSI, Telecom & IT, Healthcare, Transport & Logistics, Government & Utilities, and Others) and by Application (Customer & Channel, Sales and Marketing, Finance & Risk, and Other Applications): Global Industry Perspective, Comprehensive Analysis And Forecast 2020-2028. [online] www.zionmarketresearch.com. Available at: https://www.zionmarketresearch.com/report/predictive-analytic-market

Review42. (2020). 61+ Ecommerce Fraud Statistics To Help You Stay Safe in 2020. [online] Available at: https://review42.com/resources/ecommerce-fraud-statistics/.

Ruzza, M., Tiozzo, B., Mantovani, C., D'Este, F. and Ravarotto, L. (2017). Designing the information architecture of a complex website: A strategy based on news content and faceted classification. *International Journal of Information Management*, 37(3), pp.166–176. doi:10.1016/j.ijinfomgt.2017.02.001.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. Proceedings of the 2nd ACM conference on Electronic commerce - EC '00. doi:10.1145/352871.352887.

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9_9

Siegel, E. (2015). Seven Reasons You Need Predictive Analytics Today. [online] IBM. Available at: https://www.ibm.com/downloads/cas/LKMPR8AJ.

Statista. (n.d.). Global e-commerce payment fraud losses 2021. [online] Available at: https://www.statista.com/statistics/1273177/ecommerce-payment-fraud-losses-globally/#:~:text=E-commerce payment fraud losses worldwide 2020-2021&text=According to estimates%2C e-commerce.

The NYU Dispatch. (2019). *The Importance of Prototypes in Product Design*. [online] Available at: https://wp.nyu.edu/dispatch/2019/08/30/the-importance-of-prototypes-in-product-design/#:~:text=Prototypes improve the overall understanding of the design&text=Rapid prototypes services help illustrate.

Tuomas, A. (2021). Exploring value in eCommerce artificial intelligence and recommendation systems. University of Jyvaskyla Faculty of Information Technology. Available at:

https://jyx.jyu.fi/bitstream/handle/123456789/75678/URN%3ANBN%3Afi%3Ajyu-202105172951.pdf?sequence=-1&isAllowed=y

Tuteja, M. and Dubey, G., 2012. A research study on importance of testing and quality assurance in software development life cycle (SDLC) models. *International Journal of Soft Computing and Engineering (IJSCE)* , *2*(3), pp.251-257.

Vanneschi, L., Horn, D.M., Castelli, M. and Popovič, A. (2018). An artificial intelligence system for predicting customer default in e-commerce. Expert Systems with Applications, [online] 104, pp.1–21. doi:10.1016/j.eswa.2018.03.025.

Vatsal (2021). Recommendation Systems Explained. [online] Medium. Available at: https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed.

Walker, A. (2019). UML Activity Diagram: What is, Components, Symbol, EXAMPLE. [online] Guru99.com. Available at: https://www.guru99.com/umlactivity- diagram.html.

What is Encryption? | Types of Encryption | Cloudflare UK. (n.d.). *Cloudflare*. [online] Available at: https://www.cloudflare.com/en-gb/learning/ssl/what-is-encryption/.

WP Swings. (2022). How To Design An eCommerce Checkout Flow? [online] Available at: https://wpswings.com/blog/ecommerce-checkout-flow/#:~:text=A checkout flow is a.

www.allaboutcircuits.com. (n.d.). What are Integrated Development Environments? - Technical Articles. [online] Available at: https://www.allaboutcircuits.com/technical-articles/what-are-integrated-development-environments/.

www.dba-oracle.com. (n.d.). *Top-down vs. Bottom-Up Object Database Design*. [online] Available at: http://www.dba-oracle.com/t_object_top_down_bottom_up.htm#:~:text=The top-down method also.

www.oberlo.com. (2021). 19 Ecommerce Statistics You Need to Know in 2021 [New data]. [online] Available at: https://www.oberlo.com/blog/ecommerce-statistics#ecommerce-statistic-1-its-estimated-that-there-will-be-2-14-billion-global-digital-buyers-in-2021

www.tutorialrepublic.com. (n.d.). *Bootstrap Simple Login Form with Blue Background Template*. [online] Available at: https://www.tutorialrepublic.com/snippets/preview.php?topic=bootstrap&file=simple-login-form-with-blue-background [Accessed 6 Sep. 2022].

11 APPENDICES

APPENDIX A – Project Progress Reports

Project progress report 1

UNIVERSITY OF BEDFORDSHIRE

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

FINAL YEAR UG PROJECT: CIS017-3 SEM2

WEEKLY PROGRESS REPORT FORM

Student's Name: Lincia Walters	Supervisor's Name: Mr. Ravi Ragoonath
Month:	Report No. 1

Summary of progress (including any problems)	**Progress I have made to date:** • Reviewed project plan • Familiarized myself with the layout for the reflective report • Researched reflective writing style • Reviewed core and advanced techniques from successful online hardware stores like Nestegg and Amazon for idea generation and to have a better grasp on the requirements • Research the MVC model as it relates to OOP for development of the artefact • Refamiliarized myself with OOP concepts (objects, classes etc) • Researched PHP PDO **Work I've completed/experience acquired:** • Comprehensive list of core and advanced functional requirements • Gantt chart done using online tool • Refreshed knowledge of HTML, CSS and SQL • General understanding of syntax of PHP • Github account created for commits and version control **Challenges encountered:** • Initial deadline dates for completion of programming lectures had to be changed since some deadlines were missed due to work, personal responsibilities and underestimating the learning curve of PHP • Gantt chart to get redone using Microsoft Excel after missed deadlines
Plan for next week	❖ Installation of development tools (XAMPP, VSC, Composer, GIT) ❖ Development of a functional registration page ❖ Development of login page ❖ Do over Gantt chart ❖ Start work on reflective report

Supervisor's comments	Student has made very good start to the unit and seems focused and knows what to do. There is a good balance of attention to the written reports and the artefact as well. Student has been able to resolve the challenges by coming up with creative alternatives.

Project progress report 2

UNIVERSITY OF BEDFORDSHIRE

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

FINAL YEAR UG PROJECT: CIS017-3 SEM2

WEEKLY PROGRESS REPORT FORM

Student's Name: Lincia Walters	Supervisor's Name: Mr. Ravi Ragoonath
Month:	Report No. 2

Summary of progress (including any problems)	**Progress I have made to date:**
	
	• Started work on Reflective Report
	• Development tools installed:
	  o XAMPP
	  o VSC (Visual Studio Code) along with relevant extensions
	  o MySQL Workbench
	  o Composer
	  o GIT
	
	Work I've completed/experience acquired:
	
	• Learnt about version control as it relates to GIT and how to use commits
	  o Installed GIT
	  o Created GIT account
	• Learnt how to use Visual Studio Code
	• Reflective report cover page, introduction and table of contents
	• Created database model and database with "users" table
	
	Challenges encountered:
	
	• GIT was surprisingly more technical than I anticipated so some research was necessary. I spent some more time than was allocated to understanding it.
	• Making the switch from academic writing to reflective report writing needed more research and thoughtfulness than originally anticipated. After some time spent watching YouTube tutorials, I felt a little more confident to tackle it

Plan for next week	❖ Finish section 1, 3 and 4 for reflective report ❖ Construct a login page for artefact ❖ Research responsive design and bootstrap ❖ Design store page

Supervisor's comments	The Progress Report clearly outlines the progress made, work done and challenges encountered along with the plans for the next 2 weeks. The student has made fair progress in the report writing area and artefact development. The student has an ambitious list of core and advanced features and has made some inroads in getting these done however the pace needs to be picked up a bit to ensure timely completion. The student is clearly following a logical process and is demonstrating excellent planning and development skills.

Project progress report 3

UNIVERSITY OF BEDFORDSHIRE

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

FINAL YEAR UG PROJECT: CIS017-3 SEM2

WEEKLY PROGRESS REPORT FORM

Student's Name: Lincia Walters	Supervisor's Name: Mr. Ravi Ragoonath
Month: July	Report No. 3

Summary of progress (including any problems)	**Progress I have made to date:** • Finished section 1, 3 and 4 of reflective report • Installed all development tools o Researched how to use all while becoming familiar • Constructed a login page for artefact • Created ER diagram for database in MySQL Workbench • Created all tables in database **Work I've completed/experience acquired:** • Researched responsive design • Created the "items" and "cart" table in database • Designed store page • Created a functioning login page • Created a functioning registration page with password encryption using the MVC • ER diagram created in MySQL workbench • Learnt normalization which helped with creating an ER diagram **Challenges encountered:** • It was initially quite challenging to get the shop page to echo and reflect database information. I went on to fix it by examining the class's template project and several video guides. • Learning normalization was interesting but took longer than anticipated

Plan for next week	❖ View tutorials on sections 2, 5 and 6 for reflective report and start write up
	  o Section 2 - Summary of progress to date
	  o Section 5 - Conclusion
	  o Section 6 - References
	❖ Review and edit section 1 and 2 of reflective report
	❖ Design details page
	❖ Create cart functions to add, remove, update and get cart

Supervisor's comments	The Progress Report clearly outlines the progress made, work done, and challenges encountered along with the plans for the next 2 weeks. The student has made excellent progress in the report writing area and artefact development. The pace of development of the artefact needs to be picked up a bit to achieve the full list of requirements that the student intend to achieve. The student is clearly following a logical process and is demonstrating excellent planning and development skills.

APPENDIX B – Requirements Analysis (Functional and Non-functional requirements)

Functional Requirements

Requirement: Log in/Log out	
Number	FR2
Description	After registering on the website, the user needs to log into his newly established account to make a transaction. The user must enter their registered email address and password to proceed. The program verifies that the email address is correct, and that the password is consistent with what is stored in the application's database. After the information has been verified against the database's "users" record, the user is then rerouted to the login page.
Rationale	Once logged in, a user may make secure purchases, examine their order history and respective order details.
Requirement: Order history/Order details	
Number	FR3
Description	A user needs to register and have an account in order to access account features like order history and order details. Users who are guests cannot utilize this functionality.
Rationale	Enables the buyer to access up-to-date details on each online purchase, including payment information, products ordered and transaction dates.
Requirement: Cart management – Add to cart, edit cart, remove from cart	
Number	FR4
Description	A registered user will have the ability to add, edit, update, and delete a product that have been placed in their shopping cart. It keeps track of the products the buyer has decided to buy later.
Rationale	Gives the user the option to edit their cart in the event of a mistake. It enables the user to choose wisely while making a purchase. A better user experience is offered since the user may change their mind after adding something to their cart or change their mind about the quantity of a particular product.
Requirement: Product detail	
Number	FR5
Description	This feature offers a more thorough look of a product, along with options for size, specifications, and other attributes.
Rationale	Gives a user additional detail about a desired product to empower them to make a better buying decision.
Requirement: Product listing/Store Catalog	
Number	FR6
Description	The products page is one of the primary pages of any ecommerce website. This page includes a list of all the products offered, together with information about them, including descriptions, costs, pictures, brands, and other facts that can persuade a visitor to buy.
Rationale	This lists all the products that are for sale and that you may add to your shopping basket to buy, along with their pricing and descriptions.

Requirement: Checkout flow	
Number	FR7
Description	The checkout flow is the process a customer goes through on an e-commerce site to buy the products in their shopping cart.
Rationale	Good checkout flow enhances conversions, decreases checkout abandonment, and produces a wonderful user experience.
Requirement: Search, sort, filter	
Number	FR8
Description	The search bar enables users to look up certain products in the database. A notification is given if a specific product cannot be located; or else, the user is directed to the product page for their search result. The sort and filter menu will sort by category and price range.
Rationale	Without having to browse the complete product listing, a user should be able to find the things swiftly and easily they're looking for via a search bar. Further to this, the sort and filter will save the customer time narrowing down products that they are looking for without spending time browsing the entire store catalog.
Requirement: Informational Pages	
Number	FR9
Description	Informational pages is meant to supply information that might be needed for customers experiencing technical overwhelm while trying to make a purchase. This will also help with establishing authority and expertise.
Rationale	Strategically written in an effort to not only assist the customer but upsell current products in the store catalog by hyperlinking to relevant products.
Requirement: Online payment method	
Number	FR10
Description	The customer can securely enter their credit card information while using the online payment option to make purchases. Only logged-in users have access to this feature.
Rationale	Customers must have secure checkout and payment options in order to buy things from the website.
Requirement: Recommender systems/Predictive Analytics	
Number	FR11
Description	Using data from past purchases, this tool will recommend products to customers.
Rationale	This is done to persuade and guide people to buy more products, hence boosting sales.
Requirement: Chatbot	
Number	FR12
Description	By responding to straightforward questions, this ChatBot tool assists consumers and reduces the time it takes for a human agent to respond. The user will be forwarded to an agent if the ChatBot is unable to respond to a question.
Rationale	By offering a constant mode of contact, this function enhances customer service.
Requirement: Product inventory management – Add product, Update product, Delete product	
Number	FR13
Description	The user can add, update, or remove goods from the database using the admin menu after logging in as an admin user.
Rationale	To manage the product database, the administrator can add, update or remove products that are changed or out of stock. Additionally, it is utilized to modify product details like descriptions, pricing, and even photos.

Table 14: Requirements detail for functional requirements

Non-functional requirements

Requirement: Security – Encryption	
Number	NFR1
Description	The system will automatically encrypt the user password when a user registers, storing it as an odd string of characters that no one can decipher in the database.
Rationale	This feature will protect customer's data from getting into the hands of bad actors.
Requirement: Usability – Responsive design	
Number	NFR2
Description	The application's display will be able to automatically adjust to fit on various displays thanks to responsive design. For example, laptop, tablets, mobile devices.
Rationale	This feature puts the needs of the user first, makes it simple for them to interact with the site thereby fostering customer loyalty.
Requirement: Usability – Browser compatibility	
Number	NFR3
Description	The capability of a website to appear completely operational on various browsers on the market.
Rationale	It makes sure that a big portion of the target market are not being alienated if the artefact does not function properly on their browser-OS.
Requirement: Scalability	
Number	NFR4
Description	The capability of an artefact to maintain functionality when its size or volume are modified to accommodate increased user demand.
Rationale	Reduces the possibility of the artefact slowing down with seasonal bursts of traffic or increased demand.
Requirement: Maintainability	
Number	NFR5
Description	This feature is concerned with how simple it would be to change, update, fix or extend the artefact.
Rationale	A maintainable artefact will save money as the time and effort required by developers to maintain a system directly affects the maintenance budget.

Table 15: Non-functional requirements detail

APPENDIX C – Data Dictionary

Products				
Attribute	**Data Type**	**Field Length**	**Constraint**	**Description**
product_id	integer	11	Primary key	ID for product
product_title	varchar	100	Not null	Name of product
product_description	text	1000000000	Not null	Description of product
product_price	decimal	9	Not null	Cost of product
product_discount_amount	decimal	9	None	Discount on product
product_quantity	integer	11	Not null	Quantity of product
product_image1	varchar		not null	Image of product
product_image2	varchar	100	Not null	Image of product
product_image3	varchar	100	Not null	Image of product
product_image4	varchar	100	Not null	Image of product
product_created	datetime	19	Not null	Date product was enlisted
product_status	varchar	20	Not null	Availability of product
product_catergory	varchar	50	Not null	Category that is suitable for product
Orders				
Attribute	**Data Type**	**Field Length**	**Constraint**	**Description**
order_id	integer	11	Primary key	ID for the order
user_id	integer	11	Foreign key	User ID of the user; auto increment
subtotal	Decimal	8,2	Not null	Subtotal cost for order
total	Decimal	8,2	Not null	Total cost for order
total_discount_amount	Decimal	8,2	Not null	Discount on order
order_created	datetime		Current_timestamp()	Date order was created
payment	Varchar		Not null	
payment_id	varchar	11	Not null	ID for payment

Order Details				
Attribute	**Data Type**	**Field Length**	**Constraint**	**Description**
order_details_id	text		Primary key	Details of order
order_id	integer	11	Foreign key	ID for the order
product_id	integer	11	Foreign key	ID for the products
order_details_price	double	8,2	Not null	Total cost on order
order_details_quantity	integer	11	Not null	Quantity of items on order
order_details_created	datetime		Current_timestamp()	Date order details was created
Cart				
Attribute	**Data Type**	**Field Length**	**Constraint**	**Description**
cart_id	integer	11	Primary key	Cart ID
user_id	integer	11	Foreign key	User ID of the user; auto increment
product_id	integer	11	Foreign key	Product ID
cart_created	datetime		Current_timestamp()	Date cart was created
cart_quantity	integer	11	Not null	Amount of items in cart
cart_status	varchar	200	‘cart’ or ‘purchased’	Status of cart

Table 16: Data dictionary

APPENDIX D – Implementation of Nielsen's Heuristics

Principle 2: Match between system and the real world	
Description	Instead of using system-oriented jargon, the system should employ words, phrases, and concepts that are recognizable to the consumers. Ensure that information appears in a logical and natural arrangement by adhering to real-world norms.
Implementation	Informal, familiar language will be used throughout the artefact. Each product will have relevant descriptions.
Principle 3: User control and freedom	
Description	Users frequently choose system options by accident, and they will require an obvious "emergency exit" to get out of the undesirable condition without having to engage in a lengthy conversation. Support undo and redo.
Implementation	Users will be able to review their respective carts to ensure no buying errors. They will also be able to delete products or remove from their respective carts.
Principle 4: Consistency and standards	
Description	Users should not have to speculate if certain expressions, circumstances, or actions are equivalent. Observe platform customs.
Implementation	Familiar call-to-action (CTA) buttons will be used such as "buy now" or "check out". A standard products page template will be used.
Principle 5: Error prevention	
Description	Good error messages are great, but smart design that avoids an issue from happening in the first place is much better. Eliminate error-prone situations or check for them and provide consumers the chance to affirm before committing to an activity.
Implementation	Form fields will be programmed to only accept relevant information.
Principle 6: Recognition rather than recall	
Description	Reduce the amount of memory required from the user by making objects, actions, and options visible. It shouldn't be necessary for the user to recall details from one section of the dialogue to the next. When necessary, instructions for using the system should be readily visible or accessible.
Implementation	The products page will be designed using a grid layout. The log in page will be on the top right of the homepage screen.
Principle 7: Flexibility and efficiency of use	
Description	In order for the system to serve both beginner and experienced users, accelerators—unseen by the novice user—may often speed up the interaction for the expert user. Permit users to customize routine tasks.
Implementation	Form fields will have autofill. Users will be able to edit form fields and use relevant keyboard shortcuts.
Principle 8: Aesthetic and minimalistic design	
Description	Information that is unnecessary or infrequently used should not be included in dialogues. Each additional piece of information in a conversation competes with the pertinent pieces and reduces their relative exposure.
Implementation	Proper use of white space was used to design the artefact. Product descriptions was revised to not include any superfluous information.

Principle 9: Help users recognize, diagnose, and recover from errors	
Description	Error messages ought to be written in simple English (without the use of codes), identify the issue clearly, and offer a suitable alternative.
Implementation	Error messages are in plain, natural language. Prompts are concise and straightforward.
Principle 10: Help and documentation	
Description	Although it is preferable if users can operate the system without assistance, assistance and documentation may still be required. Any such material ought to be concise, focused on the user's goal, easy to search, and describe specific actions that need to be taken.
Implementation	A shopping tools page will be included to assist the user with queries. A chatbot will be included for an interactive conversation regarding queries.

Table 17: Implementation of Nielsen’s Heuristics

APPENDIX E – Wireframes

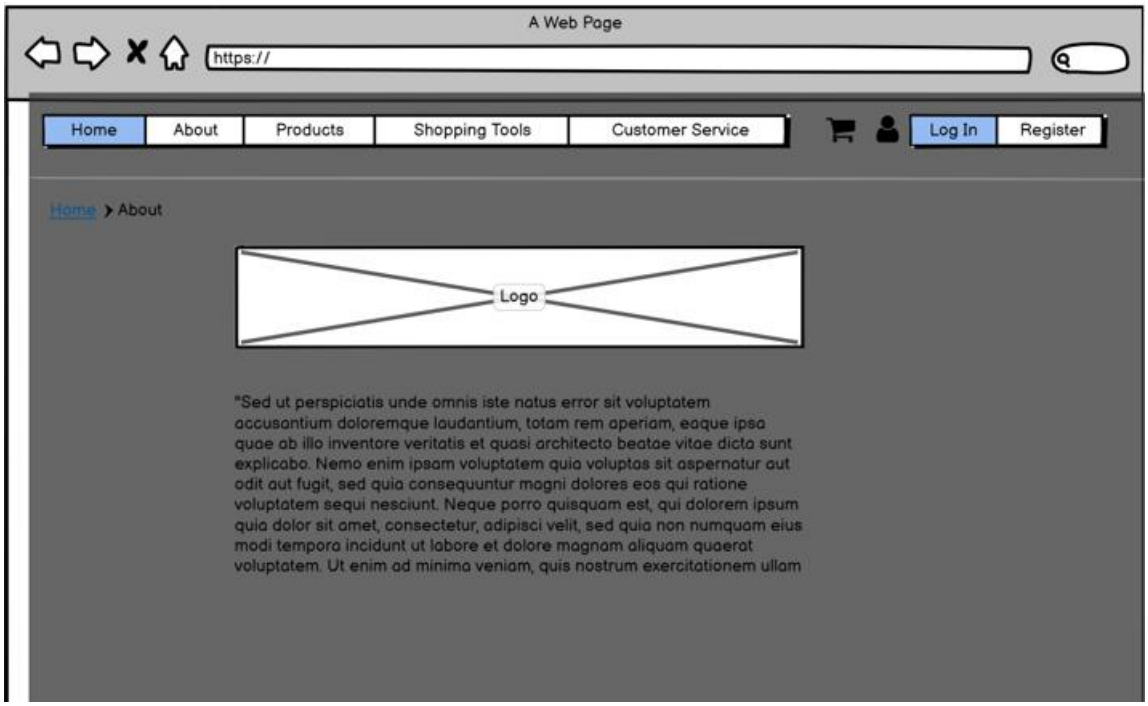


Figure 73: About us page wireframe

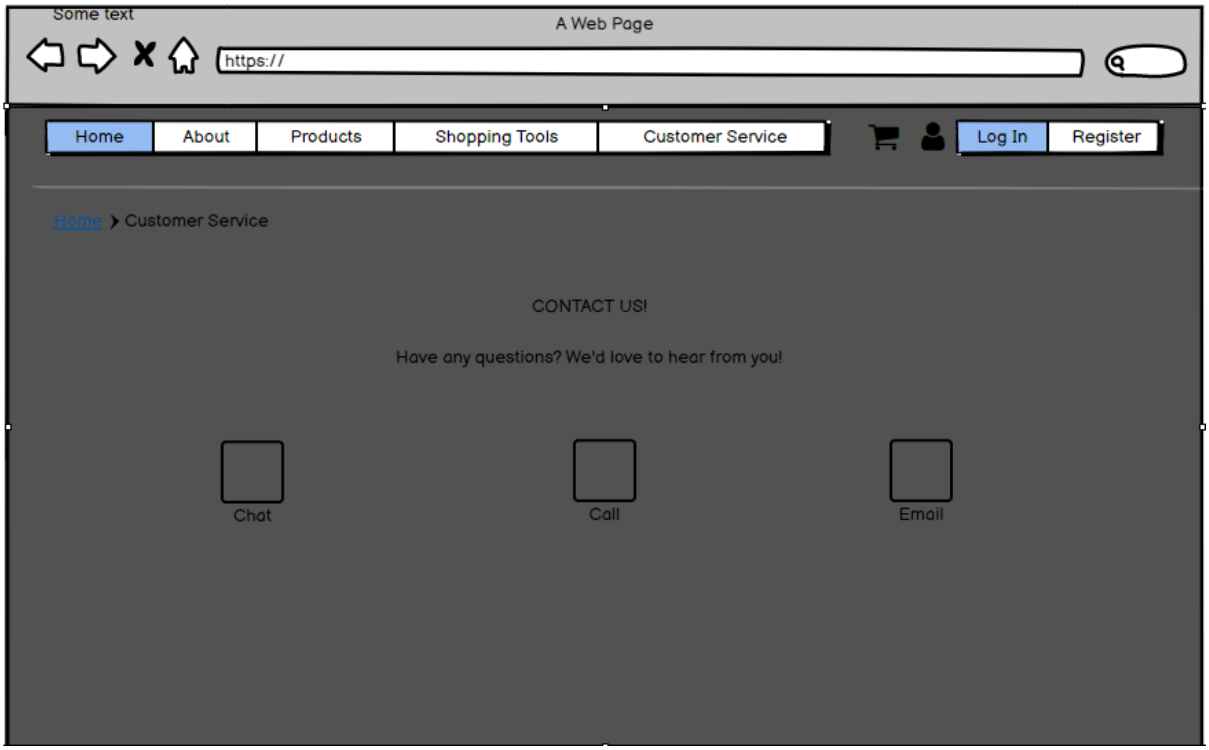


Figure 74: Customer service page/Contact us page wireframe

Figure 75: Log In page wireframe

APPENDIX F – Functional Requirements Testing

FR2 - Login

Test #	Test Description	Expected Outcome	Actual Outcome	Result
FR2.1	Login with no information inserted in fields	A message asking to please fill out require fields	A message asking to please fill out require fields	Successful
FR2.2	Login customer Lincia Walters with email address lincia.walters@hotmail.com and password: password	Customer successfully logged in with alert displayed. Redirected to home page. Navigation bar change to reflect user logged in with log out tab, cart and search bar.	Customer successfully logged in with alert displayed. Redirected to home page. Navigation bar change to reflect user logged in with log out tab, cart and search bar.	Successful
FR 2.3	Login customer Lincia Walters with email address lincia.walters@hotmail.com and password: password1	Error message indicating wrong info inserted. Please try again.	Error message indicating wrong into inserted. Please try again.	Successful
FR 2.4	User clicks on link 'Don't have an account? Sign up here!' redirects user to registration screen	User redirected to registration page	User re-directed to registration page	Successful

Table 18: FR2 testing

FR3 – Order history/order details

Test #	Test Description	Expected Outcome	Actual Outcome	Result
FR 3.1	User clicks on orders on account menu and is routed to orders page	User is taken to order page	User is taken to order page	Successful
FR 3.2	User clicks on "details" link and is routed to order details for respective product	User is routed to order details for selected order	User is routed to order details for selected order	Successful

Table 19: FR3 testing

FR4 – Cart management – Add to cart, edit cart, remove from cart

Test #	Test Description	Expected Outcome	Actual Outcome	Result
4.1	Add to cart button and cart option in navigation bar with a registered user working	Only a registered user can add items and view their cart	Only a registered user can add items and view their cart	Successful
4.2	Add to cart button redirects guest user to login page	Add to cart button redirects user to login page	Add to cart button redirects user to login page	Successful
4.2	Items can be removed from cart	Remove items button removes the items from cart	Remove items button removes the items from cart	Successful
4.3	Cart can redirect registered user to checkout page	Check out button redirects user to checkout page	Check out button redirects user to checkout page	Successful
4.4	Price of items in cart are added together correctly	Subtotal Shows correct value of all items in cart	Subtotal shows correct value of all items in cart	Successful

Table 20: FR4 testing

FR5 – Product detail

Test #	Test Description	Expected Outcome	Actual Outcome	Result
5.1	Go to product detail with a register user	Product information displayed	Product information displayed	Successful
5.2	Got to detail page using as a guest user	Product information displayed	Product information displayed	Successful

Table 21: FR5 Testing

FR6 – Product listing/store catalogue

Test #	Test Description	Expected Outcome	Actual Outcome	Result
6.1	Guest user can view product catalog	Product catalog displayed	Product catalog displayed	Successful
6.2	Registered user can view product catalog	Product catalog displayed	Product catalog displayed	Successful

Table 22: FR6 testing

FR7 – Checkout flow

Test #	Test Description	Expected Outcome	Actual Outcome	Result
7.1	Registered users can checkout	Only registered users can checkout. Guests redirected to login page if add to cart button is clicked for checkout.	Only registered users can checkout. Guests redirected to login page if add to cart button is clicked for checkout.	Successful
7.2	Registered user can complete checkout	Registered user will be redirected to stripe page after clicking checkout button.	Registered user will be redirected to stripe page after clicking checkout button.	Successful

Table 23: FR7 testing

FR8 – Search, sort and filter

Test #	Test Description	Expected Outcome	Actual Outcome	Result
8.1	Query placed for item on store page in search bar on the navigation bar for anyone	Search results displayed based on keyword input	Search results displayed based on keyword input	Successful
8.2	Query placed for item not on store page for anyone	Error message displaying no products found	Error message displaying no products found	Successful
8.2	Filter products applied to filter bar on store page by category, price max or price min and order by sorting	Filter results displayed based on criteria chosen	Filter results displayed based on criteria chosen	Successful

Table 24: FR8 testing

FR9 – Informational pages

Test #	Test Description	Expected Outcome	Actual Outcome	Result
9.2	Informational pages on navigation bar redirects to respective page	Chosen informational page displayed	Chosen informational page displayed	Successful
9.1	Anyone can view informational pages	Informational pages displayed	Informational pages displayed	Successful

Table 25: FR9 testing

FR10 – Online payment method

Test #	Test Description	Expected Outcome	Actual Outcome	Result
10.1	User can enter credit card info to pay for item	Transaction successful, removes item from cart and successful alert displayed	Transaction successful, removes item from cart and successful alert displayed	Failed

Table 26: FR10 testing

FR11 – Recommender systems/Predictive analytics

Test #	Test Description	Expected Outcome	Actual Outcome	Result
FR 11.1	Product detail page displays recommended products at bottom	Products displayed in recommended products area	Products displayed in recommended products area	Successful

Table 27: FR11 testing

FR12 – Chatbot integration

Test #	Test Description	Expected Outcome	Actual Outcome	Result
12.1	Chatbot accessed on all pages	Chatbot can be interacted it with on all pages	Chatbot can be interacted it with on all pages	Successful
12.2	Chatbot interacts in a conversational manner	Chatbot responds based on key words used	Chatbot responds based on key words used	Successful

Table 28: FR 13 testing

FR13 – Product Inventory Management

Test #	Test Description	Expected Outcome	Actual Outcome	Result

Table 29: FR13 testing

APPENDIX G – Non-Functionality Requirements Testing

NFR2 – Usability – Responsive Design

Test #	Test Description	Expected Outcome	Actual Outcome	Result
NFR2.2	User uses the artefact on various size screen devices.	Artefact pages automatically adjust content to suit device resolution	Artefact pages automatically adjust content to suit device resolution	Successful

Table 30: NFR2 testing

NFR3 – Database Integrity – Database Security

Test #	Test Description	Expected Outcome	Actual Outcome	Result
NFR3.1	Database access should only be made through the artefact's the interface	Users must not access the database, whether registered and unregistered	ONLY the artefact interface has access to the database's information for users	Successful

Table 31: NFR3 testing

NFR4 – Scalability

Test #	Test Description	Expected Outcome	Actual Outcome	Result
4.1	In order to accommodate updates and significant improvements in the future, the system should be built to be expandable.	The artefact has the capacity to register a sizable user base and maintain a sizable amount of data related goods and services.	With more users and products, the system operates without any issues.	Successful

Table 32: NFR4 testing

NFR5 - Maintainability

Test #	Test Description	Expected Outcome	Actual Outcome	Result
5.1	The artefact should be easy to change, enhance and restructure system over time	The artefact is easy to change, enhance and restructured over time	The artefact is easy to change, enhance and restructured over time	Successful

Table 33: NFR5 testing

APPENDIX H – Screenshots of artefact

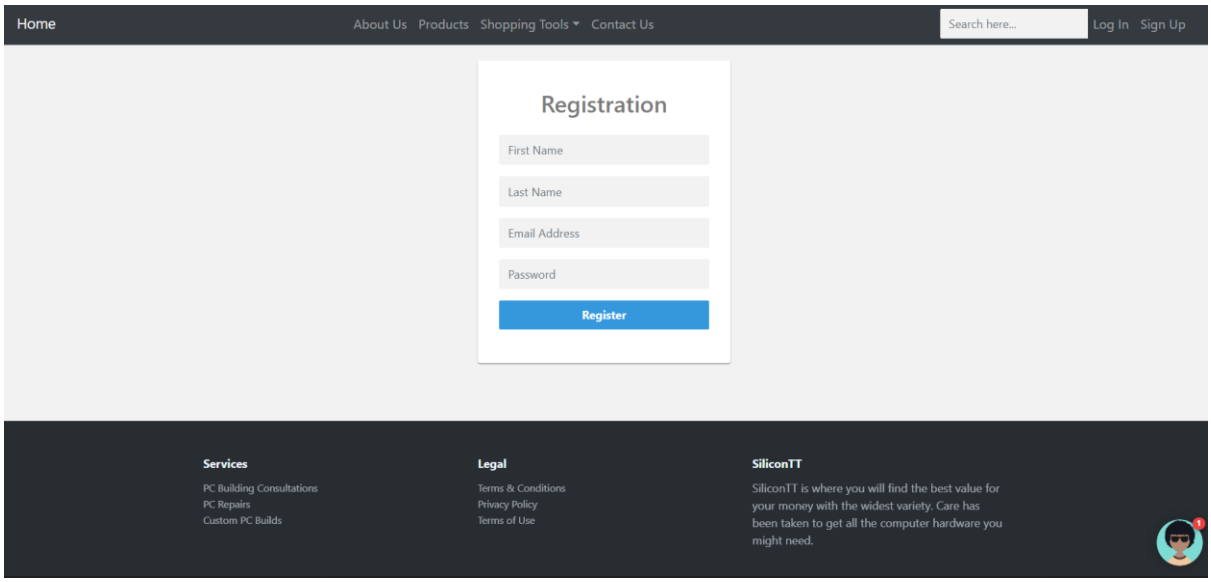


Figure 76: Registration page

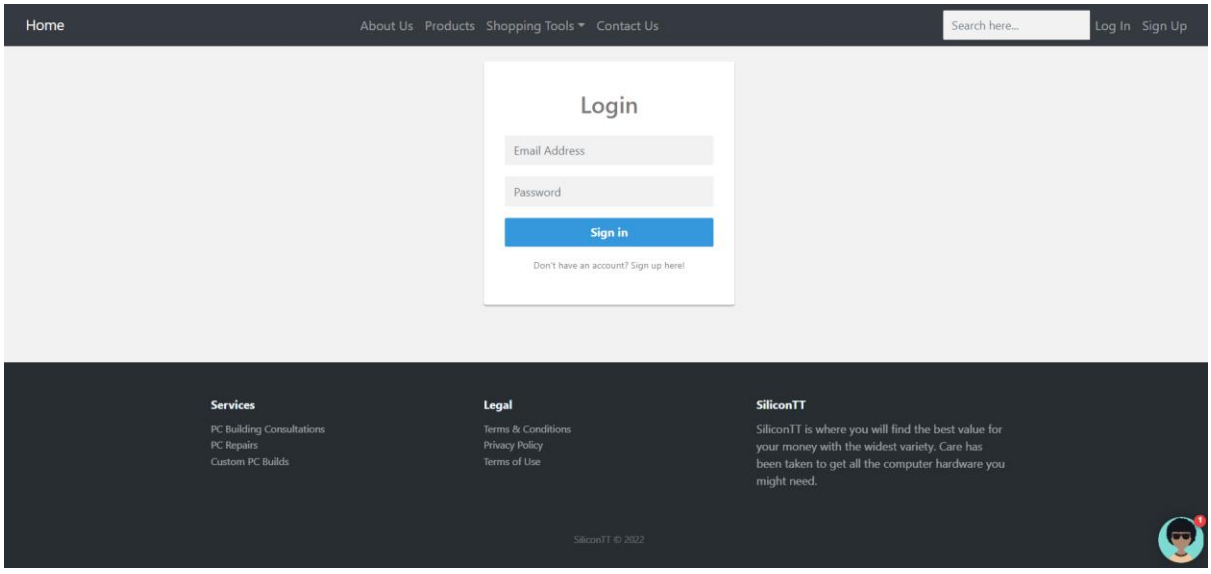


Figure 77: Login Page

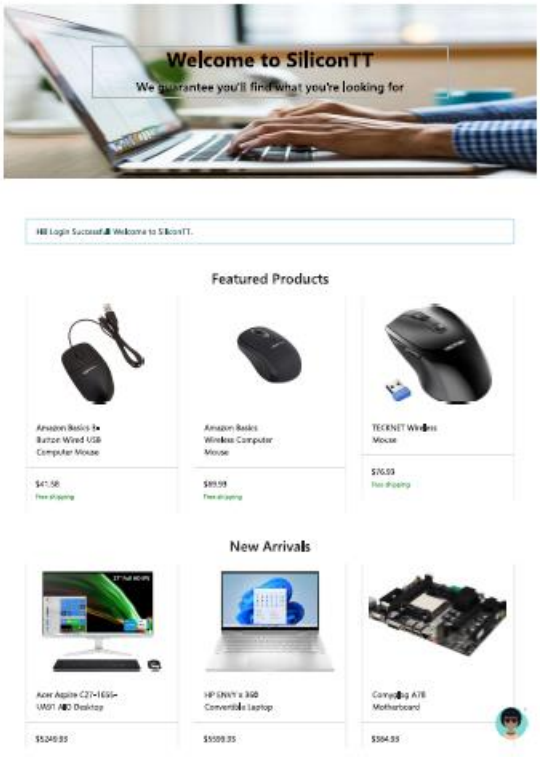


Figure 77: Homepage

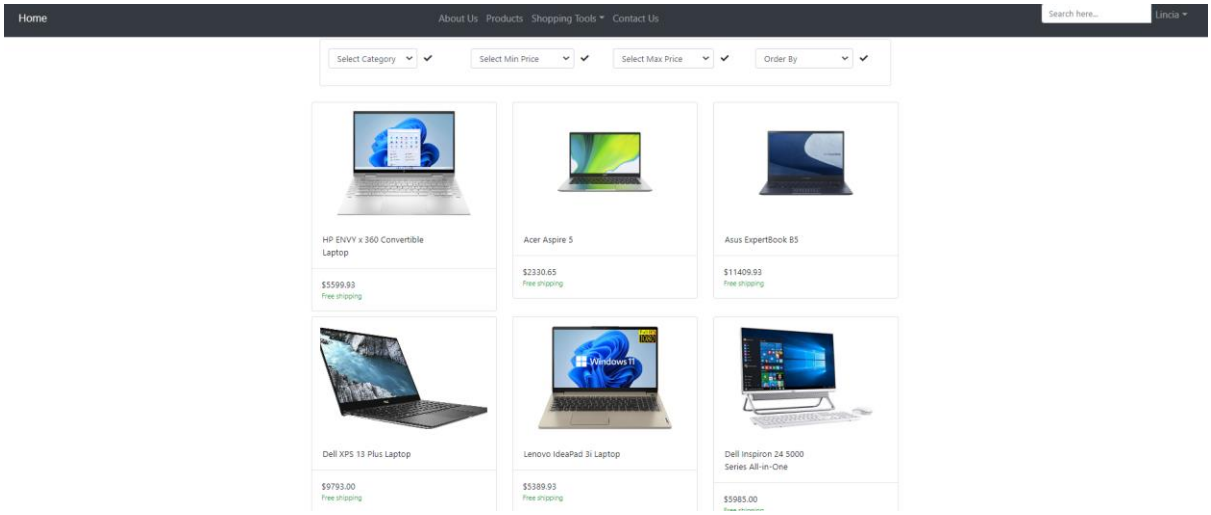


Figure 78: Store page/store catalogue

Figure 79: Product detail page

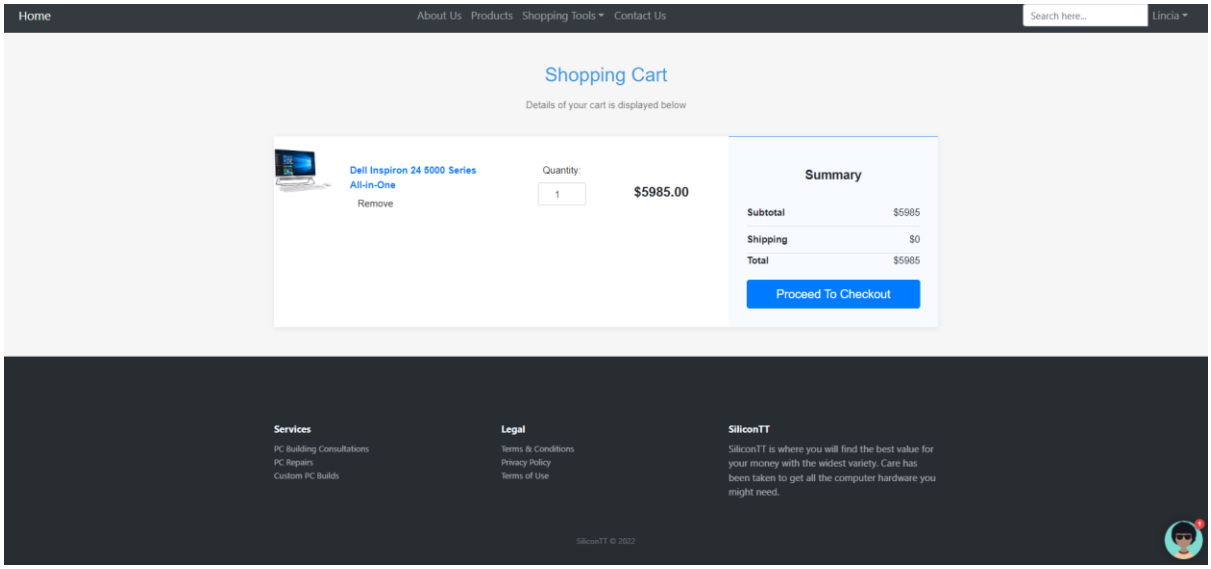


Figure 80: Cart page

APPENDIX I – Poster

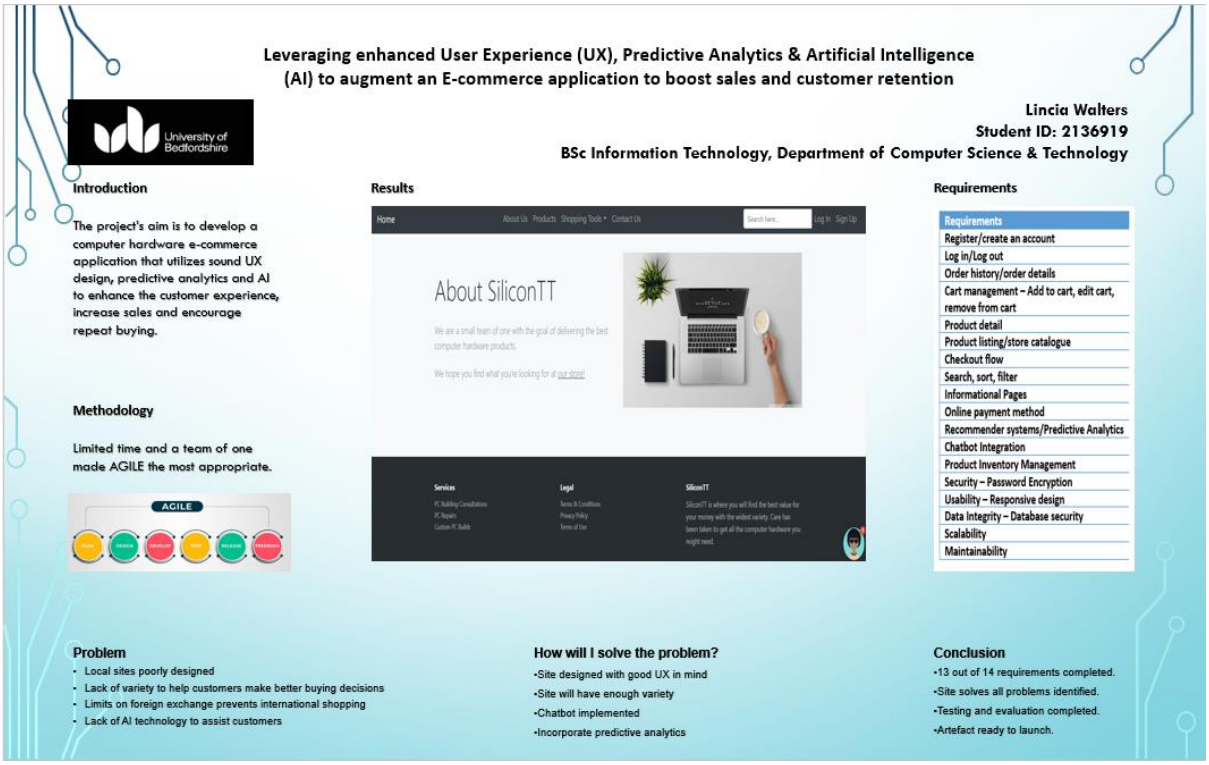


Figure 81: Poster of the artefact