

Department of Computing
Goldsmiths, University of London

Inventory Check

Final Report

by

Hardik Ramesh

June 2020

Submitted in partial fulfillment for the degree
of
Bachelor of Science in *Computer Science*

Abstract

Inventory check is a web-based application for managing the inventory for the business. The user needs to add all the product information manually using the add product form and all this data will be stored in the database securely. The application utilises six various technologies such as HTML, CSS, JavaScript, PHP, MySQL and API.

The use of inventory system is widely used by retailers/other industry and research shows that every single business use this system to manage their inventory such as eBay, Amazon, Sainsbury's, Tesco, etc. This has allowed me to take completely new approaches by solving the current problems using the system as the inventory management system widely used by a big company and no one has thought about the smaller business who cannot afford to pay big fees every year, therefore, I have made a suitable, secure and easy to use the system for smaller business and there will be no fees to use my system. This system will make it easier for them to track down the out of the stock item, analysis the profit, sales, check the number of item in stocks, fast selling item and more.

By developing a system that allows the user to see the data in real-time with database and Internet technology, this project shows the potential of use in other industries, not only in retail. Prototyping and design took place with different databases servers such as XAMPP or MongoDB, concluding with using XAMPP to implement the system.

Acknowledgements

I am extremely grateful to my supervisor, Dr. Edward Anstead who has given me advice, guidance, and encouragement during the project. All the comments given by the supervisor was valuable for me to improve my project and my report. Furthermore, I would like to thank my uncle who has been working closely with me during this project and giving me useful ideas as well as what their requirement in the program. His support was very useful for me to make the correct program as need in his business. I also like to thanks my wonderful parents who always cared for, loved, and supportive of all the work I do in my life.

Finally, to my friends especially Mr. JayKumar Halpati who has helped me with the user testing during the difficult time where the country was closed down due to COVID 19 virus.

Nomenclature

<i>API</i>	Application Program Interface
<i>CDN</i>	Content Delivery Network
<i>CSS</i>	Cascading Style Sheet
<i>HTML</i>	Hypertext Markup Language
<i>MySQL</i>	Relational Database Management System
<i>PHP</i>	Personal Home Page
<i>POS</i>	Point of Sale
<i>UI</i>	User Interface
<i>XAMPP</i>	Cross-Platform(X), Apache(A), MariaDB(M), PHP(P) and Perl(P)

Contents

Abstract	i
Acknowledgements	ii
Nomenclature	iii
Contents	iv
List of Figures	vii
1 Introduction	1
1.1 Motivation & Aims	1
1.2 Purpose & Scope	2
1.3 Section Overview	2
2 Background Research	4
2.1 Existing Systems	4
2.1.1 Dear Systems	5
2.1.2 Cin7	5
2.1.3 Unleashed	5
2.2 Back-end Software	6
2.2.1 MongoDB	6
2.2.2 MySQL	6
2.3 Libraries	6
2.3.1 jQuery	7
2.3.2 Bootstrap 4	8
2.4 Software Architecture	9
2.4.1 Brackets	9
2.4.2 Sublime Text	9

CONTENTS

2.5	Questionnaire/Survey Result	9
2.6	Project Aims	13
2.6.1	Front-end	13
2.6.2	Server-side (PHP)	14
2.6.3	Database server	14
3	System Design	15
3.1	System Requirement	15
3.1.1	Functional Requirement	15
3.1.2	Non-Functional Requirement	16
3.1.3	Database Requirement	16
3.1.4	PHP Requirement	16
3.2	Use Case Diagram	17
3.3	Wireframe	18
3.3.1	Wireframe 1	18
3.3.2	Wireframe 2	20
3.3.3	Final outcome	24
3.4	Side Map	28
4	Implementation	29
4.1	Front-End Development	29
4.1.1	HTML	29
4.1.2	CSS	30
4.1.3	JavaScript/jQuery	31
4.2	PHP	31
4.2.1	Database Connection	32
4.2.2	View all product	32
4.2.3	Add product	33
4.2.4	Edit product	34
4.2.5	Delete product	35
4.2.6	Search product	36
4.2.7	Analyse product	37
4.3	MySQL	38
4.4	Telegram API	39
4.5	Gantt Chart	40

CONTENTS

5 Testing	41
5.1 Normal Testing	41
5.2 User Testing	43
5.3 Deployment	45
6 Conclusion	46
6.1 System Successes	46
6.2 System Failures	47
6.2.1 Register	47
6.2.2 Login	47
6.2.3 Mobile Friendly (Responsive)	48
6.2.4 No alert for Internet connection	48
6.2.5 Alert for Delete product	48
6.3 Future Developments	48
6.3.1 Login & Register	48
6.3.2 More charts	49
6.3.3 Filter Features	49
6.3.4 POS system	49
6.3.5 Barcoding	49
6.3.6 Alerts	49
6.3.7 Responsive site	50
6.3.8 Alert for delete product	50
6.3.9 PDF file	50
6.3.10 Protect user data	50
6.3.11 Required hardware	50
6.4 Evaluation	50
Appendix A Questionnaire/Survey Result	52
Appendix B Gitlab program link	55
Appendix C How to run the program?	56
Bibliography	57

List of Figures

2.1	Contact form functionality using jQuery API	7
2.2	Navbar using bootstrap framework	8
2.3	Question 1: Use any system	10
2.4	Question 2: choose of system	10
2.5	Question 3: cost of current system	11
2.6	Question 4: Usage of the system	11
2.7	Question 5 & 6: Use the system for free and the ability to manage online	12
2.8	Question 7: Prefer to use my system or other	13
3.1	Use Case Diagram	17
3.2	Wireframe 1 - Homepage	18
3.3	Wireframe 1 - Product	19
3.4	Wireframe 1 - Analytic	19
3.5	Wireframe 1 - Contact us	20
3.6	Wireframe 2 - Homepage	20
3.7	Wireframe 2 - Product	21
3.8	Wireframe 2 - View all product	21
3.9	Wireframe 2 - Add product	22
3.10	Wireframe 2 - Edit product	22
3.11	Wireframe 2 - Analytic	23
3.12	Wireframe 2 - Contact us	23
3.13	Final UI - Homepage	24
3.14	Final UI - Product	25
3.15	Final UI - View all product	25
3.16	Final UI - Search Result	26
3.17	Final UI - Edit Product	26
3.18	Final UI - Add Product	27

LIST OF FIGURES

3.19 Final UI - Analytic	27
3.20 Final UI - Contact us	28
3.21 Side Map	28
4.1 Example of HTML code	30
4.2 Example of CSS code	30
4.3 Example of JavaScript code	31
4.4 Connection to the server	32
4.5 allproduct.php	33
4.6 insertitem.php	34
4.7 edit.php	35
4.8 delete.php	36
4.9 search.php	37
4.10 analytic.php	38
4.11 Product Table	38
4.12 Register/Login Table	39
4.13 Example of API code	39
4.14 Gantt Chart	40
A.1 Question 1,2 and 3	52
A.2 Question 4,5 and 6	53
A.3 Question 4,5 and 6	54

Chapter 1

Introduction

Nowadays, every single industry uses the inventory management system to manage its inventory. The most leading company that has adopted this feature in the system are dear systems, Cin7, unleashed, etc who have been charging fees between £149 - £200. By involving a higher fee the small business cannot afford to pay for their service. Therefore, I am building a system to help a small business like a corner shop with no extra costs.

1.1 Motivation & Aims

The aim of this project is to help small business specially corner shop to manage their inventory. From my personal experience, while working in my uncle's shop who uses a spreadsheet to manage his entire inventory, and sometimes he does not even know if he has run the item or not. He always goes around the shop to check which item has been out of stock and manually enter in a piece of paper to get them from cash & carry(Wholesaler) or supplier. The current system they are using not time efficient therefore I want to help my uncle and other local shops like him to use my system to save time and money. This system will require an internet connection to get the data from the database and shows on user browser.

This concept has various applications to other scenarios such as using a bar code to find a piece of item information using a scanner. It could also use machine learning to identify product availability in the inventory using the camera. Inventory check was created using HTML, CSS, JavaScript, PHP,

MySQL, and API.

1.2 Purpose & Scope

The purpose of this project is to give a solution to a local shop to manage their inventory stock. The important part of this project is achieved by adding a product, edit product, delete a product, view all products, use the search bar to find the specific item, and analyse the data which will be displayed in bar chart/table.

1.3 Section Overview

This report will contain six main sections:

1. **Introduction:** This section will outline the motivation, aims, purpose, and scope of the project.
2. **Background Research:** Analysing the existing system, technical research to find useful technology to use in this project, public survey, Software architecture, and back-end software.
3. **System Design:** This section consists of system requirements, use case diagram, wireframe(UI), side map, and accessibility features of the application.
4. **Implementation:** Discussion of implementation with Front-end development, PHP, MySQL, Telegram API, and Gantt chart. This shows how this technology helps me build this system.
5. **Testing:** A formal testing was done to see how the system was performing. This was done with Normal testing, User testing, and deployment.
6. **Conclusion:** This session evaluates the success of the implementation which was discovered in background research. It analyses the system successes, system failure, and future development that could be happening in future development.

In addition, this report contains a number of different appendices:

CHAPTER 1. INTRODUCTION

1. **Appendix A** - Full list of responses that was gathered throughout the public questionnaire.
2. **Appendix B** - This will show you where you find the inventory check code file from GitLab.
3. **Appendix C** - A list of instructions to run the server and application to view the website.

Chapter 2

Background Research

This chapter will show the research that is undertaken before developing and designing the system. The main focus will be technical research as it will allow me to find useful technologies that can be used to build the system. I will be analysing the existing system which will show what they are offering with their current system to their customers and also a public survey where it will show me if they are happy with the current system or what features they required in the new system. Through the use of an inventory check, this concept can provide an interactive solution for a local shop that cannot afford to pay higher fees.

2.1 Existing Systems

Before development, this is one of the most crucial steps that need to be taken in every project, therefore this will show us what is currently available in the market and what needs to be changed. To get the right aims and objective for the project, I have analysed a different system that is currently in the market. Most of the software has a similar feature with a decent User Interface(UI) rather than having an interactive UI where the user can feel more comfortable using the system. By analysing the current system will help me to find gaps in the design or functionality. Furthermore, all the feature they are available in the market are useful and I am making the same system with different design and service for local shopkeeper without monthly cost.

I have inspected various existing systems that are available in the market

such as Dear Systems, Cin7, and Unleashed.

2.1.1 Dear Systems

Dear Systems is an inventory software that offers a point of sale(POS), Inventory management, financial management, and warehousing. The POS system that they offer is compatible with Android, PC, or Mac. Overall I like the system and services they offer but when you check out their monthly cost where they don't give this service to small business with 1 user. The cost for this system is \$199 per month and \$2189 per annual with 5 users.

2.1.2 Cin7

Cin7 is one of the most popular management systems that are available in the market. It is fully integrated with point of sale (POS) and inventory management system where they have a real-time tracking system on inventory and sales. It is very difficult to tell the exact cost of the software as the company only gives the price by quote. The system comes with the manned tills where their software is installed and connected with the inventory. The system also has reporting features where the user can check the inventory level, top-selling item, profit, losses, cost of goods, etc. Overall the color theme of the UI seems boring as the users only interact with the site when the application has good UI. It is hard to tell what will be the monthly cost but some websites claim it starts from £199 per month for basic users. If I were a shop keeper who owns a retail business then I would not afford to pay the cost.

2.1.3 Unleashed

Unleashed has to be one of the most visited software used by big businesses with the various warehouse. It allows the user to get access to real-time data on stocks, costs, and tracking information. Real-time tracking gives accurate information on the user screen where you look at the item. They offer three different plans such as Medium, Large and Large plus. All these plans have various monthly costs such as Medium cost £149, Large cost £299 and Large plus cost £519. All these costs are higher and cannot be afforded by local shop therefore I am making this inventory software to help them not to pay any fees.

2.2 Back-end Software

It involves the choice of correct back-end software which enables me to store the data in a database using PHP and MySQL. This data will be retrieved and show the information right back into the user browser. I will be comparing two different software such as MongoDB and MySQL.

2.2.1 MongoDB

It is an open-source document-based database management program. It was used by many organisations such as Twitter, Sony, Zen desk, etc. It stores the data in JSON- like documents that can have varied structure. It does not use the usual rows and column format where we see in the relational database in MySQL. It does not require to declare the structure of documents to the system.

2.2.2 MySQL

It is an open-source relational database management system(RDMS). It stores data in the form of rows of a table and uses the structured query language to access the data. It required me to define the tables and columns before I store any data and every row in the table needs to have the same columns. I was more familiar with MySQL language as I have used this language before in other projects therefore I will be using MySQL databases for my Inventory check system.

2.3 Libraries

I have evaluated two particular libraries before developing any software. jQuery developed by the jQuery Team and bootstrap 4 developed by Bootstrap Core Team for helping with front-end web development.

2.3.1 jQuery

jQuery is a JavaScript library that allows users to add functionality to the website. It works in most of the web browsers that are available in the market. This will help to create my UI such as animating, event handling, etc. I have used this library to make a confirmation dialog box for the contact form. The following code is an example of a jquery library that I have used in the contact form:

```
/* Function that will be executed when the page is ready */
$(document).ready(function() {
    /* Event that will get triggered when the form is submitted */
    $(document).on('submit', '#form', function() {
        /* Assigns to the email variable the value of #email */
        var email = document.getElementById("email").value;
        var subject = document.getElementById("subject").value;
        var message = document.getElementById("message").value;

        var format = "*Email:* " + email + "\n*Subject:* " + subject + "\n*Message:* " + message;

        /* HTTP GET request towards Telegram */
        /* Source: URL(https://api.jquery.com/jquery.get/)*/
        $.ajax({
            url: "https://api.telegram.org/bot1128209307:AAENqh0uZejLqb1-N0dOdq3xSF4LrGQZto/sendMessage?chat_id=326097&method=sendMessage",
            type: 'GET',
            success: function (resp) {
                /* Show an alert window if the request is successful */
                alert("Thank you for your message!");
                location.reload();
            },
            error: function(e) {
                /* Show an alert window if the request is unsuccessful + the error code */
                alert('Error: ' + e);
            }
        });
    });

    /* Prevents the page from reloading when submitting the form */
    return false;
});
```

Figure 2.1: Contact form functionality using jQuery API

2.3.2 Bootstrap 4

It is an open-source front-end development framework. It has many different interface components such as navigation, forms, model, Buttons, and more. It is one of the latest version with stable UI, I have used this framework to build a flexible, mobile-friendly, and responsive website. This framework does not have any issues to work in multiple browsers. It has saved me a lot of time and effort as it has predefined classes, tables, and other models. By saving time, I can give more time to other complicated work in other development work. I have learned this framework while I was working in college. The following code is one of the examples where I have used bootstrap:

```
<!--Navbar code start here-->
<nav class="navbar navbar-expand-lg navbar navbar-dark bg-dark">
  

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown"
    aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavDropdown">
    <ul class="navbar-nav">
      <!-- home page link -->
      <li class="nav-item active">
        <a class="nav-link" href="index.php">Home <span class="sr-only">(current)</span></a>
      </li>
      <!-- product page link -->
      <li class="nav-item">
        <a class="nav-link" href="product.php">Product</a>
      </li>
      <!-- analytic page link -->
      <li class="nav-item">
        <a class="nav-link" href="analytic.php">Analytic</a>
      </li>
      <!-- contact us page link -->
      <li class="nav-item">
        <a class="nav-link" href="contact.php">Contact us</a>
      </li>
    </ul>
  </div>
</nav>
```

Figure 2.2: Navbar using bootstrap framework

2.4 Software Architecture

To aid with selecting the best code editor, two editors were analysed such as brackets and sublime text to discover the hidden features.

2.4.1 Brackets

It is a source code editor that helps the developer to write the code and created by Adobe Systems. It is the useful editor as it shows live preview your code where you have written in HTML, CSS, and JavaScript. The main features it has is extracted which allows user to extract the information from fonts, colors straight from PSDs as clean CSS. I have chosen not to use brackets to write my code as it does not a feature for command line where I can manage my git repository and also run MySQL databases.

2.4.2 Sublime Text

It is a source code editor that helps the developer to write the code in many different languages. It is one of my favorites text editor where I can use the command line to manage my git repository and also I can run my MySQL database directly from this editor. It is totally user friendly where you can modify multiple lines and look for specific code easily.

2.5 Questionnaire/Survey Result

Before developing an application, A survey was carried out to find out what they are looking for once the software has been developed. This survey was answered by my uncle and other local shopkeepers near my house. By making the questionnaire will help me to get a better understanding of the system they are using and how much do they pay. I have used the google form to create the questionnaire because, in the end, it will show me the result in the pie chart format. The full details of this survey can be found in the appendix A - Questionnaire/Survey Result.

Figure 2.3 clearly shows that 70% of the people already using some kind of system that helps them to manage their inventory. This shows that shopkeepers love to use a management system.

CHAPTER 2. BACKGROUND RESEARCH

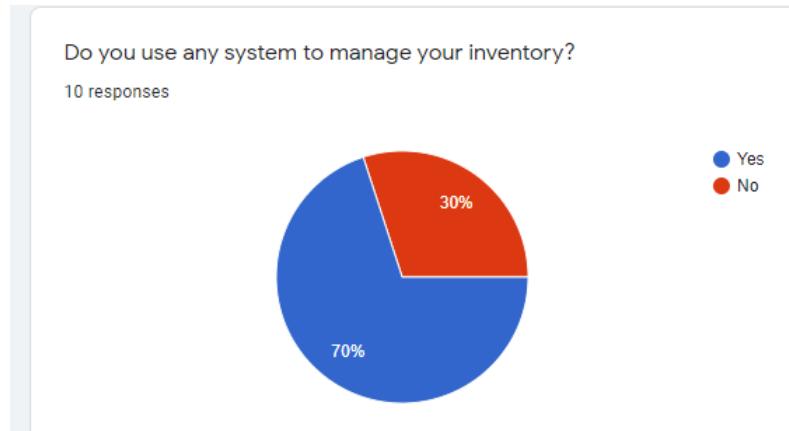


Figure 2.3: Question 1: Use any system

Figure 2.4 shows that 50% of the are using a spreadsheet to manage their inventory stocks and 37.5% of people do not use any system instead they use paper to manage. This shows that they need a stable system that can manage their stocks online.

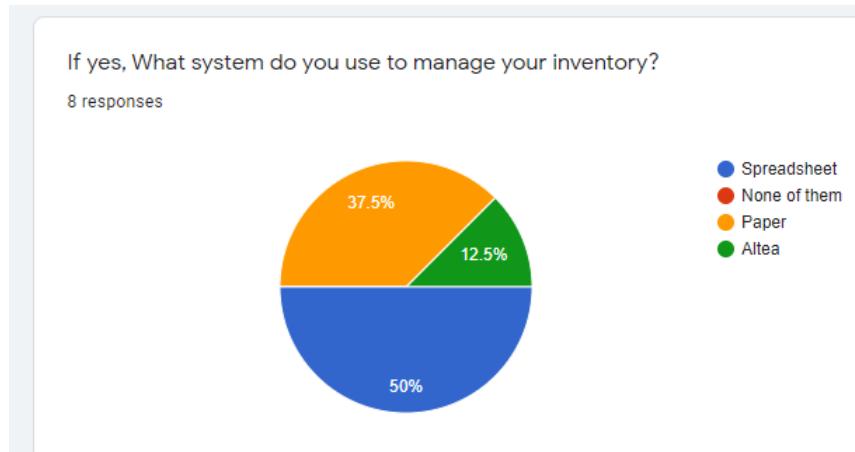


Figure 2.4: Question 2: choose of system

CHAPTER 2. BACKGROUND RESEARCH

Figure 2.5 shows that 70% of the people do not pay any fees because they used a spreadsheet to manage their inventory and it does not cost anything. Whereas 30% of the people pay between £51 - £100 Monthly and it was the cheapest cost that they found in the market. They were so happy when I told them that I am making a system to help them with no monthly cost.



Figure 2.5: Question 3: cost of current system

Figure 2.6 shows that 50% of the people use the spreadsheet daily and 50% of people use weekly. This shows that they use the system to amend the stocks one a week or daily. Therefore my system will save their time and get the report on daily bases.

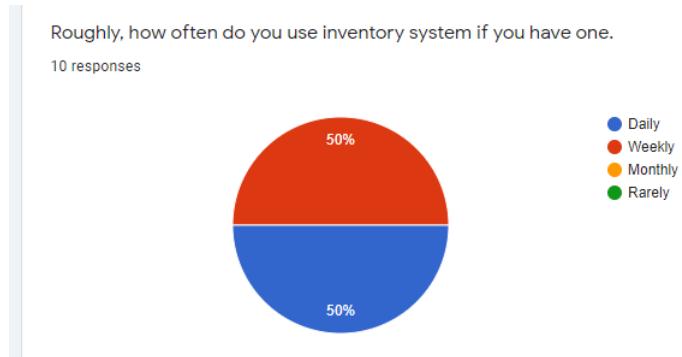


Figure 2.6: Question 4: Usage of the system

CHAPTER 2. BACKGROUND RESEARCH

Figure 2.7 was asked to understand if the user will use the system for free and also if they like to use the system from anywhere in the world. It shows that out of 10 people, 90% of them will use my system for free and 100% of the people will use it from outside their shop or from a different country.

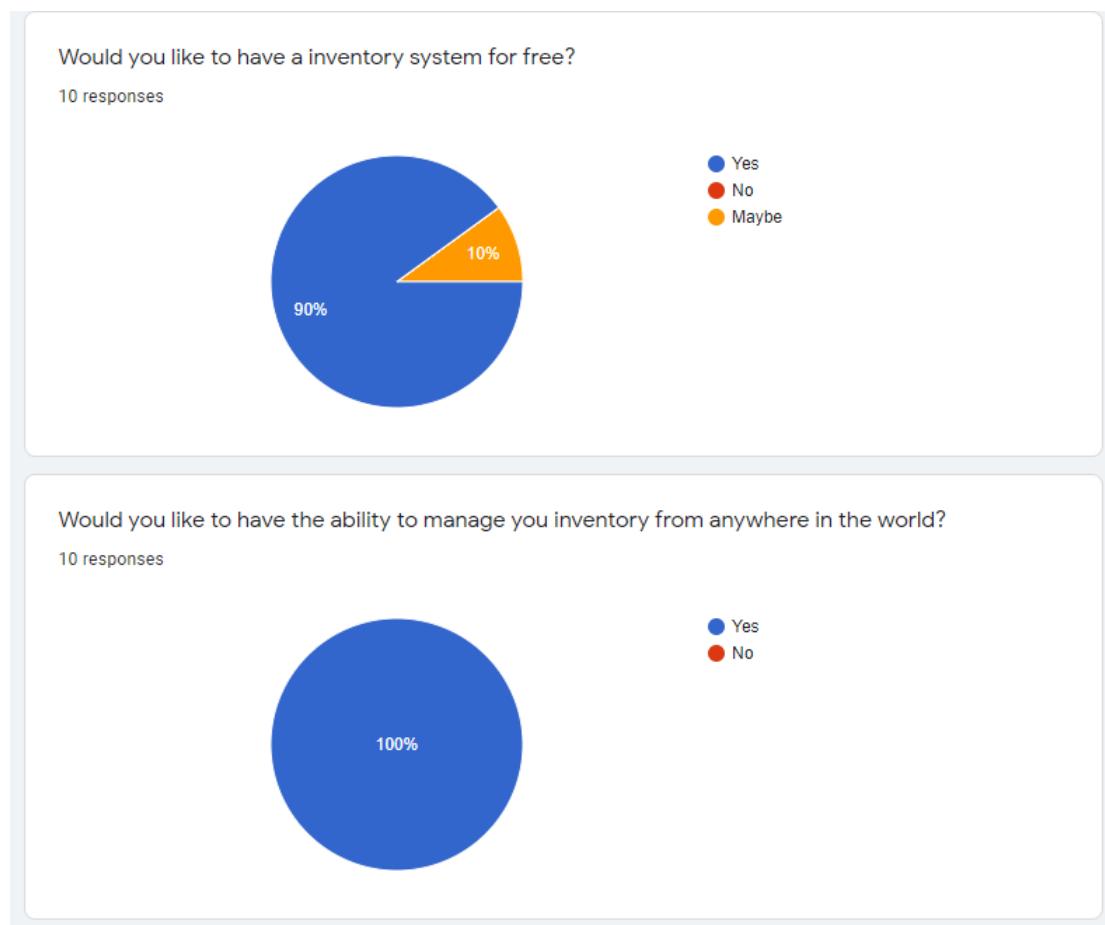


Figure 2.7: Question 5 & 6: Use the system for free and the ability to manage online

Figure 2.8 shows that out of 10 people, 90% of the people prefer to use my system to manage the inventory. This shows that I need to make a better and secure system. Therefore I need to give more time to my implementation.

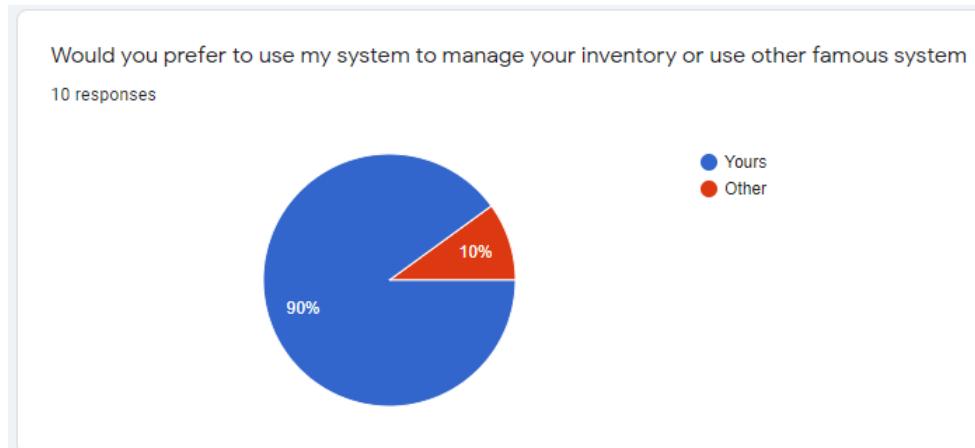


Figure 2.8: Question 7: Prefer to use my system or other

2.6 Project Aims

This project aims for three main components which will help to build the system. The first layer in programming called the presentation layer which is responsible for delivery and formatting to the system. The second layer will act as a communication between the interface and the server. The final layer is a database that will store and retrieve the data to the user. I will be writing about three various layers such as front-end, Server-side(PHP), and Database server.

2.6.1 Front-end

1. Easy to navigate throughout the website.
2. Right choice of color to make the website interactive to the user.
3. Able to show all the relevant data on the user screen.
4. Show the product information when the user uses search functionality.

5. Lock the important pages using the login and logout method.
6. Ensure users can edit or delete the product.

2.6.2 Server-side (PHP)

1. Get the data from POST and GET method.
2. Submit the MySQL query into the database when submitted by the user.
3. Handle multiple request from the user.

2.6.3 Database server

1. Store and retrieve the data when required.
2. Keep the table name up to date with the right key (Primary or Foreign Key).
3. Able to submit the data when the user adds the product in the database.

Chapter 3

System Design

This process is essential to be taken when building an application. This involves designing the front-end as well as a back-end system. This section will explain the design process for each part of the system such as system requirement, wireframe, use case diagram, and more.

3.1 System Requirement

This section will give a better understanding of what the system needs to handle to make a good application. This will be done by making a functional, non-functional, Database, and PHP requirement.

3.1.1 Functional Requirement

1. The website must able to run on every single browser that is currently in the market.
2. The website should be able to display all the products that are in the database.
3. It should able to show real data in the chart.
4. The website should able to search the data using the search functionality.
5. The website should able to add products and display in real-time.

6. The site should able to delete and edit data in real-time.
7. The site should able to contact the developer using the contact form.
8. Users should able to log in and create new users.
9. Enable easy navigation around the website.

3.1.2 Non-Functional Requirement

1. The system should respond quickly to user input such as users want to find a specific product using the search functionality.
2. The website should have a simple layout with good color.
3. During the research, I have found that the user does not like too much text on the homepage.
4. The system will require an internet connection, in order to load up the website and get data from the database.
5. Ensuring the current data in the database cannot be accessed by unauthorised users.

3.1.3 Database Requirement

1. Must able to store the data without redundancy.
2. Allow quick and easy recovery when needed.
3. The database system able to delete or update when it is performed by the user form.

3.1.4 PHP Requirement

1. The system should able to handle multiple requests from the users.
2. Must able to perform POST and GET methods.
3. Must perform the update and delete data from the database.
4. User's password must be encrypted.
5. The system must able to track down the unusual activity in the database.

3.2 Use Case Diagram

The use case diagram is very important because it analyse the life cycle of a user when they use the application. The use case diagram(Figure 3.1) deals with three different scenarios where users search product, edit product, and contact us form. First scenarios where a user searches the product, they need to enter the item number or item name where the server receives that input and find all the records which will be displayed backed to the user browser. Second scenarios, users want to edit the existing record where a user clicks on the edit button which will take them to edit form and update the record from the database. Final scenarios where a user wants to send a message using the form on the contact page. Once the form is submitted, the server process the message using the telegram API and displays it in the telegram app.

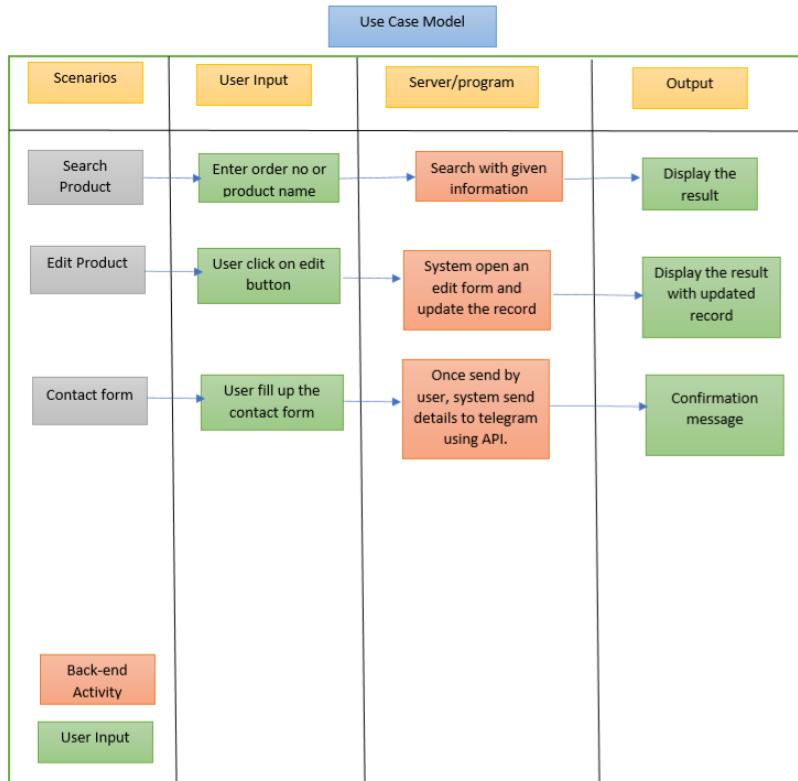


Figure 3.1: Use Case Diagram

3.3 Wireframe

This part plays a big role in the user interface and experience when the user visits my site. My aims are to make the website simple and easy to navigate without any problems because some of the shopkeepers may not able to use the system if the UI is complicated to use. Therefore I have asked my uncle what kind of interface he was looking for and he said: "simple and easy to manage everyday inventory". The first mission was to find what interface is currently looking in other inventory systems and make a difference to them. Furthermore, it will fail the design if the UI is complex.

I have created two different wireframes using an online tool(wireframe.cc). The first wireframe (Figure 3.2, 3.3, 3.4 3.5) was designed and taken backed to my uncle to gain feedback and allow me to improve them if there is something missing in the UI. The second wireframe was designed with the improvement that was suggested by my uncle.

3.3.1 Wireframe 1

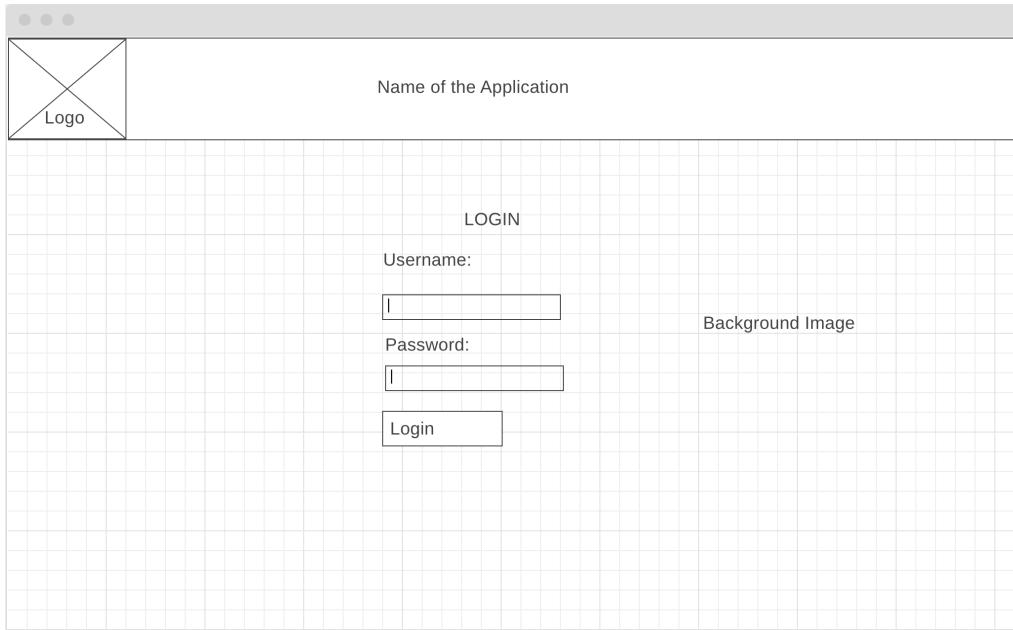


Figure 3.2: Wireframe 1 - Homepage

CHAPTER 3. SYSTEM DESIGN

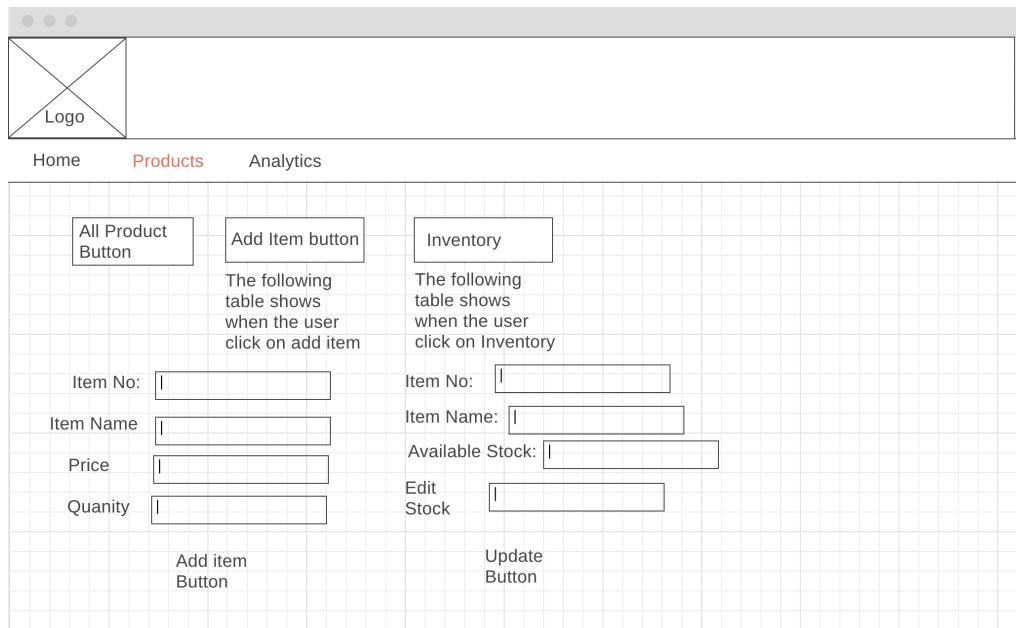


Figure 3.3: Wireframe 1 - Product

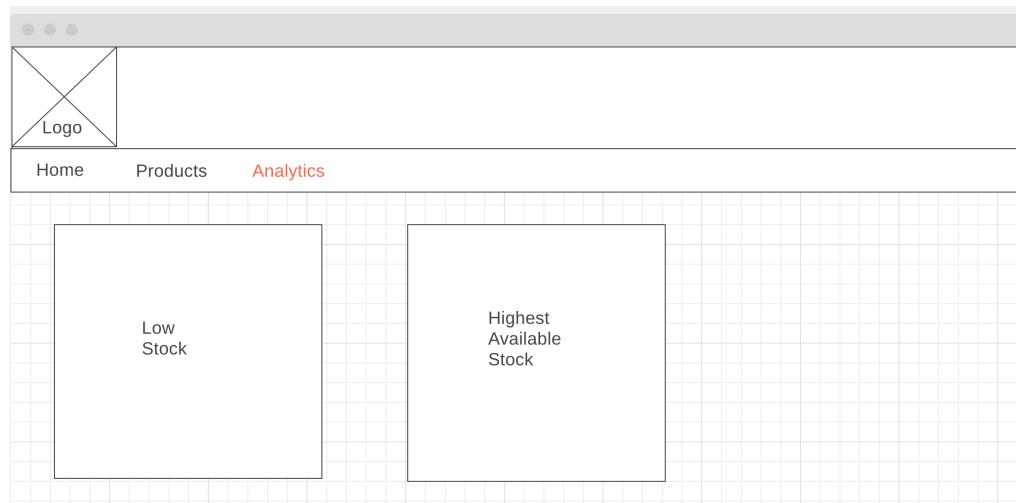


Figure 3.4: Wireframe 1 - Analytic

CHAPTER 3. SYSTEM DESIGN

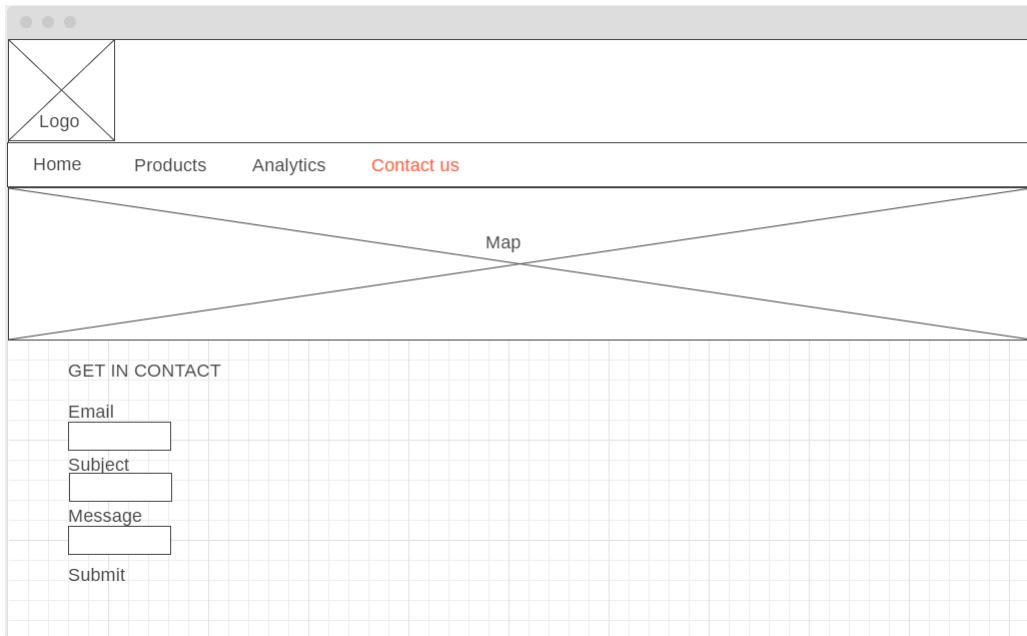


Figure 3.5: Wireframe 1 - Contact us

3.3.2 Wireframe 2

The wireframe 2 is the final UI which you will see on the actual website. To get the review of the design, shown this to my uncle and other shopkeeper and they were impressed with the design of the application.

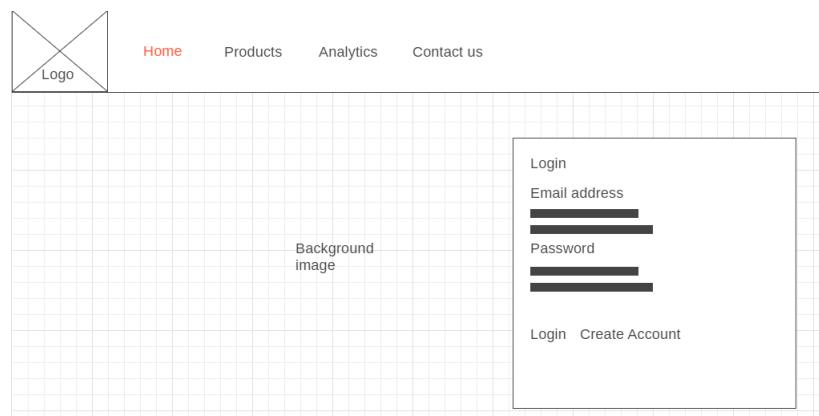


Figure 3.6: Wireframe 2 - Homepage

CHAPTER 3. SYSTEM DESIGN

Figure 3.7, below page open up and asked user to click on view all product or add product button.

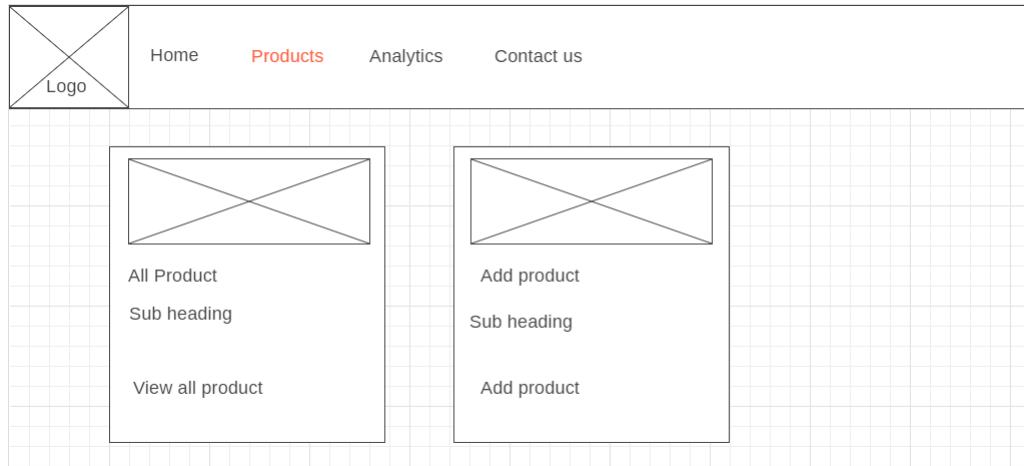


Figure 3.7: Wireframe 2 - Product

Figure 3.8, below page open up when the user click on view all product from the product page.

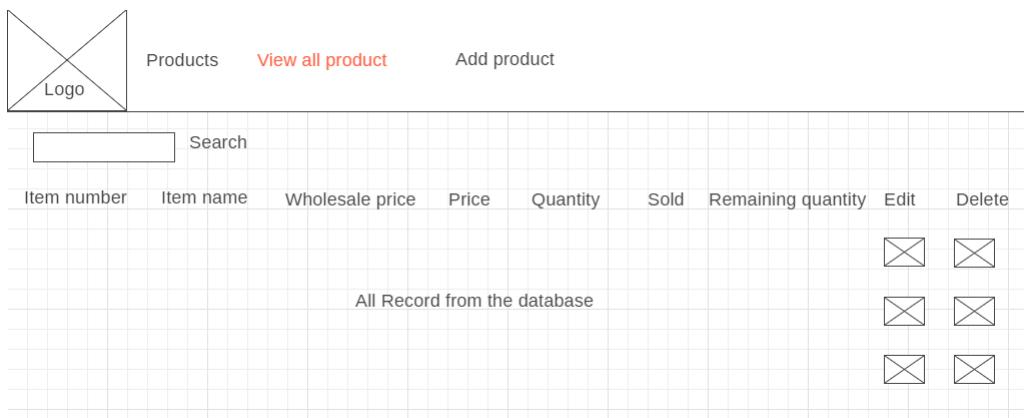
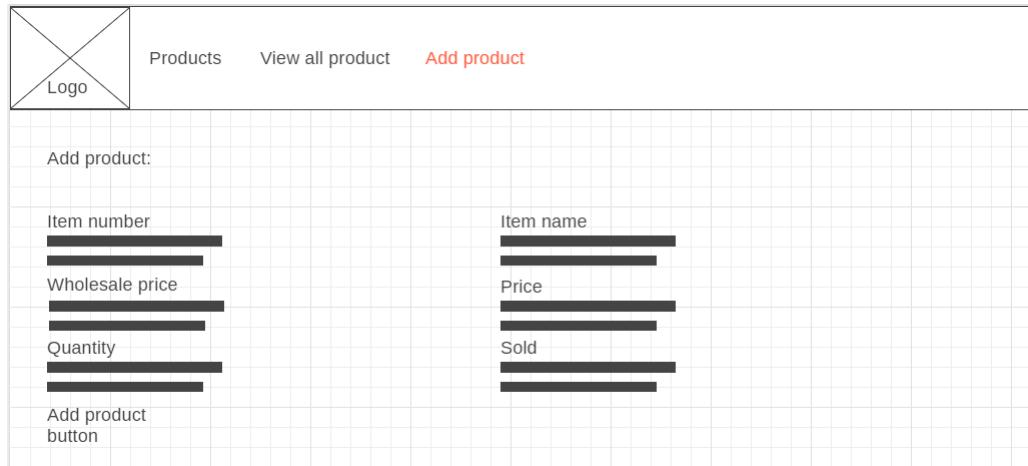


Figure 3.8: Wireframe 2 - View all product

CHAPTER 3. SYSTEM DESIGN

Figure 3.9, below page open up when the user click on add product from the product page.



A wireframe for an 'Add product' page. At the top left is a logo consisting of a square with a diagonal cross. To its right are three buttons: 'Products', 'View all product', and 'Add product' (which is highlighted in red). Below this is a section titled 'Add product:' containing two columns of input fields. The left column includes 'Item number', 'Wholesale price', 'Quantity', and an 'Add product button'. The right column includes 'Item name', 'Price', and 'Sold'.

Figure 3.9: Wireframe 2 - Add product

Figure 3.10, below page open up when the user click on edit button from view all product page.



A wireframe for an 'Edit product' page. At the top left is a title 'Edit product'. Below it are two columns of input fields. The left column includes 'Item number', 'Wholesale price', 'Quantity', and an 'Update product button'. The right column includes 'Item name', 'Price', and 'Sold'.

Figure 3.10: Wireframe 2 - Edit product

CHAPTER 3. SYSTEM DESIGN

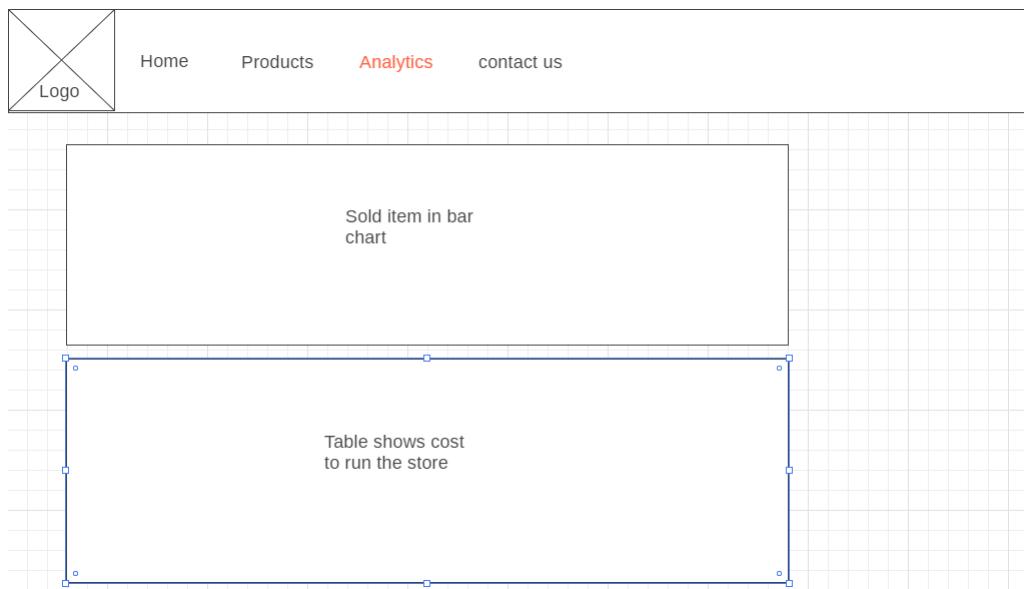


Figure 3.11: Wireframe 2 - Analytic

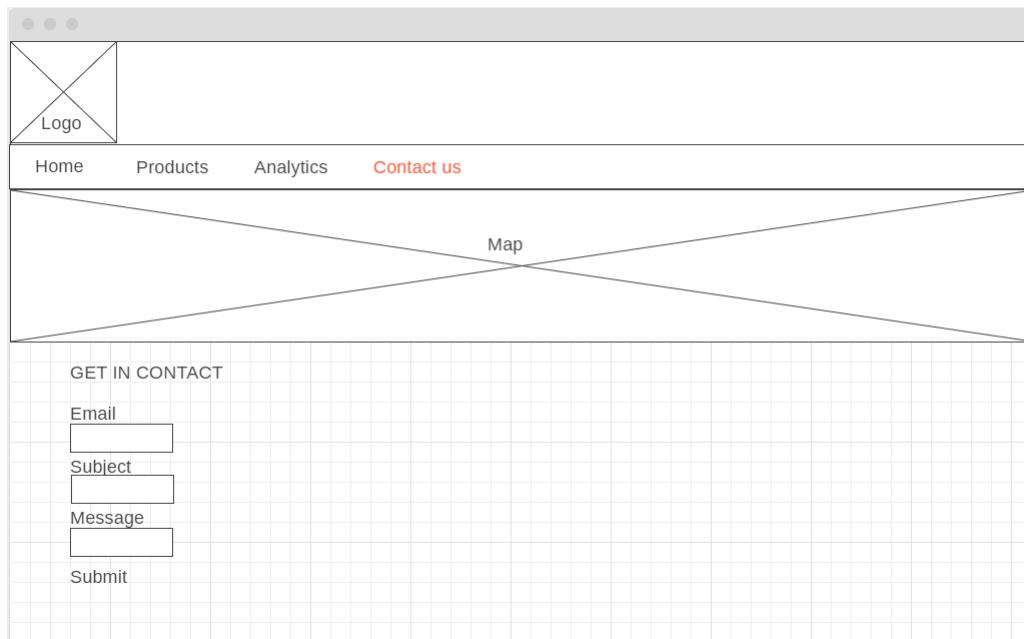


Figure 3.12: Wireframe 2 - Contact us

3.3.3 Final outcome

The following images are the final UI that was implemented after finalising the wireframe 2. The UI was made with the right choice of color, Scale and compatible with mobile browser.

Figure 3.13 is the main page of the website where the user can log in to the account and also create the account if they have not made one. This feature is currently not available at the moment but will my implemented in the feature.

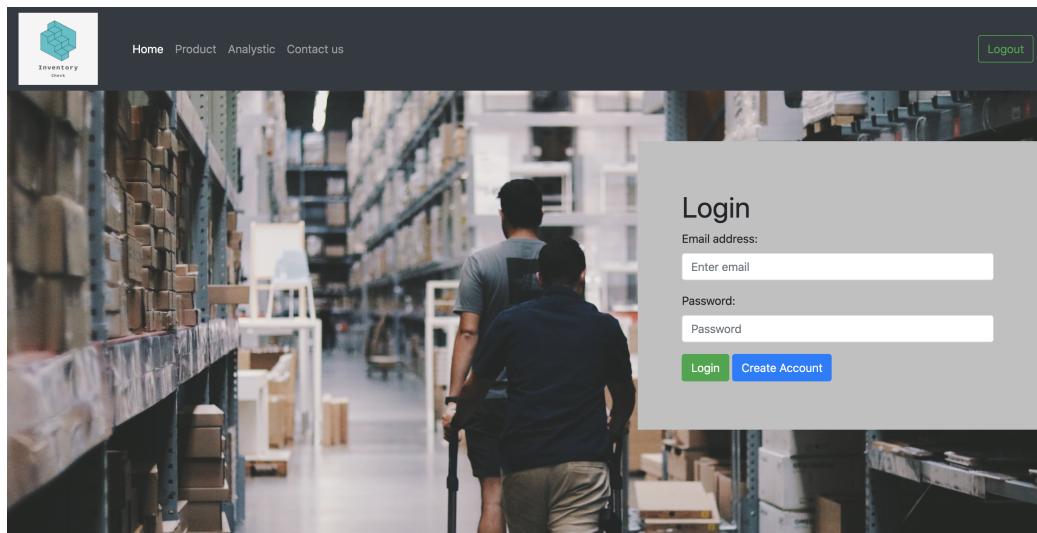


Figure 3.13: Final UI - Homepage

CHAPTER 3. SYSTEM DESIGN

Figure 3.14 is the product where user can select view all product or add product model which will take them to there destination.

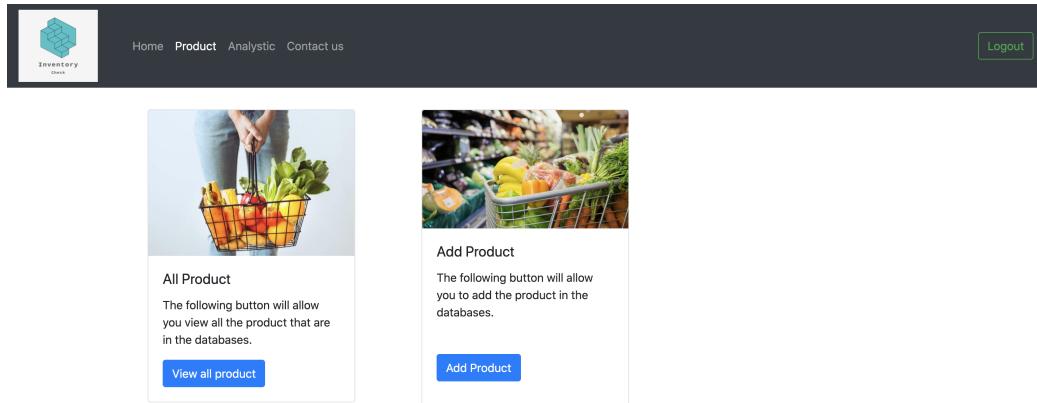


Figure 3.14: Final UI - Product

Figure 3.15 is the view all product where user can view all the product that are stored in the database, edit the item, delete the item and search for specific product using item name or item number.

The screenshot shows a web application interface for viewing all products. At the top, there is a dark header bar with a logo on the left, followed by navigation links: Product, View all Product, and Add product. Below the header, there is a search bar with a 'Search' button next to it. The main content area displays a table of products. The table has columns for Item Number, Item Name, Wholesale Price, Price, Quantity, Sold, Remaining Quantity, Edit, and Delete. Each row contains a set of data for a specific product, with 'Edit' and 'Delete' buttons in the last two columns. The data in the table is as follows:

Figure 3.15: Final UI - View all product

CHAPTER 3. SYSTEM DESIGN

Figure 3.16 is the UI of search result when user search the product from view all product page.

New search						
Item Number	Item Name	Wholesale Price	Price	Quantity	Sold	Remaining Quantity
364716	Peanut butter	£0.80	£1	15	6	9
464737	Pastas	£0.60	£1.20	20	4	16
373919	Banana	£0.20 each	£0.30 each	39	39	0
347391	Cauliflower	£0.60	£0.80	7	2	5
182625	Potato	£1 kg	£1.20 kg	5	2	3
784263	Carrots	£0.60	£0.80	10	9	1
372919	Aubergine	£0.50	£0.70	10	3	7
284426	Spinach 200g	£1	£1.55	7	4	3
127819	Tomato	£0.15	£0.20	50	15	35
243618	Oranges	£0.15 each	£0.20 each	50	10	40

Figure 3.16: Final UI - Search Result

Figure 3.17 is the edit form where user can update the product that are stored in the database.

Edit Product:

Item Number	784263	Item Name	Carrots
Wholesale Price	0.60	Price	0.80
Quantity	10	Sold	9

Figure 3.17: Final UI - Edit Product

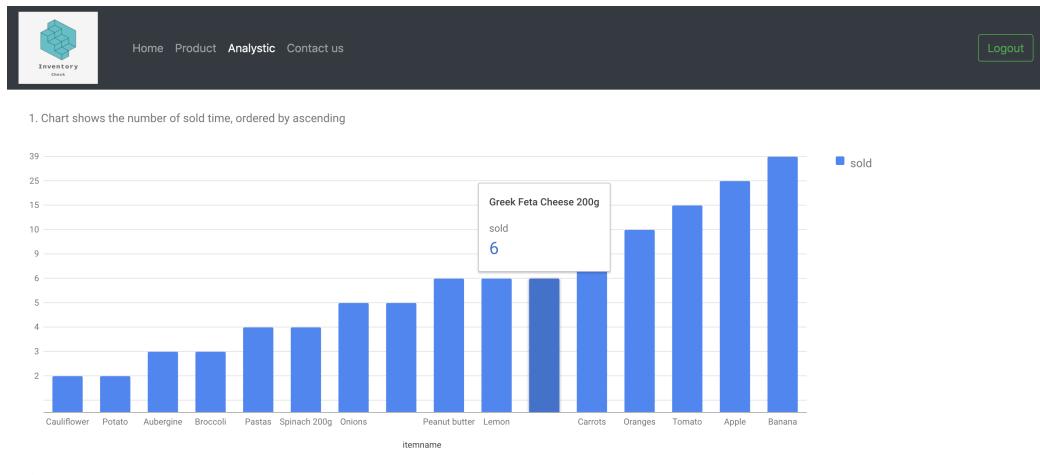
CHAPTER 3. SYSTEM DESIGN

Figure 3.18 where user can add product in the database and view the product in the product page once it has been successful stored in the database.

Add Product:

Item Number	Item Name
<input type="text"/>	<input type="text"/>
Wholesale Price	Price
<input type="text"/>	<input type="text"/>
Quantity	Sold
<input type="text"/>	<input type="text"/>

Figure 3.18: Final UI - Add Product



Name	Result
1 Total cost from wholesale price	110

Figure 3.19: Final UI - Analystic

CHAPTER 3. SYSTEM DESIGN

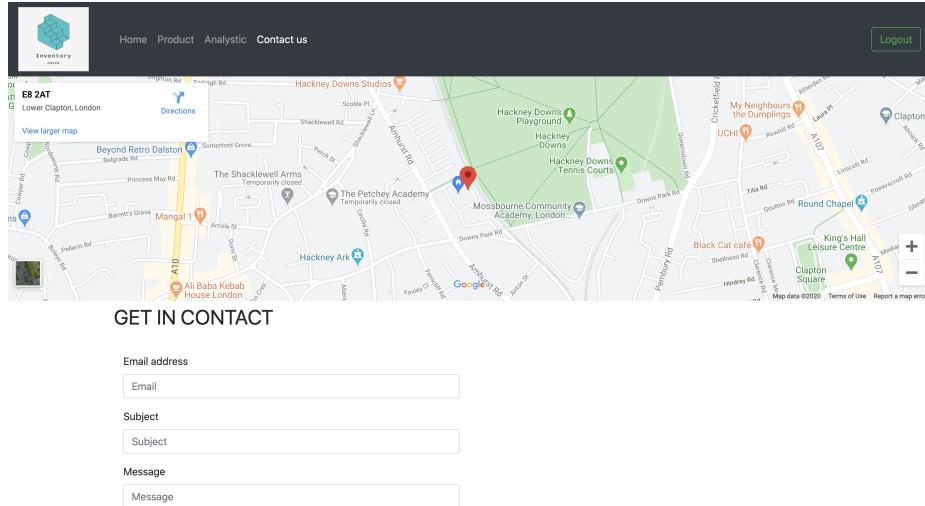


Figure 3.20: Final UI - Contact us

3.4 Side Map

It is a list of pages that are on the website. By making this diagram, it shows the developer where/how all the pages are linked together across the site.

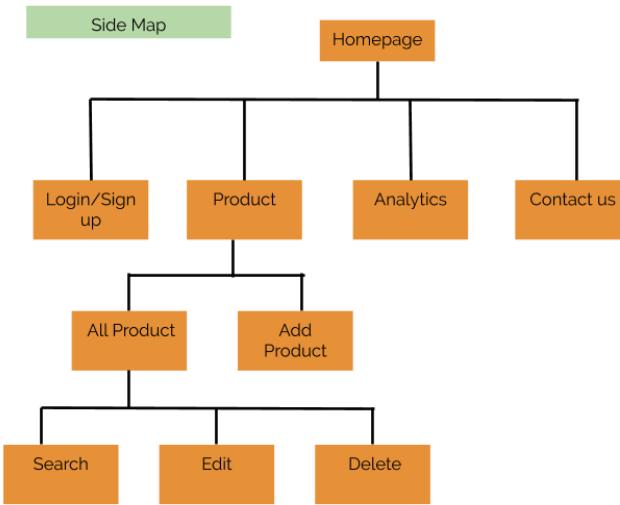


Figure 3.21: Side Map

Chapter 4

Implementation

This chapter will explain how the system was implemented to make a correct web-based application. To make the application, I have used six various technologies such as HTML, CSS, JavaScript/jQuery, PHP, MySQL, and API to receive a message through the contact form. All the program was written in Sublime text editor.

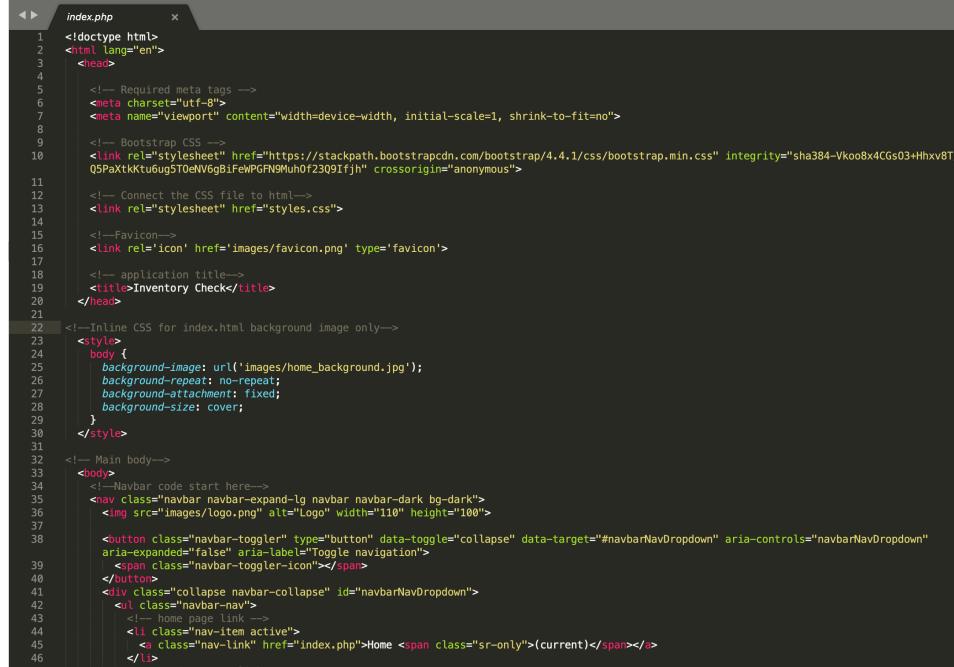
4.1 Front-End Development

Front-end development is where developers convert the data to a graphical interface via HTML, CSS, and JavaScript so the user can view and interact with the site directly. I have made the front-end interesting where the user can read the information easily.

4.1.1 HTML

In order to make the front-end quicker, I have used a bootstrap 4 framework which will help me to design my site faster and easier. I have used this framework before in other so I know what tool and function to use to make my system interactive. Figure 4.1 shows the HTML code used on the site and has been taken from the index.php page. You can also see the bootstrap framework CDN in the head tag which imports the library in my system.

CHAPTER 4. IMPLEMENTATION



The screenshot shows a code editor window with the file 'index.php' open. The code is a combination of HTML and CSS. It includes meta tags for character encoding and viewport, a link to a Bootstrap CSS file, a favicon, and a title 'Inventory Check'. The body section contains inline CSS for a background image and a navbar section. The navbar includes a logo image, a toggle button, and a dropdown menu with a single item pointing to 'index.php'.

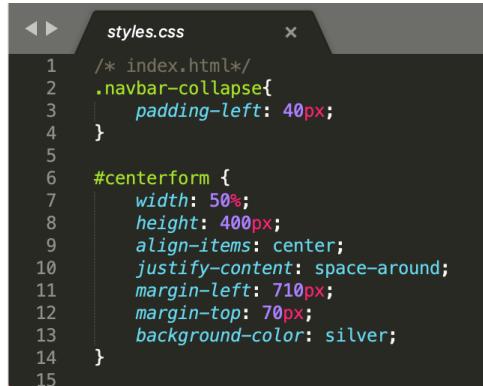
```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" integrity="sha384-QPaxXkttuug5T0eWVg81fWmFGN9Mu0f2309Ifjh" crossorigin="anonymous">
  <!-- Connect the CSS file to html-->
  <link rel="stylesheet" href="styles.css">
  <!--Favicon-->
  <link rel="icon" href='images/favicon.png' type='favicon'>
  <!-- application title-->
  <title>Inventory Check</title>
</head>
<body>
  <!-- Main body-->
  <!--Navbar code start here-->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <!-- home page Link -->
        <li class="nav-item active">
          <a class="nav-link" href="index.php">Home <span class="sr-only">(current)</span></a>
        </li>
      </ul>
    </div>
  </nav>
  <style>
    body {
      background-image: url('images/home_background.jpg');
      background-repeat: no-repeat;
      background-attachment: fixed;
      background-size: cover;
    }
  </style>
</body>

```

Figure 4.1: Example of HTML code

4.1.2 CSS

It is a style sheet language that is used for how documents are presented to users with different styles, layout, etc. Figure 4.2 shows the CSS code used in the index.php page to make the login form center-right.



The screenshot shows a code editor window with the file 'styles.css' open. The code defines two CSS classes: '.navbar-collapse' with a padding-left of 40px, and '#centerform' which has a width of 50%, height of 400px, align-items set to center, justify-content set to space-around, margin-left of 710px, and a margin-top of 70px. The background-color is set to silver.

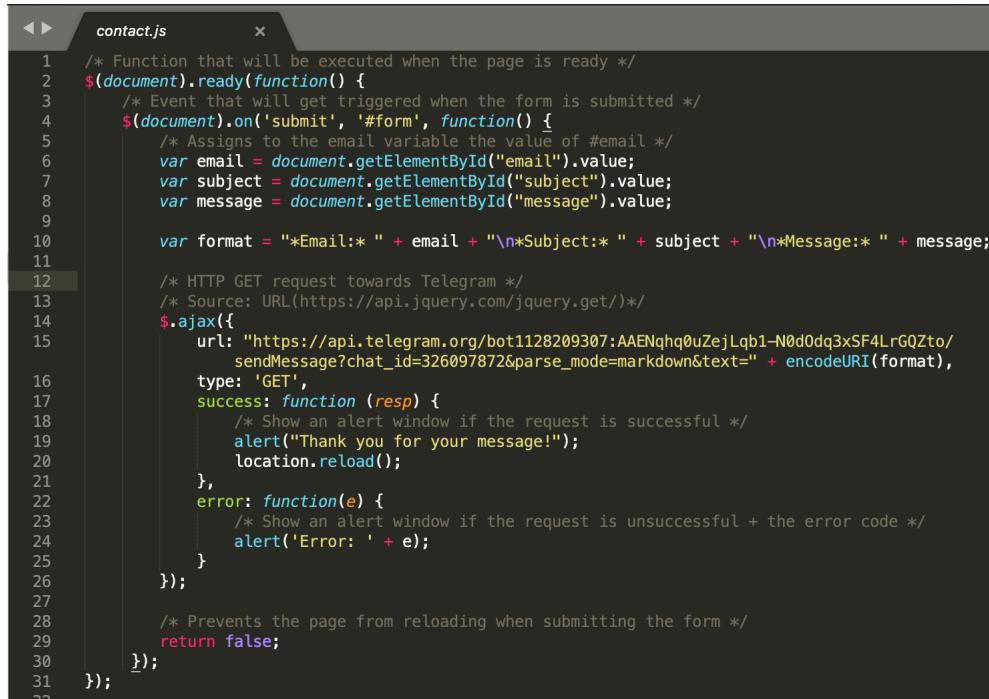
```
/* index.html*/
.navbar-collapse{
  padding-left: 40px;
}

#centerform {
  width: 50%;
  height: 400px;
  align-items: center;
  justify-content: space-around;
  margin-left: 710px;
  margin-top: 70px;
  background-color: silver;
}
```

Figure 4.2: Example of CSS code

4.1.3 JavaScript/jQuery

It is a scripting language that enables the user to implement complex features on the web pages such as interactive maps, 2D or 3D graphics, alert dialog, etc. I have used this language to make the interactive map and executing the contact form. Figure 4.3 shows the JavaScript code which will execute once the user clicks on the submit button on the contact form.



```
contact.js
```

```
1  /* Function that will be executed when the page is ready */
2  $(document).ready(function() {
3      /* Event that will get triggered when the form is submitted */
4      $(document).on('submit', '#form', function() {
5          /* Assigns to the email variable the value of #email */
6          var email = document.getElementById("email").value;
7          var subject = document.getElementById("subject").value;
8          var message = document.getElementById("message").value;
9
10         var format = "*Email:* " + email + "\n*Subject:* " + subject + "\n*Message:* " + message;
11
12         /* HTTP GET request towards Telegram */
13         /* Source: URL(https://api.jquery.com/jquery.get/)*/
14         $.ajax({
15             url: "https://api.telegram.org/bot1128209307:AAEnqhq0uZejLqb1-N0d0dq3xSF4LrGQZto/
16                 sendMessage?chat_id=326097872&parse_mode=markdown&text=" + encodeURI(format),
17             type: 'GET',
18             success: function (resp) {
19                 /* Show an alert window if the request is successful */
20                 alert("Thank you for your message!");
21                 location.reload();
22             },
23             error: function(e) {
24                 /* Show an alert window if the request is unsuccessful + the error code */
25                 alert('Error: ' + e);
26             }
27         });
28
29         /* Prevents the page from reloading when submitting the form */
30         return false;
31     });
32 }
```

Figure 4.3: Example of JavaScript code

4.2 PHP

PHP is largely used by the back-end server and executes an important feature throughout the system. I have used PHP to view all products, edit products, add products, search products, delete the product, and analyse the product which will be shown in chart form using google chart library. It is a direct communication gateway between the server-side and client-side databases.

4.2.1 Database Connection

In order for my database to interact with the system, I need to establish a connection between the database and PHP script. Figure 4.4 is the code that connect my PHP script to XAMPP server.

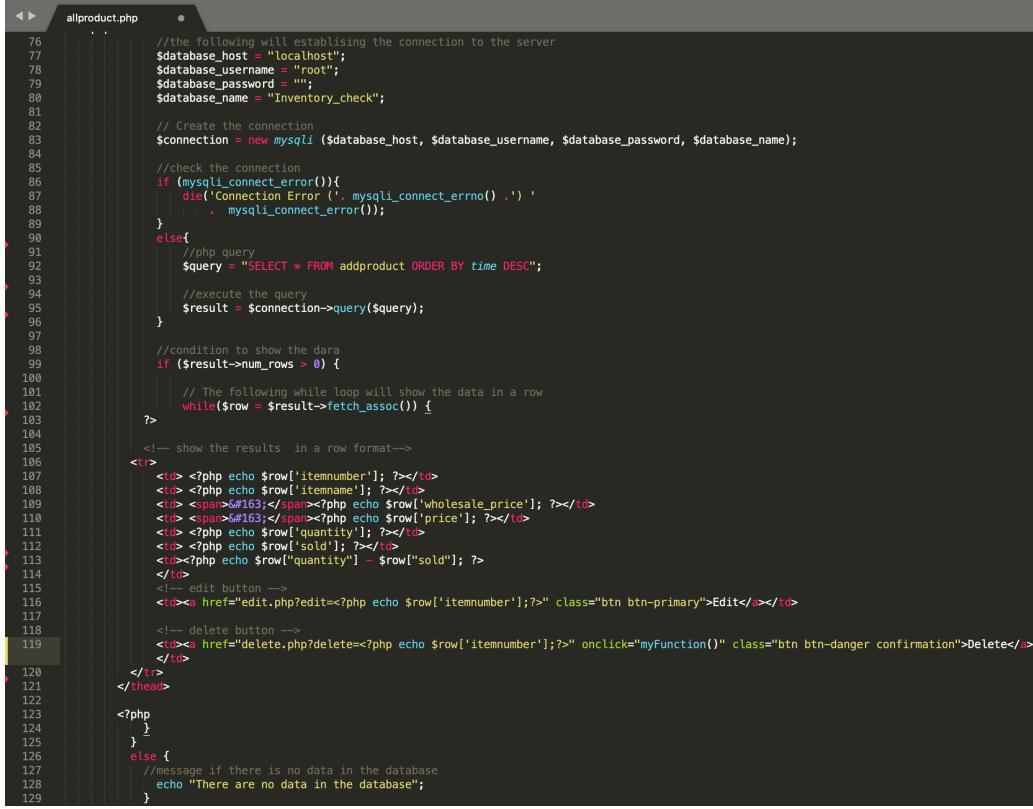
```
1 <?php
2
3     //the following will establishing the connection to the server
4     $database_host = "localhost";
5     $database_username = "root";
6     $database_password = "";
7     $database_name = "Inventory_check";
8
9     // Create the connection
10    $connection = new mysqli ($database_host, $database_username, $database_password, $database_name);
11
12    //check the connection
13    if (mysqli_connect_error()){
14        die('Connection Error ('. mysqli_connect_errno(). ') ' .
15            . mysqli_connect_error());
16    }
```

Figure 4.4: Connection to the server

4.2.2 View all product

It is essential that the user can view all the products that are in the database as a list format. Firstly, it will establish the connection between the database and the PHP script. After that, it will execute the query ("SELECT * FROM add product ORDER BY time DESC";) which will gather all the suitable data from add product table. Once the data is gathered, the PHP echo will display all the data in descending order by time to the user screen(Figure 4.5 - line 107 to 117). This will show the most recent product add in the database first. If the database has no record to display then it will show the following message "There are no data in the database" in line 134 - Figure 4.5.

CHAPTER 4. IMPLEMENTATION



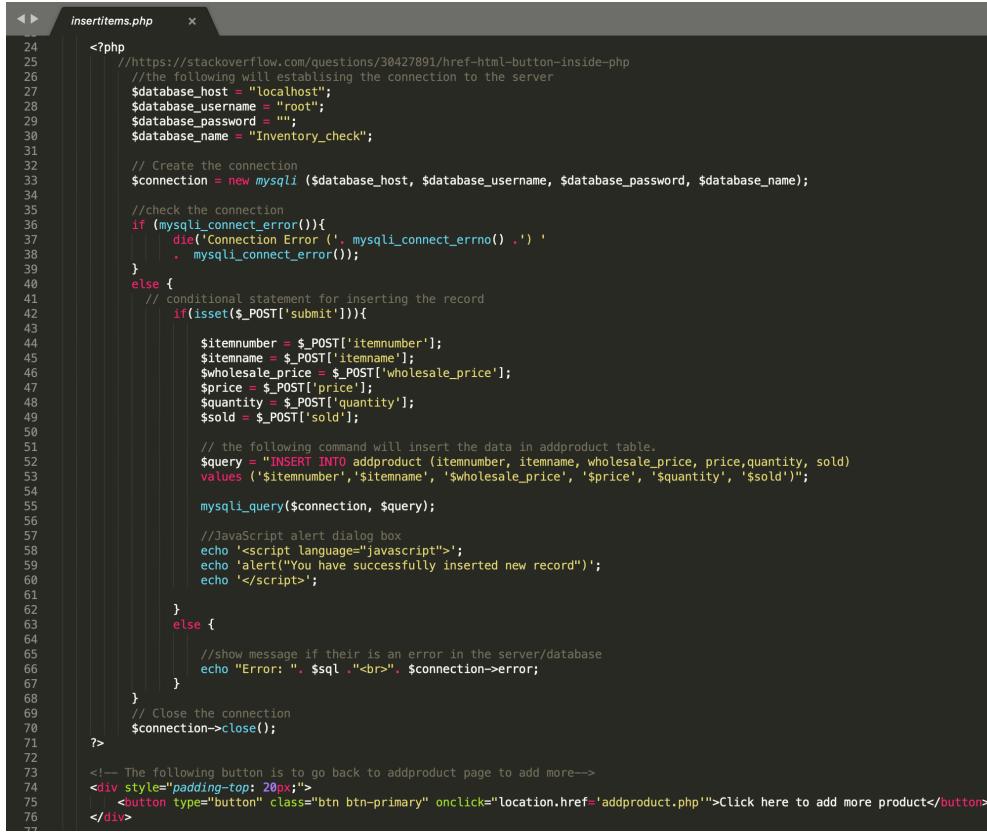
```
76 //the following will establishing the connection to the server
77 $database_host = "localhost";
78 $database_username = "root";
79 $database_password = "";
80 $database_name = "Inventory_check";
81
82 // Create the connection
83 $connection = new mysqli ($database_host, $database_username, $database_password, $database_name);
84
85 //check the connection
86 if(mysqli_connect_error()){
87     die('Connection Error (' . mysqli_connect_errno() . ') ' .
88          . mysqli_connect_error());
89 }
90 else{
91     //php query
92     $query = "SELECT * FROM addproduct ORDER BY time DESC";
93
94     //execute the query
95     $result = $connection->query($query);
96 }
97
98 //condition to show the data
99 if ($result->num_rows > 0) {
100
101     // The following while loop will show the data in a row
102     while($row = $result->fetch_assoc()) {
103
104         <!-- show the results in a row format-->
105         <tr>
106             <td><?php echo $row['itemnumber']; ?></td>
107             <td><?php echo $row['itemname']; ?></td>
108             <td><span>&#163;</span><?php echo $row['wholesale_price']; ?></td>
109             <td><span>&#163;</span><?php echo $row['price']; ?></td>
110             <td><?php echo $row['quantity']; ?></td>
111             <td><?php echo $row['sold']; ?></td>
112             <td><?php echo $row['quantity'] - $row['sold']; ?>
113             </td>
114
115         <!-- edit button -->
116         <a href="edit.php?edit=<?php echo $row['itemnumber']; ?>" class="btn btn-primary">Edit</a></td>
117
118         <!-- delete button -->
119         <a href="delete.php?delete=<?php echo $row['itemnumber']; ?>" onclick="myFunction()" class="btn btn-danger confirmation">Delete</a>
120     </tr>
121     <thead>
122
123     <?php
124         }
125     }
126     else {
127         //message if there is no data in the database
128         echo "There are no data in the database";
129     }
130 }
```

Figure 4.5: allproduct.php

4.2.3 Add product

The first most important thing for this system to have is to add the product in the database. This was created by using the POST method to transfer information through HTTP header and store data in the database. Figure 4.6, Line 42(if(isset(\$_POST['submit'])))) is a if statement which will execute when the user click on submit button from the add product HTML form. It will then run the query(line 52 to 53) that will insert the data in add product table in the database and display the message if the record has been successfully inserted or not. Once the record has been successful insert, the user can add more records using the same HTML form.

CHAPTER 4. IMPLEMENTATION



The screenshot shows a code editor window with the file name 'insertitem.php' at the top. The code is a PHP script for inserting items into a database. It includes comments explaining the connection setup and the SQL query. A JavaScript alert box is used to confirm successful insertion. Error messages are displayed if there are issues with the connection or query. The code ends with a comment indicating it's a back button to the addproduct page.

```
<?php
//https://stackoverflow.com/questions/30427891/href-html-button-inside-php
//the following will establishing the connection to the server
$database_host = "localhost";
$database_username = "root";
$database_password = "";
$database_name = "Inventory_check";

// Create the connection
$connection = new mysqli ($database_host, $database_username, $database_password, $database_name);

//check the connection
if (mysqli_connect_error()){
    die('Connection Error ('. mysqli_connect_errno() .') '
        . mysqli_connect_error());
}
else {
    // conditional statement for inserting the record
    if(isset($_POST['submit'])){

        $itemnumber = $_POST['itemnumber'];
        $itemname = $_POST['itemname'];
        $wholesale_price = $_POST['wholesale_price'];
        $price = $_POST['price'];
        $quantity = $_POST['quantity'];
        $sold = $_POST['sold'];

        // the following command will insert the data in addproduct table
        $query = "INSERT INTO addproduct (itemnumber, itemname, wholesale_price, price, quantity, sold)
values ('$itemnumber','$itemname', '$wholesale_price', '$price', '$quantity', '$sold')";

        mysqli_query($connection, $query);

        //JavaScript alert dialog box
        echo '<script language="javascript">';
        echo 'alert("You have successfully inserted new record")';
        echo '</script>';
    }
    else {
        //show message if their is an error in the server/database
        echo "Error: ". $sql ."  
". $connection->error;
    }
}
// Close the connection
$connection->close();
?>

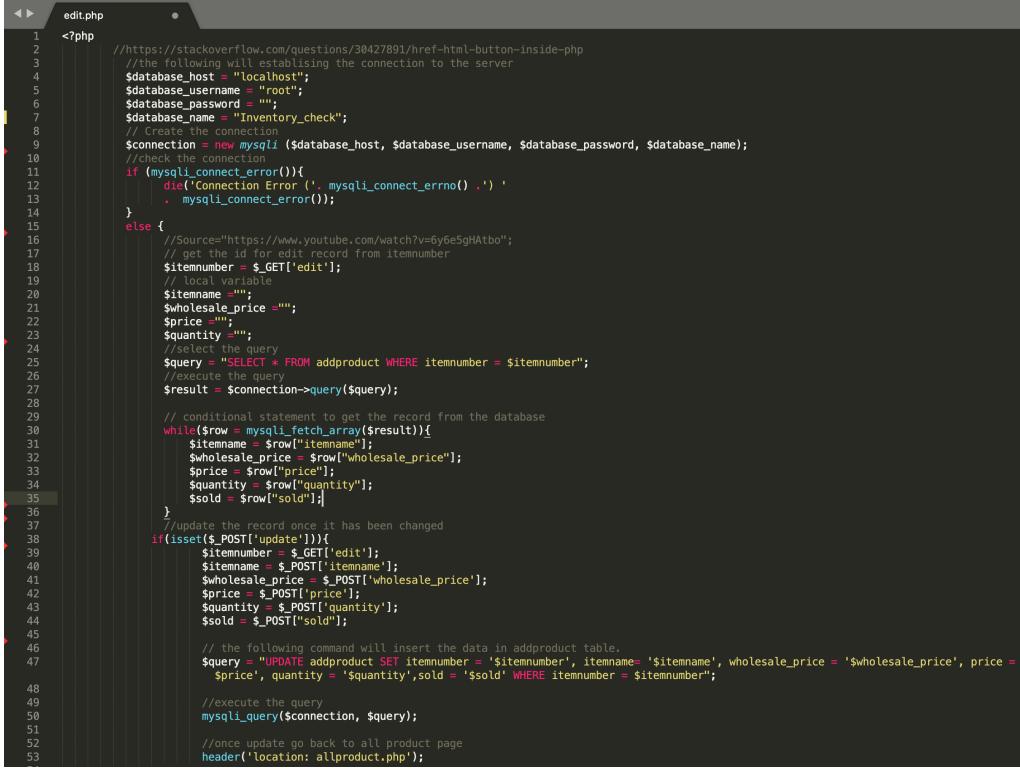
<!-- The following button is to go back to addproduct page to add more-->
<div style="padding-top: 20px;">
    <button type="button" class="btn btn-primary" onclick="location.href='addproduct.php'">Click here to add more product</button>
</div>
```

Figure 4.6: insertitem.php

4.2.4 Edit product

The edit.php will get the existing data to show in the edit form. This was done by using the GET method that uses item number as id. The while loop in figure 4.7 will fetch all the data in the form and allow the user to edit the record. After changing the record user click on update button which will than use POST method (if(isset(\$_POST['update']))) to update the data into the database. Once the record is updated the system will take the user to allproduct.php page using the header method(line 53).

CHAPTER 4. IMPLEMENTATION



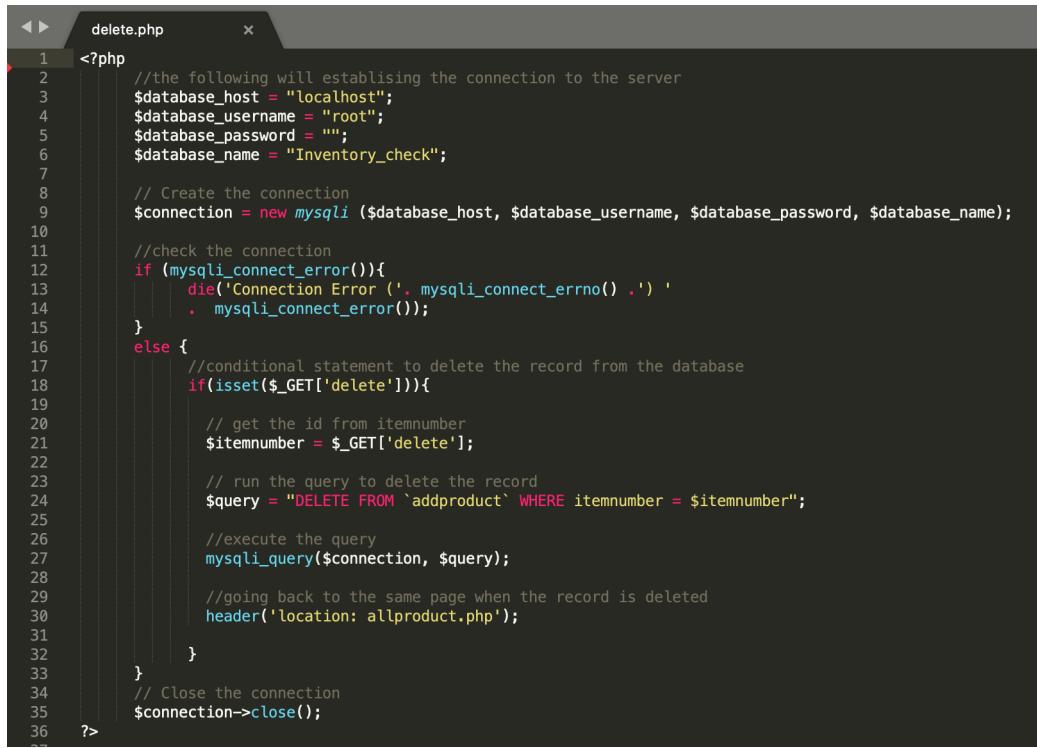
```
<?php
//https://stackoverflow.com/questions/30427891/href-html-button-inside-php
1 //the following will establishing the connection to the server
2 $database_host = "localhost";
3 $database_username = "root";
4 $database_password = "";
5 $database_name = "Inventory_check";
6 // Create the connection
7 $connection = new mysqli ($database_host, $database_username, $database_password, $database_name);
8 //check the connection
9 if (mysqli_connect_error()){
10     die('Connection Error (' . mysqli_connect_errno() . ') ' .
11         . mysqli_connect_error());
12 }
13 else {
14     //Source="https://www.youtube.com/watch?v=6y6e5gHAtbo";
15     // get the id for edit record from itemnumber
16     $itemnumber = $_GET['edit'];
17     // local variable
18     $itemname = "";
19     $wholesale_price = "";
20     $price = "";
21     $quantity = "";
22     $sold = "";
23     //select the query
24     $query = "SELECT * FROM addproduct WHERE itemnumber = $itemnumber";
25     //execute the query
26     $result = $connection->query($query);
27
28     // conditional statement to get the record from the database
29     while($row = mysqli_fetch_array($result)){
30         $itemname = $row["itemname"];
31         $wholesale_price = $row["wholesale_price"];
32         $price = $row["price"];
33         $quantity = $row["quantity"];
34         $sold = $row["sold"];
35     }
36
37     //update the record once it has been changed
38     if(isset($_POST['update'])){
39         $itemnumber = $_GET['edit'];
40         $itemname = $_POST['itemname'];
41         $wholesale_price = $_POST['wholesale_price'];
42         $price = $_POST['price'];
43         $quantity = $_POST['quantity'];
44         $sold = $_POST['sold'];
45
46         // the following command will insert the data in addproduct table.
47         $query = "UPDATE addproduct SET itemnumber = '$itemnumber', itemname= '$itemname', wholesale_price = '$wholesale_price', price = '$price', quantity = '$quantity', sold = '$sold' WHERE itemnumber = $itemnumber";
48
49         //execute the query
50         mysqli_query($connection, $query);
51
52         //once update go back to all product page
53         header('location: allproduct.php');
54     }
55 }
```

Figure 4.7: edit.php

4.2.5 Delete product

The delete.php will delete the entire record from the database. This was done by using the GET method where I have to use the item number as id. Based on item number it will look for a record in the database and once the record is found it will run a query (DELETE FROM ‘addproduct‘ WHERE item number = \$itemnumber) where it will delete the record. After deleting the record, the page will be refresh using the header method in line 30.

CHAPTER 4. IMPLEMENTATION



The screenshot shows a code editor window with the file 'delete.php' open. The code is written in PHP and performs a database delete operation. It starts by establishing a connection to a MySQL database named 'Inventory_check'. It checks if a record with a specific item number exists and if so, deletes it. Finally, it redirects the user back to the 'allproduct.php' page.

```
<?php
//the following will establishing the connection to the server
$database_host = "localhost";
$database_username = "root";
$database_password = "";
$database_name = "Inventory_check";

// Create the connection
$connection = new mysqli ($database_host, $database_username, $database_password, $database_name);

//check the connection
if (mysqli_connect_error()){
    die('Connection Error ('. mysqli_connect_errno() .') '
    . mysqli_connect_error());
}
else {
    //conditional statement to delete the record from the database
    if(isset($_GET['delete'])){
        // get the id from itemnumber
        $itemnumber = $_GET['delete'];

        // run the query to delete the record
        $query = "DELETE FROM `addproduct` WHERE itemnumber = $itemnumber";

        //execute the query
        mysqli_query($connection, $query);

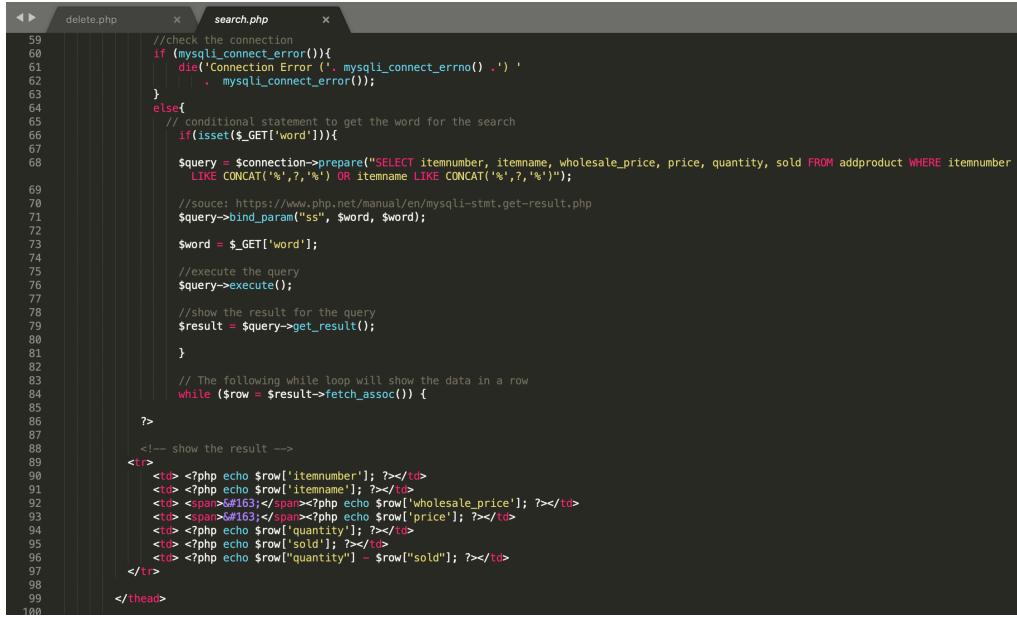
        //going back to the same page when the record is deleted
        header('location: allproduct.php');
    }
}
// Close the connection
$connection->close();
?>
```

Figure 4.8: delete.php

4.2.6 Search product

The user can able to search the specific product using the item name or item name. This is one of the main features to add in the system as there will be more than a thousand products in the database and users cannot look for every record therefore by making the search functionality will be easier to look for exact data. This was built by using the GET method where (if(isset(\$_GET['word']))) if statement will look for the record based on what user have input in the search field. It will then run the query in line 68 (Figure 4.9) and retrieve the result.

CHAPTER 4. IMPLEMENTATION



The screenshot shows a code editor window with the file 'search.php' open. The code is a PHP script for searching items in a database. It includes error handling for MySQL connections, preparing a SELECT query with a WHERE clause, executing the query, and displaying the results in an HTML table. The code is annotated with line numbers from 59 to 100.

```
59 //check the connection
60 if ($mysqli->connect_error()){
61     die('Connection Error (' . mysqli_connect_errno() . ') '
62         . mysqli_connect_error());
63 }
64 else{
65     // conditional statement to get the word for the search
66     if(isset($_GET['word'])){

67         $query = $connection->prepare("SELECT itemnumber, itemname, wholesale_price, price, quantity, sold FROM addproduct WHERE itemnumber
68             LIKE CONCAT('%',?, '%') OR itemname LIKE CONCAT('%',?, '%')");
69         //source: https://www.php.net/manual/en/mysqli-stmt.get-result.php
70         $query->bind_param("ss", $word, $word);
71
72         $word = $_GET['word'];
73
74         //execute the query
75         $query->execute();
76
77         //show the result for the query
78         $result = $query->get_result();
79
80     }

81     // The following while loop will show the data in a row
82     while ($row = $result->fetch_assoc()) {
83
84
85     ?>
86
87     <!-- show the result -->
88     <tr>
89         <td> <?php echo $row['itemnumber']; ?></td>
90         <td> <?php echo $row['itemname']; ?></td>
91         <td> <span>#163;</span><?php echo $row['wholesale_price']; ?></td>
92         <td> <span>#163;</span><?php echo $row['price']; ?></td>
93         <td> <?php echo $row['quantity']; ?></td>
94         <td> <?php echo $row['sold']; ?></td>
95         <td> <?php echo $row['quantity'] - $row['sold']; ?></td>
96
97     </tr>
98
99     </thead>
```

Figure 4.9: search.php

4.2.7 Analyse product

In order to create a chart to analyse the data, I have used the google chart API that allows me to create a graphical chart from a database record. There is a number of different charts with this library such as line, spline, area, pie, table, bar charts, etc. In my system, I have used bar chart and table chart, where bar chart will show the number of sold time, ordered by ascending and table chart shows information about the total cost from the wholesale price, in-store price, profit between wholesale product/in-store and total sale on a sold item. Figure 4.10 (Line 93), fetch all the results from the database and convert those data in a chart format.

CHAPTER 4. IMPLEMENTATION

```

79      <!-- source: https://developers.google.com/chart/interactive/docs/gallery/columnchart -->
80      <!-- I have got the following code from google chart -->
81      <script type="text/javascript">
82          google.charts.load('current', {'packages':['bar']});
83          google.charts.setOnLoadCallback(drawChart);
84
85          // function for drawchart
86          function drawChart() {
87              var data = google.visualization.arrayToDataTable([
88                  ['itemname','sold'],
89
90                  <?php
91
92                      // conditional statement to get the result of sold item
93                      if(mysqli_num_rows($result) > 0){
94
95                          while($row = mysqli_fetch_array($result)){
96
97                              //display a message
98                              echo "[".$row['itemname'].",".$row['sold']."],";
99
100                         }
101
102                     ?>
103                 ]);
104
105             // give a title to the chart
106             var options = {
107                 chart: {
108                     title: '1. Chart shows the number of sold time, ordered by ascending',
109                 }
110             };
111
112             // show the graph in column wise
113             var chart = new google.charts.Bar(document.getElementById('columnchart_material'));
114
115             chart.draw(data, google.charts.Bar.convertOptions(options));
116         }
117     </script>
118

```

Figure 4.10: analystic.php

4.3 MySQL

The application use a back-end database which was implemented using MySQL database. The following figures are the list of tables that has been used in the system.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	itemnumber	int(6)			No	None			Change Drop More
2	itemname	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	wholesale_price	text	utf8mb4_general_ci		No	None			Change Drop More
4	price	text	utf8mb4_general_ci		No	None			Change Drop More
5	quantity	int(11)			No	None			Change Drop More
6	sold	int(11)			No	None			Change Drop More
7	time	timestamp			No	current_timestamp()	ON UPDATE CURRENT_TIMESTAMP()		Change Drop More

Figure 4.11: Product Table

CHAPTER 4. IMPLEMENTATION

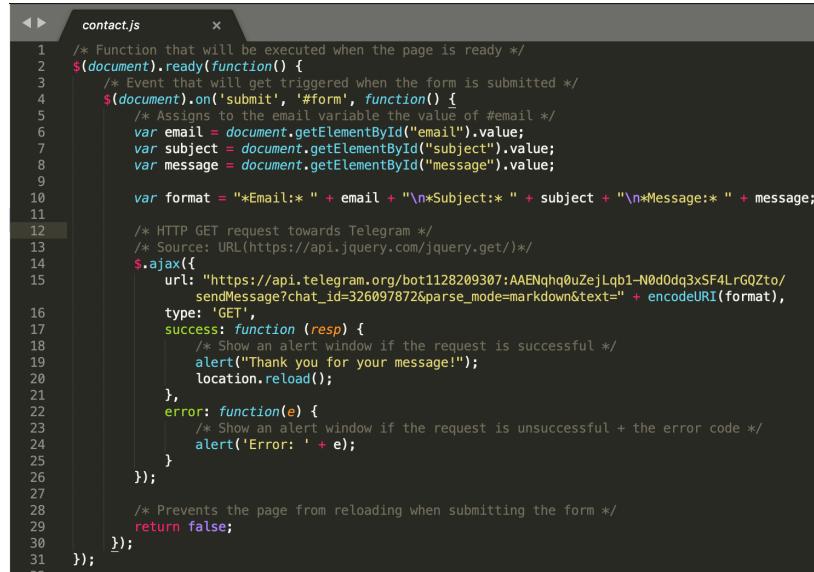
This feature has been implemented in the system yet but will be added in further development.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop ▾ More
2	firstname	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More
3	lastname	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More
4	email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More
5	password	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More

Figure 4.12: Register/Login Table

4.4 Telegram API

To make the contact form interact with the developer, I have used the Telegram API which will allow me to make a program that uses a telegram message for an interface. This means all the message submit in the contact form will be displayed in the telegram app. Inline 14-26, where HTTP GET the request toward telegram and show the alert when the message has successfully sent in line 19. In line 10 is the format that enables how to show the message in the telegram app.



```

1  /* Function that will be executed when the page is ready */
2  $(document).ready(function() {
3      /* Event that will get triggered when the form is submitted */
4      $(document).on('submit', '#form', function() {
5          /* Assigns to the email variable the value of #email */
6          var email = document.getElementById("email").value;
7          var subject = document.getElementById("subject").value;
8          var message = document.getElementById("message").value;
9
10         var format = "*Email:* " + email + "\n*Subject:* " + subject + "\n*Message:* " + message;
11
12         /* HTTP GET request towards Telegram */
13         /* Source: URL(https://api.jquery.com/jquery.get/)*/
14         $.ajax({
15             url: "https://api.telegram.org/bot1128209307:AAENqhq0uZejLqb1-N0d0dq3xSF4LrGQZto/
16                 sendMessage?chat_id=326097872&parse_mode=markdown&text=" + encodeURIComponent(format),
17             type: 'GET',
18             success: function (resp) {
19                 /* Show an alert window if the request is successful */
20                 alert("Thank you for your message!");
21                 location.reload();
22             },
23             error: function(e) {
24                 /* Show an alert window if the request is unsuccessful + the error code */
25                 alert('Error: ' + e);
26             }
27         });
28
29         /* Prevents the page from reloading when submitting the form */
30         return false;
31     });
32 });

```

Figure 4.13: Example of API code

CHAPTER 4. IMPLEMENTATION

4.5 Gantt Chart

A Gantt chart is mainly used for managing the project and is one of the best ways of showing activities displayed against time. It is an effective tool when managing a complex project with many dependencies. I have used this methodology to manage my time during the project so I can complete all the tasks that I have given to myself on time. I had shared this chart with my supervisor so he will know what task I have finished and what task I am currently working on. In figure 4.14 has the number of activities along with the time scale. Each activity is represented by a task name, duration, start date, end date, and comment if I missed any of the deadlines. This will ensure that I need to complete each activity within the time that I have specified in the chart, Therefore I can deploy the project on time.

Start date: 15/09/2019 AND End Date: 12/06/2020				
Task Name	Duration	Start Date	End Date	Comment
Project Plan				
■ Project Blog (Setting Up)	1 day	15/09/2019	01/11/2019	
■ Project Ideation	10 days	29/09/2019	30/09/2019	N/A
■ Market Research	1 week	20/09/2019	30/09/2019	N/A
■ Looking similar Project	5 days	31/10/2019	07/11/2019	N/A
■ Speaking to supervisor	1 day	08/10/2019	13/10/2019	N/A
■ Asking Friends	2 days	14/10/2019	15/10/2019	N/A
■ Research about the software	2 days	16/10/2019	18/10/2019	N/A
■ Project Specification	10 days	19/10/2019	21/10/2019	N/A
		22/10/2019	01/11/2019	N/A
Requirement Specification				
■ Interview Local Shops	3 days	02/11/2019	05/11/2019	N/A
■ Observing similar system	1 week	07/11/2019	14/11/2019	N/A
■ Create Gantt Chart	4 days	15/11/2019	19/11/2019	N/A
■ Wireframe	2 days	20/11/2019	22/11/2019	N/A
■ Break from this project to complete other module assignment	3 weeks	23/11/2019	14/12/2019	N/A
■ Interim Report	2 weeks	15/12/2019	29/12/2019	This task has not been complete because of new year break.
	1 week	30/12/2019	05/01/2020	Therefor it is been overdue by 1 week
Implementation				
■ Build the program	9 weeks	06/01/2020	08/03/2020	
■ Draft Report	18 days	09/03/2020	27/03/2020	
■ Test the program	2 weeks	28/03/2020	11/04/2020	
■ Solve the Program if their is any during testing	1 weeks	12/04/2020	19/04/2020	
Review				
■ Ask friend to test the program	2 days	20/04/2020	22/04/2020	
Create Final Report				
■ Create the presentation	30 days	23/04/2020	31/05/2020	
■ Prof read the report	4 days	01/06/2020	04/06/2020	
	5 days	05/06/2020	10/06/2020	
END OF THE TIMELINE				

Figure 4.14: Gantt Chart

Chapter 5

Testing

After development different testing was conducted to improve the system and limit the bugs. When features were implemented, I have reviewed the code that ensures any defective areas in the code and also improves the quality of the code such as adding comments, better code layout, etc. I have implemented two different types of testing such as normal testing (Tested by myself) and user testing (Tested by my friend). User testing was supposed to be done by my uncle but due to the Covid-19 virus, I could not able to go outside my house and my uncle does not know how to install the required software to run the program. Therefore I have decided to give my program to one of my friend JayKumar who has tested my program using a conference call on the Zoom app.

5.1 Normal Testing

Test No	Description	Expected Result	Actual Result	Pass/Fail
1	Check all the navigation link	Should take to right page	Has taken to correct page	Pass
2	User should able to login	Should able to redirect to product page	Does not take anywhere	Fail
3	User should able to create a new account	Should able to redirect to login page	Does not take anywhere	Fail

Continued on next page

CHAPTER 5. TESTING

Table 5.1 – Continued From Previous Page

Test No	Description	Expected Result	Actual Result	Pass/Fail
4	User clicked on View all product button from product page	Should able to redirect to view all product page	It has taken to view all product page	Pass
5	User clicked on add product button from product page	Should able to redirect to add product page where user can add product	It has taken to add product page where user can add product	Pass
6	User use the search functionality to search for specific product	Should able to show relevant product based on user search	It shows all the relevant product	Pass
7	User should able to edit the product using edit button from view all product page	Should able to take the user to edit form	It has taken the user to edit form where they can edit the data	Pass
8	User should able to delete the product using delete button from view all product page	Should able to delete entire record from the database	It has deleted the entire record from the database	Pass
9	User should able to see correct remaining quantity value	Should able to change the value automatically	It has changed the remaining quality automatic	Pass
10	Database need to display all the product in the view all product page	User should able to see all the product	User can see all the product	Pass
11	User should able to add product in the database from add product page	Ability to add the product in the database	It has successfully added the product	Pass

Continued on next page

CHAPTER 5. TESTING

Table 5.1 – Continued From Previous Page

Test No	Description	Expected Result	Actual Result	Pass/Fail
12	User should able to see an error message if there is no record in the database	Display the message	The message shown when there is no records	Pass
13	Error message if the user forgot to fill the field when adding product	Display alert ("Please fill in this field")	Shows the alert in user screen	Pass
14	User should able to view the chart from analytic page	Display the correct chart	shows the right chart	Pass
15	User should able to send a message using contact form	Message should be sent successfully	It has sent the message successfully	Pass
16	User should able to logout from the system	Take back the user to login page	Does not take anywhere	Fail

5.2 User Testing

It is used to help me to get a better understanding of the system from my friend. The feedback that was received by my friend will help me to improve the system performance, design in future development. This could be added before deploying the system in the market. I have asked one of my friends to test the program and give useful feedback through an online conference in Zoom. I have asked a number of different question related to user experience and features of the program.

The following lists are the question and his reply while testing the program.

1. How easy was it to navigate around the website?

It was very easy to navigate around the site as you have kept the site simple and easy to use.

2. Does all the navigation links takes you to the right page?

It does take me to the right page when I click on the links but the

CHAPTER 5. TESTING

logout button does not do anything. This feature will be added in future development.

3. Were you able to view all product?

Yes, I can see all the items in the view of all product pages that I have stored in the database using add product form.

4. Were you able to add the product in the database?

Yes, I was able to add the product using the form in the add product page and I can see the newly added product in view all product page at the beginning of all the items.

5. Were you able to edit the product?

Yes, when I clicked on edit button it took me to edit form where I have edited and update the product.

6. Were you able to delete the product?

Yes, I was able to delete the product but it could be useful to add a confirmation dialog box. This feature will come in future development.

7. Were you able to search a product using search box?

Yes, I was able to see all the relevant product when I type the word "A". It shows me all the products that contain the letter "A". I have even tried to search a product using item numbers to find the specific product and in return, it shows me the correct data.

8. How useful was the chart?

It was very useful to see the number of sold items in a bar chart format and also to see the total cost from the wholesale price, in-store price, profit between wholesale product/in-store, and total sale on a sold item. This was the best feature that you have added to your website.

9. How easy to contact the developer?

It was very easy to contact the developer using the contact form and once I submit the form it gave me a useful message as "Thankyou for your message!"

10. Do you consider the app to be helpful?

Yes of course as it will help all the shop keeper to manage their inventory who are using spreadsheets until now.

11. **Will you recommend this website to your friend and family who own business?**

Yes, this was the best app I have ever seen for the local shopkeeper.

12. **Is there anything else I should know about the system?**

Yes, Your login, logout, and create new account functionality is not working. I know and this feature will be added in a later version of the website.

5.3 Deployment

Once the application is ready and successfully released on the server, an email will be sent to my clients. This will ensure that the system is successful reaches the users and the required features are delivered successfully. The following list is a procedure for deploying the system:

1. Make sure everything is done which includes adding all the important features.
2. Making sure all the functions are working properly.
3. Email the clients for a meeting where I will present the application.
4. If everything goes well then sign off documents to use the system free of charge.

Chapter 6

Conclusion

This chapter classifies the system in terms of successes, failures, evaluation, and future development for an inventory check. While making the system, I have gained a lot of useful knowledge, skills, and experience when making software by my self. When deploying the system to my uncle and other shopkeepers, I have received so many positive feedback via message.

6.1 System Successes

Overall the project has been successful where design, technical requirements, and system development meet all the requirements that were set out. The main success was using a system to store the data in the database are met.

The implementation of the front-end allows the user to interact throughout the site with a number of different features. This includes the right choice of color, easy to navigate around the site, the best bar chart to show relevant data, and easy contact options to the developer. The system loads up very quickly when the user has a stable internet connection of less than 1Mbps speed. This was one of the successes with the front-end development.

The implementation of the back-end server using PHP Script and MySQL to interact with the database to retrieve and store the data. The back-end has all the ability to store all kinds of data such as storing products in the database. All the features for back-end service are working perfectly such as view all products, add products, edit products, delete the product, and show

this data in chart-wise. The contact form has Telegram API ability to send the message in developer telegram messenger app.

In the user survey, inventory check was satisfied with the result received through the survey. This was very helpful which gave me an understanding of what their requirement is. The majority of people love to use my system. By gathering all this data has helped me to make a successful system.

During the user testing, I have received a lot of positive feedback for my system. It has found that the application is easy to navigate and use throughout the site. The tester has no difficulties with all the features that are currently available within the system. The user can access the map showing the location on the contact page where they can see the developer location with a contact form as well.

6.2 System Failures

There are some failures in the system which cannot be resolved due to time constraint. There are many features that are not has been implemented with this version of the software but will be implemented in future developments. The following list is the system failures that were found during the user testing and normal testing.

6.2.1 Register

User has found that they cannot able to see the create new account page. There was an error with the code due to the bootstrap 4 library model. They have also found that the cannot able to register the new user as this feature has not been implemented with the database.

6.2.2 Login

User has found that they cannot able to login into the system. The system currently does not have this feature where the user can log in to the system. They can access all the pages without login credentials.

6.2.3 Mobile Friendly (Responsive)

The user has found that some of the website pages are not mobile-friendly especially the homepage. This feature could be solved by using media queries for different viewport so the user can view the website in any device without any difficulties.

6.2.4 No alert for Internet connection

User has found that they cannot see any alert message for no internet connection. This feature has not been added in the system but this can be in future development for reminding the user to turn on the internet.

6.2.5 Alert for Delete product

The user has found that there is no confirmation box when deleting the product from the database. This feature has been implemented in the system but this could be done using JavaScript alert command.

6.3 Future Developments

Inventory check is ready to deploy in the market as a finished product but there are some features that will come in future development. The following list will be built in future developments.

6.3.1 Login & Register

Currently, The login and register form has been build where user can add their credential information to login into the system but the functionality for back-end system will be built in future development where all the pages will be locked until user login into the system. When making the login and register feature, I will need to use a password hashing method to make sure that login credentials are secure on the server-side and protected from the hackers. In the database, the password will be encrypted with the hashing method which makes it complex for a malicious person to view the password.

6.3.2 More charts

There are only two charts available in the system for now where user can analyse the data. In the future, my plan is to build a customisation feature where the user chooses the list of the data which will then convert into the charts.

6.3.3 Filter Features

I have tried to add this feature in view of all product pages where users can filter the product in ascending order, descending order, time, stock, quantity, remaining stock, etc. But due to time constrain I have decided to add this feature in future development. Currently, The user can use search functionality until the filter features are added in the next application update.

6.3.4 POS system

This will be one of the most useful features to add in the future update where the database will update automatically with POS (Point of Sale). This system will update the stock availability, sold item, etc.

6.3.5 Barcoding

This feature will help the user to not making a mistake when adding a product in the database. They can add an item by scanning the barcode to give reliable and fast solutions to record the item. Which this feature they can easily find the stock availability and price by scanning the barcode of the item.

6.3.6 Alerts

By adding a feature in the system will give alert to the user via on-screen notification when the item is low in stock, daily sales, and any error in the system. This will help the user to find the issues more quickly and receive alerts.

6.3.7 Responsive site

Currently, there are some pages that is not mobile-friendly which was found in user testing by my friend. This will be fixed using the media query in future development so the user can view the website on their mobile without any problem.

6.3.8 Alert for delete product

While doing the unit testing, the website does not give a confirmation dialog box when deleting the product. This feature will be added in a later version where the system asks for confirmation users such as "Do you want to delete this product" and users need to select by two given options "YES" or "NO".

6.3.9 PDF file

This feature will allow the user to download all products in PDF file where they can find useful items such as low stock items, reaming stock, etc.

6.3.10 Protect user data

Inventory check application holds user data in the database server and not fully protected. There will be a risk for data theft so in future versions all the important data will be encrypted to save from hackers.

6.3.11 Required hardware

Many features will come in future development will require specific hardware to interact with the database such as POS till for automatic update the sold item, scanning and barcoding mechanism for adding item or view item information.

6.4 Evaluation

This project was aimed to tackle inventory management in local retail shop-keepers. Inventory check was selected to enhance the issues that they are currently facing and providing them with the best system for free. Stakeholder gathering was valuable which laid out the bunch of problems that they

CHAPTER 6. CONCLUSION

are currently facing with their current system. This gave me an idea of what their requirement is with the new system. Having strong planning across the project was the main success of the project to finished on time. When I fall sick, I had to manage my time more efficiently and finish the outstanding task to keep my self on target. During the design stage of the project, I need to ensure that stakeholder is happy with the design.

It was a complex system to build that uses three various technologies such as Front-end, back-end, and MySQL database. This was the biggest achievement I have made when making complex software with limited time on my own. It was a fun project to make which will helps hundred of a shopkeeper who use the spreadsheet to manage their inventory and I am happy that I have solved their problem by making this application. I have faced a lot of problems when building this application such as learning new technologies, fixing the problem within the time specified in the Gantt chart, fixing the database server connection with PHP script, etc. Using Gantt chart was allowed me to manage my time and finished the project on time.

I have never thought at the start of the project that I would build a complicated system, But now I am happy that I have made this system.

Appendix A

Questionnaire/Survey Result

Inventory Check

The aim of this questionnaire to collect information before started building a program. It is help me to find the current situation that you are facing in your shop. Thankyou!

*Required

Name *

Your answer

Do you use any system to manage your inventory?

Yes
 No

If yes, What system do you use to manage your inventory?

Spreadsheet
 None of them
 Other: _____

Figure A.1: Question 1,2 and 3

APPENDIX A. QUESTIONNAIRE/SURVEY RESULT

How much does it cost you to use inventory management system?

£30 - £50 Monthly
 £51 - £100 Monthly
 £101 - £150 Monthly
 £150 - £200 Monthly
 None

Roughly, how often do you use inventory system if you have one.

Daily
 Weekly
 Monthly
 Rarely

Would you like to have a inventory system for free?

Yes
 No
 Maybe

Figure A.2: Question 4,5 and 6

APPENDIX A. QUESTIONNAIRE/SURVEY RESULT

Would you like to have the ability to manage your inventory from anywhere in the world?

Yes

No

Would you prefer to use my system to manage your inventory or use other famous system

Yours

Other

Submit

Figure A.3: Question 4,5 and 6

Full responses for the survey can be found in git-lab. Please use the following link where you can find all the responses in the spreadsheet file:
https://gitlab.doc.gold.ac.uk/hrame001/inventory-check-system/-/blob/master/Final%20Report/Inventory_Check_Responses.xlsx

Appendix B

Gitlab program link

The following links are the inventory check program link where you can find all the program files to run the program. I have already permitted you as a maintainer to check out the application codes in gitlab.

https://gitlab.doc.gold.ac.uk/hrame001/inventory-check-system/-/tree/master/Inventory_Check

Programme folder called Inventory check where you will find all the related program files.

Appendix C

How to run the program?

The following command is a list of instructions that you need to take to run the file.

1. Download the XAMPP software in your system.
2. Open the application and start the server from the general list.
3. Start all the services from the services tag.
4. Click on Volumes, select on the mount, and click on explore.
5. Last command opens a new window where you need to select htdocs and paste the Inventory_check folder from GitLab.
6. once the file is uploaded in the server you can run using this link:
http://192.168.64.2/Inventory_Check/index.php

Bibliography

- [1] Dearsystems: [Online] (Last updated: Unknown) <https://dearsystems.com/pricing-table/>)
- [2] Unleashed: [Online] (Last updated: Unknown) <https://www.unleashedsoftware.com/gb/pricing>
- [3] Panoply Blog: [Online] (Last updated: Unknown) <https://blog.panoply.io/mongodb-and-mysql>
- [4] Existing system: [Online] (Last updated: unknown) <https://www.theukdomain.uk/6-powerful-inventory-tools-for-your-small-business/>
- [5] Cin7: [Online] (Last updated: unknown) <https://www.business.org/finance/cost-management/best-inventory-management-software/>
- [6] jQuery hide and show method: [Online] (Last updated: unknown) https://www.w3schools.com/whatis/whatis_jquery.asp
- [7] Bootstrap: [Online] (Last updated: unknown) [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- [8] Brackets & Sublime: [Online] (Last updated: unknown) <https://www.elegantthemes.com/blog/resources/best-code-editors>)
- [9] Michael Severn report: [report] (Last updated: unknown) <https://learn.gold.ac.uk/course/view.php?id=11997>)
- [10] Abdul Wahid report: [report] (Last updated: unknown) <https://learn.gold.ac.uk/course/view.php?id=11997>)