Faculty of Electronic & Computer Technology and Engineering

STOCK MANAGEMENT SYSTEM USING WEB AND DATABASE SERVER

MUHAMMAD AMIRUL AMIN BIN ARSAH

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2024

STOCK MANAGEMENT USING WEB AND DATABASE SERVER

MUHAMMAD AMIRUL AMIN BIN ARSAH

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electronics & Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Stock Management System Using Web and Database Server
Sesi Pengajian : Semester 1 2023/2024

Saya MUHAMMAD AMIRUL AMIN BIN ARSAH mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

□ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

Amin

(TANDATANGAN PENULIS)

TS. MOHD SAAD BIN HAMID
Pensyarah
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: No 176 Lorong 10 Taman Muhibbah 27000 Jerantut Pahang

Tarikh: 14/1/2024

Tarikh: 16/12024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled “Stock Management System Using Web and Database Server” is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : Amin

Student Name : Muhammad Amirul Amin Bin Arsah

Date : 14/1/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature	:	
Supervisor Name	:	TS. MOHD SAAD BIN HAMID
Date	:	16/1/2024

DEDICATION

To my beloved mother, Norehan Binti Hashim,

whose unwavering support, wisdom, and love have been my guiding light throughout this academic journey. Your sacrifices and encouragement have shaped my pursuit of knowledge, and I dedicate this thesis to you with profound gratitude and immense love.

To my coursemates in 4BEEC,

whose camaraderie and shared challenges have made this academic endeavor both enriching and memorable. Together, we've navigated the complexities of our coursework, and I am grateful for the friendships forged and the collective strength we've found in each other.

To my friends,

for their constant encouragement, laughter, and understanding. Your presence has been a source of joy and inspiration, providing balance to the demands of academic life. This thesis is dedicated to the cherished connections that have made this journey brighter and more fulfilling.

ABSTRACT

The project aims to verify that the system accurately reflects the number of things currently housed in the storage area. It is not necessary to perform the stock check once every week. Then, the system requires each member of the F&B staff to state what is take out from the store daily. The system displayed what remains. Every piece of data will be saved in a safe place and made available to everyone for periodic updates. System developed using MySQL as database to save data and the user interface for this system created for a web app optimised for mobile device usage and a web browser for use on tablets and personal computers. Then, every workforce member can utilise their mobile phone and a browser to check on what is still available from any location and at any time. Any member of the F&B staff can use the graphical user interface. This will allow the management or the administrative staff to keep track of the number of things still in storage while saving sometime during the weekly stock check.

ABSTRAK

Projek ini bertujuan untuk mengesahkan bahawa sistem itu menggambarkan dengan tepat bilangan perkara yang kini ditempatkan di kawasan storan. Ia tidak perlu melakukan pemeriksaan stok sekali setiap minggu. Kemudian, sistem memerlukan setiap ahli kakitangan F&B menyatakan apa yang dibawa keluar dari kedai setiap hari. Sistem telah memaparkan apa yang tinggal. Setiap bahagian data akan disimpan di tempat yang selamat dan disediakan kepada semua orang untuk kemas kini berkala. MySQL telah digunakan untuk menyimpan setiap jumlah data. Antara muka pengguna untuk sistem ini telah dibuat untuk aplikasi web yang dioptimumkan untuk penggunaan peranti mudah alih dan pelayar web untuk digunakan pada tablet dan komputer peribadi. Kemudian, setiap ahli tenaga kerja boleh menggunakan telefon mudah alih dan penyemak imbas mereka untuk meny emak perkara yang masih tersedia dari mana-mana lokasi dan pada bila-bila masa. Mana-mana ahli kakitangan F&B boleh menggunakan antara muka pengguna grafik. Ini membolehkan pihak pengurusan atau kakitangan pentadbiran menjejaki bilangan barang yang masih dalam simpanan sambil menjimatkan masa semasa pemeriksaan stok mingguan.

ACKNOWLEDGEMENTS

I extend my profound gratitude to the esteemed Ts. Mohd Saad Bin Hamid, whose invaluable guidance, words of wisdom, and unwavering patience have been the bedrock of my journey through this project. His mentorship has been nothing short of instrumental in shaping the trajectory of my academic pursuit.

Heartfelt appreciation is also extended to Universiti Teknikal Malaysia Melaka (UTeM) and Majlis Amanah Rakyat (MARA) for their generous financial support through the MARA scholarship, a beacon that illuminated the path to the successful completion of this ambitious project.

I am indebted to my esteemed colleagues in 4BEEC and my gracious housemates, whose collaborative spirit and willingness to share insights and ideas have enriched the tapestry of this endeavor.

A pinnacle of thanks ascends to my parents and family members for their unwavering love and prayers, serving as a steadfast pillar of support throughout the arduous yet rewarding period of my academic journey. Special acknowledgment is reserved for my academic beacon, Ts. Dr. Abdul Halim Bin Dahalan, whose motivational prowess and empathetic understanding have been a driving force behind my academic accomplishments.

This acknowledgment is not just a mere expression of thanks; it is a resounding tribute to the constellation of mentors, supporters, and well-wishers who have propelled me to triumph.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter includes an overview of the project's background, issue statements, objectives, and scope, among other vital details. Each subtopic comprises the necessary information, notable publications, and supporting evidence from previous researchers to support the effort and emphasis of this project.

1.2 Project background

Stock management has been changed by looking at how it developed nowadays. Commonly we see every data needs to be handled by the manager. This system has helped many businesses track how their business progresses. Many things should take part in this system to take it to another level. They allow users to monitor their products stored in the storeroom, keep track of their product stories, and manage their suppliers and customers. The system has provided real-time visibility to ensure users make informed decisions about making orders to suppliers and whether any item has been left in the storeroom. The system is vital to increasing Food and Beverages (F&B) productivity.

On the other hand, to make the staff job easier. Traditionally the manager needs to update the counting of the stock by calculating it manually[1]. This way is not efficient because the person who needs to figure it out manually is the staff. Staff need to handle their job instead of being given other tasks, which can burden the staff. Besides, staff shortage usually happens since the team needs to help the manager finish another job. In F&B, time is the key. A lack of staff may usually happen because some of them got emergency leave and have some personal problems. In that case, a stock management or inventory system was developed to solve the in-storeroom problems.

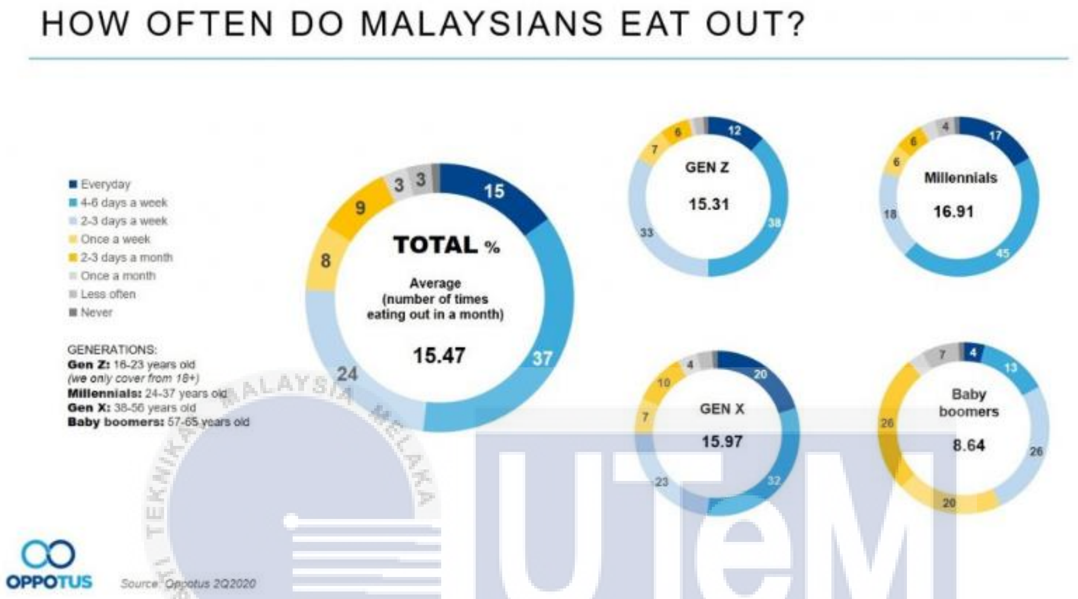


Figure 1-1 Figure statistic of how often do Malaysian eat out[2].

This statistic shows how frequent Malaysian eat out in a month. How can F&B not be affected by the crowd of customers? That is one of the main reasons why this system needs to be improvised to make the customer process to buy the food and the staff problem on lack of members can be solved.

Many ways to make this system are using a barcode scan, a key in the design, or another system already suggested by the developer. However, people's main issue is double-checking the stock; everything must be counted again to make the exact numbers. This system required staff to key in their stock daily to make consistent stock checks. Management needs to know what is left in store to ensure what they will need in future. In this way, even staff can understand what is left without checking details in the storeroom. Systems like this can help managers, administrators, or organizations know what they need to sustain to make money. It is an excellent way to use HTML to design the system and make it understandable for the staff. Applying MySql to stored data is an efficient way to ensure data is in a safe place.

1.3 Problem Statement

In a typical system, humans need to calculate stock manually weekly. The problem that will face is not the exact value due to human error. Keeping this habit also makes the restaurant quite busy because even if they need to count it. Lack of staff in the peak periods will always be the main problem in the restaurant[3]. Avoid staff shortage, this problem happens when stocks take during busy hours. Some of the staff need to count the items manually during working hours. Paper and pen can be saved instead of making an extensive logbook monthly just to keep the data in stock. Other than that, the record of products in the storeroom may be misplaced if written on paper or in a physical logbook[4]. Lost product always happens in a storeroom of F&B because some workers cannot apply First in First Out (FIFO). This may cause some loss to the company. The product in the storeroom is also not well managed. What will happen? Some items may be misplaced, and FIFO cannot be implemented.

To maintain accurate inventory records and streamline operations, staff must log the products they take out of the storeroom daily. This eliminates manual counting during busy periods and ensures up-to-date inventory information. Hence, it can sustain and consider waste reduction methods, such as using portion control techniques, monitoring and optimising ingredient usage, and applying eco-friendly procedures. Environmentally friendly, no more paper to waste since every data needs to be key in the system and save human energy. Examine how stock management can support environmental activities and advance sustainability goals.

Implementing such a system where staff members input the items they retrieve from the storeroom daily offers significant benefits, especially for larger companies. By consistently running this system, the company can effectively manage its inventory, which is essential for success.

1.4 Project Objective

The main aim of this project is to propose a system that can help humans count what is left in the storeroom. Specifically, the objectives are as follows:

1. To perform a literature study on how inventory or stock management work in the industry.

2. To develop a stock management system for the food and beverages industry.

3. To analyse the reliability of the stock management system

1.5 Scope of Project

Many aspects can be related to stock management in F&B (Food and Beverages), like inventory control, supply chain management, and overall operational efficiency—some of the areas that can be explored. The scope of the project can be specific:

a) The use of HTML as GUI.

b) MySQL is a database server to secure all the data.

c) Target users use web browsers to get into the system.

d) Inventory management: FIFO (First In, First Out) and JIT (Just-In-Time) to monitor stock and avoid many wastes.

e) To link PHP, HTML, and phpMyAdmin, set up a local working environment, create a database in phpMyAdmin, connect PHP to the database, work with the data, and embed HTML code in PHP.

f) CSS controls the colour, font, size, spacing between elements, how elements are placed and laid out, what background images or colours to use, and different displays for different devices and screen sizes.

CHAPTER 2

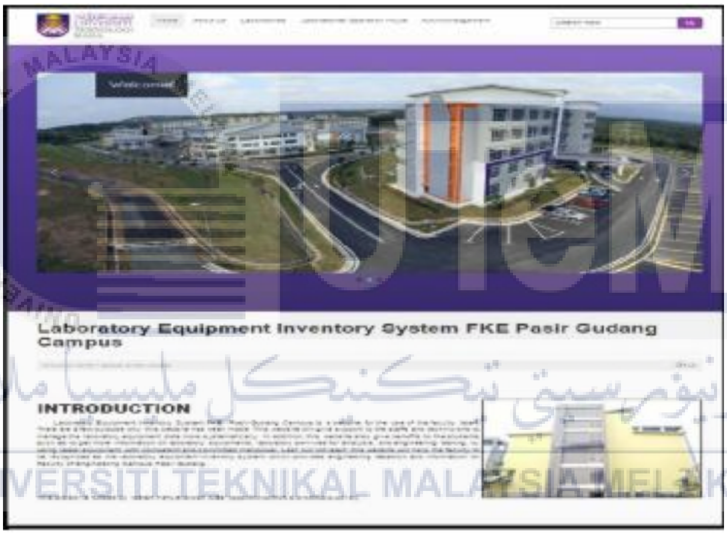
LITERATURE REVIEW

2.1 Introduction

The stock management system is a system that is used in the food and beverage sector to manage storage to keep track of the quantity of stock that is stored there. If it does not function effectively, this could result in a loss. There will be many issues, such as the paper with the data being misplaced, no FIFO at all, and other issues. Sometimes, the F&B crew does not warn customers of low stock. It's possible that restaurant patrons won't order what they want because of this media. The inventory system was designed to solve this problem precisely so that it would not continue to be a problem in the future. The information will be stored on a database server, which only certain people can access. It is only accessible to a select group of persons with authority, such as staff, managers, and administrative personnel.

2.2 Development of FKE UiTM Kampus Pasir Gudang Lab Equipment Web Database[5].

It was using Wampserver 2 and Joomla! Content management system, the document explains how a web database of laboratory equipment was created for the Fakulti Kejuruteraan Elektrik Kampus Pasir Gudang (FKE PG) organisation. The database was designed to assist laboratory personnel, lecturers, and students in recording, managing, and retrieving laboratory equipment and components information. To provide students with real-world experiences, FKE PG is home to 22 laboratories, each serving a unique purpose and outfitted with an impressive collection of apparatus and electronic components. As a result, it has been suggested that an online database be used to maintain the records of all of these electronic equipment and apparatus pieces. The deployment of the webserver, the creation of the database for the laboratory equipment website using Joomla!, and the development of the database itself are the three operations that make up the development of the database for the laboratory equipment website. There is information on the website regarding FKE PG and its laboratory facilities, as well as the laboratory staff and the equipment and components in each laboratory. Users can search for information on the lab's equipment and parts on the established website, including their location, quantity, serial number, and availability. The person in charge can provide an up-to-date status report on the components and equipment of the laboratory. At long last, the webserver is operated by connecting the website and the database using WampServer 2. The homepage for the system is stated below.



Laboratory Equipment Inventory System FKE Pasir Gudang Campus

INTRODUCTION

Figure 2-1 Homepage of Laboratory Equipment Inventory System FKE Pasir Gudang Campus[5]**.**

2.3 Fridge Inventory Management to Reduce Household Food Wastage[6].

This system aims to find a study on managing the inventory of refrigerators, which intends to cut down on food waste. The problem states that households frequently fail to maintain their refrigerators' list, resulting in perishable food that goes bad and is thrown away. The technique is broken down into two most important steps: analysis and design. The analysis process consists of data required to be collected via methods such as interviews, surveys, and the reading of relevant literature, in addition to system analysis. The notion of Satzinger, Jackson, and Burd's Waterfall System Development Lifecycle is used as the foundation for the design process, incorporating the detailed object-oriented design. The preliminary findings suggest that a cloud-based inventory management system for refrigerators can help decrease food waste, and one model under consideration enables users to manage the refrigerator in collaboration with other members of the community. In general, this study sheds light on the possible benefits that could result from a refrigerator inventory management system that is both more efficient and more effective.

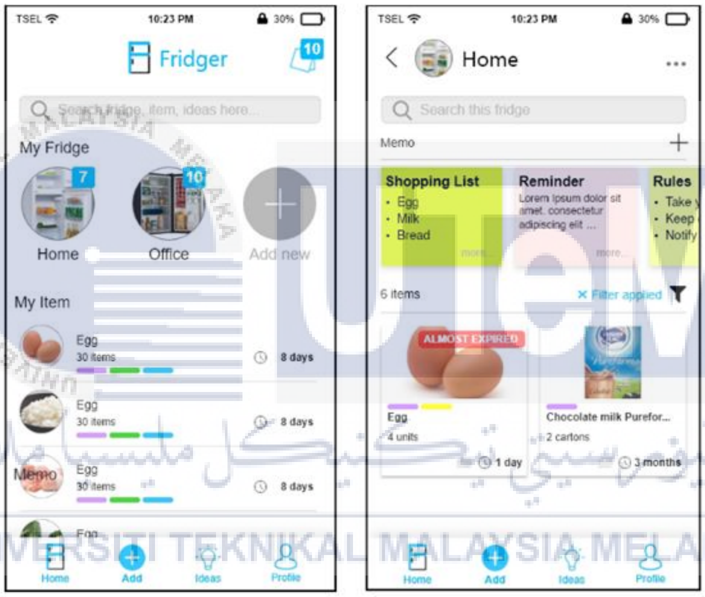


Figure 2-2 Sample of User Interface of apps Fridger[6]**.**

2.4 IoT-Based Smart Inventory Management System for Kitchen[7].

The Internet of Things (IoT) is being used in a project called Smart Kitchen Inventory (SKI), which intends to simplify and improve managing a home's food supply and other household goods.

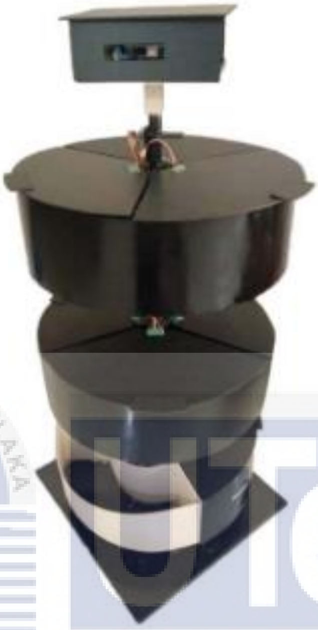


Figure 2-3 Figure of Smart Kitchen Inventory (SKI)[7].

The project entails the construction of a cabinet with nine separate compartments, each of which is fitted with a unique sensor to determine the number of food items that are kept inside it. However, there were several difficulties with the project on a technological level, such as developing sensors that were accurate and low-cost and having unstable connectivity between the hardware and the server. To overcome these obstacles, the project utilized LDR/LED sensors in conjunction with NodeMCU (ESP8266) with stable firmware. TCP and I2C connections convey information between the hardware system and the software, and a responsive website was developed with Bootstrap and core PHP. The system also has an administrative control panel that allows the user to monitor the database and change the order status and food inventory. According to the findings, the system is functioning correctly and does not include any flaws; furthermore, it is possible to cut costs by increasing the number of goods produced on a larger scale. Overall, the SKI project offers a straightforward and effective method of managing one's grocery inventory, contributing to increased digitalization in everyday life.

2.5 Inventory Management System for Education Institution[8].

The mechanism that is now in place is not only ineffective but also prone to errors. The inventory management system (IMS) that has been suggested is a software-based solution that handles inventory records effectively. Forms for logging in and registering, search filters, and stock management are included. MySQL, PHP, CSS, JS, WAMP, and LDAP are the technologies utilized in developing this system. The document above highlights the need to optimize resource distribution through information in a world with limited resources. The Object-Oriented Approach, the Unified Modelling Language (UML), and Rapid Application Development (RAD) are the development methodologies utilized to create this system. The system incorporates several features, such as a secure database for recording inventory records, centralized updating, and an unobtrusive user interface that is aesthetically pleasing and convenient. The document contains an entity-relationship diagram, which illustrates the relationships between entities and the attributes they possess in the database. The research concludes that the IMS is a practical, dependable, and economical option for inventory management in academic institutions. Soon, the scope will expand to include both an external implementation for stockyard management systems and a mobile application. The UI/UX of the Working Website is show below.

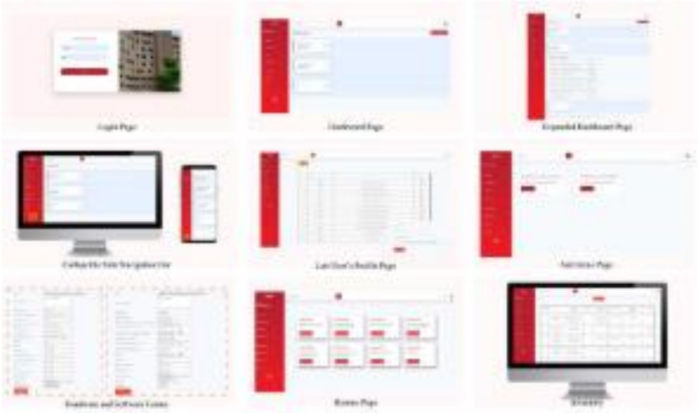


Figure 2-4 The UI/UX of the Working Website[8]**.**

2.6 Radio Frequency Identification System for Asset Tracking and Inventory Management in Hospitals[9].

The purpose of the system is to provide information in real-time on movable assets, with the end goal of combating the problem of theft within the medical industry. The system utilizes radio frequency identification (RFID) tags connected to assets like intravenous pumps, surgical equipment, beds, and wheelchairs. RFID readers scan these tags at checkpoints throughout the facility. After then, the data is checked with the prior checklist of items in the database, and the timing of when the tags arrived and left the database is recorded. The system has a straightforward user interface and requires no specialized training. The entirety of the work is carried out with the programming language C#, and the system is constructed using Microsoft Visual Studio 2012 in conjunction with Microsoft SQL Server 2008. Tracking assets can be accomplished with the help of the system in a hospital setting and retail and industrial environments. This system's contributions to the healthcare business include reducing the likelihood that products would be misplaced, saving both time and money, and introducing a novel, accurate, and less labour-intensive method of controlling inventory and avoiding theft in the healthcare sector that deals with patient care. Below write operation and asset price calculations are shown.

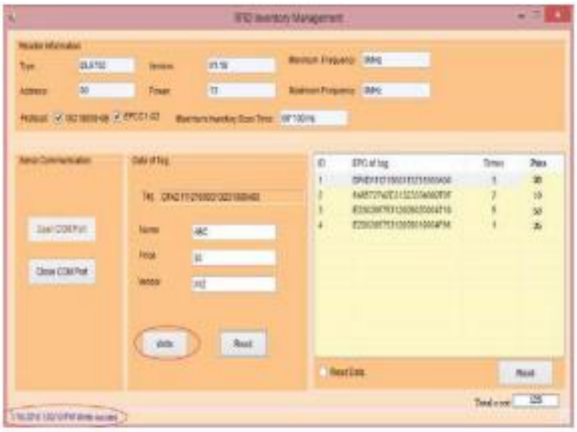
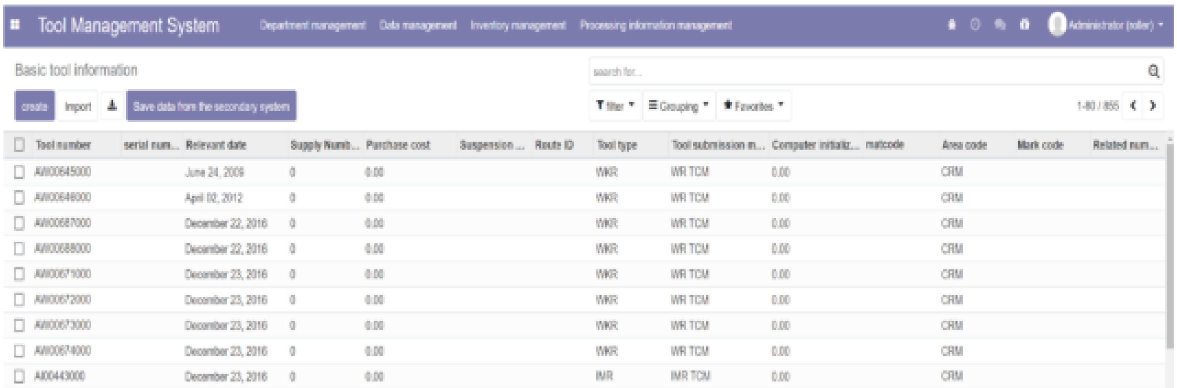


Figure 2-5 Figure of the write operation and asset price calculation[9]**.**

2.7 Visual Management System of Intelligent Factory Tools Based on Odoo[10].

An intelligent tool management system is developed for manufacturing facilities to enhance operational efficiency, optimise resource utilisation, and reduce costs. The software architecture is derived from the open-source Odoo framework, while the PostgreSQL database facilitates data storage and management. The system comprises three key components: the tool management system itself, data visualisation capabilities, and optical character recognition (OCR) for data recognition.



Basic tool information

Tool number	serial num...	Relevant date	Supply Numb...	Purchase cost	Suspension ...	Route ID	Tool type	Tool submission m...	Computer initializ...	matcode	Area code	Mark code	Related num...
AW00645000		June 24, 2009	0	0.00			WKR	WR TCM	0.00		CRM		
AW00648000		April 02, 2012	0	0.00			WKR	WR TCM	0.00		CRM		
AW00687000		December 22, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AW00688000		December 22, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AW00671000		December 23, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AW00672000		December 23, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AW00673000		December 23, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AW00674000		December 23, 2016	0	0.00			WKR	WR TCM	0.00		CRM		
AI00443000		December 23, 2016	0	0.00			IMR	IMR TCM	0.00		CRM		

Figure 2-6 Figure of the main interface of a tool management system[10]**.**

The tool management system plays a vital role in standardising tool management within the manufacturing environment, offering a visually appealing large-screen display for easy monitoring. It enables dynamic control of tool information, thereby contributing to improved operational efficiency. Additionally, the system incorporates a mobile OCR feature, allowing the users to access the system and retrieve essential tool information using their mobile devices. One notable strength of the system is its adaptability. It can be tailored to align with specific company plans, customer requirements, technological advancements, and environmental conditions. This adaptability ensures that the system remains relevant and responsive to the evolving needs of the manufacturing facility. The intelligent tool management system built upon the Odoo framework provides a comprehensive solution for manufacturing facilities. By fostering operational efficiency, optimising resource utilisation, and reducing costs through dynamic tool management, data visualisation, and mobile OCR capabilities, it empowers businesses to streamline their operations effectively. Moreover, its adaptability enables customisation based on unique business needs and changing external factors.

2.8 Process Monitoring and Inventory Management in Plastic Injection Molding[11].

The application was built with MySQL and Visual Web Developer Express Edition and tested using a genuine table database populated with experimental data. The experiment's findings demonstrated favourable satisfaction from both the users and the managing director, with higher productivity and reduced errors. The problem of in-house programmers producing custom software to meet the demands of an organisation is introduced in this section of the document. This problem arises because off-the-shelf software cannot meet individual users' requirements. Users are afforded the comfort of accessing the required data without contacting the workers of other departments, who are frequently out of the office. This is made possible by specialised software. This article provides an overview of developing a web application for monitoring and managing stock in the plastic molding injection industry. The issue discussed in this article is the necessity for a custom software solution for process monitoring and stock management in plastic molding injection. This is because off-the-shelf software cannot fulfill each customer's specific user requirements. Screen shot of the first page of the system shown below.

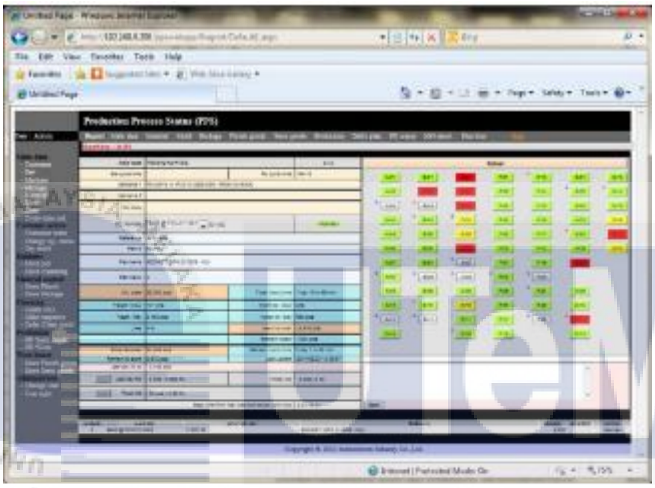


Figure 2-7 Screenshot of the first page[11].

The technique utilised in this paper includes selecting data from the database containing user records, verifying user passwords, and displaying menus per the user's rights. This paper describes the operating procedure, which begins with picking data from the database corresponding to the user record, then verifying the user password and displaying the menu per the user's entitlements. The experimental process, which includes testing and debugging the program, posting the web application on the server, and collecting user needs, is also described in the text. Software and Hardware Utilised The following pieces of software were utilised in creating this document: MySQL, MySQL GUI, and MySQL Connector 5.1.

Visual Web Developer Express Edition was also employed. This document makes use of the Windows Server 2003 operating system that is hosted on the company intranet.

Therefore, the experiment's findings demonstrated favourable satisfaction from both the users and the managing director, along with greater productivity and reduced errors. The users can complete their tasks more quickly and successfully, with fewer mistakes. Other specifics, such as mould maintenance, mould clean time before injection as sequence, mould clean when over keep age, clean mould express when change sequence, and current location of decay, are included in the programme.

2.9 Inventory Placement Mapping using Bluetooth Low Energy Beacon Technology for Warehouses[12].

An investigation into using Bluetooth Low Energy (BLE) beacons for warehouse inventory management, particularly in corrugated box manufacture. The project intends to solve the difficulties of locating items in massive warehouses that frequently rearrange their inventory. BLE beacons are utilised for zone mapping, indoor navigation, and asset tracking with the approach that has been proposed. The document also includes a comparison of BLE beacons with various other technologies used for indoor positioning and a discussion of the benefits and drawbacks of each. In the context of this project, the goal is to improve inventory management in manufacturing businesses by utilising BLE beacons and to give code snippets for implementing these solutions using Estimote's software development kit (SDK).Figure 2-8 shows the indoor navigation mobile application user interface.

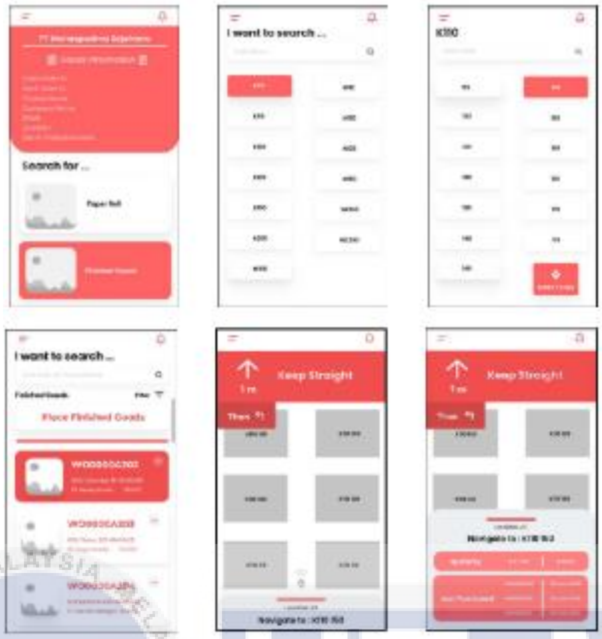


Figure 2-8 Figure of indoor navigation mobile application user interface[12].

2.10 Implementation of RFID based Lab Inventory System[13].

The system authenticates users by reading information from a contactless card or tag and utilising powerful RFID tags that do not require batteries. The plan that has been developed can be improved in the future by using RFID tags and antennae that are more efficient, and that can display the closeness of the equipment. This might help develop dependable machine-learning algorithms that can precisely localise laboratory equipment across the institution. Within the scope of this paper, they cover the complete system design, which encompasses both the software and the hardware components that have been implemented.

It is discussed how the proposed approach works, illustrating how the system is connected to its many different features.

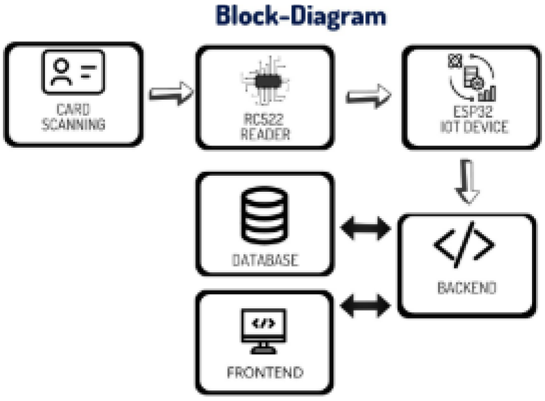


Figure 2-9 Block diagram of the project[13]**.**

In front-end development, technologies such as HTML, CSS, and JavaScript; in back-end development, MongoDB is used for data storage. Among the several pieces of hardware that were utilised were an ESP8266, an ESP32, an MRC522, and a Raspberry Pi. Because of this project, the authors learned about working with Arduino modules, NodeMCU modules, and Raspberry Pi and interfacing these devices with various sensors.

2.11 IoT In Inventory Management[14].

It discusses various methods and approaches, including the ABC analysis, the VED analysis, the safety stock method, and the Just In Time strategy, among others, for managing inventory. The application of radio-frequency identification (RFID) technology to inventory management is also covered in this document, along with an explanation of the various components that make up an RFID system, such as RFID tags, antennae, and readers. In addition, the text emphasises the significance of inventory management in warehouses, which has developed into an essential activity for companies over the past few decades. When it comes to making decisions concerning inventory management, it highlights how important it is for warehouse managers to balance the tangible and intangible aspects of the business. In addition, the document delves into the merits of utilising an Internet of Things cloud system for inventory management, optimisation of space, and intelligent shipping. This document offers helpful insights into the many facets of warehouse inventory management and the various technologies used to optimise it.

2.12 Internet of Things (IoT).

IoT is the term used to describe devices that can be controlled online. The Internet of Things (IoT) industry has proliferated over the past several years due to the increased need for connection and control across various gadgets and devices. Effective connectivity is the main criterion for evaluating IoT devices to enable dependable remote connection and data transmission in a wireless environment. We are urged by the Internet of Thing(IoT)principles to envision enormous, open networks of countless sentient things that can communicate with one another and their environment, share information and resources

with the cloud, and react to environmental changes. By 2025, there may be 75 billion devices connected to the Internet of Things. One of the enabling technologies for the IoT is radio frequency identification (RFID).[9] One part of the value chain that aids in bridging the gap between consumers and producers is the managing inventory system.

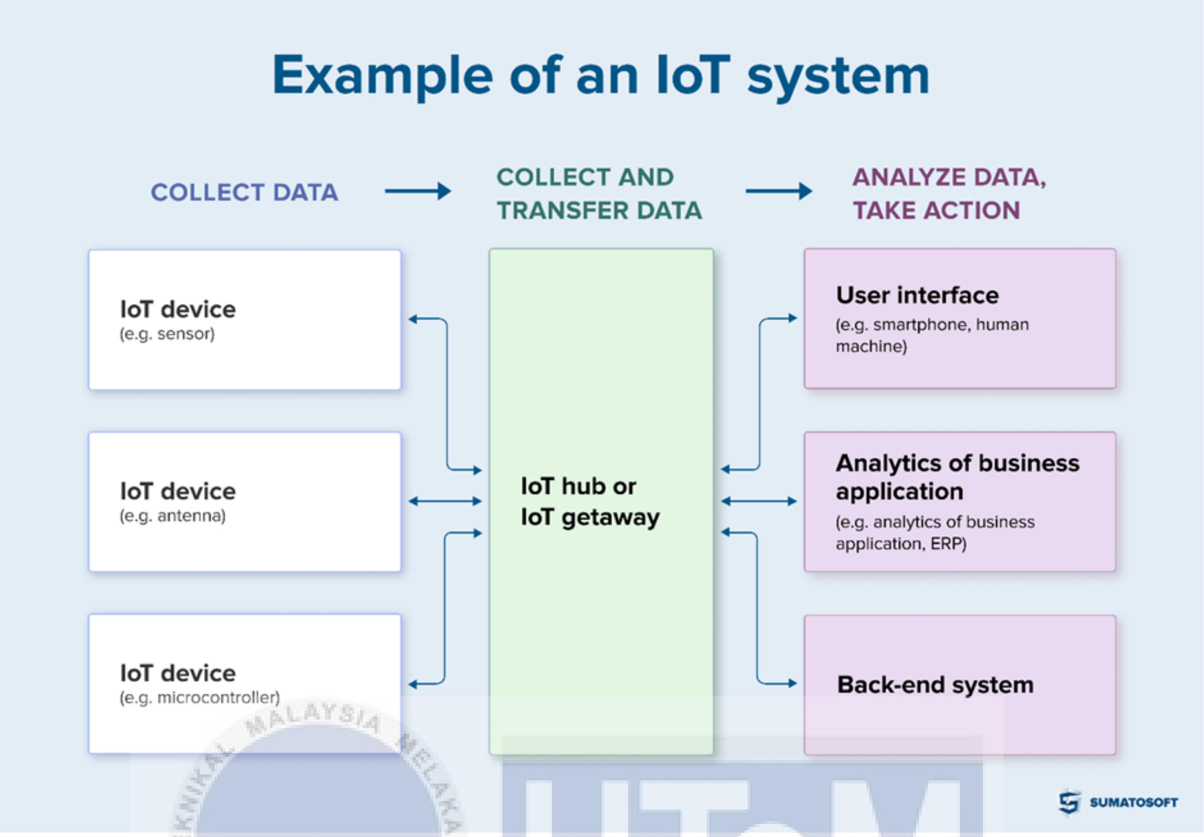

Example of an IoT system
COLLECT DATA
COLLECT AND TRANSFER DATA
ANALYZE DATA, TAKE ACTION
IoT device (e.g. sensor)
IoT device (e.g. antenna)
IoT device (e.g. microcontroller)
IoT hub or IoT getaway
User interface (e.g. smartphone, human machine)
Analytics of business application (e.g. analytics of business application, ERP)
Back-end system
SUMATOSOFT

Figure 2-10 Example of an IoT system [15]

Inventory management systems built on the Internet of Things assist in real-time item tracking. Additional barcode labels or RFID tags are added to the system to scan and recognise the products. This system, now available on the market, can keep track of several things, including inventory levels, product locations, demand projections, expiration dates, and more. The capacity to track and connect with products improves with IoT integration[14]. In this case, IoT helps organise and track the product by using a website on the mobile phone. This day, everyone uses smartphones to assist in daily life. So, the development of inventory management using IoT can help restaurant staff[16].

2.13 System helps solve the issue.

Establishing a monitoring system enables us to respond to various difficulties efficiently, should they ever materialise. One of the most essential aspects that may be checked is the economic element, which ensures that resources are distributed effectively. In addition, the system enables us to monitor the shelf life of products, which allows us to determine whether they are appropriate for continuing use or if their expiration dates are rapidly approaching. Even if items have been kept in the storage room for a considerable time, administrators and other staff members can acquire accurate information regarding the product's expiration dates if they use this system. It is now unnecessary for staff employees to rely on their memories to keep track of expiration dates because the information can be easily accessible through a web browser from any location[17]. This makes it much more convenient for customers as well. As a result of the creation of the Virtual Fridge, checking the refrigerator's contents has become substantially less time-consuming and more effective

2.14 Comparison of Previous research

Table 2-1 Table of comparison previous research

No.	Title	Problem Statement	Software/Tools	Future Work	Methodology	Advantages	Disadvantages
1	Development of FKE UiTM Kampus Pasir Gudang Lab Equipment Web Database [5]	Manual ledger books store inventory management system data. Ledgers can get lost.	Wampserver 2, Apache,PHP,MySQL	Security and enhance the website to get it more user-friendly	FKE PG Lab Equipment web database development. It involves MySQL database development, Joomla! front-page development, and webserver deployment.	No longer misplace ledger because it already stores in the database	The security needs to be developed more, and redesign the portal
2	Fridge Inventory Management to reduce Household Food Wastage [6]	Most of the food waste comes from expired or unused goods not consumed by the household.	UML, mobile application developer, database	Apply the QR Code, and add features like chat, message, and Photo.	Fridge inventory management is examined and designed. Data collecting, system analysis, and UML diagram modelling. Waterfall	The problem on expires food is lower. They will notify when the nutrition approach expiry date.	The database needs to be more secure.

					design includes comprehensive design and user interface design. A customisable system that proposes appealing food and drink uses is the objective.		
3	IoT Based Smart Inventory Management System for Kitchen Using Weight Sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi Module with Website and App. [7]	Inventory in the kitchen is not well managed	Arduino, database, embedded system, esp8266, IoT, nodemcu, Wi-Fi.	The Smart Kitchen Inventory (SKI) prototype can be mass-produced after testing. It will become more user-friendly and affordable. The future ambition is to install chips in kitchen cabinets to deliver novel kitchen services without major renovations.	IoT-based SIMS includes Smart Kitchen Inventory (SKI). Users can order groceries from the website or app and manage inventory with sensors. The system alerts users of containers about to be complete and can be accessed worldwide, making life more efficient	Inexpensive, accessible, rechargeable batteries, ECO friendly, less power consuming	Need more space to store items, heavy

					and manageable with limited time.		
4	Inventory Management System for Education Institutions [8]	Catalogues and inventory information are handwritten on paper and not in a database. Inter-departmental lending/borrowing, scrapping, and renewal are done manually or verbally, which can generate redundancy, repetitive input, and undocumented transactions.	MySQL, PHP, CSS, JS, WAMP, LDAP	Other applications and the future of this project include an IoT integration, where RFID chips can be pasted on hardware and with automatic scanning and tracking, no item/inventory will ever go amiss.	The system used an Object-Oriented Approach with UML and RAD. The database was centralised, and users had different permissions. Safe logout redirected users to the login page.	Cost-effective, simple data entry. This registry includes product definitions, serial numbers for all stock/equipment, equipment expiry dates, stock location, and packing. One can also check which products are broken, need repair, are borrowed, and are needed for the lab.	The system is relatively backward, but in the future, more attributes will be added to ensure the system's reliability in the industry.
5	Radio Frequency Identification System for Asset Tracking and Inventory	The high-value assets in hospitals are misplaced, lost or purloined, which affects	Microsoft Visual Studio, database, RFID,RAD,UML	Cloud or any database system can be included in this system to improve it.	This system manages inventory and assets via UHF RFID. ISO18000 and	The problem of tag collision is solved by reading multiple tags accurately. Overall, the	The security needs to be improved to avoid any problems with the data.

	Management in Hospitals [9]	hospital revenue and patients security.			EPC Class 1 and Gen2 standards enable reader-tag communication. RFID readers track hospital consumables, update the database, and generate bills through the user-friendly interface.	system offers a more efficient and accurate solution for inventory management and pilfering prevention in the healthcare industry.	
6	Visual Management system of Intelligent Factory tools based on Odoo [10]	Aiming at the problems of inefficient, error-prone, and complex operation of the traditional factory tool management, this paper puts forward a tool management scheme with the complete life cycle, state visualisation, and real-time dynamic management.	OCR; Odoo open source framework. (B/S) Browse/Server architecture, PostgreSQL database	3D visualisation may be applied in this system in the future to enhance the efficiency	This B/S system is web-based. Tool management, data visualisation, and OCR are its components. The tool management system has department, data, inventory, and processing components. Echarts	The intelligent industrial tool management system with visual large screen display and dynamic tool information management will standardise workshop device control. This solution improves workplace automation and	This system required the actual visual product, meaning it may be the same product but has a different serial number or effect.

					visualises and mobile OCR queries tool location. OCR recognises and laser prints the tool number.	enterprise tool management.	
7	Process Monitoring and Inventory Management in Plastic Injection Molding [11]	This research develops a software programme to solve plastic moulding injection production and stock management issues. Web pages and a database created the app. Any LAN-connected computer's web browser can display production and inventory status.	Web page, Dynamic page, Web application, Windows application, Freeware, MS Visual Web Developer, MySQL, FI-FO, GUI	The GUI needs to be more user-friendly since it is not a good interface and user easy to get confused	A table database was used for programme testing and debugging. After testing, the programme was posted to a Windows Server 2003 web application for user trials. User feedback was used to improve programme and report functionality. Since it was a web app, posting new edited files to	The users can work faster and more efficiently with fewer errors.	The problem comes when The LAN is not functioning. May use other plans to back up the data of the system.

					the server was easy.		
8	Inventory Placement Mapping using Bluetooth Low Energy Beacon Technology for Warehouses [12]	Numerous inventories are stored in a warehouse where locating goods can be troublesome for the employees. In a corrugated box company, lists of paper rolls are placed in a vast warehouse with hundreds of paper roll stock ranging from different weights and sizes.	Bluetooth low energy, Bluetooth low energy beacon, Bluetooth beacon, inventory mapping, indoor navigation, asset tracking	RFID can be used since, currently, mainly RFID is used.	Bluetooth beacons in manufacturing warehouses help inventory management by identifying and tracking items. Bluetooth beacons use RFID technology to send data quickly. The study shows warehouse Bluetooth beacon benefits with prototype figures.	Beacons in each item zone help warehouse workers zone strategically. A smartphone map and realtime position coordination can improve indoor wayfinding and workforce efficiency.	It is limited at some range since it only tracks stock in the warehouse.
9	Implementation of RFID-based Lab Inventory System [13]	Items are stolen or taken out of the lab without being entered in the inventory register, costing the institution and the student. Lost inventory registers	IoT, ESP32, MCR522, HTML, CSS, MongoDB	the project can be taken forward by using long-range tags, which have a more extensive reading radius.	Due to connectivity difficulties, ESP32 replaced ESP8266. RFID card UIDs identify it. ESP32 IP packets deliver	The lab assistant can also regularly update if the hardware faces any issues. Since there are multiple quantities of a	It was limited only to the radius of the room. Longer than that, it will be an error.

		destroy all records. It doesn't secure inventory.			card data to a database and communicate with the server. Lab assistants read RFID tags to provide students with the components they request online. Journal databases track component usage.	single instrument. The assistant can also state on the website which device is usable and which equipment is facing problems.	
10	IOT IN INVENTORY MANAGEMENT [14]	Inventory management lets customers and manufacturers connect. Real-time inventory management solutions use IoT. The technology scans barcode labels or RFID tags on the products.	RFID, IoT, IoT Cloud, ERP	We need to follow the era, which means from time to time to improve in terms of security need to be updated.	IoT and real-time analytics optimise store inventory. RFID chips and scanners provide reliable tracking. MRP dictates materials. RFID systems read tags using radio waves. IoT inventory management gives actionable insights.	Inventory visibility streamlines restocking. Many firms are using IoT devices to manage inventory cost-effectively. Real-time data from IoT devices can help companies enhance logistics, inventory, and	Must consider the cost of integration, security, and scalability of IoT devices into their existing inventory management system. Despite these reservations, the cost of IoT technology

						supply chain management.	continues to fall as the number of accessible IoT integrations for inventory management grows.

.

2.15 Summary

In summary, the stock management system in the food and beverage sector plays a vital role in efficiently managing storage and tracking stock quantities. Issues such as misplaced data, lack of FIFO practices, and failure to inform customers of low stock can result in losses and dissatisfaction. They include the development of lab equipment databases, mobile applications for fridge inventory management, smart inventory management systems for kitchens, software-based inventory management systems for educational institutions, RFID-based systems for hospital asset tracking, intelligent tool management systems for factories, web applications for process monitoring and stock management, the use of BLE beacons for warehouse inventory management, RFID-based systems for university labs, and discussions on inventory management strategies and technologies. This research emphasise the importance of efficient inventory control and resource utilisation in various industries. A dedicated inventory system has been designed to address these challenges to ensure accurate and accessible information. The system securely stores data on a database server accessible only to authorised personnel. Implementing this system improves operational efficiency, minimises errors, and enhances customer satisfaction by enabling effective stock monitoring and informed decision-making. The goal is to optimise inventory control and mitigate stock-related issues for better future management.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, explain the procedure and method for the system. It includes the flowchart and Gantt Chart of PSM, Flowcharts of PSM and the system, the Entity Relationship

In this chapter, explain the procedure and method for the system. It includes the flowchart and Gantt Chart of PSM, Flowcharts of PSM and the system, the Entity Relationship Diagram (ERD), the software used method used in the system. The main goal is to give the user a clear vision of how to use the Stock Management System, which needs to brief the understanding of how to use the system step by step. Furthermore, the method applied in this system includes some knowledge of programming.

3.2 Flowchart of PSM

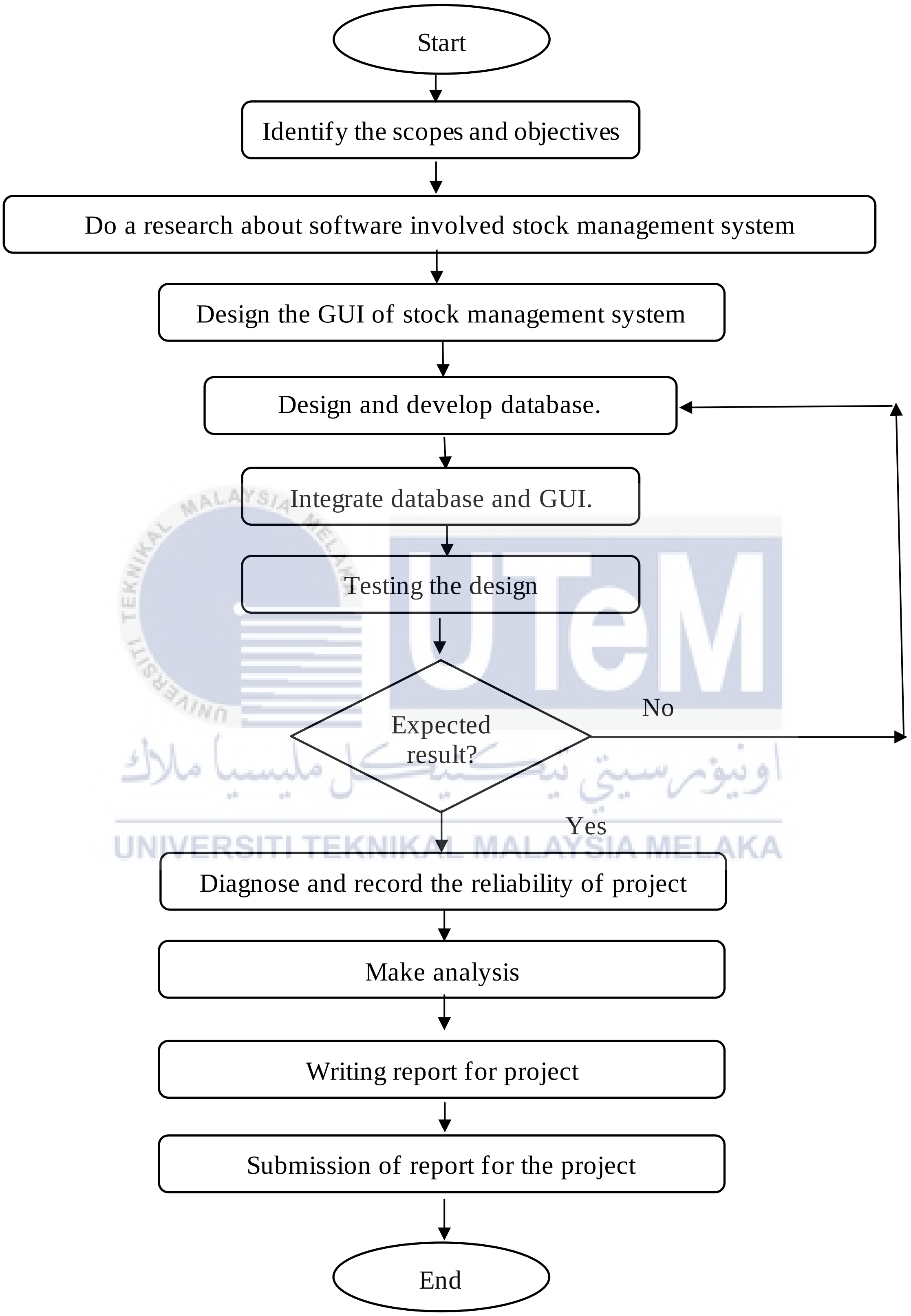
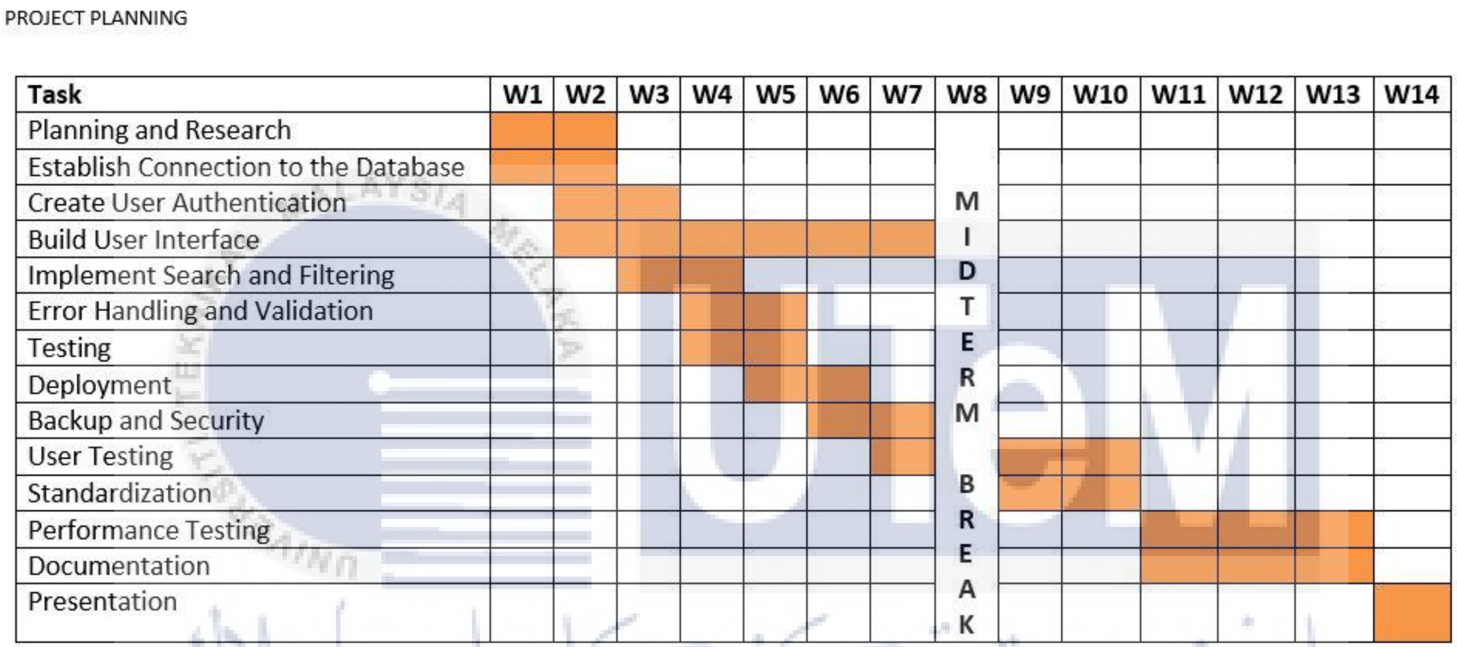

Start
Identify the scopes and objectives
Do a research about software involved stock management system
Design the GUI of stock management system
Design and develop database.
Integrate database and GUI.
Testing the design
Expected result?
No
Yes
Diagnose and record the reliability of project
Make analysis
Writing report for project
Submission of report for the project
End

Figure 3-1 Flowchart of PSM

The PSM starts by identifying the scopes and objectives—research about software needed to validate the system's suitable software. The GUI needs to be designed but using a storyboard adequate. Next, design and develop the database based on the system. After that, we need to test the design and compare the expected results, to see whether they are likely. If not, repeat process design and development to identify the problem or debugging process happen. The reliability of the project needs to be recorded and diagnosed. Analysis made. All components are recorded in the report and submitted as a report. That is the flow for PSM 1.

3.2.1 Gantt Chart



PROJECT PLANNING

Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Planning and Research	🟧	🟧						MIDTERM BREAK						
Establish Connection to the Database	🟧	🟧												
Create User Authentication		🟧	🟧											
Build User Interface		🟧	🟧	🟧	🟧	🟧	🟧							
Implement Search and Filtering			🟧	🟧										
Error Handling and Validation				🟧	🟧									
Testing				🟧	🟧									
Deployment					🟧	🟧								
Backup and Security						🟧	🟧							
User Testing							🟧		🟧	🟧				
Standardization									🟧	🟧				
Performance Testing											🟧	🟧	🟧	
Documentation											🟧	🟧	🟧	
Presentation														🟧

Figure 3-2 Gantt Chart of PSM

The comprehensive 14-week project plan outlines a systematic approach to developing a stock management system. The initial weeks focus on meticulous planning, including defining project scope, researching existing systems, and creating a detailed project plan with timelines, milestones, and risk management strategies. The subsequent weeks delve into technical implementation, starting with establishing a connection to the database and progressing through user authentication, interface design, and search/filter functionalities. Robust error handling and validation mechanisms are introduced, ensuring the application's reliability and security. Thorough testing, both manual and automated, is conducted in Week 7, followed by the deployment phase, which involves setting up production environments and optimizing server configurations. Weeks 9 and 10 address crucial aspects of backup, security, and user testing, incorporating measures for data protection and continuous improvement based on user feedback. The project progresses into standardization, embracing coding standards, version control, and comprehensive documentation. The final weeks focus on performance testing, ensuring the system's scalability under varying loads. The documentation phase, in Week 13, includes user guides, developer documentation, and system diagrams. The project concludes with a presentation to stakeholders, showcasing the application's features, benefits, and future plans. The mid-term break serves as an opportunity for catching up on any outstanding tasks and recharging, maintaining a balance between productivity and well-being throughout the project's lifecycle.

3.3 Methodology

The systematic methodology and collection of techniques used to successfully design, develop, and execute a system are discussed in this chapter. The entire process, from analysing requirements to testing and maintenance, is guided by a systematic framework. The technique ensures the system is developed and managed methodically, effectively, and trustworthy. Reaching the desired goals and outcomes includes several phases, including needs assessment, analysis, design, implementation, and evaluation. Organisations may create and manage reliable systems that satisfy user requirements, are scalable, and offer top performance by adhering to a well-defined process.

3.3.1 Flowchart of the system

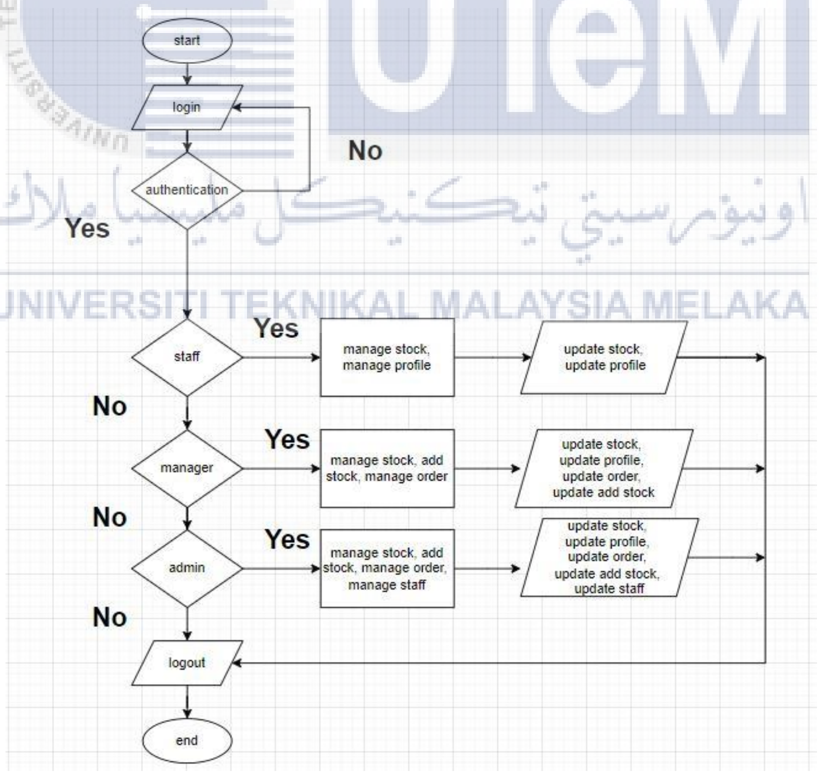

start
login
No
authentication
Yes
staff
Yes
manage stock, manage profile
update stock, update profile
No
manager
Yes
manage stock, add stock, manage order
update stock, update profile, update order, update add stock
No
admin
Yes
manage stock, add stock, manage order, manage staff
update stock, update profile, update order, update add stock, update staff
No
logout
end

Figure 3-3 Flowchart of Stock Management System

Based on Figure 3-3, the user must first log in as a staff member. Then authentication will identify whether the user is staff, manager, or admin. Next, they will execute the next step by their role group as staff can update stock management, showing a report that the team already update the stock. Managers can monitor, edit, and manage the product, which means managers can add stock orders. Hence, the system will generate a report about the activity then the manager can also do the inventory check to monitor whether the product's expiring date is approaching. As an admin, they can monitor the progress, update the stock based on the manager's orders, and verify whether the product can be ordered. Admin can also give access to manager or staff to continue the process. If not any of the choices are chosen, the system will log out, and the system ends.

3.3.2 Use Case Diagram

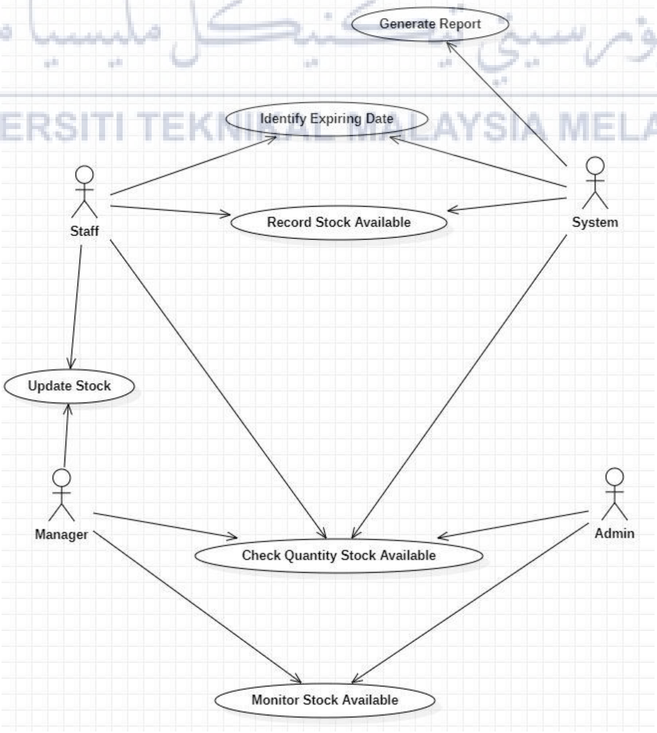


Figure 3-4 Use Case diagram of the system

The figure above involved four actors: Staff, Manager, Admin, and System. The use case in the system is to generate reports, identify expiring dates, record stock available, check quantity stock available, and monitor stock available. The system can create reports. Staff and system can locate the expiring date and record stock available. Staff and managers can update stock. Every actor can do the check quantity of stock available. Managers and Admin were able to monitor stock available.

3.3.3 Entity Relatioship Digram(ERD)

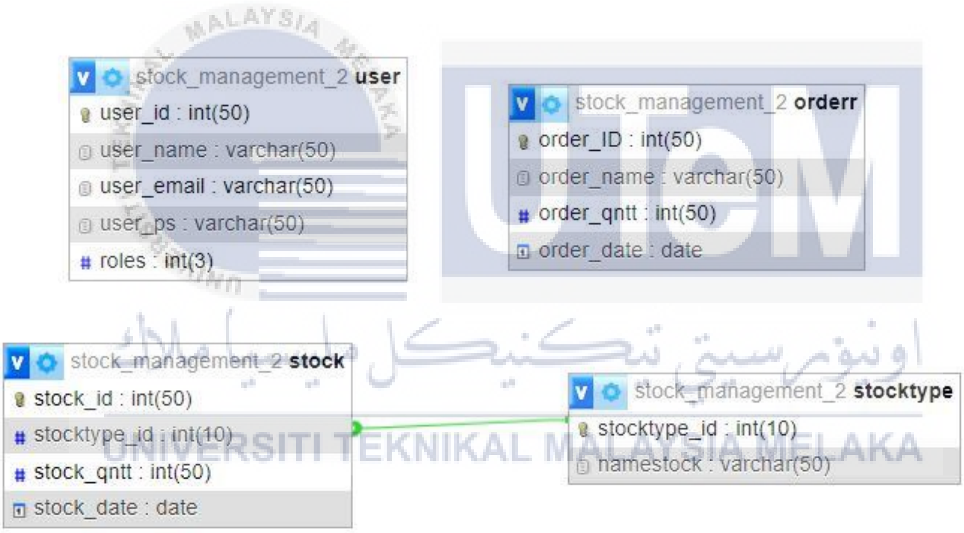


Figure 3-5 Figure of System ERD

The Entity Relationship Diagram (ERD) delineates the configuration and associations within a hypothetical stock management system, encompassing four fundamental entities: user, stock, orderr (order), and stocktype. The 'user' entity includes attributes such as user_id, user_name, user_email, user_ps (password), and roles, which represent the specific information about individuals who are engaging with the system. The 'stock' entity consists of stock_id, stocktype_id, stock_qntt (quantity), and stock_date, which provide details about the stock items being managed in the system. The 'orderr' entity comprises attributes such as order_ID, order_name, and order_date, which capture specific information pertaining to orders made within the system. The 'stocktype' entity serves as a reference table, containing stocktype_id and namestock, which classify various types of stock items.

These entities are interconnected through explicit relationships. Users can be linked to multiple stock items and orders, which reflects the dynamic role of users in overseeing the system. Every individual stock item is associated with a distinct stocktype, enabling effective categorization and organisation. On the other hand, a stocktype can be linked to multiple stock items. Regarding orders, an order can encompass multiple stock items, with each stock item being linked to a single order.

This Entity-Relationship Diagram (ERD) represents a simple stock management system that allows users to monitor stock items and make orders. It includes a stocktype lookup table to categorise different types of stock. The diagram establishes relationships that enable a thorough comprehension of the organisation and connectivity of data within the system.

3.3.4 Purpose of GUI

Table 3-1 Table of Purpose GUI

Interface	Function
Login	For the user to access the system using their ID and password. It also provides a clickable button.
Admin	Admin can monitor the current progress of the manager and staff. Other than that, they can delete IDs for managers and staff since they are on top of the hierarchy.
Manager	The order can be made by a manager. Staff roles are also within the manager role.
Staff	The staff's job is to update the stock they take out daily. The date need to be written during the update of the inventory.

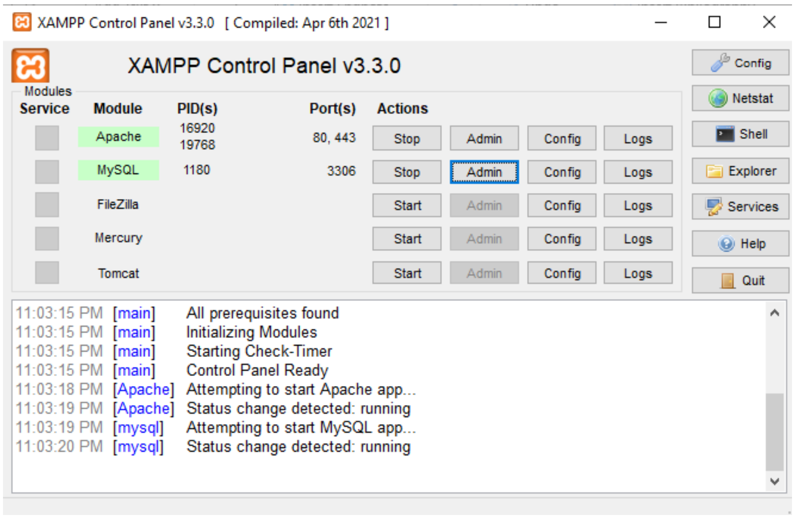
The purpose of the GUI goal is to provide the user with decision points that are simple to locate, understand, and apply. Easy to say as a guide for developers to know the function of each interface, which can be easily used for future work.

3.4 Software Used

The system consists of one part, which uses software components only and needs to get knowledge about them. Let's get through one by one.

3.4.1 Xampp

The XAMPP software package is free and open-source software that streamlines the process of developing websites on a local computer. It provides a complete environment for testing and developing web applications by including Apache, MySQL, PHP, and Perl as part of its functionality[18]. XAMPP is often used for local development before it is deployed to a live server since it has simple controls and extra utilities such as phpMyAdmin. XAMPP, on the other hand, should not be used in environments intended for production because it is not encouraged.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	16920 19768	80, 443	Stop	Admin	Config	Logs
	MySQL	1180	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
11:03:15 PM  [main]     All prerequisites found
11:03:15 PM  [main]     Initializing Modules
11:03:15 PM  [main]     Starting Check-Timer
11:03:15 PM  [main]     Control Panel Ready
11:03:18 PM  [Apache]   Attempting to start Apache app...
11:03:19 PM  [Apache]   Status change detected: running
11:03:19 PM  [mysql]    Attempting to start MySQL app...
11:03:20 PM  [mysql]    Status change detected: running
```

Figure 3-6 Xampp Control Panel version 3.3.0

3.4.2 phpMyAdmin

One of the most well-known web-based management tools for MySQL databases is called phpMyAdmin. It has a straightforward graphical user interface that makes it easy for users to execute a wide range of database activities, including creating, modifying, and removing databases, tables, and records[19]. phpMyAdmin is a program that helps simplify the process of maintaining databases by providing capabilities such as importing and exporting data, executing SQL queries, controlling user privileges, and generating visual representations of the database's structure. It enables developers and database administrators to interact with MySQL databases effectively and carry out routine actions in the context of database administration. MySQL has been chosen as a database server because it is one of the top ranks voted by professional developers.[20]

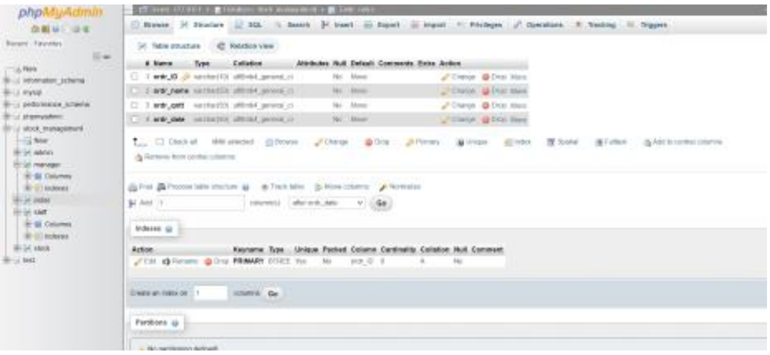


Figure 3-7 phpMyAdmin example

3.4.3 PHP

PHP, "Hypertext Preprocessor," is a widely used open-source server-side scripting language[21]. Its name comes from the acronym PHP. Because of its tight integration with HTML, programmers can incorporate PHP code directly into HTML markup, resulting in dynamic web page development. For web development, PHP offers a wide range of capabilities, such as built-in support for form management, file uploads, sessions, cookies, and more. It also features good database integration, allowing it to function effectively with standard database management systems such as MySQL. Because it is not dependent on any particular platform to work, PHP can be deployed in various contexts thanks to its portability. PHP continues to be a well-liked option among software developers who want to create dynamic and feature-packed online applications because it has a sizable and lively community, abundant documentation, and constant advances in performance and scalability.

3.4.4 HTML

Markup languages come in various flavors, but the most used one for developing websites and software is HTML (Hypertext Markup Language). It specifies the organisation and layout of a website's content, which may include text, photos, links, and other multimedia components. HTML uses tags, or labels, to specify various elements and their associated characteristics. This enables browsers to comprehend and display the content correctly. The majority of websites are written in HTML, which is a markup language. Pages can be created and given functionality through the use of HTML[22]. Web page designers can arrange web pages using HTML components such as headings, paragraphs, lists, tables, etc. Using semantic components in a website's structure, such as the header>, nav>, section>, and footer> tags, lends the material additional meaning and context, thereby improving its search engine optimisation and accessibility. The construction of hyperlinks, which are used to connect web pages and browse through websites, is enabled by HTML. Forms may be constructed quickly and easily using form elements such as "input" and "select," allowing user data entry and submission. Using HTML tags, multimedia information such as photographs, movies, and audio files can be included without interruptions. HTML is designed to be as accessible as possible by having features such as alternative text for picture files and properties for use with screen readers. It makes the content of websites accessible to users with various kinds of impairments. The fact that all major web browsers support HTML contributes to its compatibility, which is another advantage of using HTML. However, developers are responsible for ensuring that their code is compatible with various browsers when using particular HTML capabilities. In a nutshell, HyperText Markup Language (HTML) is the bedrock upon which web development is built. It is responsible for the structure, layout, and organisation of material. It makes it possible to create web pages and applications that are interactive and available across multiple platforms, which in tum improves the user experience.

3.4.5 CSS

Cascading Style Sheets, or CSS for short, is essential to developing websites that focus on the visual display and arrangement of items created with HTML. It functions with HTML to specify styles for improving web page layout and overall look. These styles include colours, fonts, spacing, and positioning[23]. Developers can target particular items and apply their desired styles by using selectors and declarations in their code. CSS offers many capabilities, including layout and placement, font modification, colour and background control, transitions and animations, and responsive design. It makes it possible to separate the style from the content, resulting in a uniform and aesthetically pleasing layouts across all web pages. CSS is essential in developing interactive and aesthetically pleasant user experiences on the web.

3.4.6 Apache Bench

ApacheBench (often referred to as ab) stands as a command-line utility within the Apache HTTP Server project, offering a straightforward means to benchmark the performance of web servers. Typically bundled with Apache installations, this tool enables users to assess a server's capabilities by simulating various levels of concurrent requests. The fundamental usage involves specifying the total number of requests (-n) and the concurrency level (-c). For instance, a command like ab -n 100 -c 10 http://example.com/ would dispatch 100 requests to the specified URL with a concurrency of 10 at a time. Output metrics include requests per second and connection times, aiding in gauging server performance under different loads. While ApacheBench provides valuable insights into a server's responsiveness, it's essential to consider that real-world performance can be influenced by factors beyond simulated loads, such as network conditions and application complexity. ApacheBench (ab), a command-line tool primarily associated with the Apache HTTP Server, is often considered more seamlessly integrated with Linux systems. This preference arises from its native inclusion with the Apache server commonly used on Linux. Leveraging the powerful command-line interface of Linux, users can easily incorporate ApacheBench into scripts, automation processes, and perform efficient performance testing. The historical association of ApacheBench with Apache HTTP Server, predominantly deployed on Linux servers, contributes to its strong presence and support in the Linux ecosystem. Aligning with the open-source philosophy, both ApacheBench and Apache HTTP Server are well-suited for Linux environments, ensuring compatibility, straightforward installation, and active community support. Linux's package management systems further simplify the handling of dependencies, enhancing the ease of setup and utilization of ApacheBench. Additionally, the scripting capabilities of Linux allow users to seamlessly integrate ApacheBench into scripts for automated and large-scale testing scenarios. Although ApacheBench is not exclusive to Linux, its historical ties and compatibility make it particularly well-suited for performance testing in Linux environments.

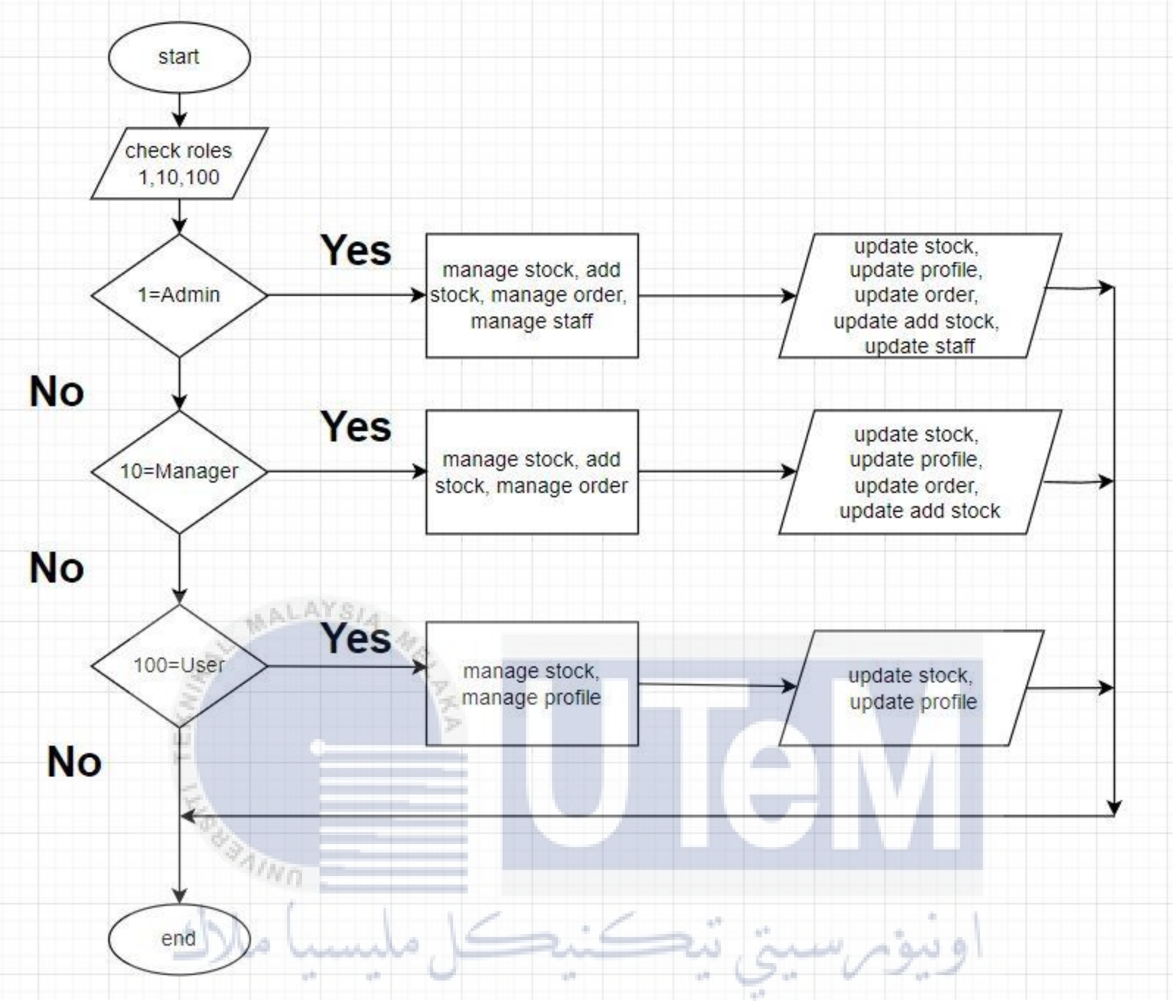
3.4.7 SQL Injection

SQL injection represents a significant web security vulnerability, enabling malicious actors to disrupt an application's database queries. Exploitation of this vulnerability empowers attackers to access, manipulate, or delete data beyond their authorized scope, potentially compromising the integrity of the underlying server or other backend infrastructure. This type of cyber threat has played a pivotal role in numerous high-profile data breaches, leading to unauthorized retrieval of sensitive information like personal details, financial records, and confidential business data.

The susceptibility to SQL injection arises when an application forms SQL queries by concatenating user inputs. Exploiting this vulnerability, attackers inject malevolent SQL code into the input, subsequently executed by the database. This illicit maneuver grants the attacker the ability to inspect or modify data within the database and, in extreme cases, execute system commands on the server infrastructure.

Preventing SQL injection attacks necessitates the adoption of security measures such as prepared statements or parameterized queries. These methods help ensure the proper escaping and quoting of user input, thereby mitigating the risk of malicious code execution. Additionally, validating and sanitizing user input is crucial to preclude any inclusion of harmful code. Keeping software consistently updated and conducting regular security testing are vital practices to identify and rectify potential vulnerabilities in a timely manner.

3.5 Method of System



start
check roles 1,10,100
1=Admin
Yes
manage stock, add stock, manage order, manage staff
update stock, update profile, update order, update add stock, update staff
No
10=Manager
Yes
manage stock, add stock, manage order
update stock, update profile, update order, update add stock
No
100=User
Yes
manage stock, manage profile
update stock, update profile
No
end

Figure 3-8 Figure of Check Roles Flowchart

3.5.1 Case users

Initially, it performs a user permission check, ensuring that only users with ADMIN privileges can access the user management section; otherwise, it gracefully presents an error message. The code encompasses form submission handling for three distinct forms: updateDetails for modifying user information, deleteUser for removing a user from the database, and addUser for adding a new user. The View Users section provides a tabular presentation of existing users, featuring essential details such as user ID, name, email, and role, accompanied by options for user editing and deletion. The Edit Users section caters to three scenarios: editing an existing user, adding a new user, or deleting a user. Leveraging MySQLi for database interactions and incorporating HTML, CSS, and JavaScript for the user interface, this code snippet efficiently manages user-related functionalities. However, it is imperative to address certain shortcomings, notably the absence of input validation and sanitization, which could potentially expose the application to security risks such as SQL injection and cross-site scripting (XSS). Rectifying these issues is crucial to fortify the application's security and overall robustness.

3.5.2 Case profile

Triggered when users opt to modify their information, it redirects them to a form enabling updates to their email and password. Upon form submission, the code meticulously checks if both the email and password fields are completed. If so, it adeptly updates both entries in the database. Alternatively, if only the email field is populated, it exclusively updates the email information. Subsequently, the code fetches the user's existing details from the database, encompassing their name, email, and position, presenting them in a well-formatted manner. The position is ascertained by interpreting the 'roles' field in the database, mapping various numeric values to distinct roles (1 for Admin, 10 for Manager, and 100 for Staff). In cases where the 'roles' value deviates from these parameters, the position defaults to 'Unknown.' Additionally, the code includes a commented-out section designed for displaying user details when a link to view their information is clicked. This code not only furnishes users with an intuitive interface for profile updates but also prioritizes secure storage of data in the database.

3.5.3 Case order

This PHP code facilitates the execution of CRUD operations (Create, Read, Update, Delete) to manage "orders" in a database. The process begins by validating the user's authorization to access the order page. When given permission, the page displays a structured representation of current orders, along with choices to include, modify, or remove orders. The code presents a multifunctional editOrder function that accepts parameters including order ID, operation type (add, edit, or delete), database connection object, and script name. This function generates the necessary HTML form for adding, editing, or deleting an order, depending on the specified operation. Subsequently, the viewOrder function is implemented, which is tasked with fetching all orders from the database and displaying them in a well-organized table format. Every row in the table contains crucial information such as order ID, name, quantity, and date. Additionally, there are edit and delete buttons provided to make it easier to modify or remove orders. The code subsequently examines whether the user accessed the page using a particular operation (add, edit, or delete) and a corresponding order ID. If the answer is yes, the corresponding editOrder function is called to display the relevant form. Finally, the variable $isi is assigned the HTML code for the order page, which includes the table of existing orders and the invocation of the viewOrder function. If there are not enough permissions, a message that is easy for the user to understand will be shown instead. This code provides users with efficient order management capabilities and enforces access control by verifying permissions prior to allowing interactions with the database.

3.5.4 Case stock

Admins, managers, and regular users gain access to functionalities such as adding, editing, deleting, and viewing stock details. Leveraging mysqli_query for seamless interaction with a MySQL database, the code meticulously checks for errors during the querying process. Users can augment stock levels by specifying type, quantity, and date, while the edit functionality facilitates updates to quantity and date for specific stock items. The delete feature enables the removal of stock items from the system, and the view functionality presents stock information in a structured table format. Additionally, the code incorporates a valuable low-stock alert, dynamically highlighting items with a quantity of 10 or less in red. Serving as a vital component within a broader stock and inventory management system, this code ensures effective control and monitoring of stock levels while prioritizing user-friendly interactions.

3.5.5 Case addStock

This PHP code snippet efficiently manages the addition, editing, and deletion of stock types in a system, with access restricted to users possessing manager-level permissions or higher. The implementation relies on SQL queries for seamless interaction with the database, allowing users to retrieve and modify stock type data through a carefully crafted HTML and CSS-based user interface. The pivotal functions, `ejasStok` and `tengokStok`, dynamically generate HTML code for forms and tables. The former adapts its display based on the operation parameter, enabling users to add or edit stock types, while the latter generates an HTML table displaying existing stock types retrieved from the database. To ensure secure access, the code checks user permissions using the `$auth->getUserPerm()` method and authorizes access only to those with the required managerial permissions, displaying an error message otherwise. The code also handles form submissions, updating the database with new or edited stock type data. Additionally, it manages deletion requests, presenting a confirmation message and proceeding with the removal of the stock type from the database upon user confirmation. Overall, this well-organized and permission-aware code snippet serves as a fundamental component for effective stock type management within a broader system.

3.5.6 Session time out

Utilising the jQuery library, is specifically designed to manage session expiration in a web application. The `countdown` function is used to initialise a timer based on the `PAGESESSIONTIME` PHP constant. It displays the remaining session time in minutes and seconds. Once the expiration time is reached, the `sessionTimeOut` function is called, which triggers an alert to inform users that their session has ended, and prompts them to log in again. The purpose of this script is to improve user experience by delivering prompt notifications for session management in the application.

3.6 Summary

The methodology suggested for developing a new, efficient system for the food and beverage industry is presented in this chapter. The correct approach must be disseminated to everyone for every already-defined program so the restaurant can remain profitable. Errors caused by humans will be eliminated from the data. Toward the end of this methodology, the evaluation process that should be used for all software is described.

CHAPTER 4
RESULTS AND DISCUSSIONS

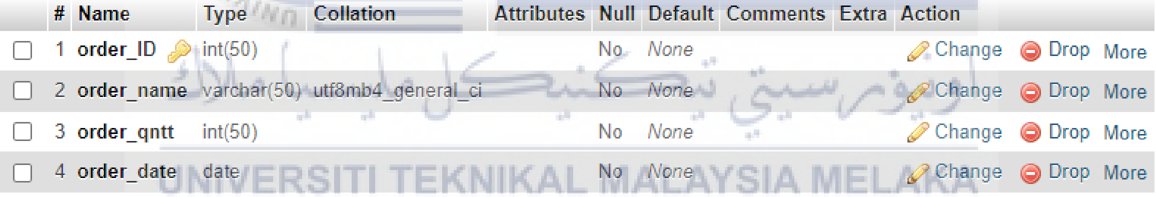
4.1 Introduction

The outcomes of the system's implementation and an analysis of its execution are shown in the part titled "Results and Discussion" of the project "Implementing a Stock Management System Using MySQL, phpMyAdmin, HTML, and CSS." The purpose of the stock management system is to track and manage inventory effectively, guaranteeing that stock levels are accurate and operations streamlined. The results gathered were reported in this area, including information on stock tracking and revisions to the inventory. The system's functionality and performance are evaluated, considering aspects such as how quickly it responds, how reliable it is, and how user-friendly it is. The discussion digs deeper into the ramifications of the results, analysing how the system has improved stock management operations, decreased error rates, and enhanced overall efficiency. In addition, it discusses any difficulties or restrictions that arose during the implementation process and offers alternative solutions and areas that could use further refinement. In addition, the user interface is built with HTML and CSS described, and an analysis of its usability, responsiveness, and aesthetics is performed based on the comments and suggestions of actual end users. In part, under "Results and Discussion," you will find a detailed assessment of the stock management system's performance, effectiveness, and user satisfaction. This section will also offer ideas for potential improvements and optimisations in the future.

4.2 Results and Analysis

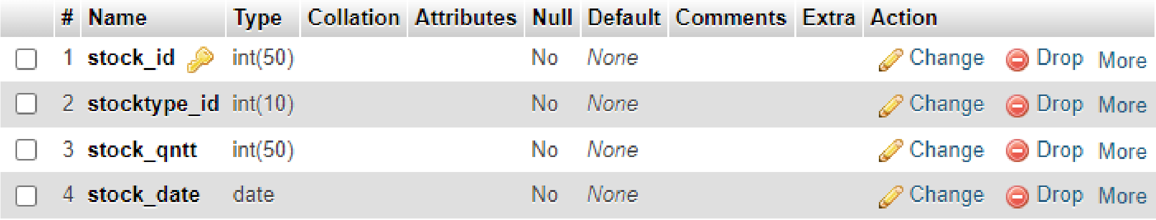
In this section, a comprehensive presentation of results before and after a particular intervention will be provided, followed by a thorough analysis. The purpose is to illustrate the discernible changes and evaluate the impact of the implemented measures or changes. This comparative approach allows for a clear understanding of the transformation and facilitates insights into the effectiveness of the undertaken actions. The analysis will delve into key metrics, outcomes, and any observable shifts in order to draw informed conclusions about the success or areas that may require further attention. This structured examination aims to offer a nuanced perspective on the progression from the initial state to the modified condition, contributing valuable insights for decision-making and future planning.

4.2.1 Preliminary Result and Analysis



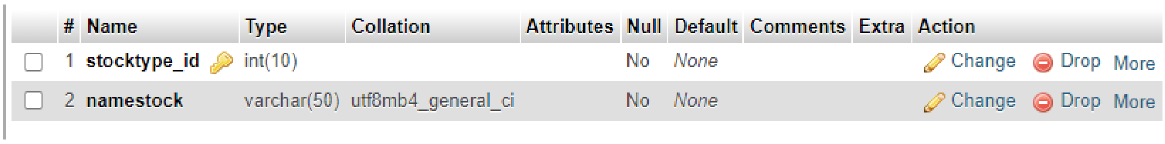
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	order_ID	int(50)			No	None			Change	Drop	More
□	2	order_name	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
□	3	order_qntt	int(50)			No	None			Change	Drop	More
□	4	order_date	date			No	None			Change	Drop	More

Figure 4-1 Table Structure of order



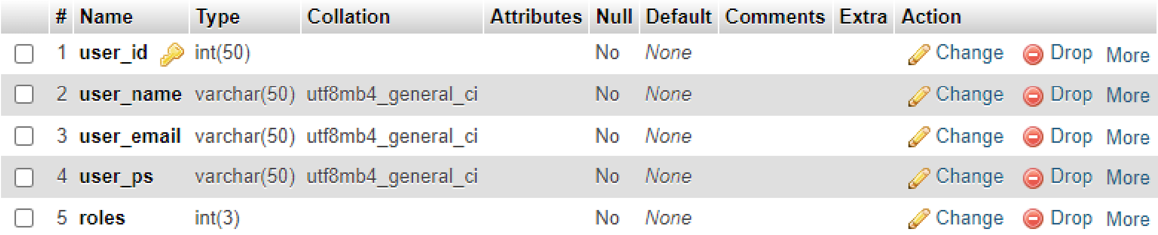
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	stock_id	int(50)			No	None			Change	Drop	More
□	2	stocktype_id	int(10)			No	None			Change	Drop	More
□	3	stock_qntt	int(50)			No	None			Change	Drop	More
□	4	stock_date	date			No	None			Change	Drop	More

Figure 4-2 Table Structure of Stock



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	stocktype_id	int(10)			No	None			Change	Drop	More
□	2	namestock	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More

Figure 4-3 Table Structure of stocktype



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	user_id 🔑	int(50)			No	*None*			Change	Drop	More
□	2	user_name	varchar(50)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	3	user_email	varchar(50)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	4	user_ps	varchar(50)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	5	roles	int(3)			No	*None*			Change	Drop	More

Figure 4-4 Table Structure of user

The table shows the database for order and all names based on information needed in the database. The primary keys for all databases are already set. The table is set for order, stock, stocktype, and user. All of this will be integrated to HTML and PHP later.

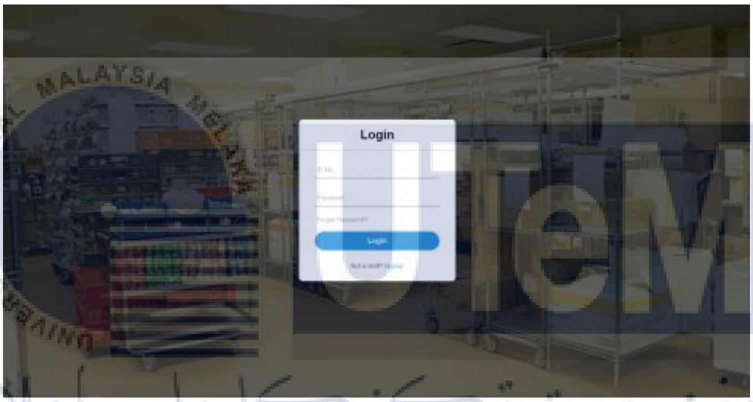
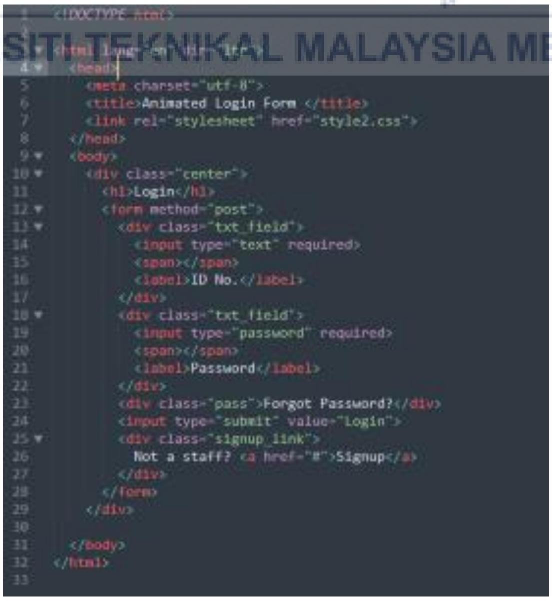
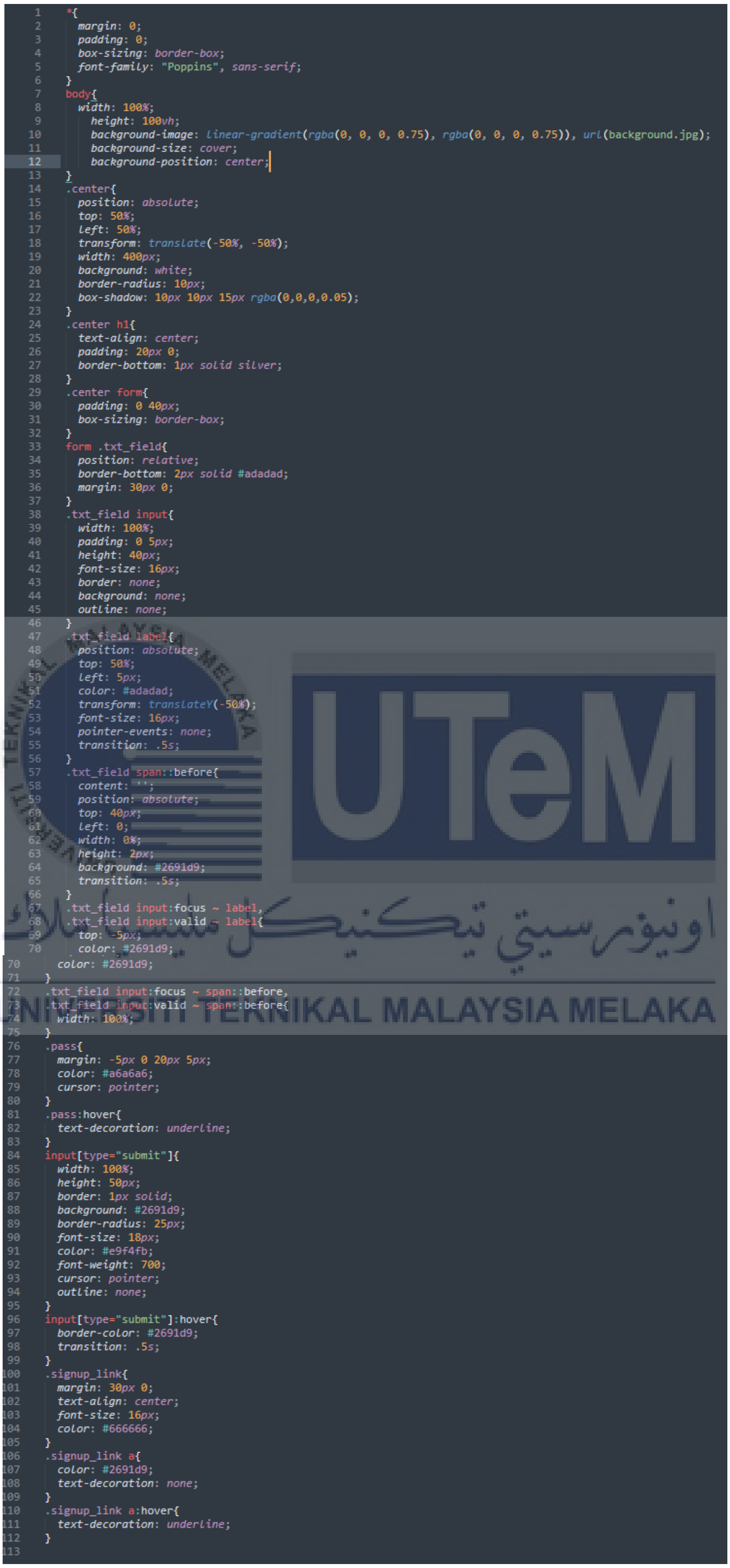


Figure 4-5 Figure of Login Form



```
<!DOCTYPE html>

<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Animated Login Form </title>
    <link rel="stylesheet" href="style2.css">
  </head>
  <body>
    <div class="center">
      <h1>Login</h1>
      <form method="post">
        <div class="txt_field">
          <input type="text" required>
          <span></span>
          <label>ID No.</label>
        </div>
        <div class="txt_field">
          <input type="password" required>
          <span></span>
          <label>Password</label>
        </div>
        <div class="pass">Forgot Password?</div>
        <input type="submit" value="Login">
        <div class="signup_link">
          Not a staff? <a href="#">Signup</a>
        </div>
      </form>
    </div>

  </body>
</html>

```

Figure 4-6 Figure of HTML coding of Login Form



```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
body{
  width: 100%;
    height: 100vh;
    background-image: linear-gradient(rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0.75)), url(background.jpg);
    background-size: cover;
    background-position: center;
}
.center{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 400px;
  background: white;
  border-radius: 10px;
  box-shadow: 10px 10px 15px rgba(0,0,0,0.05);
}
.center h1{
  text-align: center;
  padding: 20px 0;
  border-bottom: 1px solid silver;
}
.center form{
  padding: 0 40px;
  box-sizing: border-box;
}
form .txt_field{
  position: relative;
  border-bottom: 2px solid #adadad;
  margin: 30px 0;
}
.txt_field input{
  width: 100%;
  padding: 0 5px;
  height: 40px;
  font-size: 16px;
  border: none;
  background: none;
  outline: none;
}
.txt_field label{
  position: absolute;
  top: 50%;
  left: 5px;
  color: #adadad;
  transform: translateY(-50%);
  font-size: 16px;
  pointer-events: none;
  transition: .5s;
}
.txt_field span::before{
  content: '';
  position: absolute;
  top: 40px;
  left: 0;
  width: 0%;
  height: 2px;
  background: #2691d9;
  transition: .5s;
}
.txt_field input:focus ~ label,
.txt_field input:valid ~ label{
  top: -5px;
  color: #2691d9;
  color: #2691d9;
}
.txt_field input:focus ~ span::before,
.txt_field input:valid ~ span::before{
  width: 100%;
}
.pass{
  margin: -5px 0 20px 5px;
  color: #a6a6a6;
  cursor: pointer;
}
.pass:hover{
  text-decoration: underline;
}
input[type="submit"]{
  width: 100%;
  height: 50px;
  border: 1px solid;
  background: #2691d9;
  border-radius: 25px;
  font-size: 18px;
  color: #e9f4fb;
  font-weight: 700;
  cursor: pointer;
  outline: none;
}
input[type="submit"]:hover{
  border-color: #2691d9;
  transition: .5s;
}
.signup_link{
  margin: 30px 0;
  text-align: center;
  font-size: 16px;
  color: #666666;
}
.signup_link a{
  color: #2691d9;
  text-decoration: none;
}
.signup_link a:hover{
  text-decoration: underline;
}
```

Figure 4-7 Figure of CSS coding of HTML

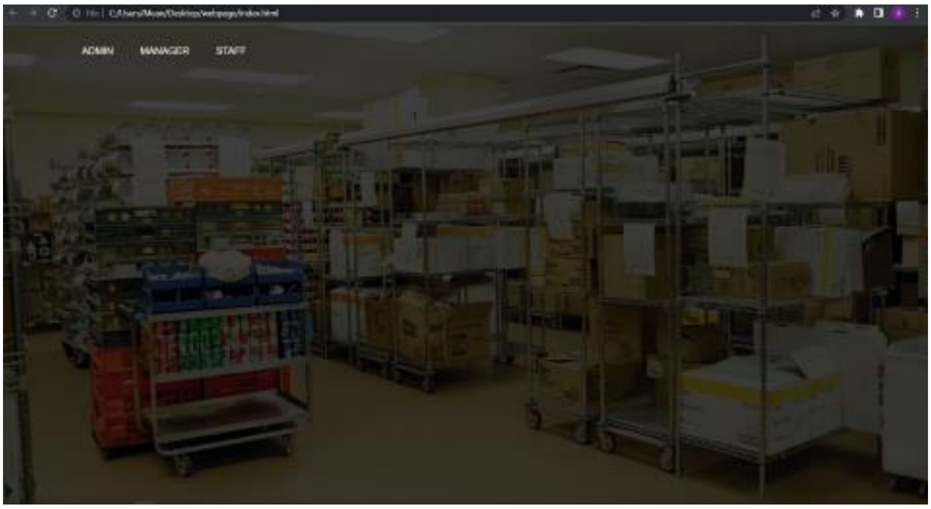
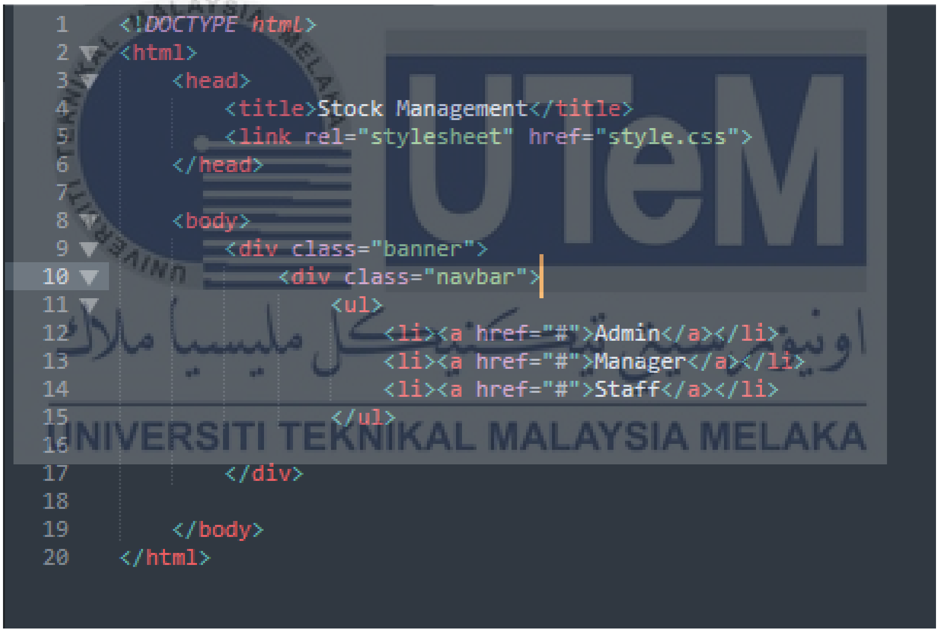
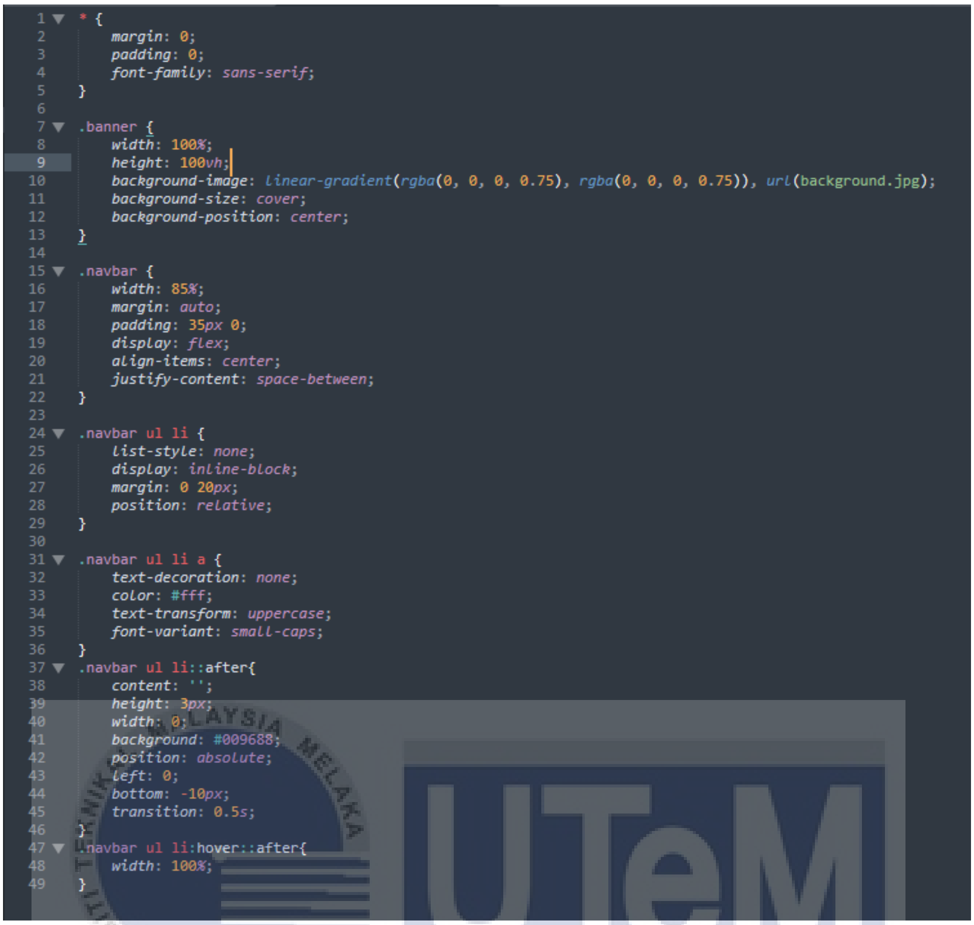


Figure 4-8 Figure of Dashboard



```
<!DOCTYPE html>
<html>
    <head>
        <title>Stock Management</title>
        <link rel="stylesheet" href="style.css">
    </head>

    <body>
        <div class="banner">
            <div class="navbar">
                <ul>
                    <li><a href="#">Admin</a></li>
                    <li><a href="#">Manager</a></li>
                    <li><a href="#">Staff</a></li>
                </ul>

        </div>

    </body>
</html>
```

Figure 4-9 Figure of HTML Coding of Dashboard



```css
* {
    margin: 0;
    padding: 0;
    font-family: sans-serif;
}

.banner {
    width: 100%;
    height: 100vh;
    background-image: linear-gradient(rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0.75)), url(background.jpg);
    background-size: cover;
    background-position: center;
}

.navbar {
    width: 85%;
    margin: auto;
    padding: 35px 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.navbar ul li {
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}

.navbar ul li a {
    text-decoration: none;
    color: #fff;
    text-transform: uppercase;
    font-variant: small-caps;
}
.navbar ul li::after{
    content: '';
    height: 3px;
    width: 0;
    background: #009688;
    position: absolute;
    left: 0;
    bottom: -10px;
    transition: 0.5s;
}
.navbar ul li:hover::after{
    width: 100%;
}
```

Figure 4-10 Figure of CSS Coding of Dashboard

Figure 4-5 until 4-7 shows the User Login Form Interface also CSS and HTML coding. Suppose staff forgets their password and can signup for the Login Form. Figure 4-8 until 4-10 shows the dashboard for user to choose their roles to get into the database with the HTML and CSS coding.

4.2.2 Result

When examining the results generated by our Database Management System (DBMS), it is crucial to acknowledge the integration of data, analysis, and system operations. The subsequent exhaustive outcomes summarise the culmination of the DBMS's capabilities, providing valuable insights, patterns, and performance measurements that aid in making informed decisions and enhancing our comprehension of the database environment.

4.2.3 Table structure in phpMyAdmin

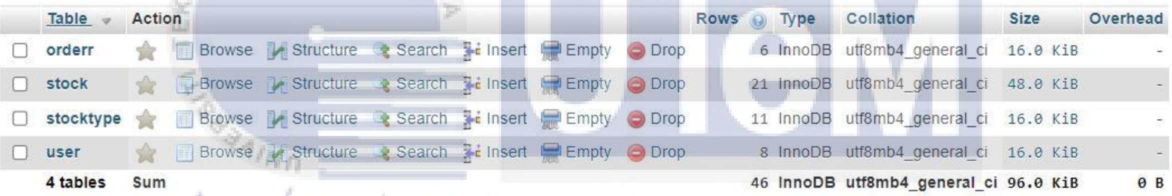
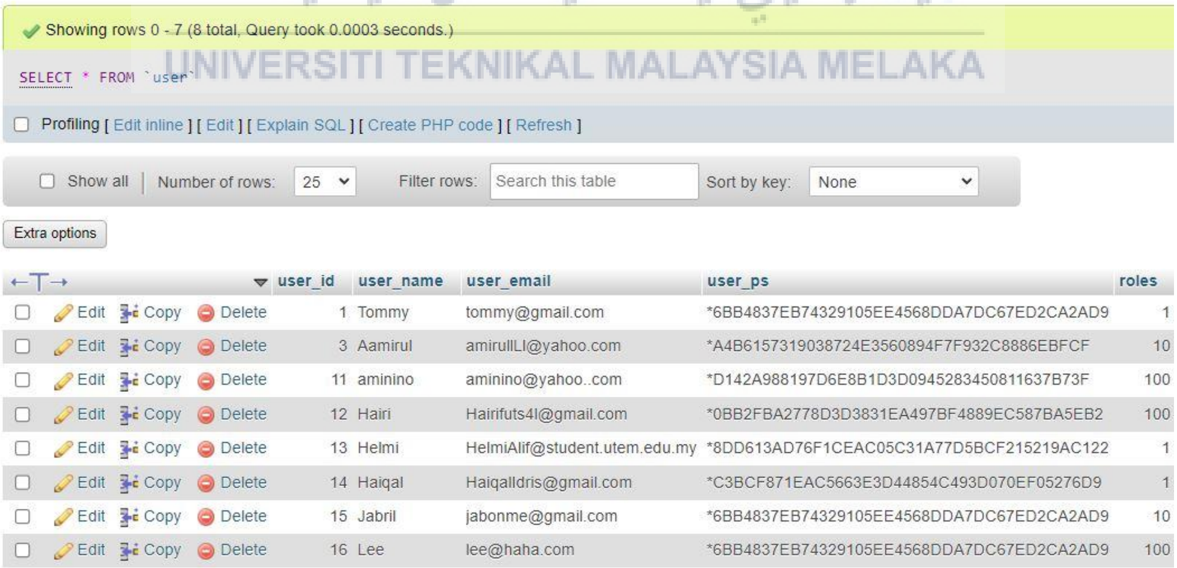


	Table	Action							Rows	Type	Collation	Size	Overhead
□	orderr	☆	Browse	Structure	Search	Insert	Empty	Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	stock	☆	Browse	Structure	Search	Insert	Empty	Drop	21	InnoDB	utf8mb4_general_ci	48.0 KiB	-
□	stocktype	☆	Browse	Structure	Search	Insert	Empty	Drop	11	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	user	☆	Browse	Structure	Search	Insert	Empty	Drop	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	4 tables	Sum							46	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

Figure 4-11 Figure of Table stock_management



✓ Showing rows 0 - 7 (8 total, Query took 0.0003 seconds.)

```
SELECT * FROM `user`
```

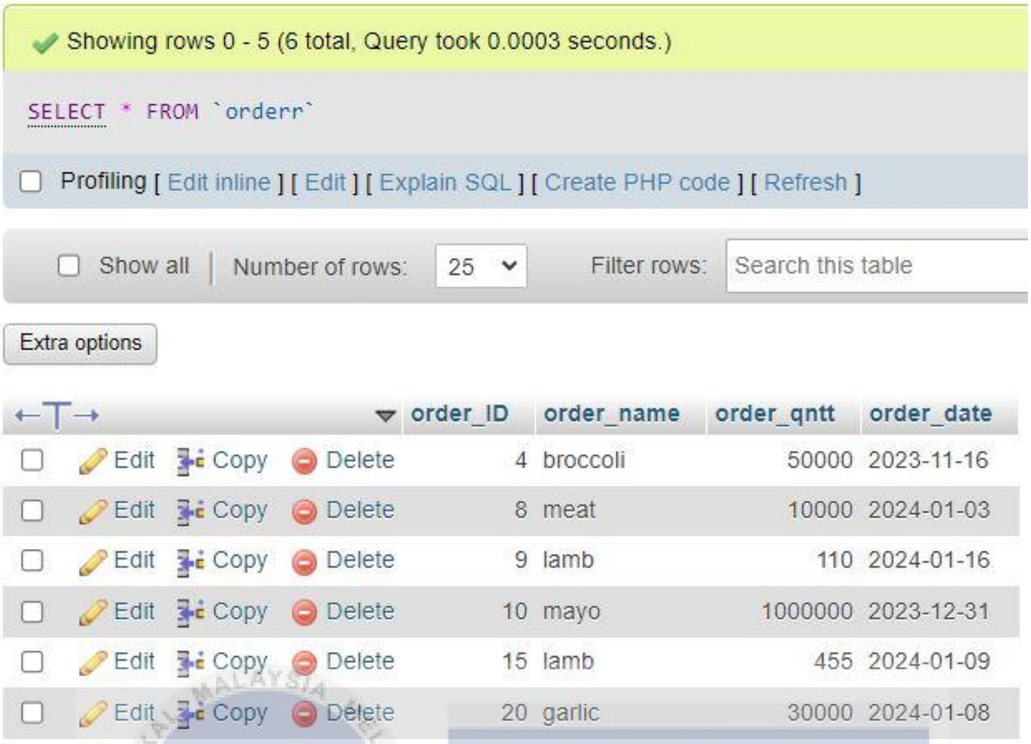
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	←T→			user_id	user_name	user_email	user_ps	roles
□	Edit	Copy	Delete	1	Tommy	tommy@gmail.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9	1
□	Edit	Copy	Delete	3	Aamirul	amirullLI@yahoo.com	*A4B6157319038724E3560894F7F932C8886EBFCF	10
□	Edit	Copy	Delete	11	aminino	aminino@yahoo..com	*D142A988197D6E8B1D3D0945283450811637B73F	100
□	Edit	Copy	Delete	12	Hairi	Hairifuts4l@gmail.com	*0BB2FBA2778D3D3831EA497BF4889EC587BA5EB2	100
□	Edit	Copy	Delete	13	Helmi	HelmiAlif@student.utem.edu.my	*8DD613AD76F1CEAC05C31A77D5BCF215219AC122	1
□	Edit	Copy	Delete	14	Haiqal	HaiqalIdris@gmail.com	*C3BCF871EAC5663E3D44854C493D070EF05276D9	1
□	Edit	Copy	Delete	15	Jabril	jabonme@gmail.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9	10
□	Edit	Copy	Delete	16	Lee	lee@haha.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9	100

Figure 4-12 Figure of Table of user



✓ Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)

```
SELECT * FROM `orderr`
```

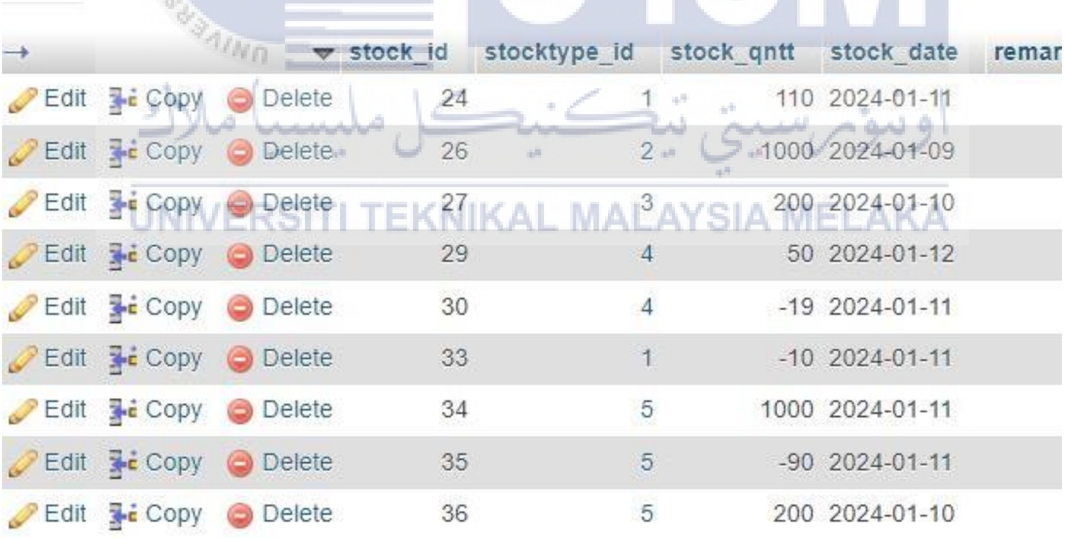
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

Extra options

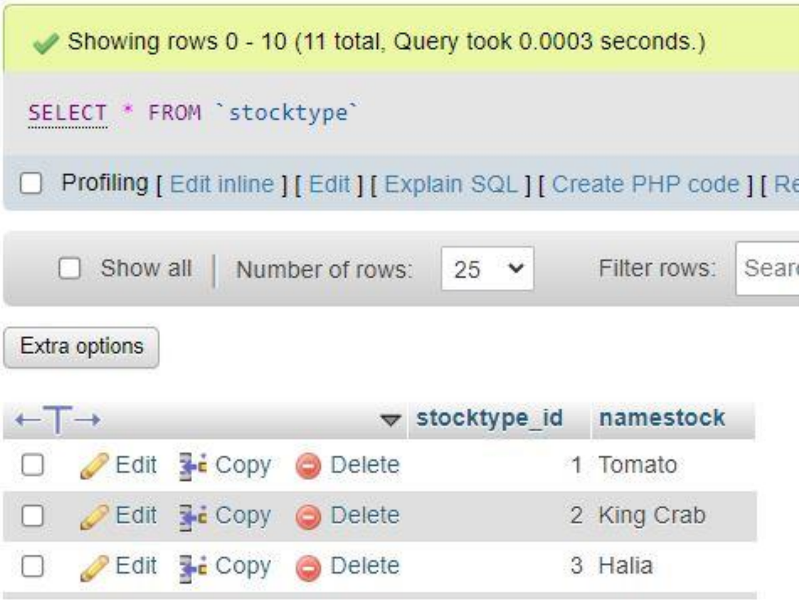
				order_ID	order_name	order_qntt	order_date
□	Edit	Copy	Delete	4	broccoli	50000	2023-11-16
□	Edit	Copy	Delete	8	meat	10000	2024-01-03
□	Edit	Copy	Delete	9	lamb	110	2024-01-16
□	Edit	Copy	Delete	10	mayo	1000000	2023-12-31
□	Edit	Copy	Delete	15	lamb	455	2024-01-09
□	Edit	Copy	Delete	20	garlic	30000	2024-01-08

Figure 4-13 Figure of Table order



			stock_id	stocktype_id	stock_qntt	stock_date	remar
Edit	Copy	Delete	24	1	110	2024-01-11	
Edit	Copy	Delete	26	2	1000	2024-01-09	
Edit	Copy	Delete	27	3	200	2024-01-10	
Edit	Copy	Delete	29	4	50	2024-01-12	
Edit	Copy	Delete	30	4	-19	2024-01-11	
Edit	Copy	Delete	33	1	-10	2024-01-11	
Edit	Copy	Delete	34	5	1000	2024-01-11	
Edit	Copy	Delete	35	5	-90	2024-01-11	
Edit	Copy	Delete	36	5	200	2024-01-10	

Figure 4-14 Figure of Table stock


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Re
□ Show all
Number of rows: 25
Filter rows: Sear
Extra options
Edit
Copy
Delete

Showing rows 0 - 10 (11 total, Query took 0.0003 seconds.)

```
SELECT * FROM `stocktype`
```

stocktype_id	namestock
1	Tomato
2	King Crab
3	Halia

Figure 4-15 Figure of Table stocktype

PhpMyAdmin is a web-based administration tool designed for managing MySQL databases. It provides a user-friendly graphical interface that allows users to perform various tasks related to database management without the need for extensive knowledge of SQL commands. Some key features include creating and modifying databases, managing tables and their structures, executing SQL queries, manipulating data records, and handling user privileges. It is a valuable tool for developers, administrators, and anyone working with MySQL databases, offering convenience and efficiency in database administration tasks. Whether you're creating databases, designing tables, or executing queries, phpMyAdmin simplifies the process, making it accessible to a broad range of users in web development and database management.

4.2.4 Login

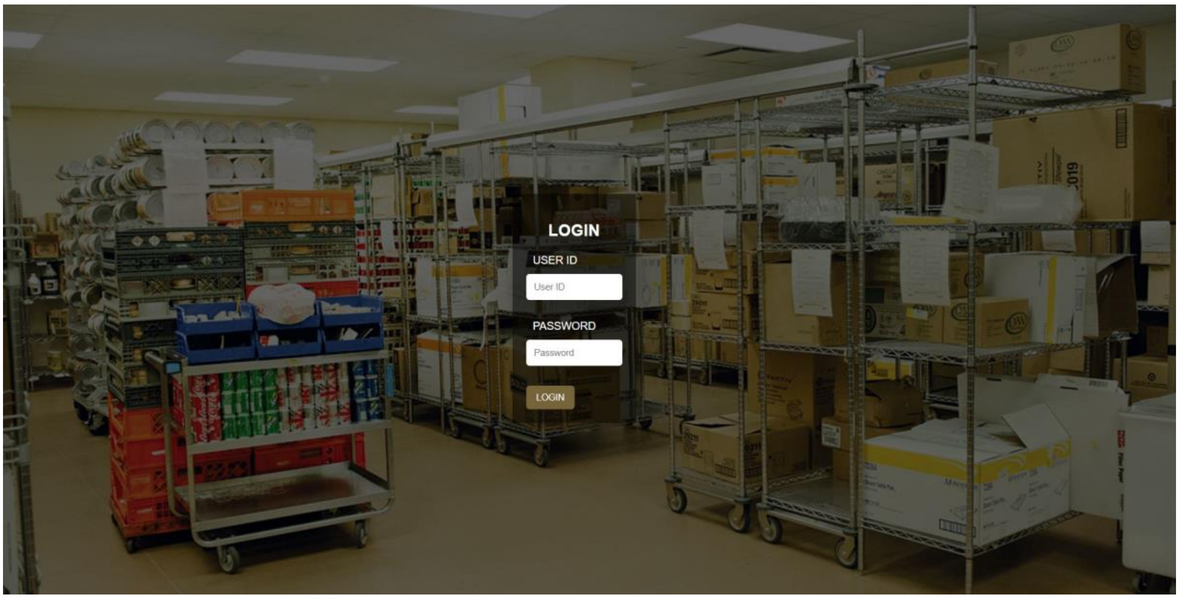
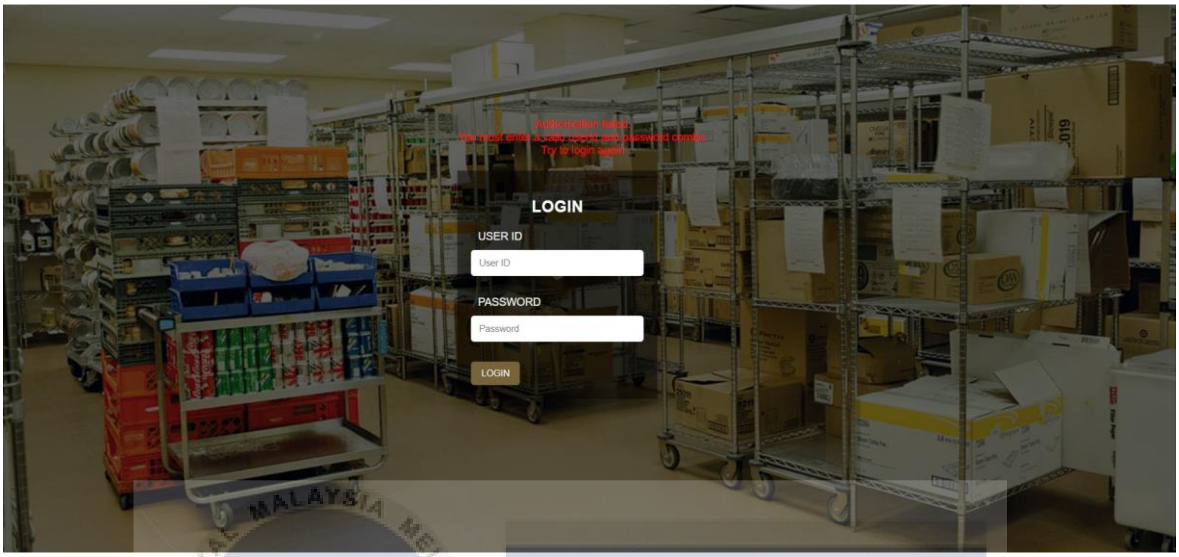


Figure 4-16 Figure of Login Form

This login form presents a straightforward yet secure login form with two input fields, one for the user ID and another for the password. When a user submits the form, the provided input values are sent to the server, where PHP code manages the data. Notably, the form employs the POST method for submission, enhancing security by avoiding the exposure of sensitive information in the URL. All user level will be serve with same interface of login form.

Breaking down the code, it starts with the standard HTML document declaration and language specifications. The head section includes meta-information and sets the document's character set, followed by a title that appears in the browser's title bar. The code links an external CSS file for styling, incorporating a mechanism to ensure the browser fetches the latest version.

Within the body, a div element centers the login form, featuring input fields for the user ID and password. Labels provide clarity to users, and input types are appropriately set to "text" and "password" for user ID and password, respectively. A submit button finalizes the form, triggering the data submission to the server.



Authorization failed.
You must enter a valid userid and password combo.
Try to login again

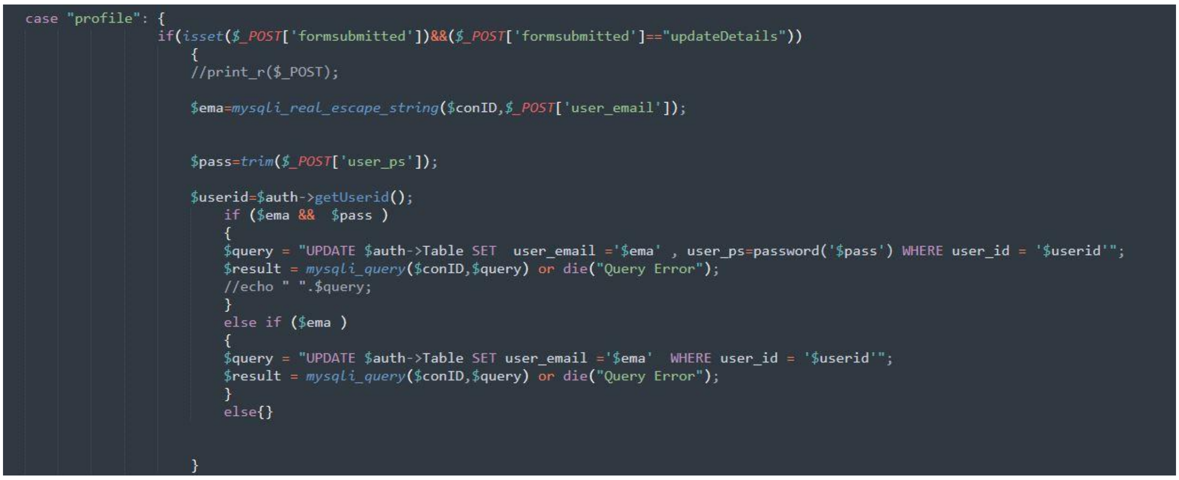
Figure 4-17 Figure of when either password or user ID incorrect

Here when the user put the incorrect password or user ID. It will return to login form. The red caution “Authorization failed. You must enter a valid userid and password combo. Try to login again” will appear. It need the user to enter the right combo of user ID and password to get in the web page.

In essence, this login form is designed to offer a user-friendly experience while maintaining security standards. The PHP backend, not explicitly detailed here, handles authentication processes, permitting access to authorized users exclusively, ensuring the protection of sensitive content on the website.

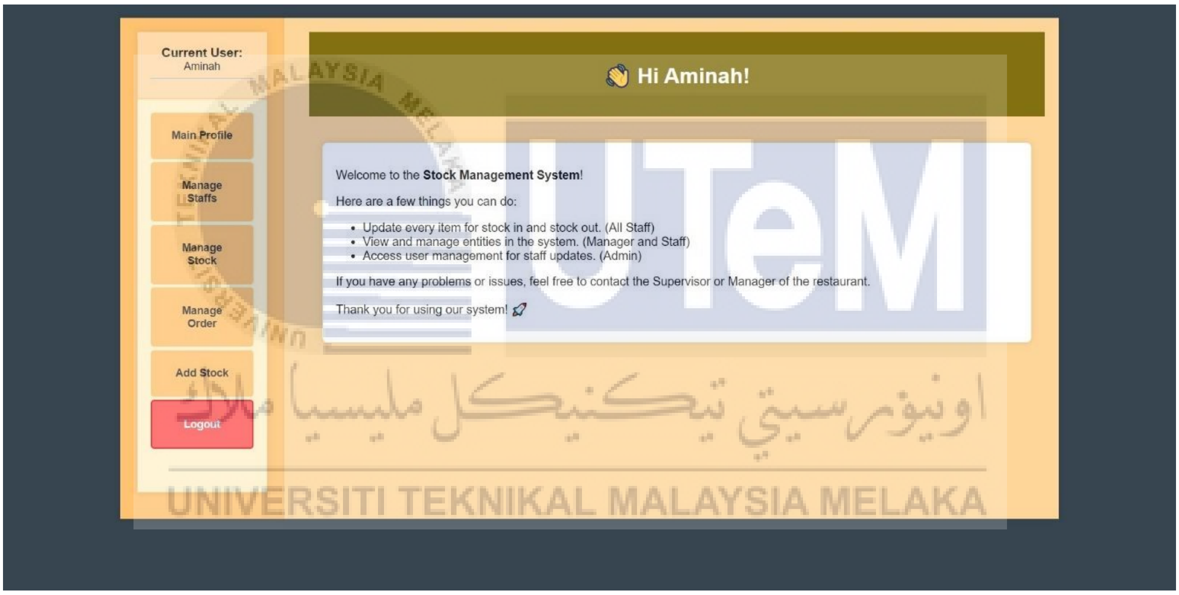
4.2.5 Main Profile

The menu is available to anyone who wants to use it, and anyone can activate the function by clicking the button that is explicitly designated for that purpose.



```
case "profile": {
                if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
                    {
                    //print_r($_POST);

                    $ema=mysqli_real_escape_string($conID,$_POST['user_email']);


                    $pass=trim($_POST['user_ps']);

                    $userid=$auth->getUserid();
                        if ($ema &&  $pass )
                        {
                        $query = "UPDATE $auth->Table SET  user_email ='$ema' , user_ps=password('$pass') WHERE user_id = '$userid'";
                        $result = mysqli_query($conID,$query) or die("Query Error");
                        //echo " ".$query;
                        }
                        else if ($ema )
                        {
                        $query = "UPDATE $auth->Table SET user_email ='$ema'  WHERE user_id = '$userid'";
                        $result = mysqli_query($conID,$query) or die("Query Error");
                        }
                        else{}


                    }
```

Figure 4-18 Figure of Snippet profile case source code



Welcome to the **Stock Management System**!

Here are a few things you can do:

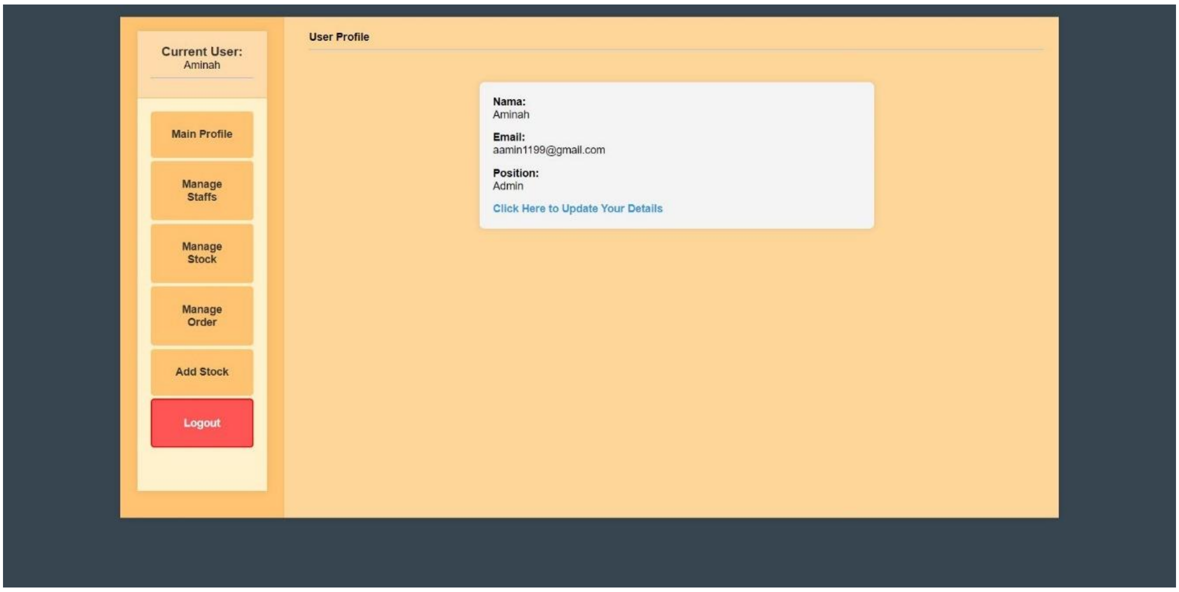
- Update every item for stock in and stock out. (All Staff)
- View and manage entities in the system. (Manager and Staff)
- Access user management for staff updates. (Admin)

If you have any problems or issues, feel free to contact the Supervisor or Manager of the restaurant.

Thank you for using our system! 🚀

Figure 4-19 Figure of Admin first page

The initial page that the administrator will encounter is this particular page, which provides a visual representation of various elements. The administrator has access to five menu options: Main Profile, Manage Staff, Manage Stock, Add Stock, and the universal Log Out button, which is displayed as a red button for all user levels.



User Profile

Nama:
Aminah

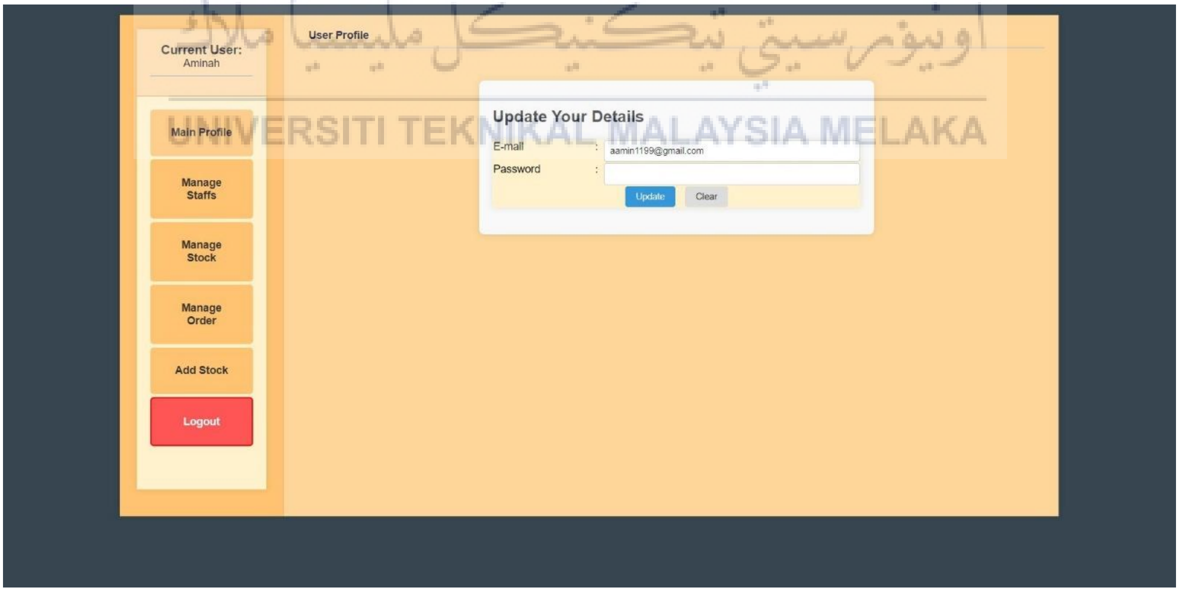
Email:
aamin1199@gmail.com

Position:
Admin

Click Here to Update Your Details

Figure 4-20 Figure of Admin main profile

The primary profile contains vital details, such as the person's name, email address, and job title. The position can only be selected from three options: administrator, manager, and staff. The choice will be made only by Admin. Users possess the capability to modify their name and email address.



User Profile

Update Your Details

E-mail : aamin1199@gmail.com

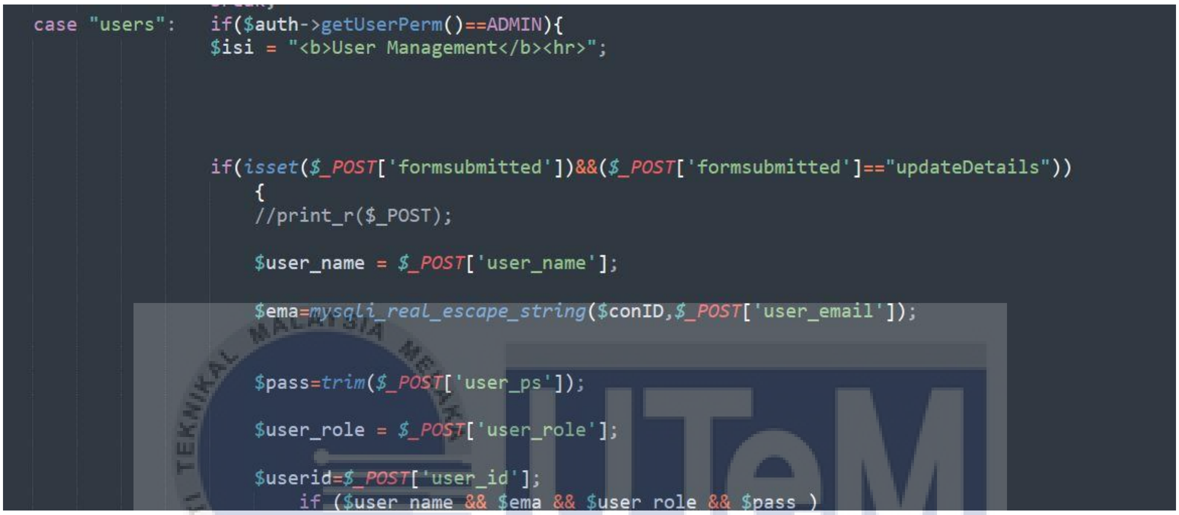
Password :

Figure 4-21 Figure of Admin Update Profile

On this page, every user has the option to change their password and email address if they so desire.

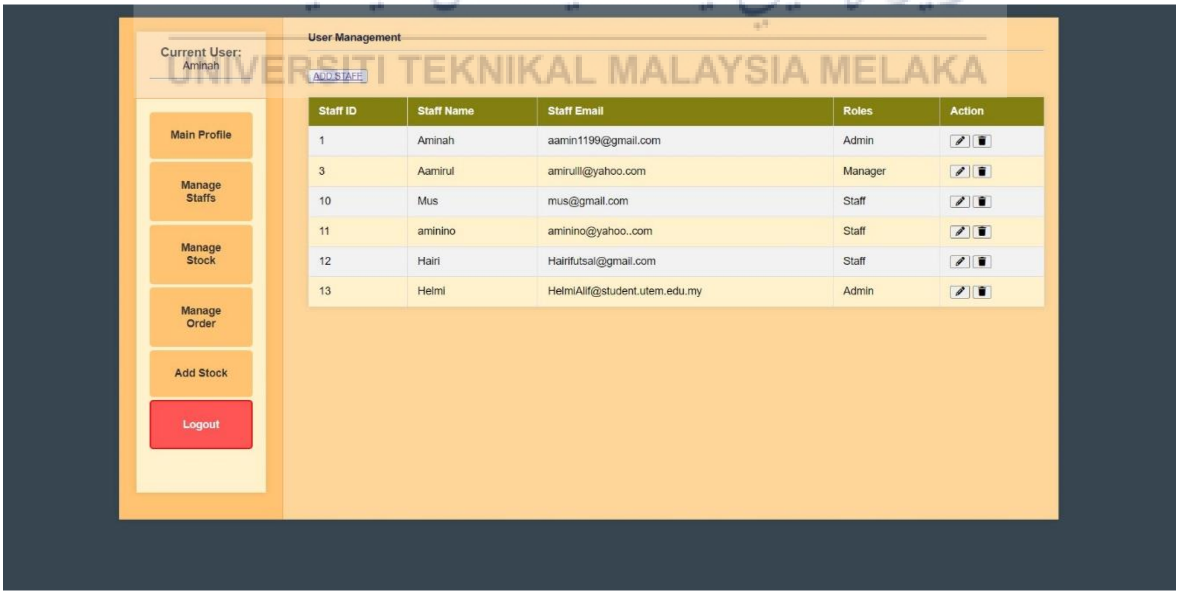
4.2.6 Manage Staffs

Managing staff members is a responsibility that is reserved exclusively for administrators. When it comes to assigning positions, they have the authority to configure passwords and choose from one of three options: administrator, manager, or staff.



```
case "users":   if($auth->getUserPerm()==ADMIN){
                $isi = "<b>User Management</b><hr>";



                if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
                    {
                    //print_r($_POST);

                    $user_name = $_POST['user_name'];

                    $ema=mysqli_real_escape_string($conID,$_POST['user_email']);


                    $pass=trim($_POST['user_ps']);

                    $user_role = $_POST['user_role'];

                    $userid=$_POST['user_id'];
                        if ($user_name && $ema && $user_role && $pass )
```

Figure 4-22 Figure of Snippet case users source code

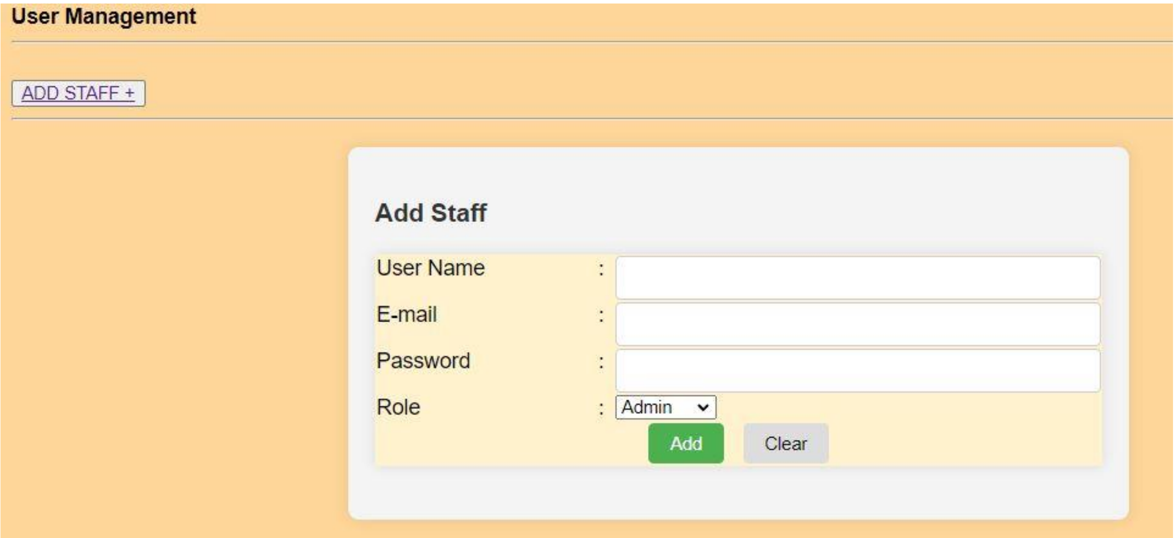


User Management

Staff ID	Staff Name	Staff Email	Roles	Action
1	Aminah	aamin1199@gmail.com	Admin	✎ 🗑
3	Aamirul	amirulll@yahoo.com	Manager	✎ 🗑
10	Mus	mus@gmail.com	Staff	✎ 🗑
11	aminino	aminino@yahoo..com	Staff	✎ 🗑
12	Hairi	Hairifutsal@gmail.com	Staff	✎ 🗑
13	Helmi	HelmiAlif@student.utem.edu.my	Admin	✎ 🗑

Figure 4-23 Figure of manage Staff page

The administrators are the only ones who can access the primary page for managing staff.


User Management

Add Staff

User Name :

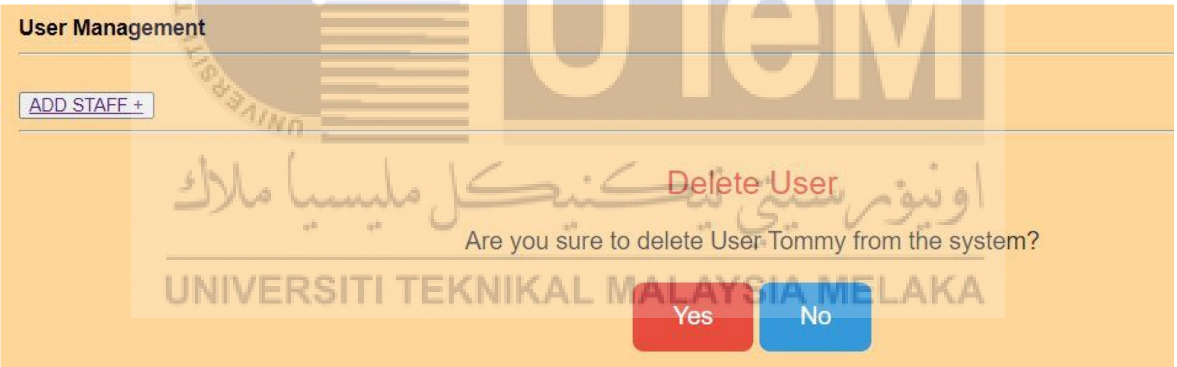
E-mail :

Password :

Role : Admin

Figure 4-24 Figure of Add Staff page

The administrator can add new staff by clicking the button labelled "Add Staff," and the process must be completed before the new staff is added to the table.

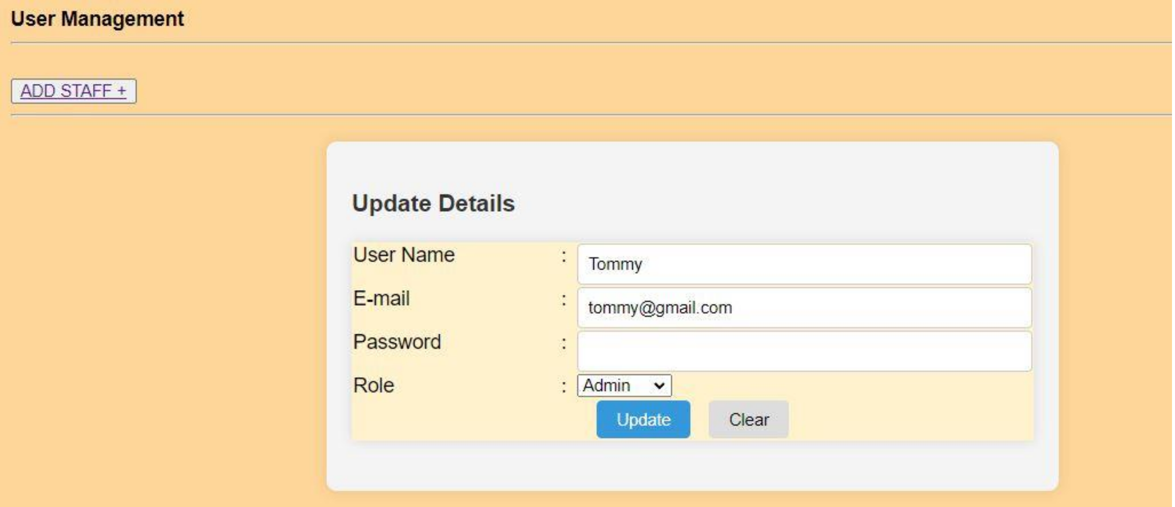

User Management

Delete User

Are you sure to delete User Tommy from the system?

Figure 4-25 Figure of Delete Staff

When the user presses the icon that looks like a trash can, they will be asked to confirm whether or not they want to delete the row in question.


User Management

ADD STAFF +

Update Details

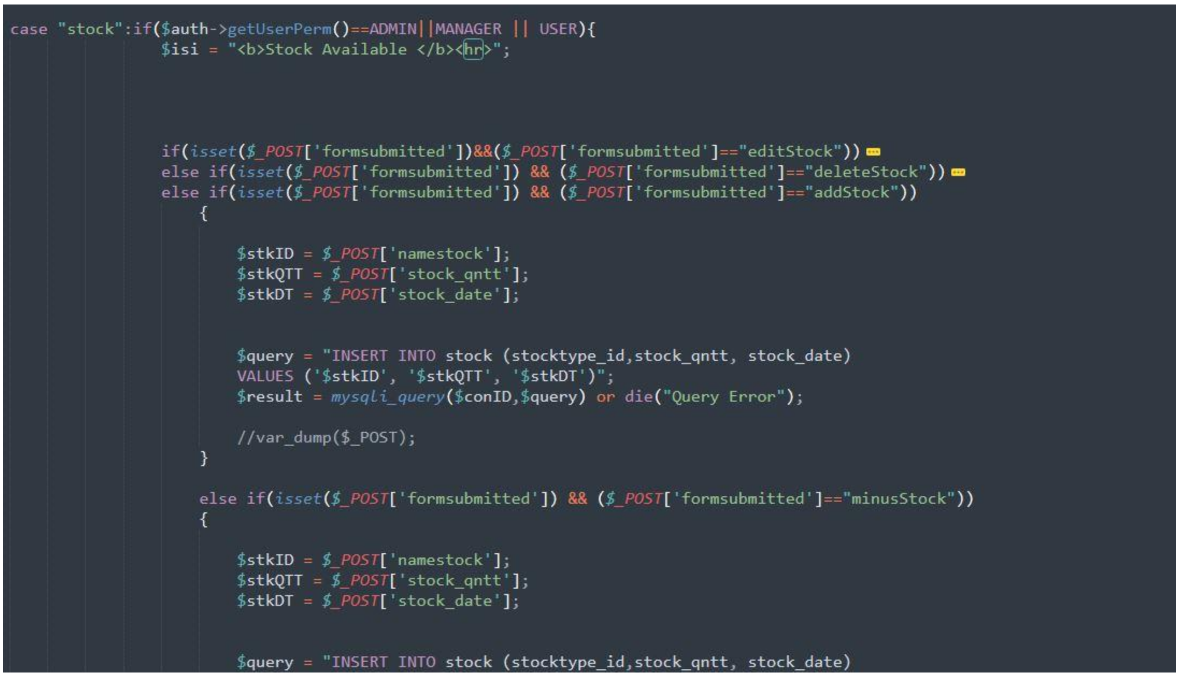
User Name	:	Tommy
E-mail	:	tommy@gmail.com
Password	:	
Role	:	Admin

Figure 4-26 Figure of Update Staff Details

Administrators also have the ability to make changes to all records pertaining to staff members. There are a number of predetermined options from which the role can be selected.

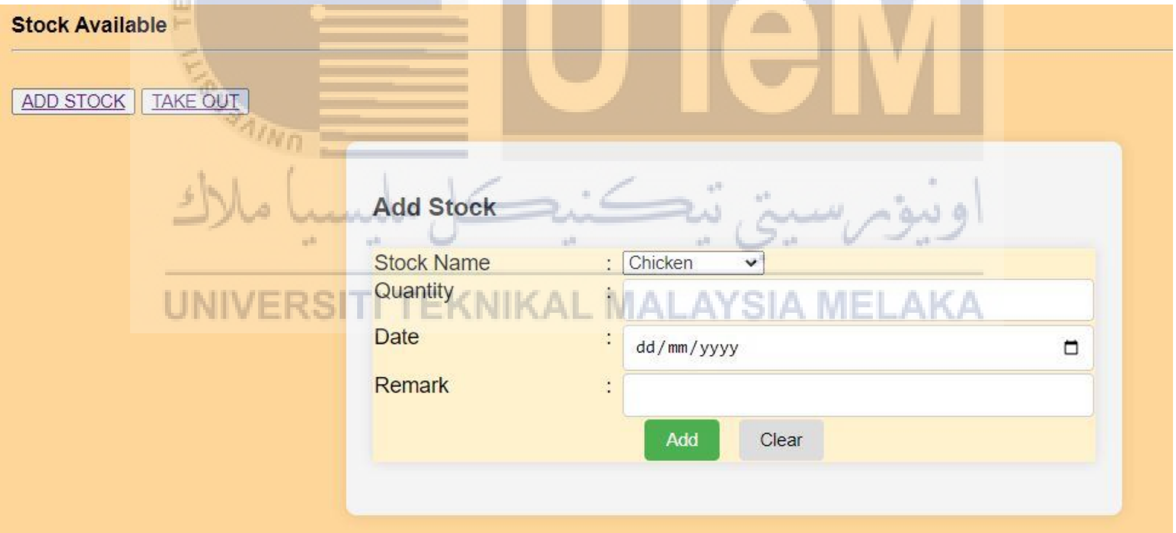
4.2.7 Manage Stock

Given that the management of stock is the primary focus of the system, everyone is able to utilise this capability. With the exception of inventory, the term "stock" in this context refers to all of the ingredients that are kept in the storeroom. The items that fall under this category are those that need to be carefully monitored for their expiration dates.



```
case "stock":if($auth->getUserPerm()==ADMIN||MANAGER || USER){
                $isi = "<b>Stock Available </b><hr>";



                if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="editStock"))
                else if(isset($_POST['formsubmitted']) && ($_POST['formsubmitted']=="deleteStock"))
                else if(isset($_POST['formsubmitted']) && ($_POST['formsubmitted']=="addStock"))
                    {

                        $stkID = $_POST['namestock'];
                        $stkQTT = $_POST['stock_qntt'];
                        $stkDT = $_POST['stock_date'];


                        $query = "INSERT INTO stock (stocktype_id,stock_qntt, stock_date)
                        VALUES ('$stkID', '$stkQTT', '$stkDT')";
                        $result = mysqli_query($conID,$query) or die("Query Error");

                        //var_dump($_POST);
                    }

                    else if(isset($_POST['formsubmitted']) && ($_POST['formsubmitted']=="minusStock"))
                    {

                        $stkID = $_POST['namestock'];
                        $stkQTT = $_POST['stock_qntt'];
                        $stkDT = $_POST['stock_date'];


                        $query = "INSERT INTO stock (stocktype_id,stock_qntt, stock_date)
```

Figure 4-27 Figure of Snippet case stock source code



Stock Available

Stock Name : Chicken

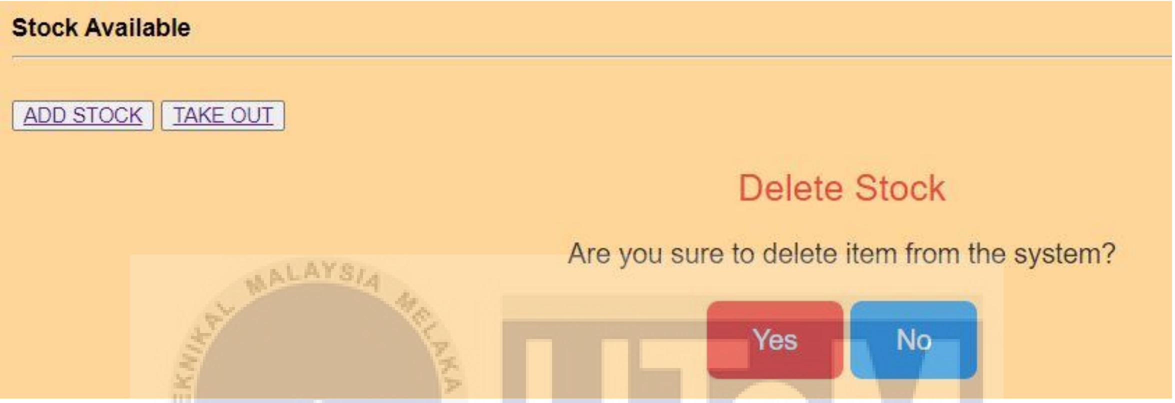
Quantity :

Date : dd/mm/yyyy

Remark :

Figure 4-28 Figure of Add Stock into Storeroom

The function known as "Add Stock" is responsible for the implementation of the addition of stock names, and only the Admin and Manager roles are granted access to this function. The stock name is fixed and retrieved from the function within the "Add" button, which provides users with a selection pane for the purpose of making their choice more convenient. Quantities can only be entered by users exclusively in the form of numerical values, with the exception of administrators and managers. Additionally, a calendar function is incorporated immediately after the blank space, which makes it easier for users to select dates using the calendar. When you have finished selecting the date, it will be saved in the format dd/mm/yy. By utilising this configuration, the stock management system is able to maintain its data integrity while simultaneously providing a user-friendly experience.



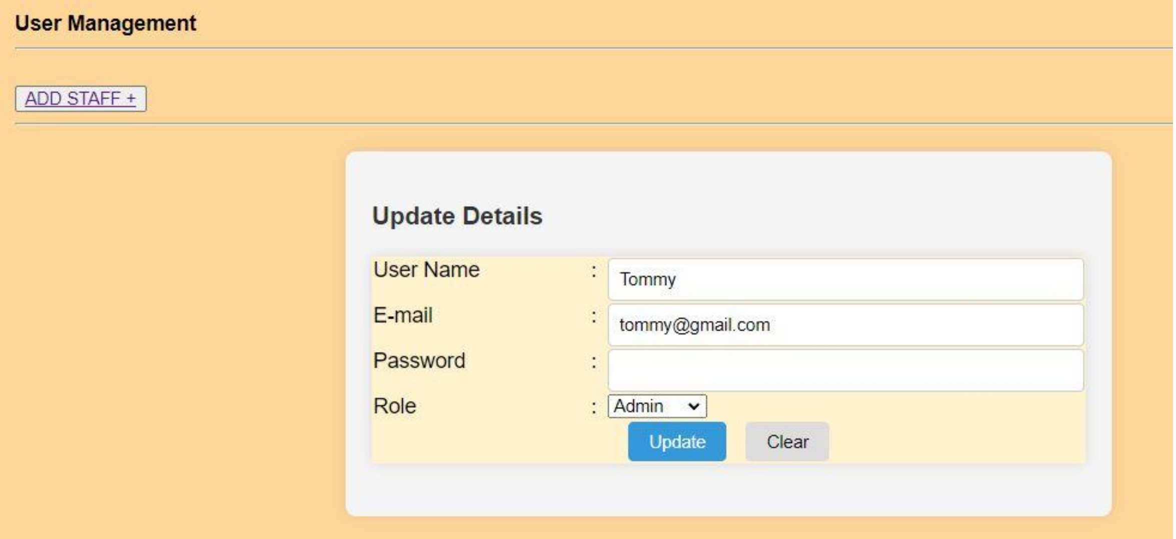
Stock Available

Delete Stock

Are you sure to delete item from the system?

Figure 4-29 Figure of Delete Stok from Storeroom

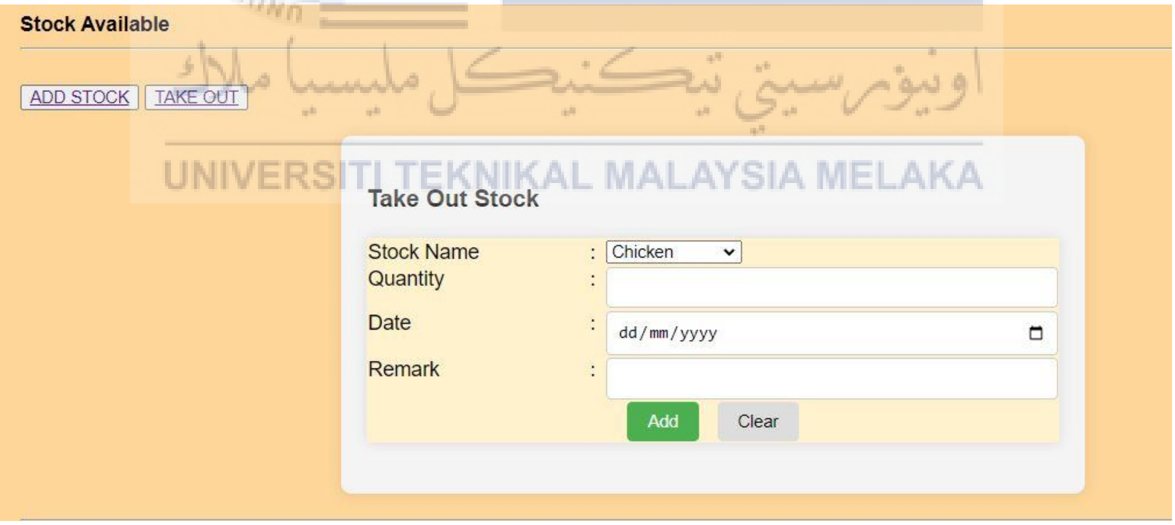
To carry out the function of deleting rows, a button that is emblematic of a garbage can is implemented. This functionality is analogous to the functionality that was previously implemented. The user will be presented with a confirmation prompt prior to the deletion of data, which will ensure that the row in question is removed in a manner that is both intentional and secure. This strategy improves the security of the data and offers users an additional layer of assurance before any actions that cannot be undone are carried out within the system.



User Name	:	Tommy
E-mail	:	tommy@gmail.com
Password	:	
Role	:	Admin

Figure 4-30 Figure of Update Stok

The "Update Stock" functionality facilitates modifications to the quantity and date of a stock entry. This feature empowers users to adjust stock details, ensuring accurate and up-to-date information in the system. The update process encompasses changes to both the quantity and the associated date, allowing for comprehensive and precise stock management.



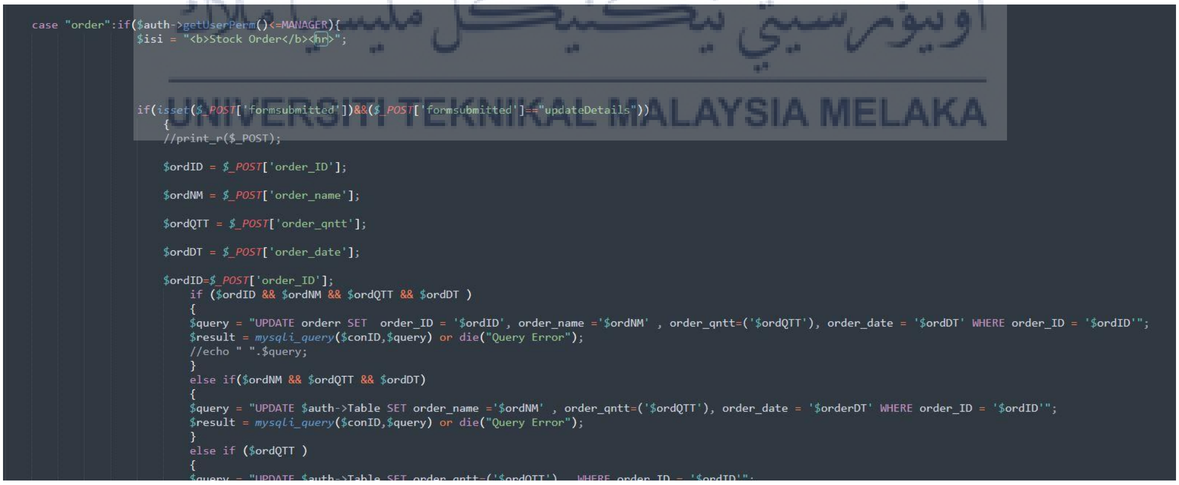
Stock Name	:	Chicken
Quantity	:	
Date	:	dd/mm/yyyy
Remark	:	

Figure 4-31 Figure of Take Out Stock From Storeroom

When the "Take Out Stock" operation is carried out, the items that are in stock are removed from the storage area. To carry out this action, users must first select the items they want to remove, and then they must input the quantity of items they want to remove. Note that the quantity value must be expressed as a negative figure, as this indicates a reduction from the stock that is already present. This is an extremely important point to keep in mind. In the subsequent step, the system will update the recorded table by subtracting the quantity that was specified. This will ensure that the stock management is accurate and reflective.

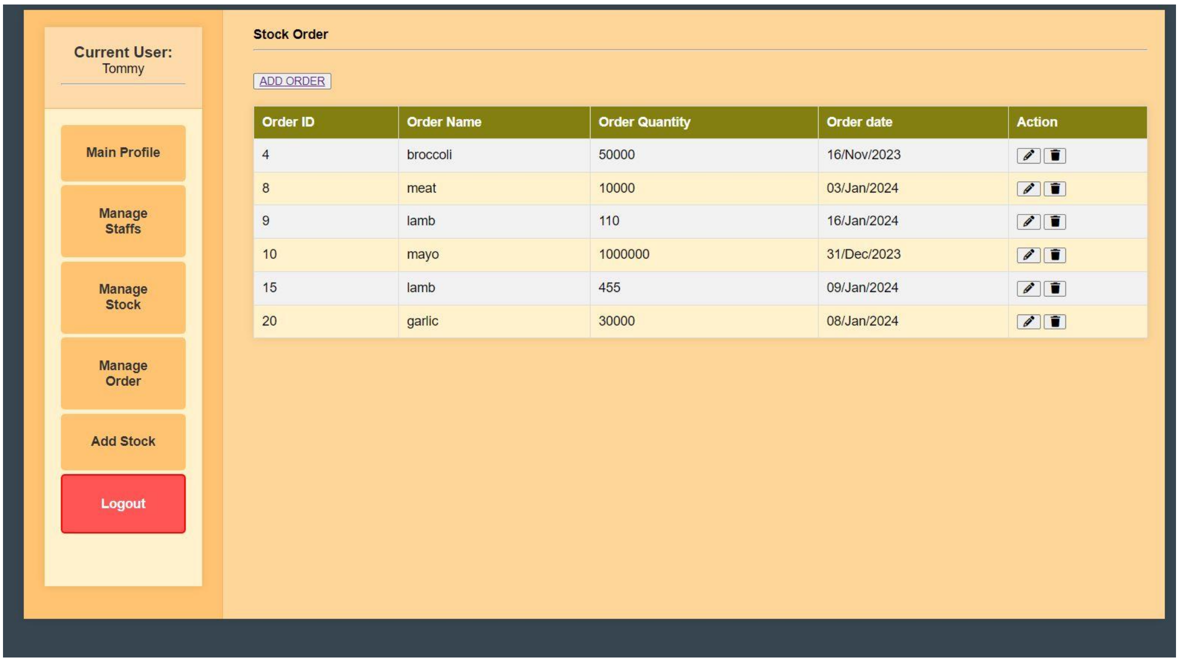
4.2.8 Manage Order

Access to this particular operation, specifically the "Take Out Stock" functionality, is restricted to users with Admin and Manager roles. This feature is designed for future use, allowing the addition of new products or items to be recorded preemptively before they are formally ordered. By limiting access to authorized personnel, such as administrators and managers, the system ensures controlled and secure management of stock-related activities. This approach maintains data integrity and aligns with a proactive strategy for stock recording and future procurement planning.



```
case "order":if($auth->getUserPerm()<=MANAGER){
            $isi = "<b>Stock Order</b><hr>";


            if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
                {
                //print_r($_POST);

                $ordID = $_POST['order_ID'];

                $ordNM = $_POST['order_name'];

                $ordQTT = $_POST['order_qntt'];

                $ordDT = $_POST['order_date'];

                $ordID=$_POST['order_ID'];
                    if ($ordID && $ordNM && $ordQTT && $ordDT )
                    {
                    $query = "UPDATE orderr SET  order_ID = '$ordID', order_name ='$ordNM' , order_qntt=('$ordQTT'), order_date = '$ordDT' WHERE order_ID = '$ordID'";
                    $result = mysqli_query($conID,$query) or die("Query Error");
                    //echo " ".$query;
                    }
                    else if($ordNM && $ordQTT && $ordDT)
                    {
                    $query = "UPDATE $auth->Table SET order_name ='$ordNM' , order_qntt=('$ordQTT'), order_date = '$orderDT' WHERE order_ID = '$ordID'";
                    $result = mysqli_query($conID,$query) or die("Query Error");
                    }
                    else if ($ordQTT )
                    {
                    $query = "UPDATE $auth->Table SET order_qntt=('$ordQTT')  WHERE order_ID = '$ordID'";
```

Figure 4-32 Figure of Snippet case order source code



Current User:
Tommy

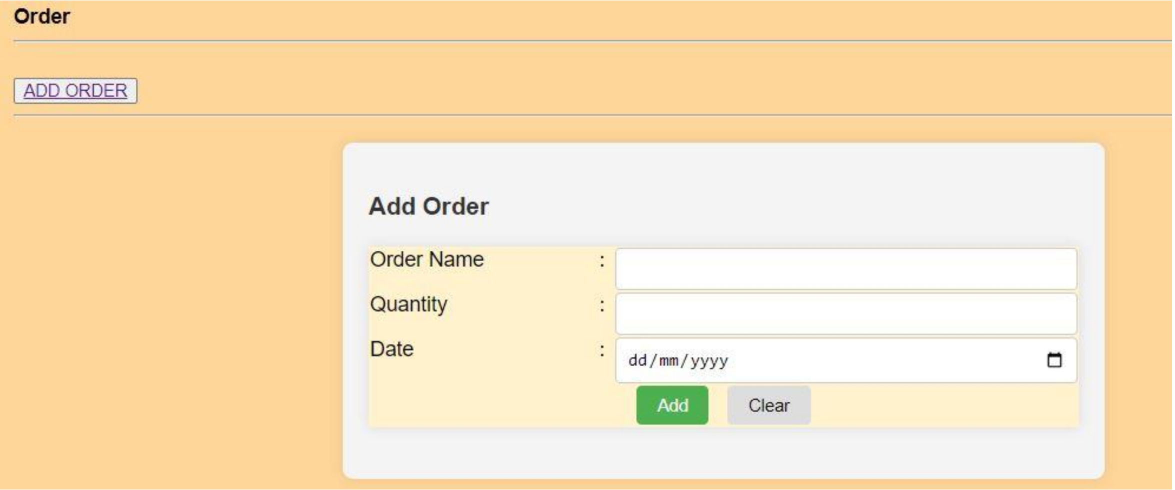
Stock Order

ADD ORDER

Order ID	Order Name	Order Quantity	Order date	Action
4	broccoli	50000	16/Nov/2023	✎ 🗑
8	meat	10000	03/Jan/2024	✎ 🗑
9	lamb	110	16/Jan/2024	✎ 🗑
10	mayo	1000000	31/Dec/2023	✎ 🗑
15	lamb	455	09/Jan/2024	✎ 🗑
20	garlic	30000	08/Jan/2024	✎ 🗑

Figure 4-33 Figure of Manage Order

The user interface includes an "Add Order" button, complemented by icons for editing and deleting orders. This design offers a clear and concise visual representation of available actions, empowering users to seamlessly add new orders, edit existing ones, or remove orders as needed. The combination of these features provides a user-friendly and efficient means to manage and interact with the order-related functions within the system.



Order

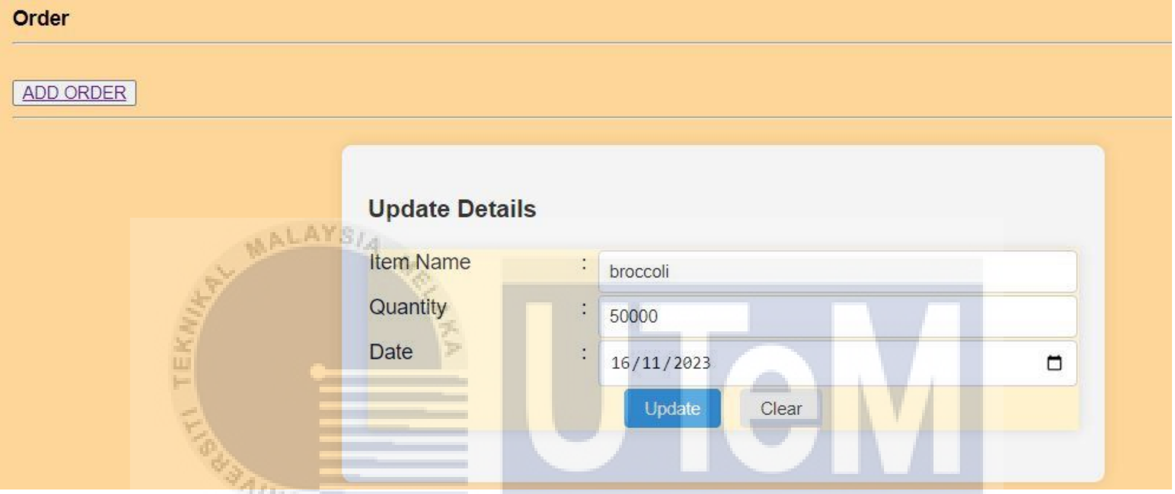
ADD ORDER

Add Order

Order Name	:	
Quantity	:	
Date	:	dd/mm/yyyy

Figure 4-34 Figure of add order in Manage Order

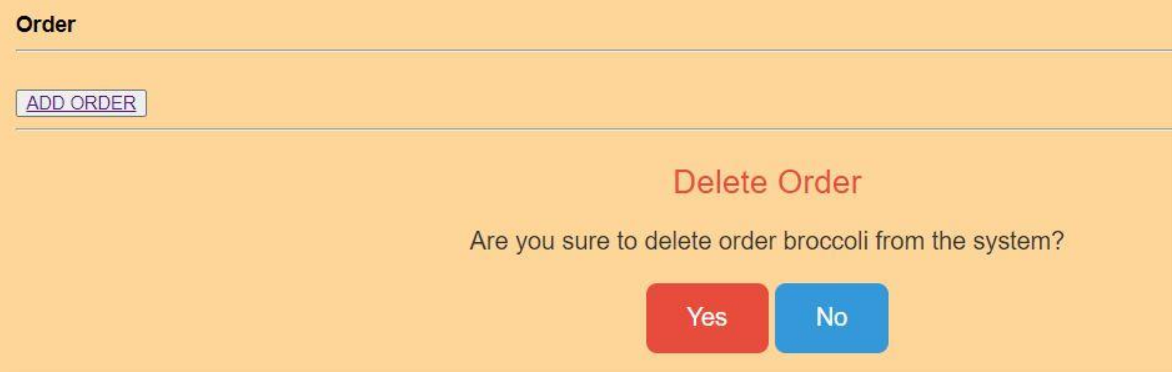
Upon clicking the "Add Order" button, the system activates a form labeled "Add Order." Users are required to complete the form by providing essential details, including the order name, quantity, and date. Only after furnishing this requisite information will the system proceed to add the order entry into the table. This approach ensures that each order is accompanied by comprehensive and accurate information, contributing to a well-organized and systematically recorded set of orders within the system.



Item Name	:	broccoli
Quantity	:	50000
Date	:	16/11/2023

Figure 4-35 Figure of update details in Manage Order

The "Update Details" functionality is designed to facilitate modifications to specific details of an item within the system. Users can initiate updates to the item name, quantity, and associated date through this feature. This capability ensures the system's adaptability to changes over time, allowing for accurate and current information about each item to be maintained in the system.



Delete Order

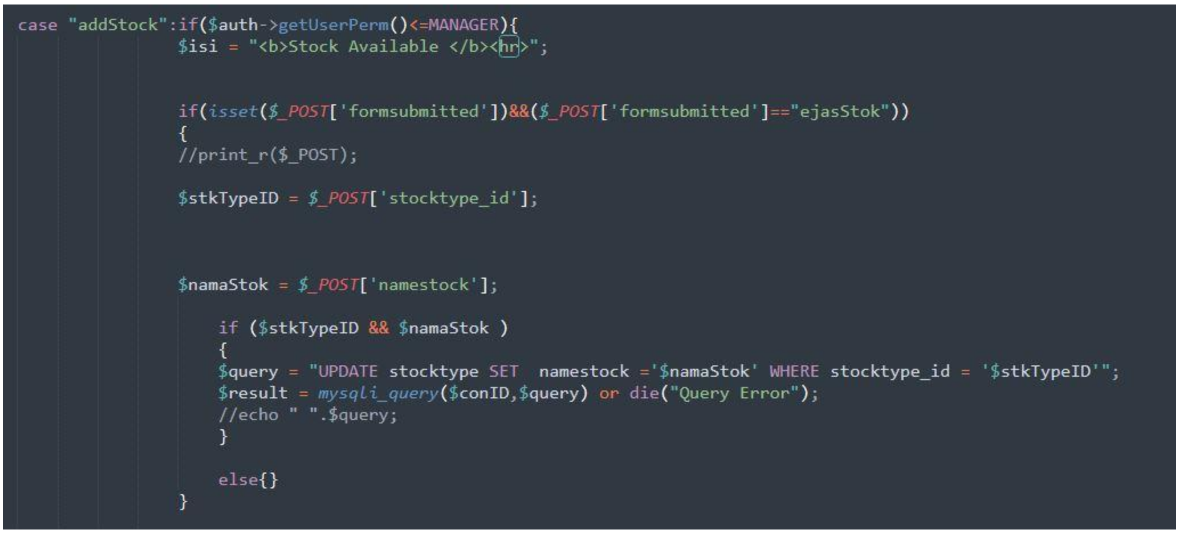
Are you sure to delete order broccoli from the system?

Figure 4-36 Figure of delete order in Manage Order

Delete icon will delete data. The presence of the delete icon serves as a mechanism to remove data within the system. Upon activating this icon, the associated data will be deleted, providing users with a straightforward means to manage and declutter information as needed. Prior to the deletion, a confirmation step is typically implemented to ensure intentional and secure removal of the specified data.

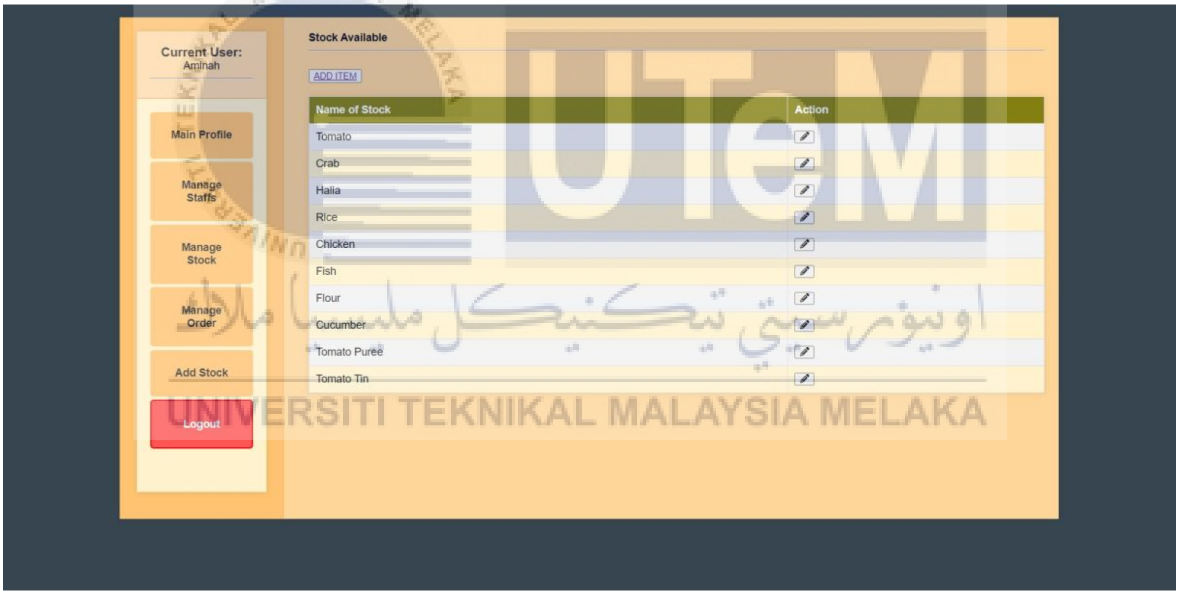
4.2.9 Add Stock

The "Add Stock" function is inherently connected with the broader context of managing stock. This integration serves the purpose of preventing staff members from independently renaming item names. By consolidating the stock addition process within the overarching framework of stock management, the system ensures consistency and control over the naming conventions of items. This approach contributes to maintaining uniformity in the system's stock records and mitigates the risk of inadvertent or unauthorized alterations to item names by staff members.



```
case "addStock":if($auth->getUserPerm()<=MANAGER){
                $isi = "<b>Stock Available </b><hr>";


                if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="ejasStok"))
                {
                //print_r($_POST);

                $stkTypeID = $_POST['stocktype_id'];



                $namaStok = $_POST['namestock'];

                    if ($stkTypeID && $namaStok )
                    {
                    $query = "UPDATE stocktype SET  namestock ='$namaStok' WHERE stocktype_id = '$stkTypeID'";
                    $result = mysqli_query($conID,$query) or die("Query Error");
                    //echo " ".$query;
                    }

                    else{}
                }
```

Figure 4-37 Figure of Snippet case addStock source code

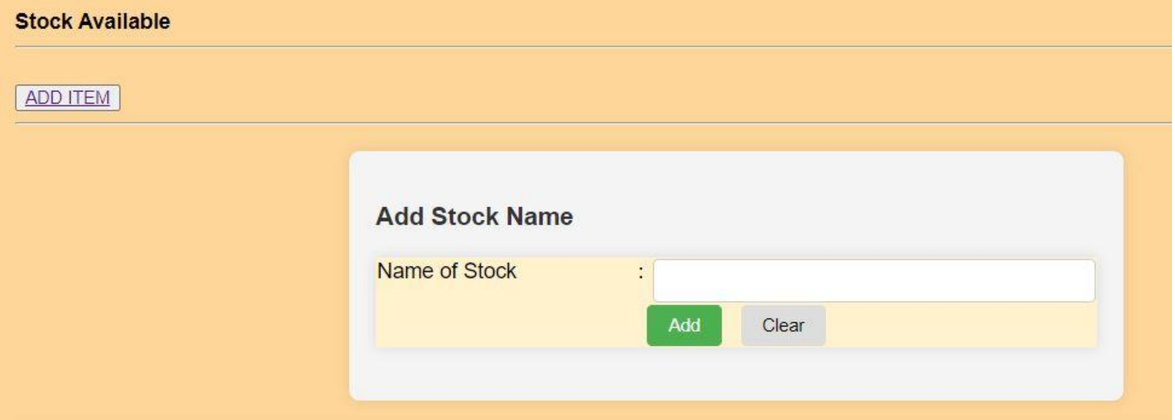


Stock Available

Name of Stock	Action
Tomato	✎
Crab	✎
Halia	✎
Rice	✎
Chicken	✎
Fish	✎
Flour	✎
Cucumber	✎
Tomato Puree	✎
Tomato Tin	✎

Figure 4-38 Figure of Add Stock

The functionality is streamlined and comprises two main actions: the "Add Item" button and the option to "Update Item Name." This simplicity in functionality ensures a focused approach within the system. The "Add Item" button facilitates the incorporation of new items, while the "Update Item Name" feature allows for modifications to existing item names. This straightforward design aims to provide users with clear and efficient tools for managing item-related information within the system.

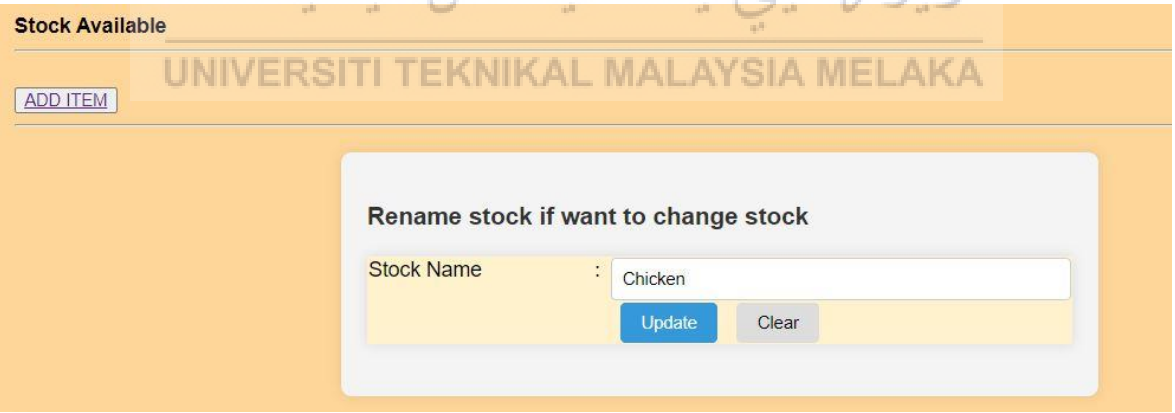

Stock Available

Add Stock Name

Name of Stock :

Figure 4-39 Figure of Add Stock for add item

The "Add Stock Name" feature becomes relevant when users click the "Add Item" button to introduce new items. This functionality allows users to seamlessly include a stock name for the newly added item. By associating stock names during the item creation process, the system ensures that each item is distinctly identified within the stock management system. This integrated approach simplifies the addition of new items while maintaining consistency in stock naming conventions.


Stock Available

Rename stock if want to change stock

Stock Name : Chicken

Figure 4-40 Figure of Rename item

The "Update Details" function serves the purpose of renaming an item within the system. Users can utilize this feature to replace the existing item name with a new one, allowing for efficient management of item nomenclature and optimizing space for future functionalities.

This capability ensures the adaptability of the system to evolving requirements, providing a means to update and refine item details as needed.

4.3 Analysis

For analysis part, stress test has been made to show how reliable the website. Stress testing, including the type conducted using ApacheBench, is a valuable method for assessing the reliability and robustness of a website or web application. By subjecting the system to a simulated high load of concurrent requests, stress testing helps evaluate how well the website can handle increased traffic, identify potential bottlenecks, and determine its overall stability under stressful conditions.

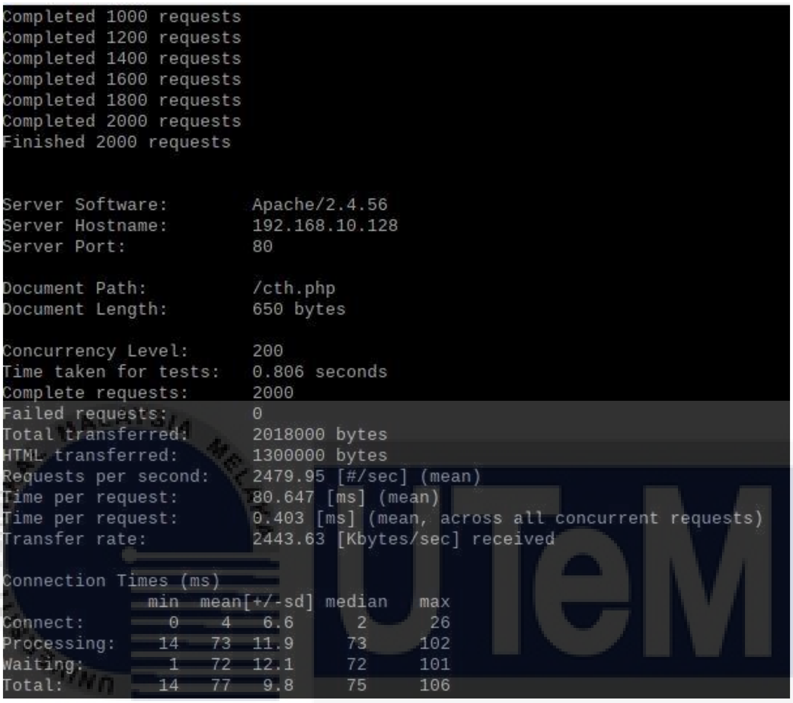
The primary goal of stress testing is to understand the breaking point or limitations of the system, ensuring that it can maintain acceptable performance levels even during peak usage. By analyzing metrics such as Requests per Second (RPS), Time per Request, and Transfer Rate, you gain insights into the system's responsiveness, efficiency, and scalability.

In summary, stress testing provides crucial information about the website's reliability and performance under challenging circumstances, helping developers and administrators make informed decisions about optimizations, scaling, and resource allocation.

Importantly, hashing is one-way. This property makes it difficult to decipher the original data if someone gets the hashed version.

Hashing is used to protect sensitive data like passwords in computers and security, keeping the original data secure and difficult to decipher even if hashed.

4.3.1 Stress test using Apache Bench (ab)

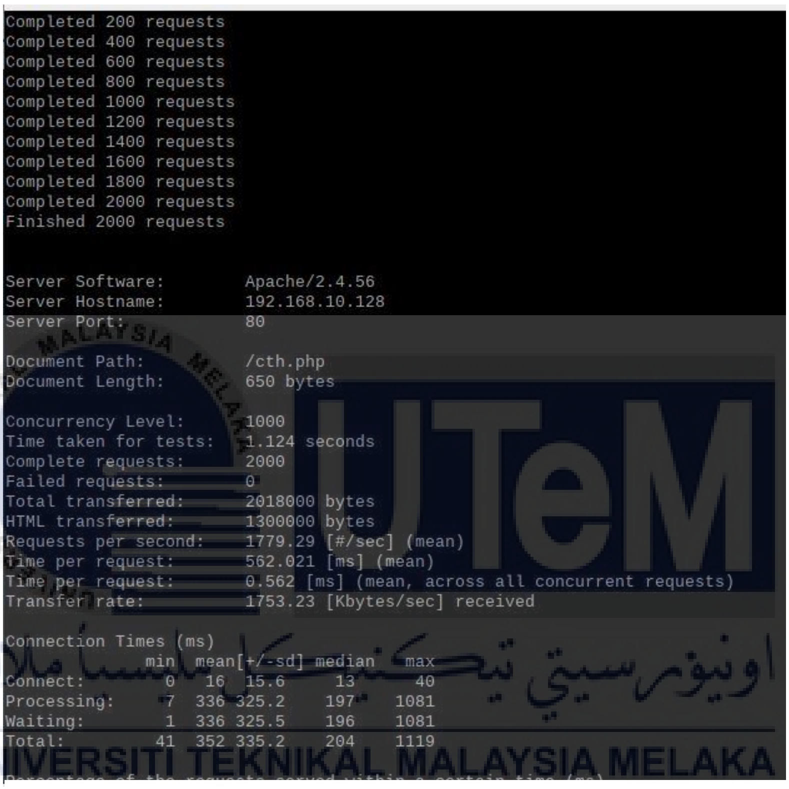


```
Completed 1000 requests
Completed 1200 requests
Completed 1400 requests
Completed 1600 requests
Completed 1800 requests
Completed 2000 requests
Finished 2000 requests


Server Software:        Apache/2.4.56
Server Hostname:        192.168.10.128
Server Port:            80

Document Path:          /cth.php
Document Length:        650 bytes

Concurrency Level:      200
Time taken for tests:   0.806 seconds
Complete requests:      2000
Failed requests:        0
Total transferred:      2018000 bytes
HTML transferred:       1300000 bytes
Requests per second:    2479.95 [#/sec] (mean)
Time per request:       80.647 [ms] (mean)
Time per request:       0.403 [ms] (mean, across all concurrent requests)
Transfer rate:          2443.63 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    4   6.6      2      26
Processing:    14   73  11.9     73     102
Waiting:        1   72  12.1     72     101
Total:         14   77   9.8     75     106
```

Figure 4-41 Figure of Stress Test 1

Here where the test 1 made using command line ab -n 2000 -c 200 “localhost/Stock_Management/cth.php”. Concurrency level here is 200 and request for 2000 users. The test resulted in 2000 successful requests, with no failed requests. The total amount of data transferred was 2018000 bytes, with 1300000 bytes being HTML content. The server was able to complete the requests at a rate of 2479.95 requests per second, with an average time of 80.647 milliseconds per request. The transfer rate was 2443.63 Kbytes/sec.

The connection times are also reported, with a connect time of 0 ms, processing time of 14 ms, waiting time of 0 ms, and a total time of 1 ms.



```
Completed 200 requests
Completed 400 requests
Completed 600 requests
Completed 800 requests
Completed 1000 requests
Completed 1200 requests
Completed 1400 requests
Completed 1600 requests
Completed 1800 requests
Completed 2000 requests
Finished 2000 requests


Server Software:        Apache/2.4.56
Server Hostname:        192.168.10.128
Server Port:            80

Document Path:          /cth.php
Document Length:        650 bytes

Concurrency Level:      1000
Time taken for tests:   1.124 seconds
Complete requests:      2000
Failed requests:        0
Total transferred:      2018000 bytes
HTML transferred:       1300000 bytes
Requests per second:    1779.29 [#/sec] (mean)
Time per request:       562.021 [ms] (mean)
Time per request:       0.562 [ms] (mean, across all concurrent requests)
Transfer rate:          1753.23 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   16  15.6     13      40
Processing:     7  336 325.2    197    1081
Waiting:        1  336 325.5    196    1081
Total:         41  352 335.2    204    1119
```

Figure 4-42 Figure of Stress Test 2

Here where the test 2 made using command line ab -n 2000 -c 1000 “localhost/Stock_Management/cth.php”. Concurrency level here is 1000 and request for 2000 users. The tool also reports various statistics about the requests, such as the total number of bytes transferred (2018000), the number of bytes consisting of HTML content (1300000), and the average number of requests completed per second (1779.29). It also provides detailed information about the time taken for each request, including the time taken to establish a connection (Connect), the time spent processing the request (Processing), the time spent waiting for a response (Waiting), and the total time taken (Total).

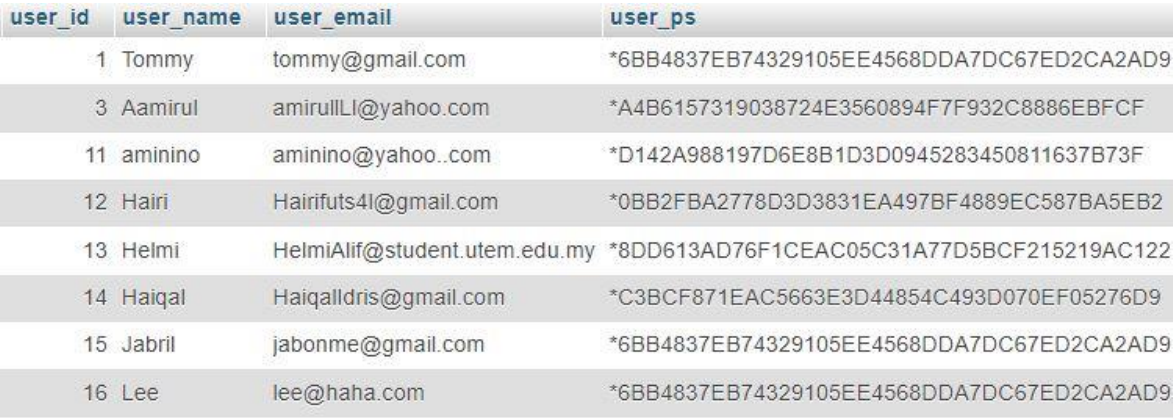
The output also includes a table showing the percentage of requests completed within certain time intervals, ranging from 13 ms (the minimum time taken for any request) to 1081 ms (the maximum time taken for any request).

Over the course of 2000 requests, the server demonstrated exceptional reliability with zero failed requests. The average time per request was 80.647 milliseconds, showcasing efficient processing. The data transfer rate reached 2443.63 Kbytes/sec, resulting in a total data transfer of 2018000 bytes. Impressively, the server achieved a processing speed of 2479.95 requests per second.

To further enhance the server's performance, various optimization strategies can be considered. Implementing load balancing, caching mechanisms, database optimization, code refinement, and hardware upgrades are potential avenues for improvement. It's crucial to acknowledge that optimizing a web server involves a holistic approach, considering factors like scalability, security, and user experience. While this analysis provides valuable insights, a thorough examination of server logs and consultation with web services experts may offer a more comprehensive understanding of performance dynamics.

4.3.2 Hashing technique

Hashing is a process used to convert plaintext data, such as passwords or other sensitive information, into a fixed-size string of characters, typically a hash value. The primary purpose of hashing is to provide a secure and irreversible transformation of data, making it difficult for attackers to retrieve the original information.



user_id	user_name	user_email	user_ps
1	Tommy	tommy@gmail.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
3	Aamirul	amirullLI@yahoo.com	*A4B6157319038724E3560894F7F932C8886EBFCF
11	aminino	aminino@yahoo..com	*D142A988197D6E8B1D3D0945283450811637B73F
12	Hairi	Hairifuts4I@gmail.com	*0BB2FBA2778D3D3831EA497BF4889EC587BA5EB2
13	Helmi	HelmiAlif@student.utem.edu.my	*8DD613AD76F1CEAC05C31A77D5BCF215219AC122
14	Haiqal	Haiqalldris@gmail.com	*C3BCF871EAC5663E3D44854C493D070EF05276D9
15	Jabril	jabonme@gmail.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
16	Lee	lee@haha.com	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9

Figure 4-43 Figure of Table in phpMyAdmin

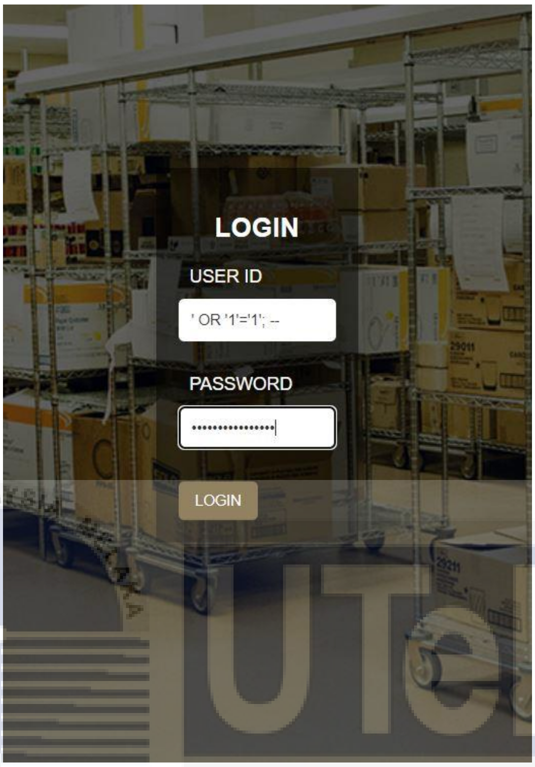
In secure database storage practices, user passwords are not stored as plaintext but rather in a hashed form. This fundamental approach ensures that the actual passwords are not directly visible or accessible within the database, enhancing overall data security. The process involves employing cryptographic hash functions to convert user passwords into fixed-size hash values, introducing an element of irreversibility. As a result, even if unauthorized access to the database occurs, attackers cannot readily obtain the original plaintext passwords. Instead, they would only see the hashed representations, which are computationally challenging to reverse.

To further fortify this security measure, a random value known as a "salt" is often incorporated before hashing each user's password. This additional layer of complexity ensures that identical passwords yield different hash values due to the unique salts, thereby thwarting precomputed table attacks. In practical terms, when a user creates or updates their password, the system applies the hash function along with a unique salt, generating a hashed value that is stored in the database.

During the authentication process, when a user attempts to log in, the entered password undergoes the same hashing process, and the resulting hash is compared with the stored hash in the database. If there is a match, the entered password is validated, allowing for a secure and effective means of user authentication while maintaining the confidentiality of sensitive information.

4.3.3 SQL Injection

SQL injection is a technique that can be used to attack data-driven applications. This technique involves inserting malicious SQL statements into an entry field in order to bring about their execution. The actions in question have the potential to result in the unauthorised access to confidential information, the manipulation of already existing information, difficulties in verifying the origin of the actions, the complete disclosure of data, and the destruction of data. SQL injection attacks can be mitigated by implementing fundamental measures such as properly escaping special characters in SQL queries, employing parameterized statements, and restricting database permissions for web application users. These measures together can help prevent SQL injection attacks. The identification and resolution of SQL injection vulnerabilities can be accomplished successfully through the use of automated web application security scanners as well as manual code reviews.



LOGIN

USER ID

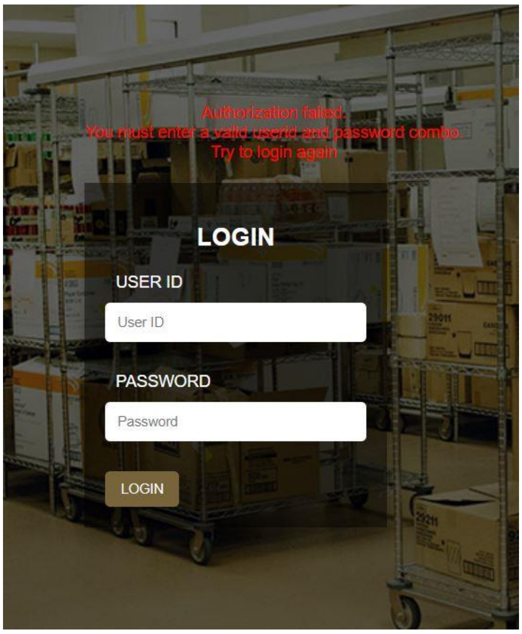
' OR '1'='1'; --

PASSWORD

••••••••••••••••

Figure 4-44 Figure of SQL Injection

To perform SQL injection, a common technique is to use a condition that will always be true, such as ' OR 1=1 --', which would change the query to 'SELECT * FROM user WHERE user_id = 'USER ID' OR 1=1 --' AND user_ps = 'password''.



Authorization failed.
You must enter a valid userid and password combo.
Try to login again

Figure 4-45 Figure of after SQL Injection

The system is safeguarded against SQL injection by utilising MySQL functions, which effectively prevent unauthorised manipulation of queries through user input.

4.4 Summary

The project "Implementing a Stock Management System Using MySQL, phpMyAdmin, HTML, and CSS" displays the system's implementation results in the "Preliminary Results and Discussion" section. It contains information about stock tracking, inventory updates, and transaction data. The section assesses the system's performance, functionality, and usability. The debate focuses on how to enhance stock management systems, decrease errors, and increase efficiency. The difficulties found during implementation were discussed, and potential solutions will be proposed. The developments will also examine the usefulness and aesthetics of the HTML and CSS-created user interface. The section thoroughly evaluates the stock management system's performance and offers suggestions for future improvements and here assumption can be made that the system is reliable after stress test, hashing technique and SQL injection.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The existing manual stock counting system in the Food and Beverages (F&B) sector presents significant challenges, including human errors in calculations, increased workload during peak hours, and the risk of misplaced records. These issues can lead to inefficiencies, staff shortages, and potential losses for the company, particularly when First In First Out (FIFO) practices are not consistently applied. To address these challenges, a proposed automated stock management system aims to streamline operations and improve accuracy in inventory tracking. By allowing staff to log daily stock movements digitally, the system eliminates the need for manual counting during busy periods, reduces the risk of misplaced records, and promotes FIFO practices. Additionally, the system contributes to waste reduction by eliminating the need for paper logbooks, supporting environmental sustainability. The project's objectives include conducting a literature study on industry stock management practices, developing a system tailored to the F&B sector's needs, and analyzing the reliability of the proposed stock management system. All the objectives were achieved in Chapter 4. Overall, the project seeks to enhance storeroom management in the F&B sector, offering a more efficient and sustainable solution to inventory challenges.

The project achieved its main goals:

First, a detailed literature study examined industry inventory and stock management. This required a thorough examination of inventory control methods, technologies, and practices.

Second, a Stock Management System was created. This system improves storeroom management with system.

The project concluded with a thorough analysis of the stock management system's reliability. The system's ability to streamline operations, and its ability to provide timely inventory information.

The project contributed to inventory management knowledge and provided a tangible solution to improve storeroom management efficiency and reliability for Food and Beverages businesses.

5.2 Future Works

In the work ahead, the primary objectives are to develop with PHP, integrate the tools, and bring the system to its conclusion. To accomplish this, analyze must be taken, code using PHP, set up the development environment, and adhere to best practises. Integration between the tools will be established, and extensive testing will be carried out to guarantee that communication and data exchange will go smoothly. Reviewing the system's requirements, performing testing, obtaining user feedback, documenting the process, and getting ready for deployment are all necessary steps in completing the system. The successful completion of future work depends critically on efficient management and communication of projects. Here are some future work that should be done in future:

- Barcode scanning

  Future work involves implementing barcode scanning functionality to enhance the efficiency of the stock management system. This feature will allow users to quickly and accurately input product information by scanning barcodes, reducing manual data entry errors and streamlining the overall inventory management process.

- Mobile App development

  Expanding the stock management system through mobile app development is a key focus. This initiative aims to provide users with greater flexibility and accessibility by enabling them to manage inventory, track stock levels, and input data directly from their mobile devices. The mobile app will offer a user-friendly interface, ensuring convenience and real-time updates on stock status.

- Security enhancements

  In the future, the stock management system will undergo security enhancements to safeguard sensitive inventory data. This involves implementing advanced encryption protocols, user authentication mechanisms, and access controls. The goal is to ensure the confidentiality, integrity, and availability of the stored information, protecting it from unauthorized access or data breaches.

- Report and analytics

  Enhancing the system's reporting and analytics capabilities is a crucial aspect of future development. This includes the integration of comprehensive reporting tools that generate insightful analytics on stock levels, turnover rates, and other relevant metrics. Users will benefit from customizable reports, facilitating informed decision-making and strategic planning based on the system's data analysis features.

- Add notifications

  Adding notifications to alert management when an item is below a certain quantity in the store involves implementing a system that monitors inventory levels and sends notifications when thresholds are reached.

- RFID

  RFID technology revolutionizes stock management systems by embedding unique identifiers in inventory items through RFID tags. As items equipped with RFID tags move within the read range of strategically placed RFID readers, real-time data capture occurs, updating the stock management system instantly. This not only eliminates manual data entry errors but also enables accurate, automated tracking of inventory levels, locations, and movements. RFID enhances efficiency by allowing bulk reading of multiple tags simultaneously, providing increased accuracy and visibility. The technology also facilitates batch and expiry tracking, crucial for managing perishable or batch-sensitive items. The seamless integration of RFID data into stock management software ensures timely decision-making, reduces stockouts, and enhances overall inventory control.

The presentation of suggested recommendations offers valuable insights for prospective considerations and strategic planning. These recommendations encapsulate informed perspectives that can serve as a foundation for future decision-making processes. The consideration of these recommendations is pivotal for enhancing the comprehensiveness and efficacy of future initiatives. The nuanced insights provided aim to contribute to a thoughtful and well-informed approach in addressing challenges or pursuing opportunities that may arise in subsequent endeavors.

REFERENCES

[1] magenest, "TOP 11 CHALLENGES IN INVENTORY MANAGEMENT: PROBLEMS AND SOLUTIONS." https://magenest.com/en/inventory-management-challenges/

[2] OPPOTUS, "Revealing Malaysia's Biggest Food Trends," 2020. https://www.oppotus.com/revealing-malaysias-food-trends/

[3] FASTJOBS, "6 COMMON PROBLEMS FACED BY F&B STAFF AND HOW TO SOLVE THEM." https://blog.fastjobs.sg/6-common-problems-faced-by-fnb-staff-and-how-to-solve-them

[4] Abby Jenkins, "20 Inventory Management Challenges and Solutions for 2022 and Beyond." https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management-challenges.shtml

[5] F. Y. A. Rahman, N. Kasuan, M. R. Dzulkifli, E. H. M. Saat, M. F. Dahari, and M. A. Saidi, "Development of FKE UiTM Kampus Pasir Gudang lab equipment web database," *2016 7th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2016 - Proceeding*, no. August, pp. 57–61, 2017, doi: 10.1109/ICSGRC.2016.7813301.

[6] B. S. Wijaya, I. Yohansen, Jeremy, and T. Oktavia, "Fridge inventory management to reduce household food wastage," *Proc. 2020 Int. Conf. Inf. Manag. Technol. ICIMTech 2020*, no. August, pp. 153–158, 2020, doi: 10.1109/ICIMTech50083.2020.9211142.

[7] S. Rezwan, W. Ahmed, M. A. Mahia, and M. R. Islam, "IoT Based Smart Inventory Management System for Kitchen Using Weight Sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi Module with Website and App," *Proc. - 2018 4th Int. Conf. Adv. Comput. Commun. Autom. ICACCA 2018*, 2018, doi:

10.1109/ICACCAF.2018.8776761.

[8] S. Singh, K. Kunder, V. Jain, and V. Sagvekar, “Inventory Management System for Education Institutions,” *5th IEEE Int. Conf. Adv. Sci. Technol. ICAST 2022*, pp. 206–209, 2022, doi: 10.1109/ICAST55766.2022.10039520.

[9] R. Patil, S. Nema, and S. Kadam, “Radio frequency identification system for asset tracking and inventory management in hospitals,” *2017 4th Int. Conf. Signal Process. Integr. Networks, SPIN 2017*, pp. 188–192, 2017, doi: 10.1109/SPIN.2017.8049941.

[10] M. C. Si, Y. Hong Ding, X. Ma, P. H. Xue, J. Bin Huang, and Y. J. Ma, “Visual management system of intelligent factory tools based on odoo,” *Proc. - 2021 4th Int. Conf. Adv. Electron. Mater. Comput. Softw. Eng. AEMCSE 2021*, pp. 938–941, 2021, doi: 10.1109/AEMCSE51986.2021.00191.

[11] J. Kongsumran and W. Premchaiswadi, “Process Monitoring and Inventory Management in Plastic Injection Molding,” pp. 192–195, 2011.

[12] P. Octaviani and W. Ce, “Inventory placement mapping using bluetooth low energy beacon technology for warehouses,” *Proc. 2020 Int. Conf. Inf. Manag. Technol. ICIMTech 2020*, no. August, pp. 354–359, 2020, doi: 10.1109/ICIMTech50083.2020.9211206.

[13] V. Chokshi, M. Shah, P. Mehta, M. Vairale, S. Mehta, and K. Talele, “Implementation of RFID-based Lab Inventory System,” *IBSSC 2022 - IEEE Bombay Sect. Signat. Conf.*, pp. 1–5, 2022, doi: 10.1109/IBSSC56953.2022.10037518.

[14] Smitha and A. Aslekar, “IoT in Inventory Management,” *2022 Int. Conf. Decis. Aid Sci. Appl. DASA 2022*, pp. 1045–1050, 2022, doi: 10.1109/DASA54658.2022.9765137.

[15] Nikita Zubovich, “What is IoT (Internet of Things)? – Explained by SumatoSoft.” https://sumatosoft.com/blog/what-is-iot-internet-of-things-explained-by-sumatosoft

[16] V. Mathur, “IoT In Inventory Management: Impact and Benefits.” https://www.analyticssteps.com/blogs/iot-inventory-management-impact-and-benefits

[17] O. / NetSuite, “Benefits of Inventory Management and Inventory Management Systems.” https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management-benefits.shtml#:~:text=Inventory management helps companies identify,proper warning of a shortage.

[18] A. K, “What is XAMPP? and How to Install XAMPP?” https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/

[19] J. Dobry, “What is phpMyAdmin.” https://www.nexcess.net/help/what-is-phpmyadmin/

[20] Stackoverflow, “2022 Developer Survey.” https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof

[21] K. Chris, “What is PHP? The PHP Programming Language Meaning Explained.” https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/

[22] F. Moraes, “HTML For Beginners The Easy Way: Start Learning HTML & CSS Today.” https://html.com/

[23] J. Juliver, “The Ultimate CSS Tutorial for Beginner Programmers.” https://blog.hubspot.com/website/css-tutorial

APPENDICES

AuthClassed.php

```
<?php
// <$Id: AuthClassed.php,v 1.5 2002/12/12 12:25:11 robhanssen Exp $>

// let's define some constants for the database
define("__DEFAULTHOST__", "localhost");
define("__DEFAULTNAME__", "stock_management");
define("__DEFAULTUSER__", "root");
define("__DEFAULTPASS__", "");
define("__DEFAULTTABLE__", "user");
define("PAGESESSIONTIME",60*10);

// the default permission; since 'permission' is an int(2), all permissions are lower than 100.
define("__DEFAULTPERMISSION__", 100);

Class AuthClassed
{
   var $authUserid;          // string: the userid, a unique name in the database
   var $authUsername;        // string: the real name of the user
   var $authUserPass;        // string: password
   var $authUserPermission;  // int(2): user permission level
   var $LoggedOn;            // bool  : logged on or not
   var $PageLevel;           // int   : the auth level of the protected page

   var $Host;
   var $Name;
   var $Table;
   var $User;
   var $Pass;
   var $DBlink;
   var $errorStr;

   function __construct($PageLevel)//AuthClassed($PageLevel)
   {

      $Host="";
     $Name="";
     $Table="";
     $User="";
      $Pass="";

     // first do all the database stuff
     // all the $Host,$Name,User,$Pass and $Table vars could be stored in
     // a config file, or set in the function call below
```

```
       if (!$this->DBlink = $this->setDBInfo($Host,$Name,$User,$Pass,$Table)) echo die
("unable to use the database");
        if(session_status() == PHP_SESSION_NONE)
        {
        // then start the authentification
        session_start();
        }


        $this->authUserid = $this->setUserid();//dapatkan userid
        $this->LoggedOn = $this->setLoggedOn(); //samada dah masuk or belum
        $this->authUsername = $this->setUsername();//dapatkan usernamae dr SESSION
        if (!$this->LoggedOn) $this->authUserPass = $this->setUserPass(); //kalo tak
loggedon
        $this->authUserPermission = $this->setUserPerm(); //dapatkan permission level utk
user
        $this->PageLevel = $this->setPageLevel($PageLevel);// setkan Page level

        //print_r($_SESSION);
        //echo time();

    if($this->LoggedOn){
        if($this->isSessionTimeValid())
      {
         $_SESSION["authuserlogintime"]=time();
      }
      else
      {
            $this->_login();
      }

    }


        if (!$this->LoggedOn || !$this->isPermitted($this->PageLevel))
        //kalo tak loggedon or tak dibenarkan sbb page level
        {
           if ($this->getUserid() && ($this->getUserPass() || $this->isLoggedOn()))
              { $userinfo = $this->_authenticate();}
           else $this->_login();

           if (is_array($userinfo) && $userinfo[1] <= $this->PageLevel)
           {
              $this->authUsername = $userinfo[0];
              $this->authUserPermission = (int)$userinfo[1];
              $this->LoggedOn = TRUE;
              $this->_initSession();
           }
           else
```

```
          {
              $this->LoggedOn = FALSE;
              $this->errorStr = $this->_errorMsg();
              $this->_login();
          }
      }
  }


  function setDBInfo($host="",$name="", $user="", $pass="", $table="")
  {
      $this->Host  = ($host !="")  ? $host  : __DEFAULTHOST__;
      $this->Name  = ($name !="")  ? $name  : __DEFAULTNAME__;
      $this->User  = ($user !="")  ? $user  : __DEFAULTUSER__;
      $this->Pass  = ($pass !="")  ? $pass  : __DEFAULTPASS__;
      $this->Table = ($table!="")  ? $table : __DEFAULTTABLE__;
      $_connect = mysqli_connect($this->Host,$this->User,$this->Pass,$this->Name);
      //$_select  = mysql_select_db($this->Name,$_connect);
      return ($_connect );
  }


  function setUserid()
  {
      if (isset($_SESSION['authuserid'])) $authuserid = $_SESSION['authuserid'];
      else if (isset($_POST['authuserid'])) $authuserid = mysqli_real_escape_string($this-
>DBlink,htmlentities($_POST['authuserid']));
 else $authuserid = "";
      if (preg_match("/[][{}()*+?.\\^$|]/",$authuserid)) $authuserid = "";
      return $authuserid;
  }

  function getUserid()
  {
      if ($this->authUserid) return $this->authUserid; else return "Unknown";
  }

  function getConID()
  {
      if ($this->DBlink) return $this->DBlink; else return "Not Connected";
  }



  function setUsername()
  {
      if (isset($_SESSION['authusername'])) $authusername =
$_SESSION['authusername'];
      else $authusername = "";
      return $authusername;
```

```
    }

    function getUsername()
    {
       if ($this->authUsername) return $this->authUsername; else return "Unknown";
    }

    function setUserPass()
    {
       if (isset($_POST['authuserpass'])) $authuserpass = mysqli_real_escape_string($this-
>DBlink,htmlentities($_POST['authuserpass']));
       else $authuserpass = "";
       return $authuserpass;
    }

    function getUserPass()
    {
       if ($this->authUserPass) return true; else return false;
    }

    function setLoggedOn()
    {
       if (isset($_SESSION['authisset'])) $authisset = $_SESSION['authisset'];
       else $authisset = FALSE;
       return $authisset;
    }

    function setUserPerm()
    {
       if (isset($_SESSION['authuserperm'])) $authuserperm =
(int)$_SESSION['authuserperm'];
       else $authuserperm = __DEFAULTPERMISSION__;
       return $authuserperm;
    }

    function getUserPerm()
    {
      if ($this->authUserPermission) return $this->authUserPermission; else return
__DEFAULTPERMISSION__;
    }

    function setPageLevel($PageLevel)
    {
        return $PageLevel;
    }

    function getPageLevel()
    {
        if ($this->PageLevel) return $this->PageLevel; else return
__DEFAULTPERMISSION__;
```

```
    }


    function isLoggedOn()
    {
        if ($this->LoggedOn) return TRUE; else return FALSE;
    }

  function isSessionTimeValid()
    {
    //echo "<br>time now:".time(). " User login at:".$_SESSION["authuserlogintime"]."
Elapsed ".(time()-$_SESSION["authuserlogintime"])." ";
    $diff = (time()-$_SESSION["authuserlogintime"]);
      if($diff < PAGESESSIONTIME)
      {return TRUE; }
      else
      { return FALSE;}
    }

    function isPermitted($reqlevel,$key="")
    {
        if ($reqlevel && $key && $key == EXACT)
        {
            if ($this->authUserPermission == $reqlevel) $permission = 1; //TRUE;
            else $permission = 0; //FALSE;
        }
        else if ($reqlevel)
        {
            if ($this->authUserPermission <= $reqlevel) $permission = 1; //TRUE;
            else $permission = 0; //FALSE;
        }
        else $permission = 0; //FALSE;
        return $permission;
    }

    function Logout()
    {
    global $Style;
    foreach($_SESSION as $key => $value)
         unset($key);
    session_destroy();
   header("Location: cth.php");
    exit;
    }



    function _login()
    {
        global $PHP_SELF, $Style;
```

```
 if(isset($_SESSION["authuserid"])){unset($_SESSION["authuserid"]);}
   if(isset($_SESSION["authusername"])){unset($_SESSION["authusername"]);}
   if(isset($_SESSION["authuserisset"])){unset($_SESSION["authuserisset"]);}
   if(isset($_SESSION["authuserperm"])){unset($_SESSION["authuserperm"]);}

   if(session_status() == PHP_SESSION_ACTIVE)
   {
      session_destroy();
   }

   include_once("login.php");
exit;
}

function _authenticate()
{
     if ($this->LoggedOn)
        $query = "SELECT user_name,roles FROM $this->Table WHERE user_id = 
'$this->authUserid' AND roles <= $this->PageLevel";
     else
        $query = "SELECT user_name,roles,user_ps FROM user WHERE user_id = 
'$this->authUserid' and user_ps = password('$this->authUserPass') AND roles <= $this-
>PageLevel";

     $result = mysqli_query($this->DBlink,$query) or die("Query failed");
     if(mysqli_num_rows($result) == 0) return 0;
     else
     {
        $query_data = mysqli_fetch_row($result);
        return $query_data;
     }
}

function _errorMsg()
{
   global $PHP_SELF;
   $invalid = "<div align=\"center\">Authorization failed.<br>
            You must enter a valid userid and password combo.<br>
            Try to login again</div>";
   $lowpermission = "<div align=\"center\">Authorization failed.<br>
            You are not authorized to use this program.<br>
            The mininum required userlevel is ". $this->PageLevel."<br>
            Try to logon again</div>";

   if (!$this->isPermitted($this->PageLevel))
       $error = $lowpermission;
   else $error = $invalid;
   return $error;
}
```

```
    function _initSession()
    {
        $_SESSION["authuserid"] = $this->authUserid;
        $_SESSION["authusername"] = $this->authUsername;
        $_SESSION["authisset"] = $this->LoggedOn;
        $_SESSION["authuserperm"] = $this->authUserPermission;
        $_SESSION["authuserlogintime"] = time();
        //session_register("authuserid", "authusername","authisset","authuserperm");
    }

    function _endSession()
    {
        session_unregister("authuserid");
        session_unregister("authusername");
        session_unregister("authuserisset");
        session_unregister("authuserperm");
        session_destroy();
    }

}
?>
```

connectDB.php

```
<?php

//nak connect ke database
class connectDB
{
    private $connectionID;

    public function __construct() {
      $this->connection = $this->connectDB();
    }

    public function connectDB()
  {
     $usr = "root";
     $pwd = "";
     $host = "localhost";
     $db = "stock_management";

     $this->connectionID = mysqli_connect($host,$usr,$pwd,$db);
     if (!$this->connectionID) { echo("ERROR! \n"); }
     return $this->connectionID;

  }

  public function getConID()
  {
      if (!$this->connectionID) { echo("ERROR! \n"); }
    return $this->connectionID;
  }
}



?>
```

Login.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>

  <meta charset="utf-8">
  <title>Login</title>

  <link rel="stylesheet" href="style4.css?v=<?php echo time();?>">

</head>

<body>
  <div class="center">

    <font color =red><div class=divStyle width=200px>
    <?php echo $this->errorStr; ?> </font>

  <form action ="<?php echo $PHP_SELF ?>" method="post">
    <h2>LOGIN</h2>
    <?php if(isset($_GET['error'])){ ?>
      <p class="error"><?php echo $_GET['error']; ?></p>
    <?php } ?>
    <label>USER ID</label>
    <input type="text" name="authuserid" placeholder="User ID">
    <br>
    <label>PASSWORD</label>
    <input type="password" name="authuserpass" placeholder="Password">
    <br>

    <button type="submit">LOGIN</button>
</div>
  </form>

</body>
</html>
```

Cth.php

```
    <?php

  define("ADMIN",1);
  define("MANAGER",10);
  define("USER",100);


  require_once("AuthClassed.php");
  $auth = new AuthClassed(USER );

  include_once "connectDB.php";
  $conDB = new connectDB();

  $userid = $auth->getUserid();
  $conID = $conDB->getConID();

  //wajib tambah as authen 3-12

  $isi="";
  $details = "";
  $op = "";
  $lists = "";
  $state="";

  $namaScript = basename($_SERVER['SCRIPT_NAME']);

  if(isset($_GET['menu']))
  {
  switch ($_GET['menu'])
  {
    case "logout": $auth->Logout();
          break;
    case "users":if($auth->getUserPerm()==ADMIN){
          $isi = "<b>User Management</b><hr>";



 if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
            {
            //print_r($_POST);

            $user_name = $_POST['user_name'];

            $ema=mysqli_real_escape_string($conID,$_POST['user_email']);

            $pass=trim($_POST['user_ps']);

            $user_role = $_POST['user_role'];
```

```
        $userid=$_POST['user_id'];
          if ($user_name && $ema && $user_role && $pass )
          {
          $query = "UPDATE $auth->Table SET  user_name = '$user_name',
user_email ='$ema' , user_ps=password('$pass'), roles = '$user_role' WHERE user_id =
'$userid'";
          $result = mysqli_query($conID,$query) or die("Query Error");
          //echo " ".$query;
          }
          else if($user_name && $ema && $user_role)
          {
          $query = "UPDATE $auth->Table SET user_name = '$user_name',
user_email='$ema', roles = '$user_role' WHERE user_id = '$userid'";
          $result = mysqli_query($conID,$query) or die("Query Error");
          }
          else if ($ema )
          {
          $query = "UPDATE $auth->Table SET user_email ='$ema'  WHERE user_id
= '$userid'";
          $result = mysqli_query($conID,$query) or die("Query Error");
          }
          else{}

        //var_dump($_POST);


        }
       else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="deleteUser"))
        {
          $user_id = $_POST['user_id'];

          $query = "DELETE FROM user WHERE user_id = '$user_id'";
          $result = mysqli_query($conID,$query) or die("Query Error");

          //var_dump($_POST);
        }
       else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="addUser"))
        {
          $user_name = $_POST['user_name'];
          $user_email = $_POST['user_email'];
          $pass = $_POST['user_ps'];
          $role = $_POST['user_role'];



          $query = "INSERT INTO user (user_name, user_email,user_ps, roles)
          VALUES ('$user_name', '$user_email', password('$pass'), '$role')";
          $result = mysqli_query($conID,$query) or die("Query Error");
```

```
           //var_dump($_POST);
         }


       function viewUsers($conID) {


         $details = "
         <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css'>
         <style>
           .styled-table {
             border-collapse: collapse;
             width: 100%;
             margin-bottom: 20px;
             font-family: 'Open Sans', sans-serif; /* Improved font choice */
           }

           .styled-table th, .styled-table td {
             border: 1px solid #ddd;
             padding: 12px 15px; /* Increased padding for better readability */
             text-align: left;
           }

           .styled-table th {
             background-color: #838011;
             color: #fff;
           }

           .styled-table tr:nth-child(even) {
             background-color: #f2f2f2;
           }

           .styled-table tr:hover {
             background-color: #ddd;
             cursor: pointer; /* Indicate interactivity on hover */
           }

           .styled-table td a {
             text-decoration: none; /* Remove underline from links */
             color: #000; /* Color for actions */
           }

           .action-buttons {
             display: flex;
             justify-content: space-around;
           }

           .action-buttons button {
```

```
   padding: 8px 12px;
   margin: 0 5px;
   cursor: pointer;
   border-radius: 5px; /* Rounded corners for buttons */
  }

  .add-button {
   background-color: #5d00ff;
   color: #fff;
   border: none;
  }


</style>



    <table class='styled-table'>
        <tr>
           <th>Staff ID</th>
           <th>Staff Name</th>
           <th>Staff Email</th>
           <th>Roles</th>
           <th>Action</th>
        </tr>";



$query = "SELECT user_id, user_name, user_email, roles FROM user ;";
$result = mysqli_query($conID, $query) or die(mysqli_error($conID));
//$roleslist .= "<option value='$roles'>$roleslistt</option>";

while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {


$roles = $row['roles'];
  if ($roles == 1) {
     $role = 'Admin';
  } else if ($roles == 10) {
     $role = 'Manager';
  } else if ($roles == 100) {
     $role = 'Staff';
  } else {
     $role = 'Unknown';
  }

   $details .= "<tr>
                <td>{$row['user_id']}</td>
                <td>{$row['user_name']}</td>
                <td>{$row['user_email']}</td>
```

```
                              <td>{$role}</td>
                              <td>
                                 <button type='button'><a
href='?menu=users&operation=edit&user_id={$row['user_id']}' class='edit-button'><i
class='fas fa-pencil'></i></a></button>
                                 <button type='button'><a
href='?menu=users&operation=delete&user_id={$row['user_id']}' class='delete-button'><i
class='fas fa-trash'></i></a>
                         </td>
                              </tr>";
                }

                $details .= "</table>";

                return $details;




            }



            function editUsers($user_id, $operation, $conID, $namaScript){



              $details = "";


              if($operation == "edit")
              {

                $query = "select user_name, user_email, user_id, roles from user WHERE
user_id = '".$user_id."'";
                $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                 while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
                {$nama=$row['user_name'];$ema=$row['user_email'];$pos=$row['roles'];}

                $details = "



                    <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                        <h3 style='color: #333;'>Update Details</h3>
                        <form action='$namaScript?menu=users' method='POST'>
                            <input type='hidden' name='user_id' value='$user_id'>
                            <table style='width: 100%;'>
                                <tr>
                                    <td>User Name</td>
                    <td>:</td>
                   <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' value='$nama' name='user_name' /></td>
                  </tr>
                  <tr>
                    <td>E-mail</td>
                    <td>:</td>
                   <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' value='$ema' name='user_email' /></td>
                  </tr>
                  <tr>
                    <td>Password</td>
                    <td>:</td>
                    <td><input type='password' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='user_ps' /></td>
                  </tr>
                  <tr>
                    <td>Role</td>
                      <td>:</td>
                      <td>
                        <select name='user_role'>
                          <option value='1' " . ($pos == 1 ? 'selected' : '') .
"'>Admin</option>
                          <option value='10' " . ($pos == 10 ? 'selected' : '') .
"'>Manager</option>
                          <option value='100' " . ($pos == 100 ? 'selected' : '') .
"'>Staff</option>
                        </select>
                      </td>
                  </tr>
                  <tr>
                    <td align='center' colspan='3'>
                      <font color='red'></font>
                      <input type='hidden' name='formsubmitted'
value='updateDetails'>
                     <input type='submit' value='Update' style='margin-right: 10px;
padding: 8px 16px; background-color: #3498db; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                      <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                    </td>
                  </tr>
                </table>
              </form>
            </div>
            <hr>
          ";

        }
```

```
                elseif ($operation == "add")
                {
                   $query = "select user_name, user_email, user_id, roles from user WHERE
user_id = '".$user_id."'";
                 $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                 $pos = "";
                  while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
                 {$nama=$row['user_name'];$ema=$row['user_email'];$pos=$row['roles'];}


                   $details = "


                  <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                      <h3 style='color: #333;'>Add Staff</h3>
                      <form action='$namaScript?menu=users' method='POST'>
                          <input type='hidden' name='formsubmitted' value='addUser'>
                          <table style='width: 100%;'>
                              <tr>
                                  <td>User Name</td>
                                  <td>:</td>
                                 <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='user_name' /></td>
                              </tr>
                              <tr>
                                  <td>E-mail</td>
                                  <td>:</td>
                                 <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='user_email' /></td>
                              </tr>
                              <tr>
                                  <td>Password</td>
                                  <td>:</td>
                                  <td><input type='password' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='user_ps' /></td>
                              </tr>
                              <tr>
                                  <td>Role</td>
                                    <td>:</td>
                                    <td>
                                        <select name='user_role'>
                                            <option value='1' " . ($pos == 1 ? 'selected' : '') .
"'>Admin</option>
                                            <option value='10' " .($pos == 10 ? 'selected' : '') .
"'>Manager</option>
                                            <option value='100' " . ($pos == 100 ? 'selected' : '') .
"'>Staff</option>
                                        </select>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td align='center' colspan='3'>
                                            <font color='red'></font>
                                            <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                            <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                        </td>
                                    </tr>
                                </table>
                            </form>
                        </div>
                        <hr>
                    ";


                }
                else if($operation == "delete")
                {
                  $query = "select user_name from user WHERE user_id = '".$user_id."'";
                  $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                   while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
                  {$nama=$row['user_name'];}

                  $details = "

                    <div style='text-align: center;'>
                        <p style='font-size: 24px; color: #e74c3c; margin-bottom: 10px;'>Delete
User</p>
                        <p style='font-size: 18px; color: #333; margin-bottom: 20px;'>Are you
sure to delete User $nama from the system?</p>
                        <form action='$namaScript?menu=users' method='POST'>
                            <input type='hidden' name='user_id' value='$user_id'></input>
                            <input type='hidden' name='formsubmitted'
value='deleteUser'></input>
                            <div style='margin-top: 20px;'>
                               <button type='submit' style='background-color: #e74c3c; color: #fff;
padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size: 18px;
transition: background-color 0.3s;'>Yes</button>
                                <a href='?menu=users'>
                                    <button type='button' style='background-color: #3498db; color:
#fff; padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size:
18px; transition: background-color 0.3s;'>No</button>
                                </a>
                            </div>
                        </form>
                    </div>
                    <hr></hr>";



                }


                return $details;


              }


                $details .= "<button type='button'><a
href='?menu=users&operation=add'>ADD STAFF +</a></button> </br> <hr>";



                if(isset($_GET['operation']))
                {
                  switch($_GET['operation'])
                    {
                      case"edit" : $details .= editUsers($_GET['user_id'],"edit",$conID,
$namaScript); //namascript kenapa?
                            break;
                      case"add" : $details .= editUsers("","add",$conID, $namaScript);
                            break;
                      case"delete" : $details .= editUsers($_GET['user_id'],"delete",$conID,
"");

                            break;
                    }
                }

                $details .= "".viewUsers($conID);




              }
              else($isi="<font color=red>Sorry, You are not authorized to use this
function.</font>");

              break;
```

```
      case "profile": {

    if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
                {
                //print_r($_POST);

                $ema=mysqli_real_escape_string($conID,$_POST['user_email']);


                $pass=trim($_POST['user_ps']);

                $userid=$auth->getUserid();
                  if ($ema &&  $pass )
                  {
                  $query = "UPDATE $auth->Table SET  user_email ='$ema' ,
user_ps=password('$pass') WHERE user_id = '$userid'";
                  $result = mysqli_query($conID,$query) or die("Query Error");
                  //echo " ".$query;
                  }
                  else if ($ema )
                  {
                  $query = "UPDATE $auth->Table SET user_email ='$ema'  WHERE user_id
= '$userid'";
                  $result = mysqli_query($conID,$query) or die("Query Error");
                  }
                  else{}


                }

              $isi .= "<b>User Profile</b><hr>";
              $query = "select user_name, user_email, user_id, roles from $auth->Table
WHERE user_id = '".$auth->getUserid()."'";
              $result = mysqli_query($conID, $query) or die(mysqli_error($conID));

              while ($row =  mysqli_fetch_array($result, MYSQLI_ASSOC)) {
                 $nama = $row['user_name'];
                 $ema = $row['user_email'];
                 $pos = $row['roles'];

                 $details = "
                    <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                       <div style='margin-bottom: 15px;'>
                          <div style='font-weight: bold;'>Nama:</div>
                          <div>$nama</div>
                       </div>

                       <div style='margin-bottom: 15px;'>
                          <div style='font-weight: bold;'>Email:</div>
```

```
                         <div>$ema</div>
                      </div>

                      <div style='margin-bottom: 15px;'>
                         <div style='font-weight: bold;'>Position:</div>";

                if ($pos == 1) {
                   $pos = 'Admin';
                } elseif ($pos == 10) {
                   $pos = 'Manager';
                } elseif ($pos == 100) {
                   $pos = 'Staff';
                } else {
                   $pos = 'Unknown';
                }

                $details .= "<div>$pos</div>";

                $details .= "
                      </div>

                      <div>
                         <a href='cth.php?menu=profile&operation=update' style='text-
decoration: none; color: #3498db; font-weight: bold;'>
                            Click Here to Update Your Details
                         </a>
                      </div>
                   </div>";
             }
```

```
             if(isset($_GET['operation']))
             {
               switch ($_GET['operation'])
                  {
                  case "": /*
                     $qphoto = "select photo from $auth->Table WHERE userid = '$userid'";
                     $result = mysqli_query($conID,$qphoto) or die("Query Error");
                     while($row =
mysqli_fetch_array($result,MYSQLI_ASSOC)){$userphoto = $row['photo'];}
                     $userphoto = "files/".$userphoto;
                     $mysock = getimagesize($userphoto);
                          if ($userphoto == "files/"){$details= "This is your
details<br><br>Sorry, your photo is not available yet";}
                          else{ $details = "This is your details<br><br><a href='$userphoto'
target='_blank'><img src='$userphoto' ".imageResize($mysock[0],$mysock[1],
250)."></a>"; }
                     */
```

```
                    break;
                case "update":  $userid=$auth->getUserid();
                        $query = "select user_email, user_name from $auth->Table WHERE
user_id = '$userid' limit 1";
                        $result = mysqli_query($conID,$query) or die(mysqli_error($conID));
                          $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
                          $ema=$row['user_email'];

                         $details = "
                           <div style='width: 50%; margin: 20px auto; background-color:
#f9f9f9; padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                               <h2 style='color: #333;'>Update Your Details</h2>
                               <form action='$namaScript?menu=profile' method='POST'>
                                   <table style='width: 100%;'>
                                       <tr>
                                           <td>E-mail</td>
                                           <td>:</td>
                                          <td><input type='text' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' value='$ema' name='user_email' /></td>
                                       </tr>
                                       <tr>
                                           <td>Password</td>
                                           <td>:</td>
                                          <td><input type='password' style='width: 100%; padding:
8px; border: 1px solid #ccc; border-radius: 4px;' name='user_ps' /></td>
                                       </tr>
                                       <tr>
                                           <td align='center' colspan='3'>
                                               <font color='red'></font>
                                               <input type='hidden' name='formsubmitted'
value='updateDetails'>
                                               <input type='submit' value='Update' style='margin-
right: 10px; padding: 8px 16px; background-color: #3498db; color: #fff; border: none;
border-radius: 4px; cursor: pointer;' />
                                               <input type='reset' value='Clear' style='padding: 8px
16px; background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                           </td>
                                       </tr>
                                   </table>
                               </form>
                           </div>
                        ";




                        break;

                }
```

```
              }

            }
            break;

    case "order":if($auth->getUserPerm()<=MANAGER){
            $isi = "<b>Order</b><hr>";




  if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="updateDetails"))
              {
              //print_r($_POST);

              $ordID = $_POST['order_ID'];

              $ordNM = $_POST['order_name'];

              $ordQTT = $_POST['order_qntt'];

              $ordDT = $_POST['order_date'];

              $ordID=$_POST['order_ID'];
                if ($ordID && $ordNM && $ordQTT && $ordDT )
                {
                $query = "UPDATE orderr SET  order_ID = '$ordID', order_name ='$ordNM'
, order_qntt=('$ordQTT'), order_date = '$ordDT' WHERE order_ID = '$ordID'";
                $result = mysqli_query($conID,$query) or die("Query Error");
                //echo " ".$query;
                }
                else if($ordNM && $ordQTT && $ordDT)
                {
                $query = "UPDATE $auth->Table SET order_name ='$ordNM' ,
order_qntt=('$ordQTT'), order_date = '$orderDT' WHERE order_ID = '$ordID'";
                $result = mysqli_query($conID,$query) or die("Query Error");
                }
                else if ($ordQTT )
                {
                $query = "UPDATE $auth->Table SET order_qntt=('$ordQTT'),  WHERE
order_ID = '$ordID'";
                $result = mysqli_query($conID,$query) or die("Query Error");
                }
```

```
                  else{}

                //var_dump($_POST);


                }
              else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="deleteOrder"))
                {
                  $ordID = $_POST['order_ID'];

                  $query = "DELETE FROM orderr WHERE order_ID = '$ordID'";
                  $result = mysqli_query($conID,$query) or die("Query Error");

                  //var_dump($_POST);
                }
              else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="addOrder"))
                {

                  $ordNM = $_POST['order_name'];
                  $ordQTT = $_POST['order_qntt'];
                  $ordDT = $_POST['order_date'];


                  $query = "INSERT INTO orderr (order_name,order_qntt, order_date)
                  VALUES ('$ordNM', '$ordQTT', '$ordDT')";
                  $result = mysqli_query($conID,$query) or die("Query Error");

                  //var_dump($_POST);
                }


                function editOrder($order_ID, $operation, $conID, $namaScript){


                $details = "";


                if($operation == "edit")
                {

                  $query = "select order_name, order_ID, order_qntt, order_date from orderr
WHERE order_ID = '".$order_ID."'";
                  $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                   while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))

  {$ordNM=$row['order_name'];$ordQTT=$row['order_qntt'];$ordDT=$row['order_date'];
}
```

```
                $details = "
                  <hr>

                  <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                      <h3 style='color: #333;'>Update Details</h3>
                      <form action='$namaScript?menu=order' method='POST'>
                          <input type='hidden' name='order_ID' value='$order_ID'>
                          <table style='width: 100%;'>
                              <tr>
                                  <td>Item Name</td>
                                  <td>:</td>
                                 <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' value='$ordNM' name='order_name' /></td>
                              </tr>
                              <tr>
                                  <td>Quantity</td>
                                  <td>:</td>
                                  <td><input type='number' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' value='$ordQTT' name='order_qntt' min=0
/></td>
                              </tr>
                              <tr>
                                  <td>Date</td>
                                  <td>:</td>
                                 <td><input type='date' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' value='$ordDT' name='order_date' /></td>
                              </tr>
                              <tr>
                                  <td align='center' colspan='3'>
                                      <font color='red'></font>
                                      <input type='hidden' name='formsubmitted'
value='updateDetails'>
                                    <input type='submit' value='Update' style='margin-right: 10px;
padding: 8px 16px; background-color: #3498db; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                      <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                  </td>
                              </tr>
                          </table>
                      </form>
                  </div>
              ";

              }

                elseif ($operation == "add")
                {
                    $query = "select order_name, order_ID, order_qntt, order_date from orderr
WHERE order_ID = '".$order_ID."'";
                  $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                  while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))

  {$ordNM=$row['order_name'];$ordQTT=$row['order_qntt'];$ordDT=$row['order_date'];
}



                    $details = "
                    <hr>

                    <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                        <h3 style='color: #333;'>Add Order</h3>
                        <form action='$namaScript?menu=order' method='POST'>
                            <input type='hidden' name='formsubmitted' value='addOrder'>
                            <table style='width: 100%;'>
                                <tr>
                                    <td>Order Name</td>
                                    <td>:</td>
                                   <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='order_name' /></td>
                                </tr>
                                <tr>
                                    <td>Quantity</td>
                                    <td>:</td>
                                   <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='order_qntt' /></td>
                                </tr>
                                <tr>
                                    <td>Date</td>
                                    <td>:</td>
                                    <td><input type='date' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='order_date' /></td>
                                </tr>
                                <tr>
                                    <td align='center' colspan='3'>
                                        <font color='red'></font>
                                        <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                        <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                    </td>
                                </tr>
                            </table>
                        </form>
                    </div>
                    ";


                    }
                  else if($operation == "delete")
                  {
                    $query = "select order_name from orderr WHERE order_ID =
'".$order_ID."'";
                    $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                     while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
                    {$ordNM=$row['order_name'];}

                    $details = "
                      <hr></hr>
                      <div style='text-align: center;'>
                          <p style='font-size: 24px; color: #e74c3c; margin-bottom: 10px;'>Delete
Order</p>
                          <p style='font-size: 18px; color: #333; margin-bottom: 20px;'>Are you
sure to delete order $ordNM from the system?</p>
                          <form action='$namaScript?menu=order' method='POST'>
                              <input type='hidden' name='order_ID' value='$order_ID'></input>
                              <input type='hidden' name='formsubmitted'
value='deleteOrder'></input>
                              <div style='margin-top: 20px;'>
                                  <button type='submit' style='background-color: #e74c3c; color: #fff;
padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size: 18px;
transition: background-color 0.3s;'>Yes</button>
                                  <a href='?menu=order'>
                                      <button type='button' style='background-color: #3498db; color:
#fff; padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size:
18px; transition: background-color 0.3s;'>No</button>
                                  </a>
                              </div>
                          </form>
                      </div>";

                  }


                  return $details;


                }


                function viewOrder($conID){


                  $details = "
```

```
        <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css'>
        <style>
          .styled-table {
            border-collapse: collapse;
            width: 100%;
            margin-bottom: 20px;
          }

          .styled-table th, .styled-table td {
            border: 1px solid #ddd;
            padding: 10px;
            text-align: left;
          }

          .styled-table th {
            background-color: #838011;
            color: #fff;
          }

          .styled-table tr:nth-child(even) {
            background-color: #f2f2f2;
          }

          .styled-table td a {
           text-decoration: none; /* Remove underline from links */
           color: #000; /* Color for actions */
          }

          .action-buttons {
           display: flex;
           justify-content: space-around;
          }

          .action-buttons button {
           padding: 8px 12px;
           margin: 0 5px;
           cursor: pointer;
           border-radius: 5px; /* Rounded corners for buttons */
          }

          .add-button {
           background-color: #5d00ff;
           color: #fff;
           border: none;

          }
        </style>

            <table class='styled-table'>
```

```
                    <tr>
                        <th>Order ID</th>
                        <th>Order Name</th>
                        <th>Order Quantity</th>
                        <th>Order date</th>
                        <th>Action</th>
                    </tr>";

            $query = "SELECT order_ID, order_name, order_qntt, order_date FROM
orderr ORDER BY order_date ;";
            $result = mysqli_query($conID, $query) or die(mysqli_error($conID));


            while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
            $formattedDate = date_create($row['order_date'])->format('d/M/Y');
                $details .= "<tr>
                            <td>{$row['order_ID']}</td>
                            <td>{$row['order_name']}</td>
                            <td>{$row['order_qntt']}</td>
                            <td>{$formattedDate}</td>
                            <td>
                            <button type='button' class='edit-button'><a
href='?menu=order&operation=edit&order_id={$row['order_ID']}'><i class='fas fa-pencil-
alt'></i></a></button>
                        <button type='button' class='delete-button'><a
href='?menu=order&operation=delete&order_id={$row['order_ID']}'><i class='fas fa-
trash'></i></a></button>
                            </td>
                        </tr>";
            }

            $details .= "</table>";

            return $details;




        }



            $details .= "<button type='button'><a
href='?menu=order&operation=add'>ADD ORDER</a></button>";




            if(isset($_GET['operation']))
            {
              switch($_GET['operation'])
                  {
                        case"edit" : $details .= editOrder($_GET['order_id'],"edit",$conID,
$namaScript);
                              break;
                        case"add" : $details .= editOrder("","add",$conID, $namaScript);
                              break;
                        case"delete" : $details .= editOrder($_GET['order_id'],"delete",$conID,
"");
                              break;
                      }
                }


                $details .= "".viewOrder($conID);

              }
              else($isi="<font color=red>Sorry, You are not authorized to use this
function.</font>");




              break;



      case "stock":if($auth->getUserPerm()==ADMIN||MANAGER || USER){
              $isi = "<b>Stock Available </b><hr>";

              if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="editStock"))
                {
                //print_r($_POST);

                $stkID = $_POST['stock_id'];

                $stkQTT = $_POST['stock_qntt'];

                $stkDT = $_POST['stock_date'];

                  if ($stkID && $stkQTT && $stkDT )
                  {
                  $query = "UPDATE stock SET stock_qntt=('$stkQTT'), stock_date =
'$stkDT' WHERE stock_id = '$stkID'";
                  $result = mysqli_query($conID,$query) or die("Query Error");
                  //echo " ".$query;
                  }
                  else if($stkQTT && $stkDT)
                  {
                  $query = "UPDATE $auth->Table SET stk_qntt=('$ordQTT'), stock_date =
'$stkDT' WHERE stock_id = '$stkID'";
                  $result = mysqli_query($conID,$query) or die("Query Error");
```

```
                 }
                 else if ($stkQTT )
                 {
                 $query = "UPDATE $auth->Table SET stk_qntt=('$stkQTT'),  WHERE
stock_id = '$stkID'";
                 $result = mysqli_query($conID,$query) or die("Query Error");
                 }
                 else{}

                //var_dump($_POST);



                }
              else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="deleteStock"))
                {
                 $stkID = $_POST['stock_ID'];

                 $query = "DELETE FROM stock WHERE stock_id = '$stkID'";
                 $result = mysqli_query($conID,$query) or die("Query Error");

                 //var_dump($_POST);
                }
              else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="addStock"))
                {

                 $stkID = $_POST['namestock'];
                 $stkQTT = $_POST['stock_qntt'];
                 $stkDT = $_POST['stock_date'];
                 $rmk = $_POST['remark'];


                 $query = "INSERT INTO stock (stocktype_id,stock_qntt, stock_date,remark)
                 VALUES ('$stkID', '$stkQTT', '$stkDT','$rmk')";
                 $result = mysqli_query($conID,$query) or die("Query Error");

                 //var_dump($_POST);
                }

                else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="minusStock"))
                {

                 $stkID = $_POST['namestock'];
                 $stkQTT = $_POST['stock_qntt'];
                 $stkDT = $_POST['stock_date'];
                 $rmk = $_POST['remark'];
```

```
          $query = "INSERT INTO stock (stocktype_id,stock_qntt, stock_date,remark)
          VALUES ('$stkID', '$stkQTT', '$stkDT','$rmk')";
          $result = mysqli_query($conID,$query) or die("Query Error");

        }

        function editStock($stock_id, $operation, $conID, $namaScript){


        $details = "";

        $namestocklist = "<select name='namestock' id='namestock'>";

        $query = "select * from stocktype order by namestock";
        $result = mysqli_query($conID,$query) or die(mysqli_error($conID));

        while($row = mysqli_fetch_array($result,MYSQLI_ASSOC)){

          $stocktype_id = $row['stocktype_id'];
          $namestock = $row['namestock'];

          $namestocklist .= "<option value='$stocktype_id'>$namestock</option>";
        }

        $namestocklist .= "</select>";


        if($operation == "edit")
        {

          $query = "SELECT stock.stock_id, stocktype.namestock, stock.stock_qntt,
stock.stock_date FROM stock INNER JOIN stocktype ON
stock.stocktype_id=stocktype.stocktype_id WHERE stock_id = '".$stock_id."'";
          $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
           while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
          {$stkName = $row['namestock'];$stkQTT = $row['stock_qntt'];$stkDT =
$row['stock_date'];}

          $details = "



            <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
              <h3 style='color: #333;'>Update Stock</h3>
              <form action='$namaScript?menu=stock' method='POST'>
                <input type='hidden' name='formsubmitted' value='editStock'>
                <input type='hidden' name='stock_id' value=$stock_id >
                <table style='width: 100%;'>
```

```
                                <tr>
                                    <td>Quantity</td>
                                    <td>:</td>
                                    <td><input type='number' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='stock_qntt'min=0 /></td>
                                </tr>
                                <tr>
                                    <td>Date</td>
                                    <td>:</td>
                                   <td><input type='date' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='stock_date' /></td>
                                </tr>
                                <tr>
                                    <td align='center' colspan='3'>
                                        <font color='red'></font>
                                        <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                        <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                    </td>
                                </tr>
                            </table>
                        </form>
                    </div>
                    <hr>
                ";

                  }
                elseif ($operation == "add")
                  {

                $details = "



                  <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                      <h3 style='color: #333;'>Add Stock</h3>
                      <form action='$namaScript?menu=stock' method='POST'>
                          <input type='hidden' name='formsubmitted' value='addStock'>
                          <table style='width: 100%;'>
                              <tr>
                                  <td>Stock Name</td>
                                  <td>:</td>
                                  <td>$namestocklist</td>
                              </tr>
                              <tr>
                                  <td>Quantity</td>
                                  <td>:</td>
                                  <td><input type='number' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='stock_qntt' min='1' required /></td>
                              </tr>
                              <tr>
                                  <td>Date</td>
                                  <td>:</td>
                                  <td><input type='date' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='stock_date' /></td>
                              </tr>
                              <tr>
                                  <td>Remark</td>
                                  <td>:</td>
                                  <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='remark' /></td>
                              </tr>
                              <tr>
                                  <td align='center' colspan='3'>
                                      <font color='red'></font>
                                      <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                      <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                                  </td>
                              </tr>

                          </table>
                      </form>
                  </div>
                  <hr>
";


                  }
                else if($operation == "delete")
                {
                  $query = "select stock.stock_id, stocktype.namestock from stock inner join
stocktype on stock.stocktype_id=stocktype.stocktype_id WHERE stock_id =
'".$stock_id."'";
                  $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                  while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
                  {$stkID = $row['stock_id'];}
                  $details = "


                      <div style='text-align: center;'>
                          <p style='font-size: 24px; color: #e74c3c; margin-bottom: 10px;'>Delete
Stock</p>
```

```
                        <p style='font-size: 18px; color: #333; margin-bottom: 20px;'>Are you
sure to delete item from the system?</p>
                        <form action='$namaScript?menu=stock' method='POST'>
                            <input type='hidden' name='stock_ID' value='$stock_id'></input>
                            <input type='hidden' name='formsubmitted'
value='deleteStock'></input>
                            <div style='margin-top: 20px;'>
                                <button type='submit' style='background-color: #e74c3c; color:
#fff; padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size:
18px; transition: background-color 0.3s;'>Yes</button>
                                <a href='?menu=stock'>
                                    <button type='button' style='background-color: #3498db; color:
#fff; padding: 15px 30px; border: none; border-radius: 8px; cursor: pointer; font-size:
18px; transition: background-color 0.3s;'>No</button>
                                </a>
                            </div>
                        </form>
                    </div>
                    <hr>";


                }
                else if ($operation == 'minus')
                {
                  $details = "


                <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                    <h3 style='color: #333;'>Take Out Stock</h3>
                    <form action='$namaScript?menu=stock' method='POST'>
                        <input type='hidden' name='formsubmitted' value='minusStock'>
                        <table style='width: 100%;'>
                            <tr>
                                <td>Stock Name</td>
                                <td>:</td>
                                <td>$namestocklist</td>
                            </tr>
                            <tr>
                                <td>Quantity</td>
                                <td>:</td>
                                <td><input type='number' style='width: 100%; padding: 8px;
border: 1px solid #ccc; border-radius: 4px;' name='stock_qntt' max='-1' /></td>
                            </tr>
                            <tr>
                                <td>Date</td>
                                <td>:</td>
                                <td><input type='date' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='stock_date' /></td>
                            </tr>
                <tr>
                  <td>Remark</td>
                  <td>:</td>
                  <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='remark' /></td>
                </tr>
                <tr>
                  <td align='center' colspan='3'>
                    <font color='red'></font>
                    <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                    <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                  </td>
                </tr>
              </table>
            </form>
          </div>
          <hr>
        ";
         }

         return $details;

        }



        function viewStock($conID){



         $details = "<style>
           .styled-table {
             border-collapse: collapse;
             width: 100%;
             margin-bottom: 20px;
           }

           .styled-table th, .styled-table td {
             border: 1px solid #ddd;
             padding: 10px;
             text-align: left;
           }
```

```
  .styled-table th {
      background-color: #838011;
      color: #fff;
  }

  .styled-table tr:nth-child(even) {
      background-color: #f2f2f2;
  }

  .styled-table td a {
    text-decoration: none; /* Remove underline from links */
    color: #000; /* Color for actions */
  }

  .action-buttons {
    display: flex;
    justify-content: space-around;
  }

  .action-buttons button {
    padding: 8px 12px;
    margin: 0 5px;
    cursor: pointer;
    border-radius: 5px; /* Rounded corners for buttons */
  }

  .add-button {
    background-color: #5d00ff;
    color: #fff;
    border: none;
  }

  .styled-table .low-stock-alert {
  color: red; /* Text color for low-stock items */
 }

  }

</style>

     <table class='styled-table'>
        <tr>

           <th>Stock Name</th>
           <th>Stock In-store</th>
        </tr>";
```

```
                        $query = "SELECT stocktype.namestock, SUM(stock.stock_qntt) AS
stocksum FROM stock INNER JOIN stocktype ON
stock.stocktype_id=stocktype.stocktype_id GROUP BY stocktype.namestock ;";
                $result = mysqli_query($conID, $query) or die(mysqli_error($conID));

                while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
                $stockQuantity = $row['stocksum'];
                  $alertClass = $stockQuantity <= 10 ? 'low-stock-alert' : '';

                   $details .= "<tr>
                                <td>{$row['namestock']}</td>
                                <td class='$alertClass'>{$stockQuantity}</td>
                             </tr>";
                };

                $details .= "</table>";


               $details .= "
               <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css'>

               <style>
                  .styled-table {
                     border-collapse: collapse;
                     width: 100%;
                     margin-bottom: 20px;
                  }

                  .styled-table th, .styled-table td {
                     border: 1px solid #ddd;
                     padding: 10px;
                     text-align: left;
                  }

                  .styled-table th {
                     background-color: #838011;
                     color: #fff;
                  }

                  .styled-table tr:nth-child(even) {
                     background-color: #f2f2f2;
                  }

                  .styled-table td a {
                    text-decoration: none; /* Remove underline from links */
                    color: #000; /* Color for actions */
                  }

                  .action-buttons {
                    display: flex;d
                    justify-content: space-around;
                  }

                  .action-buttons button {
                    padding: 8px 12px;
                    margin: 0 5px;
                    cursor: pointer;
                    border-radius: 5px; /* Rounded corners for buttons */
                  }

                  .add-button {
                    background-color: #5d00ff;
                    color: #fff;
                    border: none;
                  }

                  }
                </style>
                <table class='styled-table'>
                        <tr>
                            <th>Stock ID</th>
                            <th>Stock Name</th>
                            <th>Stock Quantity</th>
                            <th>Date</th>
                            <th>Remark</th>
                            <th>Action</th>
                        </tr>";

                $query = "SELECT stock.stock_id, stocktype.namestock, stock.stock_qntt,
stock.stock_date , stock.remark FROM stock INNER JOIN stocktype ON
stock.stocktype_id=stocktype.stocktype_id ORDER BY stock.stock_date DESC ;";
                $result = mysqli_query($conID, $query) or die(mysqli_error($conID));

                while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
                 $formattedDate = date_create($row['stock_date'])->format('d/M/Y');
                   $details .= "<tr>
                                 <td>{$row['stock_id']}</td>
                                 <td>{$row['namestock']}</td>
                                 <td>{$row['stock_qntt']}</td>
                                 <td>{$formattedDate}</td>
                                 <td>{$row['remark']}</td>
                                 <td>


                                    <button type='button'><a
href='?menu=stock&operation=edit&stock_id={$row['stock_id']}' class='edit-button'><i
class='fas fa-pencil'></i></a></button>
```

```
                    <button type='button'><a
href='?menu=stock&operation=delete&stock_id={$row['stock_id']}' class='delete-
button'><i class='fas fa-trash'></i></a>
                          </td>
                       </tr>";
             }

             $details .= "</table>";

             return $details;




          }

          $details = "<button type='button'><a href='?menu=stock&operation=add'>ADD
STOCK</a></button>
                  <button type='button'><a href='?menu=stock&operation=minus'>TAKE
OUT</a></button>";

            if(isset($_GET['operation']))
            {
              switch($_GET['operation'])
                 {
                  case"edit" : $details .= editStock($_GET['stock_id'],"edit",$conID,
$namaScript);
                       break;
                  case"add" : $details .= editStock("","add",$conID, $namaScript);
                       break;
                  case"delete" : $details .= editStock($_GET['stock_id'],"delete",$conID,
"");
                       break;
                  case"minus" : $details .= editStock("","minus",$conID, $namaScript);
                       break;
                 }
            }




            $details .= "".viewStock($conID);







          }
```

```
            else($isi="<font color=red>Sorry, You are not authorized to use this
function.</font>");


             break;



        case "addStock":if($auth->getUserPerm()<=MANAGER){
               $isi = "<b>Stock Available </b><hr>";



 if(isset($_POST['formsubmitted'])&&($_POST['formsubmitted']=="ejasStok"))
               {
               //print_r($_POST);

               $stkTypeID = $_POST['stocktype_id'];



               $namaStok = $_POST['namestock'];

                if ($stkTypeID && $namaStok )
                {
                $query = "UPDATE stocktype SET  namestock ='$namaStok' WHERE
stocktype_id = '$stkTypeID'";
                $result = mysqli_query($conID,$query) or die("Query Error");
                //echo " ".$query;
                }

                else{}
               }

               /*
               else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="buangStok"))
               {
                $stkTypeID = $_POST['stocktype_id'];

                $query = "DELETE FROM stocktype WHERE stocktype_id =
'$stkTypeID'";
                $result = mysqli_query($conID,$query) or die("Query Error");

                //var_dump($_POST);
               }*/
               else if(isset($_POST['formsubmitted']) &&
($_POST['formsubmitted']=="tambahStok"))
               {
```

```
          $namaStok = $_POST['namestock'];



          $query = "INSERT INTO stocktype (namestock)
          VALUES ( '$namaStok')";
          $result = mysqli_query($conID,$query) or die("Query Error");

          //var_dump($_POST);
        }

        function ejasStok($stkTypeID, $operation, $conID, $namaScript){


      $details = "";


      if($operation == "edit")
      {

        $query = "select stocktype_id, namestock from stocktype WHERE
stocktype_id = '".$stkTypeID."'";
        $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
         while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
        {$stkTypeID=$row['stocktype_id'];$namaStok=$row['namestock'];}

        $details = "
          <hr>

          <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
              <h3 style='color: #333;'>Rename stock if want to change stock</h3>
              <form action='$namaScript?menu=addStock' method='POST'>
                  <input type='hidden' name='stocktype_id' value='$stkTypeID'>
                  <table style='width: 100%;'>
                      <tr>
                          <td>Stock Name</td>
                          <td>:</td>
                         <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' value='$namaStok' name='namestock' /></td>
                      </tr>


                      <tr>
                          <td align='center' colspan='3'>
                              <font color='red'></font>
                             <input type='hidden' name='formsubmitted' value='ejasStok'>
```

```
                                <input type='submit' value='Update' style='margin-right: 10px;
padding: 8px 16px; background-color: #3498db; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                                 <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                              </td>
                           </tr>
                        </table>
                     </form>
                  </div>
              ";

              }

               elseif ($operation == "add")
               {
                  $query = "select stocktype_id, namestock from stocktype WHERE
stocktype_id = '".$stkTypeID."'";
               $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo
$query;
                while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
               {$stkTypeID=$row['stocktype_id'];$namaStok=$row['namestock'];}


                  $details = "
                 <hr>

                <div style='width: 50%; margin: 20px auto; background-color: #f4f4f4;
padding: 20px; border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
                   <h3 style='color: #333;'>Add Stock Name</h3>
                   <form action='$namaScript?menu=addStock' method='POST'>
                      <input type='hidden' name='formsubmitted' value='tambahStok'>
                      <table style='width: 100%;'>
                         <tr>
                            <td>Name of Stock</td>
                            <td>:</td>
                           <td><input type='text' style='width: 100%; padding: 8px; border:
1px solid #ccc; border-radius: 4px;' name='namestock' /></td>
                         </tr>


                         <tr>
                            <td align='center' colspan='3'>
                               <font color='red'></font>
                               <input type='submit' value='Add' style='margin-right: 10px;
padding: 8px 16px; background-color: #4caf50; color: #fff; border: none; border-radius:
4px; cursor: pointer;' />
                               <input type='reset' value='Clear' style='padding: 8px 16px;
background-color: #ddd; border: none; border-radius: 4px; cursor: pointer;' />
                            </td>
                         </tr>
                </table>
            </form>
        </div>
    ";


     }




     /*
    else if($operation == "delete")
    {
      $query = "select namestock from stocktype WHERE stocktype_id = 
'".$stkTypeID."'";
      $result = mysqli_query($conID,$query) or die(mysqli_error($conID));//echo 
$query;
       while($row =  mysqli_fetch_array($result,MYSQLI_ASSOC))
      {$namaStok=$row['namestock'];}

      $details = "

          <hr></hr>
          Delete Stock
          Are you sure to delete Stock $namaStok from the system?
          <form action='$namaScript?menu=addStock' method=POST>
          <input type=hidden name=stocktype_id value='$stkTypeID'></input>
          <input type=hidden name=formsubmitted value='buangStok'></input>
          <table>
          <tr><td><input type=submit value=Yes /></td></form><td><a 
href='?menu=addStock'><input type=button value=No /></a></td></tr>";
    }
    */

    return $details;


  }



  function tengokStok($conID){


    $details = "
    <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css'>
    <style>
        .styled-table {
            border-collapse: collapse;
            width: 100%;
            margin-bottom: 20px;
        }

        .styled-table th, .styled-table td {
            border: 1px solid #ddd;
            padding: 10px;
            text-align: left;
        }

        .styled-table th {
            background-color: #838011;
            color: #fff;
        }

        .styled-table tr:nth-child(even) {
            background-color: #f2f2f2;
        }

        .styled-table td a {
          text-decoration: none; /* Remove underline from links */
          color: #000; /* Color for actions */
        }

        .action-buttons {
          display: flex;
          justify-content: space-around;
        }

        .action-buttons button {
          padding: 8px 12px;
          margin: 0 5px;
          cursor: pointer;
          border-radius: 5px; /* Rounded corners for buttons */
        }

        .add-button {
          background-color: #5d00ff;
          color: #fff;
          border: none;
        }
    </style>


            <table class='styled-table'>
                <tr>

                    <th>Name of Stock</th>
                    <th>Action</th>
                </tr>";
            $query = "SELECT stocktype_id, namestock FROM stocktype ORDER BY
namestock;";
              $result = mysqli_query($conID, $query) or die(mysqli_error($conID));

              while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
                $details .= "<tr>
                          <td>{$row['namestock']}</td>
                          <td>
                            <button type='button'><a
href='?menu=addStock&operation=edit&stocktype_id={$row['stocktype_id']}' class='edit-
button'><i class='fas fa-pencil'></i></a></button>
                          </td>
                        </tr>";
              }

              $details .= "</table>";

              return $details;
            }

            $details .= "<button type='button'><a
href='?menu=addStock&operation=add'>ADD ITEM</a></button>";

            if(isset($_GET['operation']))
            {
             switch($_GET['operation'])
               {
                case"edit" : $details .= ejasStok($_GET['stocktype_id'],"edit",$conID,
$namaScript);
                    break;
                case"add" : $details .= ejasStok("","add",$conID, $namaScript);
                    break;
                case"delete" : $details .=
ejasStok($_GET['stocktype_id'],"delete",$conID, "");
                    break;
               }


            }

            $details .= "".tengokStok($conID);

           }



          else($isi="<font color=red>Sorry, You are not authorized to use this
function.</font>");
```

```
              break;
            }

    }




    else
    {
    $isi = "
    <div style='background-color: #837011; padding: 20px; color: #fff; text-align: center;'>
        <h1>👋 Hi " . ucfirst($auth->getUsername()) . "!</h1>
    </div>";

$details = "
    <div style='margin: 20px; font-size: 18px; background-color: #fff; padding: 20px;
border-radius: 8px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);'>
        <p>Welcome to the <strong>Stock Management System</strong>!</p>
        <p>Here are a few things you can do:</p>
        <ul>
            <li>Update every item for stock in and stock out. (All Staff)</li>
            <li>View and manage entities in the system. (Manager and Staff)</li>
            <li>Access user management for staff updates. (Admin)</li>
        </ul>
       <p>If you have any problems or issues, feel free to contact the Supervisor or Manager
of the restaurant.</p>
        <p>Thank you for using our system! 🚀 </p>
    </div>";

    }





     //include_once "header.php";?>




    <script type="text/javascript">

    $(function () {
     var shortly = new Date();
        shortly.setSeconds(shortly.getSeconds() + <?php echo PAGESESSIONTIME; ?>);
     $('#sessionTimeDisplay').countdown({until: shortly,format: 'MS',expiryUrl: '<?php
echo $namaScript."?menu=logout"; ?>',onExpiry: sessionTimeOut});
```

```
  });

  function sessionTimeOut() {
    alert('Session Expired! Please Login Again. ');
  }


 function confirmDelete(url) {
  if (confirm("!!! Do you really want to delete the record?") == true) {
   window.location.replace(url);
  } else {
  alert('Abort delete operation');
  }
 }




  </script>

 <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    body {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      background-color: #36454f;
    }

    table {
      width: 80%;
      margin: 20px auto;
      border-collapse: collapse;
      background-color: #fff2cc;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    td {
      vertical-align: top;
    }

    .menu-container {
      width: 15%;
      background-color: #ffc371;
      border-right: 1px solid #ffaa5e;
      padding-bottom: 20px;
    }

    .user-info {
        text-align: center;
        padding: 20px;
        border-bottom: 1px solid #ffaa5e;
        background-color: #ffdbac;
    }

    .user-info h3 {
        margin: 0;
        color: #333;
    }

    .navigation {
        text-align: center;
        padding: 20px;
    }

    .navigation a {
        display: block;
        padding: 25px;
        text-decoration: none;
        color: #333;
        font-weight: bold;
        background-color: #ffc371;
        transition: background-color 0.3s ease;
        margin-bottom: 5px;
        border-radius: 5px;
    }

    .navigation a:hover {
        background-color: #ffaa5e;
    }

    .content-container {
        width: 70%;
        background-color: #ffd699;
    }

    .content {
        padding: 20px;
    }

    #sessionTimeDisplay {
        margin-top: 20px;
        font-weight: bold;
        color: #333;
    }
</style>
```

```
</head>
<body>

<table>
    <tbody>
    <tr>
        <!-- Menu -->
        <td class="menu-container">
            <table width="100%" border="0" cellpadding="0" cellspacing="0">
                <tbody>
                <tr>
                    <td class="user-info">
                        <div>
                          <?php echo "<h3>Current User:</h3>" . $auth->getUsername() . "<br>";
?>
                            <hr>
                        </div>
                    </td>
                </tr>
                <tr>
                    <td class="navigation">
                        <div>
                            <a href="cth.php?menu=profile">Main Profile</a>
                            <?php if($auth->getUserPerm()==ADMIN) { ?>
                                <a href="cth.php?menu=users">Manage Staffs</a>
                            <?php } ?>

                            <a href="cth.php?menu=stock">Manage Stock</a>

                            <?php if($auth->getUserPerm()<=MANAGER) { ?>
                            <a href="cth.php?menu=order">Manage Order</a>
                            <?php } ?>

                            <?php if($auth->getUserPerm()<=MANAGER) { ?>
                            <a href="cth.php?menu=addStock">Add Stock</a>
                            <?php } ?>
                          <a href="cth.php?menu=logout" target="_top" style="background-color:
#ff5555; color: #fff; font-weight: bold; border: 2px solid #ff0000; border-radius:
5px;">Logout</a>
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <center><br><div id="sessionTimeDisplay"></div></center>
                    </td>
                </tr>
                </tbody>
            </table>
        </td>
        <!-- Content Area -->
        <td width="1%" valign="top" bgcolor="#ffd699"></td>
        <td class="content-container" valign="top">
            <div class="content">
                <?php echo "$isi <br> $details"; ?>
            </div>
        </td>
        <!-- End of Content -->
    </tr>
    </tbody>
</table>


</body>
</html>



    <?php

    //include_once "footer.php";



    ?>
```