

Credit Based Order Management System

MediTech

Name	Karasinghe Aarachchige Hiruka Tharupathi Karasinghe
BCS Membership number	995093301
Level of Project	Diploma in IT
Submitted in	February 2024
Word Count	4440

Contents

<u>Abstract</u>	9
<u>1.0 Introduction</u>	10
<u>1.1 Context</u>	10
<u>1.2 Current system</u>	10
<u>1.3 Weaknesses of the current system</u>	10
<u>1.4 Proposed System</u>	11
<u>1.5 How Weaknesses are overcome</u>	11
<u>2.0 Requirements Analysis and specification</u>	12
<u>2.1 Background</u>	12
<u>2.2 Requirements Elicitation</u>	12
<u>2.3 Functional Units</u>	13
<u>2.4 Functional Requirements</u>	14
<u>2.5 Non-functional Requirements</u>	15
<u>2.6 Software Development Methodology</u>	16
<u>3.0 Design</u>	17
<u>3.1 data Modeling techniques</u>	17
<u>3.1.1 Use Case Diagrams</u>	17
<u>3.1.2 Activity Diagram</u>	19
<u>3.1.3 Class Diagram</u>	21
<u>3.1.4 Sequence Diagram</u>	22
<u>3.2 Database Design</u>	25
<u>3.2.1 Entity Relationship Diagram</u>	25
<u>3.3 User Interface</u>	27
<u>4.0 Implementation</u>	41
<u>4.1 Introduction</u>	41
<u>4.2 System development tools</u>	41
<u>4.2.1 Software tools</u>	41

<u>4.3 Selecting program languages and styles</u>	42
<u>4.3.1 Languages and packages used</u>	43
<u>4.3.1.1 Languages</u>	43
<u>4.3.1.2 Packages</u>	43
<u>4.4 Selecting development framework</u>	44
<u>4.4.1 Model-View-Controller framework</u>	44
<u>4.5 Codeignitor Framework</u>	45
<u>4.5.1 Introduction</u>	45
<u>4.5.1.1 Codeignitor Controller</u>	45
<u>4.5.1.2 Codeignitor Model</u>	46
<u>4.5.1.3 Codeignitor View</u>	46
<u>4.6 Database Implementation</u>	47
<u>4.6.1 Introduction.....</u>	47
<u>4.6.2 Relational Database</u>	47
<u>4.6.3 Database Development</u>	47
<u>4.7 Application Development</u>	51
<u>4.7.1 Model</u>	51
<u>4.7.2 Controller</u>	54
<u>4.7.3 View</u>	57
<u>4.7.3.1 Use of HTML (Hypertext Markup Language).....</u>	58
<u>4.7.3.2 Use of CSS (Cascading Style Sheets).....</u>	60
<u>4.7.3.3 Use of Bootstrap</u>	62
<u>4.7.3.4 Use of JavaScript.....</u>	63
<u>4.7.3.5 Error Pages.....</u>	64
<u>4.8 Documentation.....</u>	71
<u>5.0 Testing</u>	72
<u>5.1 Testing and verification</u>	72
<u>5.2 Test Cases</u>	73
<u>6.0 Critical Appraisal</u>	82

<u>6.1 Critical appraisal</u>	82
<u>6.2 Future Enhancements</u>	82
<u>6.3 Experience gained</u>	82
<u>6.4 Work Division</u>	83
<u>6.5 Conclusion.....</u>	83
<u>7.0 Appendix.....</u>	84
<u> 7.1 Model.....</u>	84
<u> 7.2 Controller.....</u>	96
<u> 7.3 View.....</u>	124
<u> 7.4 Test Cases.....</u>	157
<u> 7.5 Timeline.....</u>	177
<u> 7.6 Check List.....</u>	177
<u>References.....</u>	178

Figures

Figure 1: Agile Software Development Methodology (www.linkedin.com)	16
Figure 2: Use case diagram for the Credit Based Order Management System	18
Figure 3: Activity diagram (i)	19
Figure 4: Activity diagram (ii)	20
Figure 5: Class Diagram	21
Figure 6: Sequence Diagram (i).....	22
Figure 7: Sequence Diagram (ii)	23
Figure 8: Sequence Diagram (iii)	23
Figure 9: Sequence Diagram (iv).....	24
Figure 10: Entity Relationship Diagram	25
Figure 11: Main Database Tables	26
Figure 12: Login Page	27
Figure 13:Dashboard – Yearly view	28
Figure 14: Yearly view – extended	28
Figure 15: Monthly View	29
Figure 16: User Profile	29
Figure 17: Edit User Profile	30
Figure 18: Customer Management view	30
Figure 19: Customer Management view – Add customer	31
Figure 20: Customer Management view – Update customer	31
Figure 21: Blacklist	32
Figure 22: Order Management	32
Figure 23: Order Management – Add order	33
Figure 24: Order Management – create order(i)	33
Figure 25: Order Management – create order(ii)	34

Figure 26: User Management	34
Figure 27: User Management – Add user	35
Figure 28: User Management – Add user success.....	35
Figure 29: User Management – Edit user	36
Figure 30: User Management – Edit user success.....	36
Figure 31: Product Management	37
Figure 32: Product Management – Add product.....	37
Figure 33: Product Management – Add product success.....	38
Figure 34: Credit Term Management	38
Figure 35: Credit Term Management – Add credit term.....	39
Figure 36: Report Generator	39
Figure 37: Report Generator – success	40
Figure 38: MVC process	44
Figure 39: Database Design	47
Figure 40: Database Main table.....	48
Figure 41: Credit Terms Table	48
Figure 42: Customer Table.....	48
Figure 43: Order Table.....	49
Figure 44: Order_Products Table	49
Figure 45: Payement_records table.....	50
Figure 46: Users table.....	50
Figure 47: Products Table	50
Figure 48: Project View Structure	57
Figure 49: Pre-Condition – Login	73
Figure 50: Post-Condition – Dashboard	74
Figure 51: Pre-Condition – Customer Management Module	75
Figure 52: Post-Condition – Customer Management Module	76
Figure 53: Pre-Condition – Customer Management – Update Customer	77
Figure 54: Post-Condition – Customer Management – Update Customer	77
Figure 55: Pre-Condition – Customer Management – Delete Customer	78

Figure 56: Post-Condition – Customer Management – Delete Customer.....	79
Figure 57: Pre-Condition – Order Management	80
Figure 58: Pre-Condition – Order Management – Add order	80
Figure 59: Post-Condition – Order Management – Add Order – Success Message	81
Figure 60: Post-Condition – Order Management – Add Order	81
Figure 61: Pre-Condition – Order Management – Delete Order	157
Figure 62: Post-Condition – Order Management – Delete Order – success message	158
Figure 63: Post-Condition – Order Management – Delete order..	158
Figure 64: Pre-Condition – User Management – Add user.....	159
Figure 65: Post-Condition – User Management – Add user.....	160
Figure 66: Pre-Condition – User Management – Edit User	161
Figure 67: Pre Condition – User Management – Edit User Form	161
Figure 68: Post-Condition – User Management – Edit user	162
Figure 69: Pre-Condition – User Management – Delete Order	163
Figure 70: Post-Condition – User Management – Delete Order	163
Figure 71: Pre-Condition – Product Management – Add Product.....	164
Figure 72: Pre-Condition – Product Management – Add Product Form	165
Figure 73: Post-Condition – Product Management – Add Product- success message.....	165
Figure 74: Post-Condition – Product Management – Add product	166
Figure 75: Pre-condition – Product Management – Edit Product	167
Figure 76: Pre-condition – Product Management – Edit Product – Form	167
Figure 77: Post-condition – Product Management – Edit Product -success message	168
Figure 78: Post-Condition – Product Management-Edit Product	168
Figure 79: Pre-Condition – Product Management – Delete Product	169
Figure 80: Post-Condition – Product Management – Delete Product	170
Figure 81: Pre-Condition – Credit Term Management – Add Credit Term	171
Figure 82: Pre-Condition – Credit Term Management – Add Credit Term – Form	171
Figure 83: Post-Condition – Credit Term Management – Add Credit Term	172
Figure 84: Pre-Condition – Credit Term Management – Deactivate	173
Figure 85: Post-Condition – Credit Term Management – Deactivate	173

Figure 86: Pre-Condition – Report Generator	174
Figure 87: Post-Condition – Report Generator	175
Figure 88: Pre-Condition – Logout.....	176
Figure 89: Post-Condition – Logout	176

Abstract

With the modernization and development in the information technology over the years, the concept of using digitalized systems for managing financial information has gained traction among other competitors in the market. The accuracy, speed and cost reduction that can be achieved using a digitalized system for managing financial, credit-based information allows for a very large improvement in productivity for the business. The issues that lie within the cooperation were identified during research which grasped the attention of the development team, therefore engaging them to develop an efficient solution to the problem. The software solution will help MediTech realign with it's competitors in the market, making them pioneers of the market again.

1. Introduction

1.1 Context

Medi-Tech is a highly recognized medical consumables and equipment supplier that is registered in the National Medical Regulatory Authority. They mainly supply medical items to government hospitals and other governmental medical institutions. Furthermore, they are a prominent competitor in the dental and orthodontic medical items market as well. Medi-Tech's mission is to ensure that the product quality standards of the products that they provide are satisfiable for both the patients, doctors and other medical staff that will be involved in the use of their products.

1.2 Current system

The current system is entirely paper-based & excel-based and utilizes a very disarrayed file system to organize their records. Data regarding Customers, Staff and inventory are recorded in excel files. This storage mechanism imposes great threats to the business and can cause severe losses in case of loss or unauthorized access of files . Order records are recorded by staff and inserted into a file under the customer name. once a customer makes the payment, a bill or a proof of document will be attached to the file suggesting the closure of the order. There is no alternative method to monitor payments and overdue orders which is the main cause of the business's problem.

1.3 Weaknesses of the current system

Medi-Tech, as previously mentioned, relies on a paper based system for managing and keeping track of their credit-based sales. This has caused unfavorable consequences for both the customers and the business admins, as the currently implemented paper-based system is prone to errors that may cause financial harm for the cooperation. The aspect of credit-negligence by the customer end cannot be properly addressed using the currently installed paper based system as the system lacks a proper mechanism to identify fraudulent customers. This has ultimately driven down the turnover rates of credit-based orders and has negatively impacted the business financial statements as a consequence.

1.4 Proposed System

Considering the issues outlined above, the following system was proposed;

The system will consist of the following functional units.

- Order management – this module will ensure that orders can be created under the name of a certain customer. A credit period can be allocated to each order and if the payment is not paid during the given credit period, the customer will proceed to get blacklisted,

- Blacklisting – as mentioned above, the failure to make due payment within the given time will automatically list the customer as a blacklisted customer. Blacklisted customer will be excluded from the order management module and will not be able to place an order until they are whitelisted again.
- customer management – Basic data regarding customer will be stored in a separate data base.
- Inventory management – this module is used to manage stocks within the business. Users will be able to view the inventory and edit the inventory depending on their access rights, in turn helping employees understand demand and would allow them to cater to this demand effectively.
- report generation –admin staff will be able to view and download reports regarding blacklisted customer, customer details and employee details.
- user management – basic data regarding employees will be stored in a separate database. Their level of access authority will also be filtered at this point.

1.5 How weaknesses are overcome

The proposed system will help MediTech overcome many of their weaknesses such as:

The will now be able to monitor the relevant payments related to each order and whether they are made on time. This will surely help MediTech recover their money, therefore reducing their losses due to credit negligence.

Moreover, customer, user and inventory data are stored in a secure manner and can only be accessed by authorized users.

Unfaithful customer are identified and eliminated from the business process to prevent further harm.

The management of the business are able to produce analytical reports to aid them in their decision making.

2. Requirements Analysis and Specification

2.1 Background

The Business process within MediTech is rather self-explanatory. A customer will first reach out to a sales person within the company and provide the order including the credit period. The salesperson will then place an order for the customer under the customer's name. Once credit periods and the order value has been confirmed, the order is sent to the order-fulfilment department where the order will be packed and delivered to the customer. Once delivery is done accurately, the customer has to settle the full payment of the order within the given credit period. Customers will be reminded of their due payments nearing the deadline. Once the payment is made, the order is complete and the customer's credit records will be cleared. However, if the payments are not made on time, the customer will be listed as a blacklisted customer and will not be allowed to place any other orders until their due payments are cleared.

The following users are expected to interact with the system;

Business owner/managers: The owner/s and manager/s would be allowed full authorized access to the system, allowing them to get an in-depth analysis of the function of the business and its operations. Business Managers and owners will benefit from various functionalities of the system such as report generation. These functionalities will aid decision making and in-depth analysis of the business operations.

Data Entry operators: Data Entry operators are the staff that ensure that the system is updated with the most recent data. They would update inventory stocks, etc, which keeps the system's data accurate at all times.

Sales Operators: Sales operators are the staff that represent the customer within the system. They perform tasks under the name of the customer, for example, creating orders. They have full authorized access to see a customer's past records to ensure that the customers are reliable.

2.2 Requirements Elicitation

This is the mechanism of gathering and defining requirements for a software. (www.geeksforgeeks.org) In the development stages, Requirements elicitation is a significant process because it establishes a clear understanding of the customer's requirements. This is important for the development team to ensure customer satisfaction. Richer quality of information can be gathered using multiple requirements elicitation techniques.

Interviews

Observations

Document Inspection

Questioners

Interviews are the most widely used requirements elicitation technique due to its versatility and ease of organizing. Interviews can help the development team get a grasp of the customer's exact needs and requirements through an array of focused questions. Open-ended questions can convey emotions of customers thus allowing developers to collect qualitative data along the process.

The requirements elicitation process at MediTech consisted of a variety of **interviews** held with the management and the working staff of the business. These interviews helped us understand the financial threats the cooperation is facing, the lack of organization and even more practical aspects such as usability requirements, and more.

Observations can convey more data and information than interviews on certain occasions. This is because some business processes are extremely difficult to explain and point down. Moreover, behaviors of staff and customers during daily tasks can sometimes be unpredictable and observations can help developers get a clear idea of these processes. The observations done at MediTech helped our developers get an understanding about the daily run of the businesses thus opening us up to a plethora of added functionalities that needed to be considered.

Document Inspections at MediTech helped our team understand and analyze the current system, allowing us to engage with the current business process. Through this, we understood what data and information they record and how they record it. These factors were all considered during development to ensure the smooth transition from a paper-based to a digital system.

By using **questioners**, we were able to reach almost all the employees in the business. This allowed the development team to gather details about the staff's overall dissatisfaction about the paper-based system and helped us understand what areas needed improvement. With this data we were able to satisfy staff with what they requested.

2.3 Functional Units

The major functional units of the project were deduced to be as follows;

User management

Customer management

Order management

Blacklist Management

Product Management

Report generation

Credit term management

2.4 Functional Requirements

Functional requirements are the requirements that the end user specifically demands as basic facilities that has to be included within the system. It is a must that all these functionalities are present in the end product to meet the basic agreement level between developer and customer. Usually functional requirements describe the behavior of the system under very specific conditions. (www.geeksforgeeks.org)

The functional requirements of the system can be listed as follows

Create user login – The Admins of the system will be able to create new user logins for new staff members such as sales operators or data entry operators. This can be done using the user management component.

Availability of multiple user permissions – The administrators will be able to assign permissions to users depending on their role in the company and will be able to edit permissions using the user management component.

Create a customer – The administrators and sales staff will possess the power to create a new customer and edit an existing customer's details as they have access to the Customer management component.

Create and edit products- Admin users and data entry operators will be able to add new products to the system, edit product details or prices and delete products through the product management component.

Amending credit terms – Admin users and data entry operators will be granted permission to amend the credit terms through the Credit term management components.

Placing orders – admin users and sales operators will be able to place an order on behalf of any whitelisted customer using the order management component.

Updating payment status of orders – the admin staff and sales operators will be granted permission to update the payment status of an order by submitting the necessary documents of proof using the order management component.

Blacklist – A blacklist will be automatically generated and will consist of the customer who have been unable to settle due payment/s on time. Admin staff and sales operators will be able to view this pre-produced blacklist

Reports – admin users will be able to view and download reports that will be automatically produced through the system. These reports will consist of data regarding customer, orders placed and the blacklist itself.

2.5 Non-functional requirements

These are the quality constraints that must be satisfied by the newly developed system. These outline how the system should perform when in use rather than the functionalities of the system.

Non-functional requirements are what ensures the system's usability, reliability and efficiency which will ultimately affect the user experience.

The non-functional requirements are mentioned as follows:

Security :

Security determines the safety of the data and information stored within the system and is absolutely crucial when developing any system. Security should be a component tightly knit with each and every component within the development process. The data and information stored in MediTech's system is also surrounded by a considerable level of confidentiality, and therefore necessary steps should be taken to safeguard this valuable information.

Security is maintained within the system by authenticating the user before logging in. Usernames and Passwords are used to make sure that only authorized users get to view data.

Strong database passwords were used to ensure that no confidential data regarding customer or users are leaked.

Finally, the system completely complies with the *Data Protection Act 2018* and the *UK General Data Protection Regulation (UK GDPR)*

Performance:

Performance is a measure of the efficiency with which the system produces outputs for the inputs of a user. Performance is usually measured under a predefined workload to test its ability to function under these conditions. It is assured that each module of the newly built system will provide maximum efficiency to its users.

Moreover, clear error messages will be displayed that the occurrence of errors will be minimized overall.

Usability:

Usability is a key non-functional requirement that determines the user's satisfaction. The new system has a simple GUI that is very easily understandable and is easy to navigate, thus providing users with a good user experience.

Availability:

The system is fully available to any user given that a good internet connection is available.

Correctness:

Users can be assured that every function within the system provides accurate results as these functions have been individually tested by the use of unit testing.

Reusability:

Data stored can be reused if needed

2.6 Software Development Methodology

The agile software development methodology allows developers to break down the system into smaller components and phases allowing them to individually develop each with a high attention to customer collaboration to each component. Furthermore, this methodology facilitates continuous improvements of the system whilst allowing an iterative approach for delivering the final solution.

Advantages	Disadvantages
Continuous delivery of the software can be achieved, leading to customer satisfaction.	Less predictable, this is caused by the extremely flexible nature of the agile model.
The agile methodology allows the customers to change the requirements easily	Very time consuming as the collaborative approach of the agile methodology requires a high amount of time.
The thorough attention to the technical and design related aspects of the software solution lead to greater product quality.	The agile documentation structure is not very robust.
The use of face-to-face communication and high collaboration results in the elimination of miscommunication.	A high deal of focus and concentration is expected both from the customers and the developers, if not satisfied, the project may fail.

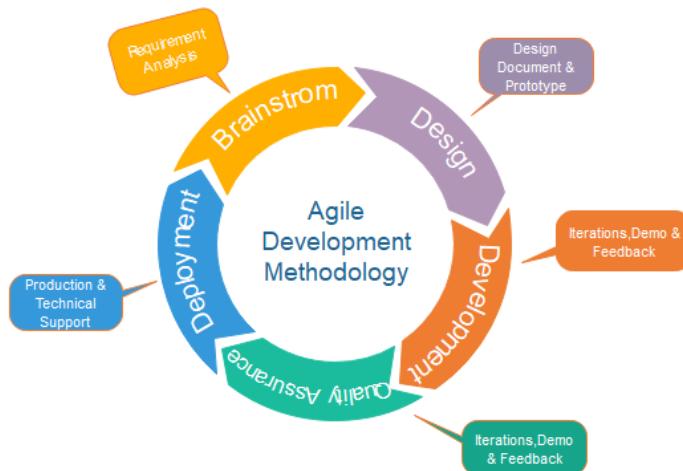


Figure 1: Agile Software Development Methodology

3. Design

After the system requirements have been gathered and understood by the development team, a blueprint of the system solution will be developed. This includes architectural designs, user interphase designs, database designs and security designs.

3.1 Data Modeling Techniques

Data modeling is the process of curating a visual representation of the entire system of parts of it to communicate connections between data points and structures. The purpose of Data modelling is to identify the types of data used and to form connections between these types of data and the ways the data can be grouped and organized and its formats and attributes. (www.ibm.com). Unified Modeling Language (UML) was used to create these data models within the system. The UML diagrams that were used made it simple to specify, visualize, modify, construct and document the artifacts of the system.

3.1.1 Use case diagrams

The use case diagram mainly revolves around 2 key details. The use case and the actor. The user cases represent a functional requirement expected from the system whilst the actor represents the users of the system which give and take information from and to the system. The use case diagram describes the interactions between the actor and the use cases within the confines of the system.

Use case diagrams are an elaborate analysis of the user requirements collected during the early stages of the software development cycle. The creation of the use cases requires a high amount of user involvement and the uses cases that are produced must reflect the actual functional requirements of the system clearly. This in turn, should minimizes the risk of failure during the latter stages of the development.

The use case diagram serves the purpose of portraying an estimation of the system's complexity and scale for the project manager and business analyst. This helps them plan, schedule, estimate resources, staff, and create rough cost estimates as a result.

This diagram can also be utilized by the developers for the purposes of development or as a tool to test the final product against.

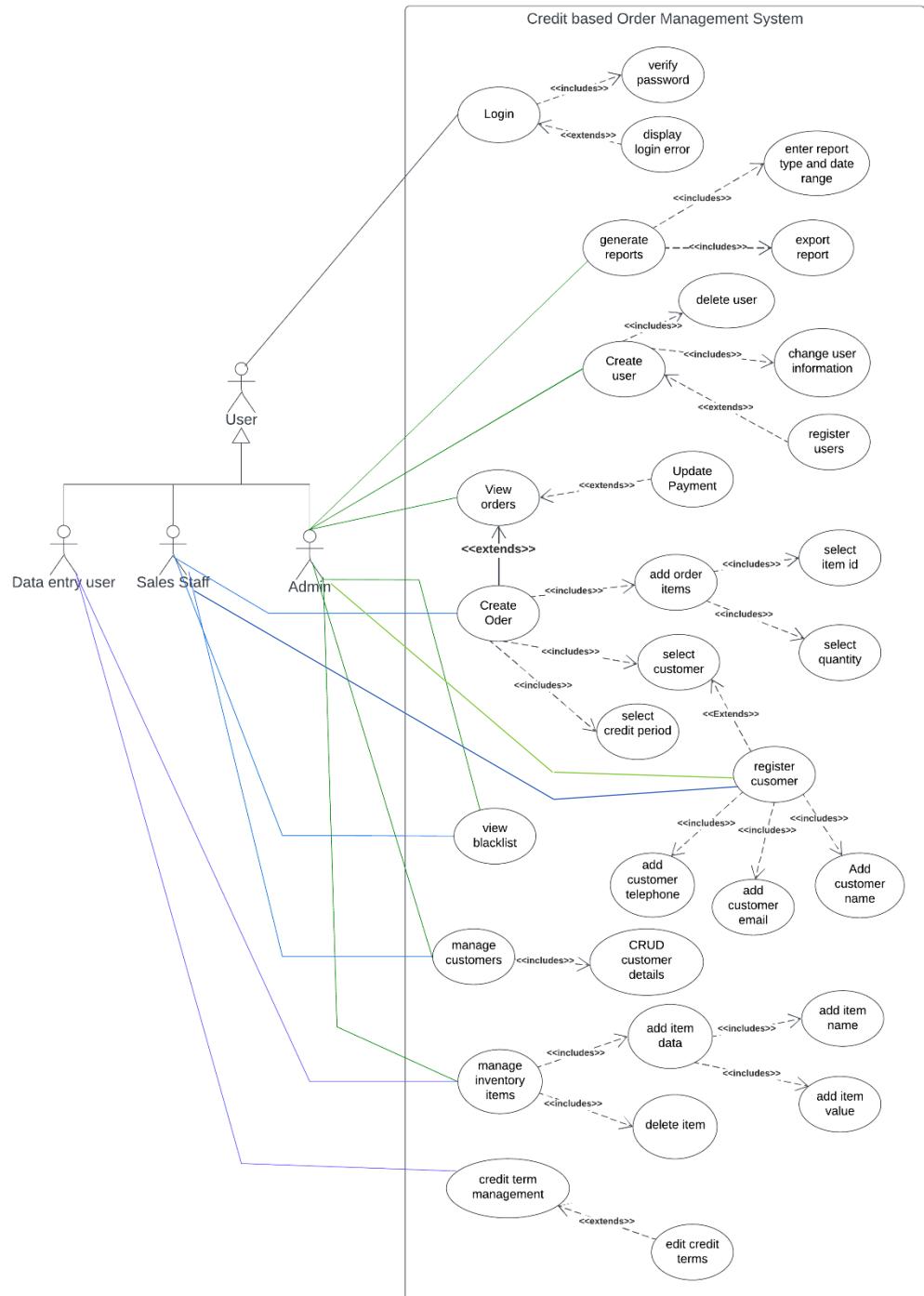


Figure 2: Use case diagram for the Credit Based Order Management System

3.1.2 Activity Diagrams

Activity diagram is a behavioral diagram that describes the dynamic aspect of a system. this diagram is a more advanced version of a flow chart and it models the flow from one activity to another. this flow can be sequential, branched or concurrent. (www.visual-paradigm.com)

an activity is a particular operation of the system. We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a event using an activity diagram. (www.geeksforgeeks.org)

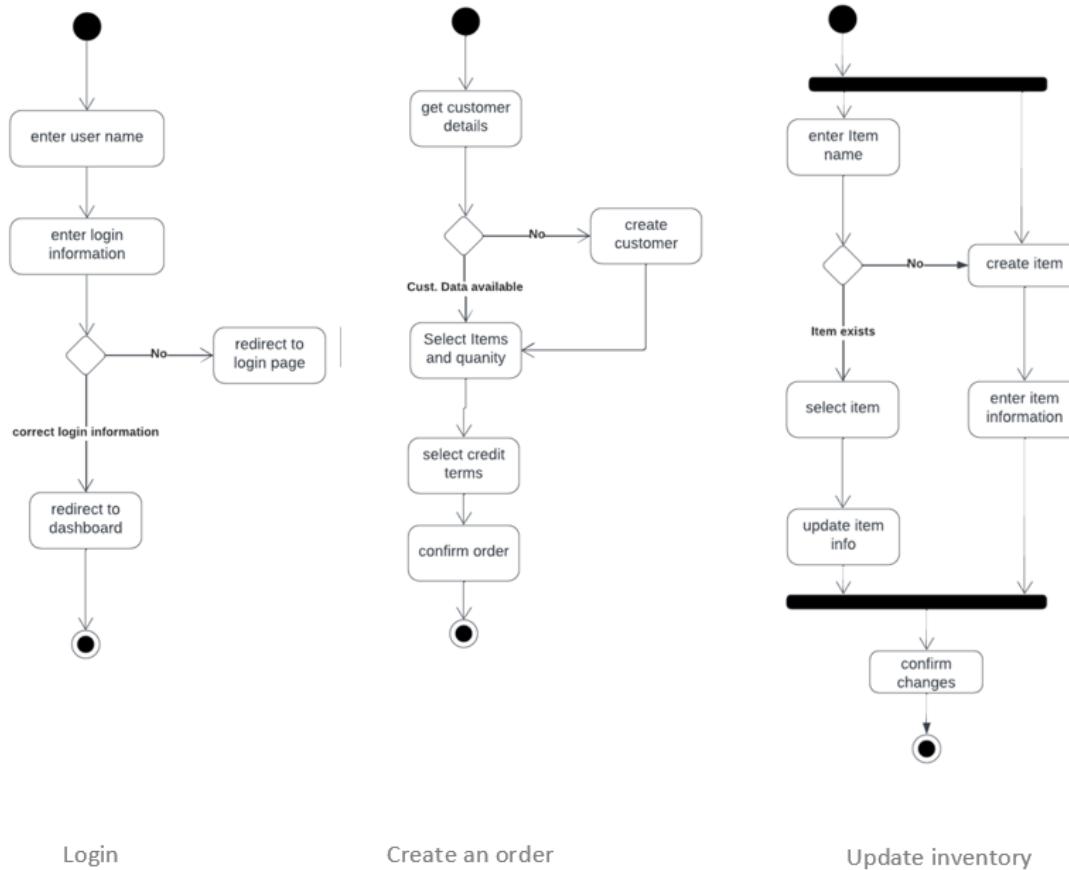


Figure 3: Activity diagram (i)

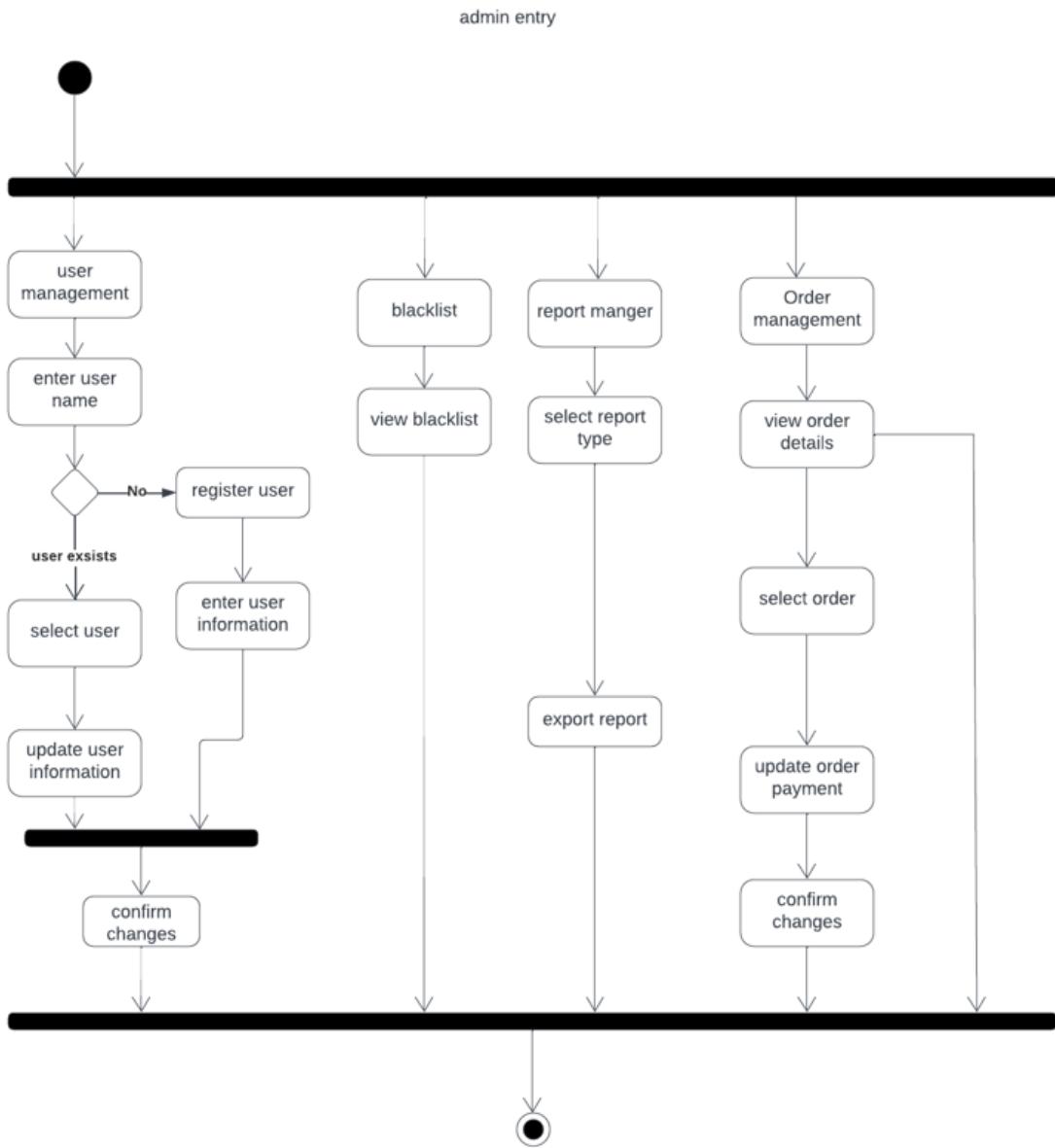


Figure 4: Activity diagram (ii)

The “Create an order” activity diagram showcases an important function of the system. When placing an order, if the customer is not already registered with the system, this functionality helps the user redirect from the order module page to the customer module and register the customer before placing the order.

3.1.3 Class Diagrams

Class diagrams are a type of UML diagrams that are specifically designed for illustrating the structure of the system and therefore is very valuable for the development of the software solution. It provides a static representation of the system with regards to the relationships between classes of a system.

A class is generally used as a blueprint for the creation of an object. The object, therefore, is an instance of a given class. Classes can house its own attributes and methods which can be used by each of its objects.

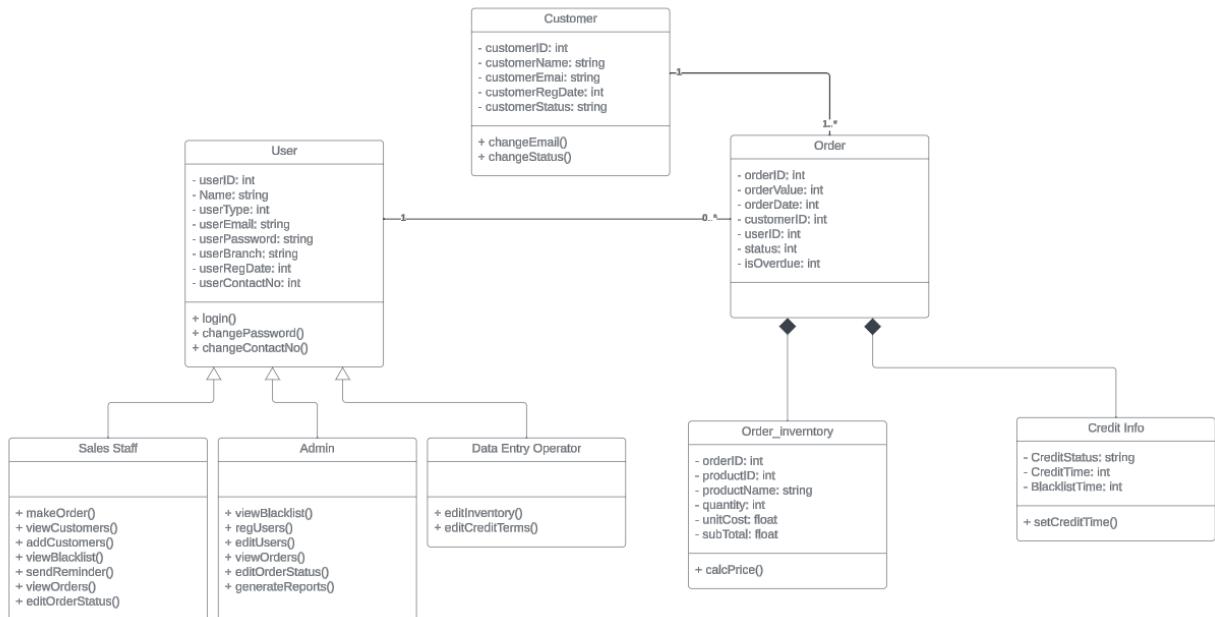


Figure 5: Class Diagram

3.1.4 Sequence Diagrams

The sequence diagram is an interaction diagram that is used to reflect how the objects within a system interact with each other and in what sequence. The diagram can be used to understand when a object is active and in what sequence it interacts with each other object and what are the operations that are done by each. This helps developers identify certain problems during the early stages of the development, particularly problems related to user interface and problems in system logic.

The sequence diagrams for each user component for the Credit Based Order Management System is given below ;

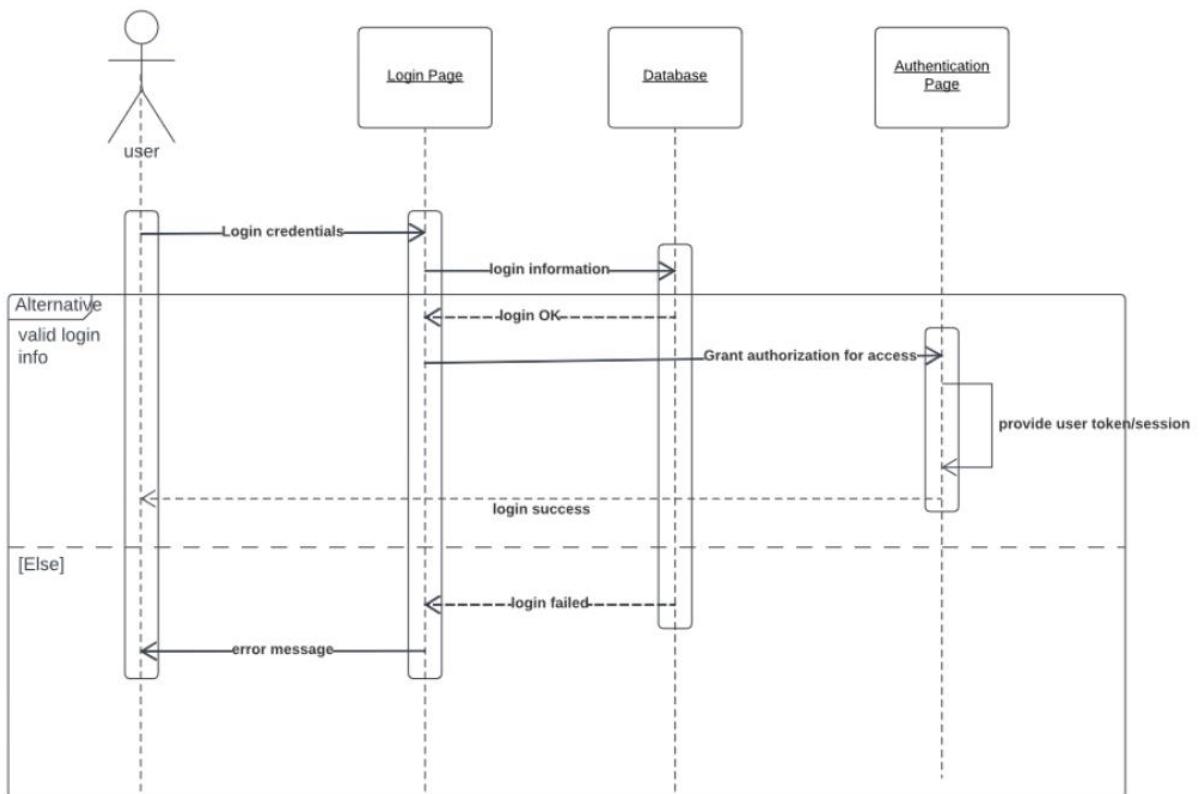


Figure 6: Sequence Diagram (i)

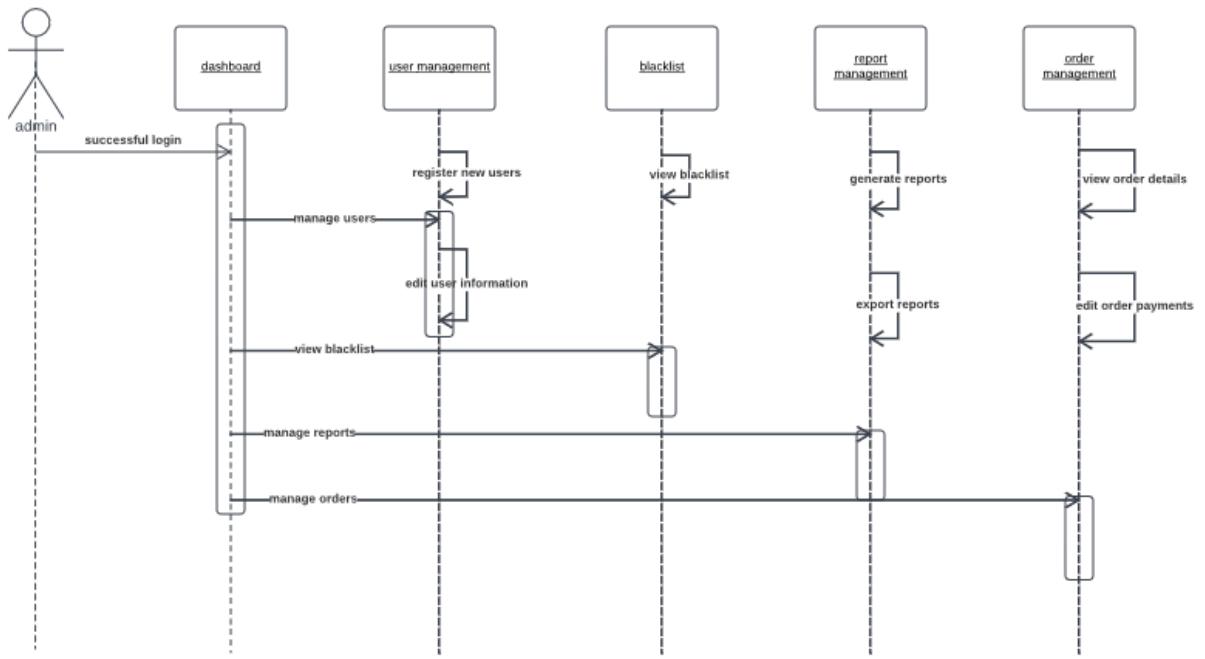


Figure 7: Sequence Diagram (ii)

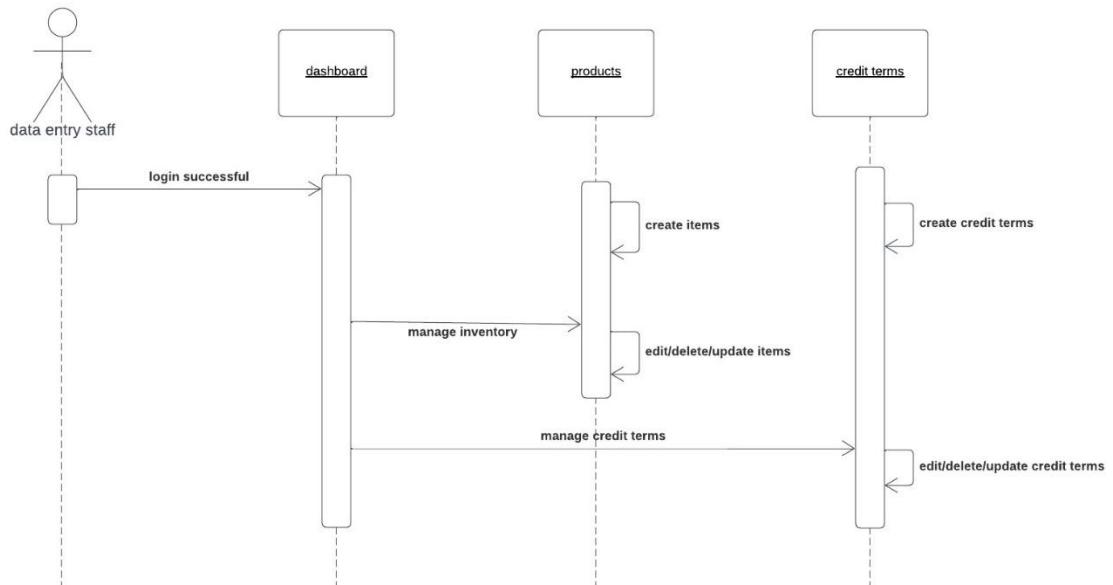


Figure 8: Sequence Diagram (iii)

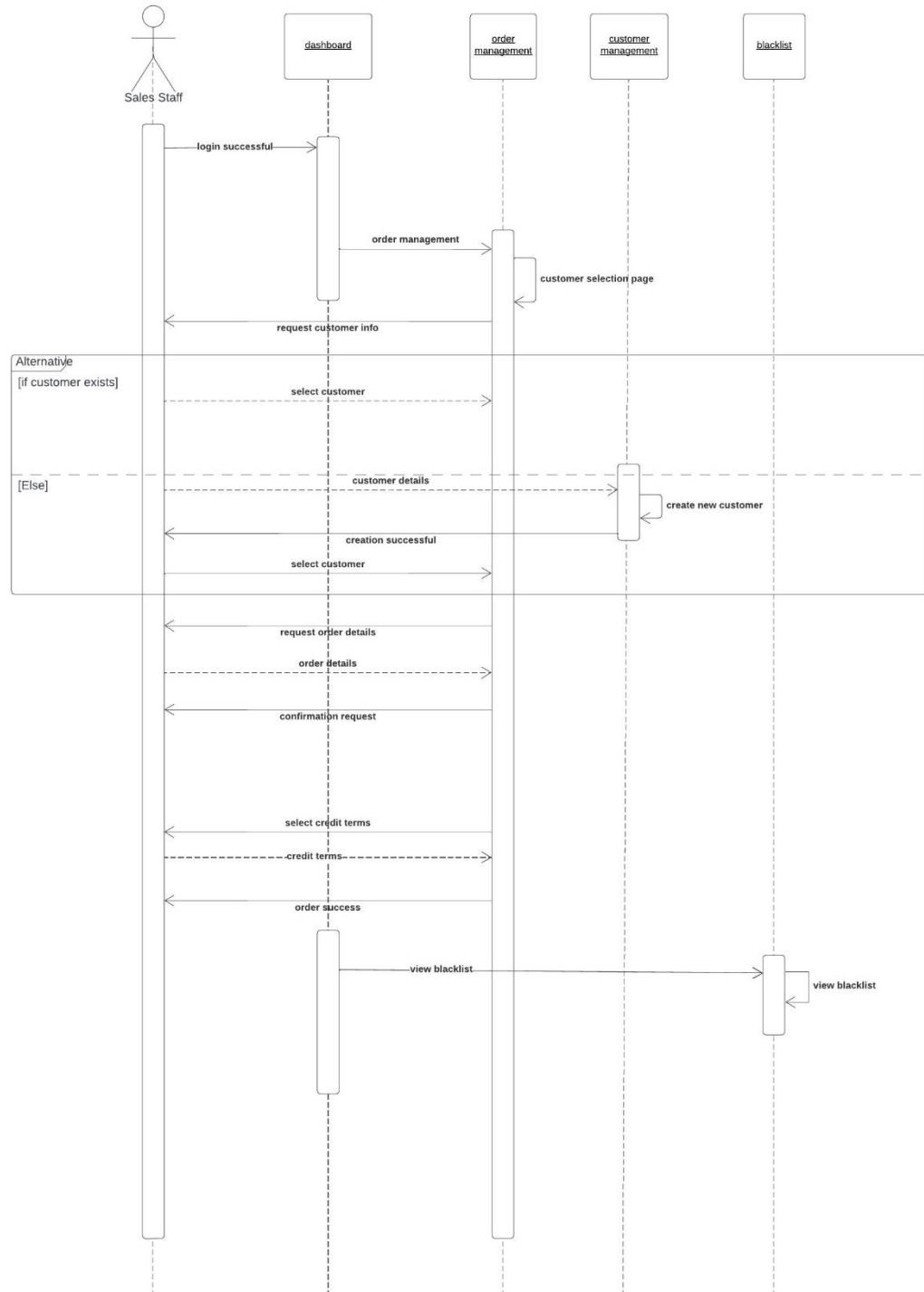


Figure 9: Sequence Diagram (iv)

3.2 Database Design

3.2.1 Entity Relationship Diagrams

Entity relationship diagram is utilized for the purpose of reflecting the relationships between the various entities stored in the database structure. This acts as a visual representation for the relationships between different people, objects, places and concepts or events with a system. Entities carry different attributes that outlines their properties.

ERD helps define the business process by means of data modelling techniques and therefore can be used as a foundation procedure of a relational database.

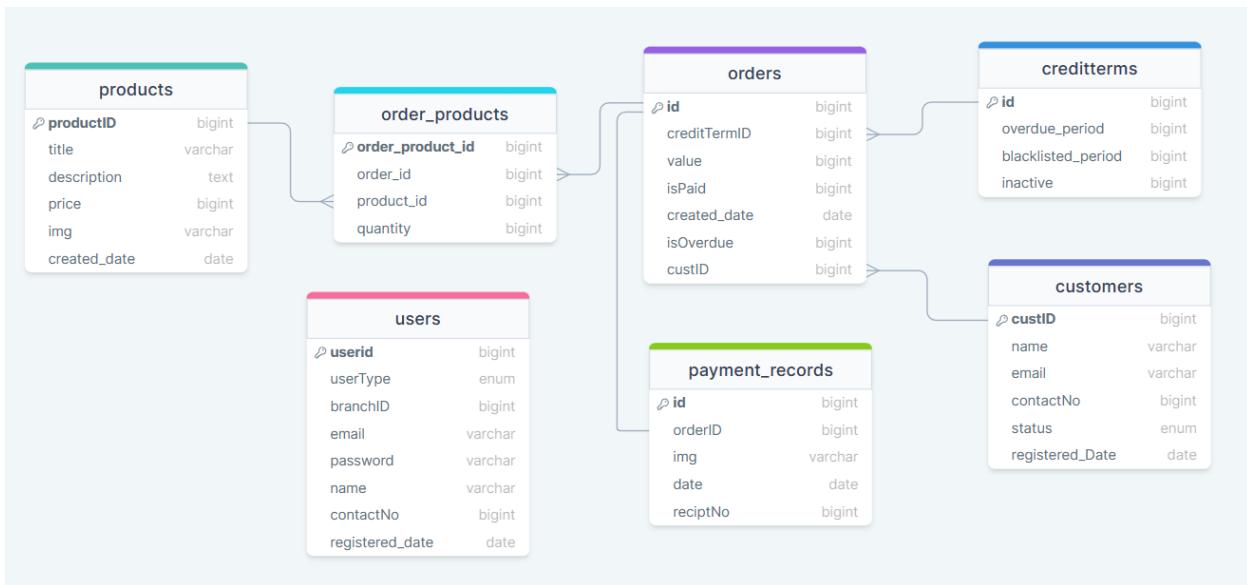


Figure 10: Entity Relationship Diagram

Server: 127.0.0.1 > Database: meditech

Structure	SQL	Search	Query	Export	Import	Operations	Privileges	Routines	Events	Trig
Filters										
Containing the word: <input type="text"/>										
Table	Action	Rows	Type	Collation	Size	Overhead				
<input type="checkbox"/> credittersms	 Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-				
<input type="checkbox"/> customers	 Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-				
<input type="checkbox"/> orders	 Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	48.0 KiB	-				
<input type="checkbox"/> order_products	 Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	48.0 KiB	-				
<input type="checkbox"/> payment_records	 Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-				
<input type="checkbox"/> products	 Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-				
<input type="checkbox"/> users	 Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-				
7 tables	Sum		42	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B			
<input type="checkbox"/> Check all		With selected: <input type="button" value="▼"/>								

Figure 11: Main Database Tables

3.3 User Interface

The user interface of a system may be one of the most important factors of a system as it directly deals with the user. Even if all the functional requirements of the user have been achieved, if the user interface is unusable, the system may not serve any purpose to the user at all. To ensure that users are comfortable with the design of the user interface, developers should use simple designs, dialog boxes, scroll bars, icons and mouse interactions. Therefore, the decision was made to use a very simple and descriptive UI for the system to help with easier and efficient navigation of the system.

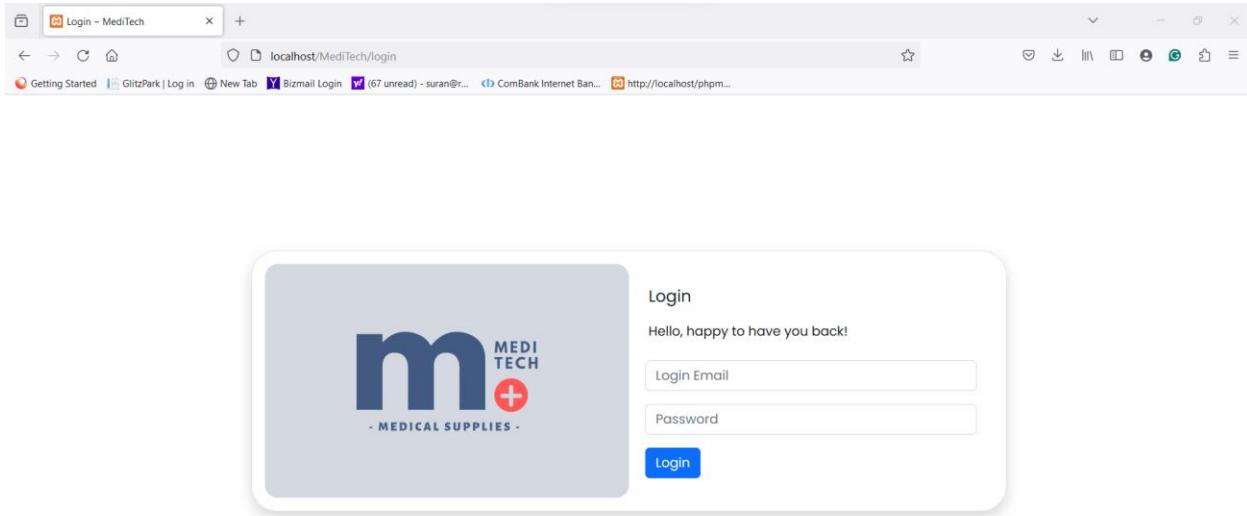


Figure 12: Login Page

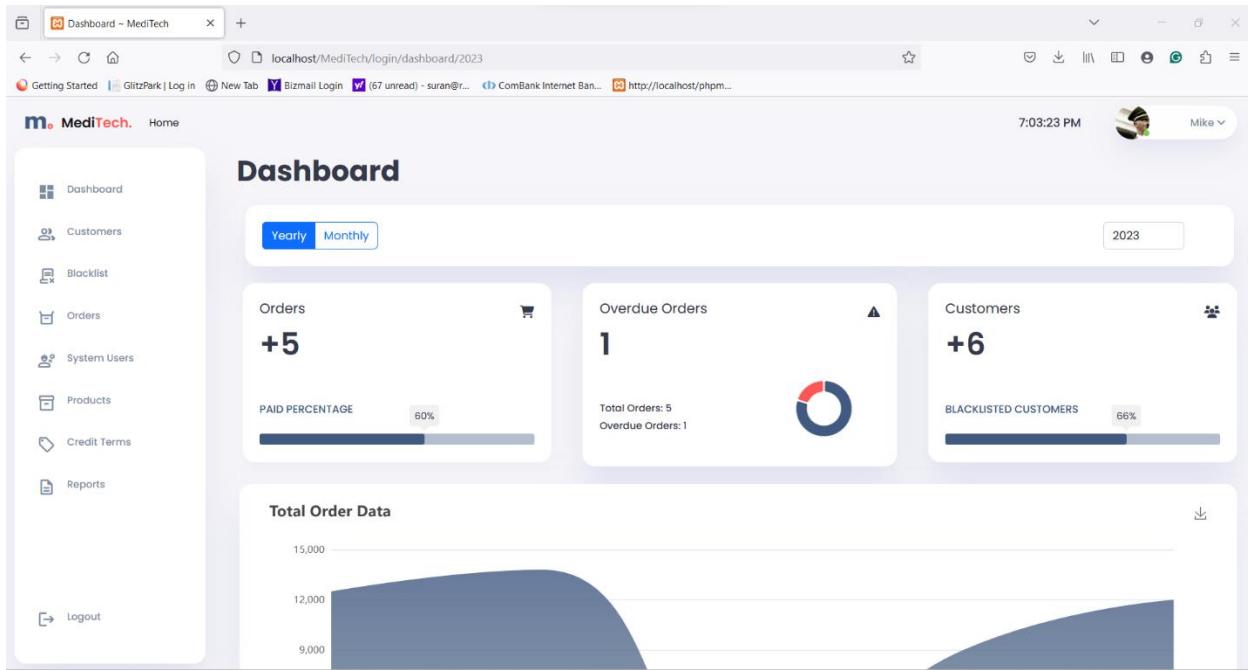


Figure 13: Dashboard – Yearly view

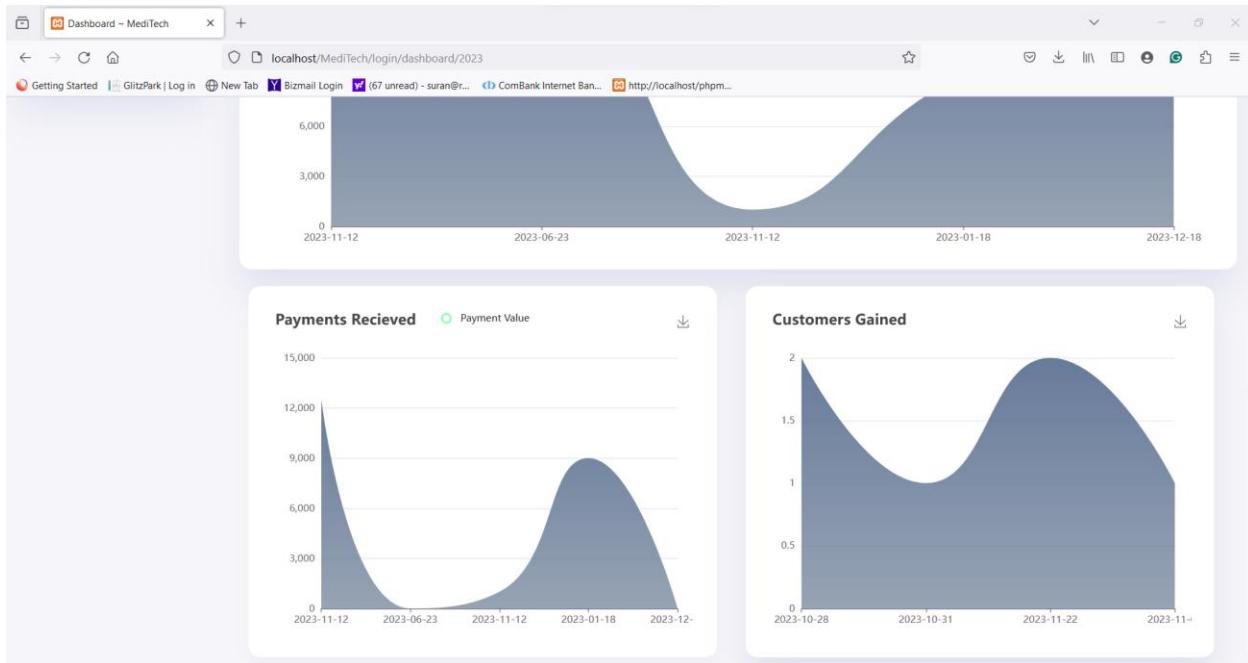


Figure 14: Yearly view - extended

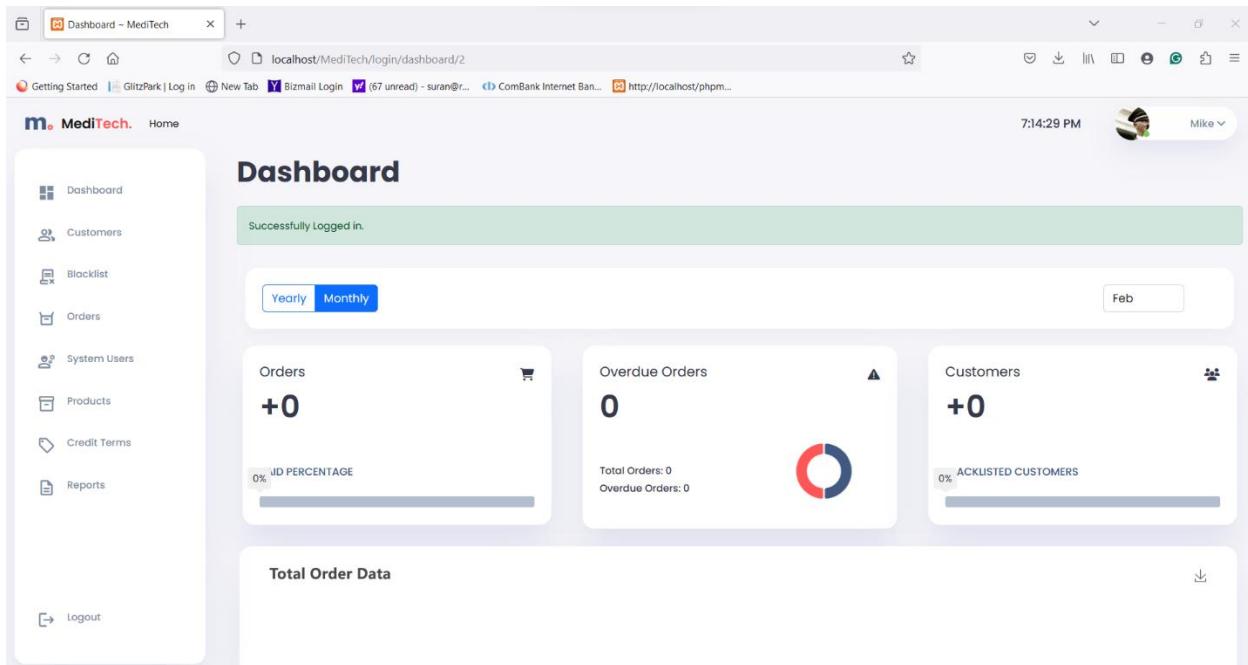


Figure 15: Monthly View

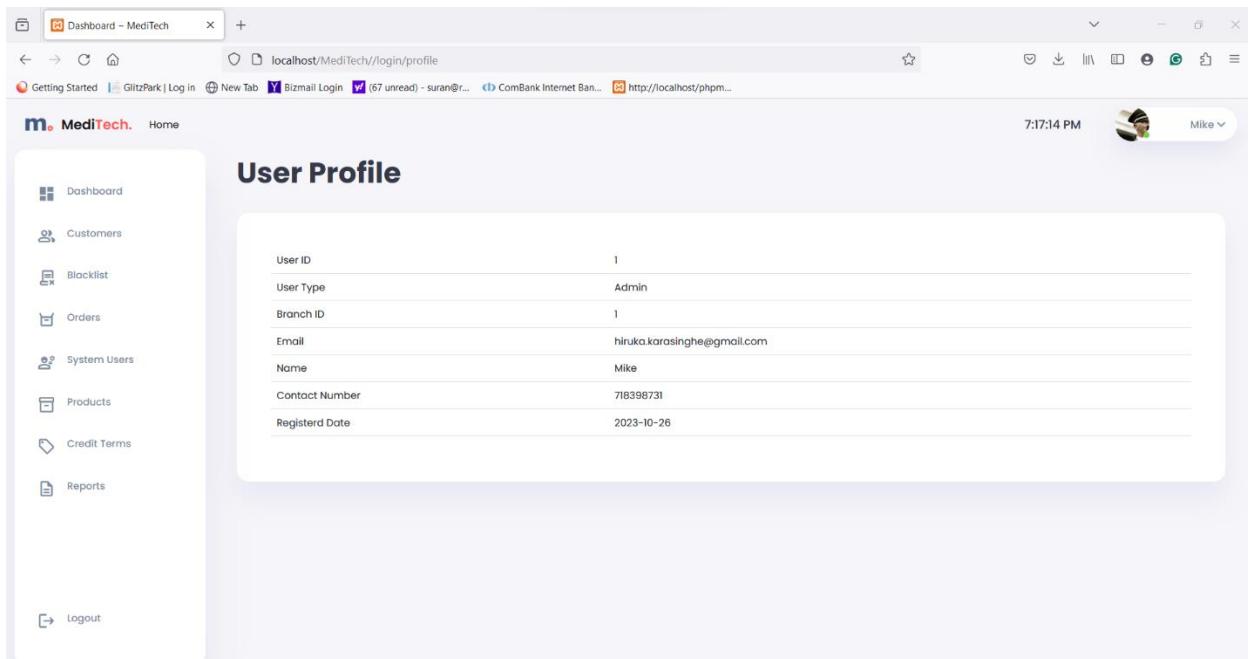


Figure 16: User Profile

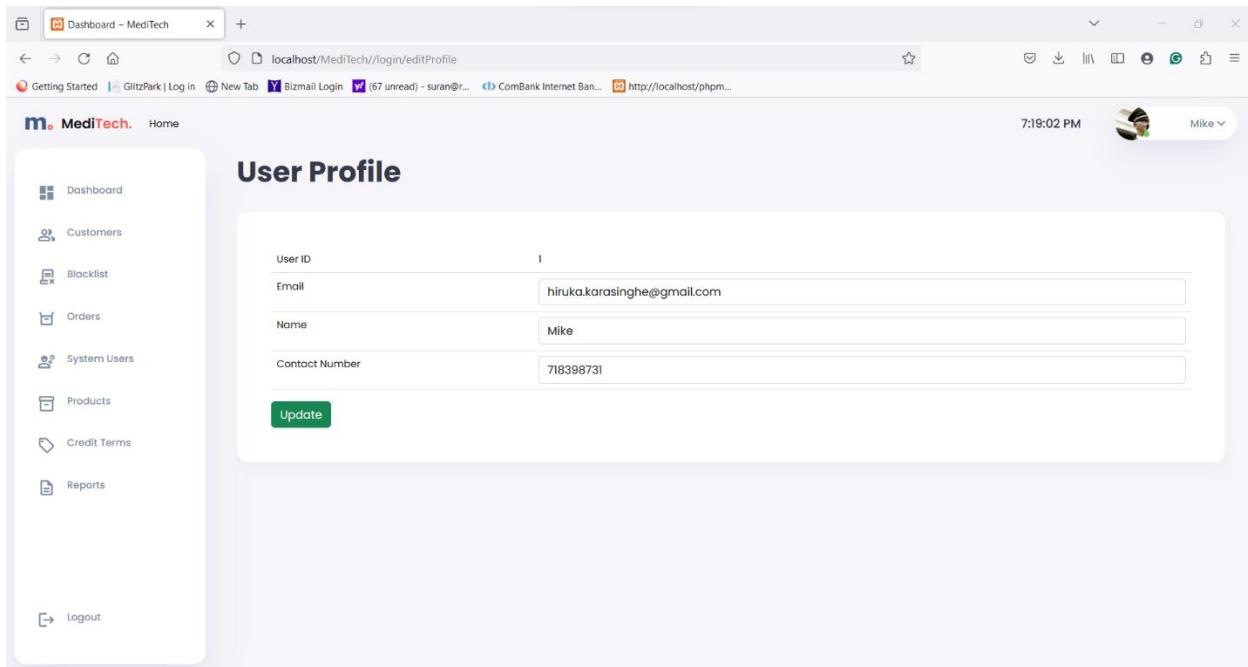


Figure 17: Edit User Profile

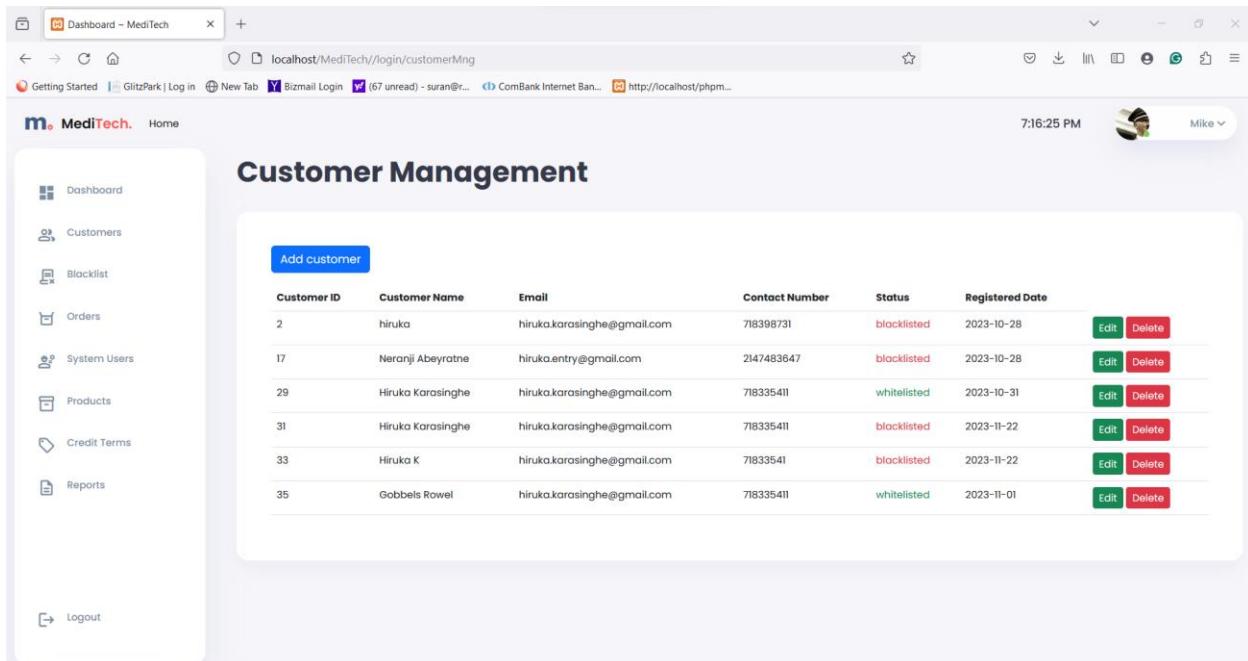


Figure 18: Customer Management view

Customer Management

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	blocklisted	2023-10-28
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blocklisted	2023-10-28
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22
33	Hiruka K	hiruka.karasinghe@gmail.com	71833541	blocklisted	2023-11-22

Figure 19: Customer Management view – Add customer

Customer Management

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	blocklisted	2023-10-28
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blocklisted	2023-10-28
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22
33	Hiruka K	hiruka.karasinghe@gmail.com	71833541	blocklisted	2023-11-22

Figure 20: Customer Management view – Update customer

The screenshot shows the MediTech application's Blacklist page. The left sidebar contains navigation links for Dashboard, Customers, Blocklist (which is selected), Orders, System Users, Products, Credit Terms, and Reports. The main content area has a title 'Blacklist' and a search bar labeled 'Search customer name'. Below the search bar is a table with columns: Customer ID, Customer Name, Email, Contact Number, and Registered Date. The table data is as follows:

Customer ID	Customer Name	Email	Contact Number	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	2023-10-28
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	2023-10-28
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	2023-11-22
33	Hiruka K	hiruka.karasinghe@gmail.com	71833541	2023-11-22

At the bottom left is a 'Logout' link.

Figure 21: Blacklist

The screenshot shows the MediTech application's Order Management page. The left sidebar contains navigation links for Dashboard, Customers, Blocklist, Orders (selected), System Users, Products, Credit Terms, and Reports. The main content area has a title 'Order Management' and two buttons: 'Add Order' and 'New Customer'. Below these buttons is a search bar labeled 'Search order by customer'. A table lists orders with columns: Order ID, Customer Name, Value, Status, and Created Date. The table data is as follows:

Order ID	Customer Name	Value	Status	Created Date	Action	
16	Hiruka Karasinghe	12500	Paid	2023-11-12	<button>Details</button>	
17	hiruka	13800	Unpaid	Overdue	2023-08-23	<button>Update</button>
18	Hiruka Karasinghe	1000	Paid	2023-11-12	<button>Details</button>	
19	Gobbels Rowel	31200	Paid	2022-11-08	<button>Details</button>	
20	Gobbels Rowel	9000	Paid	2023-01-18	<button>Details</button>	
21	hiruka	12000	Unpaid		2023-12-18	<button>Update</button>

At the bottom left is a 'Logout' link.

Figure 22: Order Management

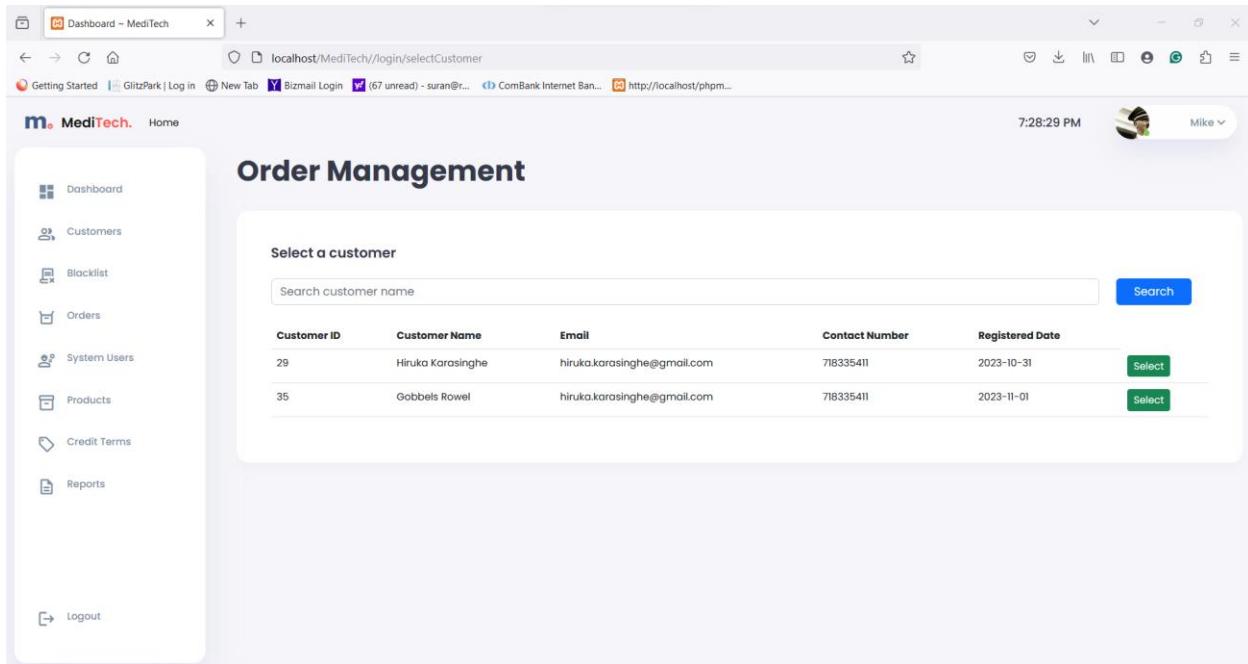


Figure 23: Order Management – Add order

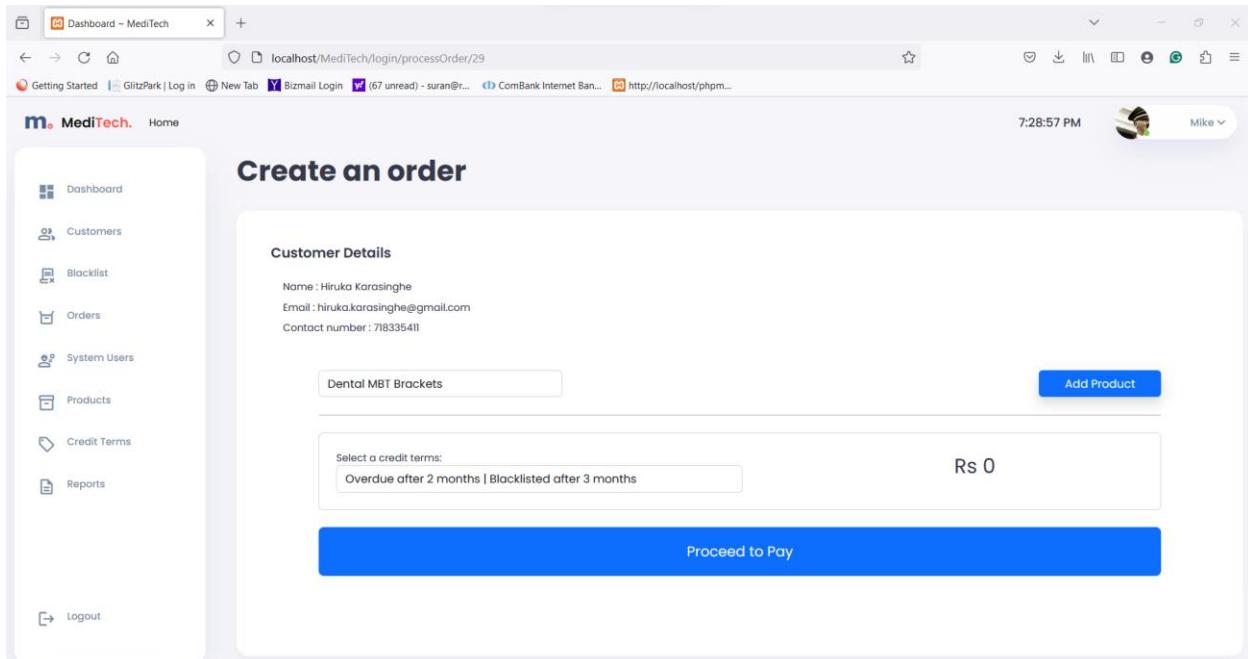


Figure 24: Order Management – create order(i)

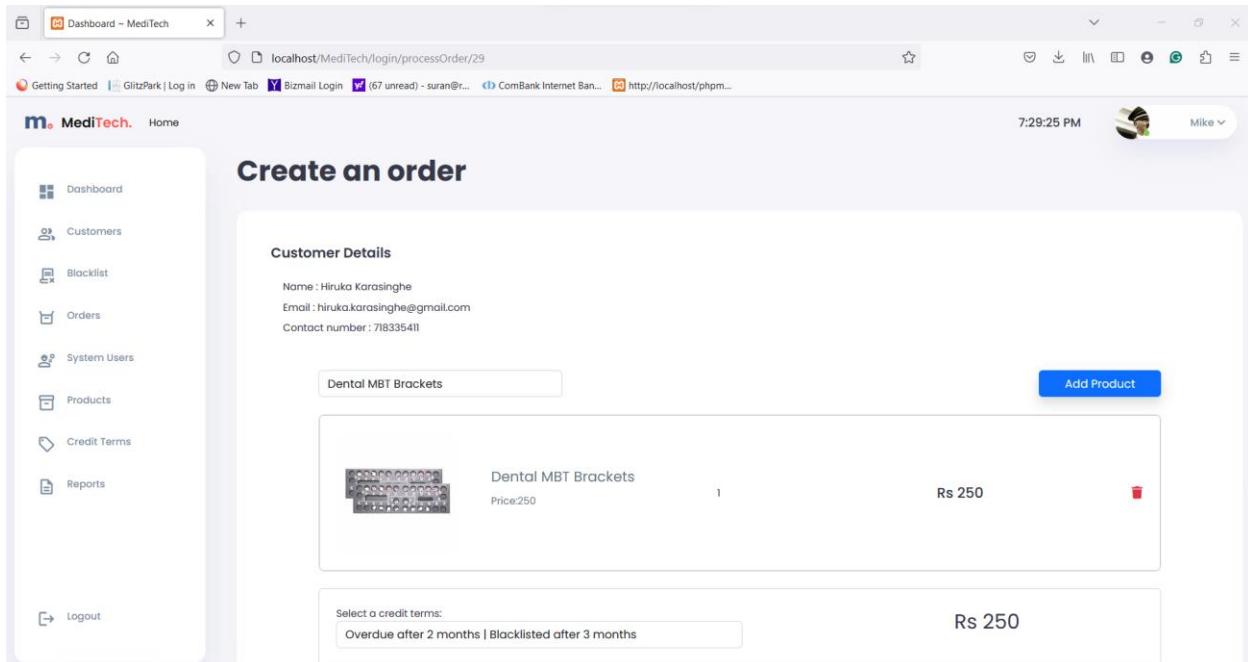


Figure 25: Order Management – create order(ii)

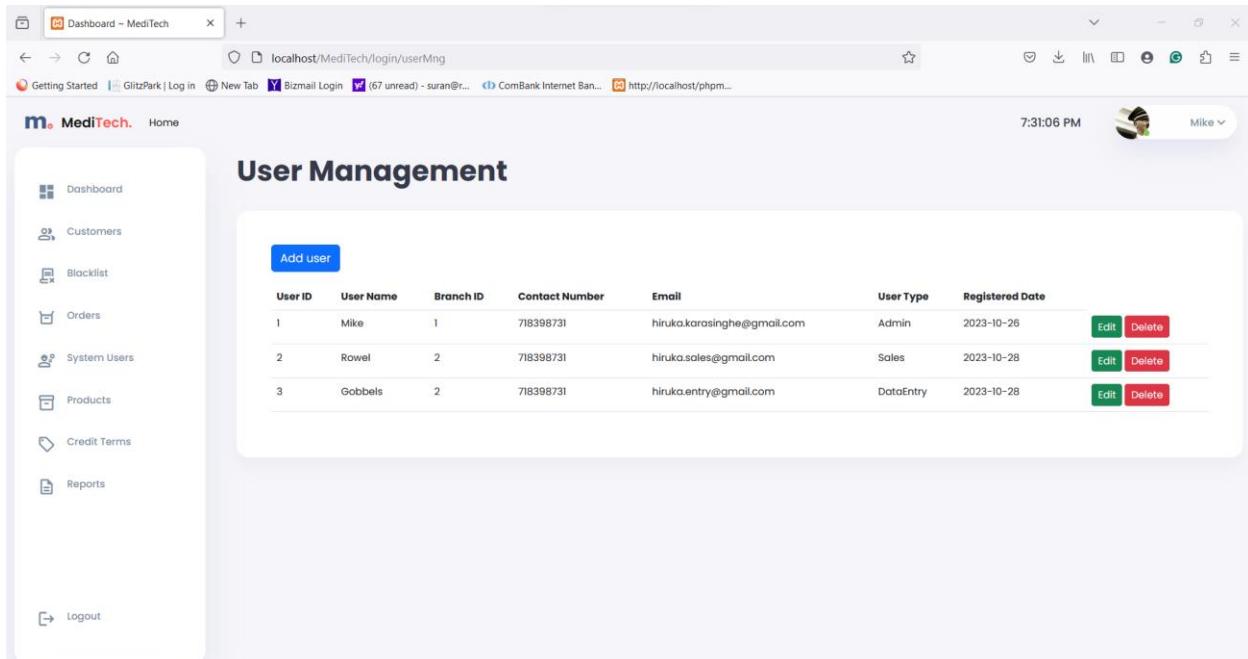


Figure 26: User Management

User Management

Name	Email
Nicky	pinindiweeraratne@gmail.com

Contact Number	Branch ID
7789450678	1

Password	Select a user type:
*****	DataEntry

Add User

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date
1	Mike	1	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28

Figure 27: User Management – Add user

User Management

User added successfully.

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date
1	Mike	1	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28
10	Nicky	1	2147483647	pinindiweeraratne@gmail.com	DataEntry	0000-00-00

Figure 28: User Management – Add user success

User Management

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date
1	Mike	1	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28

Figure 29: User Management – Edit user

User Management

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	DataEntry	2023-10-26
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28
10	Nicky	1	2147483647	pinindiweeraratne@gmail.com	DataEntry	0000-00-00

Figure 30: User Management – Edit user success

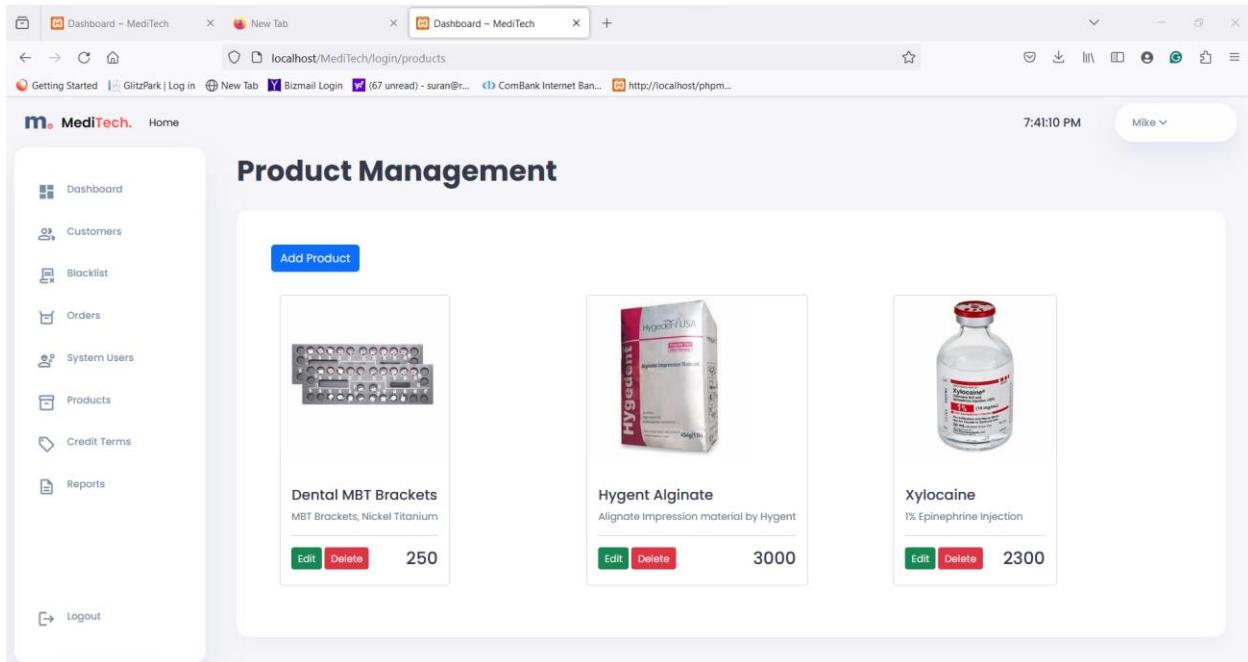


Figure 31: Product Management

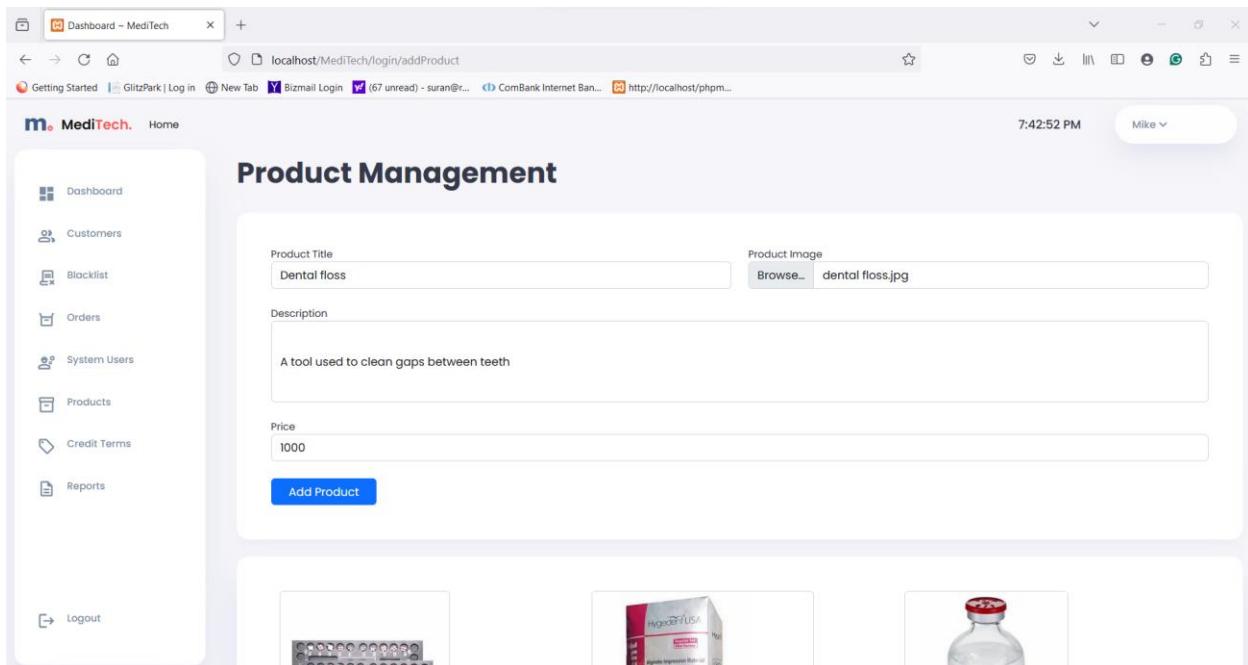


Figure 32: Product Management – Add product

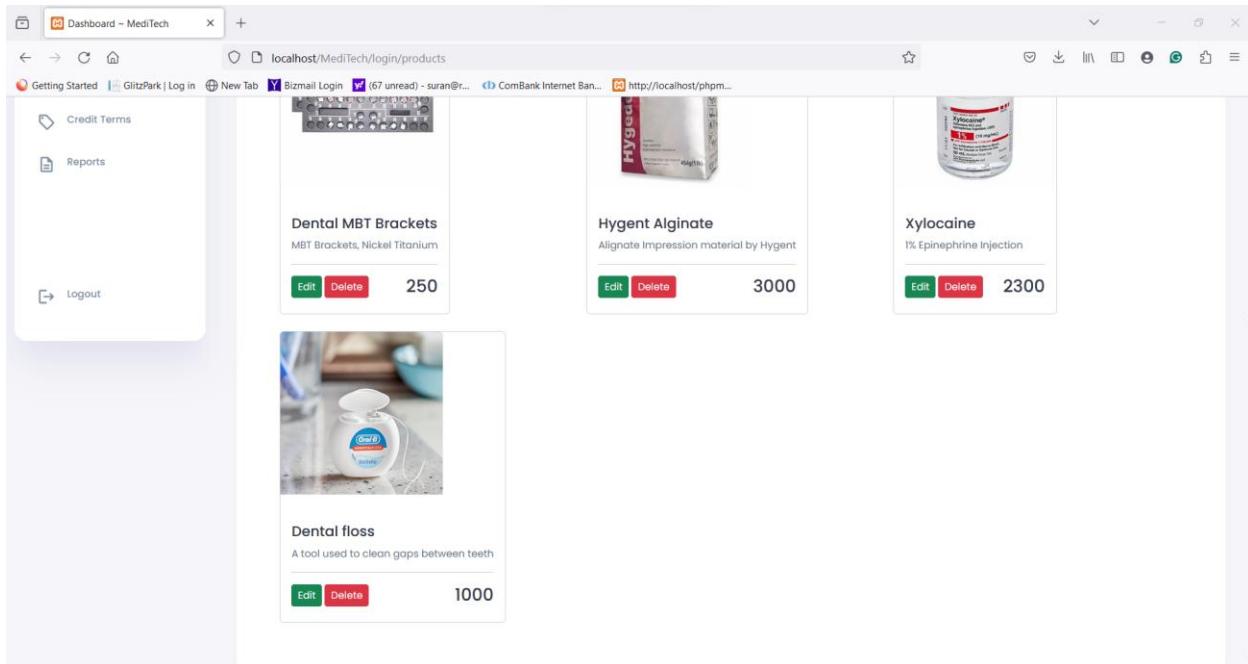


Figure 33: Product Management – Add product success

The screenshot shows a web browser window titled "Dashboard - MediTech". The URL is "localhost/MediTech/login/credit". The page title is "Credit Term Management". On the left, there is a sidebar with links: "Dashboard", "Customers", "Blacklist", "Orders", "System Users", "Products", "Credit Terms", and "Reports". The main area features a "Add Credit Term" button and a table for managing credit terms:

Credit Term ID	Overdue Time Period (M)	Blacklisting Time Period (M)	Action
1	2	3	<button>Deactivate</button>
2	3	6	<button>Deactivate</button>

A user profile "Mike" is visible in the top right corner.

Figure 34: Credit Term Management

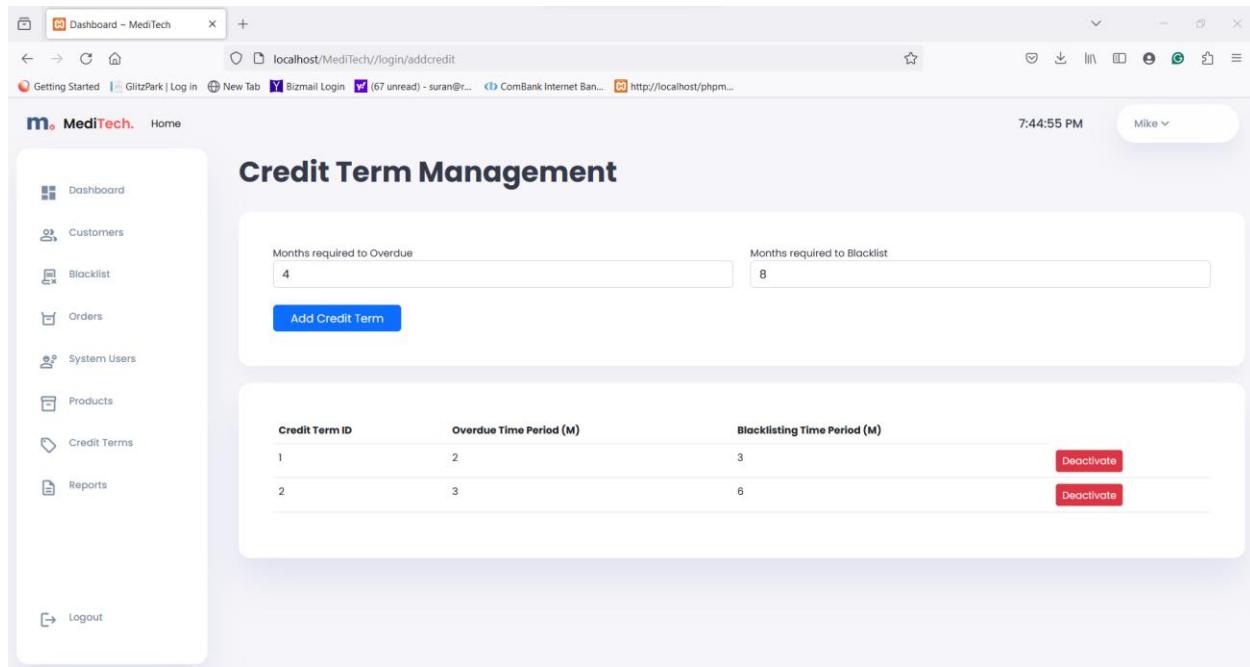


Figure 35: Credit Term Management – Add credit term

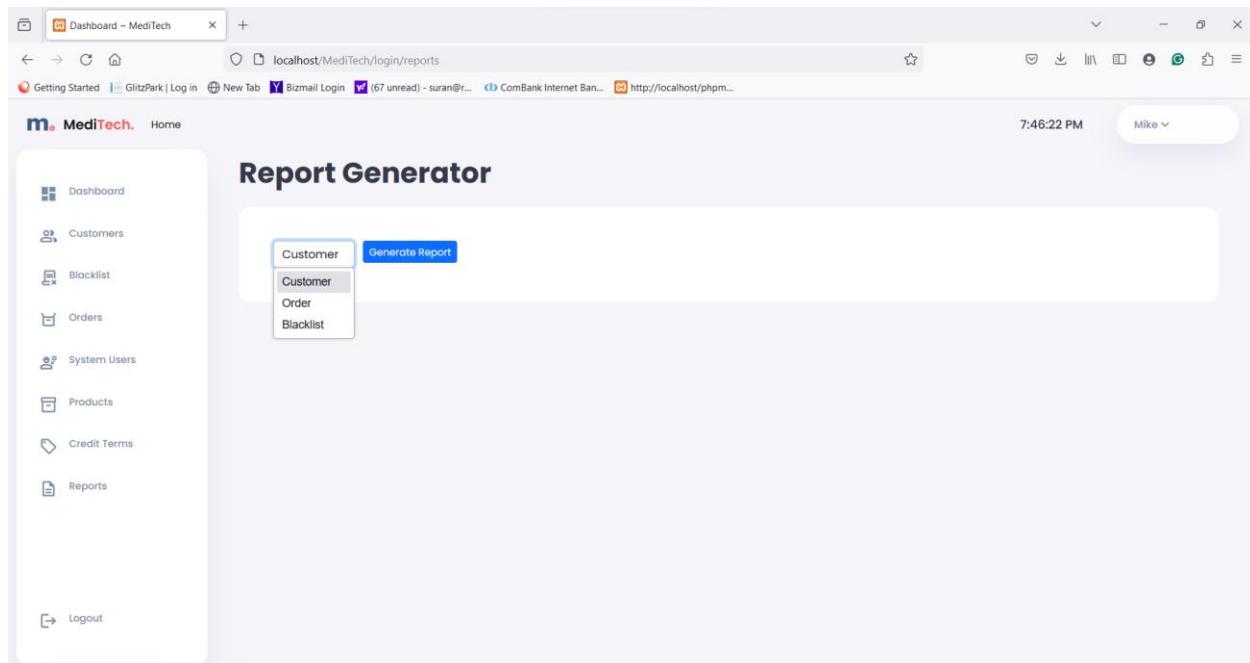


Figure 36: Report Generator

The screenshot shows a web browser window with a title bar "customerpdf" and a URL "localhost/MediTech/reports/customerpdf". The main content area displays a report titled "Customer Details" with a table of data. The table has columns: Customer ID, Customer Name, Email, Contact Number, Status, and Registered Date. The data is as follows:

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	whitelisted	2023-10-28
17	Neranji Abeyratne	hiruka.stry@gmail.com	2147483647	blacklisted	2023-10-28
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blacklisted	2023-11-22
33	Hiruka K.	hiruka.karasinghe@gmail.com	718335411	blacklisted	2023-11-22
35	Gobbel Rowel	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-11-01

Figure 37: Report Generator - success

4. Implementation

4.1 Introduction

Implementation is naturally the most important stage of the software development life cycle. After the designing phase is over, developer will receive a finalized design document to which they will adhere while developing the system. During this stage, developers will turn the design they received into workable code, thus producing the final functioning system. Programming involves many programming languages and database management systems.

4.2 System development tools

4.2.1 Software tools

Front End Design Technologies

- HTML
- CSS
- Java Script

Back End Development Technologies

- PHP 7
- JQuery
- Ajax

Frameworks

- Front End Design -Bootstrap
- Development – CodeIgnitor 3 Web Framework using MVC framework

Database Technologies

- MySQL 5.5.2.7

Server

- XAMPP 3.2.2

Editing Tool

- Visual Code

Operating System

- Windows 10 Pro

Web Browsers

- Latest version of Firefox or Chrome

HARDWARE TOOLS

- Intel Pentium core i7.0 GHz Processor

- 4GB RAM

- 1 VGA Card

- 500 GB Hard Disk

- Network Adapter

- Internet Connection

- Monitor

- Keyboard

- Mouse

4.3 Selecting program languages and styles

The process of selecting a programming language for particular project is a hard task as every programming language has its own set of strengths and weakness making it nearly impossible to assess what programming languages are better than the rest. Therefore, the choice of the programming language should be done with care to better suit the project requirements.

4.3.1 Language and package used

4.3.1.1 Languages

1. PHP & HTML
2. CSS
3. JavaScript
4. Bootstrap
5. jQuery & Ajax
6. MySQL 5.2.0

The programming languages mainly used for the coding was PHP with MySQL. The reason for selecting PHP as the main programming language for this project is:

- PHP is an open source web development language with high performance and reliability.
- It is platform independent language which is a major advantage so that it can run on various platform such as Windows, Unix, Linux and it supports all the main browsers namely google chrome, Firefox and etc.
- PHP is compatible with wide range of servers like Microsoft IIS, apache and Netscape Servers.
- Most of the new developers use PHP as the Server-Side scripting language for web development because it is easy to learn and runs effectively and efficiently on the server.
- PHP takes few lines of codes while other languages needs longer lines of codes for the same work, therefore it significantly reduces the coding time of a programmer.
- Supports for various types of database such as Oracle, SQL Server, MS Access and MySQL further enhance support for MySQL with a new MySQL extension, it has built-in database connection modules which can connect to database easily using PHP. The system was developed using the following programming/scripting languages and software packages.

4.3.1.2 Packages

1. XAMPP 3.3.0

XAMPP is a free and open-source cross-platform web server solution developed by the Apache team. It mainly consists of the Apache HTTP Server, MariaDB database and relevant interpreters for scripts written in PHP and Perl languages.

2. Visual Studio Code IDE

VS Code is an integrated development environment that has been used for the development of the system.

4.4 Selecting development framework

The MediTech credit-based order management system is developed using the model view controller (MVC) framework. CodeIgniter is the MVC based framework which is used by PHP and is used for the development of the system.

4.4.1 Model-View-Controller framework

MVC is an architectural pattern which works by splitting the application into three major components as Model, View and Controller. (medium.com)

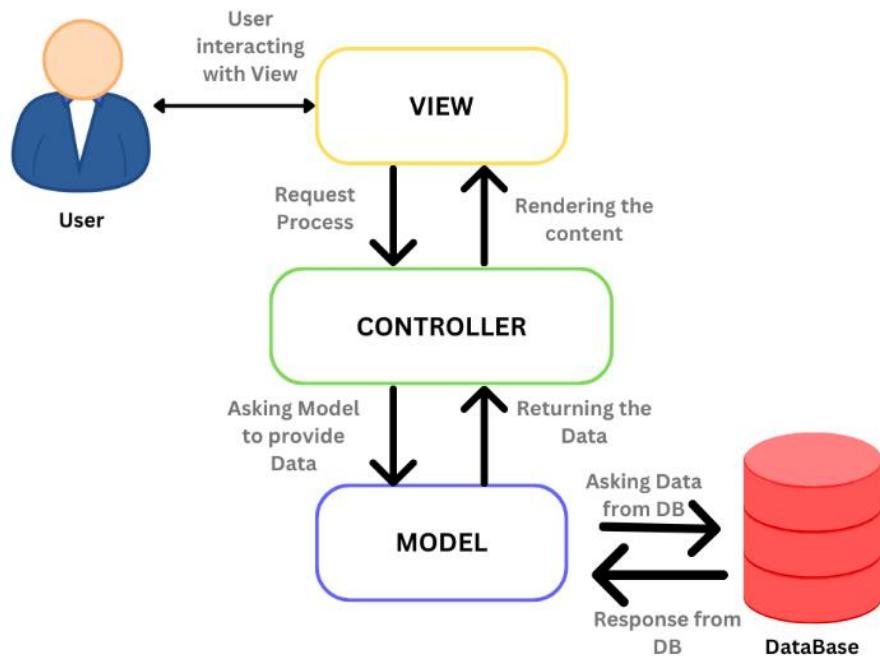


Figure 38: MVC process

The benefits of using the MVC framework are as follows:

- Loose coupling -the components exist and function independently of each other.
- Flexibility – one can easily make changes to individual components
- Increased productivity – multiple people can work on the project at the same time.

4.5 CodeIgniter Framework

4.5.1 Introduction

CodeIgniter simplifies PHP syntax and streamlines application source code on your web pages, which will help reduce development time and repetitive scripting of code blocks. CodeIgniter is a PHP driven framework and contains libraries, plug-ins, helpers and other resources that help developers solve complex PHP functions easily (www.javatpoint.com)

4.5.1.1 CodeIgniter Controller

```
public function userMng()
{
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchUserDB();

        if ($result) {
            $data['user'] = $result;
            $this->load->view('user-management/user-management', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}
```

4.5.1.2 CodeIgniter Model

```
public function fetchUserDB()
{
    $query = $this->db->query("SELECT * FROM `users`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}
```

4.5.1.3 CodeIgniter View

```
<div class="container">
<div class="row">
    <div class="mt-4 innercont p-5 me-4">
        <div class="d-flex mb-3">
            <a href="=site_url(); ?/login/adduser" class="btn btn-primary">Add user</a>
        </div>
        <!-- table view -->
        <table class="table">
            <thead>
                <tr>
                    <th>User ID</th>
                    <th>User Name</th>
                    <th>Branch ID</th>
                    <th>Contact Number</th>
                    <th>Email</th>
                    <th>User Type</th>
                    <th>Registered Date</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($user as $row) : ?>
                <tr>
                    <td><?php echo $row->userid; ?></td>
                    <td><?php echo $row->name; ?></td>
                    <td><?php echo $row->branchID; ?></td>
                    <td><?php echo $row->contactNo; ?></td>
                    <td><?php echo $row->email; ?></td>
                    <td><?php echo $row->userType; ?></td>
                    <td>
                        <div class="flex">
                            <a href="=site_url(); ?/login/editUser/<?php echo $row->userid; ?>" class="btn btn-success">
                                <a href="=site_url(); ?/login/delUser/<?php echo $row->userid; ?>" class="btn btn-danger">
                                    </div>
                            </td>
                        </div>
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
```

4.6 Database Implementation

4.6.1 Introduction

The relational data model is a second generation Data Base Management System which was used for data modelling database structures. MySQL was used for the development of the database.

4.6.2 Relational Database

The relational data model helps the data to be presented in a table format with regards to their logical relationships. This overcomes the problem of physical address handling. It was formulated upon the mathematical set theory and has a very robust mathematical foundation. Relational Databases needs to possess 3 main factors ;

1. Data structure – Data are stored a tabular format.
2. Data Manipulation – Powerful operations are made to manipulate data.
3. Data integrity – Facilitates data integrity.

4.6.3 Database Development

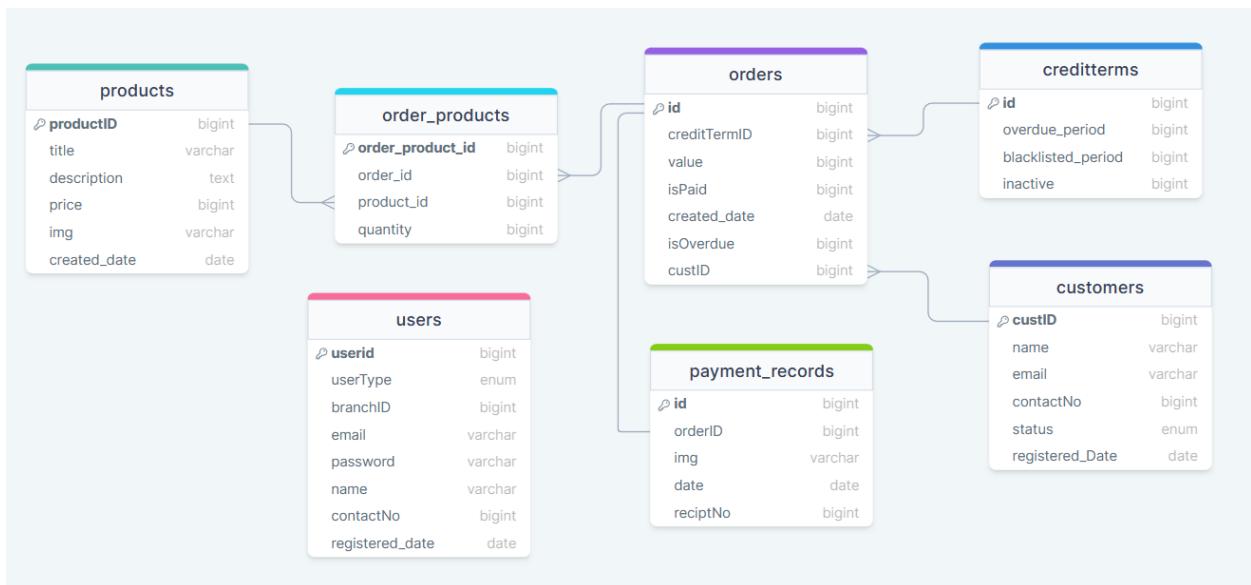


Figure 39: Database Design

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** meditech
- Toolbar:** Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers.
- Filters:** A search bar containing "Containing the word:" followed by an empty input field.
- Table List:**

Table	Action	Rows	Type	Collation	Size	Overhead
creditterms	Browse, Structure, Search, Insert, Empty, Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customers	Browse, Structure, Search, Insert, Empty, Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
orders	Browse, Structure, Search, Insert, Empty, Drop	7	InnoDB	utf8mb4_general_ci	48.0 KiB	-
order_products	Browse, Structure, Search, Insert, Empty, Drop	11	InnoDB	utf8mb4_general_ci	48.0 KiB	-
payment_records	Browse, Structure, Search, Insert, Empty, Drop	6	InnoDB	utf8mb4_general_ci	32.0 KiB	-
products	Browse, Structure, Search, Insert, Empty, Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	Browse, Structure, Search, Insert, Empty, Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
7 tables	Sum	42	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B
- Buttons:** Up, Check all, With selected: dropdown menu.

Figure 40: Database Main table

The screenshot shows the MySQL Workbench interface with the following details:

- Table:** creditterms
- Columns:** id, overdue_period, blacklisted_period, inactive
- Data:**

		id	overdue_period	blacklisted_period	inactive
<input type="checkbox"/>	<input type="checkbox"/> Edit	1	2	3	0
<input type="checkbox"/>	<input type="checkbox"/> Edit	2	3	6	0
<input type="checkbox"/>	<input type="checkbox"/> Edit	4	6	2	1
<input type="checkbox"/>	<input type="checkbox"/> Edit	5	3	8	1
- Buttons:** Up, Check all, With selected: dropdown menu, Edit, Copy, Delete, Export.

Figure 41: Credit Terms Table

The screenshot shows the MySQL Workbench interface with the following details:

- Table:** customers
- Columns:** custID, name, email, contactNo, status, registered_date
- Data:**

		custID	name	email	contactNo	status	registered_date
<input type="checkbox"/>	<input type="checkbox"/> Edit	2	hiruka	hiruka.karasinghe@gmail.com	718398731	whitelisted	2023-10-28
<input type="checkbox"/>	<input type="checkbox"/> Edit	17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blacklisted	2023-10-28
<input type="checkbox"/>	<input type="checkbox"/> Edit	29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
<input type="checkbox"/>	<input type="checkbox"/> Edit	31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blacklisted	2023-11-22
<input type="checkbox"/>	<input type="checkbox"/> Edit	33	Hiruka K	hiruka.karasinghe@gmail.com	718335411	blacklisted	2023-11-22
<input type="checkbox"/>	<input type="checkbox"/> Edit	35	Gobbel Rowel	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-11-01
- Buttons:** Up, Check all, With selected: dropdown menu, Edit, Copy, Delete, Export.

Figure 42: Customer Table

	<input type="button" value="← T →"/>	<input type="button" value="▼ id"/>	<input type="button" value="custID"/>	<input type="button" value="creditTermID"/>	<input type="button" value="value"/>	<input type="button" value="isPaid"/>	<input type="button" value="created_date"/>	<input type="button" value="isOverdue"/>
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	31	2	12500	1 2023-11-12 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	2	2	13800	1 2023-06-23 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	31	1	1000	1 2023-11-12 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	19	35	1	31200	1 2022-11-08 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	20	35	1	9000	1 2023-01-18 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	21	2	2	12000	0 2023-12-18 0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	22	29	1	6250	0 2024-01-15 0

Check all With selected:

Figure 43: Order Table

	<input type="button" value="← T →"/>	<input type="button" value="▼ order_product_id"/>	<input type="button" value="order_id"/>	<input type="button" value="product_id"/>	<input type="button" value="quantity"/>		
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	16	29	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	16	28	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	17	30	6
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	18	28	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	19	30	9
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	19	29	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	19	28	6
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	20	29	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	21	29	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	22	28	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	22	29	2

Check all With selected:

Figure 44: Order_Products Table

	<input type="button" value="← T →"/>		<input type="button" value="▼"/>	id	orderID	img	date	receiptNo
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	16	1699780242_344024.jpeg	2023-11-12	269044
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	18	1699790378_630194.jpeg	2023-11-12	269045
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	17	1702833424_735398.png	2023-12-17	269044
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	20	1702911157_267191.png	2023-12-18	269044
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	19	1702911185_852583.png	2023-12-18	269044
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	17	1705303388_354799.png	2024-01-15	43534536

Check all With selected:

Figure 45: Payment_records table

	<input type="button" value="← T →"/>		<input type="button" value="▼"/>	userid	userType	branchID	email	password	name	contactNo	registered_date
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	Admin	1	hiruka.karasinghe@gmail.com	Hiruka	Mike	718398731	2023-10-26
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	Sales	2	hiruka.sales@gmail.com	Hiruka	Rowel	718398731	2023-10-28
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	DataEntry	2	hiruka entry@gmail.com	Hiruka	Gobbels	718398731	2023-10-28
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	DataEntry	2	hiruka.karasinghe@gmail.com	Hiruka	Pinindi Edited	93764901	2024-01-15

Check all With selected:

Figure 46: Users table

	<input type="button" value="← T →"/>		<input type="button" value="▼"/>	productID	title	description	price	img	created_date
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	28	Dental MBT Brackets	MBT Brackets, Nickel Titanium	250	1699257988_423078.jpg	2023-11-05
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	29	Hygent Alginate	Alignate Impression material by Hygent	3000	1699258000_622025.jpg	2023-11-05
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	30	Xylocaine	1% Epinephrine Injection	2300	1699258007_509815.jpg	2023-11-05
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	31	Adrenaline 1ml	Dental adrenaline supply of 50 uses of each being ...	5000	1705314012_143450.jpeg	2024-01-15

Check all With selected:

Figure 47: Products Table

4.7 Application Development

4.7.1 Model

```
<?php

class User_model extends CI_Model
{

    // -----Product Model-----

    public function editProductData($id, $data)
    {

        $condition = "productID='{$id}'";
        //$this->db->set('userType', $data['userType']);
        $this->db->set('title', $data['title']);
        $this->db->set('description', $data['description']);
        $this->db->set('img', $data['img']);
        $this->db->set('price', $data['price']);

        $this->db->where($condition);
        $this->db->update('products');

        if ($this->db->affected_rows() == 1) {
            return (1);
        } else if ($this->db->affected_rows() == 0) {
            return (0);
        } else {
            return (-1);
        }
    }

    public function delProduct($id)
    {
        //DELETE FROM customers WHERE `customers`.`custID` = 19
        $condition = "productID='{$id}'";
        $this->db->where($condition);
        $this->db->delete('products');
        if ($this->db->affected_rows() == 1) {
            return true;
        } else {
            return false;
        }
    }

    public function fetchProducts()
    {

        $query = $this->db->query("SELECT * FROM `products`");

        if ($query) {
            return $query->result();
        } else {
            return false;
        }
    }
}
```

```

public function addProduct($data)
{
    $this->db->set('title', $data['title']);
    $this->db->set('description', $data['description']);
    $this->db->set('img', $data['img']);
    $this->db->set('price', $data['price']);

    $query = $this->db->insert('products');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function getProductByID($id)
{
    $condition = "productID='{$id}'";
    $query = $this->db->select('*')
        ->where($condition)
        ->get('products');

    return $query->result();
    if ($query->num_rows() == 1) {
        return $query->result();
    } else {
        return false;
    }
}

// -----Customer Model-----

public function fetchBlacklistedCustomers()
{
    $query = $this->db->query("SELECT * FROM `customers` WHERE `status` = 'blacklisted'");
    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchCustomerDB()
{
    $query = $this->db->query("SELECT * FROM `customers`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

public function fetchWhitelistedCustomerDB()
{
    $condition = "`status` = 'whitelisted'";

    $query = $this->db->select('*')
        ->where($condition, NULL, FALSE)
        ->get('customers');

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function getCustomerDataByID($id)
{
    $condition = "custID='{$id}'";
    $query = $this->db->select('*')
        ->where($condition)
        ->get('customers');

    return $query->result();
    if ($query->num_rows() == 1) {
        return $query->result();
    } else {
        return false;
    }
}

public function getCustomerNameByID($id)
{
    $condition = "custID='{$id}'";

    $query = $this->db->select('*')
        ->where($condition)
        ->get('customers');

    return $query->result();
    if ($query->num_rows() == 1) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchOrderNames()
{
    $query = $this->db->query("SELECT `name` FROM `customers`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

```

4.7.2 Controller

```
<?php

class Login extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
        $this->load->helper('form');
        $this->load->library('form_validation');
        $this->load->model('user_model');
        $this->load->model('data_model');
        // $this->load->controller('mail');
        $this->load->library('session');
        $this->load->helper('url');

        date_default_timezone_set("Asia/colombo");

        $this->load->library('upload');
        $this->load->library('phpmailer_lib');
        $this->load->library('email');
        // $this->load->library('openai');
    }

    function sendReminder($name, $amount, $date, $email)
    {
        $config = array(
            'protocol' => 'smtp',
            'smtp_host' => 'smtp.office365.com',
            'smtp_port' => 587,
            'smtp_user' => 'sales@meditech.work',
            'smtp_pass' => 'meditech@123',
            'smtp_crypto' => 'tls',
            'mailtype' => 'html',
            'charset' => 'utf-8',
            'newline' => "\r\n",
        );
        $this->email->initialize($config);

        $this->email->from('sales@meditech.work', 'MediTech');
        $this->email->to($email);
        $this->email->subject('Overdue payment notification | MediTech');

        $reminder = "A credit order placed on $date from Meditech has been overdue recently. The amount
overdue is $amount. Please make the necessary payment timely to avoid being blacklisted from our
services.";
        // reminder configurations

        $mailContent = "
<h1>Overdue Order Notice</h1>
<h3>Hi $name,</h3>
<p>$reminder</p>
<p>Thank you for your attention.</p>
<p>MediTech</p>";
        // reminder message

        $this->email->message($mailContent);

        if ($this->email->send()) {
            echo 'Email sent.';
        } else {
            echo 'Email could not be sent.';
            echo $this->email->print_debugger();
        }
        // sending message / capturing errors
    }
}
```

```

public function index()
{
    $this->load->view('auth/login');
}

public function register()
{
    $this->load->view('auth/register');
}

public function profile()
{
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    //load user profile page
    $sessionData = $this->session->userdata('userinfo');

    if ($this->checkSessionExist()) {

        $result = $this->user_model->getUserDataByID($sessionData['id']);
        if ($result) {
            $data['myprofile'] = $result;
            $this->load->view('profile', $data);
        }
    }
}

public function editProfile()
{
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    //loads user profile edit page
    $sessionData = $this->session->userdata('userinfo');

    if ($this->checkSessionExist()) {

        $result = $this->user_model->getUserDataByID($sessionData['id']);
        if ($result) {
            $data['myprofile'] = $result;
            $this->load->view('edit-profile', $data);
        }
    }
}

```

```

public function editProfileSubmit()
{
    //user profile edit submission function
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('name', 'Name', 'required');
    $this->form_validation->set_rules('contact', 'Contact', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect("/login/editProfile/");
    } else {

        $sessionData = $this->session->userdata('userinfo');
        $data = array(
            'id' => $sessionData['id'],
            'name' => $_POST['name'],
            'email' => $_POST['email'],
            'contact' => $_POST['contact']
        );

        //submission of the edits
        $result = $this->user_model->updateProfile($data);

        if ($result == 1) {
            $this->session->set_flashdata('success', 'Profile data updated successfully.');
            redirect("/login/profile/");
        } elseif ($result == 0) {
            $this->session->set_flashdata('success', 'Profile data upto date.');
            redirect("/login/profile/");
        } else {
            $this->session->set_flashdata('error', 'Error occured please try again');
            redirect("/login/editProfile/");
        }
    }
}

public function userlogin()
{
    //login page controller
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    //error
    if (isset($data['error']) || isset($data['success'])) {
        $this->load->view('auth/login', $data);
    } else {
        if ($this->checkSessionExist()) {
            $this->load->view('dashboard', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

```

4.7.3 View

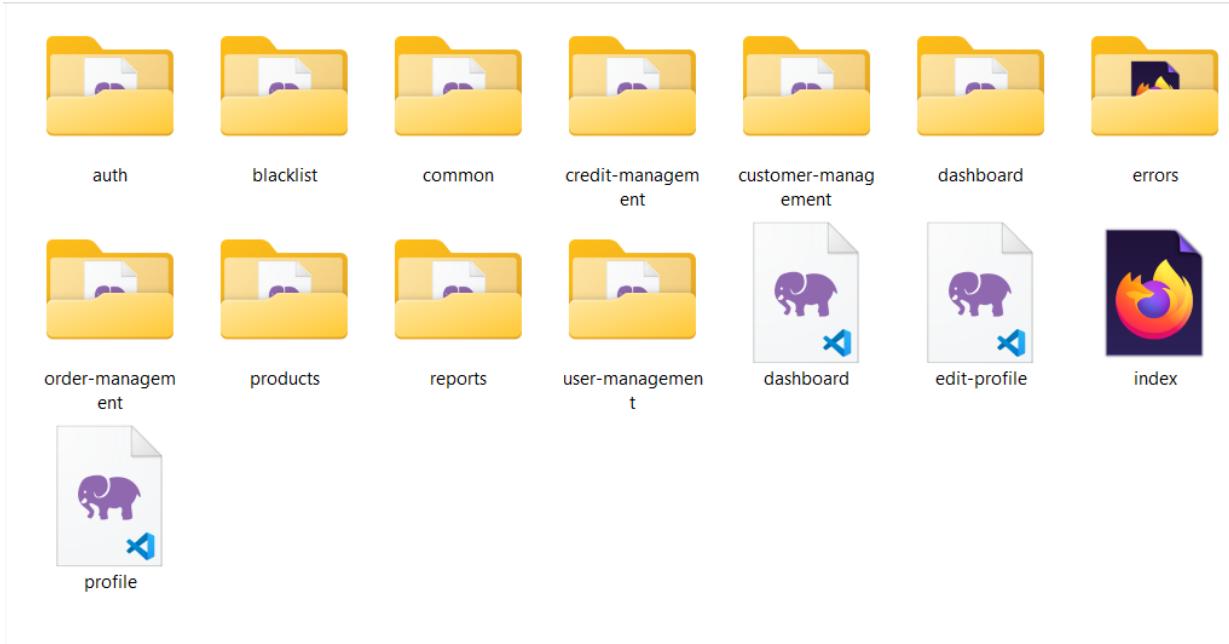


Figure 48: Project View Structure

4.7.3.1 Use of HTML (Hypertext Markup Language)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login ~ MediTech</title>
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/bootstrap.min.css">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/custom/login.css">
</head>

<body>
    <div class="container d-flex justify-content-center align-items-center min-vh-100">
        <div class="row border rounded-5 p-3 bg-white shadow box-area">
            <div class="col-md-6 rounded-4 d-flex justify-content-center align-items-center left-box flex-column" style="background: #f0f0f0; background-color: #f0f0f0; color: #f0f0f0; font-weight: bold; font-size: 1.2em; text-align: center; padding: 10px; margin-bottom: 20px;">
                <div class="featured-image">
                    <img src=<?php echo site_url(); ?>images/logo-transparent.png class="img-fluid" style="width: 250px;">
                </div>
            </div>
            <div class="col-md-6 right-box">
                <div class="row align-items-center">
                    <div class="header-text mt-4 ps-4">
                        <p class="fs-5 nice-font">Login</p>
                        <p class="fs-6">Hello, happy to have you back!</p>
                    </div>
                    <div class="reserved">
                        <?php
                            if (isset($success)) {
                                echo "<div class='alert alert-success px-6'>";
                                echo $success;
                                echo "</div>";
                            }
                            if (isset($error)) {
                                echo "<div class='alert alert-danger px-6'>";
                                echo $error;
                                echo "</div>";
                            }
                        ?>
                    <?php echo validation_errors('<div class="alert alert-danger px-6">', '</div>'); ?>
                </div>
                <?php echo form_open('login/loginSubmit') ?>
                <table class="table table-borderless">
                    <tr>
                        <td>
```

```
                <input class="form-control" type="text" name="email"
placeholder="Login Email">
                </td>
            </tr>
            <tr>
                <td>
                    <input class="form-control" type="password" name="password"
placeholder="Password">
                </td>
            </tr>
            <tr>
                <td>
                    <input class="btn btn-primary" type="submit" name="submit"
value="Login">
                </td>
            </tr>
        </table>
        <?php echo form_close(); ?>
    </div>
</div>
</div>
</body>
</html>
```

4.7.3.2 Use of CSS (Cascading Style Sheet)

```
<style>
  .skills {
    width: 100%;
    padding: 0;
  }

  .skill-name {
    font-size: 13px;
    font-weight: 500;
    color: #425A82;
    margin: 20px 0;
  }

  .skill-bar {
    height: 14px;
    background: #b3bfce;
    border-radius: 3px;
  }

  .skill-per {
    height: 14px;
    background: #425A82;
    border-radius: 3px;
    position: relative;
    animation: fillBars 2.5s 1;
  }

  .skill-per::before {
    content: attr(per);
```

```
position: absolute;
padding: 4px 6px;
background: #f1f1f1;
border-radius: 4px;
font-size: 12px;
top: -35px;
right: 0;
transform: translateX(50%);
}

.skill-per::after {
  content: "";
  position: absolute;
  width: 10px;
  height: 10px;
  background: #f1f1f1;
  top: -15px;
  right: 0;
  transform: translateX(50%) rotate(45deg);
  border-radius: 2px;
}

@keyframes fillBars {
  from {
    width: 0;
  }

  to {
    width: 100%;
  }
}
</style>
```

4.7.3.3 Use of Bootstrap

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet"
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/custom/nav.css">
    <link href=<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/nav.php'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">

            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Blacklist</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>
            </div>
        </div>
    </div>
</body>
```

4.7.3.4 Use of JavaScript

```
<script>
let selectedValue = document.getElementById("productDropdown").value;
//console.log(<?php echo $selectedMonth ?>);
// Function to update the href attribute of the "Add Product" link
function changeMonth() {
    const dropdown = document.getElementById("productDropdown");
    const newSelectedValue = dropdown.value;

    console.log(selectedValue);
    //const selectedProduct = $products.find(product => product.productID === selectedValue);
    //console.log(selectedValue);
    //console.log(selectedProduct);

    if (newSelectedValue !== selectedValue) {
        selectedValue = newSelectedValue;

        const data = {
            value: selectedValue
        };

        // Create an XMLHttpRequest object
        $.ajax({
            url: "<?php echo site_url(); ?>login/refreshdash/",
            data: data,
            method: "GET",
            success: function(response) {

                console.log(response);
                window.location.href = "<?php echo site_url(); ?>login/dashboard/" + selectedValue;
                //break;
            },
            error: function(jqXHR, textStatus, errorThrown) {
                console.log("AJAX Request Failed: " + errorThrown);
            }
        });
    }
}

$("#productDropdown").change(changeMonth);

//changeMonth();
</script>
```

4.7.3.5 Error Pages

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

echo "\nERROR: ",
$heading,
"\n\n",
$message,
"\n\n";
```

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

echo "\nDatabase error: ",
$heading,
"\n\n",
$message,
"\n\n";
```

```
<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

An uncaught Exception was encountered

Type:      <?php echo get_class($exception), "\n"; ?>
Message:   <?php echo $message, "\n"; ?>
Filename:  <?php echo $exception->getFile(), "\n"; ?>
Line Number: <?php echo $exception->getLine(); ?>

<?php if (defined('SHOW_DEBUG_BACKTRACE') && SHOW_DEBUG_BACKTRACE === TRUE): ?>

Backtrace:
<?php  foreach ($exception->getTrace() as $error): ?>
<?php    if (isset($error['file']) && strpos($error['file'], realpath(BASEPATH)) !== 0): ?>
    File: <?php echo $error['file'], "\n"; ?>
    Line: <?php echo $error['line'], "\n"; ?>
    Function: <?php echo $error['function'], "\n\n"; ?>
<?php    endif ?>
<?php  endforeach ?>

<?php endif ?>
```

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

echo "\nERROR: ",
$heading,
"\n\n",
$message,
"\n\n";
```

```
<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

A PHP Error was encountered

Severity:    <?php echo $severity, "\n"; ?>
Message:     <?php echo $message, "\n"; ?>
Filename:    <?php echo $filepath, "\n"; ?>
Line Number: <?php echo $line; ?>

<?php if (defined('SHOW_DEBUG_BACKTRACE') && SHOW_DEBUG_BACKTRACE === TRUE): ?>

Backtrace:
<?php foreach (debug_backtrace() as $error): ?>
<?php     if (isset($error['file']) && strpos($error['file'], realpath(BASEPATH)) !== 0): ?>
    File: <?php echo $error['file'], "\n"; ?>
    Line: <?php echo $error['line'], "\n"; ?>
    Function: <?php echo $error['function'], "\n\n"; ?>
<?php     endif ?>
<?php endforeach ?>

<?php endif ?>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>403 Forbidden</title>
</head>
<body>

<p>Directory access is forbidden.</p>

</body>
</html>
```

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?><!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>404 Page Not Found</title>
<style type="text/css">

::selection { background-color: #E13300; color: white; }
::-moz-selection { background-color: #E13300; color: white; }

body {
    background-color: #fff;
    margin: 40px;
    font: 13px/20px normal Helvetica, Arial, sans-serif;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 19px;
    font-weight: normal;
    margin: 0 0 14px 0;
    padding: 14px 15px 10px 15px;
}

code {
    font-family: Consolas, Monaco, Courier New, Courier, monospace;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
    padding: 12px 10px 12px 10px;
}

#container {
    margin: 10px;
    border: 1px solid #D0D0D0;
    box-shadow: 0 0 8px #D0D0D0;
}

p {

```

```

        margin: 12px 15px 12px 15px;
    }

```

```

</style>
</head>
<body>
    <div id="container">
        <h1><?php echo $heading:_?></h1>
        <?php echo $message:_?>
    </div>
</body>
</html>

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?><!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Database Error</title>
<style type="text/css">

::selection { background-color: #E13300; color: white; }
::-moz-selection { background-color: #E13300; color: white; }

body {
    background-color: #fff;
    margin: 40px;
    font: 13px/20px normal Helvetica, Arial, sans-serif;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 19px;
    font-weight: normal;
    margin: 0 0 14px 0;
    padding: 14px 15px 10px 15px;
}

code {
    font-family: Consolas, Monaco, Courier New, Courier, monospace;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
}

```

```

        padding: 12px 10px 12px 10px;
    }

#container {
    margin: 10px;
    border: 1px solid #D0D0D0;
    box-shadow: 0 0 8px #D0D0D0;
}

p {
    margin: 12px 15px 12px 15px;
}
</style>
</head>
<body>
    <div id="container">
        <h1><?php echo $heading; ?></h1>
        <?php echo $message; ?>
    </div>
</body>
</html>

```

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>

<div style="border:1px solid #990000;padding-left:20px;margin:0 0 10px 0;">

<h4>An uncaught Exception was encountered</h4>

<p>Type: <?php echo get_class($exception); ?></p>
<p>Message: <?php echo $message; ?></p>
<p>Filename: <?php echo $exception->getFile(); ?></p>
<p>Line Number: <?php echo $exception->getLine(); ?></p>

<?php if (defined('SHOW_DEBUG_BACKTRACE') && SHOW_DEBUG_BACKTRACE === TRUE): ?>

    <p>Backtrace:</p>
    <?php foreach ($exception->getTrace() as $error): ?>

        <?php if (isset($error['file']) && strpos($error['file'], realpath(BASEPATH)) !== 0): ?>

            <p style="margin-left:10px">
                File: <?php echo $error['file']; ?><br />
                Line: <?php echo $error['line']; ?><br />
                Function: <?php echo $error['function']; ?>
            </p>
        <?php endif ?>

    <?php endforeach ?>

<?php endif ?>

</div>

```

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?><!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
<style type="text/css">

::selection { background-color: #E13300; color: white; }
::-moz-selection { background-color: #E13300; color: white; }

body {
    background-color: #fff;
    margin: 40px;
    font: 13px/20px normal Helvetica, Arial, sans-serif;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 19px;
    font-weight: normal;
    margin: 0 0 14px 0;
    padding: 14px 15px 10px 15px;
}
```

```
code {
    font-family: Consolas, Monaco, Courier New, Courier, monospace;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
    padding: 12px 10px 12px 10px;
}

#container {
    margin: 10px;
    border: 1px solid #D0D0D0;
    box-shadow: 0 0 8px #D0D0D0;
}

p {
    margin: 12px 15px 12px 15px;
}
</style>
</head>
<body>
    <div id="container">
        <h1><?php echo $heading; ?></h1>
        <?php echo $message; ?>
    </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>403 Forbidden</title>
</head>
<body>

    <p>Directory access is forbidden.</p>

</body>
</html>
```

4.6 Documentation

Documentation is very significant for any development process. It outlines the system's body as a whole and explains its functionalities, its designs, its usability and many more. It is also very important to have during maintenance and after care of the system. Producing Documentation is absolutely necessary.

The types of documents are as follows

Requirements documentation: outlining the functional and non-functional requirements of the system

Architectural documentation: contains the design of the system

Technical documentation: focuses on the technical aspects of the system.

End-user documentation: this is prepared for the end users and provides users with the necessary help regarding system navigation.

As the development team developed the Credit Based Order Management system, all of these reports were produced.

5 Testing

5.1 Testing and verification

Testing Is the process of identifying whether a system performs the functions it was intended to perform in an accurate and reliable manner. This involves executing a program with the intention of finding errors and is a critical element of the software development lifecycle.

Functional Testing – checks the application to ensure that it performs that task that its supposed to do.

Non-functional testing – performance, usability, reliability, etc are all tested under non-functional testing.

Test cases were run on the Credit-Based Order Management system to ensure that the intended software was delivered to the customer in an error free manner.

During the test process of the newly developed system, the following testing methods were utilized:

Unit testing

Integration testing

System testing

5.2 Test Cases

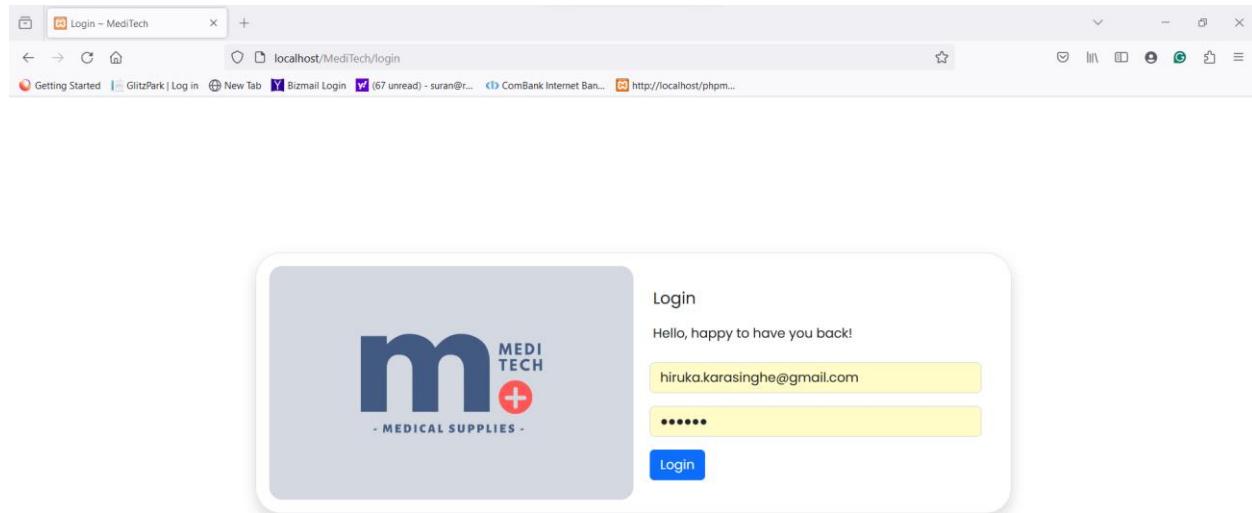


Figure 49: Pre-Condition – Login

Test Case ID	TC_01
Description	Verify that user can successfully login to the application
Pre-Condition	User should open the application
Test Procedure	Enter user email Enter password of account Click login button
Expected Result	Dashboard should open
Actual Result	Dashboard opens
Status	Passed

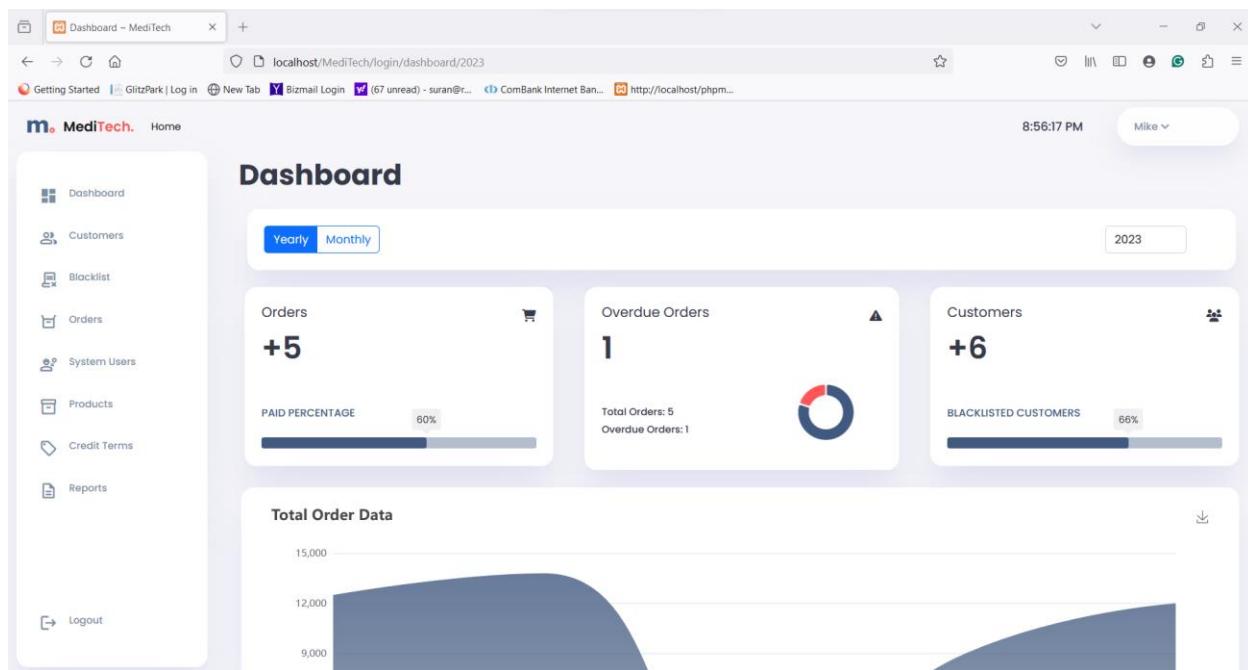


Figure 50: Post-Condition – Dashboard

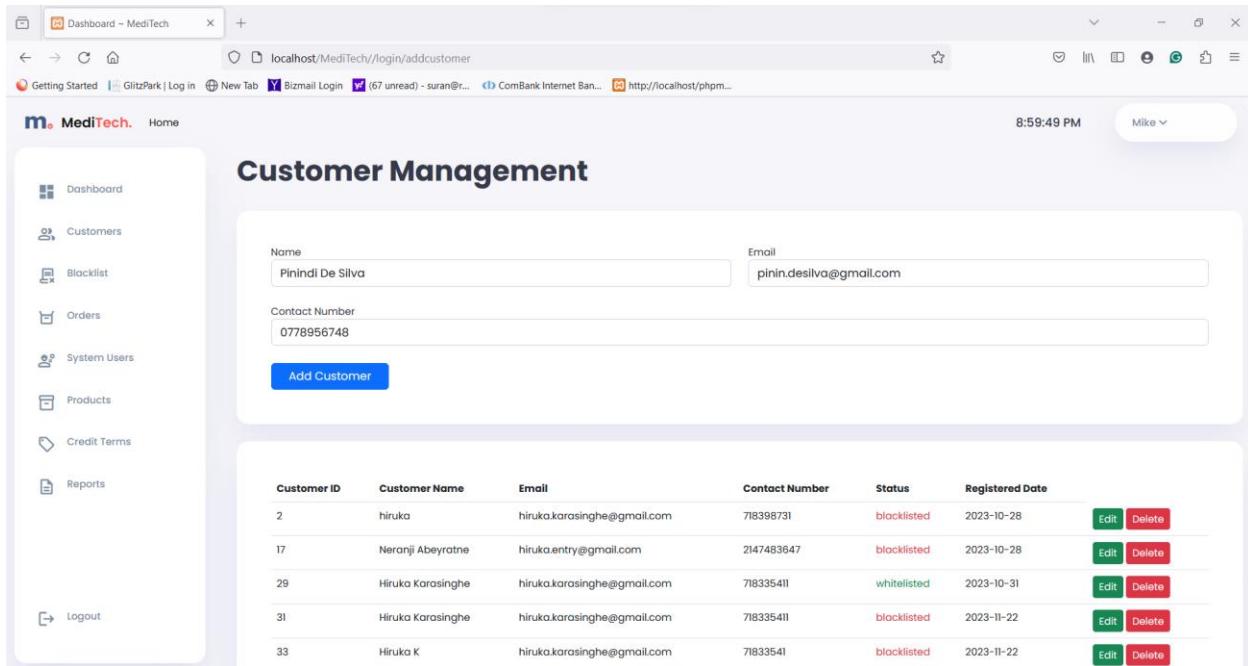


Figure 51: Pre-Condition – Customer Management Module

Test Case ID	TC_02
Description	Verify that a new customer can be added
Pre-Condition	User should be logged into the system and in the customer management module
Test Procedure	<p>Click on "Add customer"</p> <p>Enter Customer details</p> <p>Click add customer button</p>
Expected Result	New Customer should be visible with a success message
Actual Result	New customer is visible with a success message
Status	Passed

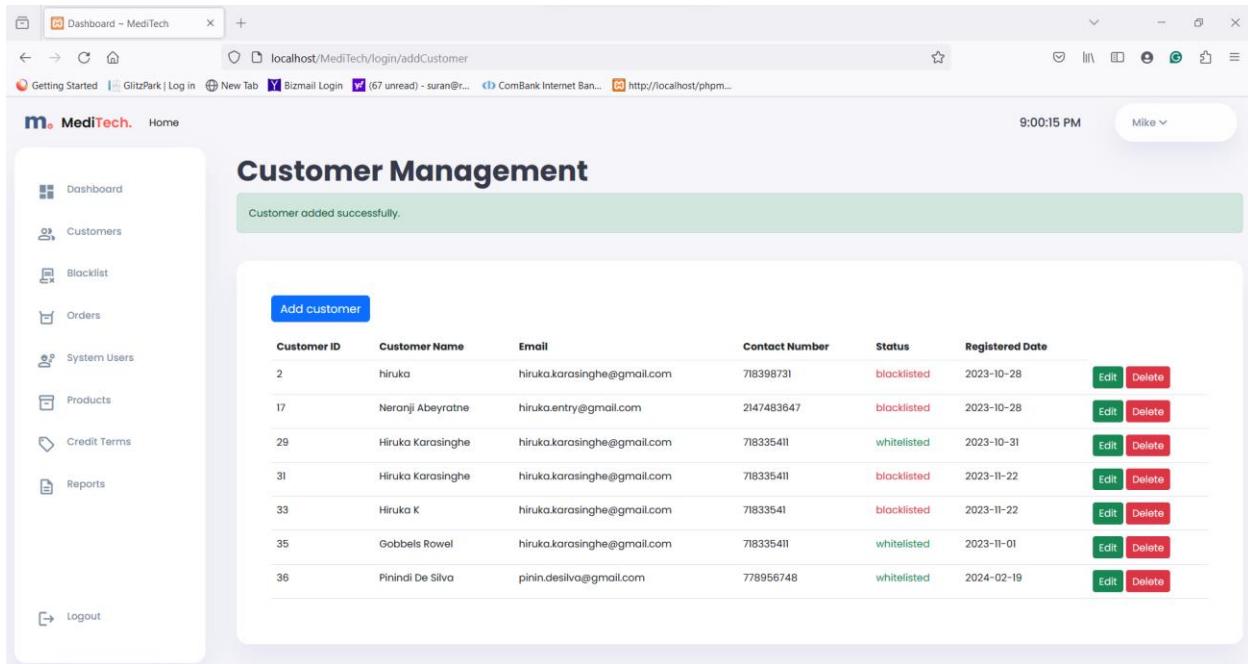


Figure 52: Post-Condition – Customer Management Module

Test Case ID	TC_03
Description	Verify that a Customer details can be edited
Pre-Condition	User should be logged into the system and in the customer management module
Test Procedure	<p>Click on “Edit” next to customer record</p> <p>Enter Edited data</p> <p>Click update customer button</p>
Expected Result	Customer details should be edited In the <u>record</u> with success message
Actual Result	Edited customer records are visible in the records with success message
Status	Passed

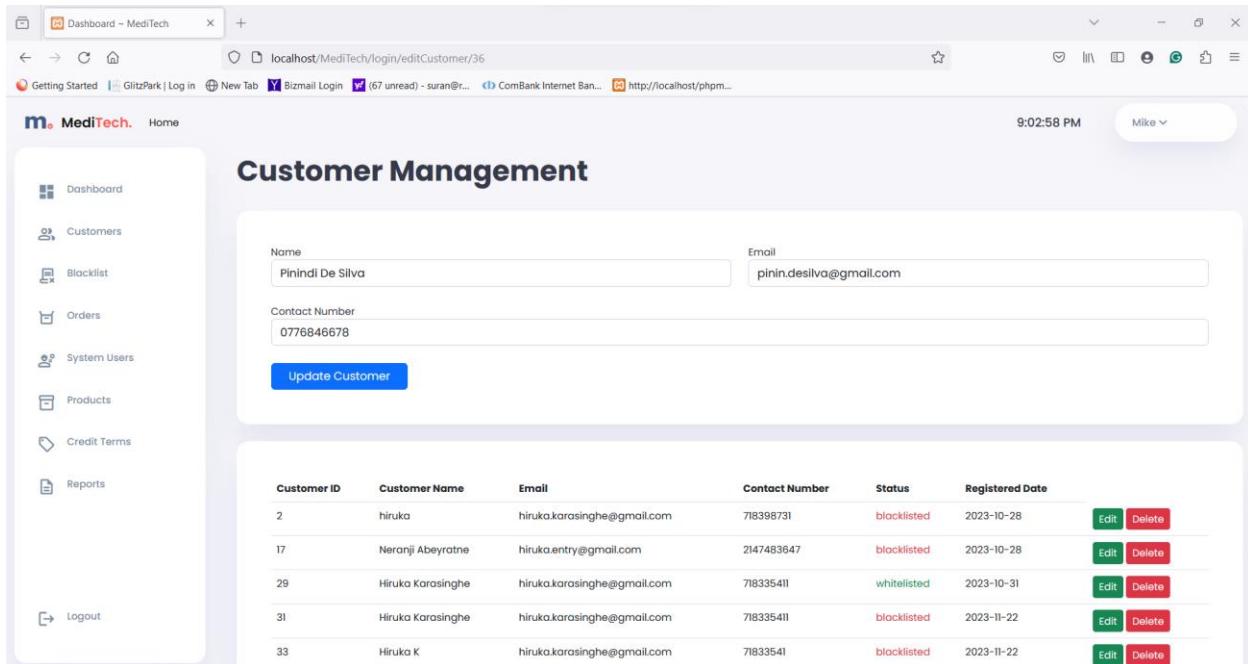


Figure 53: Pre-Condition – Customer Management – Update Customer

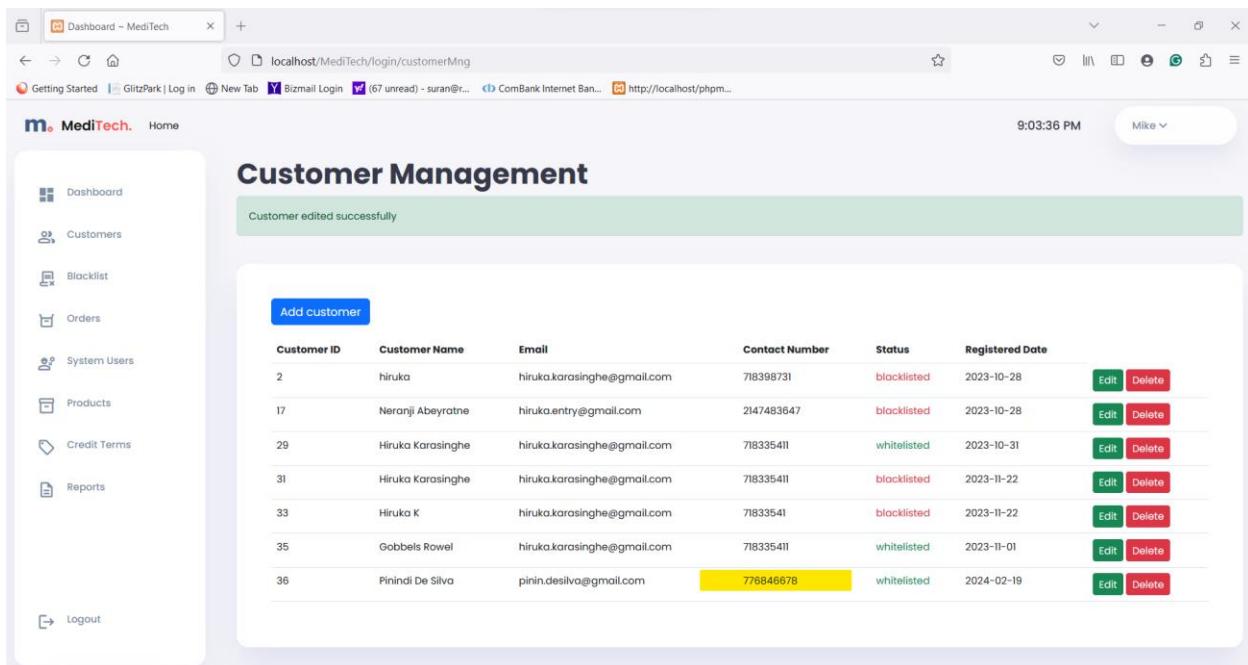


Figure 54: Post-Condition – Customer Management – Update Customer

Test Case ID	TC_04
Description	Verify that a Customer can be deleted
Pre-Condition	User should be logged into the system and in the customer management module
Test Procedure	Click on “Delete” next to customer record
Expected Result	Customer details should be removed from the record with success message
Actual Result	Customer was removed from the records with a success message
Status	Passed

The screenshot shows the MediTech Customer Management interface. On the left is a sidebar with navigation links: Dashboard, Customers (selected), Blocklist, Orders, System Users, Products, Credit Terms, and Reports. The main area has a title 'Customer Management' and a sub-section 'Add customer'. Below this is a table listing 8 customers:

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date	Action
2	hiruka	hiruka.karasinghe@gmail.com	718398731	blocklisted	2023-10-28	<button>Edit</button> <button>Delete</button>
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blocklisted	2023-10-28	<button>Edit</button> <button>Delete</button>
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31	<button>Edit</button> <button>Delete</button>
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22	<button>Edit</button> <button>Delete</button>
33	Hiruka K	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22	<button>Edit</button> <button>Delete</button>
35	Gobbels Rowel	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-11-01	<button>Edit</button> <button>Delete</button>
37	Pinindi De Silva	pinin.desilva@gmail.com	778956748	whitelisted	2024-02-19	<button>Edit</button> <button>Delete</button>

At the bottom left of the sidebar is a 'Logout' link.

Figure 55: Pre-Condition – Customer Management – Delete Customer

The screenshot shows a web browser window for 'Dashboard - MediTech' at the URL 'localhost/MediTech/login/customerMng'. The page title is 'Customer Management'. A green success message box displays 'Record deleted successfully.' Below this, a table lists customer data with columns: Customer ID, Customer Name, Email, Contact Number, Status, and Registered Date. The table contains 6 rows of data. At the bottom of the page, a yellow success message box displays 'Record deleted'.

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	blocklisted	2023-10-28
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blocklisted	2023-10-28
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22
33	Hiruka K	hiruka.karasinghe@gmail.com	718335411	blocklisted	2023-11-22
35	Gobbels Rowel	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-11-01

Figure 56: Post-Condition – Customer Management – Delete Customer

Test Case ID	TC_05
Description	Verify that a new order can be created
Pre-Condition	User should be logged into the system and in the order management module
Test Procedure	<p>Click on “Add order”</p> <p>Select Customer</p> <p>Select products and Click “Add product”</p> <p>Click “Proceed to pay”</p>
Expected Result	New Order should be created with a success message
Actual Result	New order created with a success message
Status	Passed

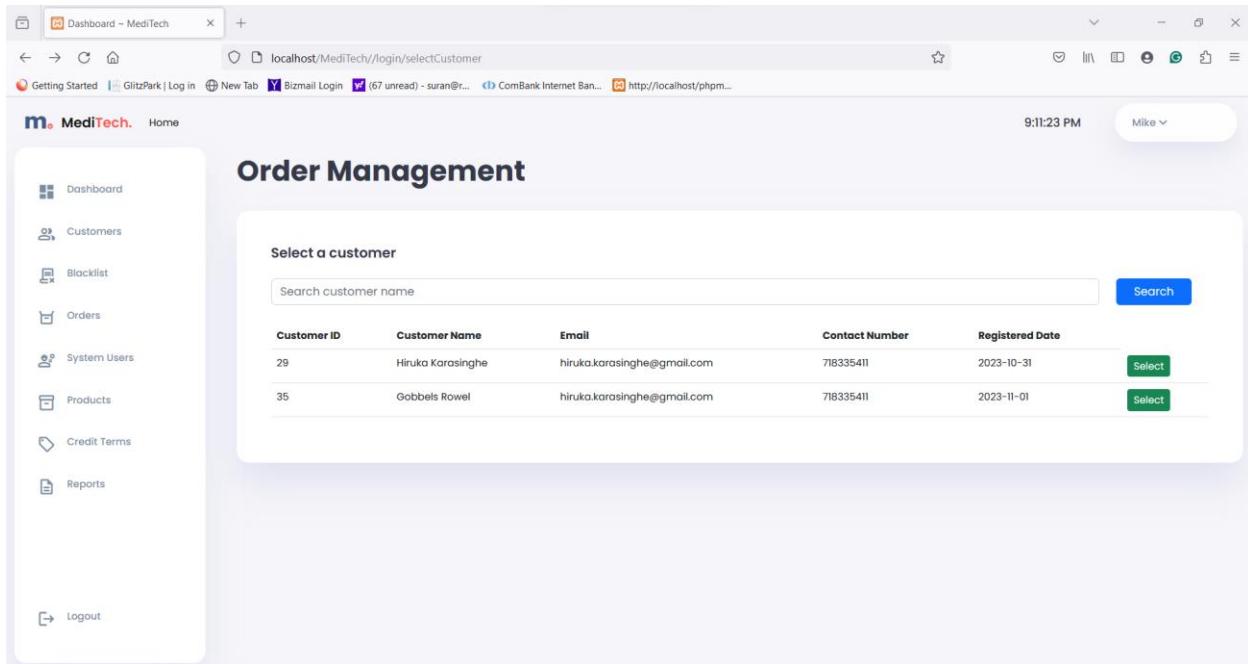


Figure 57: Pre-Condition – Order Management

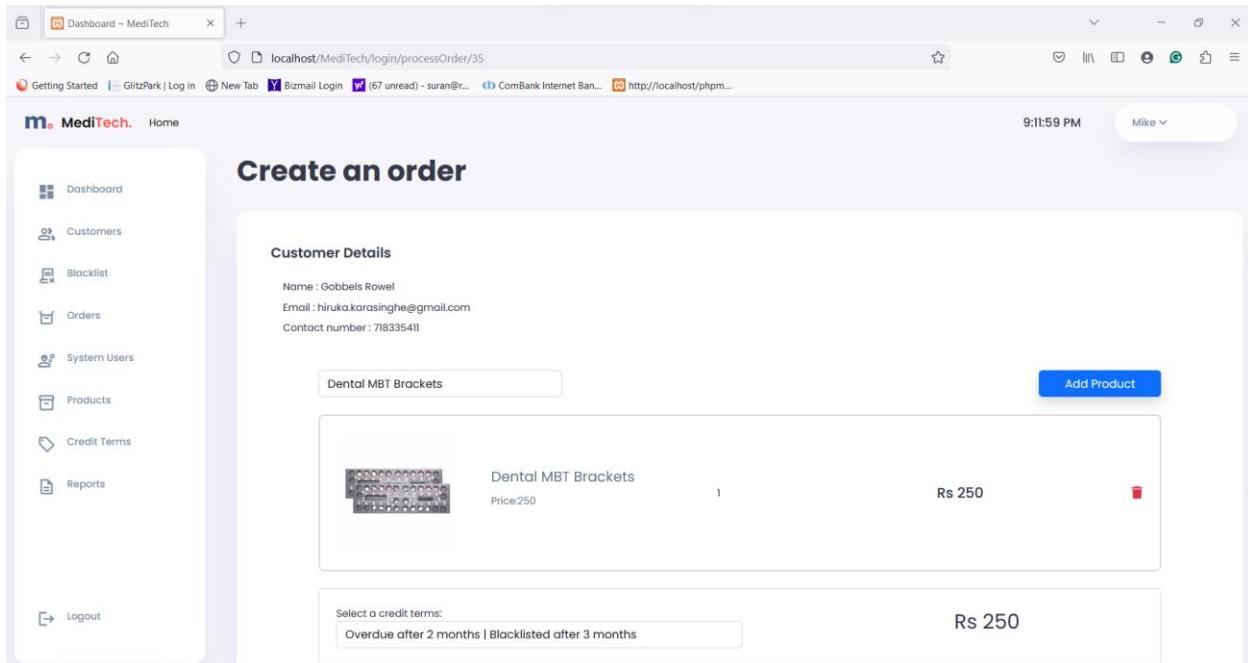


Figure 58: Pre-Condition – Order Management – Add order

The screenshot shows the MediTech Order Management interface. On the left is a sidebar with links: Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms, Reports, and Logout. The main area has a title 'Order Management' and a green success message 'Order added successfully.' Below this are two buttons: 'Add Order' and 'New Customer'. A search bar labeled 'Search order by customer' is followed by a table of order data. The table columns are Order ID, Customer Name, Value, Status, and Created Date. The rows show the following data:

Order ID	Customer Name	Value	Status	Created Date
16	Hiruka Karasinghe	12500	Paid	2023-II-12
17	hiruka	13800	Unpaid Overdue	2023-06-23
18	Hiruka Karasinghe	1000	Paid	2023-II-12
19	Gobbels Rowel	31200	Paid	2022-II-08
20	Gobbels Rowel	9000	Paid	2023-01-18

Each row has 'Details' and 'Update' buttons.

Figure 59: Post-Condition – Order Management – Add Order – Success Message

This screenshot is similar to Figure 59 but shows a different state. The sidebar and top navigation are identical. The main area shows the same 'Order Management' title and buttons. The search bar and table are also the same. However, the last row of the table is highlighted with a yellow background, indicating the most recent addition. The data for this row is:

Order ID	Customer Name	Value	Status	Created Date
16	Hiruka Karasinghe	12500	Paid	2023-II-12
17	hiruka	13800	Unpaid Overdue	2023-06-23
18	Hiruka Karasinghe	1000	Paid	2023-II-12
19	Gobbels Rowel	31200	Paid	2022-II-08
20	Gobbels Rowel	9000	Paid	2023-01-18
21	hiruka	12000	Unpaid	2023-II-18
22	Hiruka Karasinghe	3500	Unpaid	2024-02-18
23	Gobbels Rowel	250	Unpaid	2024-02-19

The last row (Order ID 23) has 'Details' and 'Update' buttons.

Figure 60: Post-Condition – Order Management – Add Order

6 Critical Appraisal

6.1 Critical appraisal

A critical appraisal of any given system involves the identification of any drawbacks or improvements that can be done. All the functions of the system are analyzed to collect these data and this process developer with very valuable information.

The major drawbacks that a designer can identify in the credit based order management system are as follows;

There is no payment gateway in the system – there is no platform to make direct payments through the system and it has been noted that proving a payment through the existing mechanism is a tedious task.

The new system has to be tested under stress, therefore, there is room for bottlenecks to occur.

The reporting functionality can be improved to ensure that data under certain conditions can be filtered out to make more unique reports.

6.2 Future Enhancements

The following enhancements may be done in the future to further develop on the existing system.

Firstly, a payment portal should be implemented.

Secondly, automated reminder generators can be used to send automatic reminders to customer.

A module to assign packing staff to an order will be useful to maintain responsibility within the business.

A portal for customers to directly place orders with no intermediary sales staff will be implemented in the future.

6.3 Experience gained

A bank of knowledge and experience working with various software tools and programming languages.

An in-depth understanding of the model view controller framework.

Skills related to collaboration, teamwork and task division.

Experience on how to coordinate and cooperate with cooperate level staff was gained.

6.4 Work Division

The following project was completed as a group project in collaboration with Hiruka Karasinghe of candidate number:

The following table described the work division amongst the two members

Task	Undertaken by member
Requirements Gathering	Pinindi Weeraratne Hiruka Karasinghe
Development of UML Diagrams	Pinindi Weeraratne Hiruka Karasinghe
Feasibility Study	Pinindi Weeraratne
User Interface Design	Pinindi Weeraratne
Customer Module development	Hiruka Karasinghe
User Module Development	Hiruka Karasinghe Pinindi Weeraratne
Order Management Module Development	Hiruka Karasinghe Pinindi Weeraratne
Products Module Development	Hiruka Karasinghe
Orders Module Development	Hiruka Karasinghe
Blacklist module Development	Pinindi Weeraratne Hiruka Karasinghe
Credit Terms module Development	Pinindi Weeraratne Hiruka Karasinghe
Reports Module Development	Pinindi Weeraratne
Test Case development	Pinindi Weeraratne

6.5 Conclusion

The development of this system for MediTech has been a great opportunity to explore and learn various tools and languages.

The team was able to deliver a well-rounded system that provided a good solution to the problem MediTech was facing. They now possess a centralized platform where all of their data and information are managed in a centric form, which in turn will help reduce their operating costs and make them a more reliable business in the long run.

The fully functional system met all its customer's requirements and was delivered within the agreed timeline. However, further improvements to the system will be made in the future and the scalability of the system will aid the team in further enhancements.

7 Appendix

7.1 Model

```
public function searchBlacklistByName($name)
{
    $condition = "name LIKE '%{$name}%'";
    $condition2 = "`status` = 'blacklisted'";
    $query = $this->db->select('*')
        ->where($condition, NULL, FALSE)
        ->where($condition2, NULL, FALSE)
        ->get('customers');

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

public function searchCustomerByName($name)
{
    $condition = "name LIKE '%{$name}%'";
    $condition2 = "`status` = 'whitelisted'";

    $query = $this->db->select('*')
        ->where($condition, NULL, FALSE)
        ->where($condition2, NULL, FALSE)
        ->get('customers');

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

public function editCustomerData($id, $customer)
{
    $condition = "custID='{$id}'";
    // $this->db->set('userType', $data['userType']);
    $this->db->set('name', $customer['name']);
    $this->db->set('email', $customer['email']);
    $this->db->set('contactNo', $customer['contactNo']);

    $this->db->where($condition);
    $this->db->update('customers');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}
```

```

public function addCustomer($data)
{
    $this->db->set('email', $data['email']);
    $this->db->set('name', $data['name']);
    $this->db->set('contactNo', $data['contactNo']);

    $query = $this->db->insert('customers');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function delCustomer($id)
{
    //DELETE FROM customers WHERE `customers`.`custID` = 19
    $condition = "custID='{$id}'";
    $this->db->where($condition);
    $this->db->delete('customers');
    if ($this->db->affected_rows() == 1) {
        return true;
    } else {
        return false;
    }
}

// -----User Profile Model-----

public function getUserTypes()
{
    $sql = "SELECT SUBSTRING(COLUMN_TYPE, 6, LENGTH(COLUMN_TYPE) - 6) AS enum_values "
        . "FROM information_schema.COLUMNS "
        . "WHERE TABLE_NAME = 'users' AND COLUMN_NAME = 'userType';";

    $query = $this->db->query($sql);

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function getUserData($data)
{
    $this->db->where('email', $data['email']);
    $this->db->where('password', $data['password']);
    $query = $this->db->get('users');

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

public function getUserDataByID($id)
{
    $condition = "userid='{$id}'";
    $query = $this->db->select('*')
        ->where($condition)
        ->get('users');

    // return $query->result();
    if ($query->num_rows() == 1) {
        return $query->result();
    } else {
        return false;
    }
}

public function updateProfile($data)
{
    $condition = "userid='{$data['id']}'";
    // $this->db->set('userType', $data['userType']);
    $this->db->set('email', $data['email']);
    $this->db->set('name', $data['name']);
    $this->db->set('contactNo', $data['contact']);

    $this->db->where($condition);
    $this->db->update('users');

    // echo $this->db->last_query();
    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function fetchUserDB()
{
    $query = $this->db->query("SELECT * FROM `users`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function addUser($data)
{
    $this->db->set('email', $data['email']);
    $this->db->set('name', $data['name']);
    $this->db->set('contactNo', $data['contactNo']);
    $this->db->set('branchID', $data['branchID']);
    $this->db->set('password', $data['password']);
    $this->db->set('userType', $data['userType']);

    $query = $this->db->insert('users');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

```

```

public function editUserData($id, $data)
{
    $condition = "userid='{$id}'";
    // $this->db->set('userType', $data['userType']);
    $this->db->set('email', $data['email']);
    $this->db->set('name', $data['name']);
    $this->db->set('contactNo', $data['contactNo']);
    $this->db->set('branchID', $data['branchID']);
    $this->db->set('password', $data['password']);
    $this->db->set('userType', $data['userType']);

    $this->db->where($condition);
    $this->db->update('users');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function delUser($id)
{
    // DELETE FROM customers WHERE `customers`.`custID` = 19
    $condition = "userid='{$id}'";
    $this->db->where($condition);
    $this->db->delete('users');
    if ($this->db->affected_rows() == 1) {
        return true;
    } else {
        return false;
    }
}

// -----Login Model-----

public function loginCheck($data)
{
    $this->db->where('email', $data['email']);
    $this->db->where('password', $data['password']);
    $query = $this->db->get('users');

    if ($query->num_rows() > 0) {
        return true;
    } else {
        return false;
    }
}

```

```

// -----Credit Model-----

public function deactivateCredit($id)
{
    $condition = "id='{$id}'";
    $this->db->set('inactive', '1');

    $this->db->where($condition);
    $this->db->update('creditterms');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function addCredit($data)
{
    $this->db->set('overdue_period', $data['overdue']);
    $this->db->set('blacklisted_period', $data['blacklist']);

    $query = $this->db->insert('creditterms');

    if ($this->db->affected_rows() > 0) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

// -----Order Model-----

public function fetchCreditTermsDB()
{
    $query = $this->db->query("SELECT * FROM `creditterms`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

public function fetchodersDB()
{
    $query = $this->db->query("SELECT * FROM `orders`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchOrdersWCustomers()
{
    $query = $this->db->query("SELECT orders.*,customers.name FROM orders orders, customers
customers where customers.custID = orders.custID");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchOrdersWCustomerID($id)
{
    $query = $this->db->query("SELECT * FROM orders JOIN customers ON customers.custID =
orders.custID WHERE orders.id = $id");
    //print_r($query->result());
    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchOrdersWCreditTerms($id)
{
    $query = $this->db->query("SELECT * FROM orders JOIN credittterms ON credittterms.id =
orders.creditTermID WHERE orders.id = $id");
    //print_r($query->result());
    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function getOrderProducts($id)
{
    $sql = "SELECT products.*, order_products.*
    FROM order_products
    INNER JOIN products ON order_products.product_id = products.productID
    WHERE order_products.order_id = $id";

    $query = $this->db->query($sql);
    //print_r($query->result());

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

public function addOrder($data)
{
    $this->db->set('custID', $data['custID']);
    $this->db->set('value', $data['value']);
    $this->db->set('creditTermID', $data['creditTermsID']);

    $query = $this->db->insert('orders');

    $orderID = $this->db->insert_id();

    $orderProductData = array();
    foreach ($data['productID'] as $key => $productID) {
        $orderProductData[] = array(
            'order_id' => $orderID,
            'product_id' => $productID,
            'quantity' => $data['quantity'][$productID]
        );
    }

    $this->db->insert_batch('order_products', $orderProductData);

    if ($this->db->affected_rows() > 0) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function updatePayment($id)
{
    $condition = "id='{$id}'";

    $this->db->set('isPaid', 1);
    $this->db->set('isOverdue', 0);
    $this->db->where($condition);
    $this->db->update('orders');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

public function addConfirmationRecord($data)
{
    $this->db->set('orderID', $data['orderID']);
    $this->db->set('img', $data['img']);
    $this->db->set('receiptNo', $data['receiptNo']);

    $query = $this->db->insert('payment_records');

    if ($this->db->affected_rows() == 1) {
        return (1);
    } else if ($this->db->affected_rows() == 0) {
        return (0);
    } else {
        return (-1);
    }
}

```

```

public function searchOrderByName($name)
{
    $condition = "name LIKE '%{$name}%'";

    $query = $this->db->query("SELECT orders.*,customers.name FROM orders orders, customers
customers where customers.custID = orders.custID AND $condition");

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchPaymentRecord($id)
{
    $condition = "orderID='{$id}'";

    $query = $this->db->select('*')
        ->where($condition)
        ->get('payment_records');

    return $query->result();
    if ($query->num_rows() == 1) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

<?php

class Data_model extends CI_Model
{

    public function getMonthlySales($month)
    {
        if ($month <= 12) {
            $query = $this->db->query("SELECT * FROM `orders` WHERE MONTH(`created_date`) = $month");
            if ($query) {
                return $query->result();
            } else {
                echo $this->db->error();
                return false;
            }
        } else {
            $query = $this->db->query("SELECT * FROM `orders` WHERE YEAR(`created_date`) = $month");
            if ($query) {
                return $query->result();
            } else {
                echo $this->db->error();
                return false;
            }
        }
    }
}

```

```

public function getMonthlyCust($month)
{
    if ($month <= 12) {
        $query = $this->db->query("SELECT * FROM `customers` WHERE MONTH(`registered_date`) = $month");
        if ($query) {
            return $query->result();
        } else {
            echo $this->db->error();
            return false;
        }
    } else {
        $query = $this->db->query("SELECT * FROM `customers` WHERE YEAR(`registered_date`) = $month");
        if ($query) {
            return $query->result();
        } else {
            echo $this->db->error();
            return false;
        }
    }
}

public function getCreditTermsNo()
{
    $query = $this->db->query("SELECT COUNT(*) AS total_entries FROM creditterms");
    if ($query) {
        $result = $query->row();
        return (int)$result->total_entries;
    } else {
        return false;
    }
}

public function fetchCreditTerms()
{
    $query = $this->db->query("SELECT * FROM `creditterms`");

    if ($query) {
        return $query->result();
    } else {
        return false;
    }
}

public function fetchOverdueOrders($m)
{
    if ($m <= 12) {
        $condition = "isOverdue='1'";
        $condition2 = "MONTH(`created_date`) = $m";
        $query = $this->db->select('*')
            ->where($condition)
            ->where($condition2)
            ->get('orders');

        // return $query->result();
        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return 0;
        }
    } else {
        $condition = "isOverdue='1'";
        $condition2 = "YEAR(`created_date`) = $m";
        $query = $this->db->select('*')
            ->where($condition)
            ->where($condition2)
            ->get('orders');

        // return $query->result();
        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return 0;
        }
    }
}

```

```

public function findOverdueMonth($id)
{
    $query = $this->db->select('overdue_period')->where('id', $id)->get('creditterms');
    if ($query->num_rows() > 0) {
        $result = $query->row();
        $overduePeriod = $result->overdue_period;
        return $overduePeriod;
    } else {
        return false;
    }
}

public function findBlacklistMonth($id)
{
    $query = $this->db->select('blacklisted_period')->where('id', $id)->get('creditterms');
    if ($query->num_rows() > 0) {
        $result = $query->row();
        $blacklistedPeriod = $result->blacklisted_period;
        return $blacklistedPeriod;
    } else {
        return false;
    }
}

public function updateOverdueOrderPerTerm($term, $m)
{
    $condition = "creditTermID='{$term}' AND isPaid = 0 AND isOverdue = 0 AND created_date <
DATE_SUB(NOW(), INTERVAL {$m} MONTH)";

    $query = $this->db->select('*')
        ->where($condition)->get('orders');
    // Update the orders that meet the conditions
    $this->db->set('isOverdue', 1)
        ->where($condition)
        ->update('orders');

    // Check if any rows were affected
    return $query->result();
}

public function findBlacklistableUsers($term, $m)
{
    $condition = "creditTermID='{$term}' AND isPaid = 0 AND isOverdue = 1 AND created_date <
DATE_SUB(NOW(), INTERVAL {$m} MONTH)";

    // Update the orders that meet the conditions
    $query = $this->db->select('*')
        ->where($condition)
        ->get('orders');

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

```

```

public function blacklistCustomers($id)
{
    $condition = "custID='{$id}'";

    // Update the orders that meet the conditions
    $this->db->set('status', 'blacklisted')
        ->where($condition)
        ->update('customers');

    // Check if any rows were affected
    return $this->db->affected_rows();
}

public function getOrderCustomerID($id)
{
    $condition = "id='{$id}'";

    // Update the orders that meet the conditions
    $query = $this->db->select('custID')
        ->where($condition)
        ->get('orders');

    if ($query->num_rows() > 0) {
        $result = $query->row();
        $cust = $result->custID;
        return $cust;
    } else {
        return false;
    }
}

public function orderByCustomers($custID, $id)
{
    $condition1 = "custID='{$custID}'";
    $condition2 = "creditTermID='{$id}'";

    // Update the orders that meet the conditions
    $query = $this->db->select('*')
        ->where($condition1)
        ->where($condition2)
        ->get('orders');

    if ($query->num_rows() > 0) {
        return $query->result();
    } else {
        return false;
    }
}

public function isBlacklisted($id, $m)
{
    $condition = "id='{$id}' AND created_date < DATE_SUB(NOW(), INTERVAL {$m} MONTH)";

    // Update the orders that meet the conditions
    $query = $this->db->select('*')
        ->where($condition)
        ->get('orders');

    if ($query->num_rows() > 0) {
        return true;
    } else {
        return false;
    }
}

```

```
public function whitelistCustomer($id)
{
    $condition = "custID='{$id}'";

    // Update the orders that meet the conditions
    $this->db->set('status', 'whitelisted')
        ->where($condition)
        ->update('customers');

    // Check if any rows were affected
    return $this->db->affected_rows();
}
```

7.2 Controller

```
public function loginSubmit()
{
    //login submission proressing controller function
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('password', 'Password', 'required');
    if ($this->form_validation->run() == FALSE) {
        $this->load->view('auth/login');
    } else {
        $data = array(
            'email' => $_POST['email'],
            'password' => $_POST['password'],
        );
        $result = $this->user_model->loginCheck($data);
        if ($result) {

            $resutlUserData = $this->user_model->getUserData($data);

            $session_user = array(
                'id' => $resutlUserData[0]->userid,
                'login' => true,
                'name' => $resutlUserData[0]->name
            );

            if ($resutlUserData[0]->userType == 'Admin') {
                $actions = array(
                    'dashboard' => [
                        'myprofile' => true,
                        'teamprofile' => true,
                        'product' => true,
                        'reports' => true
                    ],
                    'customers' => [
                        'edit' => true,
                        'delete' => true
                    ],
                    'user-management' => [
                        'view' => true,
                        'edit' => true,
                        'delete' => true
                    ],
                    'products' => [
                        'view' => true,
                        'edit' => true,
                        'delete' => true
                    ],
                    'blacklist' => [
                        'view' => true,
                    ],
                    'orders' => [
                        'view' => true,
                        'edit' => true,
                        'delete' => true
                    ],
                    'credit' => [
                        'view' => true,
                        'edit' => true,
                        'delete' => true
                    ],
                    'reports' => [
                        'view' => true,
                        'edit' => true,
                        'delete' => true
                    ],
                );
            }
        }
    }
}
```

```

} elseif ($resutlUserData[0]->userType == 'Sales') {
    $actions = array(
        'dashboard' => [
            'myprofile' => true,
            'teamprofile' => true,
            'product' => true,
            'reports' => true
        ],
        'customers' => [
            'edit' => true,
            'delete' => false
        ],
        /*
        'user-management' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        */
        'blacklist' => [
            'view' => true,
        ],
        'orders' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        /*
        'reports' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],*/
    );
} elseif ($resutlUserData[0]->userType == 'DataEntry') {
    $actions = array(
        'dashboard' => [
            'myprofile' => true,
            'teamprofile' => true,
            'product' => true,
            'reports' => true
        ],
        'customers' => [
            'edit' => false,
            'delete' => false
        ],
        'user-management' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        /*
        'blacklist' => [
            'view' => true,
        ],
        */
        'orders' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        'credit' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        'products' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],
        /*
        'reports' => [
            'view' => true,
            'edit' => true,
            'delete' => true
        ],*/
    );
}
);
}

```

```

// update session object with new session data
    $this->session->set_userdata('userinfo', $session_user);
    $this->session->set_userdata('routing', $actions);
    $this->session->set_flashdata('success', "Successfully Logged in.");

    $termsNo = $this->data_model->getCreditTermsNo();
    $terms = $this->data_model->fetchCreditTerms();

    //----- Update the overdue status of orders/sending overdue reminders -----
    foreach ($terms as $term) {
        $month = $this->data_model->findOverdueMonth($term->id);
        $overdueOrder = $this->data_model->updateOverdueOrderPerTerm($term->id, $month);

        foreach ($overdueOrder as $ovOrder) {
            $cust = $this->user_model->getCustomerDataByID($ovOrder->custID);
            $overdueAmount = $ovOrder->value;
            $this->sendReminder($cust[0]->name, $overdueAmount, $ovOrder->created_date,
$cust[0]->email);
        }
    }

    //----- Blacklist blacklistable users
    foreach ($terms as $term) {
        $month = $this->data_model->findBlacklistMonth($term->id);
        $bresult = $this->data_model->findBlacklistableUsers($term->id, $month);
        if ($bresult != false) {
            foreach ($bresult as $customer) {
                $result = $this->data_model->blacklistCustomers($customer->custID);
            }
        }
    }

    $currentMonth = date('n');

    redirect('login/dashboard/' . $currentMonth);
} else {
    $this->session->set_flashdata('error', 'Invalid email/password. Please try again.');
    redirect('login/userlogin');
}
}
}
}

```

```

public function refreshdash()
{
    //update selected month on refresh
    $currentMonth = $_GET['value'];
    redirect('login/dashboard/' . $currentMonth);
}

public function dashboard($currentMonth)
{
    //dashboard page controller
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if (isset($data['error']) || isset($data['success'])) {

        $data['orders'] = $this->data_model->getMonthlySales($currentMonth);
        $data['customers'] = $result2 = $this->data_model->getMonthlyCust($currentMonth);
        $data['overdue'] = $this->data_model->fetchOverdueOrders($currentMonth);
        $data['month'] = $currentMonth;

        $this->load->view('dashboard/dashboard', $data);
    } else {
        if ($this->checkSessionExist()) {

            $data['orders'] = $this->data_model->getMonthlySales($currentMonth);
            $data['customers'] = $this->data_model->getMonthlyCust($currentMonth);
            $data['overdue'] = $this->data_model->fetchOverdueOrders($currentMonth);
            $data['month'] = $currentMonth;

            $this->load->view('dashboard/dashboard', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

public function userMng()
{
    //user management page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchUserDB();

        if ($result) {
            $data['user'] = $result;
            $this->load->view('user-management/user-management', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function adduser()
{
    //add user section with the user management page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if (isset($data['error']) || isset($data['success'])) {
        $result = $this->user_model->fetchUserDB();

        if ($result) {
            $data['user'] = $result;

            $this->load->view('user-management/user-management', $data);
        }
    } else {
        if ($this->checkSessionExist()) {

            $result = $this->user_model->fetchUserDB();
            $userType = $this->user_model->getUserTypes();

            if ($result) {
                $data['user'] = $result;
                $data['dropdown'] = $userType;

                $this->load->view('user-management/add-user-management', $data);
            }
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

public function addUserSubmit()
{
    //user addition processing function
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('name', 'Name', 'required');
    $this->form_validation->set_rules('contact', 'Contact', 'required');
    $this->form_validation->set_rules('branch', 'Branch', 'required');
    $this->form_validation->set_rules('password', 'Password', 'required');
    $this->form_validation->set_rules('type', 'Type', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect("/login/adduser/");
    } else {

        $data = array(
            'name' => $_POST['name'],
            'email' => $_POST['email'],
            'contactNo' => $_POST['contact'],
            'branchID' => $_POST['branch'],
            'password' => $_POST['password'],
            'userType' => $_POST['type']
        );

        $result = $this->user_model->addUser($data);

        if ($result == 1) {
            $data['success'] = $this->session->set_flashdata('success', 'User added successfully.');
            redirect('login/userMng');
        } elseif ($result == 0) {
            $data['success'] = $this->session->set_flashdata('success', 'User added successfully.');
            redirect('login/userMng');
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'Error occured please try again');
            redirect("login/addUser");
        }
    }
}

```

```

public function editUser($id)
{
    //user management with user edit functions
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchUserDB();
        $customerFP = $this->user_model->getUserDataByID($id);
        $userType = $this->user_model->getUserTypes();

        if ($result && $customerFP && $userType) {

            $data['dropdown'] = $userType;
            $data['user'] = $result;
            $data['table'] = $customerFP;

            $this->load->view('user-management/edit-user-management', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

public function editUserSubmit($id)
{
    //user edit submission processing function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchUserDB();
        $customerFP = $this->user_model->getUserDataByID($id);

        if ($result && $customerFP) {

            $data = array(
                'name' => $_POST['name'],
                'email' => $_POST['email'],
                'contactNo' => $_POST['contact'],
                'branchID' => $_POST['branch'],
                'password' => $_POST['password'],
                'userType' => $_POST['type'],
                'userid' => $id
            );

            $result = $this->user_model->editUserData($id, $data);

            if ($result == 1) {
                $data['success'] = $this->session->set_flashdata('success', 'User edited successfully');

                redirect('login/userMng');
            } elseif ($result == 0) {
                $data['success'] = $this->session->set_flashdata('error', 'No edits were made.');
                redirect('login/userMng');
            } else {
                $this->session->set_flashdata('error', 'Error occurred please try again');
                redirect('/login/userMng');
            }
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function delUser($id)
{
    //delete user function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {
        $result = $this->user_model->delUser($id);
        if ($result) {
            $data['success'] = $this->session->set_flashdata('success', 'User deleted successfully.');
            redirect('login/userMng');
        } else {
            $data['success'] = $this->session->set_flashdata('error', 'Error, the record was not deleted.');
            redirect("/login/userMng");
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function customerMng()
{
    //customer management controller function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchCustomerDB();

        if ($result) {
            $data['customer'] = $result;
            $this->load->view('customer-management/customer-management', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function addcustomer()
{
    // add customer section with customer mng page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if (!isset($data['error']) || !isset($data['success'])) {
        $result = $this->user_model->fetchCustomerDB();

        if ($result) {
            $data['customer'] = $result;
            $this->load->view('customer-management/customer-management', $data);
        }
    } else {
        if ($this->checkSessionExist()) {

            $result = $this->user_model->fetchCustomerDB();

            if ($result) {
                $data['customer'] = $result;
                $this->load->view('customer-management/add-customer-management', $data);
            }
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

```

```

public function addCustomerSubmit()
{
    //submission of customer addition processing function
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('name', 'Name', 'required');
    $this->form_validation->set_rules('contact', 'Contact', 'required');
    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect('/login/addcustomer/');
    } else {
        $data = array(
            'name' => $_POST['name'],
            'email' => $_POST['email'],
            'contactNo' => $_POST['contact']
        );
        $result = $this->user_model->addCustomer($data);

        if ($result == 1) {
            $data['success'] = $this->session->set_flashdata('success', 'Customer added successfully.');
            redirect('login/addCustomer');
        } elseif ($result == 0) {
            $data['success'] = $this->session->set_flashdata('success', 'Customer added successfully.');
            redirect('login/addCustomer');
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'Error occurred please try again');
            redirect("login/addCustomer");
        }
    }
}

public function editCustomer($id)
{
    //customer edit section with customer mng page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchCustomerDB();
        $customerFP = $this->user_model->getCustomerDataByID($id);

        if ($result && $customerFP) {
            $data['customer'] = $result;
            $data['table'] = $customerFP;

            $this->load->view('customer-management/edit-customer-management', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

public function editCustomerSubmit($id)
{
    //submit customer edit function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchCustomerDB();
        $customerFP = $this->user_model->getCustomerDataByID($id);

        if ($result && $customerFP) {

            $data = array(
                'custID' => $id,
                'name' => $_POST['name'],
                'email' => $_POST['email'],
                'contactNo' => $_POST['contact']
            );

            $result = $this->user_model->editCustomerData($id, $data);

            if ($result == 1) {
                $data['success'] = $this->session->set_flashdata('success', 'Customer edited successfully');
                redirect('login/customerMng');
            } elseif ($result == 0) {
                $data['success'] = $this->session->set_flashdata('error', 'No edits were made.');
                redirect('login/customerMng');
            } else {
                $this->session->set_flashdata('error', 'Error occurred please try again');
                redirect("/login/customerMng");
            }
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

```

```

public function delCustomer($id)
{
    //delete customer function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {
        $result = $this->user_model->delCustomer($id);
        if ($result) {
            $data['success'] = $this->session->set_flashdata('success', 'Record deleted successfully.');
            redirect('login/customerMng');
        } else {
            $data['success'] = $this->session->set_flashdata('error', 'Error, the record was not deleted.');
            redirect("/login/customerMng");
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function products()
{
    //product management page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchProducts();
        if ($result == null) {
            $data['product'] = null;
            $this->load->view('products/add-product', $data);
        }
        if ($result) {
            $data['product'] = $result;
            $this->load->view('products/products', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function addProduct()
{
    //add product section with product management page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if (isset($data['error']) || isset($data['success'])) {
        $result = $this->user_model->fetchProducts();

        if ($result) {
            $data['product'] = $result;
            $this->load->view('products/products', $data);
        }
    } else {
        if ($this->checkSessionExist()) {

            $result = $this->user_model->fetchProducts();

            if ($result) {
                $data['product'] = $result;
                $this->load->view('products/add-product', $data);
            }
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

```

```

public function addProductSubmit()
{
    //product addition submit controller function
    $this->form_validation->set_rules('title', 'Title', 'required');
    $this->form_validation->set_rules('description', 'Description', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect("login/addProduct/");
    } else {
        $file_extension = pathinfo($_FILES["userfile"]["name"], PATHINFO_EXTENSION);

        // Generate a unique file name
        $new_name = time() . '_' . mt_rand(100000, 999999) . '.' . $file_extension;

        $config = array(
            'upload_path' => 'uploads/images/',
            'allowed_types' => "gif|jpg|png|jpeg|pdf",
            'overwrite' => TRUE,
            'max_size' => "2048000",
            // 'max_height' => "768",
            // 'max_width' => "1024",
            'file_name' => $new_name
        );
        $this->load->library('upload', $config);
        $this->upload->initialize($config);
        if ($this->upload->do_upload()) {
            $data = array('upload_data' => $this->upload->data());

            $data = array(
                'title' => $_POST['title'],
                'description' => $_POST['description'],
                'img' => $new_name,
                'price' => $_POST['price']
            );
            $result = $this->user_model->addProduct($data);

            if ($result == 1) {
                $data['success'] = $this->session->set_flashdata('success', 'Product added successfully.');
                redirect('login/products');
            } elseif ($result == 0) {
                $data['error'] = $this->session->set_flashdata('success', 'No products were added.');
                redirect('login/products');
            } else {
                $data['error'] = $this->session->set_flashdata('error', 'Error occurred please try again');
                redirect('login/products');
            }
        } else {
            $error = $this->upload->display_errors();
            $data['error'] = $this->session->set_flashdata('error', $error);
            redirect('login/products');
        }
    }
}

public function editProduct($id)
{
    //edit product section with product management section
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];

    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchProducts();
        $productsFP = $this->user_model->getProductByID($id);

        if ($result && $productsFP) {

            $data['product'] = $result;
            $data['info'] = $productsFP;

            $this->load->view('products/edit-products', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function editProductSubmit($id)
{
    //edit product submit processing function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchProducts();
        $productFP = $this->user_model->getProductByID($id);

        $file_extension = pathinfo($_FILES["userfile"]["name"], PATHINFO_EXTENSION);

        $new_name = time() . '_' . mt_rand(100000, 999999) . '.' . $file_extension;

        $config = array(
            'upload_path' => 'uploads/images/',
            'allowed_types' => "gif|jpg|png|jpeg|pdf",
            'overwrite' => TRUE,
            'max_size' => "2048000",
            // 'max_height' => "768",
            // 'max_width' => "1024",
            'file_name' => $new_name
        );
        $this->load->library('upload', $config);
        $this->upload->initialize($config);
        if ($this->upload->do_upload()) {
            $data = array('upload_data' => $this->upload->data());

            $data = array(
                'title' => $_POST['title'],
                'description' => $_POST['description'],
                'img' => $new_name,
                'price' => $_POST['price'],
            );
        }

        $result = $this->user_model->editProductData($id, $data);

        if ($result == 1) {
            $data['success'] = $this->session->set_flashdata('success', 'Product edited successfully.');
            redirect('login/products');
        } elseif ($result == 0) {
            $data['error'] = $this->session->set_flashdata('error', 'No edits were made.');
            redirect('login/products');
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'Error occured please try again');
            redirect('login/products');
        }
    } else {
        $error = array('error' => $this->upload->display_errors());
        print_r($error);
    }
    $this->load->view('auth/login', $data);
}
}

```

```

public function delProduct($id)
{
    //delete product function
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }

    if ($this->checkSessionExist()) {
        $result = $this->user_model->delProduct($id);
        if ($result) {
            $data['success'] = $this->session->set_flashdata('success', 'Product deleted
successfully.');
            redirect('login/products');
        } else {
            $data['success'] = $this->session->set_flashdata('error', 'Error, the record was not
deleted.');
            redirect("/login/products");
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function blacklist()
{
    //blacklist view page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->fetchBlacklistedCustomers();

        if ($result) {
            $data['customer'] = $result;

            $this->load->view('blacklist/blacklist', $data);
        } else {
            $data['error'] = "No records were found";
            $this->load->view('blacklist/empty-blacklist', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function searchBlacklistSubmit()
{
    //view searched blacklisted users in the blacklist module
    $searchTerm = $_POST['value'];
    $result = $this->user_model->searchBlacklistByName($searchTerm);

    if ($result) {
        $data['customer'] = $result;

        $this->load->view('blacklist/blacklist', $data);
    } else {
        $data['error'] = $this->session->set_flashdata('error', 'No results found');
        redirect('login/blacklist');
    }
}

```

```

public function orderMng()
{
    //opens order management page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchOrdersWCustomers();

        if ($result) {

            $data['orders'] = $result;

            $this->load->view('order-management/order-management', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

public function searchOrderSubmit()
{
    //searches orders by customer name submit processing function
    $searchTerm = $_POST['value'];
    $result = $this->user_model->searchOrderByName($searchTerm);

    if ($result) {
        $data['orders'] = $result;

        $this->load->view('order-management/order-management', $data);
    } else {
        $data['error'] = $this->session->set_flashdata('error', 'No results found');
        redirect('login/orderMng');
    }
}

public function selectCustomer()
{
    //selects a customer by their name from the order creation page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->fetchWhitelistedCustomerDB();
        $result2 = $this->user_model->fetchOrdersWCustomers();

        if ($result) {
            $data['customer'] = $result;
            $data['orders'] = $result2;

            $this->load->view('order-management/select-customer', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function searchCustomerSubmit()
{
    //searches customers that are eligible to have a new order made
    $searchTerm = $_POST['value'];
    $result = $this->user_model->searchCustomerByName($searchTerm);
    $result2 = $this->user_model->fetchOrdersWCustomers();

    if ($result) {
        $data['customer'] = $result;
        $data['orders'] = $result2;

        $this->load->view('order-management/select-customer', $data);
    } else {
        $data['error'] = $this->session->set_flashdata('error', 'No results found');
        redirect('login/selectCustomer');
    }
}

public function processOrder($id)
{
    //open order creation page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->getCustomerDataByID($id);
        $crediterms = $this->user_model->fetchCreditTermsDB();
        $products = $this->user_model->fetchProducts();
        if ($result) {
            $data['customer'] = $result;
            $data['terms'] = $crediterms;
            $data['products'] = $products;
            $data['orderId'] = $id;

            $this->load->view('order-management/create-order', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function processOrderSubmit($id)
{
    //add new order to the database table
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->getCustomerDataByID($id);
        $crediterms = $this->user_model->fetchCreditTermsDB();
        $products = $this->user_model->fetchProducts();
        print_r($products);
        if ($result) {
            $data['customer'] = $result;
            $data['terms'] = $crediterms;
            $data['products'] = $products;
            $data['orderId'] = $id;
            $this->load->view('order-management/create-order', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function addProductToOrder($id)
{
    //add a given product to the processOrder page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        if ($id) {
            $result = $this->user_model->getCustomerDataByID($id);
            $creditterms = $this->user_model->fetchCreditTermsDB();
            $products = $this->user_model->fetchProducts();
            print_r($products);
            if ($result) {
                $data['customer'] = $result;
                $data['terms'] = $creditterms;
                $data['products'] = $products;
                $data['orderID'] = $id;

                $this->load->view('order-management/create-order', $data);
            }
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'No results found');
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function ajaxRequestHandler()
{
    //update product
    $out = $_GET['value'];
    $products = $this->user_model->getProductByID($out);
    $data['products'] = $products;

    // Set the response type to JSON
    header('Content-Type: application/json');
    echo json_encode($data);
    exit;
}

```

```

public function addOrderSubmit($id)
{
    //the order addition submission process
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $this->form_validation->set_rules('total_order_value', 'Total_order_value', 'required');
        $this->form_validation->set_rules('type', 'Type', 'required');

        if ($this->form_validation->run() == FALSE) {
            $this->session->set_flashdata('error', 'Form details cannot be empty');
            redirect("/login/processOrder/" . $id);
        } else {

            $data = array(
                'custID' => $id,
                'creditTermsID' => $_POST['type'],
                'value' => $_POST['total_order_value'],
                'productID' => $_POST['product_ids'],
                'quantity' => $_POST['product_quantities']
            );

            if (!($data['productID'] && $data['quantity'])) {
                $this->session->set_flashdata('error', 'Form details cannot be empty');
                redirect("/login/processOrder/" . $id);
            }

            $result = $this->user_model->addOrder($data);

            if ($result == 1) {
                $data['success'] = $this->session->set_flashdata('success', 'Order added successfully.');
                redirect('login/orderMng');
            } elseif ($result == 0) {
                $data['success'] = $this->session->set_flashdata('success', 'User added successfully.');
                redirect('login/orderMng');
            } else {
                $data['error'] = $this->session->set_flashdata('error', 'Error occured please try again');
                redirect("login/selectCustomer");
            }
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function orderUpdate($id)
{
    //views the order update page along with order data
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->fetchOrdersWCustomerID($id);
        $result2 = $this->user_model->getOrderProducts($id);
        $result3 = $this->user_model->fetchOrdersWCreditTerms($id);
        if ($result) {
            $data['orders'] = $result;
            $data['products'] = $result2;
            $data['credit'] = $result3;

            $this->load->view('order-management/order-view', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function orderUpdateForm($id)
{
    //load the order confirmation section to the order page
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {
        $result = $this->user_model->fetchOrdersWCustomerID($id);
        $result2 = $this->user_model->getOrderProducts($id);
        $result3 = $this->user_model->fetchOrdersWCreditTerms($id);
        if ($result) {
            $data['orders'] = $result;
            $data['products'] = $result2;
            $data['credit'] = $result3;

            $this->load->view('order-management/order-view-confirm', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

```

```

public function orderUpdateSubmit()
{
    //submit the order confirmation details and confirm order
    $this->form_validation->set_rules('receipt', 'Receipt', 'required');
    $id = $_POST['id'];
    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect("login/orderUpdateForm/$id");
    } else {
        $file_extension = pathinfo($_FILES["userfile"]["name"], PATHINFO_EXTENSION);

        // Generate a unique file name
        $new_name = time() . '_' . mt_rand(100000, 999999) . '.' . $file_extension;

        $config = array(
            'upload_path' => 'uploads/recipts/',
            'allowed_types' => 'gif|jpg|png|jpeg|pdf',
            'overwrite' => TRUE,
            'max_size' => "2048000",
            // 'max_height' => "768",
            // 'max_width' => "1024",
            'file_name' => $new_name
        );
        $this->load->library('upload', $config);
        $this->upload->initialize($config);
        if ($this->upload->do_upload()) {
            $data = array('upload_data' => $this->upload->data());

            $data = array(
                'orderID' => $_POST['id'],
                'receiptNo' => $_POST['receipt'],
                'img' => $new_name
            );

            $result = $this->user_model->addConfirmationRecord($data);
            $result2 = $this->user_model->updatePayment($id);
            $result3 = $this->data_model->getOrderCustomerID($id); //customer_id

            $terms = $this->data_model->fetchCreditTerms();
            $isWhitelistable = 2; //null

            foreach ($terms as $term) {
                $custOrders = $this->data_model->orderByCustomers($result3, $term->id);
                $month = $this->data_model->findBlacklistMonth($term->id);
                foreach ($custOrders as $order) {
                    if ($order->isOverdue == 1) {
                        if ($this->data_model->isBlacklisted($order->id, $month)) {
                            $isWhitelistable = 0;
                        } else {
                            if ($isWhitelistable == 0) {
                                break;
                            } else {
                                $isWhitelistable = 1;
                            }
                        }
                    } else {
                        if ($isWhitelistable == 0) {
                            break;
                        } else {
                            $isWhitelistable = 1;
                        }
                    }
                }
            }

            if ($isWhitelistable == 1) {
                $result4 = $this->data_model->whitelistCustomer($result3);
            }

            if ($result && $result2 == 1) {
                $data['success'] = $this->session->set_flashdata('success', 'Payment Confirmation Success');
                redirect('login/orderUpdate/' . $id);
            } elseif ($result && $result2 == 0) {
                $data['error'] = $this->session->set_flashdata('error', 'Confirmation Not Added');
                redirect('login/orderUpdate/' . $id);
            } else {
                $data['error'] = $this->session->set_flashdata('error', 'Error occurred please try again');
                redirect('login/orderMng');
            }
        } else {
            $error = $this->upload->display_errors();
            $data['error'] = $this->session->set_flashdata('error', $error);
            redirect("login/orderUpdateForm/$id");
        }
    }
}

```

```

public function orderPaymentDetails($id)
{
    //loads the payment details of a particular order
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if ($this->checkSessionExist()) {

        $result = $this->user_model->fetchPaymentRecord($id);

        if ($result) {
            $data['records'] = $result;

            $this->load->view('order-management/payment-details.php', $data);
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function reports()
{
    //open reports page
    if ($this->checkSessionExist()) {

        $this->load->view('reports/reports.php');
    } else {

        $this->load->view('auth/login', $data);
    }
}

public function credit()
{
    //opens the credit management page
    if ($this->checkSessionExist()) {

        $data['credit'] = $this->user_model->fetchCreditTermsDB();

        $this->load->view('credit-management/credit-management.php', $data);
    } else {

        $this->load->view('auth/login', $data);
    }
}

```

```

public function delCredit($id)
{
    //deactivate a selected credit term
    if ($this->checkSessionExist()) {

        $result = $this->user_model->deactivateCredit($id);

        if ($result == 1) {
            $data['success'] = $this->session->set_flashdata('success', 'Successfully deactivated');
            redirect('login/credit/');
        } elseif ($result == 0) {
            $data['error'] = $this->session->set_flashdata('error', 'Deactivation unsuccessful');
            redirect('login/credit/');
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'Error occured please try again');
            redirect('login/credit/');
        }
    } else {
        $this->load->view('auth/login', $data);
    }
}

public function addcredit()
{
    //credit term management page with the credit term addition section
    $success = $this->session->flashdata('success');
    $error = $this->session->flashdata('error');
    $data = [];
    if (!empty($success)) {
        $data['success'] = $success;
    }
    if (!empty($error)) {
        $data['error'] = $error;
    }
    if (isset($data['error']) || isset($data['success'])) {

        $data['credit'] = $this->user_model->fetchCreditTermsDB();

        $this->load->view('credit-management/credit-management.php', $data);
    } else {
        if ($this->checkSessionExist()) {

            $data['credit'] = $this->user_model->fetchCreditTermsDB();

            $this->load->view('credit-management/add-credit-management.php', $data);
        } else {
            $this->load->view('auth/login', $data);
        }
    }
}

```

```

public function addCreditSubmit()
{
    //submit the new credit term details and create a new credit term
    $this->form_validation->set_rules('ov', 'ov', 'required');
    $this->form_validation->set_rules('bl', 'bl', 'required');
    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('error', 'Form details cannot be empty');
        redirect("/login/credit/");
    } else {
        //print_r($_POST);

        $data = array(
            'overdue' => $_POST['ov'],
            'blacklist' => $_POST['bl'],
        );

        $result = $this->user_model->addCredit($data);

        if ($result == 1) {
            $data['success'] = $this->session->set_flashdata('success', 'Credit Term added successfully.');
            redirect('login/credit');
        } elseif ($result == 0) {
            $data['success'] = $this->session->set_flashdata('success', 'Action Unsuccessful');
            redirect('login/credit');
        } else {
            $data['error'] = $this->session->set_flashdata('error', 'Error occured please try again');
            redirect('login/credit');
        }
    }
}

private function checkSessionExist()
{
    //check if the session of a given user exists
    if (!$this->session->has_userdata('userinfo')) {
        $this->session->set_flashdata('error', 'Please login first to access the page');
        redirect('login/userlogin');
    } else {
        return true;
    }
}

public function logout()
{
    //log out of the session
    $this->session->unset_userdata('userinfo');
    $this->session->set_flashdata('success', 'Logout successful.');

    redirect('login/userlogin');
}

public function getCharts()
{
    //open the charts page
    $result = $this->product_model->getMarks();
    $data['students'] = $result;
    $this->load->view('/chart/charts_view', $data);
}
}

```

```

<?php

class Reports extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('user_model');
    }

    public function customerPdf()
    {
        //generate customer pdf file
        $bootstrapCss = file_get_contents("http://localhost/MediTech/css/bootstrap.min.css");
        $customCss = file_get_contents("http://localhost/MediTech/css/custom/public-dashboard.css");
    }
}

```

```

$result = $this->user_model->fetchCustomerDB();

$html = "<html>
    <head>
        <style>
            {$bootstrapCss}
            {$customCss}
            body {
                font-size: 14px;
            }
            .mt-4 {
                margin-top: 1rem !important;
            }
            .innercont {
                margin: 1rem !important;
                padding: 1rem;
            }
            .table {
                width: 100%;
                margin-bottom: 1rem;
                color: #212529;
                background-color: transparent;
            }
            .table-bordered {
                border: 1px solid #dee2e6;
            }
            .table-bordered th, .table-bordered td {
                border: 1px solid #dee2e6;
            }
            .table-striped tbody tr:nth-of-type(odd) {
                background-color: rgba(0, 0, 0, 0.05);
            }
            .thead-dark th {
                color: #fff;
                background-color: #343a40;
            }
            .text-danger {
                color: #dc3545 !important;
            }
            .text-success {
                color: #28a745 !important;
            }
        </style>
    </head>
    <body>

```

```

        <div class='mt-4 innercont p-5'>
            <h2> Customer Details </h2>
            <table class='table table-striped'>
                <thead class='thead-dark'>
                    <tr>
                        <th>Customer ID</th>
                        <th>Customer Name</th>
                        <th>Email</th>
                        <th>Contact Number</th>
                        <th>Status</th>
                        <th>Registered Date</th>
                    </tr>
                </thead>
                <tbody>";
            }

            foreach ($result as $customer) {
                $statusClass = ($customer->status == "blacklisted") ? "text-danger" : "text-success";

                $html .= "<tr>
                            <td>{$customer->custID}</td>
                            <td>{$customer->name}</td>
                            <td>{$customer->email}</td>
                            <td>{$customer->contactNo}</td>
                            <td class='{$statusClass}'>{$customer->status}</td>
                            <td>{$customer->registered_date}</td>
                        </tr>";
            }

            $html .= "</tbody></table></div></body></html>";

            //dompdf

            $customPaper = array(0, 0, 800, 700);

            $this->load->library('pdf');

            $this->pdf->loadHtml($html);
            $this->pdf->setPaper($customPaper, 'landscape');
            $this->pdf->render();
            $this->pdf->stream("sample.pdf", array("Attachment" => 0));
        }
    }
}

```

```
public function orderPdf()
{
    //generate order pdf file
    $bootstrapCss = file_get_contents("http://localhost/MediTech/css/bootstrap.min.css");
    $customCss = file_get_contents("http://localhost/MediTech/css/custom/public-dashboard.css");

    $result = $this->user_model->fetchodersDB();

    $html = "<html>
        <head>
            <style>
                {$bootstrapCss}
                {$customCss}
                body {
                    font-size: 14px;
                }
                .mt-4 {
                    margin-top: 1rem !important;
                }
                .innercont {
                    margin: 1rem !important;
                    padding: 1rem;
                }
                .table {
                    width: 100%;
                    margin-bottom: 1rem;
                    color: #212529;
                    background-color: transparent;
                }
                .table-bordered {
                    border: 1px solid #dee2e6;
                }
                .table-bordered th, .table-bordered td {
                    border: 1px solid #dee2e6;
                }
                .table-striped tbody tr:nth-of-type(odd) {
                    background-color: rgba(0, 0, 0, 0.05);
                }
                .thead-dark th {
                    color: #fff;
                    background-color: #343a40;
                }
                .text-danger {
                    color: #dc3545 !important;
                }
                .text-success {

```

```

        color: #28a745 !important;
    }
    .text-success {
        color: #28a745 !important;
    }

```

```

</style>
</head>
<body>
    <div class='mt-4 innercont p-5'>
        <h2> Order Details </h2>
        <table class='table table-striped'>
            <thead class='thead-dark'>
                <tr>
                    <th>Order ID</th>
                    <th>Customer ID</th>
                    <th>Value</th>
                    <th>Status</th>
                    <th>Created Date</th>
                </tr>
            </thead>
            <tbody>";

```

```

foreach ($result as $order) {

    $isPaidClass = ($order->isPaid == "0") ? "text-danger" : "text-success";
    $statusClass = ($order->isPaid == "0") ? "unpaid" : "paid";

    $html .= "<tr>
        <td>{$order->id}</td>
        <td>{$order->custID}</td>
        <td>{$order->value}</td>
        <td class='{$isPaidClass}'>{$statusClass}</td>
        <td>{$order->created_date}</td>
    </tr>";
}

```

```

$html .= "</tbody></table></div></body></html>";

//dompdf

$customPaper = array(0, 0, 800, 700);

$this->load->library('pdf');

$this->pdf->loadHtml($html);
$this->pdf->setPaper($customPaper, 'landscape');
$this->pdf->render();
$this->pdf->stream("sample.pdf", array("Attachment" => 0));

}

```

```
public function blacklistpdf()
{
    //generate pdf file for the blacklisted customers
    $bootstrapCss = file_get_contents("http://localhost/MediTech/css/bootstrap.min.css");
    $customCss = file_get_contents("http://localhost/MediTech/css/custom/public-dashboard.css");

    $result = $this->user_model->fetchBlacklistedCustomers();

    $html = "<html>
        <head>
            <style>
                {$bootstrapCss}
                {$customCss}
                body {
                    font-size: 14px;
                }
                .mt-4 {
                    margin-top: 1rem !important;
                }
                .innercont {
                    margin: 1rem !important;
                    padding: 1rem;
                }
                .table {
                    width: 100%;
                    margin-bottom: 1rem;
                    color: #212529;
                    background-color: transparent;
                }
                .table-bordered {
                    border: 1px solid #dee2e6;
                }
                .table-bordered th, .table-bordered td {
                    border: 1px solid #dee2e6;
                }
                .table-striped tbody tr:nth-of-type(odd) {
                    background-color: rgba(0, 0, 0, 0.05);
                }
                .thead-dark th {
                    color: #fff;
                    background-color: #343a40;
                }
                .text-danger {
                    color: #dc3545 !important;
                }
            </style>
        </head>
        <body>
            <div class="innercont">
                <table border="1" style="width: 100%; border-collapse: collapse;">
                    <thead>
                        <tr style="background-color: #343a40; color: white; font-weight: bold;">
                            <th>Customer ID</th>
                            <th>Customer Name</th>
                            <th>Email Address</th>
                            <th>Phone Number</th>
                            <th>Address</th>
                            <th>City</th>
                            <th>State</th>
                            <th>Zip Code</th>
                            <th>Country</th>
                            <th>Last Login</th>
                            <th>Status</th>
                            <th>Actions</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>1</td>
                            <td>John Doe</td>
                            <td>johndoe@example.com</td>
                            <td>(123) 456-7890</td>
                            <td>123 Main St</td>
                            <td>Anytown</td>
                            <td>CA</td>
                            <td>95201</td>
                            <td>USA</td>
                            <td>2022-01-01</td>
                            <td>Active</td>
                            <td><a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a></td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </body>
    </html>";
}
```

```

        .text-success {
            color: #28a745 !important;
        }
    </style>
</head>
<body>
<div class='mt-4 innercont p-5'>
<h2> Blacklisted Customers</h2>
    <table class='table table-striped'>
        <thead class='thead-dark'>
            <tr>
                <th>Customer ID</th>
                <th>Customer Name</th>
                <th>Email</th>
                <th>Contact No</th>
                <th>Registered Date</th>
            </tr>
        </thead>
        <tbody>";
    foreach ($result as $customer) {

        $html .= "<tr>
                    <td>{$customer->custID}</td>
                    <td>{$customer->name}</td>
                    <td>{$customer->email}</td>
                    <td>{$customer->contactNo}</td>
                    <td>{$customer->registered_date}</td>
                </tr>";
    }

    $html .= "</tbody></table></div></body></html>";

    $customPaper = array(0, 0, 800, 700);

    $this->load->library('pdf');

    $this->pdf->loadHtml($html);
    $this->pdf->setPaper($customPaper, 'landscape');
    $this->pdf->render();
    $this->pdf->stream("sample.pdf", array("Attachment" => 0));
}
}

```

```
<?php
session_start();
class User_Authentication extends CI_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->helper('form');
        $this->load->library('form_validation');
        $this->load->model('user_model');
        $this->load->library('session');
    }
    public function index() {
        $this->load->view('contactus');

    }

    // Show registration page
    public function user_registration_show() {
        $this->load->view('registration_form');
    }

    // Validate and store registration data in database
    public function new_user_registration() {

    }

    // Check for user login process
    public function user_login_process() {

    }

    // Logout from user page
    public function logout() {

    }
}

?>
```

7.3 View

This section includes the view of two of the most critical modules of the system. The remaining codes will not be disclosed due to confidentiality issues.

1. Blacklist

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/icon?family=MaterialIcons" rel="stylesheet" />
    <link href="<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet" />
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/custom/nav.css">
    <link href="<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/header.php'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">

            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Blacklist</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="row">
                        <div class="mt-4 jumbotron p-5 me-4">
                            <?php echo form_open('login/searchBlacklistsSubmit'); ?>

                            <div class="form-group mb-4 d-flex">
                                <input type="text" name="value" class="form-control" placeholder="Search customer name">

```

```

        <input type="submit" class="btn btn-primary px-4 mx-4"
value="Search"></input>
    </div>
    <?php echo form_close(); ?>

    <!-- table view -->

    <table class="table">

        <thead>
            <tr>
                <th>Customer ID</th>
                <th>Customer Name</th>
                <th>Email</th>
                <th>Contact Number</th>
                <th>Registered Date</th>
            </tr>
        </thead>

        <tbody>
            <?php foreach ($customer as $row) : ?>

            <tr>
                <td><?php echo $row->custID; ?></td>
                <td><?php echo $row->name; ?></td>
                <td><?php echo $row->email; ?></td>
                <td><?php echo $row->contactNo; ?></td>
                <td><?php echo $row->registered_date; ?></td>
            </tr>
            <?php endforeach; ?>
        </tbody>

    </table>

    </div>

    </div>
    </div>
    </div>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/icon?family=MaterialIcons" rel="stylesheet" />
    <link href="<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet"
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/custom/nav.css">
    <link href="<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/nav.php'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">

            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Blacklist</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="row">
                        <div class="mt-4 jumbotron p-5 me-4">
                            <?php echo form_open('login/searchBlacklistSubmit'); ?>

                            <div class="form-group mb-4 d-flex">
                                <input type="text" name="value" class="form-control"
placeholder="Search customer name">

```

```
        <input type="submit" class="btn btn-primary px-4 mx-4" value="Search"></input>
    </div>
    <?php echo form_close(); ?>

    <!-- table view -->

    <table class="table">
        <thead>
            <tr>
                <th>Customer ID</th>
                <th>Customer Name</th>
                <th>Email</th>
                <th>Contact Number</th>
                <th>Registered Date</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>John Doe</td>
                <td>john.doe@example.com</td>
                <td>(123) 456-7890</td>
                <td>2023-10-01</td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

2. Order Module

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/bootstrap.min.css>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet"
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/custom/nav.css">
    <link href=<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />
    <script src="https://code.jquery.com/jquery-3.7.1.js"
        integrity="sha256-eKhayi8LEQwp4NKxN+CfCh+3qOVUtJn3QNz0TciWLP4="
        crossorigin="anonymous"></script>

    <script>
        var eimage = null;
        var aprica = null;
        var etitle = null;
        var eproductID = null;
    </script>
    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/navbar'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">
            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Create an order</h1>

                <div class="container">
                    <div class="row">

                        <div class="mt-4 jumbotron p-5 me-4">
                            <?php
                                if (isset($success)) {
                                    echo "<div class='alert alert-success'>";
                                    echo $success;
                                    echo "</div>";
                                }
                            </?php
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
```

```

        }

        if (isSet($error)) {
            echo "<div class='alert alert-danger'>";
            echo $error;
            echo "</div>";
        }

    ?>

        <!-- order addition card -->

        <h2>Customer Details</h2>
        <div class="row alert">
            <label class="">Name : <?php echo $customer[0]->name ?></label>
            <label class="mt-2">Email : <?php echo $customer[0]->email ?></label>
            <label class="mt-2">Contact number : <?php echo $customer[0]->contactNo ?></label>
        </div>

        <div class="row">

            <section class="">
                <div class="container py-3 mb-5">
                    <div class="row d-flex justify-content-center align-items-center">
                        <div class="col-11">
                            <?php echo form_open('login/addOrderSubmit/' . $customer[0]->custID, array('method' => 'post')) ?>
                            <div class="d-flex justify-content-between align-items-center mb-4">

                                <select id="productDropdown"
name="product_id" class="form-control" style="width:300px">
                                    <?php $i = 0; ?>
                                    <?php foreach ($products as $product) : ?>
                                        <option value="<?php echo $product->productID; ?>"><?php echo $product->title; ?>
                                    </option><span></span>
                                    <?php $i++; ?>
                                <?php endforeach; ?>
                            </select>

                            <div>
                                <script>
                                    selectedValue = null;
                                </script>
                            </div>
                        </div>
                    </div>
                </div>
            </section>
        </div>
    </div>
</body>
</html>

```

attribute of the "Add Product" link

// Function to update the href

function updateAddProductLink() {

```
document.getElementById("productDropdown");

products.find(product => product.productID === selectedValue);

object

?>login/ajaxRequestHandler",
{

ducts);

RESPONSE.products;

{
products[0]; // Assuming you want the first product

product.price;
product.title;
product.productID;

the img, price, and title variables as needed
}

atitle, img, productID);

textStatus, errorThrown) {
Request Failed: " + errorThrown);
}

const dropdown =
selectedValue = dropdown.value;
//const selectedProduct =
//console.log(selectedValue);
//console.log(selectedProduct);

const data = {
value: selectedValue
};

// Create an XMLHttpRequest

$.ajax({
url: "<?php echo site_url('",
data: data,
method: "GET",
success: function(response)
{
console.log(response);

var products =
if (products.length > 0)
var product =
img = product.img;
price =
title =
productID =
// Now you can use
}

console.log(price,
atitle, img, productID);

},
error: function(jqXHR,
jqXHR.responseText));
});
```



```

echo "<option
value='".$option->id."'>$msg</option>";
}
}
}

?>

</select>

</div>

<div class="form-group col-md-3">
Rs 0</span>
</div>
</div>
</div>

<input type="hidden" name="total_order_value"
id="total_order_value" value="0">

<button type="submit" class="btn btn-primary
btn-block btn-lg" onclick="calculateTotalOrderValue()">
Proceed to Pay</button>

<?php echo form_close(); ?>

</div>
</div>
</div>
</section>

</div>
</div>

<!-- <div class=" mt-4 dropdown p-5 me-4"> -->
<!-- table view -->
<!-- TODO -->

```



```

const newProductCard = document.createElement("div");
newProductCard.className = "card";
newProductCard.innerHTML = cardTemplate;
newProductCard.id = productCardId;
newProductCard.setAttribute("data-product-id", product.productId);
newProductCard.setAttribute("data-quantity", quantity);
newProductCard.setAttribute("data-price", price);
newProductCard.setAttribute("data-total", total);
newProductCard.setAttribute("data-image", image);
newProductCard.setAttribute("data-label", label);
newProductCard.setAttribute("data-quantity-label", quantityLabel);
newProductCard.setAttribute("data-total-label", totalLabel);

// Append the new product card to the container
const container = document.getElementById("cont2");
container.appendChild(newProductCard);
}*/
```

function updatePrice(cardID, price) {

```

let cont = "form" + cardID;
const quantityContainer = document.getElementById(cont);

let currentQuantity = parseInt(quantityContainer.textContent, 10);
currentQuantity++;
quantityContainer.textContent = currentQuantity;

console.log(cardID);
```

let labelID = "price" + cardID;

```

console.log(labelID);
const label = document.getElementById(labelID);

label.textContent = "Rs " + price * currentQuantity;
let t = calculateTotalOrderValue();
document.getElementById("totalIndicator").innerHTML = "Rs " + t;
```

}

function updateQuantity(productId, quantityID) {

```

let cont = "form" + quantityID;
const quantityContainer = document.getElementById(cont);

let currentQuantity = parseInt(quantityContainer.textContent, 10);
document.getElementById('quantity_' + productId).value = currentQuantity + 1;
}
```

function addProductCard() {

```

// Generate a unique product card ID and increment the counter
const productCardId = `product-card-${nextProductCardId}`;

// Create a new product card element
const newProductCard = document.createElement("div");
newProductCard.id = productCardId;
newProductCard.className = "card";

// Create card body
const cardBody = document.createElement("div");
cardBody.className = "card-body p-4";

// Create the row for content
const contentRow = document.createElement("div");
contentRow.className = "row d-flex justify-content-between align-items-center";

// Image column
const imageColumn = document.createElement("div");
imageColumn.className = "col-md-2 col-lg-2 col-xl-2";
```

```

const image = document.createElement("img");
image.className = "img-fluid rounded-3";
image.src = "<?php echo site_url(); ?>/uploads/images/" + img;
image.alt = "Product Image";
imageColumn.appendChild(image);

// Title and Price column
const titlePriceColumn = document.createElement("div");
titlePriceColumn.className = "col-md-3 col-lg-3 col-xl-3";
const title = document.createElement("p");
title.className = "lead fw-normal mb-2";
title.textContent = title;
const price = document.createElement("p");

price.innerHTML = '<span class="text-muted">Price:' + price + '</span>';

titlePriceColumn.appendChild(title);
titlePriceColumn.appendChild(price);

// Quantity controls column
const quantityColumn = document.createElement("div");
quantityColumn.className = "col-md-3 col-lg-3 col-xl-2 d-flex";

const idlabel = document.createElement("label");
idlabel.className = "invisible";
idlabel.value = productID;
idlabel.textContent = nextProductCardId;
idlabel.id = price;

const quantityInput = document.createElement("input");
quantityInput.type = "hidden";
quantityInput.name = "product_quantities[" + idlabel.value + "]";
quantityInput.value = 1; // Set default quantity
quantityInput.id = "quantity_" + idlabel.value;
quantityInput.className = "quantity";

const productIdInput = document.createElement("input");
productIdInput.type = "hidden";
productIdInput.name = "product_ids[]";
productIdInput.value = productID;
productIdInput.className = "";

const quantityContainer = document.createElement("div");
quantityContainer.id = "form" + nextProductCardId; // Generate a unique container ID
quantityContainer.className = "quantity-container";
quantityContainer.textContent = 1;
quantityContainer.onclick = function() {
    let qnt = idlabel.textContent;
    let id = idlabel.value;
    let pric = idlabel.id;
    updateQuantity(id, qnt);
    updatePrice(qnt, pric);
}

```

```

        calculateTotalOrderValue()
    };
    quantityContainer.style = "height: 50px; line-height: 50px; width:100px;"

    quantityColumn.appendChild(productIdInput);
    quantityColumn.appendChild(quantityInput);

    quantityColumn.appendChild(idlabel);
    quantityColumn.appendChild(quantityContainer);

    // Price column
    const priceColumn = document.createElement("div");
    priceColumn.className = "col-md-3 col-lg-2 col-xl-2 offset-lg-1";
    priceColumn.id = "priceCol" + nextProductCardId;
    const priceValue = document.createElement("h5");
    priceValue.className = "mb-0";
    priceValue.textContent = "Rs " + arrPrice * i;
    priceValue.id = "price" + nextProductCardId;
    priceColumn.appendChild(priceValue);

    // Remove button column
    const removeColumn = document.createElement("div");
    removeColumn.className = "col-md-1 col-lg-1 col-xl-1 text-end";
    const removeButton = document.createElement("a");
    removeButton.href = "javascript:void(0)";
    removeButton.className = "text-danger";
    removeButton.onclick = function() {
        removeProductCard(productCardId);
        removeValues(idlabel.value);
    };
    removeButton.innerHTML = '<i class="fas fa-trash fa-lg"></i>';
    removeColumn.appendChild(removeButton);

    // Append all columns to the content row
    contentRow.appendChild(imageColumn);
    contentRow.appendChild(titlePriceColumn);
    contentRow.appendChild(quantityColumn);
    contentRow.appendChild(priceColumn);
    contentRow.appendChild(removeColumn);

    // Append the content row to the card body
    cardBody.appendChild(contentRow);

    // Append the card body to the new product card
    newProductCard.appendChild(cardBody);

    // Append the new product card to the container
    const container = document.getElementById("cont2");

    container.appendChild(newProductCard);

```

```

        let i = calculateTotalOrderValue();
        document.getElementById("totalIndication").innerHTML = "Rs " + i;
        nextProductCardId++;
    }

    function calculateTotalOrderValue() {
        let totalOrderValue = 0;

        // Get all product cards
        const productCards = document.querySelectorAll('[id^="product-card-"]');

        // Iterate through all product cards
        productCards.forEach(productCard => {
            // Get the quantity and price elements
            const quantityElement = productCard.querySelector('.quantity-container');
            const priceElement = productCard.querySelector('.text-muted');

            // Check if elements are found
            if (quantityElement && priceElement) {
                // Parse quantity and price values
                const quantity = parseInt(quantityElement.textContent, 10);
                const price = parseFloat(priceElement.textContent.replace('Price:', ''));

                // Check if parsing was successful
                if (!isNaN(quantity) && !isNaN(price)) {
                    // Calculate total price for the current product
                    const totalPrice = quantity * price;

                    // Add to the total order value
                    totalOrderValue += totalPrice;
                }
            }
        });

        // Display or use the total order value as needed
        console.log('Total Order Value: Rs ' + totalOrderValue.toFixed(2));
        document.getElementById('total_order_value').value = totalOrderValue.toFixed(2);
        return totalOrderValue;
    }

    // Example usage (you can trigger this function when needed, e.g., on button click)
    document.querySelector('.btn-warning').addEventListener('click', function() {
        const totalOrderValue = calculateTotalOrderValue();
        // Perform further actions with the total order value as needed
        console.log("Proceeding to pay with Total Order Value: Rs " + totalOrderValue);
    });
}

```

```

        function removeProductCard(productCardId) {
            // Find the product card by its unique ID
            const productCard = document.getElementById(productCardId);

            if (productCard) {
                // Remove the specified product card from the DOM
                productCard.remove();
            }
        }

        function removeValues(productId) {
            const productIdInput = document.querySelector('input[name="product_ids[]"][value="' + productId + '"]');
            if (productIdInput) {
                productIdInput.remove();
            }

            const quantityInput = document.getElementById("quantity_" + productId);
            if (quantityInput) {
                quantityInput.remove();
            }

            let t = calculateTotalOrderValue();
            document.getElementById("totalIndicator").innerHTML = "Rs " + t;
        }
    
```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet" />
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css" rel="stylesheet" />
    <link href="https://code.jquery.com/jquery-3.5.1.slim.min.js" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

        <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/header.php')?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php')?>
        </div>
        <div class="dashboard-content col-10">
            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Order Management</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="row">

                        <div class="mt-4 innercont p-5 me-4">

```

```

        <div class="d-flex box-button">

            <a href="=&lt;?php echo site_url(); ?&gt;/login/selectCustomer"
class="btn btn-primary shadow"&gt;
                Add Order&lt;/a&gt;
            &lt;a href="<?=&lt;?php echo site_url(); ?&gt;/login/addCustomer" class="btn
btn-primary mx-4 shadow"&gt;
                &lt;span class="material-icons custom-icon"&gt;person&lt;/span&gt; New
Customer&lt;/a&gt;

        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="mt-4 jumbotron p-5 me-4"&gt;

        &lt;?php echo form_open('login/searchOrderSubmit'); ?&gt;

        &lt;div class="form-group mb-4 d-flex"&gt;
            &lt;input type="text" name="value" class="form-control"
placeholder="Search order by customer"&gt;
            &lt;input type="submit" class="btn btn-primary px-4 mx-4"
value="Search"&gt;&lt;/input&gt;
        &lt;/div&gt;
        &lt;?php echo form_close(); ?&gt;

        &lt;!-- table view --&gt;

        &lt;table class="table"&gt;

            &lt;thead&gt;
                &lt;tr&gt;
                    &lt;th&gt;Order ID&lt;/th&gt;
                    &lt;th&gt;Customer Name&lt;/th&gt;
                    &lt;th&gt;Value&lt;/th&gt;
                    &lt;th&gt;Status&lt;/th&gt;
                    &lt;th&gt;Created Date&lt;/th&gt;
                &lt;/tr&gt;
            &lt;/thead&gt;
            &lt;tbody&gt;

                &lt;?php foreach ($orders as $order); ?&gt;

                &lt;tr&gt;

                    &lt;td&gt;&lt;?php echo $order-&gt;id; ?&gt;&lt;/td&gt;
                    &lt;td&gt;&lt;?php echo $order-&gt;name; ?&gt;&lt;/td&gt;
                    &lt;td&gt;&lt;?php echo $order-&gt;value; ?&gt;&lt;/td&gt;
                    &lt;td &lt;?php

                        if ($order-&gt;isPaid == 1) {
                            echo "class='text-success'";
                        }
                    &lt;/?php&gt;&gt;&lt;/td&gt;
                &lt;/tr&gt;
            &lt;/tbody&gt;
        &lt;/table&gt;
    &lt;/div&gt;
</pre

```

```

        } else {

            echo "class='text-danger'";
        }
    ?><?php

    if ($order->isPaid == 1) {

        echo "Paid";
    } else {

        echo "Unpaid";


        if($order->isOverdue == 1){
            echo "<div class='ms-3 pt-1 pb-1 px-2 alert
            alert-warning alert-dismissible
            type='button'
            label='Close' aria-hidden='true'>
                Overdue
            </div>";
        }
    }
    ?></td>
<td><?php echo $order->created_date; ?></td>
<td>
<div class="flex">
    <a href=<?php echo site_url();>
        <?php echo $order->id; ?>" class="btn btn-
        <?php if ($order->isPaid == 1) {?>Details <?php
    & else{?>Update<?php }?></a>
    </div>
</td>

<td>
    </td>
</td>

</tr>
<?php endforeach; ?>
</tbody>
</table>

```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href=<?php echo site_url(); ?>css/bootstrap.min.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=MaterialIcons" rel="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet" />
/>
    <link href="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link href="stylesheet" href=<?php echo site_url(); ?>css/custom/nav.css" rel="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/nav.php')?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php')?>
        </div>
        <div class="dashboard-content col-10">

            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Update Order</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">

                    <div class="row">
                        <div class=" mt-4 jumbotron p-5 me-4">

                            <h2 class="mb-5">Confirm Order Payment</h2>

```

```

<?php echo form_open_multipart('login/orderUpdateSubmit'); ?>
    <div class="row">
        <div class="col-md-6">
            <div class="form-group mb-4">
                <label>Receipt / Cheque Image</label>
                <input type="file" name="userfile" class="form-control"
placeholder="Enter customer email address" size="20" required>
                <input type="text" name="id" class="form-control
invisible d-none" value="<?php echo $orders[0]->id; ?>">
            </div>
        </div>

        <div class="col-md-6">
            <div class="form-group mb-4">
                <label>Receipt / Cheque No.</label>
                <input type="text" name="receipt" class="form-control"
placeholder="Enter receipt / cheque no.">
            </div>
        </div>

        <div class="col-md-12">
            <div class="form-group">
                <input type="submit" class="btn btn-success shadow-md me-6"></input>
            </div>
        </div>
    </div>
    <?php echo form_close(); ?>
</div>

<div class="mt-4 container p-5 me-4">

    <div class="card" style="border: none;">
        <div class="card-body">
            <div class="container mb-6 mt-3">
                <div class="row d-flex align-items-baseline">
                    <div class="col-xl-9">
                        <p style="font-size: 20px;">Invoice &gt;&gt;</p>
<strong>ID: <?php echo $orders[0]->id; ?></strong></div>
                    </div>
                </div>
                <div class="container">

                    <div class="row mt-5 mb-5">
                        <div class="col-xl-9">
                            <ul class="list-unstyled">
                                <li class="text-muted mt-2">To: <span
style=""><?php echo $orders[0]->name; ?></span></li>
                                <li class="text-muted mt-2"><i
class="fas fa-envelope"></i> <?php echo $orders[0]->email; ?></li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

                <li class="text-muted mt-2"><i
class="fas fa-phone"></i> <?php echo $orders[0]->contactNo; ?></li>
                </ul>
            </div>
            <div class="col-xl-3">
                <p class="text-muted">Invoice</p>
                <ul class="list-unstyled">
                    <li class="text-muted">
                        <i class="fas fa-circle">
                            </i> <span class="fw-bold">ID:</span>
                    <li class="text-muted">
                        <i class="fas fa-circle">
                            </i> <span class="fw-bold">Creation Date:</span>
                    <li class="text-muted">
                        <i class="fas fa-circle">
                            </i> <span class="me-1 fw-bold">Status:</span><span <?php
                                if ($orders[0]->isPaid == 1) {
                                    echo "class='text-success'";
                                } else {
                                    echo "class='text-danger'";
                                }
                            ?>><?php
                                if ($orders[0]->isPaid == 1) {
                                    echo "Paid";
                                } else {
                                    echo "Unpaid";
                                }
                            ><?php
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </div>

```

```

?></span></l
1>
        </ul>
    </div>
</div>
<bc>

<?php foreach ($products as $product): ?>
    <div class="row my-2 mx-1 justify-content-
center">
        <div class="col-md-2 mb-4 mb-md-0">
            <div class="bg-image ripple rounded-5
mb-4 overflow-hidden d-block" data-ripple-color="light">
                description; ?>" />
                <a href="#">
                    <div class="hover-overlay">
                        <div class="mask"
style="background-color: hsla(0, 0%, 98.4%, 0.2)"></div>
                    </div>
                </a>
            </div>
            <div class="col-md-7 mb-4 mb-md-0">
                <p class="fw-bold"><?php echo $product-
>title; ?> x <?php echo $product->quantity; ?></p>
                <p class="mb-1">
                    <span><?php echo $product-
>description; ?></span>
                </p>
                <p class="mb-1">
                    <span>x <?php echo $product-
>quantity; ?></span>
                </p>
            </div>
            <div class="col-md-3 mb-4 mb-md-0">
                <h5 class="mb-2">
                    <span class="align-middle">Rs <?php
echo $product->price; ?></span>
                </h5>
            </div>
        <bc>
            <?php endforeach; ?>
        <div class="row">
            <div class="col-x1-8">
                <?php $msg = "Overdue after " . $credit[0]-
>overdue_period . ' months | Blacklisted after ' . $credit[0]->blacklisted_period . ' months'; ?>
                <p class="ms-3"><?php echo $msg; ?></p>
            </div>

```

```
<div class="col-xl-3">
    <ul class="list-unstyled">
        <li class="text-muted ms-3"><span
class="text-black me-4">SubTotal</span>Rs <?php echo $orders[0]->value; ?></li>
    </ul>
    <p class="text-black float-start"><span
class="text-black me-3"> Total Amount</span><span style="font-size: 25px;">Rs <?php echo $orders[0]->value; ?></span></p>
</div>
</div>

</div>
</div>
</div>
</div>
</body>
</html>
```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/bootstrap.min.css>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet" />
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href=<?php echo site_url(); ?>css/custom/nav.css">
    <link href=<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

        <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/nav.php'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">
            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Update Order</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="row">
                        <div class="col mt-4 innercont p-5 me-4">

                            <div class="card" style="border: none;">
                                <div class="card-body">

```

```


<div class="row d-flex align-items-baseline">
        <div class="col-11-9">
            <p style="font-size: 20px;">Invoice &gt;&gt;
            <strong>ID: <?php echo $orders[0]->id;
        ?></strong></p>
    </div>
    <?php if ($orders[0]->isPaid == "0") { ?>
        <div class="col-11-3">
            <a href="<?php echo site_url('login/orderUpdateForm/'); ?>
                login/orderUpdateForm/<?php
                echo $orders[0]->id ?>">
                class="btn btn-success shadow me-6"
                style="width=100px;">Confirm Payment
                <i class="fas fa-check-circle ms-2"></i></a>
        </div>
        <?php } else { ?>
        <div class="col-11-3">
            <i class="far fa-check-circle fa-lg text-
success ms-2 pr-2"></i>
            <a href="<?php echo site_url('login/orderPaymentDetails/'); ?>
                login/orderPaymentDetails/<?php
                echo $orders[0]->id ?>">
                class="btn btn-success shadow me-6"
                style="width=100px; margin-left: 7px;">
                Payment Details<i class="fas fa-file ms-
2"></i>
            </a>
        </div>
        <?php } ?>
    </div>
    <div class="container">
        <div class="row mt-5 mb-5">
            <div class="col-11-9">
                <ul class="list-unstyled">
                    <li class="text-muted mt-2">To: <span
style="">
                    ?><span></li>
                    <?></span></li>
                    <li class="text-muted mt-2">
                        <i class="fas fa-envelope"></i>
                        <?php echo $orders[0]->email;
                    ?></li>
                    <li class="text-muted mt-2">
                        <i class="fas fa-phone"></i>
                        <?php echo $orders[0]->contactNo;
                    ?></li>
                </ul>
            </div>
            <div class="col-11-3">
                <p class="text-muted">Invoice</p>
                <ul class="list-unstyled">
                    <li class="text-muted">


```

```

        <i class="fas fa-circle"
style="color:#8f8061 ;"></i>
<span class="fw-bold">ID:<br/>
</span><?php echo $orders[0]->id;
?></li>

<li class="text-muted">
<i class="fas fa-circle"
style="color:#8f8061 ;"></i>
<span class="fw-bold">Creation Date:<br/>
</span><?php echo $orders[0]->created_date;
?></li>
<li class="text-muted">
<i class="fas fa-circle"
style="color:#8f8061;"></i>
<span class="me-1 fw-bold">
Status:</span><span <?php
if ($orders[0]->isPaid == 1) {
    echo
    "class='text-success'";
} else {
    echo
    "class='text-danger'";
}
?>><?php
if ($orders[0]->isPaid == 1) {
    echo
    "Paid";
} else {
    echo
    "Unpaid";
}
?></span></li>

```

```

<bo>

<?php foreach ($products as $product): ?>
    <div class="row my-2 mx-1 justify-content-
center">
        <div class="col-md-2 mb-4 mb-md-0">
            <div class="bg-image ripple rounded-5
mb-4 overflow-hidden d-block "
data-ripple-color="light">
                description_2>" />
                <a href="#">
                    <div class="hover-overlay">
                        <div class="mask"
style="background-color:
black; 0, 0%, 98.4%,
0.2)"></div>
                </div>
            </a>
        </div>
        <div class="col-md-7 mb-4 mb-md-0">
            <p class="fw-bold"><?php echo
$product->title_2>
            <?php echo $product->quantity_2></p>
            <p class="mb-1">
                <span><?php echo
$product->description_2></span>
            </p>
            <p class="mb-1">
                <span>x <?php echo
$product->quantity_2></span>
            </p>
        </div>
        <div class="col-md-3 mb-4 mb-md-0">
            <h5 class="mb-2">
                <span class="align-middle">Rs
                <?php echo $product->price_2>
            </span>
            </h5>
        </div>
    <bo>
        <?php endforeach; ?>
    <div class="row">
        <div class="col-xl-8">
            <?php $msg = 'Overdue after ' .
$credit[0]->overdue_period . '
months | Blacklisted after ' .

```

```

$credit[0]->blacklisted_period .
' months':=2>
<p class="ms-3"><?php echo $msg:?=*></p>
</div>
<div class="col-xl-3">
    <ul class="list-unstyled">
        <li class="text-muted ms-3">
            <span class="text-black me-4">
                SubTotal</span>Rs <?php
                echo $orders[0]->value; ?></li>
    </ul>
    <p class="text-black float-start">
        <span class="text-black me-3">
            Total Amount</span><span
            style="font-size: 25px;">Rs <?php
            echo $order[0]->value;
?></span></p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    <link href="<?php echo site_url(); ?>css/custom/dashboard.css" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" rel="stylesheet"
/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css">
    <link rel="stylesheet" href="<?php echo site_url(); ?>css/custom/nav.css">
    <link href="<?php echo site_url(); ?>css/custom/public-dashboard.css" rel="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('common/nav.php')?>
        </div>
        <div class="row">
            <?php $this->load->view('common/sidebar.php')?>
        </div>
        <div class="dashboard-content col-10">

            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Order Management</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="mt-4 mb-4 p-5 me-4" style="box-shadow: 0 0 10px rgba(0, 0,
0, 0.1);">
                        <div class="d-flex justify-content-center align-items-center">

                            <div class="rounded p-3 shadow" style="max-height: 400px; max-width:
300px;">

```

```
        <img src=<?php echo site_url(); ?>/uploads/receipts/<?php echo  
$records[0]->img; ?>" alt="Receipt Image" class="img-fluid" style="border-radius: var(--card-border-  
radius);">  
    </div>  
  
    <div class="ms-4 row">  
        <h4 class="fw-bold mb-0 col">Receipt Number:</h4>  
        <h4 class="fw-bold mb-0 col"><?php echo $records[0]->receiptNo;  
?></h4>  
  
        </div>  
        </div>  
    </div>  
    </div>  
</body>  
</html>
```

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href=<?php echo site_url(); ?>css/bootstrap.min.css>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" href="stylesheet" />
    <link href=<?php echo site_url(); ?>css/custom/dashboard.css href="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Symbols+Rounded" href="stylesheet" />
/>
    <link href="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
    <link href="stylesheet" href=<?php echo site_url(); ?>css/custom/nav.css>
    <link href=<?php echo site_url(); ?>css/custom/public-dashboard.css href="stylesheet" />

    <title>Dashboard ~ MediTech</title>
</head>

<body>

    <div class="container-fluid px-6">
        <div class="row">
            <?php $this->load->view('/common/nav.php'); ?>
        </div>
        <div class="row">
            <?php $this->load->view('/common/sidebar.php'); ?>
        </div>
        <div class="dashboard-content col-10">
            <div class="dynamic-page container m-2">

                <h1 style="font-weight: 800; font-size: 2.5rem;">Order Management</h1>
                <?php

                    if (isset($success)) {
                        echo "<div class='alert alert-success' style='z-index: -1;'>";
                        echo $success;
                        echo "</div>";
                    }
                    if (isset($error)) {
                        echo "<div class='alert alert-danger' style='z-index: -1;'>";
                        echo $error;
                        echo "</div>";
                    }
                ?>

                <div class="container">
                    <div class="row">
                        <div class="mt-4 innercont p-5 me-4">

                            <h2 class="mb-4">Select a customer</h2>
                            <?php echo form_open('login/searchCustomerSubmit'); ?>

```

```

        <div class="form-group mb-4 d-flex">
            <input type="text" name="value" class="form-control"
placeholder="Search customer name">
            <input type="submit" class="btn btn-primary px-4 mx-4"
value="Search"></input>
        </div>
        <?php echo form_close(); ?>

        <!-- table view -->

        <table class="table">

            <thead>
                <tr>
                    <th>Customer ID</th>
                    <th>Customer Name</th>
                    <th>Email</th>
                    <th>Contact Number</th>
                    <th>Registered Date</th>
                </tr>
            </thead>
            <tbody>

                <?php foreach ($customer as $row) : ?>

                <tr>
                    <td><?php echo $row->custID; ?></td>
                    <td><?php echo $row->name; ?></td>
                    <td><?php echo $row->email; ?></td>
                    <td><?php echo $row->contactNo; ?></td>
                    <td><?php echo $row->registeredDate; ?></td>
                    <td>
                        <div class="flex">
                            <a href=<?php echo site_url('
login/processOrder/'); ?>
                                login/processOrder/<?php echo $row->custID; ?>
                            " class="btn btn-success btn-sm">Select</a>
                        </div>
                    </td>
                </tr>
            <?php endforeach; ?>
            </tbody>
        </table>

    </div>
    <!--
    <div class="mt-4 mb-4 p-5 me-4">

```

```


| Order ID                   | Customer Name                | Value                         | Status                         | Created Date                        |
|----------------------------|------------------------------|-------------------------------|--------------------------------|-------------------------------------|
| <?php echo \$order->id; ?> | <?php echo \$order->name; ?> | <?php echo \$order->value; ?> | <?php echo \$order->isPaid; ?> | <?php echo \$order->createdDate; ?> |


```

7.4 Test Cases

Test Case ID	TC_06
Description	Verify that an order can be updated
Pre-Condition	User should be logged into the system and in the order management module
Test Procedure	<p>Click on “Update” next to an order record</p> <p>Select “confirm Payment”</p> <p>Select products and Click “Add product”</p> <p>Upload the receipt and enter cheque or receipt number</p> <p>Click “Submit Query”</p>
Expected Result	Order status should be updated with a success message and payment confirmation
Actual Result	Order status updated with a success message and payment confirmation
Status	Passed

The screenshot shows the MediTech Order Management System interface. On the left is a sidebar with navigation links: Customers, Blocklist, Orders, System Users, Products, Credit Terms, Reports, and Logout. The main area has two buttons: 'Add Order' and 'New Customer'. Below them is a search bar labeled 'Search order by customer' with a 'Search' button. A table lists 23 orders with columns: Order ID, Customer Name, Value, Status, and Created Date. Each row has 'Details' and 'Update' buttons.

Order ID	Customer Name	Value	Status	Created Date	Action
16	Hiruka Karasinghe	12500	Paid	2023-11-12	Details
17	hiruka	13800	Unpaid Overdue	2023-06-23	Update
18	Hiruka Karasinghe	1000	Paid	2023-11-12	Details
19	Gobbels Rowel	31200	Paid	2022-11-08	Details
20	Gobbels Rowel	9000	Paid	2023-01-18	Details
21	hiruka	12000	Unpaid	2023-12-18	Update
22	Hiruka Karasinghe	3500	Unpaid	2024-02-18	Update
23	Gobbels Rowel	250	Unpaid	2024-02-19	Update

Figure 61: Pre-Condition – Order Management – Delete Order

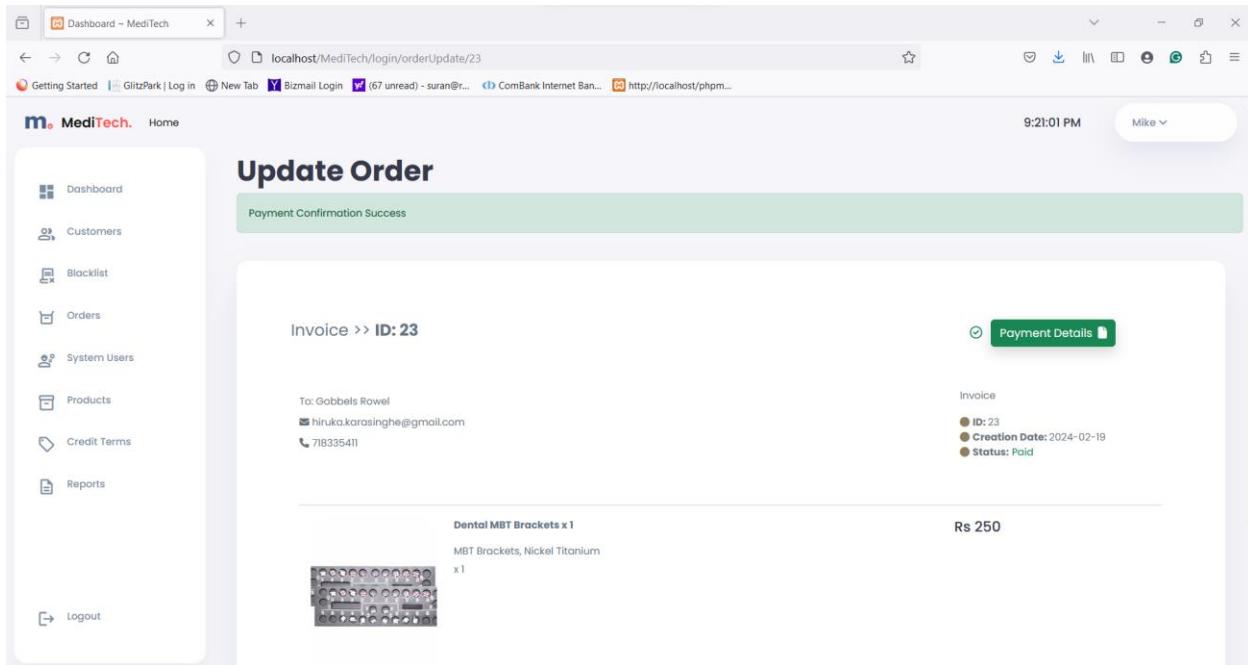


Figure 62: Post-Condition – Order Management – Delete Order – success message

The screenshot shows the MediTech application's 'Order Management' page. The header displays 'Dashboard – MediTech' and the URL 'localhost/MediTech/login/orderMng'. The main content area features two blue buttons: 'Add Order' and 'New Customer'. Below them is a search bar with placeholder 'Search order by customer' and a 'Search' button. A table lists eight orders with columns: Order ID, Customer Name, Value, Status, and Created Date. The last row, for Order ID 23, has a yellow background. Each row includes a 'Details' and an 'Update' button. The sidebar on the left is identical to Figure 61, showing 'Customers', 'Orders', and other system management options.

Order ID	Customer Name	Value	Status	Created Date	
16	Hiruka Karasinghe	12500	Paid	2023-11-12	<button>Details</button>
17	hiruka	13800	Unpaid Overdue	2023-06-23	<button>Update</button>
18	Hiruka Karasinghe	1000	Paid	2023-11-12	<button>Details</button>
19	Gobbels Rowel	31200	Paid	2022-11-08	<button>Details</button>
20	Gobbels Rowel	9000	Paid	2023-01-18	<button>Details</button>
21	hiruka	12000	Unpaid	2023-12-18	<button>Update</button>
22	Hiruka Karasinghe	3500	Unpaid	2024-02-18	<button>Update</button>
23	Gobbels Rowel	250	Paid	2024-02-19	<button>Details</button>

Figure 63: Post-Condition – Order Management – Delete order

Test Case ID	TC_07
Description	Verify that a new system user can be added
Pre-Condition	User should be logged into the system and in the user management module
Test Procedure	<p>Click on “<u>Add user</u>”</p> <p>Enter User details</p> <p>Click add user button</p>
Expected Result	New User should be visible with a success message
Actual Result	New User is visible with a success message
Status	Passed

The screenshot shows the MediTech application's User Management screen. On the left is a sidebar with navigation links: Dashboard, Customers, Blocklist, Orders, System Users (which is selected), Products, Credit Terms, and Reports. The main area has a title 'User Management'. The 'Add User' form contains fields for Name (Hanna Samarasinghe), Email (hanna@gmail.com), Contact Number (0776977790), Branch ID (1), Password (*****), and User Type (DataEntry). A blue 'Add User' button is at the bottom. Below the form is a table listing three existing users:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date	Actions
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button> <button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button> <button>Delete</button>
3	Gobbel	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button> <button>Delete</button>

Figure 64: Pre-Condition – User Management – Add user

The screenshot shows a web browser window titled "Dashboard - MediTech". The URL is "localhost/MediTech/login/userMng". The page header includes "Getting Started", "GiltzPark | Log in", "New Tab", "Bizmail Login", "(67 unread) - suran@r...", "ComBank Internet Ban...", and "http://localhost/phpm...". The time is 9:25:25 PM and the user is Mike.

The main content area is titled "User Management" and displays a message "User added successfully." Below this, there is a table titled "Add user" showing the following data:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date		
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button>	<button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button>	<button>Delete</button>
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button>	<button>Delete</button>
10	Nicky	1	2147483647	pinindiweeraratne@gmail.com	DataEntry	0000-00-00	<button>Edit</button>	<button>Delete</button>
11	Hanna Samarasinghe	1	776977790	hanna@gmail.com	DataEntry	0000-00-00	<button>Edit</button>	<button>Delete</button>

At the bottom left, there is a "Logout" link.

Figure 65: Post-Condition – User Management – Add user

Test Case ID	TC_08
Description	Verify that a system user can be edited
Pre-Condition	User should be logged into the system and in the user management module
Test Procedure	<p>Click on “Edit” next to a user record</p> <p>Enter edited user datils</p> <p>Click edit user button</p>
Expected Result	Record with edited user datils should be displayed with success message
Actual Result	Record with edited user datils is displayed with success message
Status	Passed

The screenshot shows the MediTech User Management interface. On the left is a sidebar with links: Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms, and Reports. A 'Logout' button is at the bottom. The main area has a title 'User Management' and a sub-section 'Add user'. Below is a table of users:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date	Action
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button> <button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button> <button>Delete</button>
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button> <button>Delete</button>
10	Nicky	1	2147483647	pinindiveeraratne@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>
11	Hanna Samarasinghe	1	776977790	hanna@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>

Figure 66: Pre-Condition – User Management – Edit User

The screenshot shows the MediTech User Management edit user form. The sidebar and main layout are identical to Figure 66. The 'Edit User' section contains fields for Name (Hanna Samarasinghe), Email (hanna.samara@gmail.com), Contact Number (776977790), Branch ID (1), Password (*****), and User Type (DataEntry). Below is a table of users:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date	Action
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button> <button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button> <button>Delete</button>
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button> <button>Delete</button>

Figure 67: Pre Condition – User Management – Edit User Form

The screenshot shows a web browser window for 'Dashboard - MediTech' at the URL 'localhost/MediTech/login/userMng'. The page title is 'User Management'. A green success message 'User edited successfully' is displayed. Below it is a table titled 'Add user' containing the following data:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date	Action
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button> <button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button> <button>Delete</button>
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button> <button>Delete</button>
10	Nicky	1	2147483647	pinindiweeraratne@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>
11	Hanna Samarasinghe	1	776977790	hanna.samara@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>

The right side of the table has a yellow highlight over the email address 'hanna.samara@gmail.com' for the user with ID 11. On the left sidebar, there is a navigation menu with items like Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms, Reports, and Logout.

Figure 68: Post-Condition – User Management – Edit user

Test Case ID	TC_09
Description	Verify that a system user can be deleted
Pre-Condition	User should be logged into the system and in the user management module
Test Procedure	Click on “Delete” next to a user record
Expected Result	User records should be deleted with a success message
Actual Result	User record was deleted with a success message
Status	Passed

The screenshot shows the MediTech User Management interface. On the left is a sidebar with links: Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms, and Reports. The main area has a title 'User Management' and a sub-section 'Add user'. Below this is a table with the following data:

User ID	User Name	Branch ID	Contact Number	Email	User Type	Registered Date	Action
1	Mike	2	718398731	hiruka.karasinghe@gmail.com	Admin	2023-10-26	<button>Edit</button> <button>Delete</button>
2	Rowel	2	718398731	hiruka.sales@gmail.com	Sales	2023-10-28	<button>Edit</button> <button>Delete</button>
3	Gobbels	2	718398731	hiruka.entry@gmail.com	DataEntry	2023-10-28	<button>Edit</button> <button>Delete</button>
10	Nicky	1	2147483647	pinindiweeraratne@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>
11	Hanna Samarasinghe	1	776977790	hanna@gmail.com	DataEntry	0000-00-00	<button>Edit</button> <button>Delete</button>

At the bottom left of the main area is a 'Logout' button.

Figure 69: Pre-Condition – User Management – Delete Order

The screenshot shows the MediTech User Management interface after a user has been deleted. The main area displays a green success message: 'User deleted successfully.' Below this is the same user list table as in Figure 69. A yellow banner at the bottom of the main area displays the message 'Record deleted'.

Figure 70: Post-Condition – User Management – Delete Order

Test Case ID	TC_10
Description	Verify that a new product can be added
Pre-Condition	User should be logged into the system and in the product management module
Test Procedure	<p>Click on "Add product"</p> <p>Enter product details</p> <p>Click "add product"</p>
Expected Result	A new product should be added with a success message
Actual Result	A new product was added with a success message
Status	Passed

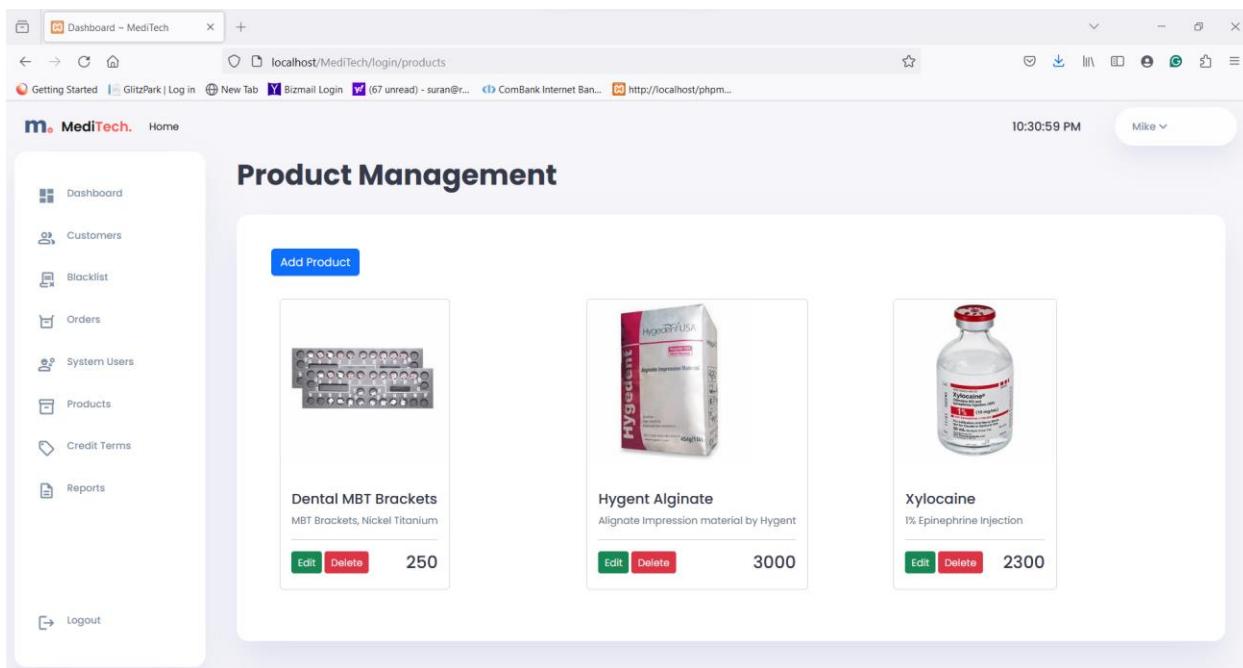


Figure 71: Pre-Condition – Product Management – Add Product

Product Management

Product Title: Dental floss

Product Image: dental floss.jpg

Description: A tool used to clean gaps between teeth

Price: 1000

Add Product

Figure 72: Pre-Condition – Product Management – Add Product Form

Product Management

Product added successfully.

Product Image	Product Name	Description	Price
	Dental MBT Brackets	MBT Brackets, Nickel Titanium	250
	Hygent Alginate	Alginate Impression material by Hygent	3000
	Xylocaine	1% Epinephrine Injection	2300

Figure 73: Post-Condition – Product Management – Add Product- success message

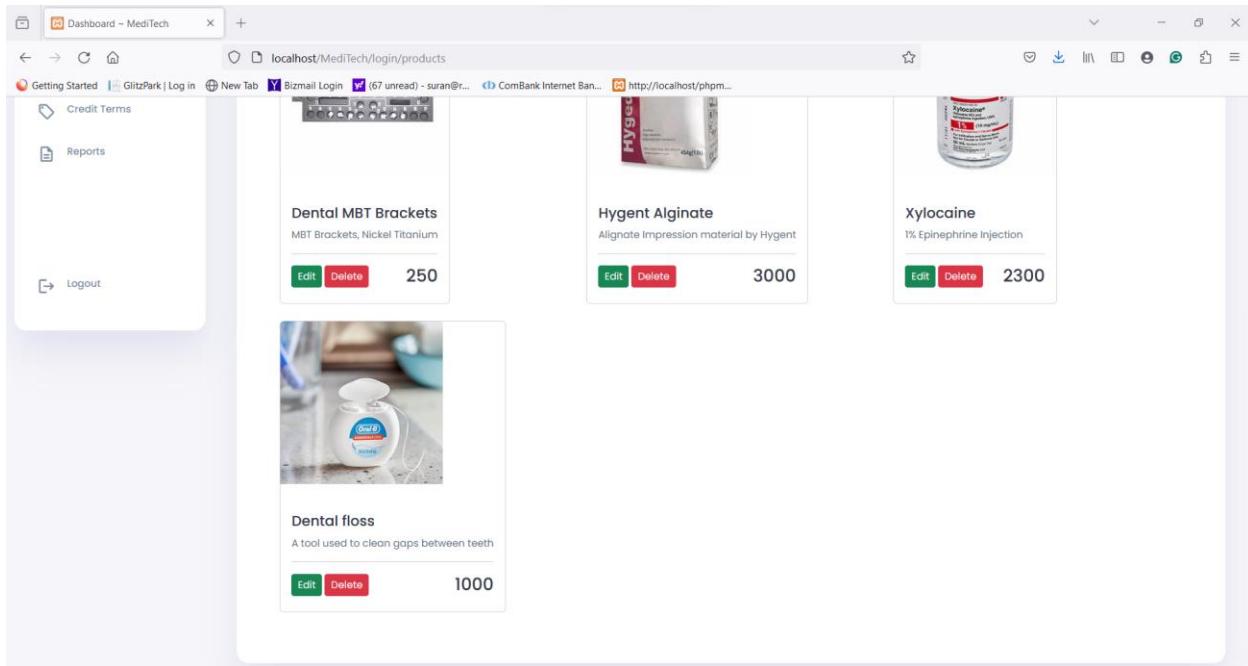


Figure 74: Post-Condition – Product Management – Add product

Test Case ID	TC_11
Description	Verify that an existing product can be edited
Pre-Condition	User should be logged into the system and in the product management module
Test Procedure	<p>Click on “Edit”</p> <p>Edit product details</p> <p>Click “edit product”</p>
Expected Result	Product details should be edited with a success message
Actual Result	Product details were edited with a success message
Status	Passed

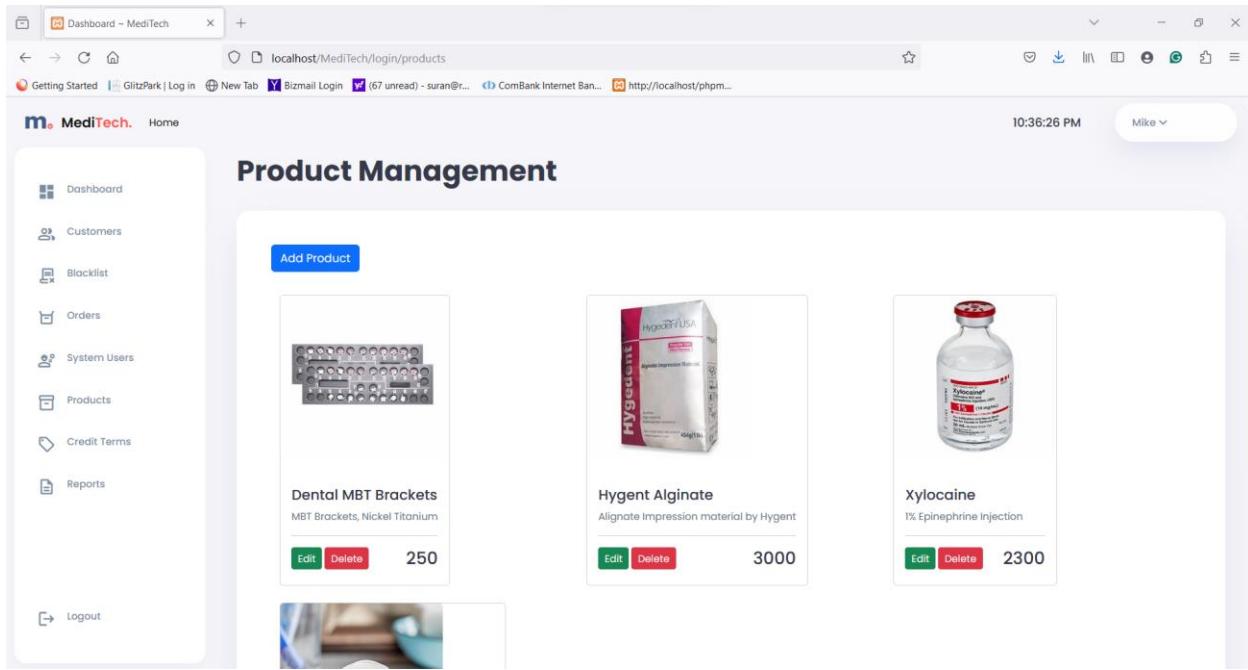


Figure 75: Pre-condition – Product Management – Edit Product

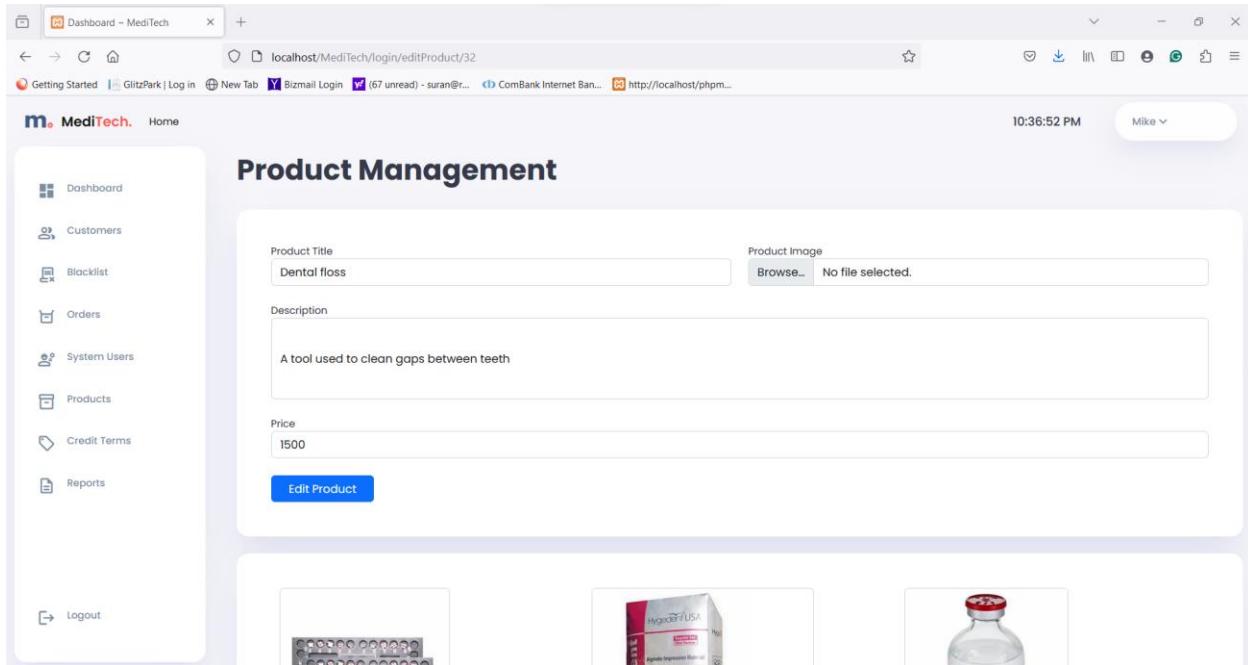


Figure 76: Pre-condition – Product Management – Edit Product - Form

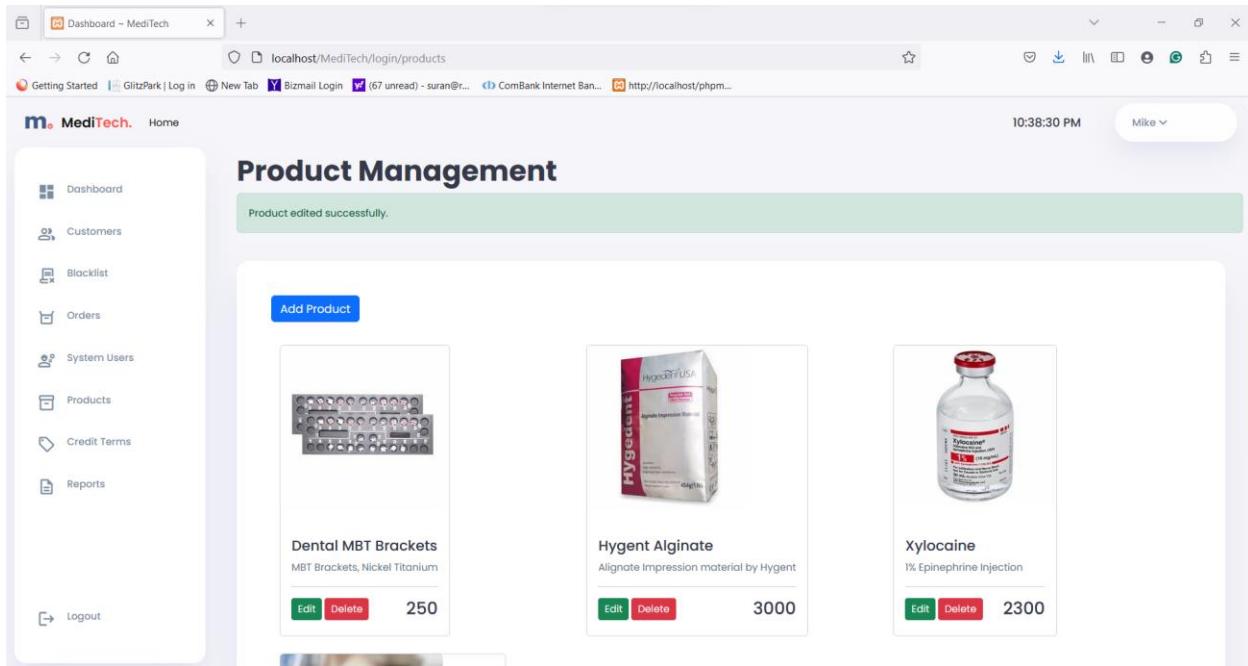


Figure 77: Post-condition – Product Management – Edit Product -success message

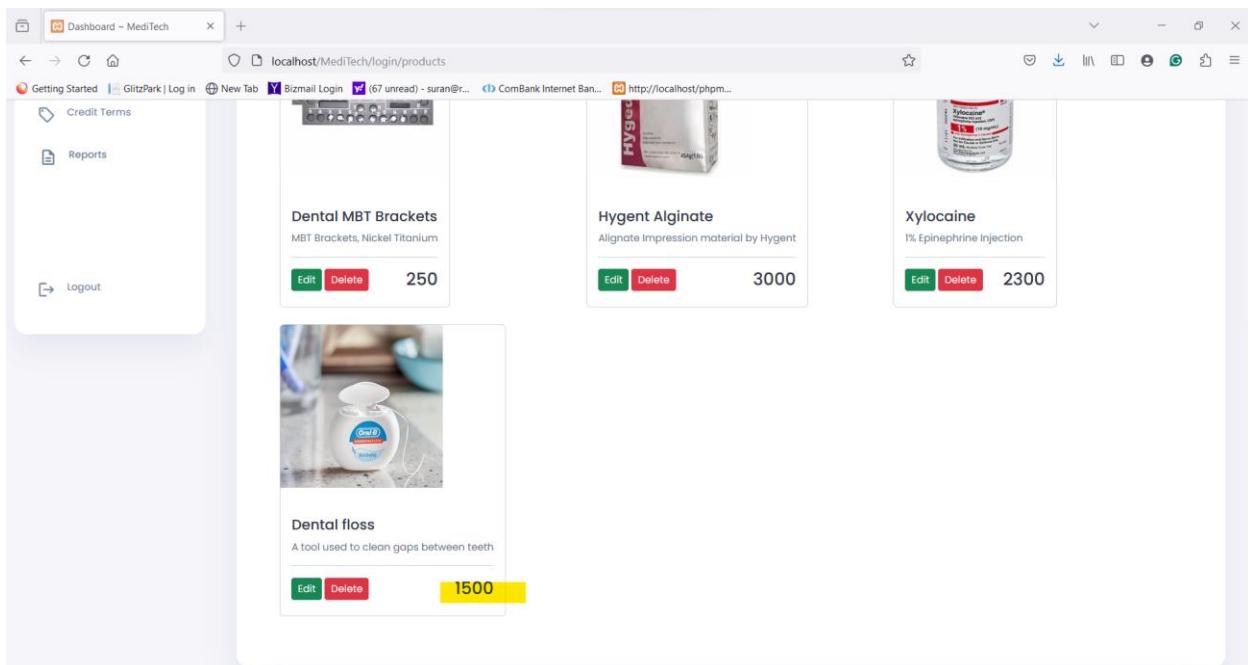


Figure 78: Post-Condition – Product Management-Edit Product

Test Case ID	TC_12
Description	Verify that an existing product can be deleted
Pre-Condition	User should be logged into the system and in the product management module
Test Procedure	Click on “delete”
Expected Result	Product should be deleted with a success message
Actual Result	Product was deleted with a success message
Status	Passed

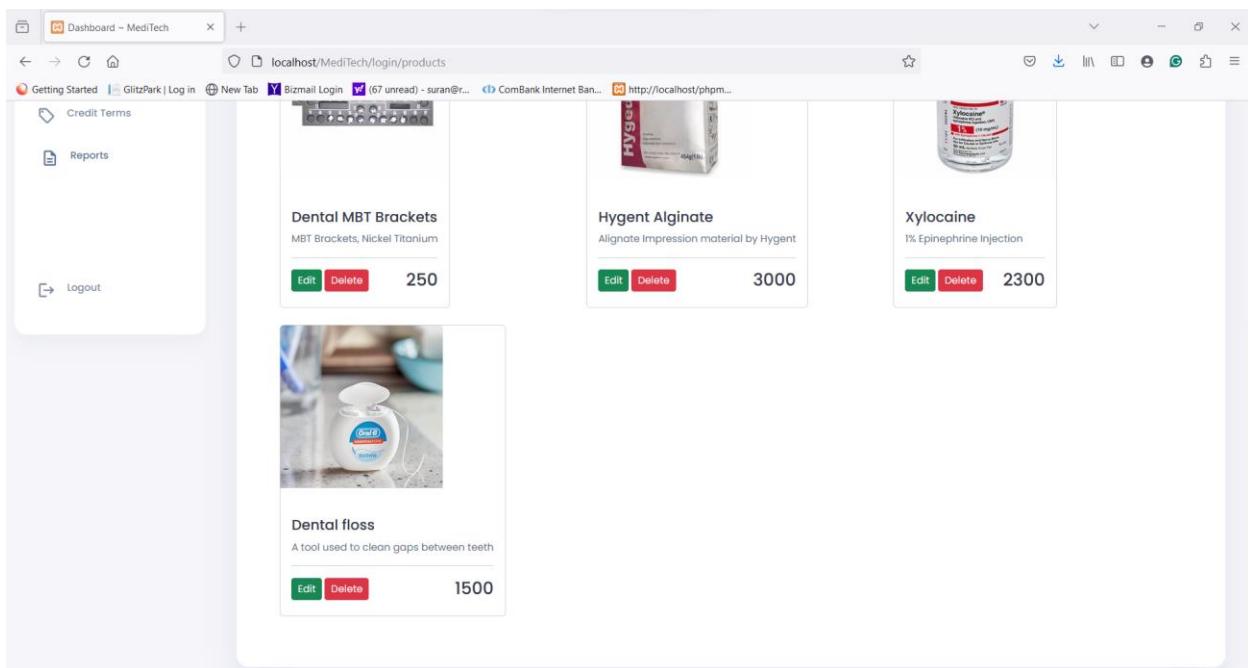


Figure 79: Pre-Condition – Product Management – Delete Product

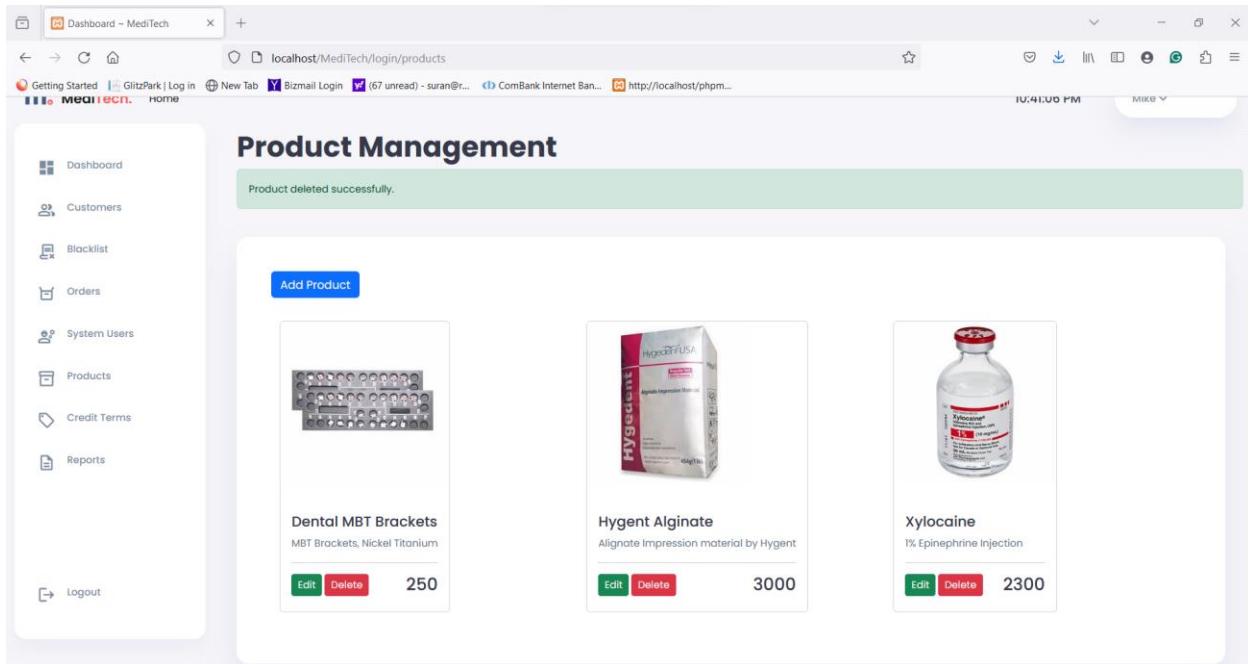


Figure 80: Post-Condition – Product Management – Delete Product

Test Case ID	TC_13
Description	Verify that new credit terms can be added
Pre-Condition	User should be logged into the system and in the credit term management module
Test Procedure	<p>Click on “Add Credit term”</p> <p>Enter credit term details</p> <p>Click “Add Credit terms”</p>
Expected Result	New credit term should be added
Actual Result	New credit term was added
Status	Passed

The screenshot shows the MediTech application's Credit Term Management page. On the left, a sidebar menu includes options like Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms (which is selected and highlighted in blue), and Reports. A Logout link is also present. The main content area has a title 'Credit Term Management' and a sub-section 'Add Credit Term'. Below this, there is a table listing three existing credit terms:

Credit Term ID	Overdue Time Period (M)	Blacklisting Time Period (M)
1	2	3
2	3	6
5	4	8

Each row in the table contains a red 'Deactivate' button on the right side.

Figure 81: Pre-Condition – Credit Term Management – Add Credit Term

This screenshot shows the same Credit Term Management page as Figure 81, but with a focus on the 'Add Credit Term' form. The sidebar and the list of existing credit terms are identical. The 'Add Credit Term' section now includes two input fields: 'Months required to Overdue' (set to 5) and 'Months required to Blacklist' (set to 7). Below these fields is a blue 'Add Credit Term' button. The rest of the page remains the same, displaying the existing credit term data.

Figure 82: Pre-Condition – Credit Term Management – Add Credit Term - Form

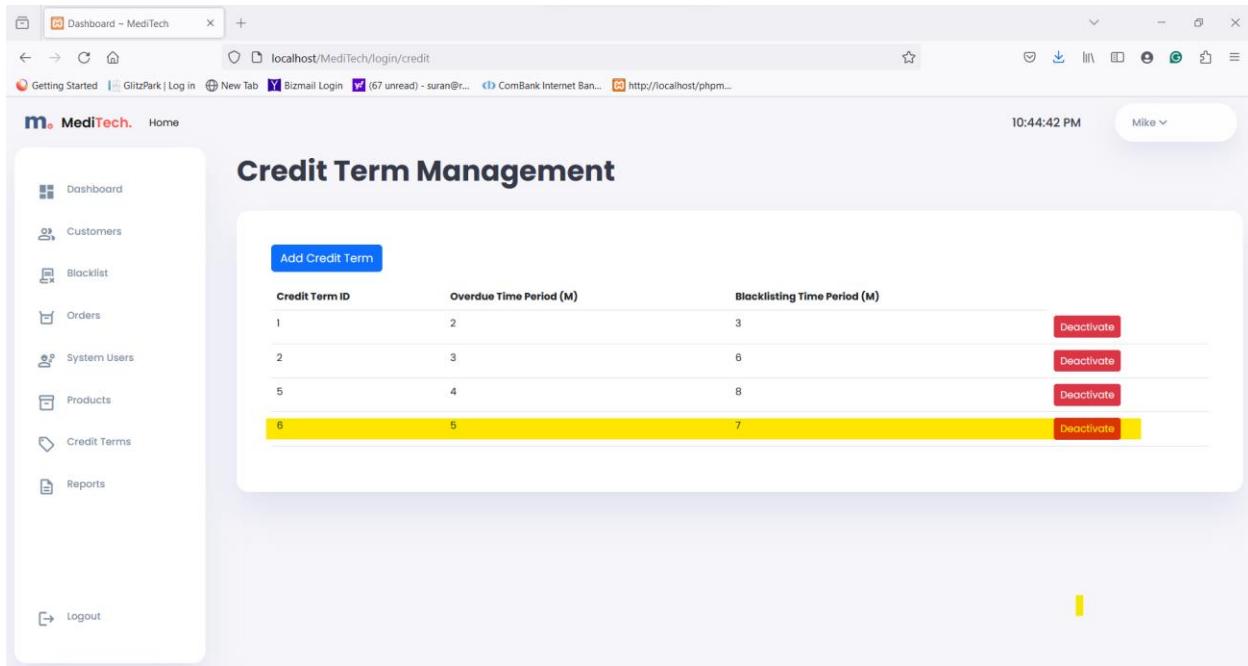


Figure 83: Post-Condition – Credit Term Management – Add Credit Term

Test Case ID	TC_14
Description	Verify that credit terms can be deleted
Pre-Condition	User should be logged into the system and in the credit term management module
Test Procedure	Click on “deactivate”
Expected Result	Credit term should be deleted
Actual Result	Credit term was deleted
Status	Passed

The screenshot shows the MediTech application's Credit Term Management screen. On the left is a sidebar with links: Dashboard, Customers, Blocklist, Orders, System Users, Products, Credit Terms (which is the active tab), and Reports. Below the sidebar is a 'Logout' link. The main content area has a title 'Credit Term Management' and a 'Add Credit Term' button. A table lists four credit terms:

Credit Term ID	Overdue Time Period (M)	Blacklisting Time Period (M)
1	2	3
2	3	6
5	4	8
6	5	7

Each row contains a red 'Deactivate' button.

Figure 84: Pre-Condition – Credit Term Management – Deactivate

This screenshot shows the same Credit Term Management screen as Figure 84, but with a yellow success message 'Record deleted' displayed prominently at the bottom of the table area.

Figure 85: Post-Condition – Credit Term Management – Deactivate

Test Case ID	TC_15
Description	Verify that reports can be generated
Pre-Condition	User should be logged into the system and in the report generator module
Test Procedure	Select report type Click on "Generate report"
Expected Result	<u>Report should be generated</u>
Actual Result	Report was generated
Status	Passed

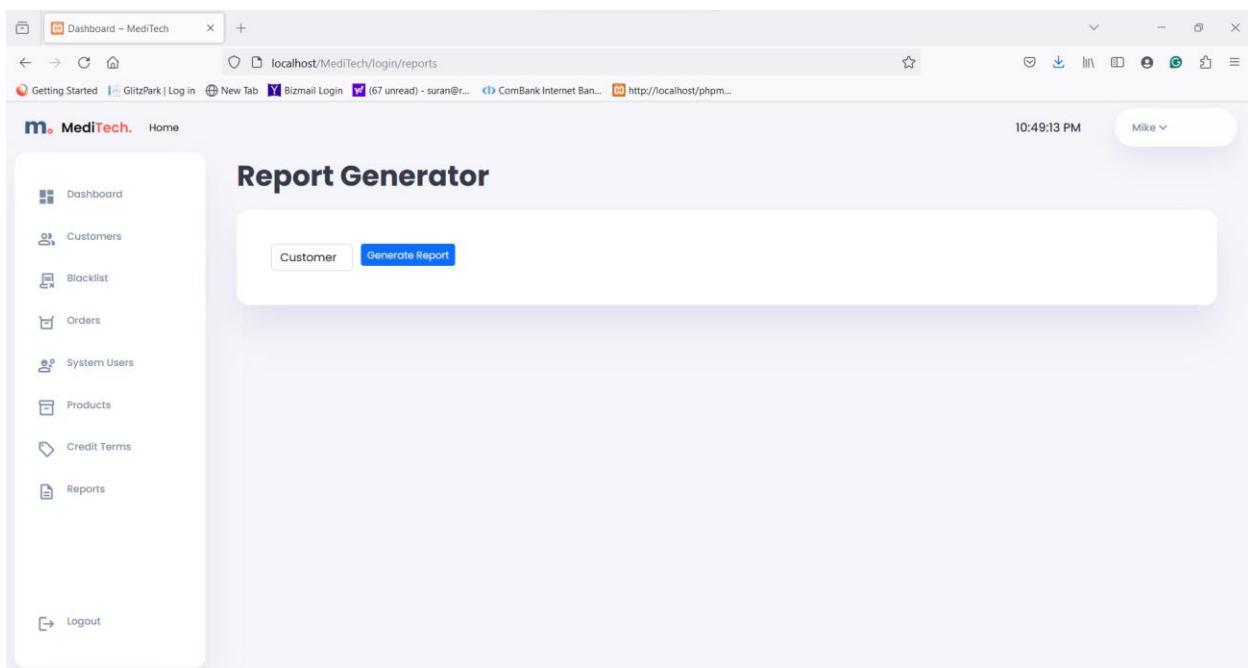
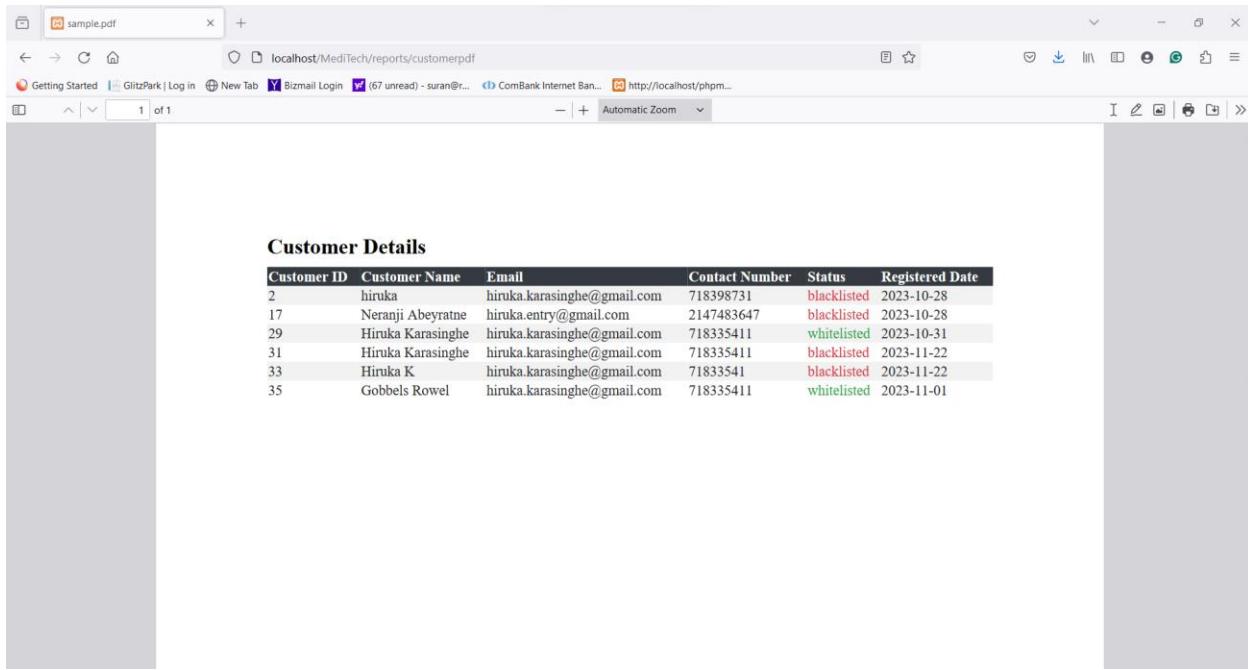


Figure 86: Pre-Condition – Report Generator

**Customer Details**

Customer ID	Customer Name	Email	Contact Number	Status	Registered Date
2	hiruka	hiruka.karasinghe@gmail.com	718398731	blacklisted	2023-10-28
17	Neranji Abeyratne	hiruka.entry@gmail.com	2147483647	blacklisted	2023-10-28
29	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-10-31
31	Hiruka Karasinghe	hiruka.karasinghe@gmail.com	718335411	blacklisted	2023-11-22
33	Hiruka K	hiruka.karasinghe@gmail.com	71833541	blacklisted	2023-11-22
35	Gobbels Rowel	hiruka.karasinghe@gmail.com	718335411	whitelisted	2023-11-01

Figure 87: Post-Condition – Report Generator

Test Case ID	TC_16
Description	Verify that users can log out
Pre-Condition	User should be logged into the system
Test Procedure	Click “Logout”
Expected Result	User should be logged out
Actual Result	User was logged out
Status	Passed

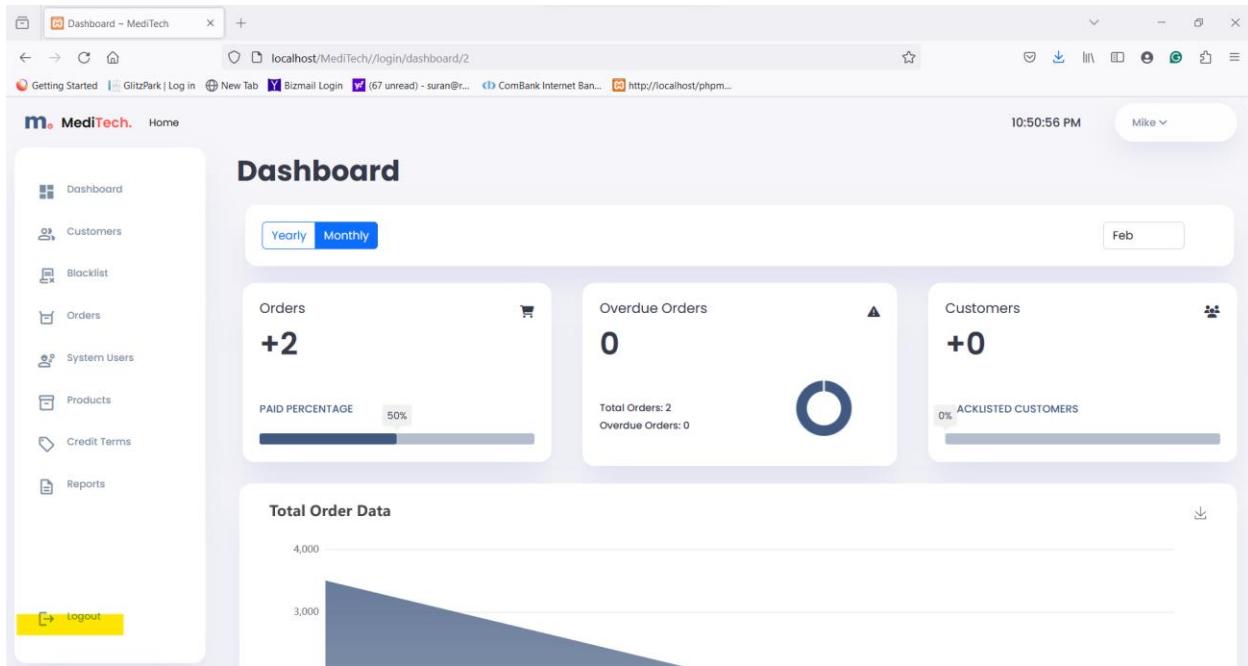


Figure 88: Pre-Condition – Logout

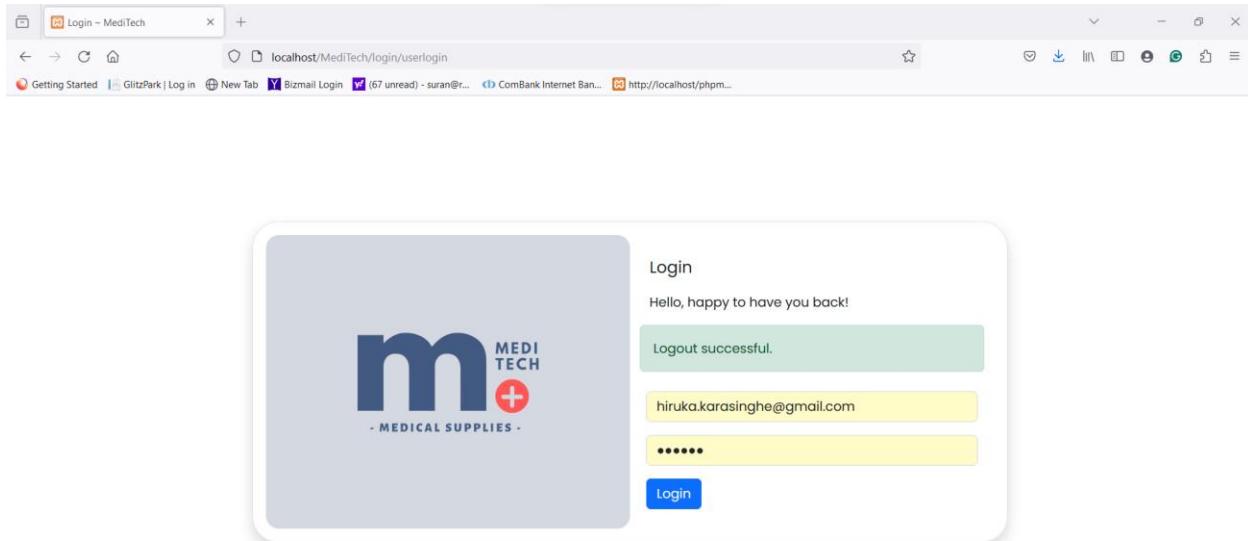


Figure 89: Post-Condition – Logout

7.5 Timeline

Task	Week Wk 37 & 38, 2023	Wk 39 & 40, 2023	Wk 41 & 42, 2023	Wk 43 & 44, 2023	Wk 45 & 46, 2023	Wk 47 & 48, 2023	Wk 49 & 50, 2023	Wk 51 & 52, 2023	Wk 01 & 02, 2024	Wk 03 & 04, 2024	Wk 05 & 06, 2024
Problem definition and requirement analysis											
Design											
Implementation											
Testing											
User Training											
Deployment											
Documentation											
System maintenance											

7.6 Check – list

1. The word count (excluding appendixes) is on the title page.
2. Correct evidence of the testing has been included.
3. Both referencing and citing of external sources have been carried out.
4. Professional issues have been correctly considered and described.
5. Appropriate timelines have been included.
6. Professional use has been made of the appendix material.
7. Appropriate advice and tutoring have been sought.
8. Group work has been appropriately described.

✓
✓
✓
✓
✓
✓
✓
✓
✓
✓

References

<https://www.youtube.com/watch?v=ybXulmeilFM&pp=ygUpc2lkZSBtZW51IGRlc2lnbiB1c2luZyBib290c3RyYXAgY3NzIGH0bWw%3D>

<https://www.youtube.com/watch?v=ybXulmeilFM&pp=ygUpc2lkZSBtZW51IGRlc2lnbiB1c2luZyBib290c3RyYXAgY3NzIGH0bWw%3D>

<https://youtu.be/bFvfqUMjvsA>

<https://www.youtube.com/watch?v=XhqTdlQJhAs>

<https://www.youtube.com/watch?v=7on734pbKxM>

<https://www.youtube.com/watch?v=aLjAyF0N-GE>

<https://www.w3schools.com/>

<https://www.techtarget.com/>