

A

**PROJECT REPORT**

**ON**

**INVENTORY MANAGEMENT SYSTEM OF MAZGAON MEDICAL**

**Submitted in partial fulfilment of the requirement for qualifying**

**National Institute of Electronics & Information Technology**

**IT 'A' LEVEL**



**Submitted by:**

**Nikash Majumdar**

**Registration No.: 1107564**

**Under the Supervision and Guidance of:**

**Amarnath Sah**

**Tezpur EC Of NIELIT Guwahati,**

**Near Indian Statistical Institute (ISI),**

**Vill: Punioni, Solmara,**

**Tezpur, Assam, 784028**

**PERFORMA FOR A / B / C LEVEL PROJECT CERTIFICATE  
FROM PROJECT GUIDE / ACCREDITED INSTITUTE**

[For Direct as well as candidate from Accredited Institute]

This is to certify that the Project / Dissertation entitled, Inventory Management System of Margaon Medical is a bonafide work done by Mr. / Ms. Bikash Majumdar (NIELIT Registration No: 1107564) in partial fulfillment of A Level / B Level / C Level examination and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

29/8/2022

Signature of Guide / Supervisor

Name:

AMARNATH SAH

Place: Tezpur

Designation: Technical Asst.

Date: 29/8/2022

Address:

Tezpur University  
Napam, Sonitpur, Assam

784028

Signature of Center Manager  
[in case of a candidate from Accredited Institute]



# CERTIFICATE BY GUIDE

---

This is to certify that **Bikash Majumdar** of **IT A Level Roll No. 1107564** has successfully completed the project entitled "**Inventory Management System Of Mazgaon Medical**" submitted in the partial fulfilment of the requirement for the degree of IT A level . This project is done under my supervision and guidance. This work has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

29/08/2022

Place: Tezpur

Date: 29-08-2022

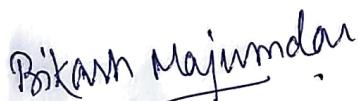
Amarnath Sah  
Technical Assistant,  
Tezpur University, Napam,  
Assam, 784028

# DECLARATION BY THE CANDIDATE

---

I hereby declare that the report for this project entitled "**Inventory Management System Of Mazgaon Medical**" is submitted by **Bikash Majumdar** to National Institute of Electronics & Information Technology (NIELIT), Tezpur in partial fulfillment of the requirement for the award of the IT 'A' level is a original record of bonafide projects work carried out by me under the guidance of **Mr. Amarnath Sah**.

I further declare that the work reported in this project has not been submitted before and will not submitted to other inistitute or university in fullfilment of the project work.



Name: **Bikash Majumdar**

Reg . No: **1107564**

Level: **IT 'A'**

Place: **Tezpur**

Date: **07 - 09 - 2022**

# ACKNOWLEDGEMENT

---

I would like to express my sincere gratitude to all those who supported me throughout in preparing this projects. I am thankful for their inspiration, guidance, invaluable constructive criticism and friendly advice during this project work. I am sincerely grateful to them for sharing their truthful and illuminating views on the number of issues related to this project.

I would like to thank Raju Sharma, centre in charge of the NIELIT, Tezpur for providing all necessary facilities and encouragement.

I would like to sincerely thank my mentor and guide Mr. Amarnath Sah for his precious time, great insight, expert guidance and valuable suggestion in every stage in this project. I would like to thank all the faculty of NIELIT, Tezpur for providing me with the required facilities and support towards the completion of the project.

Finally, it is a great pleasure to extend my sincere gratitude to my parent for their constant support and encouragement and last but not the least, friends and well wishers for their help, cooperation and solutions to problems during the course of the project.

Thank you.

Bikash Majumdar

Name: Bikash Majumdar

Reg . No: 1107564

Level: IT 'A'

(NIELIT, TEZPUR)

# List Of Chapters

CONTENT	Page
<b>Chapter 1: Introduction And Overview</b>	
1.1: Introduction	1
1.2: Objective	1
1.3: Scope	2
<b>Chapter 2: Background Study</b>	
2.1: HTML	3
2.2: PHP	4
2.2:1: Common Uses Of PHP	5
2.2:2: Characteristics Of PHP	5
2.2:3: Hello World In PHP	5
2.3: SQL	6
2.3:1: What is SQL?	6
2.3:2: Why SQL?	6
2.3:3: SQL Process	7
2.3:4: SQL Commands	8
2.4: Back end: SQL-server 2008	9
<b>Chapter 3: System Planning</b>	
3.1: Introduction	11
3.2: PERT chart	11
3.3: Conclusion	12
<b>Chapter 4: Feasibility Study</b>	
4.1: Introduction	13
4.2: Economic Feasibility	13
4.3: Technical Feasibility	13
4.4: Behavioural Feasibility	13
4.5: Conclusion	14
<b>Chapter 5: System Requirement Specification</b>	
5.1: Introduction	15
5.2: Purpose	15
5.3: Scope	15
5.4: Benefits	15
5.5: Objectives	16

CONTENT	Page
5.6: Software And Hardware Environment	16
<b>Chapter 6: System Analysis And Design</b>	
6.1: Software Engineering Paradigm	17
6.2: System Design	18
6.3: Data Flow Diagram	19
6.3.1: Types Of DFD	20
6.4: Context level DFD	20
6.5: 1st level DFD	21
6.6: 2nd level DFD	22
6.7: Data Dictionary	
6.7.1: backup_time table	27
6.7.2: customer table	27
6.7.3: inflow table	27
6.7.4: outflow table	28
6.7.5: login table	28
6.7.6: orders table	29
6.7.7: owner_info table	29
6.7.8: productid table	30
6.7.9: sale table	30
6.7.10: purchase table	31
6.7.11: supplierid table	31
6.8: E-R diagram	32
<b>Chapter 7: Input And Output Screen</b>	
7.1: Login Screen	35
7.1.1: Forgot Password Popup	35
7.2: Mazgaon Medical	
7.2.1: Owner Info Update Screen	36
7.2.2: User Management Screen	36
7.2.2.1: Update User Info Popup	37
7.2.2.2: Update Password Popup	37
7.2.2.3: Delete User Popup	38
7.2.3: Backup & Restore Screen	38
7.3: Inventory	
7.3.1: Inventory Management	39
7.3.2: Inventory Details	39
7.4: Receipts & Sales	
7.4.1: Receive Payment	40
7.4.1.1: Receive Payment Popup	40
7.4.1.2: Receive Payment History Popup	41

CONTENT	Page
7.4.2. Bill Details	41
7.4.2. Customer Details	42
7.5: Payments & Purchases	
7.5.1: Make Payment	42
7.5.1.1: Make Payment Popup	43
7.5.1.2: Make Payment History Popup	43
7.5.2: Supplier Details	44
7.6: Reports	
7.6.1: Customer Report	44
7.6.2: Purchase Report	45
7.6.3: Product Sale Report	45
7.7: Create Order	46
7.8: Generate Receipt	
7.8.1: Adding Products	46
7.8.1.1: Finalize Order Popup	47
7.8.2: Receipt Print	47
7.9: Logout	48
<b>Chapter 8: Testing</b>	
8.1: Introduction	49
8.2: Objectives Of Testing	49
8.3: Testing Methods	50
8.3.1: Unit Testing	50
8.3.2: Integration Testing	50
8.3.3: Validation	50
8.3.4: Output Testing	51
8.3.5: Random Data Testing	51
8.3.6: User Acceptance Testing	51
8.3.7: Security Mechanism	51
<b>Chapter 9: Future Scope And Further Enhancement</b>	
9.1: Introduction	53
9.1.1: Future Scope	53
9.1.2: Future Scope Of Enhancement	53
<b>Chapter 10: Code Snippets</b>	
10.1: owner_info.php	54
10.2: owner_info_process.php	56
10.3: auth.php	57
10.4: config.php	57
10.5: login_process.php	57

CONTENT	Page
<b>Chapter 11: Bibliography And Reference</b>	
<b>11.1: Bibliography</b>	<b>60</b>
<b>11.1.1: Books</b>	<b>60</b>
<b>11.1.2: Websites</b>	<b>60</b>
<b>11.1.3: YouTube Tutorials</b>	<b>60</b>

## List Of Tables

CONTENT	Page
<b>Structured Query language</b>	
Table 2.3.2: DDL-Data Definition Language	8
Table 2.3.3: DML- Data Manipulation Language	9
Table 2.3.4: DCL- Data Control Language	9
<b>System Planning</b>	
Table 3.1.1: Activities Along With Estimated Time	11
<b>System Analysis And Design</b>	
Table 6.3: Notation Of DFD	19
Table 6.3.1: Types Of DFD	20
Table 6.8.1: Notation of E-R Diagram	32
<b>Data Dictionary</b>	
Table 6.7.1: backup_time table	27
Table 6.7.2: customer table	27
Table 6.7.3: inflow table	27
Table 6.7.4: outflow table	28
Table 6.7.5: login table	28
Table 6.7.6: orders table	29
Table 6.7.7: owner_info table	29
Table 6.7.8: productid table	30
Table 6.7.9: sale table	30
Table 6.7.10: purchase table	31
Table 6.7.11: supplierid table	31

## List Of Diagrams

CONTENT	Page
Fig 2.3.1: SQL Architecture	8
Fig 3.2.1: PERT chart	11
Fig 6.1.1: Software Engineering Paradigm	18
Fig 6.4.1: Context Level DFD	20
Fig 6.5.1: 1st level DFD	21
Fig 6.6.1: Login Module	22
Fig 6.6.2: Owner Module	22
Fig 6.6.3: Customer Module	23
Fig 6.6.4: Product Module	23
Fig 6.6.5: Sale Module	24
Fig 6.6.6: Order Module	24
Fig 6.6.7: Inflow Module	25
Fig 6.6.8: Outflow Module	25
Fig 6.6.9: Reports Module	26
Fig 6.6.10: Receipts Module	26
Fig 6.8.1: E-R diagram	34
Fig 7.1: Login Screen	35
Fig 7.1.1: Forgot Password Popup	35
Fig 7.2.1: Owner Info Update Screen	36
Fig 7.2.2: User Management Screen	36
Fig 7.2.2.1: Update User Info Popup	37
Fig 7.2.2.2: Update Password Popup	37
Fig 7.2.2.3: Delete User Popup	38
Fig 7.2.3: Backup & Restore Screen	38
Fig 7.3.1: Inventory Management	39
Fig 7.3.2: Inventory Details	39
Fig 7.4.1: Receive Payment	40
Fig 7.4.1.1: Receive Payment Popup	40
Fig 7.4.1.2: Receive Payment History Popup	41
Fig 7.4.2: Bill Details	41
Fig 7.4.3: Customer Details	42
Fig 7.5.1: Make Payment	42
Fig 7.5.1.1: Make Payment Popup	43
Fig 7.5.1.2: Make Payment History Popup	43
Fig 7.5.2: Supplier Details	44
Fig 7.6.1: Customer Report	44
Fig 7.6.2: Purchase Report	45
Fig 7.6.2: Product Sale Report	45
Fig 7.7: Create Order	46
Fig 7.8.1: Adding Products	46

<b>CONTENT</b>	<b>Page</b>
Fig 7.8.2: Finalize Order Popup	47
Fig 7.8.3: Receipt Print	47
Fig 7.9: Logout	48
Fig 8.3.6: Beta Testing Report	52

## Chapter 1: Introduction And Overview

### 1.1: Introduction

IMS (Inventory Management System) is a computer software that manages all the processes and stores information of inventory starting from the purchase date to their sale, creditors and receipt generation. The IMS software can be used as a web tool or installed on a personal computer or devices and stores and manages captured data, as well as producing reports from that data.

IMS is a software that allows us to effectively manage the inventory in a store providing accurate information regarding purchases, sales, creditors and receipt generation. The software also helps in generating reports of the details in a systematic manner.

The main purpose of IMS is to improve the maintenance of inventory in the store. Using IMS would cut down the need for the manual task thereby making the task easier and simple.

The software does not only make the management of inventory easier and simple but also makes the task safe and secure as only admin and user can login with valid user id and password there by the chance of misuse of data.

### 1.2: Objective

The objective of the project is to explain and maintain clear details of various kind of medicines regarding their inventory maintenance and also the list of the customers to whom the medicines will be sold and also the suppliers from whom the medicines will be acquired, so as to minimize labour and time and make the task easier.

- To have attractive and secure login page to access.
- Only the admin and users are allowed to access the information of the inventory.

- Search inventory (medicine) details easily.
- Update inventory details easily.
- Add new inventory details easily.
- Update and maintain customer list.
- Update and maintain supplier list
- Maintain record of creditors and debtors.
- To generate sales and purchase report.
- To view bill details.
- To create backup and also restore if necessary.

### **1.3: Scope**

It may help in collecting the absolute details in a very short time. The updating and maintenance will be obvious, simple, easier and less time consuming. It will help the user to know the details of the inventory along with the bill details. It currently provides facilities such as add, update, search and delete to accomplish the desired objectives. The present project has been developed to meet the aspiration indicated in the modern age. An attempt has been made through this project to do all work, quickly and easily. The facility included in this project and the suggested activities have been organized to impart knowledge and develop skill and attitude in the inventory management in a business.

## Chapter 2: Background Study

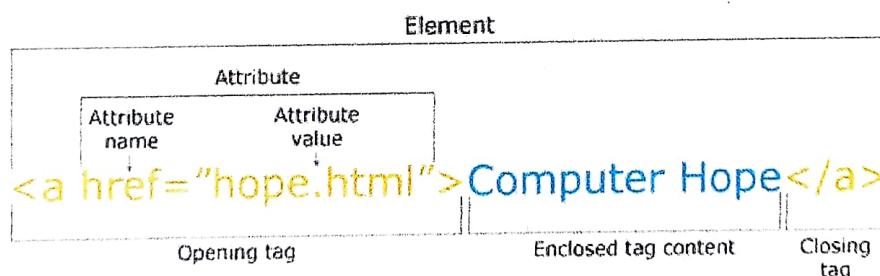
### 2.1: HTML

First developed by *Tim Berners-Lee* in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

What does an HTML tag look like?

### Breakdown of an HTML Tag



As can be seen in the HTML tag example above, there are not many components. Almost all HTML tags have an opening tag that contains the name with any attributes, a close tag that contains a forward slash, and the name of the tag that is being closed.

For tags that do not have a closing tag like <img>, it is best practice to end the tag with a forward slash.

Each tag is contained within a less than and greater than angle brackets, and everything between the opening and closing tag is displayed or affected by the tag. In the example above, the <a> tag is creating a link called "Computer Hope" that is pointing to the hope.html file.

```
<html>

<head>
    <title>Page Title</title>
</head>

<body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
</body>

</html>
```

## 2.2: PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. **Rasmus Lerdorf** unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "*PHP: Hypertext Preprocessor*".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

#### **2.2.1: Common uses of PHP**

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete or modify elements within your database through PHP.
- Access cookies, variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

#### **2.2.2: Characteristics of PHP**

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

#### **2.2.3: Hello World in PHP**

To get a feel for PHP, we will first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst our normal HTML (or XHTML if you're cutting-edge) we'll have PHP statements like this:

```
1 <html>
2   <head>
3     <title>Hello World</title>
4   </head>
5   <body>
6     <?php echo "HELLO WORLD";?>
7   </body>
8 </html>
```

It will produce following result:

HELLO WORLD

### 2.3: SQL

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language, but there are many different versions of the SQL language.

#### 2.3.1: What is SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, PostgreSQL and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as:

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

#### 2.3.2: Why SQL?

SQL is widely popular because it offers the following advantages:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.

- Allows to be embedded within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

#### A Brief History of SQL:

- 1970 – Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- 1974 – Structured Query Language appeared.
- 1978 – IBM worked to develop Codd's ideas and released a product named System/R.
- 1986 – IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

#### **2.3.3: SQL Process**

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

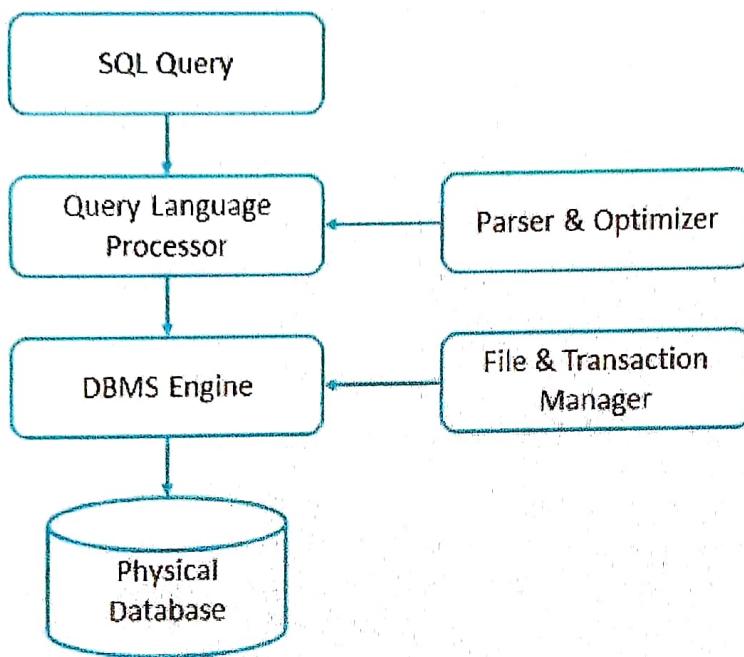
There are various components included in this process.

These components are:

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture:



*Fig: 2.3.1: SQL Architecture*

#### 2.3.4: SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature:

- **DDL - Data Definition Language:**

Sl. No.	Command	Description
1	CREATE	Creates a new table, a view of a table, or other object in the database.
2	ALTER	Modifies an existing database object, such as a table.
3	DROP	Deletes an entire table, a view of a table or other objects in the database.

*Table 2.3.2 DDL*

- **DML - Data Manipulation Language:**

Sl. No.	Command	Description
1	SELECT	Retrieves certain records from one or more tables.
2	INSERT	Creates a record.
3	UPDATE	Modifies record.
4	DELETE	Deletes record.

Table 2.3.3 DML

- **DCL - Data Control Language:**

Sl. No.	Command	Description
1	GRANT	Gives a privilege to user.
2	REVOKE	Takes back privileges granted from user.

Table 2.3.4 DCL

## 2.4: BACK END: SQL-SERVER 2008

Microsoft SQL-SERVER 2008 is a full featured relational database management system that offers a variety of administrative tools to ease the burdens of database development & administration. Enterprise manager is the main administrative console for sql server installation .We can perform high level administrative function that affects one or more servers, schedule common maintenance tasks or create & modify the structure of individual database. Query analyzer offers a quick & dirty method for performing query against sql server database. SQL profiler provides a window into inner working of our database. Service manager is used to Ms-SQL servers, Ms-Distributed transaction co-coordinator & sql server agent processes. Data transformation services provides an extremely data between a Microsoft sql server installation & large variety of other formats.

**Features:**

- Database mirroring
- Online restore opportunity
- Online indexing operations
- SQL-server management studio
- Native XML support
- Security enhancement

## Chapter 3: System Planning

### 3.1: Introduction

Proper planning is a key to success of any project. For the development of the proposed system the following activities are identified and time required to complete them are estimated. The table 3.1.1 shows the different activities along with estimated time in number of weeks.

Sl. No.	Activity	Immediate Predecessor	Immediate Successor	Time required (in weeks)
1	Project Formulation	-	2	2
2	Requirement Analysis	1	3,4,5	4
3	Feasibility analysis	2	6	1
4	System Analysis	2	6	6
5	System Design	2	6	1
6	Implementation	3,4,5	7	4
7	Testing	6	-	4

Table 3.1.1: Activities along with estimated time

### 3.2: PERT (Programme Evaluation and Review Technique) Chart

The time dependency among the different activities listed in the above table can be represented in the following PERT chart given in figure 3.2.1.

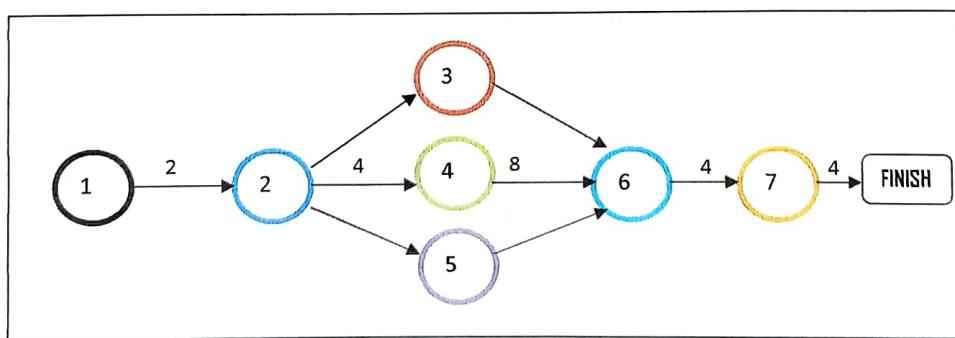


Fig 3.2.1: PERT Chart

Activities in the critical path are 1, 2, 4, 6, 7.

Total Project Duration = Length of the critical path =  $(2+4+8+4+4)$  weeks=22 weeks

### **3.3: Conclusion**

The estimated minimum duration for the project conforms to the specified duration of NIELIT "A" project work. It is expected that the project can be completed during this time.

## Chapter 4: Feasibility Analysis

### 4.1: Introduction

Feasibility study gives an idea on the workability of a system. It is the measure of impact of a system on the ability to meet the user needs and the effective use of the available resources. The object of a feasibility study is to use design and development of the proposed system. The key considerations that are involved in the feasibility analysis are economic feasibility, technical feasibility and behavioral feasibility.

### 4.2: Economic Feasibility

In economic feasibility it is determined whether the benefit expected from the proposed system outweighs the cost involved in developing it. The cost of the proposed system is much less than the benefits that can be derived from the system which mostly manual. The major advantage of the proposed system is that it will considerably reduce time, energy, and space involved in the present system. On the other hand, since no new hardware and software are to be procured for the system and the system is being developed as a student project, the cost is negligible. Hence it was concluded that the system is economically feasible.

### 4.3: Technical Feasibility

The hardware and software required to develop and deployment are already available at NIELIT, Tezpur. It has a high-speed intranet which will enable clients to access the data base from anywhere within the NIELIT intranet. Therefore, it was concluded that the system is technically feasible.

### 4.4: Behavioral Feasibility

Behavioral feasibility determines whether all the users of the organization would adopt themselves with the system easily or resist it. Since our system is very user-friendly its application can be easily taught to the users of the organization. Moreover many of

the users already have good computer skill. So, there is no reason why the users will oppose the system. Thus the proposed system is behaviorally feasible.

#### **4.5: Conclusion**

From the observations made in the feasibility study described above it was recommended that the proposed system is feasible for its development and implementation. It can be justified that this study should be followed by the analysis and system design phase of the software development.

## **Chapter 5: System Requirement Specification (SRS)**

### **5.1: Introduction**

A IMS (Inventory Management System) is a software that allows us to effectively manage the inventory in a business providing accurate information regarding stock, customers, sales etc of the commodities. The software also helps in generating reports of the details in a systematic manner.

### **5.2: Purpose**

This SRS document contains the complete software requirements for the Inventory Management System and describes the design decision, architectural design and the detailed designed needed to implement the system. It provides the visibility in the design and provides information needed for software support. The main purpose of IMS is to improve the maintenance of inventory. Using IMS would cut down the need for the manual task thereby making the task easier and simple.

### **5.3: Scope**

Inventory management system is developed for general purpose and used to replace old paper work system and project unit measurement system. Inventory management system is to built upon the existing information system projects unit management system in order to efficiently provide the inventory information along with their proper management. It currently provides facilities such as add, update, search and delete to accomplish the desired objectives. It may help in collecting the absolute details in a very short time. The upgradation and maintenance will be obvious, simple, easier and less time consuming.

### **5.4: Benefits**

- Better performance by the employees.
- Simplifying and streamlining all the tasks like updating and maintenance of inventory.
- Managing the suppliers and customers.

- Complete information of the inventory.
- Cut down the need for the manual task.
- Minimize the time required.

## 5.5: Objectives

- To have attractive and secure login page to access.
- Only the admin and users are allowed to access the information of the inventory.
- Search inventory (medicine) details easily.
- Update inventory details easily.
- Add new inventory details easily.
- Update and maintain customer list.
- Update and maintain supplier list
- Maintain record of creditors and debtors.
- To generate sales and purchase report.
- To view bill details.
- To create backup and also restore if necessary.

## 5.6: Software And Hardware Environment

### Software used

Software package	: XAMPP Server
Back-end tool	: MySQL
Front end tools	: PHP, HTML, CSS, JavaScript
Operating System	: MS Windows 10

### Hardware used

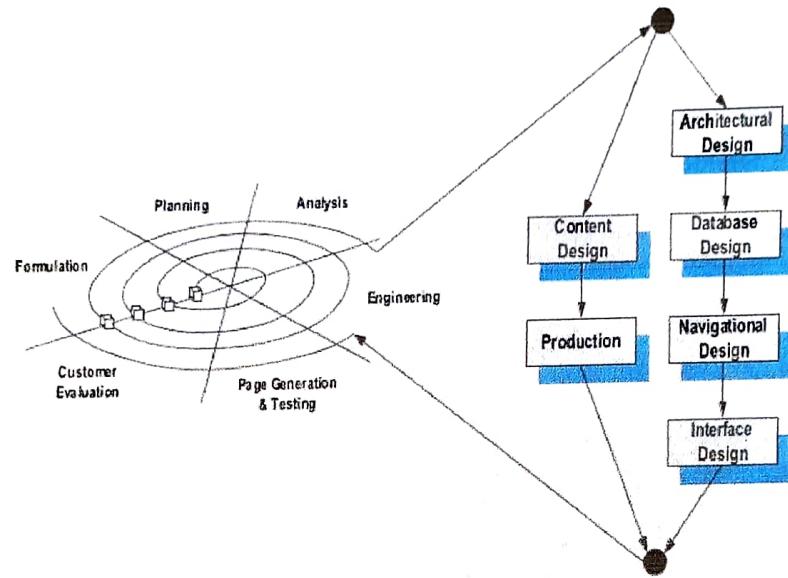
Processor	: Intel <sup>(R)</sup> Core <sup>(TM)</sup> i3-6006U Processor CPU @ 2.00 GHz
SSD	: 250 GB
RAM	: 8 GB

## Chapter 6: System Analysis And Design

### 6.1: Software Engineering Paradigm

The web application process begins with:

- Formulation, an activity that identifies the goals and objectives of the system and establishes the scope for the first increment.
- Planning estimates overall project cost, evaluates risks associated with the development effort, and defines a finely granulated development schedule for subsequent increments.
- Analysis establishes technical requirements for the system and identifies the content items that will be incorporated. Requirements for graphic design (aesthetics) are also defined.
- Engineering activity incorporates two parallel tasks, Content design and production are performed by non-technical member of the team. The intent of these tasks is to design, produce, and/or acquire all texts, graphics, audio and video contents that are to become integrated into the web application. At the same time, a technical design tasks are conducted.
- Page Generation is a construction activity that makes heavy use of automated tools. The content defined in the engineering activity is merged with the architectural, navigational and interface design to produce executable web pages.
- Testing ensures that the web application will operate correctly with different browsers.
- Customer Evaluation is the point at which changes are requested. These changes are integrated into the next path through the incremental process flow.



*Fig 6.1.1: Software Engineering Paradigm*

## 6.2: System Design

System phase concentrate on “how” it will be done. System design can be of the following two types:

- Logical Design
- Physical Design

Logical design concerns with the specifications of major features of the system that would meet the objectives. Physical design of a system take the logical design blueprints and produces the program specifications, physical file or database definition with the help of this blue print. It also includes user-interface design taking into considerations the selected hardware and software packages.

The end result of structured analysis produces a structured specification that uses several basic tools such as:

- Context Level DFD
- 1<sup>st</sup> Level DFD

- E-R Diagram
- Data Dictionary

The selection of these tables may vary from project to the system under consideration I select the tools, which I find suitable for the project.

### 6.3: Data Flow Diagram

The data flow diagram (DFD) is one of the most important tools used by the systems analysts. The use of the data flow diagrams as modelling tools was popularized by **DeMarco (1978)** **Gane and Sarson (1979)** through their structured systems analysis methodologies. They suggested that a data flow diagram should be the first tool used by the systems analyst to model systems components. These components are the systems processes, the data used by these processes, any external entities that interact with the systems, and the information flows in the system.

Following are the basic notation used to construct the DFDs.

Shape	Description
	A producer or consumer of information that resides outside the bounds of the system to be modelled
	A transformer of information (i.e. process)
	A data item or collection of data item, the arrow-head indicates the direction of flow
	The table in which information will be stored ultimately

**Table 6.3: Notation used to create Data Flow Diagram**

### 6.3.1: Types Of Data Flow Diagram

Depending on the methodology (Gane and Sarson Vs. Yourdon and Coad), DFD symbols vary slightly. However, the basic ideas remain the same.

Description	Gane & Sarson	Yourdon & Coad
Process		
Data Store		
External Entity		
Data Flow		

Table 6.3.1: Types Of DFD

### 6.4: Context Level DFD

Context Level Data Flow Diagram is the starting point of the Data Flow Diagram. They are structured to show the highest –level model of the system. This is the most general and broadest representation of the system. It does not contain any stored and it links the system to rest of the world. It is also known as a 'Bubble Chart', the process of these is to clarify the system requirements and identify the major transformations that will become the programs in the system designs. So, it is the starting point of the system design phase that functionally decomposes the requirements specifications to the lowest level.

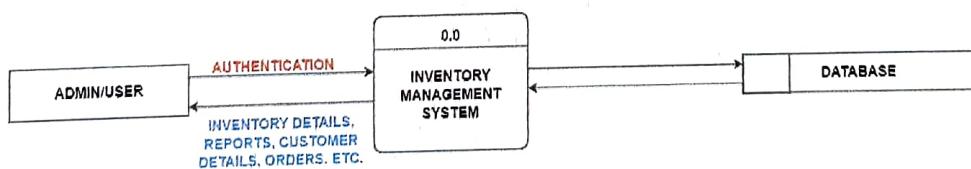


Fig 6.4.1: Context Level DFD

### 6.5: 1<sup>st</sup> Level DFD

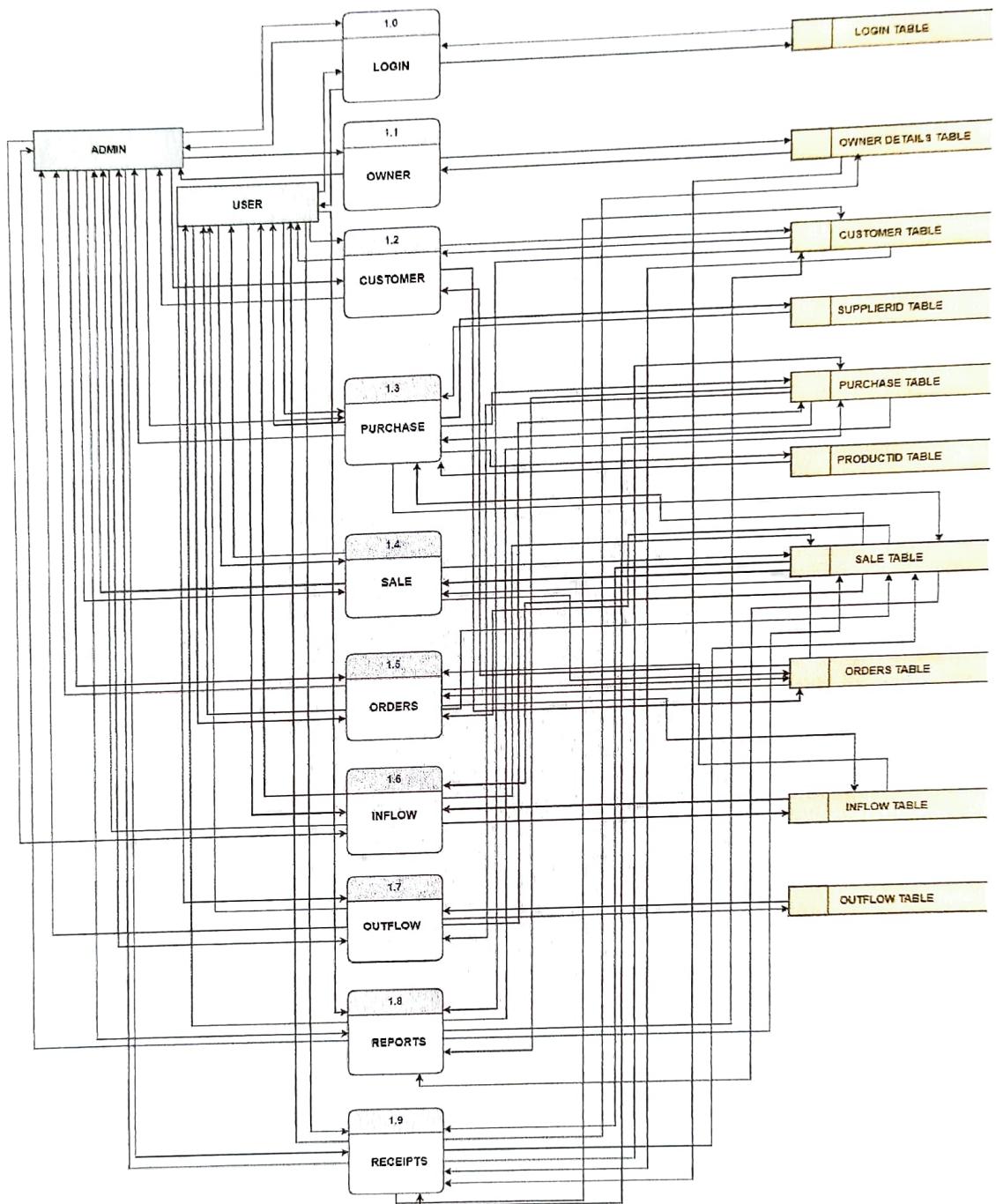


Fig. 6.5: 1<sup>st</sup> Level DFD

- Customer Module

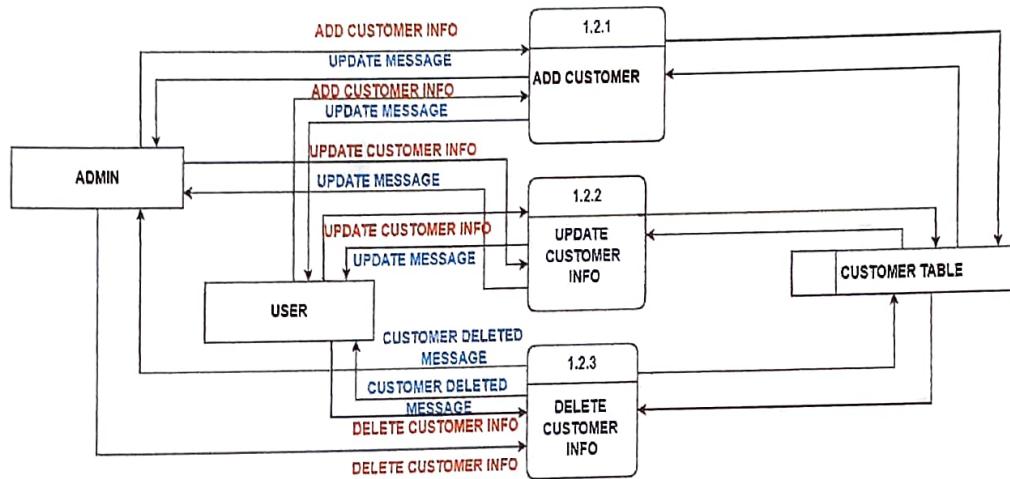


Fig 6.6.3: Customer module DFD

- Product Module

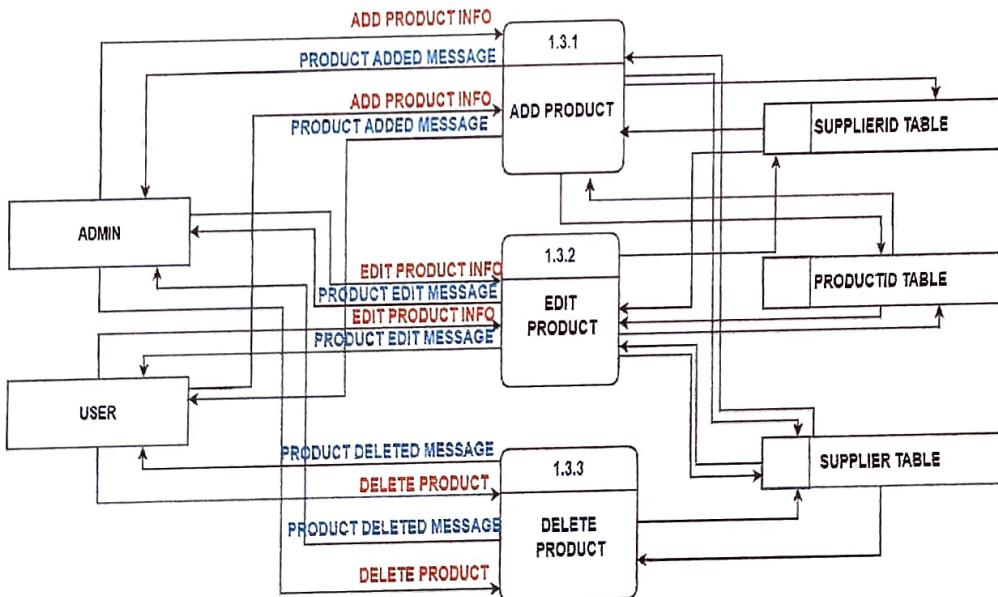


Fig 6.6.4: Product module DFD

- Sale Module

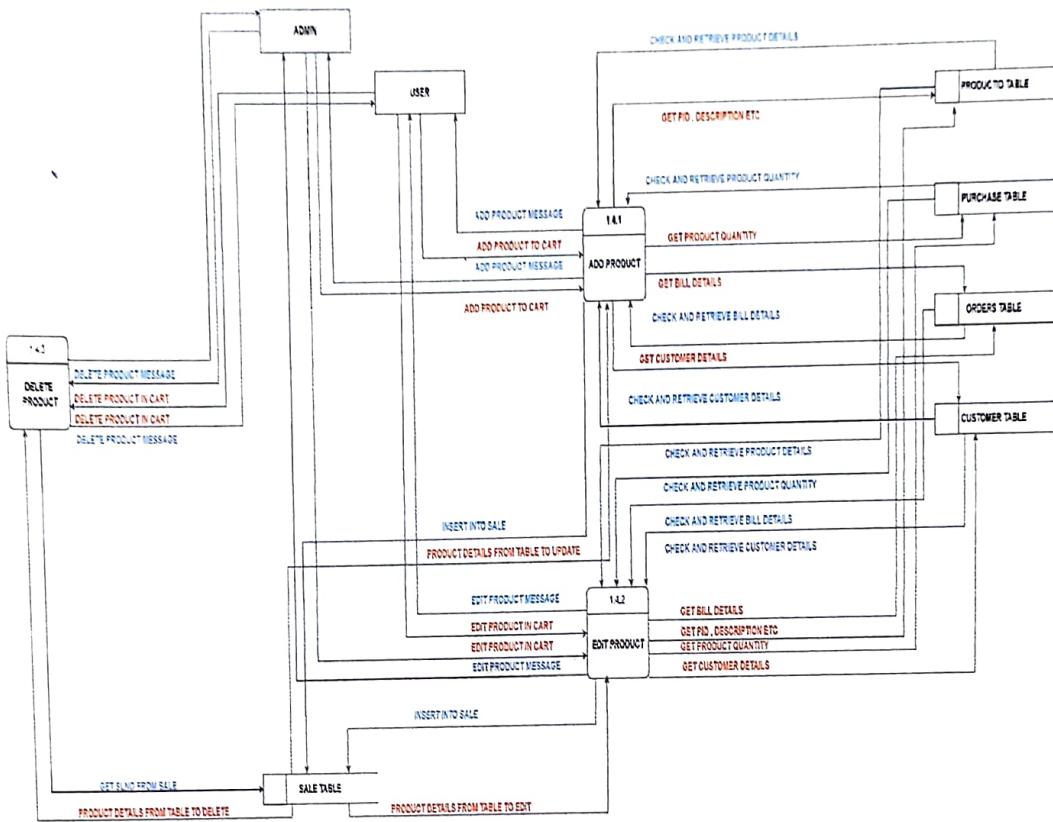


Fig 6.6.5: Sale module DFD

- Order Module

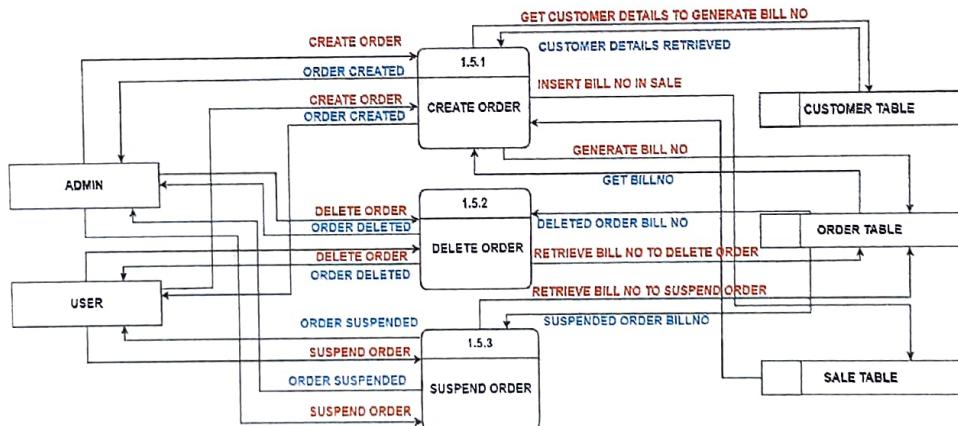


Fig 6.6.6: Order module DFD

- **Inflow Module**

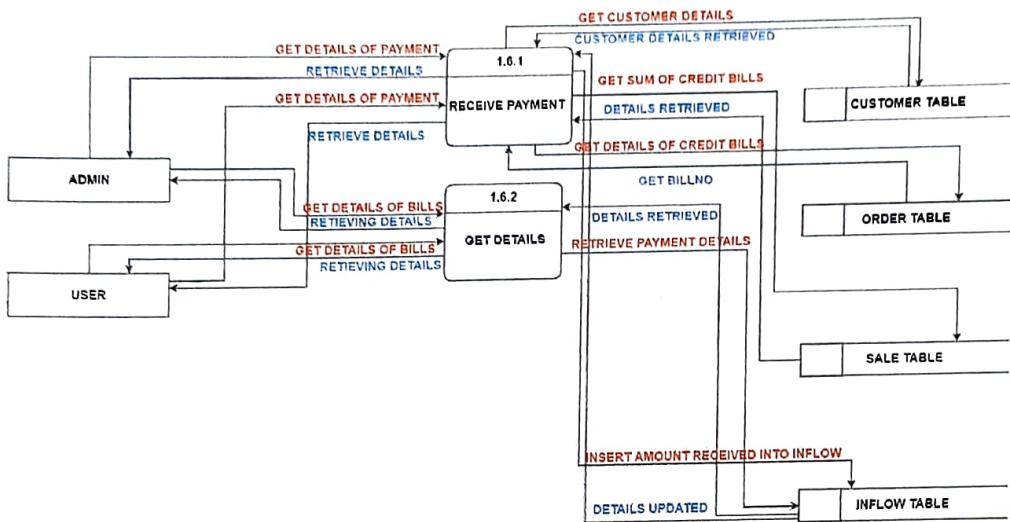


Fig 6.6.7: Inflow module DFD

- **Outflow Module**

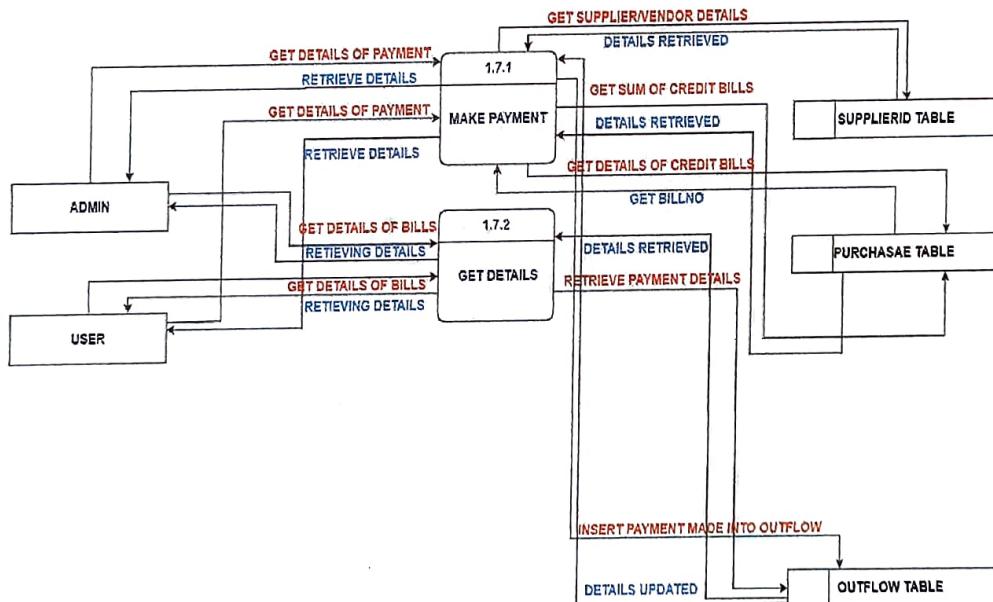
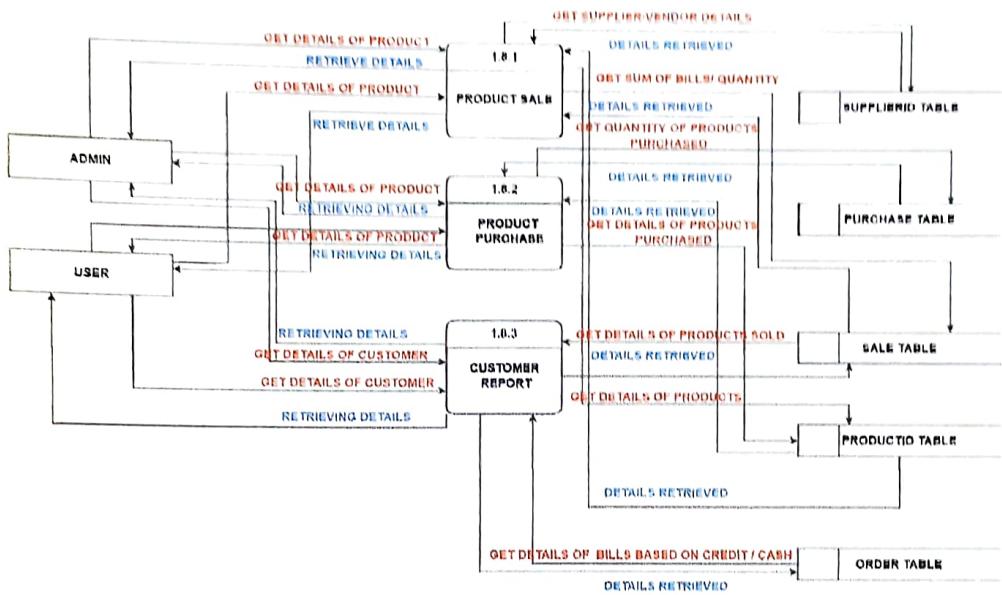


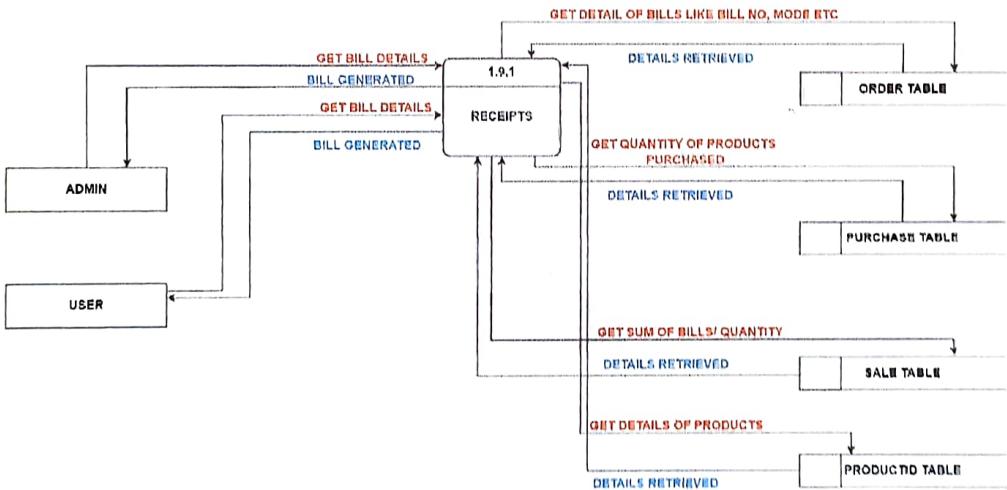
Fig 6.6.8: Outflow module DFD

- **Reports Module**



*Fig 6.6.9: Reports module DFD*

- **Receipts Module**



*Fig 6.6.10: Receipts module DFD*

## 6.7 DATA DICTIONARY

Table 6.7.1: backup\_time

Field name	Datatype	Size	Default	Constraint	Description
serial	TinyInt	1	Auto-Increment	Primary Key	Unique Identification For Backup
backup_time	time		Current Timestamp	Not null	Time When Backup Is Created
backup_date	date		Current Timestamp	Not null	Date When Backup Is Created

Table 6.7.2: customer

Field Name	Data type	Size	Default	Constraint	Description
cus_id	int	11	Auto-Increment	Primary Key	Unique Identification of the customer
add_date	date		Current Timestamp	Not null	When customer was added to database
customer_name	Varchar	20	None	Not null	Name of customer
customer_address	Varchar	20	None	Not null	Address of customer
mobile	Varchar	10	None	Not null	Mobile/Phone Number of customer
del_flag	TinyInt	1	None	Not null	Checks if customer is deleted

Table 6.7.3: inflow

Field name	Datatype	Size	Default	Constraint	Description
in_tran_id	Int	11	Auto Increment	Primary Key	Unique Identification For The Transaction
tran_details	Varchar	15	None	Not Null	Transaction Details
cus_id	Int	11	None	Not Null	Customer Identification

<b>receive_date</b>	Date		Current Timestamp	Not Null	Payment Receipt Date
<b>receive_time</b>	Time		Current Timestamp	Not Null	Payment Receipt Time
<b>mode</b>	Varchar	11	None	Not Null	Mode of Payment (Debit Card, Credit Card, UPI)
<b>amt</b>	Decimal	12,2	None	Not Null	Amount Received

**Table 6.7.4: outflow**

Field name	Datatype	Size	Default	Constraint	Description
<b>out_tran_id</b>	Int	11	Auto Increment	Primary Key	Unique Identification For The Transaction
<b>tran_details</b>	Varchar	15	None	Not Null	Transaction Details
<b>sid</b>	Int	11	None	Not Null	Supplier Identification
<b>payment_date</b>	Date		Current Timestamp	Not Null	Payment Date
<b>payment_time</b>	Time		Current Timestamp	Not Null	Payment Time
<b>mode</b>	Varchar	11	None	Not Null	Mode of Payment (Debit Card, Credit Card, UPI)
<b>amt</b>	Decimal	12,2	None	Not Null	Amount Paid

**Table 6.7.5: login**

Field name	Datatype	Size	Default	Constraint	Description
<b>user_id</b>	Int	11	Auto Increment	Primary Key	User Id Of User
<b>username</b>	Varchar	20	None	Not Null	User Name Of User
<b>password</b>	Varchar	20	None	Not Null	Password Of User
<b>dobfor</b>	Date		None	Not Null	Date Of Birth Of User
<b>power</b>	TinyInt	1	None	Not Null	Whether Admin Or User
<b>user</b>	Varchar	20	None	Not Null	Name Of User
<b>phone</b>	Varchar	10	None	Not Null	Phone Of User
<b>address</b>	Varchar	20	None	Not Null	Address Of User
<b>email</b>	Varchar	50	None	Not Null	Email Of User

<b>del_flag</b>	TinyInt	1	None	Not Null	If User Is Deleted
<b>question</b>	Varchar	50	None	Not Null	Security Question Of User
<b>answer</b>	Varchar	50	None	Not Null	Answer To Security Question
<b>otp</b>	Char	4	None	Not Null	One Time Password
<b>time_otp</b>	Timestamp		Current Timestamp	Not Null	Time To Delete OTP

Table 6.7.6: orders

Field name	Datatype	Size	Default	Constraint	Description
<b>bill_no</b>	Int	11	Auto Increment	Primary key	Bill Number
<b>cus_id</b>	Int	11	None	Not Null	Customer Identification
<b>addl_disc</b>	Decimal	12,2	None	Not Null	Additional Discount Provided After Billing
<b>bill_date</b>	Date		None	Not Null	Date Of Billing
<b>bill_time</b>	Time		None	Not Null	Time Of Billing
<b>payment</b>	Varchar	6	None	Not Null	Mode Of Bill
<b>suspend</b>	TinyInt	1	None	Not Null	If State Of Bill Is Suspended
<b>total_amount</b>	Int	11	None	Not Null	Total Amount Of Bill

Table 6.7.7: owner\_info

Field name	Datatype	Size	Default	Constraint	Description
<b>serial</b>	Int	11	Auto Increment	Primary Key	Serial Number of Owner
<b>dl_number</b>	Varchar	20	None	Not Null	Licence Number Of Business
<b>gstin</b>	Varchar	15	None	Not Null	GSTIN Of Business
<b>owner_address</b>	Varchar	20	None	Not Null	Address Of Owner
<b>owner_name</b>	Varchar	20	None	Not Null	Name of Owner
<b>owner_phone_number</b>	Varchar	10	None	Not Null	Phone Number of Owner
<b>UPI_phone</b>	Varchar	10	None	Not Null	Owner UPI Phone Number
<b>vpa</b>	Varchar	50	None	Not Null	Virtual Payment Address Of Owner

**Table 6.7.8: productid**

Field name	Datatype	Size	Default	Constraint	Description
pid	Int	11	Auto Increment	Primary Key	Unique Identification For Products
description	Varchar	20	None	Not Null	Description Of Product
manufacturer	Varchar	20	None	Not Null	Manufacturer Of Product
unit	Varchar	10	None	Not Null	Unit Of Per Product
mrp	Decimal	12,2	None	Not Null	MRP Of Product
actual_rate	Decimal	12,2	None	Not Null	Actual Price At Which Product Was Bought
expiry_date	Date		None	Not Null	Expiry Date Of Product
batch	Varchar	10	None	Not Null	Batch Number Of Product
cgst	Decimal	12,2	None	Not Null	CGST Levied On Product
sgst	Decimal	12,2	None	Not Null	SGST Levied On Product
rate	Decimal	12,2	None	Not Null	Estimated Selling Price
hsn	Varchar	10	None	Not Null	HSN of Product

**Table 6.7.9: sale**

Field name	Datatype	Size	Default	Constraint	Description
sale_id	Int	11	Auto Increment	Primary key	Unique Id Of Sale
pid	Int	11	None	Not Null	Product Id Number
bill_no	Int	11	None	Not Null	Bill Number
quantity	Int	11	None	Not Null	Quantity Sold
free_qty	Int	11	None	Not Null	Free Quantity Offered
rate	Decimal	12,2	None	Not Null	Product Was Sold At
gross_total	Decimal	12,2	None	Not Null	Total Before Discount
total	Decimal	12,2	None	Not Null	Total After Tax And Discount
discount	Decimal	12,2	None	Not Null	Discount On Products i.e.(Mrp-Rate)
tax	Decimal	12,2	None	Not Null	Total Tax Levied

**Table 6.7.10: purchase**

Field name	Datatype	Size	Default	Constraint	Description
pid	Int	11	None	Not Null	Product Identification Number
PurchasePayment	Varchar	6	None	Not Null	Whether bought on credit/cash
sid	Int	11	None	Not Null	Id of supplier
lock_tab	TinyInt	1	None	Not Null	If all items of table are locked to make them uneditable
sl_no	Int	11	Primary Key	Auto Increment	Unique identification of row
date_of_insert	Date		None	Not Null	When products were bought
del_flag	TinyInt	1	None	Not Null	If product is deleted
free_qty_sup	Int	11	None	Not Null	Free quantity of product received
invoice_supp	Varchar	10	None	Not Null	Invoice number of product
quantity	Int	11	None	Not Null	Quantity of product purchased

**Table 6.7.11: supplierid**

Field name	Datatype	Size	Default	Constraint	Description
sid	Int	11	Auto Increment	Primary Key	Unique Identification Of Supplier
supplier	Varchar	20	None	Not Null	Supplier Name
mobile	Varchar	10	None	Not Null	Mobile/Phone Number Of Supplier
address	Varchar	20	None	Not Null	Address Of Supplier
bank	Varchar	20	None	Not Null	Bank Details Of Supplier
add_date	Date		None	Not Null	Date When Supplier Was Added
del_flag	TinyInt	1	None	Not Null	If Supplier Is Deleted

## 6.8: ER Diagram

Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases. An Entity Relationship Diagram is a piece of data—an object or concept about which data is stored.

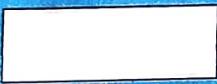
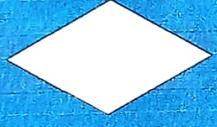
Shape	Description
	Entity
	Relationship
	Attributes

Table 6.8.1: Notations used for ER Diagram

There are three types of relationships between entities:

- **One-to-one (1 to 1):** one instance of an entity (A) is associated with one other instance of another entity (B). For example, in a database of employees, each employee name (A) is associated with only one social security number (B).
- **One-to-many (1 to N):** one instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A. For example, for a company with all employees working in one building, the building name (A) is associated with many different employees (B), but those employees all share the same singular association with entity A.

- **Many-to-many (N to N):** one instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A. For example, for a company in which all of its employees work on multiple projects, each instance of an employee (A) is associated with many instances of a project (B), and at the same time, each instance of a project (B) has multiple employees (A) associated with it

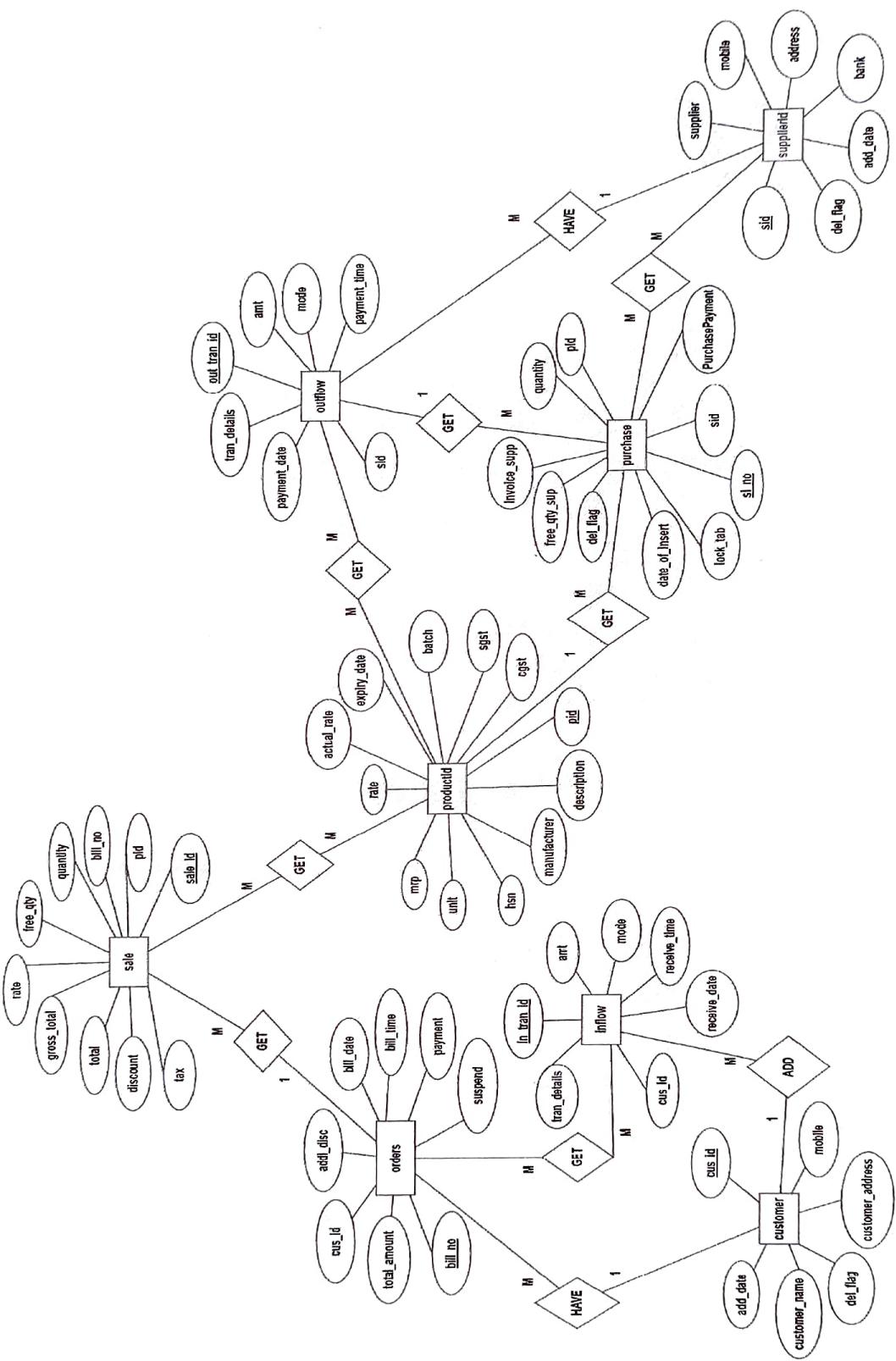


Fig 6.8.1: ER Diagram

## Chapter 7: Input And Output Screen

### 7.1: Login Screen

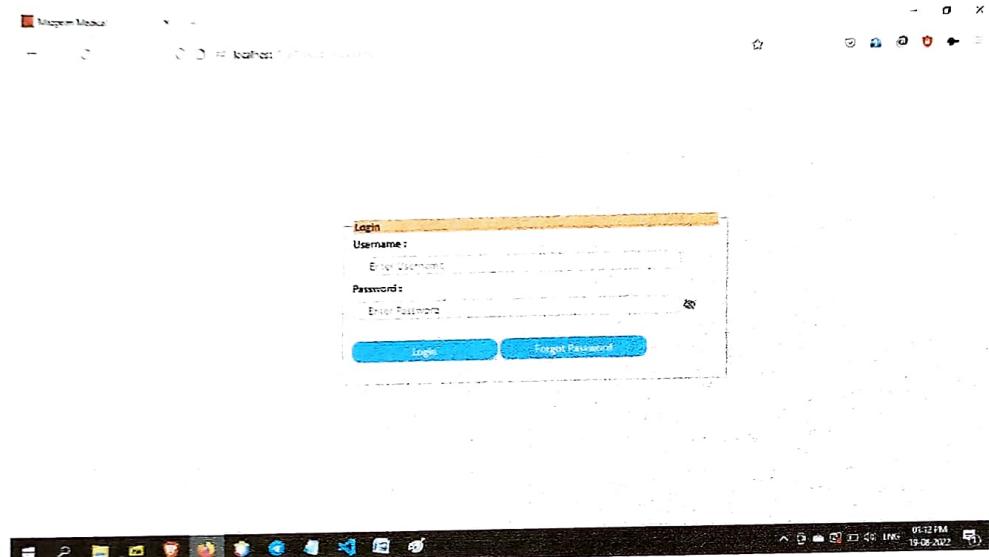


Fig 7.1: Login Screen

#### 7.1.1: Forgot Password Popup

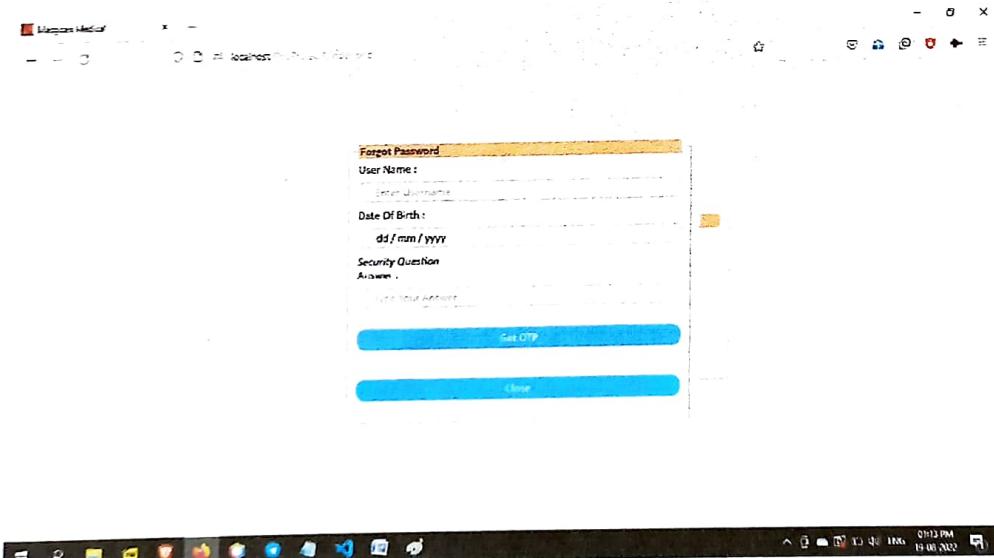


Fig 7.1.1: Forgot Password Popup

## 7.2: Mazgaon Medical

### 7.2.1: Owner Info Update Screen

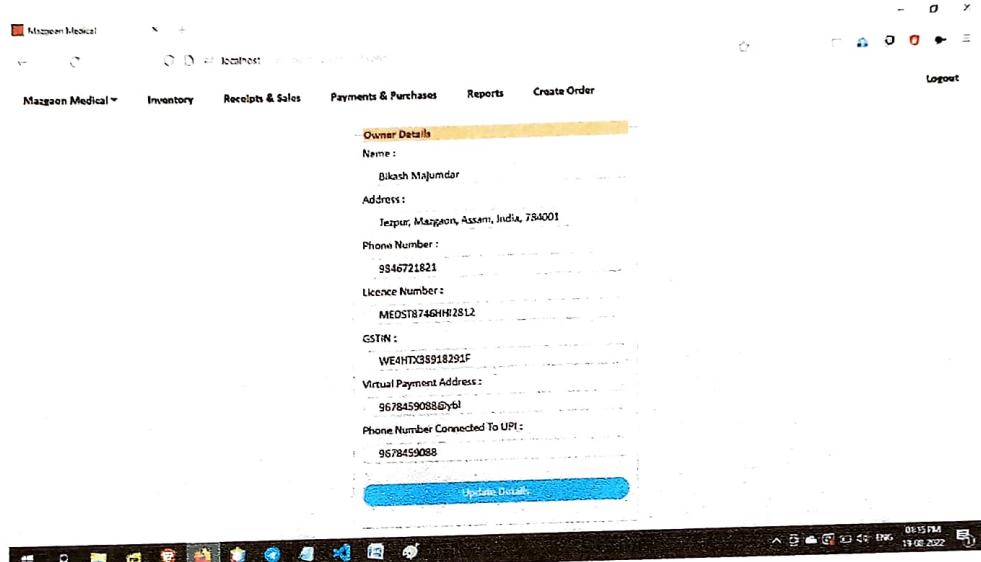


Fig 7.2.1: Owner Info Update Screen

### 7.2.2: User Management Screen

The screenshot shows a Windows application window titled "Mazgaon Medical". The menu bar includes "Inventory", "Receipts & Sales", "Payments & Purchases", "Reports", and "Create Order". A "Logout" link is in the top right. The main content area is titled "User Info" and displays a table of user data:

S No.	Privilege	Date Of Birth	Name	Phone	Address	Email	Update Info	Update Password	Delete
1	Admin	1991-03-25	BIKASH MAJUMDAR	7988734221	TEZPUR	bm19171@gmail.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	User	2000-12-25	MDURE	644751563	TEZPUR	myself@rediffmail.co	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	User	2001-03-24	KAME	70617261	ATEZ	kame@gmail.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	User	2002-07-17	PHONE	7111241562	TEZPUR	phone@gmail.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	User	1997-02-06	ITU DOKA	2812817281	SEGUANU	dokadis@gmail.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

A blue "Create User" button is at the bottom.

Fig 7.2.2.: User Management Screen

### 7.2.2.1: Update User Info Popup

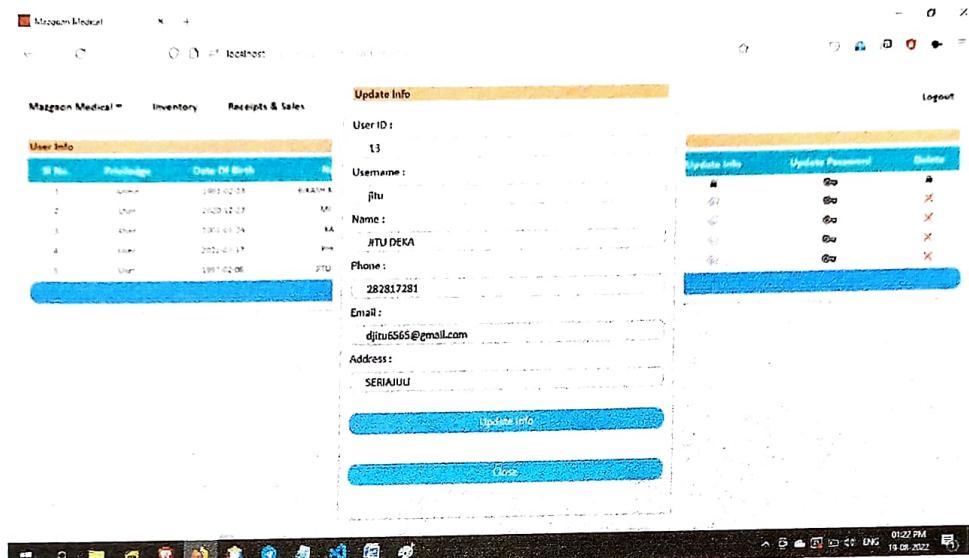


Fig 7.2.2.1: Update User Info Popup

### 7.2.2.2: Update Password Popup

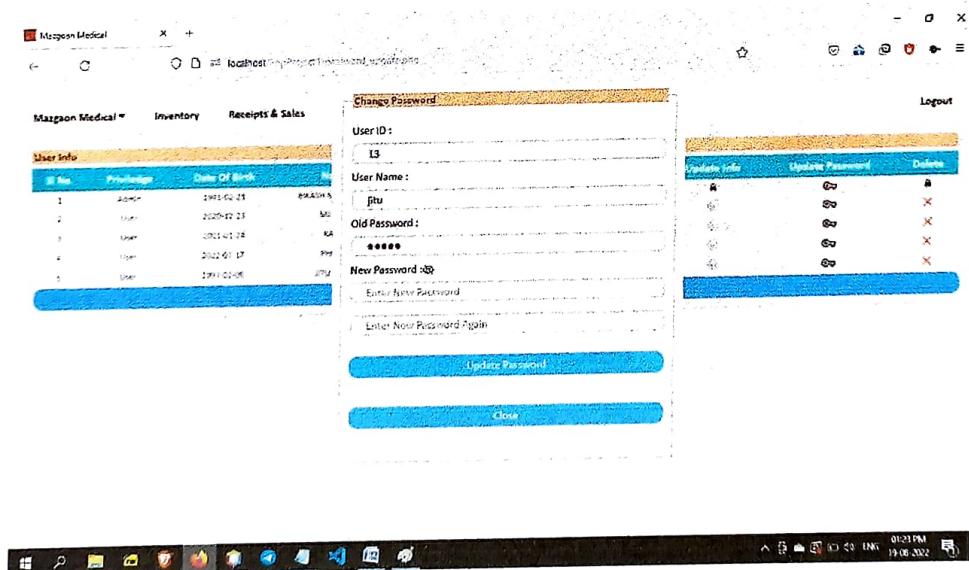


Fig 7.2.2.2: Update Password Popup

### 7.2.2.3: Delete User Popup

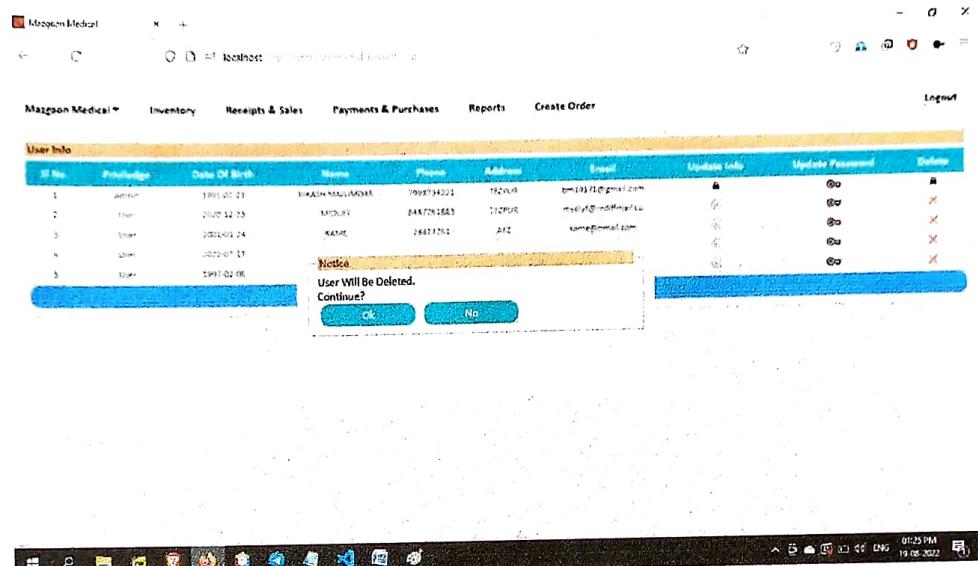


Fig 7.2.2.3: Delete User Popup

### 7.2.3: Backup & Restore Screen

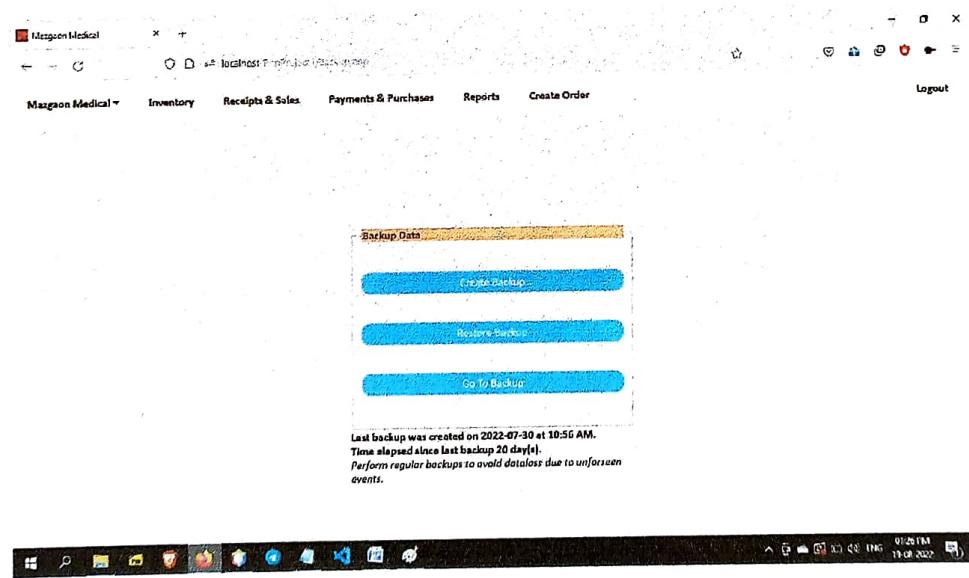


Fig 7.2.3: Backup & Restore Screen

## 7.3: Inventory

### 7.3.1: Inventory Management

The screenshot shows the 'Inventory Management' section of the Mazgaon Medical software. At the top, there are tabs for 'Enter Details', 'Supplier', 'Manufacturer', 'Unit', 'Batch No.', 'Free', 'Rate', 'CGST (%)', 'SGST (%)', 'Actual Rate', 'Purchase Type', and buttons for 'Add', 'Clear', 'Delete Details', 'Credit Purchase', and 'Print Report'. Below this is a table titled 'Stock Details' showing 16 entries. The columns include: Description, Supplier, Date In, Expiry, Batch, Qty, Free, Actual Rate, MRP, Sale Rate, Cost, and Net. The table lists various items like MAMMY POKO PANTS, BAS, C, B, and A, with details such as quantity (e.g., 20, 100), expiry dates (e.g., 2023-09-07, 2023-09-31), and rates (e.g., 17.00, 1.00). A search bar at the bottom allows users to search by item name.

Description	Supplier	Date In	Expiry	Batch	Qty	Free	Actual Rate	MRP	Sale Rate	Cost	Net
MAMMY POKO PANTS	HEINZ	2023-07-06	2023-09-07	ERT5	20	2	17.00	25.98	20.00	1.00	1.00
BAS	A	2023-07-30	2023-09-31	101	3	100	12.43	12.30	11.00	6.00	6.00
BAS	B	2023-07-30	2023-09-31	BA52	200	1	25.87	28.00	27.40	2.00	2.00
BAS	B	2023-07-30	2023-09-31	DA52	34	1	25.87	28.00	27.40	2.00	2.00
BAS	C	2023-07-30	2023-09-31	A	21	2	10.00	12.43	12.30	12.00	6.00
BAS	C	2023-07-30	2023-09-31	C	20	1	1.00	2.00	1.00	5.00	6.00
BAS	B	2023-07-30	2023-09-31	B	110	1	15.00	23.00	20.00	2.00	3.00
BAS	A	2023-07-30	2023-09-31	A	100	10	10.00	12.43	12.30	12.00	6.00
BAS	A	2023-07-30	2023-09-31	A	222	13	10.00	12.43	12.30	12.00	6.00

Fig 7.3.1: Inventory Management

### 7.3.2: Inventory Details

The screenshot shows the 'Inventory Details' section of the Mazgaon Medical software. At the top, there are tabs for 'Enter Details', 'Supplier', 'Manufacturer', 'Unit', 'Batch No.', 'Free', 'Rate', 'CGST (%)', 'SGST (%)', 'Actual Rate', 'Purchase Type', and buttons for 'Add', 'Clear', 'Delete Details', 'Credit Purchase', and 'Print Report'. Below this is a table titled 'Stock Details' showing 23 entries. The columns include: Description, Manufacturer, Hsn, Unit, Expiry, Batch, Qty, Sold, In Stock, Free, Gold, In Stock, Actual Rate, MRP, Rate, Cost, and Net. The table lists various items like MAMMY POKO PANTS, BAS, C, B, and A, with details such as quantity (e.g., 20, 100), expiry dates (e.g., 2023-09-07, 2023-09-31), and rates (e.g., 17.00, 1.00). A search bar at the bottom allows users to search by item name.

Description	Manufacturer	Hsn	Unit	Expiry	Batch	Qty	Sold	In Stock	Free	Gold	In Stock	Actual Rate	MRP	Rate	Cost	Net
MAMMY POKO PANTS	HEINZ	8004	20	2023-09-07	ERT5	20	5	35	2	0	2	17.00	25.98	20.00	1.00	1.00
BAS	B	10X1	10X1	2023-09-31	BA52	334	0	334	2	0	2	25.87	28.00	27.40	2.00	2.00
C	C	10X1	10X1	2023-09-31	C	20	0	20	1	0	1	1.00	2.00	1.00	6.00	6.00
B	B	SD10	SD10	2023-09-31	B	110	0	110	1	0	1	15.00	23.00	20.00	2.00	2.00
A	A	10X1	10X1	2023-09-31	A	222	13	210	15	0	13	10.00	12.43	12.30	12.00	6.00

Fig 7.3.2: Inventory Details

## 7.4: Receipts & Sales

### 7.4.1: Receive Payment Screen

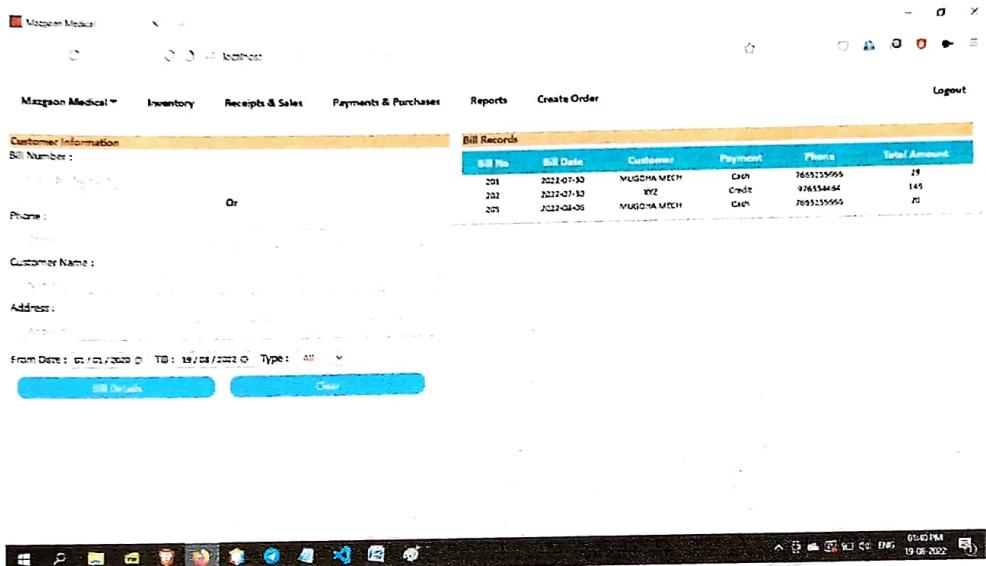


Fig 7.4.1: Receive Payment Screen

#### 7.4.1.1: Receive Payment Popup

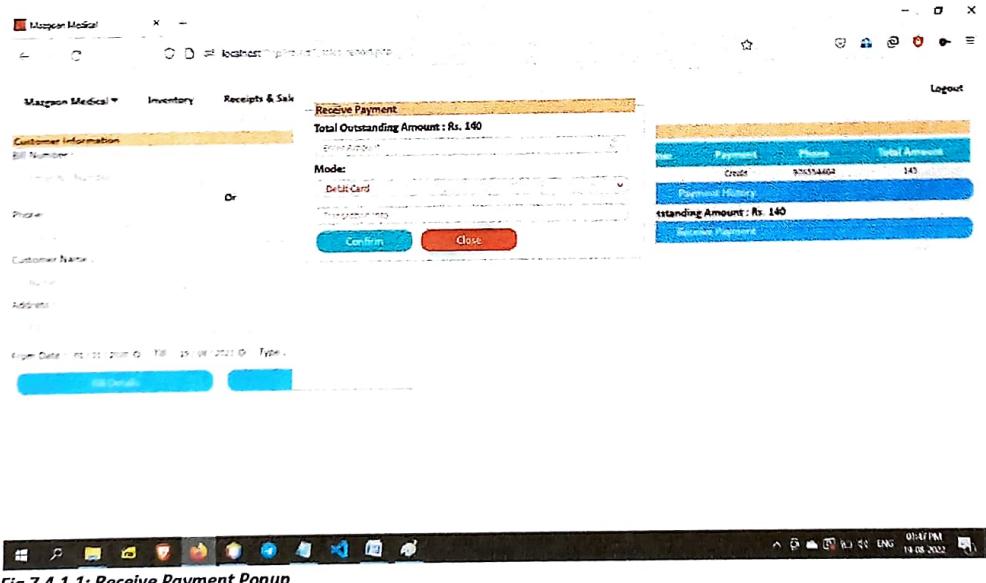


Fig 7.4.1.1: Receive Payment Popup

#### 7.4.1.2: Receive Payment History Popup

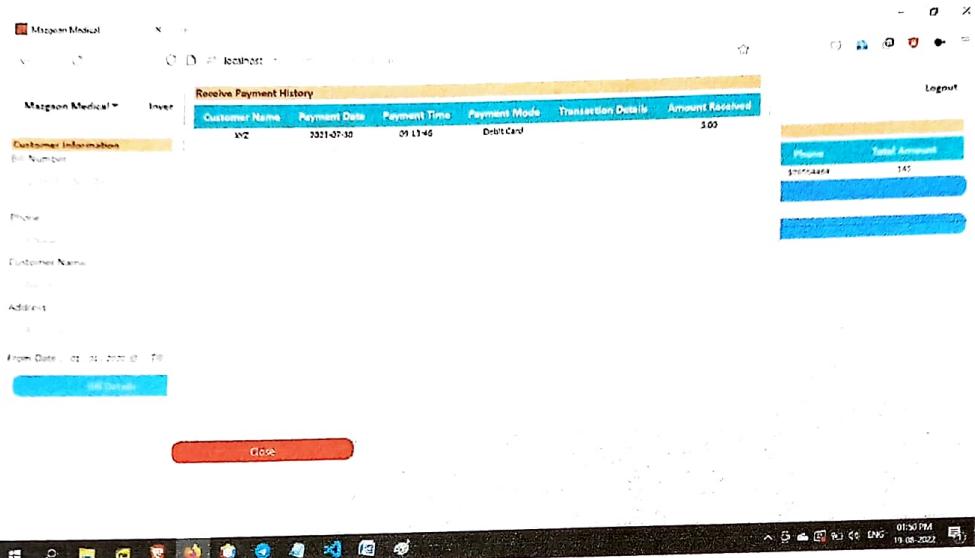


Fig 7.4.1.2: Receive Payment History Popup

#### 7.4.2: Bill Details

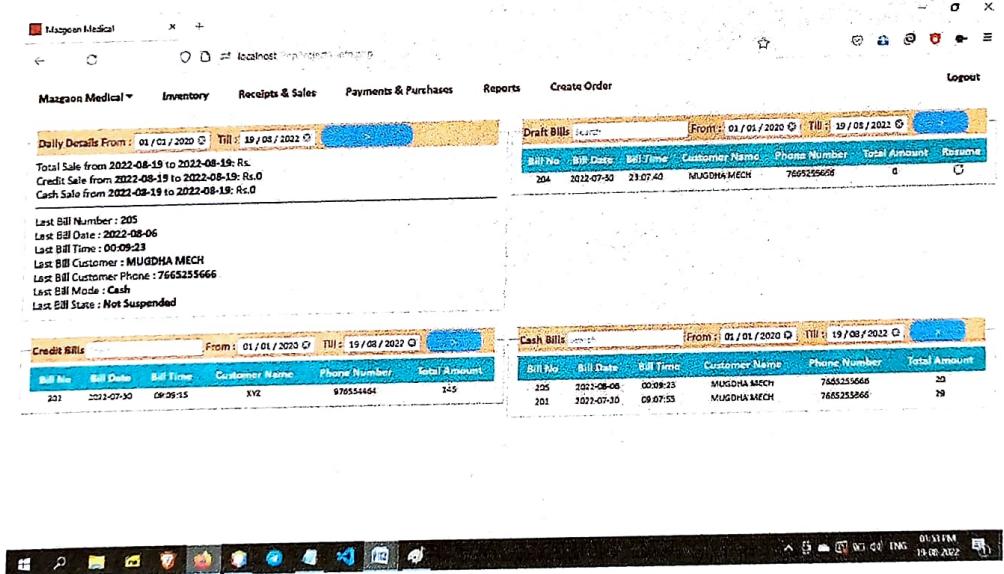


Fig 7.4.2: Bill Details

### 7.4.3: Customer Details

Customer ID	Date	Customer Name	Mobile	Customer Address	Edit	Delete
9	2012-07-03	MUGOCHA MACH	7665755665	TEZPUR	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	2012-07-04	AJED BOZA	613173675	KALPANA	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	2012-07-07	BETAL	7885624412	JORHAT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	2012-07-08	MEERA MOTTI	9441231448	TEZPUR	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	2012-07-08	ZY	9765564664	TEZPUR	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	2012-07-09				<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fig 7.4.3: Customer Details

### 7.5: Payments & Purchases

#### 7.5.1: Make Payment Screen

Invoice No	Date	Supplier	Mode	Phone	Taxable Amount	Tax	Amount After Tax
1	2012-07-30	ASDIT	Credit	67855678	2670.00	154.20	2824.20
2	2012-07-30	WOROI	Cash	6755545	1089.58	42.79	1122.37
3	2012-07-31	WOROI	Credit	6755545	6592.00	228.76	6825.76

Fig 7.5.1: Make Payment Screen

### 7.5.1.1: Make Payment Popup

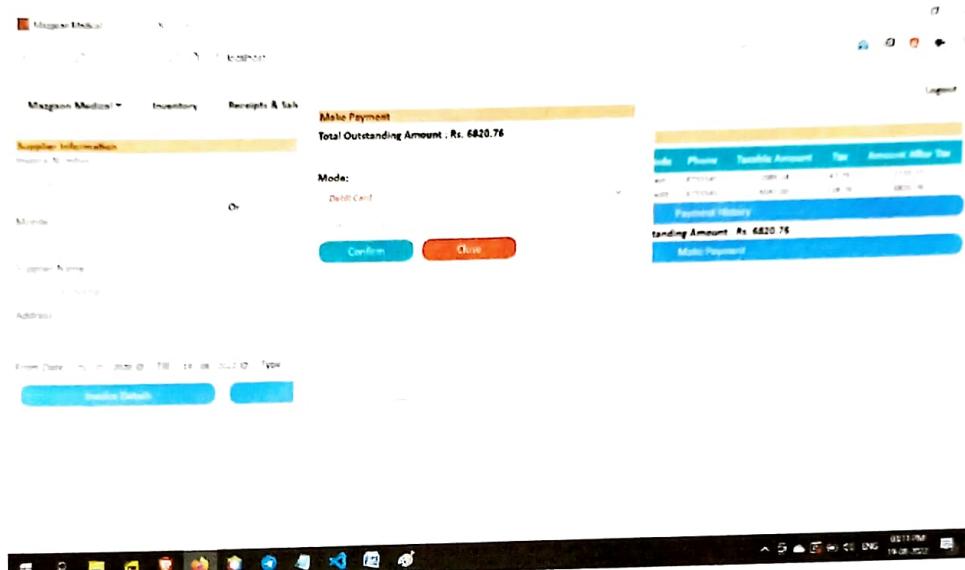


Fig 7.5.1.1: Make Payment Popup

### 7.5.1.2: Make Payment History Popup

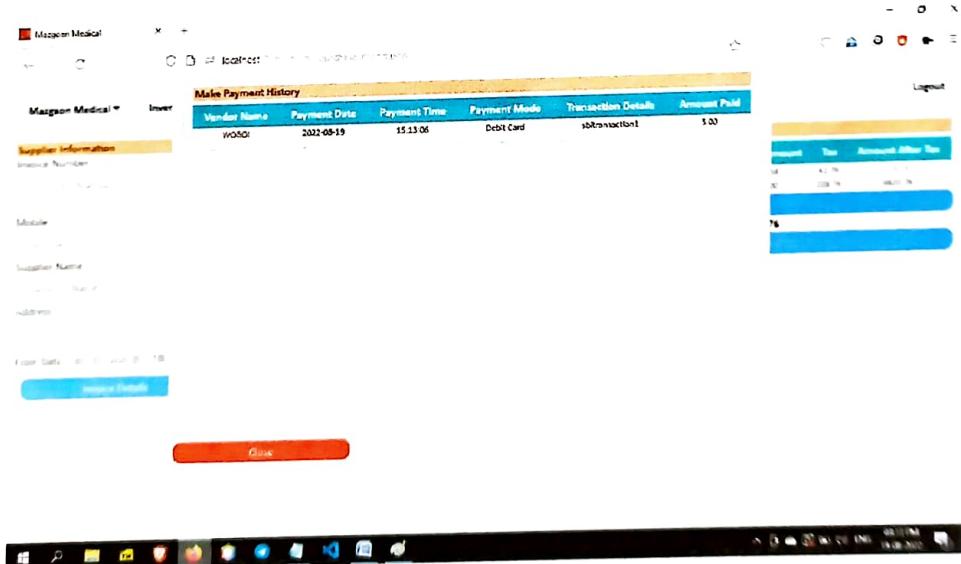


Fig 7.5.1.2: Make Payment History Popup

### 7.4.2: Supplier Details

The screenshot shows the Mazgoan Medical software interface. At the top, there is a navigation bar with links for Inventory, Receipts & Sales, Payments & Purchases, Reports, and Create Order. On the right side of the top bar is a Logout link. Below the navigation bar, there are two main sections: 'Supplier Details' on the left and 'Supplier Records' on the right.

**Supplier Details:**

- Mobile: 9876543210
- Supplier Name: ABC Ltd
- Supplier Address: 123 Main Street, New York, NY 10001
- Supplier Bank Details: Standard Chartered Bank, New York, NY 10001

**Supplier Records:**

Supplier ID	Date	Supplier Name	Mobile	Address	Bank Details	Active	Deleted
33	2022-07-30	ABC LTD	6789012345	123 Main St, New York, NY 10001	Standard Chartered Bank, New York, NY 10001	✓	✗
33	2022-07-30	WORLD	67535454	123 Main St, New York, NY 10001	Standard Chartered Bank, New York, NY 10001	✓	✗
36	2022-08-10	AEOL	7892743651	123 Main St, New York, NY 10001	Standard Chartered Bank, New York, NY 10001	✓	✗

At the bottom of the screen, there are two buttons: 'Add Supplier' and 'Cancel'.

Fig 7.4.2: Supplier Details

## 7.6: Reports

### 7.6.1: Customer Report

The screenshot shows the Mazgoan Medical software interface. At the top, there is a navigation bar with links for Inventory, Receipts & Sales, Payments & Purchases, Reports, and Create Order. On the right side of the top bar is a Logout link. Below the navigation bar, there is a search bar with the placeholder text 'Customer Report From Date : 01/01/2020 To : 22/08/2022 Sort By : Cash Chart Type : Bar'. There are also buttons for 'Print', 'Credit', and 'Debit'.

The main area displays a chart titled 'Customer Report' comparing Credit and Debit amounts over time. The chart has two bars: a green bar for Credit and a red bar for Debit. The Y-axis represents the amount, and the X-axis represents dates from January 2020 to August 2022.

At the bottom of the screen, there is a toolbar with various icons and a system tray showing the date and time as 20.08.2022 08:15 PM.

Fig 7.6.1: Customer Report

### 7.6.2: Product Purchase Report

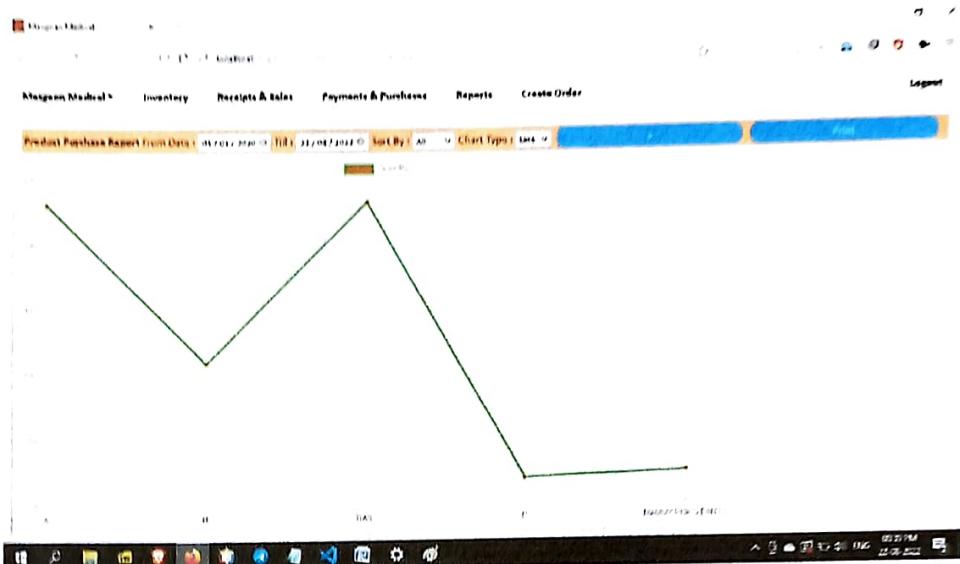


Fig 7.6.2: Product Purchase Report

### 7.6.3: Product Sale Report

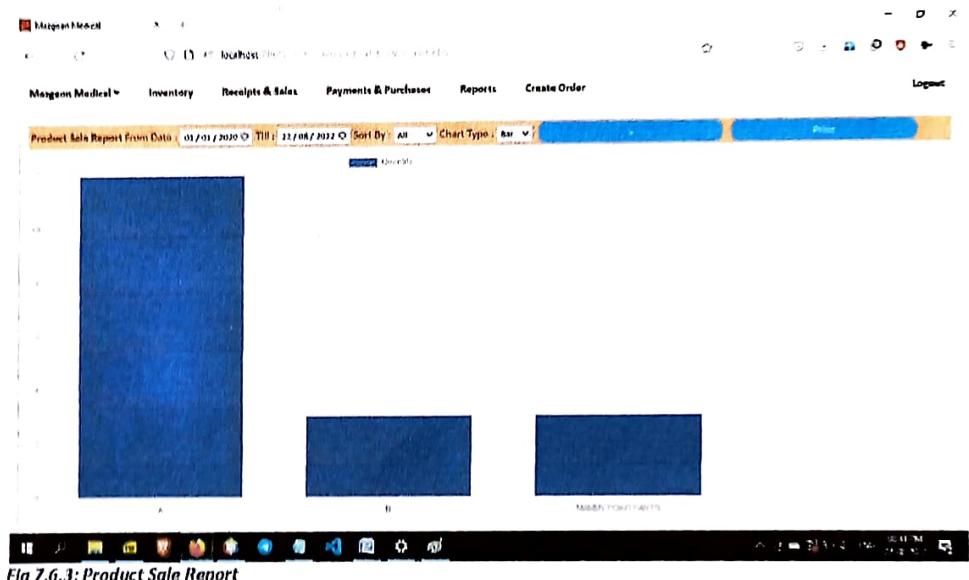


Fig 7.6.3: Product Sale Report

## 7.7: Create Order

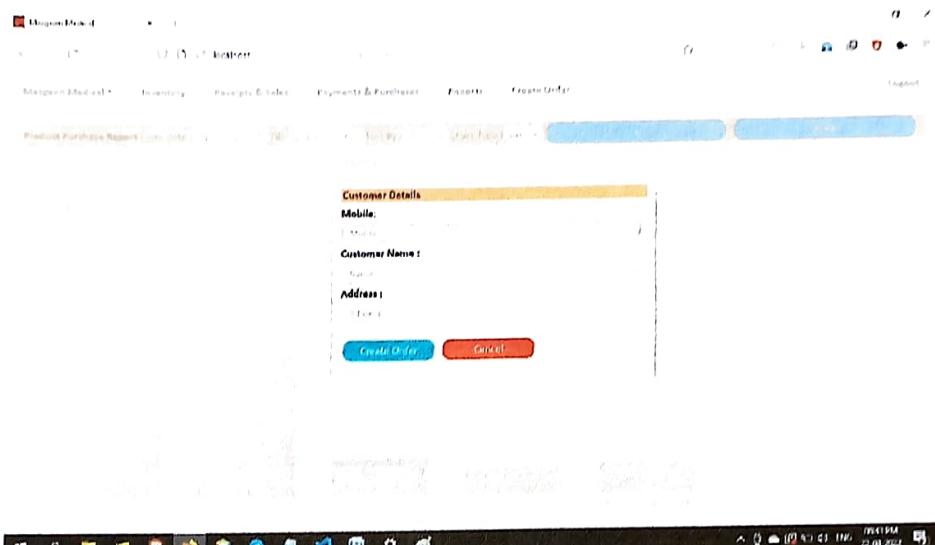


Fig 7.7: Create Order

## 7.8 Generate Receipt

### 7.8.1: Adding Products

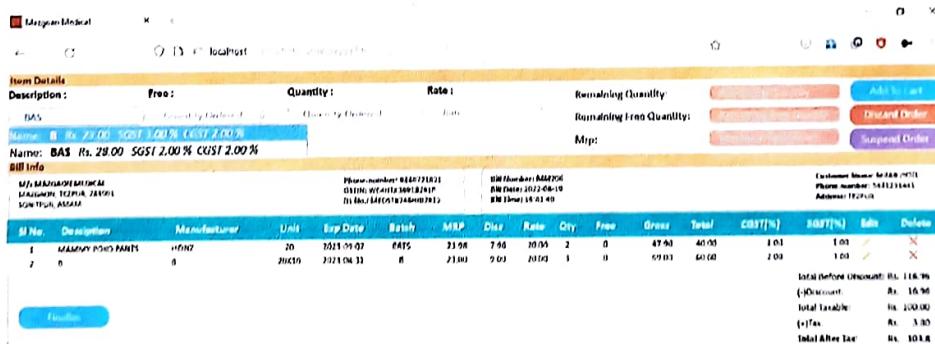


Fig 7.8.1: Adding Products

## 7.8.2: Finalize Order Popup

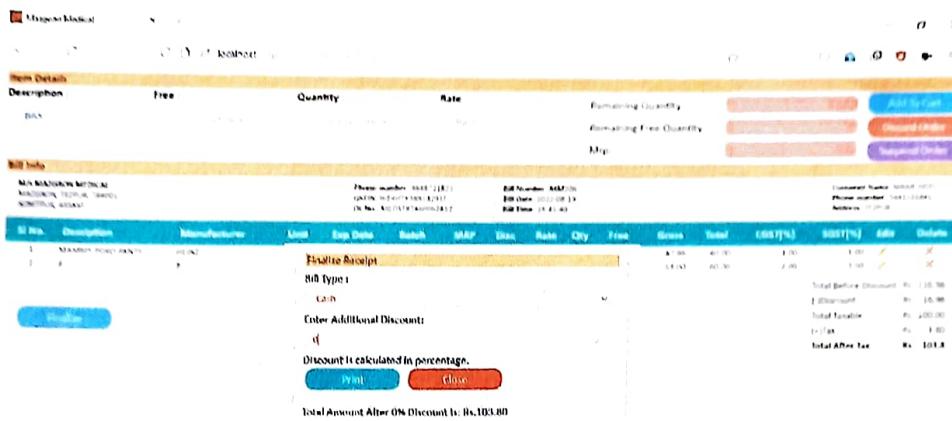


Fig 7.8.2: Finalize Order Popup

## 7.8.3: Receipt Print

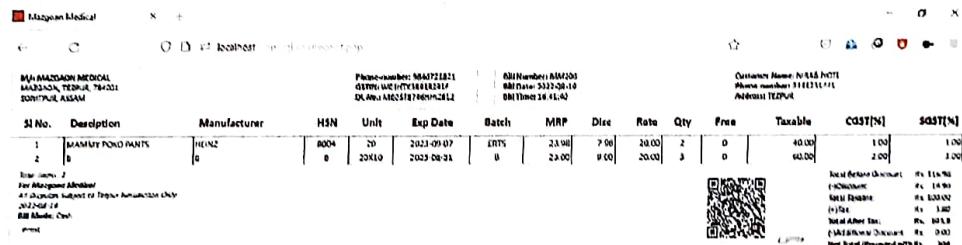


Fig 7.8.3: Receipt Print

### 7.9: Logout

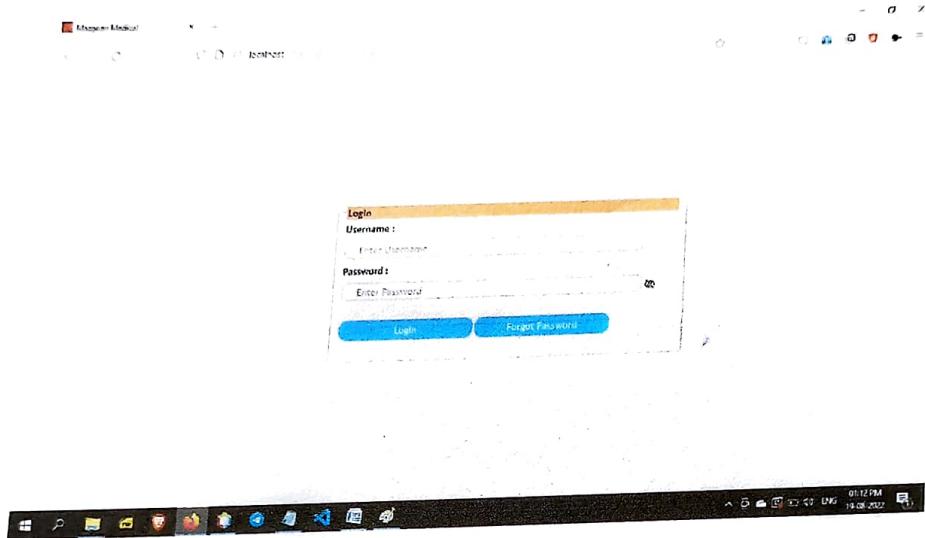


Fig 7.9: Logout

## Chapter 8: Testing

### 8.1: Introduction

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

### 8.2: Objectives of Testing

The objectives of testing are:

- Testing is a process of executing a program with the intent of finding errors.
- A successful test case is one that uncovers an as- yet-undiscovered error.

System testing is a stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences. As stated before, testing is vital to the success of a system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. A series of tests are performed before the system is ready for the user acceptance test.

### **8.3: Testing Methods**

System testing is the stage of implementation. This is to check whether the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: on line response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

The Testing Steps are:

#### **8.3.1: Unit Testing**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

#### **8.3.2: Integration Testing**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

#### **8.3.3: Validation**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

- o The function or performance characteristics confirm to specification and are accepted.

- A deviation from specification is uncovered and a deficiency lists is created.
- Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

#### **8.3.4: Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

#### **8.3.5: Random Data Testing**

Alpha testing has been done by using random data and checking the output. In this, in house testing has been done by using already available bills of inventory procurement by merchant and tested to see if the end result provided by IMS is the same as is available in the bill.

#### **8.3.6: User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

- Input Screen Design
- Output Screen Design
- Format of reports and other outputs.

The beta testing report is available below in figure 8.3.6

#### **8.3.7: Security Mechanisms**

This system is provided with authentication without which no user can pass. So only the legitimate users are allowed to use the application. If the legitimate users share

the authentication information then the system is open to outsiders. Also, in-case, a user forgets his/her password, then, the user can login via OTP sent to the email address of the user.

**Inventory Management System Evaluation Form**

1. Name: Sumeer Borseh

2. Organisation for which software is used: Mazgaon Medical

3. Testing period: 6 Month

4. How was your experience while using the software?

Bad       Good       Average       Excellent

5. Did the software work as expected?

No       Sometimes       Most of the times       Yes, always

6. Did the software satisfy all of your needs?

Not Quite       Some       Almost       Yes

7. Did you encounter any crashes during your usage?

Yes       No

8. Any problems faced during usage?

No

9. Any feature that you would like to see added in the next update?

Print- Report-

10. Any suggestions?

Designs Improvement-

MAZGAON MEDICAL

Sumeer Borseh  
Signature:  
19/08/2022

Fig 8.3.6: Beta Testing Report

## Chapter 9: Future Scope And Further Enhancement

### 9.1: Introduction

The work entitled "*Inventory Management System Of Mazgaon Medical*" presented in this report is done as a one semester student project for the course IT 'A' level. During this short period sincere efforts were put to incorporate all the user requirements. Database was carefully designed so that it has minimum possible redundancies and future enhancements and modifications, if necessary, can be done easily. Interfaces were designed and developed in such a way to make the system as user-friendly as possible.

#### 9.1.1: Future Scope

The project may be helpful in near future to explain and maintain clear details of the various kinds of inventory regarding their maintenance, updating and editing the status of the inventory.

#### 9.1.2: Future Scope Of Enhancement

Following are few possible enhancements that can be done in this project:

- Some more functionality can be added to improve the system.
- There is some scope of enhancement in few data-entry forms.
- Some more security features can be added to make the database more secure and reliable.

## Chapter 10: Code Snippets

### 10.1: owner\_info.php

```
<?php
include 'auth.php';
include 'owner_info_process.php';
include 'config.php';
?>

<html>

<head>
    <title>
        Mazgoan Medical
    </title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" href="favicon.png">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
    <link rel="stylesheet" href="mazgaonmedical.css">
    <script src="mazgaonmedical.js"></script>
    <link rel="stylesheet" href="warningmsg.css">
</head>

<body>
    <h3>
        <?php
        if (isset($_SESSION['message'])) :
            echo '<div id=' . $_SESSION["msg_type"] . '>' .
$_SESSION["message"] . '</div>';
            unset($_SESSION['message']);
        endif
        ?>

        <div id='msg' class="msg"></div>
        <?php include 'header.php'; ?>
    </h3>
    <?php
$result = mysqli_query($con, "SELECT * FROM owner_info");
$rowcount = mysqli_num_rows($result);
if ($rowcount > 0) {
    while ($row = mysqli_fetch_array($result)) {
```

```

        $owner_name = $row['owner_name'];
        $owner_address = $row['owner_address'];
        $owner_phone = $row['owner_phone_number'];
        $license = $row['dl_number'];
        $gstin = $row['gstin'];
        $vpa = $row['vpa'];
        $pn = $row['upi_phone'];
    }
} else {
    $owner_name = "";
    $owner_address = "";
    $owner_phone = "";
    $license = "";
    $gstin = "";
    $vpa = "";
    $pn = "";
}
?>

<form method="POST" action="owner_info_process.php">
    <div class="OwnerDetails">
        <fieldset>
            <legend><b>Owner Details </b></legend>
            <label for="name">Name :</label>
            <input type="text" id="name" name="name" required
value="<?php echo $owner_name; ?>" placeholder="Enter Your Name" />

            <label for="address">Address :</label>
            <input type="text" id="address" name="address" required
value="<?php echo $owner_address; ?>" placeholder="Enter Your Address" />

            <label for="my_phone_address">Phone Number :</label>
            <input type="text" id="my_phone_number"
name="my_phone_number" required value="<?php echo $owner_phone; ?>"
placeholder="Enter Your Phone Number" />

            <label for="dl_number">Licence Number :</label>
            <input type="text" id="dl_number" name="dl_number"
required value="<?php echo $license; ?>" placeholder="Enter Your License
Number" />

            <label for="gstin">GSTIN :</label>
            <input type="text" id="gstin" name="gstin" required
value="<?php echo $gstin; ?>" placeholder="Enter Your GSTIN" />

            <label for="vpa">Virtual Payment Address :</label>
            <input type="text" id="vpa" name="vpa" value="<?php echo
$vpa; ?>" placeholder="Enter Your Virtual Payment Address" />

```

```

        <label for="pn">Phone Number Connected To UPI :</label>
        <input type="text" id="pn" name="pn" value=<?php echo
$pn; ?>" placeholder="Enter Your Phone Number Connected to UPI" />

        <button type="submit" name="update_details">Update
Details</button>
    </fieldset>
</div>

</form>

</body>

</html>

```

## 10.2: owner\_info\_process.php

```

<?php
include 'auth.php';
include 'config.php';

if(isset($_POST['update_details'])){
    $owner_name=$_POST['name'];
    $owner_address=$_POST['address'];
    $owner_phone=$_POST['my_phone_number'];
    $license=$_POST['dl_number'];
    $gstin=$_POST['gstin'];
    $vpa= $_POST['vpa'];
    $pn=$_POST['pn'];

    $result=mysqli_query($con,"SELECT * FROM owner_info");
    $rowcount=mysqli_num_rows($result);
    if($rowcount > 0){
        while($row=mysqli_fetch_array($result)){
            $serial= $row['serial'];
        }
        mysqli_query($con,"UPDATE owner_info SET owner_name='$owner_name',
owner_address='$owner_address', "
        . "owner_phone_number='$owner_phone', dl_number='$license',
gstin='$gstin', vpa='$vpa', upi_phone='$pn' "
        . "WHERE serial='$serial'");
        $_SESSION['message']="Owner Info Updated.";
        $_SESSION['msg_type']="message_general";
    }else{
        mysqli_query($con,"INSERT INTO
owner_info(owner_name,owner_address,owner_phone_number,dl_number,gstin,vp
a,upi_phone) "
    
```

```

        . "VALUES
('$owner_name', '$owner_address', '$owner_phone', '$license', '$gstin', '$vpa'
,'$pn')");
$_SESSION['message']="Owner Info Added.";
$_SESSION['msg_type']="message_success";
}
header("location: owner_info.php");
}

```

### **10.3: auth.php**

```

<?php
include_once 'login_process.php';
if (!isset($_SESSION['authenticated'])) {
    header('location:index.php');
}

```

### **10.4: config.php**

```

<?php
$host = "localhost"; /* Host name */
$user = "root"; /* User */
$password = ""; /* Password */
$db1name = "db1"; /* Database name */

$con = mysqli_connect($host, $user, $password, $db1name);
// Check connection
if (!$con) {
    die("Connection failed for db1: " . mysqli_connect_error());
}

```

### **10.5: login\_process.php**

```

<?php
include 'config.php';
session_start();

if(isset($_POST['login'])){
    $username=$_POST['username'];
    $password=$_POST['password'];

    $query_login=mysqli_query($con,"SELECT * FROM login WHERE
username='$username' AND password='$password'");
    $rowcount_login=mysqli_num_rows($query_login);
    if($rowcount_login>0){
        while($erow_info = mysqli_fetch_array($query_login)){
            $_SESSION['user_id']=$erow_info['user_id'];
            $user=$erow_info['user'];
        }
    }
}

```

```

        $username=$erow_info['username'];
        $password=$erow_info['password'];
        $_SESSION['power']=$erow_info['power'];
    }

    $_SESSION['id']= $username;
    $_SESSION['authenticated']= true;
    $_SESSION['updtpass']=0;
    if($_SESSION['power']==1){
        $_SESSION['message']="Hello ".$user.". Logged In As Admin.";
        $_SESSION['msg_type']="message_success";
        header("location: supplier_info.php");
    }else{
        $_SESSION['message']="Hello ".$user.". Logged In As User.";
        $_SESSION['msg_type']="message_general";
        header("location: inventoryN.php");
    }

}
else{
    unset ($_SESSION['authenticated']);
    $_SESSION['message']="UserName And Password Donot
Match.";
    $_SESSION['msg_type']="message_error";
    header("location: index.php");
}

}

if(isset($_POST['forpass'])){
    $usernamefor=$_POST['usernamefor'];
    $dobfor=$_POST['dobfor'];
    $answerfor=$_POST['answerfor'];
    $otp=$_POST['otp'];

    mysqli_query($con,"Update login SET otp='' WHERE now()>time_otp +
interval 5 minute");
    $result=mysqli_query($con,"SELECT * FROM login WHERE
username='".$usernamefor"'
    . " && dobfor='".$dobfor' && answer='".$answerfor' && otp='".$otp'
");
    $rowcount=mysqli_num_rows($result);
    if($rowcount > 0){

        while($erow = mysqli_fetch_array($result)){
            $_SESSION['username']=$erow['username'];
            $_SESSION['user_id']=$erow['user_id'];
            $_SESSION['password']=$erow['password'];
        }
    }
}

```

```
$_SESSION['power']=$erow['power'];

}
$_SESSION['authenticated']= true;
$_SESSION['updtpass']=1;
if($_SESSION['power']==1){
    $_SESSION['message']="Logged In As Admin.";
    $_SESSION['msg_type']="message_general";
    header("location: supplier_info.php");
}
else{
    $_SESSION['message']="Logged In As User.";
    $_SESSION['msg_type']="message_general";
    header("location: inventoryN.php");
}

}
else{
    $_SESSION['message']="Couldn't Find User. Contact Admin.";
    $_SESSION['msg_type']="message_error";
    header("location: index.php");
}
}
```

## Chapter 11: Bibliography And References

### 11.1 Bibliography:

#### 11.1.1: Books

- Beginning PHP5  
Author: Dave Mercer
- PHP CookBook  
Author: David Sklar, Adam Trachtenberg
- Mysql: The Complete Reference  
Author: Vaswani
- PHP and MySQL For Dummies  
Author: Janet Valade

#### 11.1.2: Websites

- [www.mysqltutorial.org](http://www.mysqltutorial.org)
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [www.chartjs.org](http://www.chartjs.org)
- [www.w3schools.com](http://www.w3schools.com)

#### 11.1.3: YouTube Tutorials

- Web Dev Simplified <https://www.youtube.com/WebDevSimplified>
- Fireship <https://www.youtube.com/c/Fireship>
- freeCodeCamp.org <https://www.youtube.com/c/Freecodecamp>