**PHARMACY MANAGEMENT SYSTEM**

A Project ReportSubmitted in Partial Fulfillment of the

Requirements for the Degree of

**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**

**B.SC(IT)**

**by**

**YASH BHARDWAJ**

**( Enrollment no.- R200530055 )**

**Under the Supervision of**

**Internal Guide**

Dr. Suman Pant

Assistant Professor

**SGRRITS , Dehradun**

****

**School of Computer Application & Information Technology**

**Shri Guru Ram Rai University , Dehradun**

**March , 2023**

****

**SHRI GURU RAM RAI UNIVERSITY**

Patel Nagar, Dehradun-248001, Uttarakhand, India

(Estd. by Govt. of Uttarakhand, vide Shri Guru Ram Rai University Act no. 03 of 2017)

**Certificate**

This is to certify that **Yash Bhardwaj** (**R200530055**) has carried out the project work presented in this project report entitled “**Pharmacy Management System**” for the award of **Bachelor of Information Technology** from Shri Guru Ram Rai University.

He/She has done the project under my supervision. The study and work are carried out by the student and this project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature: Signature:

Head of Department Dr.Suman Pant

(CA & IT) Assistant Professor

SCAIT , SGRRU - Dehradun

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**SHRI GURU RAM RAI UNIVERSITY**

Patel Nagar, Dehradun-248001, Uttarakhand, India

(Estd. by Govt. of Uttarakhand, vide Shri Guru Ram Rai University Act no. 03 of 2017)

**ACKNOWLEDGEMENT**

At the outset, I thank God almighty for making my endeavor a success. I express my sincere gratitude to ***Dr.Suman Pant (Assistant Professor)***, **School Of CA & IT** for her constant support and valuable suggestions without which the successful completion of this thesis would not have been possible. I express my immense pleasure and thankfulness to our HOD & all the teachers and staff of the Department of Computer Application & Information Technology for their cooperation and support.

Finally, I am thankful to all my classmates for their cooperation and support or another helped me in the successful completion of this work.

Student Name : **Yash Bhardwaj** Enrollment No. : **R200530055**

****

**SHRI GURU RAM RAI UNIVERSITY**

Patel Nagar, Dehradun-248001, Uttarakhand, India

(Estd. by Govt. of Uttarakhand, vide Shri Guru Ram Rai University Act no. 03 of 2017)

**PROFORMA FOR THE APPROVAL OF PROJECT PROPOSAL**

Name of the Student \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course \_\_\_\_\_\_\_\_\_\_\_ Batch \_\_\_\_\_\_\_\_\_\_\_\_\_ University Enrolment No. \_\_\_\_\_\_\_\_\_\_\_\_\_

Title of the Project\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of the Guide \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Working / Teaching experience of the Guide \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Software used in the Project \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Student Signature of the Guide

Date: \_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Topic** | **Page No** |
| 1 | **Introduction** | 1 |
| 2 | **System Analysis**   * Identification Of Need * Identification Of Problem * Identification Of Tasks * Timeline (Gantt Chart) * Software Requirement Specification(SRS) | 2-9 |
| 3 | **Data Models**   * Data Flow Diagram (DFD) * Entity Relationship (ER Diagram) * Design Flow Chart * Data Dictionary | 10-36 |
| 4 | **Coding**   * Overview Of Frontend Language * Tags Used * Overview Of Backend Language * Source Code File | 37-93 |
| 5 | **Output Screen Layouts** | 94-101 |
| 6 | **Conclusion And Future Work** | 102-103 |
| 7 | **Bibliography** | 104 |

**INTRODUCTION**

The Project named “Pharmacy Management System”, a Medical information system is a client/server based application.

An Interactive application for managing both Stock and Billing which helps in maintaining the records of the medicine, the users and store details and also reduce the work of searching the medicine. The main aim of this application is to apply technology is supporting the pharmacist and the store to reduce the human effort on searching and automation of the billing.

The project has been developed on the basis of “Stock managing” and it’s “billing process” being presently used in the medical stores for storing and retrieving the available information in the store.

The user has to get his username and password from the admin by providing the name, address, phone no, id proof. And can get the access to the application. Without the username and password he cannot get access to the application.

In this application we have four types of users who have their specified work to perform and while logging into the application he should be knowing his designation to log in. and in this we can manage all the stocks of the store and can manage it.

In this admin and the manager are the two who have almost all the permissions to work with.

**SYSTEM ANALYSIS**

* Identification Of Need

management system which is designed to improve the

efficiency ,safety, accuracy and productivity of a pharmacy . It is web-site which helps the

pharmasist to improve inventory management, cost, medical safety etc. This The system allows

the user to enter the production date and expiration of a particular product or drug at the time of

opening the stock and sale function. The system will also provide a report showing a list of

expired products after the date specified before the expiration of the product. It also involves

direct entry when new drug sets arrive and when drugs leave the pharmacy for a period of time,

e.g. on a monthly basis, the pharmacist may want to produce a report on the release and release

of drugs from the pharmacy, receiving information about the medication e.g. expiration date,

purchase date, residual drug number, drug location at pharmacy.

Currently, the basic system is used in pharmacies. It requires the pharmacist to personally

monitor each drug available at the pharmacy. This often leads to mistakes as the pharmacist's job

grows.

The Online Pharmacy Management System is a online website which take cares of a pharmacy’s everyday needs and problems. In this world of AI and automation every sector is growing fast and to take care of this growth and improve it every sector needs a change in there working so to improve efficiency of a pharmacy store this website is the one step solution for all these common problems. Some of the main issues which every pharmacy manager face is expiry of medicine ,stock data like how much of stock is there in there warehouse is there any medicine out of stock or near expiry.

This Pharmacy management system is a management system which is designed to improve the efficiency ,safety, accuracy and productivity of a pharmacy . It is web-site which helps the pharmacist to improve inventory management, cost, medical safety etc. This The system allows the user to enter the production date and expiration of a particular product or drug at the time of opening the stock and sale function. The system will also provide a report showing a list of expired products after the date specified before the expiration of the product. It also involves direct entry when new drug sets arrive and when drugs leave the pharmacy for a period of time, e.g. on a monthly basis, the pharmacist may want to produce a report on the release and release of drugs from the pharmacy, receiving information about the medication e.g. expiration date, purchase date, residual drug number, drug location at pharmacy.

Currently, the basic system is used in pharmacies. It requires the pharmacist to personally monitor each drug available at the pharmacy. This often leads to mistakes as the pharmacist's job grows.

* Identification Of Problem

The main problem which every pharmacy faces is due to the size and quality of the pharmacy service; the pharmacy has a very large customer base. These customers often visit the pharmacy to get services especially when they close at work. During this time, the number of customers who use the pharmacy increases, thus making the pharmacist's job more tedious. This case makes it difficult for the pharmacist to care for customers in the short term.

In the meantime, the pharmacist must ensure satisfaction in the services to keep their customers. The above factors lead to delays in customer service, thereby reducing sales and the risk of losing key customers over time.

The “Pharmacy Management System” is a landmark project designed for medical and medical stores on purpose. This project equips medical stores with the tools to perform their daily tasks electronically. The website will change the process of compiling medical inventory and drug stocks, recording medical information and other medical supplies in the store, recording prescriptions issued to the Customer. Managing a medical store is particularly complicated regarding inventory and record keeping.

Medical stores need to make sure they have enough stock of support and cater to the needs of clients. Another major issues is creating invoice like in pharmacy when a medicine is sold by invoice then we need to mention the whole name of the medicine and its batch no, and expiry date and doing all this manual all by hands is sometimes not possible due to lots of customers waiting.

Taking about some other issues are like when a pharmacist or a pharmacy is associated with a doctor then the pharmacy owner also needs to take care of doctor commission that how much prescription does a doctor send and what was the amount of medicine sold to take care of this pharmacist need to keep record of things manually. In this software we will design the software like this that when a pharmacist enters new medicine it will be stored in the database and when a new purchase bill is entered then all medicine and bact id and expiry will be updated in the database this feature will help to overcome a common problem of every pharmacy.

In medical store proper handling of medicine is very necessary to ensure the success of all store operations. Medical store are in the process of searching for an effective and reliable daily help and relief plan. There is a need for electronic intervention in the management of medical stores. Pharmacy managers keep a paper record in paper cupboards. Managing a large pharmacy with paper records will be tedious and difficult to maintain in-store drug-related inventory, expiration date, number of drugs available based on categories and their functions. The pharmacist must order drugs to replenish the already depleted stock.

In addition, drug orders are hand-delivered. A significant amount of time is allocated to write an order as the pharmacist needs to exceed the stock balance and make a rough estimate of the order amount based on Statistics. Drugs should not be used after expiration. This project manager will inform the pharmacist about expired drugs, prevent the drugs from being sold and provide a solution to the problems.

* Identification Of Tasks

The main goal of the project is to develop and implement a Medical Store Management System to digitalize the operations of a medical store. In this we need to construct a fresh website by using any preferred language I have used php ,html, css, etc. While creating this we need to remember many things first of all we need to create an database for our website where the whole data of the website will be stored.

Next we need to find out the solutions to the issues which we discuss earlier that what are the major issues pharmacist face on daily basis. First of all, we need to create a basic login option for everyone related to the pharmacy like some time in a single pharmacy there are many members working in it so for that we need to create a login page where everyone can create there id password to login their own account by this the pharmacy owner can know that who billed a certain invoice and much more. Another major issues is creating invoice like in pharmacy when a medicine is sold by invoice then we need to mention the whole name of the medicine and its batch no, and expiry date and doing all this manual all by hands is sometimes not possible due to lots of customers waiting .

After that on the home page there will be many shortcut options like generate invoice add medicine add stockiest and many more. In medical store proper handling of medicine is very necessary to ensure the success of all store operations. Medical store is in the process of searching for an effective and reliable daily help and relief plan. There is a need for electronic intervention in the management of medical stores. Pharmacy managers keep a paper record in paper cupboards. Managing a large pharmacy with paper records will be tedious and difficult to maintain in-store drug-related inventory, expiration date, number of drugs available based on categories and their functions.

The pharmacist must order drugs to replenish the already depleted stock. In addition, drug orders are hand-delivered. A significant amount of time is allocated to write an order as the pharmacist needs to exceed the stock balance and make a rough estimate of the order amount based on Statistics. Drugs should not be used after expiration. This project manager will inform the pharmacist about expired drugs, prevent the drugs from being sold and provide a solution to the problems.

General Objective - The main goal of the project is to develop and implement a Medical Store Management System to digitally transform the operation of a medical store.

• Eliminate manual methods and paperwork involved in medical store management.

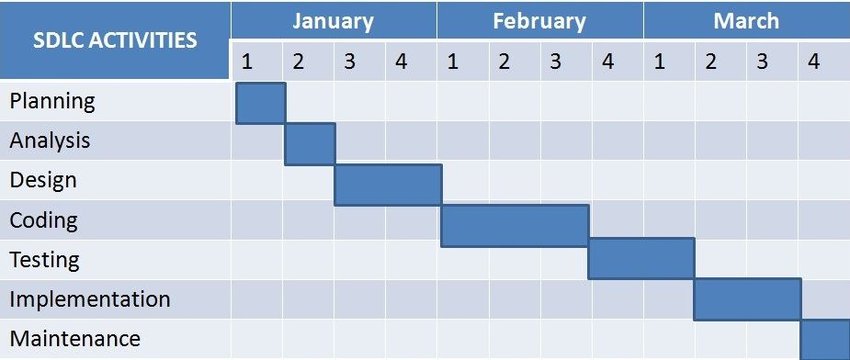
• Automatically record and store medical store data

• Allowing medical stores to compile electronic stocks to monitor stock purchases

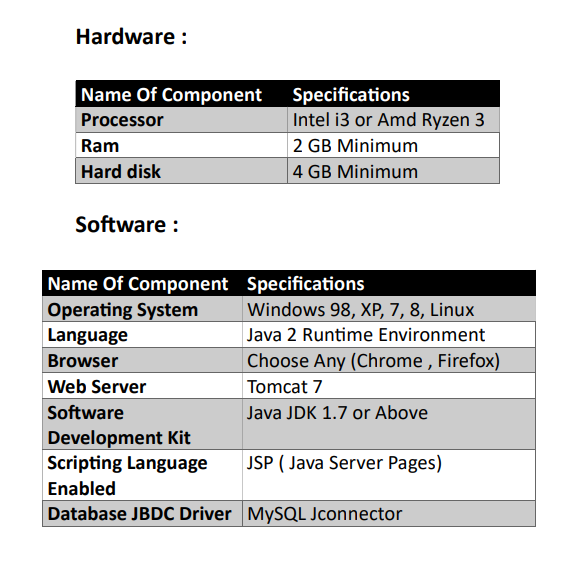
• Evaluate the system in terms of user acceptance, efficiency, quality, reliability, and productivity.

• To facilitate the reporting of store stocks, drug information (photo, expiration, etc.), release recordings, and store sales.

* Timeline (Gantt Chart)



* Software Requirement Specification



**DATA MODELS**

* Data Flow Diagram (DFD)

One of the methods used for pharmacy management system development is the **DFD (data flow diagram)**. It represents the system’s major processes and alternatives that generate the internal flow of data.

Additionally, the data was properly categorized to illustrate the pharmacy management system structure. Take note that DFD is not part of the [**Pharmacy Management System UML Diagrams**](https://itsourcecode.com/uml/pharmacy-management-system-use-case-diagram/), but they complement each other in explaining the project activities, behaviours, interactions, and structure.

**Importance of Data Flow Diagram (DFD)**

The **importance of the data flow diagram (DFD) for pharmacy management system**is to show the developers the actual happenings in the system. This is done by visualizing the system’s data management at various levels.

Furthermore, the **DFD levels**were used to discuss the **pharmacy management system** data flow. These levels have their part in expounding the system’s data flow structure details. It is then applied in creating [**Pharmacy Management System ER Diagram**](https://itsourcecode.com/uml/pharmacy-management-system-er-diagram/).

Data flow diagrams not only describe the flow of data but also denote the steps involved in transferring data from one process to another. As a result, the data was transformed from input to output.

**Advantages of Data Flow Diagram:**

The **Advantages of the Pharmacy Management System Data Flow Diagram**are as follows:

* It is a representation that aids in the visualization of system contents.
* Included in the system documentation file.
* Simple to grasp and understood by both programmers and users.
* System components were detailed and explained in DFD diagrams.
* It also aids in comprehending a system’s operation and limitations.

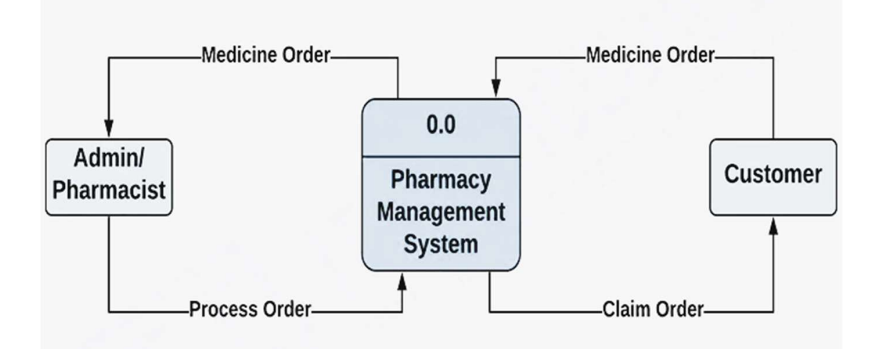
**Pharmacy Management System Data Flow Diagram (DFD)**

A thorough explanation is provided for the example **data flow diagram for pharmacy management system**. This example emphasizes the three **DFD levels (DFD Levels 0, 1, and 2)**.

**0 Level DFD for Pharmacy Management System**

The context diagram is an alternative name for the **Pharmacy Management System DFD Level 0**. Users, the main process, and data flow make up its parts. Also, the project concept is demonstrated using the single process visualization.

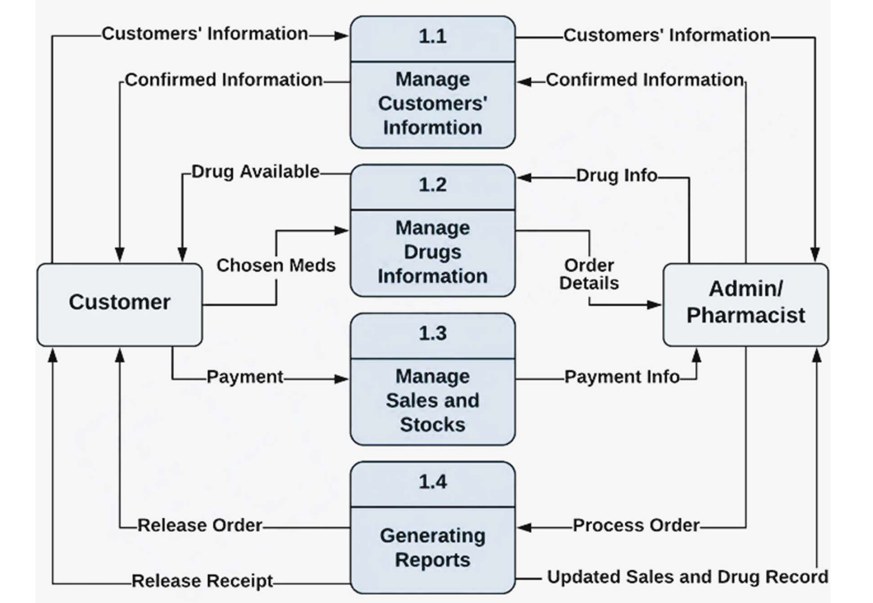
DFD Level 0 shows the entities that interact with a system and defines the border between the system and its environment. This diagram also depicts the pharmacy management system at a high level.



### ****Level 1 DFD for Pharmacy Management System****

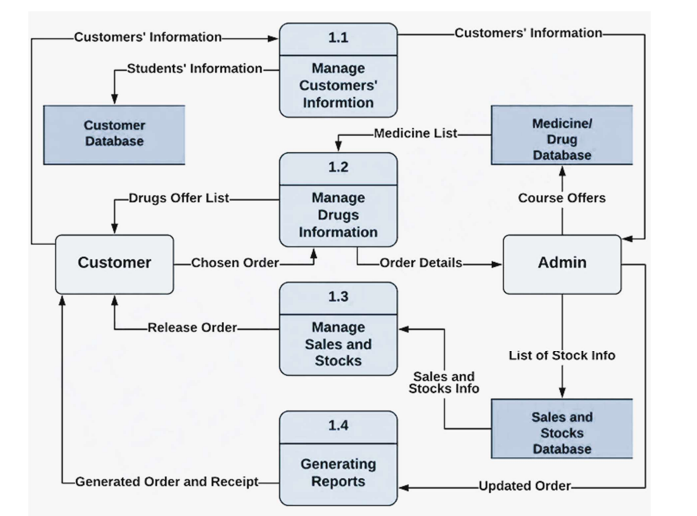
The “detonated view” of the context diagram is **Pharmacy Management System** **DFD Level 1**. Its function is to deepen the concept derive from the context diagram.

Specifically, level 1 shows the broader details of Pharmacy Management System DFD Level 0. This is to clarify the paths (flow) of data and its transformation from input to output.



### ****Level 2 DFD for Pharmacy Management System****

**Pharmacy Management System DFD Level 2** is also the highest abstraction of the data flow diagram. This level also broadens the idea from the DFD level 1. It includes the sub-processes from level 1 as well as the data that flows.



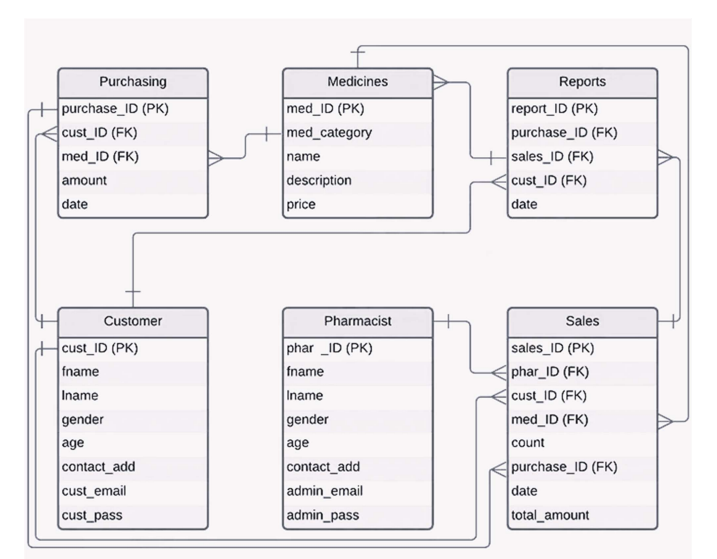
* Entity Relationship Model

The model of the **Pharmacy Management System ER Diagram** is shown in this ER (Entity Relationship) Diagram. It shows all the database tables and the connections between customers, pharmacists, medicines, purchasing, sales, and reports.

It uses structured data and relationships between structured data groups. To define the functions of the Pharmacy Management System. The Pharmacy Management System’s main parts are the Pharmacy, the Medicines, the pharmacist, the customer, the Inventory, and the reports.

**Pharmacy Management System Features**

* **Pharmacy Management** – Pharmacy Management is the main feature of this system wherein ER diagram contains the basic details needed for managing medical records and inventory. This basic information was composed of medicine records, sales, counts, and inventories. This will also monitor or check the customers’ info and purchasing or order status.
* **Customer Management** – This feature plays a big role in the system because this gathers and manages the important information of the customer. This information was used to track their orders and purchasing records and other important matters regarding the system to assure that the services were given properly.
* **Manage Medicines** – The medicine management will be done by the admin to track the activities and sales of the pharmacy. This will also monitor the performance of the pharmacy as well as their basis for inventories.
* **Manage Medicines Inventories** – Its feature will manage and monitor the inventories of the medicines as well as their sales for a given period. The activity of this will include the monitoring of the purchased medicine, the type of medicine, and their prize.

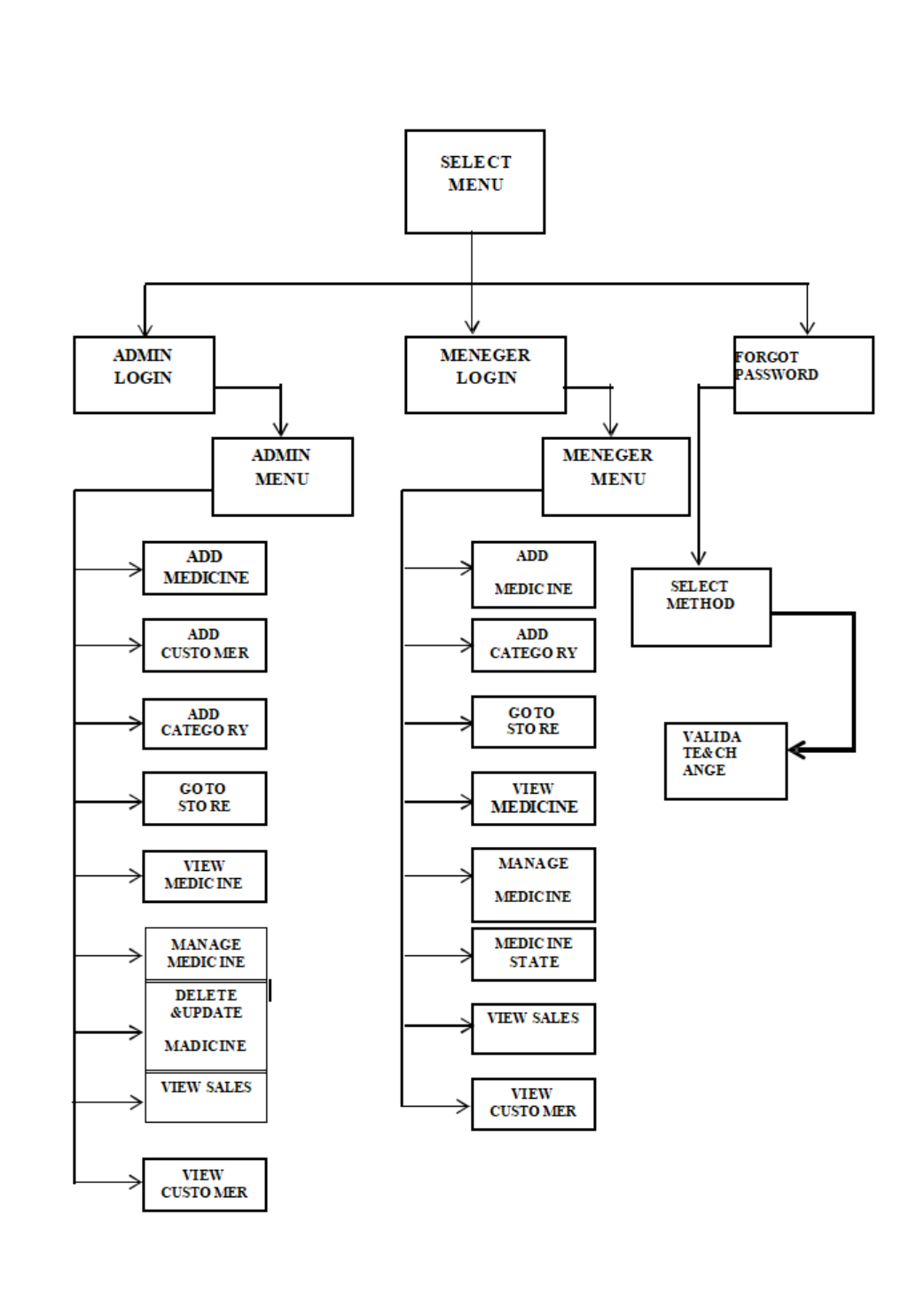


* Design Flow Chart

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements through system modeling. One could see it as the application of systems theory to produce development.

The design of this system will be user friendly. It shall be designed in such a way that employees will be able to navigate easily through the information supplied on the system. In other words, system design consists of design activities that produce system specific at ions satisfying the functional requirements that were developed in the system analysis process. System design specifies how the system will accomplish. System design is the structural implementation of the system analysis.

The proposed Pharmacy Management System Pharmacy and Stores will start with project planning by determining the users of the system, aims and objectives of the project. After these, extensive research will be done to determine how to design an effective system, as well as to review the current system. Then, the design was with an initial prototype of the system, and then refined it based on their suggestions. Phases of analysis, design and implementation were performed iteratively until users and designers agreed on a final system specification. At this point, the project could move to the final implementation phase.



* Evaluation & Selection of Specifications

At present, manual system is being utilized in the pharmacy. It requires the pharmacist to manually monitor each drug that is available in the pharmacy. Pharmacy management has kept paper record in filing cabinets. Managing a very large pharmacy with records on papers will be tedious and difficult to keep track of inventories with regards to the drugs in the store, expiry date, quantity of drugs available based on the categories and their functions. This implies that these services will be manually completed by the pharmacist. This usually leads to mistakes as the workload of the pharmacist increases. This system also ensures that there exists a level of restricted access based on functionality and role. This system also provides optimal drug inventory management by monitoring the drug movement in the pharmacy. Significant amount of time is allocated for writing the order as the pharmacist needs to go through the stock balance and make rough estimate of the amount to order based on Figures. The system will not be able to handle drug prescription, drug to drug interaction.

**Login**: There are many workers in a pharmacy so to keep a record of everyone in this Pharmacy Management System we have designed a login system for everyone like any worker of the pharmacy can create his/her account and keep his work separate by this the owner of pharmacy will have all record of every person in a pharmacy and can track the efforts of every worker in the pharmacy

**Invoice Creator**: In pharmacy when a customer purchase something and when the bill is generated this is mandatory to write expiry and batch no. Along with the medicine name and other details so doing this thing manually by the pharmacist in bill book or on the paper takes lots of time and is not the efficient way in this method like suppose if a worker sells 10 medicine then how another worker will get to known that he has already sold those medicine and are not anymore in stock, so this is the main reason where generating invoice using software is very helpful and efficient.

**Inventory management**: When lot of medicine came from any of the medical agencies the worker need to match the batch id manually and there is no record if any medicine worker ordered was delivered or not so to overcome these type of situations inventory management system comes in play where all the details like when and from which distribute the particular medicine was came and the detail of medicine will be uploaded on the website which will help to manage the stock and easy to remember that this particular medicine is out of stock.

From the problems listed in the existing system, the implementation of the proposed system shall focus on:

1. Pharmacists having access to the proposed system at any time.

2. Ensuring effective policing by providing statistics of the drugs in stock.

3. Improving the efficiency of the system by ensuring effective monitoring of services and activities.

4. Generating report within a specified period.

5. Reducing the employees’ workload.

**Design Constraints**

Regulations/Economic/Environmental/Health/manufacturability/Safety/Professional/ Ethical/Social & Political Issues/Cost considered in design.

**Analysis And Feature Finalization Subject To Constraints**

* Functional Requirements:

There are functions done by the system such as: store the necessary information of medicine, prepare bill for the medicine, easily searching of medicine, Update, delete and save data of medicine.

* Store the necessary information of the drugs:

The PMS (pharmacy management system)stores the detail information about each medicine including Actual name, formula of medicine and how it is importance and for which diseases is required. Since the information for each drug were required in some cases like the use of drugs, when use drugs and for whom is given.

* Searching Medicine and other Data’s:

The PS has easily searching of medicine which shows in which shelf is put and the behavior of the medicine. This arching process is based on the name of the give data or the identification of the item. Here when the user searches the item on search bar the related things were displayed in the screen and can select the actual item that the user needs.

* Alerting pharmacy Data in the system:

Changing medicines to another because of medicine outdated, modifying the saved medicine data for incorrect data, deleting of data of the pharmacy can be done on the system.

* Nonfunctional Requirements:

This pharmacy management system is able to operate in the following characteristics.

* Usability:

Any familiar in using windows operation can operate the system since it has user friendly user interface. Which have the instruction menu’s how to use it which self-directive application then can be used the system without ambiguity.

* Reliability:

The pharmacy system is available based on the user needs, can work properly, and do transactions efficiently including safe data management of the pharmacy. For invalid and malfunctioned operation, the system will restart in order to prevent data loose as well as safe operation within 5 seconds. The pharmacy system is password protected to change things on the system. Here the pharmacist manager control over the system by login to the pharmacy system. Any user can’t use the system but the guest user can see on general properties of the pharmacy and medicines without password. As result data is protected and controlled by only the administrator.

* Performance:

The pharmacy management system operates its function in small amount of time which is less than two seconds and can be accessed by one user at a time or concurrently. To access the user must first login to the system which must have the pharmacy system privileged and the system can store data up to 40 GB data. When the system may be busy due to malfunction operation it may wait up to one minute other ways the pharmacy system restarts.

* User interface:

the user interface is friendly which is easy to use. And having attractive frame structure which is prepared in assumption with other related systems. Also the user can change him/her user favorite interfaces that Is available in the system.

* Operation:

the pharmacy management system is operated and controlled by the pharmacy manager for safe work.

* Supportability:

This pharmacy management system operates in any version of windows operating system. Such as windows xp, windows 2003, windows 7,windows 8 and other related versions. The system can be easily maintained by the manager of the pharmacy system by using the prepared documents of the system for easy maintenance. Other ways it is maintained by the system developers for corrective and other heavy problems.

* Data Dictionary

These tables below provide the complete database table details such as **Field Name**, **Descriptions**, **data types**, and **character lengths**. Each of these tables represents the characteristics and attributes of data storage.

The **field**column presents the names of each database’s attributes, the **description**column gives the complete thought of each attribute, the **type**column is their data type and the **length**is for their character lengths.

Table Name: Admin Credentials

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| USERNAME | Username | Varchar | 50 |
| PASSWORD | Password | Varchar | 50 |

Table Name: Customers

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| ID | Customer ID | Int | 11 |
| NAME | Name | Varchar | 20 |
| CONTACT\_NUMBER | Number | Varchar | 10 |
| ADDRESS | Address | Varchar | 100 |
| DOCTOR\_NAME | Doctor Name | Varchar | 20 |
| DOCTOR\_ADDRESS | Doctor Address | Varchar | 100 |

Table Name: Invoices

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| INVOICE\_ID | Invoice ID | Int | 11 |
| NET\_TOTAL | Total | Double |  |
| INVOICE\_DATE | Invoice Date | date |  |
| CUSTOMER**\_**ID | Customer ID | Int | 11 |
| TOTAL\_AMOUNT | Age | Double |  |
| TOTAL\_DISCOUNT | Total Discount | Double |  |

Table Name: Medicines

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| ID | ID | Int | 11 |
| NAME | Name | Varchar | 100 |
| PACKING | Packing | Varchar | 20 |
| GENERIC\_NAME | Generic Name | Varchar | 100 |
| SUPPLIER\_NAME | Supplier Name | Varchar | 100 |

Table Name: Medicines Stock

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| ID | ID | Int | 11 |
| NAME | Name | Varchar | 100 |
| BATCH\_ID | Batch ID | Varchar | 20 |
| EXPIRY\_DATE | Date of expiry | Varchar | 10 |
| QUANTITY | Quantity | Int | 11 |
| MRP | Maximum Retail Price | Double |  |
| RATE | Rate | Double |  |

Table Name: Purchases

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| SUPPLIER\_NAME | Supplier Name | Varchar | 100 |
| INVOICE\_NUMBER | Invoice Number | Int | 11 |
| VOUCHER\_NUMBER | Voucher Number | Int | 11 |
| PURCHASE\_DATE | Purchase Date | Varchar | 10 |
| TOTAL\_AMOUNT | Total Amount | Double |  |
| PAYMENT\_STATUS | Payment Status | Varchar | 20 |

Table Name: Suppliers

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Description** | **Type** | **Length** |
| ID | ID | Int | 11 |
| NAME | Name | Varchar | 100 |
| EMAIL | Email Address | Varchar | 100 |
| CONTACT\_NUMBER | Contact Number | Varchar | 10 |
| ADDRESS | Address | Varchar | 100 |

**CODING**

* Overview Of the Language Used In Frontend

**PHP:**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Leardof unleashed the first version of PHP way back in 1994.PHP is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP.

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 many major protocols such as POP3, IMAP, and LDAP. PHP4added 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.

Characteristics of PHP Five important characteristics make PHP's practical nature possible –

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

**HTML:**

The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images, and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

HTML elements are delineated by tags, written using angle brackets. Tags such as <img/>and <input />directly introduce content into the page. Other tags such as<p>surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the<canvas>element, in collaboration with JavaScript.

**CSS:**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/cssis registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

**Bootstrap:**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. As of April 2022, Bootstrap is the eleventh most starred project on GitHub, with over 156,000 stars.

­­ Html tags used:

<!--...-->

These are comments, that are not visible to the user but only when you view the actual HTML. They are also useful in commenting out a block of code when you're doing testing, that way you can prevent from having to cut and paste the code elsewhere, and then copy them back.

<!doctype>

All HTML documents must start with a <!DOCTYPE> declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In HTML 5, the declaration is simple:

<!DOCTYPE html>

<a>

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

* An unvisited link is underlined and blue
* A visited link is underlined and purple
* An active link is underlined and red

<body>

The <body> tag defines the document's body.

The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

There can only be one <body> element in an HTML document.

<br>

The <br> tag inserts a single line break.

The <br> tag is useful for writing addresses or poems.

The <br> tag is an empty tag which means that it has no end tag.

The <br> tag used to enter line breaks, not to add space between paragraphs.

<button>

The <button> tag defines a clickable button.

Inside a <button> element you can put text (and tags like <i>, <b>, <strong>, <br>, <img>, etc.). That is not possible with a button created with the <input> element!

The type attribute is used for a <button> element, to tell browsers what type of button it is.

<div>

The <div> tag defines a division or a section in an HTML document.

The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The <div> tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the <div> tag

By default, browsers always place a line break before and after the <div> element.

<form>

The <form> tag is used to create an HTML form for user input.

The <form> element can contain one or more of the following form elements:

* <input>
* <textarea>
* <button>
* <select>
* <option>
* <optgroup>
* <fieldset>
* <label>
* <output>

<frame>

A <frame> tag is used with <frameset>, and it divides a webpage into multiple sections or frames, and each frame can contain different web pages.

<h1 to h6>

The <h1> to <h6> tags are used to define HTML headings.

<h1> defines the most important heading. <h6> defines the least important heading. Only used one <h1> per page - this should represent the main heading/subject for the whole page. Also, we should not skip heading levels - start with <h1>, then use <h2>, and so on.

<head>

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

Metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following elements can go inside the <head> element:

* <title> (required in every HTML document)
* <style>
* <base>
* <link>
* <meta>
* <script>
* <noscript>

<hr>

The <hr> tag defines a thematic break in an HTML page (e.g., a shift of topic).

The <hr> element is most often displayed as a horizontal rule that is used to separate content (or define a change) in an HTML page.

<html>

The <html> tag represents the root of an HTML document.

The <html> tag is the container for all other HTML elements (except for the <!DOCTYPE> tag).

The tag includes the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

<img>

The <img> tag is used to embed an image in an HTML page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

The <img> tag has two required attributes:

* src - Specifies the path to the image
* alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed.

The width and height of an image needs to be specified. If width and height are not specified, the page might flicker while the image loads.

<input>

The <input> tag specifies an input field where the user can enter data.

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the type attribute.

The different input types are as follows:

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text"> (default value)
* <input type="time">
* <input type="url">
* <input type="week">

<label>

The <label> tag defines a label for several elements:

* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="search">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">
* <meter>
* <progress>
* <select>
* <textarea>

Proper use of labels with the elements above will benefit:

* Screen reader users (will read out loud the label, when the user is focused on the element)
* Users who have difficulty clicking on very small regions (such as checkboxes) - because when a user clicks the text within the <label> element, it toggles the input (this increases the hit area).

The for attribute of <label> must be equal to the id attribute of the related element to bind them together. A label can also be bound to an element by placing the element inside the <label> element.

<link>

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Links are found in nearly all web pages. Links allow users to click their way from page to page.

<meta>

The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.

<meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parable.

Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

There is a method to let web designers take control over the viewport (the user's visible area of a web page), through the <meta> tag.

<meter>

The <meter> tag defines a scalar measurement within a known range, or a fractional value. This is also known as a gauge.

Examples: Disk usage, the relevance of a query result, etc.

The <meter> tag should not be used to indicate progress (as in a progress bar). For progress bars, use the <progress> tag.

<nav>

The <nav> tag defines a set of navigation links. Not all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

<li>

The <li> tag defines a list item.

The <li> tag is used inside ordered lists(<ol>), unordered lists (<ul>), and in menu lists (<menu>).

In <ul> and <menu>, the list items will usually be displayed with bullet points.

In <ol>, the list items will usually be displayed with numbers or letters.

<ol>

The <ol> tag defines an ordered list. An ordered list can be numerical or alphabetical.

The <li> tag is used to define each list item.

<p>

The <p> tag defines a paragraph.

Browsers automatically add a single blank line before and after each <p> element.

<script>

The <script> tag is used to embed a client-side script (JavaScript).

The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

<section>

The <section> tag defines a section in a document.

<span>

The <span> tag is an inline container used to mark up a part of a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

<style>

The <style> tag is used to define style information (CSS) for a document.

Inside the <style> element it should be specified how HTML elements should render in a browser.

<table>

The <table> tag defines an HTML table.

An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.

The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.

An HTML table may also include <caption>, <colgroup>, <thead>, <tfoot>, and <tbody> elements.

<title>

The <title> tag defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> tag is required in HTML documents!

The contents of a page title are very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

* defines a title in the browser toolbar
* provides a title for the page when it is added to favorites
* displays a title for the page in search-engine results

CSS properties used:

background-color

The background-color property specifies the background color of an element.

With CSS, a color is most often specified by:

a valid color name - like "red"

a HEX value - like "#ff0000”

an RGB value - like "rgb (255,0,0)"

Look at CSS Color Values for a complete list of possible color values.

background-image

The background-image property sets one or more background images for an element.

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

The background of an element is the total size of the element, including padding and border (but not the margin).

Example

Sets two background images for the <body> element. Let the first image appear only once (with no-repeat), and let the second image be repeated:

body {

background-image: url("img\_tree.gif"), url("paper.gif"); background-repeat: no-repeat, repeat;

background-color: #cccccc;

}

background-position

The background-position property sets the starting position of a background image.

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

Example

How to position a background-image to be centered at top:

body {

background-image: url('w3css.gif');

background-repeat: no-repeat;

background-attachment: fixed;

background-position: center top;

}

background-repeat

The background-repeat property sets if/how a background image will be repeated.

By default, a background-image is repeated both vertically and horizontally.

The background image is placed according to the background-position property. If no background-position is specified, the image is always placed at the element's top left corner.

Example

Repeat a background image both vertically and horizontally (this is default):

body {

background-image: url("paper.gif");

background-repeat: repeat

}

background-size

The background-size property specifies the size of the background images.

There are four different syntaxes you can use with this property: the keyword syntax ("auto", "cover" and "contain"), the one-value syntax (sets the width of the image (height becomes "auto"), the two-value syntax (first value: width of the image, second value: height), and the multiple background syntax (separated with comma).

Example

Specify the size of a background-image with "auto" and in pixels:

#example1 {

background: url(mountain.jpg);

background-repeat: no-repeat;

background-size: auto;

}

#example2 {

background: url(mountain.jpg);

background-repeat: no-repeat;

background-size: 300px 100px;}

Border

The border-style property specifies what kind of border to display.

The following values are allowed:

* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

border-color

The border-color property sets the color of an element's four borders. This property can have from one to four values.

If the border-color property has four values:

* + border-color: red green blue pink;
* top border is red
* right border is green
* bottom border is blue
* left border is pin

If the border-color property has three values:

* + border-color: red green blue;
* top border is red
* right and left borders are green
* bottom border is blue

If the border-color property has two values:

* + border-color: red green;
    - top and bottom borders are red
    - right and left borders are green

If the border-color property has one value:

* border-color: red;
* all four borders are red

We should declare the border-style property before the border-color property. An element must have borders before you can change the color.

border-spacing

The border-spacing property sets the distance between the borders of adjacent cells.

This property works only when border-collapse is separate

Set the border-spacing for a table:

#table1 {

border-collapse: separate;

border-spacing: 15px;

}

#table2 {

border-collapse: separate;

border-spacing: 15px 50px;

}

Display

The display property specifies the display behavior (the type of rendering box) of an element.

In HTML, the default display property value is taken from the HTML specifications or from the browser/user default style sheet. The default value in XML is inline, including SVG elements.

Example

Use of some different display values:

p.ex1 {display: none;}

p.ex2 {display: inline;}

p.ex3 {display: block;}

p.ex4 {display: inline-block;}

float

The float property is used for positioning and formatting content e.g., let an image float left to the text in a container.

The float property can have one of the following values:

* left - The element floats to the left of its container
* right - The element floats to the right of its container
* none - The element does not float (will be displayed just where it occurs in the text). This is default
* inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.

Example - float: right;

The following example specifies that an image should float to the right in a text:

Font Selection is Important

Choosing the right font has a huge impact on how the readers experience a website.

The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Generic Font Families

In CSS there are five generic font families:

1. Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.

2. Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.

3. Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.

4. Cursive fonts imitate human handwriting.

5. Fantasy fonts are decorative/playful fonts.

font-family

The font-family property specifies the font for an element.

The font-family property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

There are two types of font family names:

* family-name - The name of a font-family, like "times", "courier", "arial", etc.
* generic-family - The name of a generic-family, like "serif", "sans-serif", "cursive", "fantasy", "monospace".

Start with the font you want, and always end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

font-size

The font-size property sets the size of a font.

Example

Set the font size for different elements:

div.a {

font-size: 15px;

}

div.b {

font-size: large;

}

div.c {

font-size: 150%;

}

font-style

The font-style property specifies the font style for a text.

Example

Set different font styles for three paragraphs

: p.a {

font-style: normal;

}

p.b {

font-style: italic;

}

p.c {

font-style: oblique;

}

font-weight

The font-weight property sets how thick or thin characters in text should be displayed.

Example

Set different font weight for three paragraphs:

p.normal {

font-weight: normal;

}

p.thick {

font-weight: bold;

}

p.thicker {

font-weight: 900;

}

Height

The height property sets the height of an element.

The height of an element does not include padding, borders, or margins!

If height: auto; the element will automatically adjust its height to allow its content to be displayed correctly.

If height is set to a numeric value (like pixels, (r)em, percentages) then if the content does not fit within the specified height, it will overflow. How the container will handle the overflowing content is defined by the overflow property.

The min-height and max-height properties override the height property.

Example

Set the height of two <div> elements:

div.a {

height: auto;

border: 1px solid black;

}

div.b {

height: 50px; border: 1px solid black;

}

Left

The left property affects the horizontal position of a positioned element. This property has no effect on non-positioned elements.

* If position: absolute; or position: fixed; - the left property sets the left edge of an element to a unit to the left of the left edge of its nearest positioned ancestor.
* If position: relative; - the left property sets the left edge of an element to a unit to the left/right of its normal position.
* If position: sticky; - the left property behaves like its position is relative when the element is inside the viewport, and like its position is fixed when it is outside.
* If position: static; - the left property has no effect.

Example

Set the left edge of a positioned <div> element to 150px from the left edge of its nearest positioned ancestor:

div.c {

position: absolute;

left: 150px;

width: 200px;

height: 120px;

border: 3px solid green;

}

line-height

The line-height property specifies the height of a line.

Example

Set the line height for different <div> elements:

div.a {

line-height: normal;

}

div.b {

line-height: 1.6;

}

div.c {

line-height: 80%;

}

div.d {

line-height: 200%;

}

Margin

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

margin-top

The margin-top property sets the top margin of an element.

* Overview Of the Language Used In Backend

**MySQL**

MySQL is popular among all databases, and is ranked as the 2nd most popular database, only slightly trailing Oracle Database. Among open source databases, MySQL is the most popular database in use today. Known as one of the most reliable and performative databases out there, it was named after it’s founders daughter My, and is known for organizing data into one or more data tables in which data types are related to each other. These relations help structure data, as SQL is a language programmers use for creation, modification and extraction of data from a relational database.

MySQL uses standalone clients that allow users to interact with MySQL, and also to use it with other programs for applications that need relational database capabilities. MySQL's reputation for reliability has led to its inclusion in the popular LAMP stack (Linux, Apache, MySQL, Python/Perl/PHP) and is also used as the default DBMS in popular CMS options like Drupal, Joomla, php, and WordPress.

It is licensed under the GNU General Public License and is also available under several proprietary licenses. When Oracle bought MySQL AB in 2010, Michael "Monty" Widenius, MySQL founder, forked MySQL into a free, open source database called MariaDB -- with the intention of keeping the MariaDB project free and open source forever.

MySQL is a relational database that uses structured query language. Relational databases are a type of database that uses a structure that allows us to identify and access the data in relation to another piece of data inside of the database. This format is often organized as tables

MySQL Use Case:

Use cases for MySQL include, but are not limited to:

* Elastic Replication - Where an environment requires the number OS servers to grow and shrink dynamically.
* High Availability - Where sharding is used for write scale-out (in which each shard maps to a replication group).
* Source-Replica Replication Alternative - Which allows using a single source server to make it a single point of contention.
* Autonomic Systems - Which allows users to deploy MySQL Group Replication for the automation that is built into replication protocol.`
* Source Code

Database File (pharmacy.sql)

-- phpMyAdmin SQL Dump

-- version 5.0.2

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Oct 26, 2021 at 08:20 AM

-- Server version: 10.4.14-MariaDB

-- PHP Version: 7.2.33

**SET** SQL\_MODE **=** "NO\_AUTO\_VALUE\_ON\_ZERO"**;**

**START** **TRANSACTION;**

**SET** time\_zone **=** "+00:00"**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/**;**

/\*!40101 SET NAMES utf8mb4 \*/**;**

--

-- Database: `pharmacy`

--

-- --------------------------------------------------------

--

-- Table structure for table `admin\_credentials`

--

**CREATE** **TABLE** `admin\_credentials` **(**

`USERNAME` varchar**(**50**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`PASSWORD` varchar**(**50**)** **COLLATE** utf16\_bin **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `admin\_credentials`

--

**INSERT** **INTO** `admin\_credentials` **(**`USERNAME`**,** `PASSWORD`**)** **VALUES**

**(**'admin'**,** 'admin123'**);**

-- --------------------------------------------------------

--

-- Table structure for table `customers`

--

**CREATE** **TABLE** `customers` **(**

`ID` int**(**11**)** **NOT** **NULL,**

`NAME` varchar**(**20**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`CONTACT\_NUMBER` varchar**(**10**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`ADDRESS` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`DOCTOR\_NAME` varchar**(**20**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`DOCTOR\_ADDRESS` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `customers`

--

**INSERT** **INTO** `customers` **(**`ID`**,** `NAME`**,** `CONTACT\_NUMBER`**,** `ADDRESS`**,** `DOCTOR\_NAME`**,** `DOCTOR\_ADDRESS`**)** **VALUES**

**(**4**,** 'Kiran Suthar'**,** '1234567690'**,** 'Andheri East'**,** 'Anshari'**,** 'Andheri East'**),**

**(**6**,** 'Aditya'**,** '7365687269'**,** 'Virar West'**,** 'Xyz'**,** 'Virar West'**),**

**(**11**,** 'Shivam Tiwari'**,** '6862369896'**,** 'Dadar West'**,** 'Dr Kapoor'**,** 'Dadar East'**),**

**(**13**,** 'Varsha Suthar'**,** '7622369694'**,** 'Rani Station'**,** 'Dr Ramesh'**,** 'Rani Station'**),**

**(**14**,** 'Prakash Bhattarai'**,** '9802851472'**,** 'Pokhara-16, Dhikidada'**,** 'Hari Bahadur'**,** 'Matepani-12'**);**

-- --------------------------------------------------------

--

-- Table structure for table `invoices`

--

**CREATE** **TABLE** `invoices` **(**

`INVOICE\_ID` int**(**11**)** **NOT** **NULL,**

`NET\_TOTAL` double **NOT** **NULL** **DEFAULT** 0**,**

`INVOICE\_DATE` date **NOT** **NULL** **DEFAULT** **current\_timestamp(),**

`CUSTOMER\_ID` int**(**11**)** **NOT** **NULL,**

`TOTAL\_AMOUNT` double **NOT** **NULL,**

`TOTAL\_DISCOUNT` double **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `invoices`

--

**INSERT** **INTO** `invoices` **(**`INVOICE\_ID`**,** `NET\_TOTAL`**,** `INVOICE\_DATE`**,** `CUSTOMER\_ID`**,** `TOTAL\_AMOUNT`**,** `TOTAL\_DISCOUNT`**)** **VALUES**

**(**1**,** 30**,** '2021-10-19'**,** 14**,** 30**,** 0**),**

**(**2**,** 2626**,** '2021-10-19'**,** 6**,** 2626**,** 0**);**

-- --------------------------------------------------------

--

-- Table structure for table `medicines`

--

**CREATE** **TABLE** `medicines` **(**

`ID` int**(**11**)** **NOT** **NULL,**

`NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`PACKING` varchar**(**20**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`GENERIC\_NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`SUPPLIER\_NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `medicines`

--

**INSERT** **INTO** `medicines` **(**`ID`**,** `NAME`**,** `PACKING`**,** `GENERIC\_NAME`**,** `SUPPLIER\_NAME`**)** **VALUES**

**(**1**,** 'Nicip Plus'**,** '10tab'**,** 'Paracetamole'**,** 'BDPL PHARMA'**),**

**(**2**,** 'Crosin'**,** '10tab'**,** 'Hdsgvkvajkcbja'**,** 'Kiran Pharma'**),**

**(**4**,** 'Dolo 650'**,** '15tab'**,** 'paracetamole'**,** 'BDPL PHARMA'**),**

**(**5**,** 'Gelusil'**,** '10tab'**,** 'mint fla'**,** 'Desai Pharma'**);**

-- --------------------------------------------------------

--

-- Table structure for table `medicines\_stock`

--

**CREATE** **TABLE** `medicines\_stock` **(**

`ID` int**(**11**)** **NOT** **NULL,**

`NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`BATCH\_ID` varchar**(**20**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`EXPIRY\_DATE` varchar**(**10**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`QUANTITY` int**(**11**)** **NOT** **NULL,**

`MRP` double **NOT** **NULL,**

`RATE` double **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `medicines\_stock`

--

**INSERT** **INTO** `medicines\_stock` **(**`ID`**,** `NAME`**,** `BATCH\_ID`**,** `EXPIRY\_DATE`**,** `QUANTITY`**,** `MRP`**,** `RATE`**)** **VALUES**

**(**1**,** 'Crosin'**,** 'CROS12'**,** '12/34'**,** 2**,** 2626**,** 26**),**

**(**2**,** 'Gelusil'**,** 'G327'**,** '12/42'**,** 0**,** 15**,** 12**),**

**(**3**,** 'Dolo 650'**,** 'DOLO327'**,** '01/23'**,** 3**,** 30**,** 24**),**

**(**4**,** 'Nicip Plus'**,** 'NI325'**,** '05/22'**,** 3**,** 32.65**,** 28**);**

-- --------------------------------------------------------

--

-- Table structure for table `purchases`

--

**CREATE** **TABLE** `purchases` **(**

`SUPPLIER\_NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`INVOICE\_NUMBER` int**(**11**)** **NOT** **NULL,**

`VOUCHER\_NUMBER` int**(**11**)** **NOT** **NULL,**

`PURCHASE\_DATE` varchar**(**10**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`TOTAL\_AMOUNT` double **NOT** **NULL,**

`PAYMENT\_STATUS` varchar**(**20**)** **COLLATE** utf16\_bin **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

-- --------------------------------------------------------

--

-- Table structure for table `suppliers`

--

**CREATE** **TABLE** `suppliers` **(**

`ID` int**(**11**)** **NOT** **NULL,**

`NAME` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`EMAIL` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`CONTACT\_NUMBER` varchar**(**10**)** **COLLATE** utf16\_bin **NOT** **NULL,**

`ADDRESS` varchar**(**100**)** **COLLATE** utf16\_bin **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf16 **COLLATE=**utf16\_bin**;**

--

-- Dumping data for table `suppliers`

--

**INSERT** **INTO** `suppliers` **(**`ID`**,** `NAME`**,** `EMAIL`**,** `CONTACT\_NUMBER`**,** `ADDRESS`**)** **VALUES**

**(**1**,** 'Desai Pharma'**,** 'desai@gmail.com'**,** '9948724242'**,** 'Mahim East'**),**

**(**2**,** 'BDPL PHARMA'**,** 'bdpl@gmail.com'**,** '8645632963'**,** 'Santacruz West'**),**

**(**9**,** 'Kiran Pharma'**,** 'kiranpharma@gmail.com'**,** '7638683637'**,** 'Andheri East'**),**

**(**10**,** 'Rsrnrnrndnn'**,** 'ydj'**,** '3737355538'**,** '3fndfndfndndfnfdndfn'**),**

**(**11**,** 'Dfnsfndfndf'**,** 'fnsn'**,** '5475734385'**,** 'Ndnss4yrhrhdhrdhrh'**),**

**(**12**,** 'SS Distributors'**,** 'ssdis@gamil.com'**,** '3867868752'**,** 'Matunga West'**),**

**(**13**,** 'Avceve'**,** 'ehh'**,** '3466626226'**,** 'Eteh266266262'**),**

**(**14**,** 'Hrshrhrjher'**,** 'dzgdg'**,** '4636347335'**,** 'Rhrswjrnswjn'**),**

**(**15**,** 'Hmrxfmgtmt'**,** 'trmtrm gm tr'**,** '6553838835'**,** '38ejtdjtdxetjdt'**),**

**(**20**,** 'Dtdxtkmtdshrrhhsrjrs'**,** 'trmtrm gm tr'**,** '6553838835'**,** '38ejtdjtdxetjdt'**),**

**(**23**,** 'Fndn'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**),**

**(**24**,** 'Fndnbrwh'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**),**

**(**25**,** 'Jnentjrtj'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**),**

**(**26**,** 'Jerthjrtjtjr'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**),**

**(**28**,** 'Gahgkakbvkv'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**),**

**(**29**,** 'Hywhwhrhdw'**,** 'nena ena'**,** '3462462642'**,** 'Ebsbsdbsdndsnsdfns'**);**

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admin\_credentials`

--

**ALTER** **TABLE** `admin\_credentials`

**ADD** **PRIMARY** **KEY** **(**`USERNAME`**);**

--

-- Indexes for table `customers`

--

**ALTER** **TABLE** `customers`

**ADD** **PRIMARY** **KEY** **(**`ID`**);**

--

-- Indexes for table `invoices`

--

**ALTER** **TABLE** `invoices`

**ADD** **PRIMARY** **KEY** **(**`INVOICE\_ID`**);**

--

-- Indexes for table `medicines`

--

**ALTER** **TABLE** `medicines`

**ADD** **PRIMARY** **KEY** **(**`ID`**);**

--

-- Indexes for table `medicines\_stock`

--

**ALTER** **TABLE** `medicines\_stock`

**ADD** **PRIMARY** **KEY** **(**`ID`**),**

**ADD** **UNIQUE** **KEY** `BATCH\_ID` **(**`BATCH\_ID`**);**

--

-- Indexes for table `purchases`

--

**ALTER** **TABLE** `purchases`

**ADD** **PRIMARY** **KEY** **(**`VOUCHER\_NUMBER`**);**

--

-- Indexes for table `suppliers`

--

**ALTER** **TABLE** `suppliers`

**ADD** **PRIMARY** **KEY** **(**`ID`**);**

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `customers`

--

**ALTER** **TABLE** `customers`

**MODIFY** `ID` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**15**;**

--

-- AUTO\_INCREMENT for table `invoices`

--

**ALTER** **TABLE** `invoices`

**MODIFY** `INVOICE\_ID` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**3**;**

--

-- AUTO\_INCREMENT for table `medicines`

--

**ALTER** **TABLE** `medicines`

**MODIFY** `ID` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**6**;**

--

-- AUTO\_INCREMENT for table `medicines\_stock`

--

**ALTER** **TABLE** `medicines\_stock`

**MODIFY** `ID` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**5**;**

--

-- AUTO\_INCREMENT for table `purchases`

--

**ALTER** **TABLE** `purchases`

**MODIFY** `VOUCHER\_NUMBER` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**;**

--

-- AUTO\_INCREMENT for table `suppliers`

--

**ALTER** **TABLE** `suppliers`

**MODIFY** `ID` int**(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**30**;**

**COMMIT;**

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/**;**

Dashboard File (home.php)

<!DOCTYPE html>

<html lang=**"en"** dir=**"ltr"**>

<head>

<meta charset=**"utf-8"**>

<title>**Dashboard - Home**</title>

<link rel=**"stylesheet"** href=**"bootstrap/css/bootstrap.min.css"**>

<script src=**"bootstrap/js/jquery.min.js"**></script>

<script src=**"bootstrap/js/bootstrap.min.js"**></script>

<link rel=**"stylesheet"** href=**"https://use.fontawesome.com/releases/v5.3.1/css/all.css"** integrity=**"sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"** crossorigin=**"anonymous"**>

<link rel=**"stylesheet"** href=**"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>

<link rel=**"shortcut icon"** href=**"images/icon.svg"** type=**"image/x-icon"**>

<link rel=**"stylesheet"** href=**"css/sidenav.css"**>

<link rel=**"stylesheet"** href=**"css/home.css"**>

<script src=**"js/restrict.js"**></script>

</head>

<body>

<?php **include** "sections/sidenav.html"; ?>

<div class=**"container-fluid"**>

<div class=**"container"**>

<!-- header section -->

<?php

**require** "php/header.php";

createHeader('home', 'Dashboard', 'Home');

?>

<!-- header section end -->

<!-- form content -->

<div class=**"row"**>

<div class=**"row col col-xs-8 col-sm-8 col-md-8 col-lg-8"**>

<?php

**function** createSection1($location, $title, $table) {

**require** 'php/db\_connection.php';

$query = "SELECT \* FROM **$table**";

**if**($title == "Out of Stock")

$query = "SELECT \* FROM **$table** WHERE QUANTITY = 0";

$result = **mysqli\_query**($con, $query);

$count = **mysqli\_num\_rows**($result);

**if**($title == "Expired") {

// logic

$count = 0;

**while**($row = **mysqli\_fetch\_array**($result)) {

$expiry\_date = $row['EXPIRY\_DATE'];

**if**(**substr**($expiry\_date, 3) < **date**('y'))

$count++;

**else** **if**(**substr**($expiry\_date, 3) == **date**('y')) {

**if**(**substr**($expiry\_date, 0, 2) < **date**('m'))

$count++;

}

}

}

**echo** '

<div class="col-xs-12 col-sm-6 col-md-6 col-lg-4" style="padding: 10px">

<div class="dashboard-stats" onclick="location.href=\''.$location.'\'">

<a class="text-dark text-decoration-none" href="'.$location.'">

<span class="h4">'.$count.'</span>

<span class="h6"><i class="fa fa-play fa-rotate-270 text-warning"></i></span>

<div class="small font-weight-bold">'.$title.'</div>

</a>

</div>

</div>

';

}

createSection1('manage\_customer.php', 'Total Customer', 'customers');

createSection1('manage\_supplier.php', 'Total Supplier', 'suppliers');

createSection1('manage\_medicine.php', 'Total Medicine', 'medicines');

createSection1('manage\_medicine\_stock.php?out\_of\_stock', 'Out of Stock', 'medicines\_stock');

createSection1('manage\_medicine\_stock.php?expired', 'Expired', 'medicines\_stock');

createSection1('manage\_invoice.php', 'Total Invoice', 'invoices');

?>

</div>

<div class=**"col col-xs-4 col-sm-4 col-md-4 col-lg-4"** style=**"padding: 7px 0; margin-left: 15px;"**>

<div class=**"todays-report"**>

<div class=**"h5"**>**Todays Report**</div>

<table class=**"table table-bordered table-striped table-hover"**>

<tbody>

<?php

**require** 'php/db\_connection.php';

**if**($con) {

$date = **date**('Y-m-d');

?>

<tr>

<?php

$total = 0;

$query = "SELECT NET\_TOTAL FROM invoices WHERE INVOICE\_DATE = '**$date**'";

$result = **mysqli\_query**($con, $query);

**while**($row = **mysqli\_fetch\_array**($result))

$total = $total + $row['NET\_TOTAL'];

?>

<th>**Total Sales**</th>

<th class=**"text-success"**>**Rs.** <?php **echo** $total; ?></th>

</tr>

<tr>

<?php

//echo $date;

$total = 0;

$query = "SELECT TOTAL\_AMOUNT FROM purchases WHERE PURCHASE\_DATE = '**$date**'";

$result = **mysqli\_query**($con, $query);

**while**($row = **mysqli\_fetch\_array**($result))

$total = $total + $row['TOTAL\_AMOUNT'];

}

?>

<th>**Total Purchase**</th>

<th class=**"text-danger"**>**Rs.** <?php **echo** $total; ?></th>

</tr>

</tbody>

</table>

</div>

</div>

</div>

<hr style=**"border-top: 2px solid #ff5252;"**>

<div class=**"row"**>

<?php

**function** createSection2($icon, $location, $title) {

**echo** '

<div class="col-xs-12 col-sm-6 col-md-3 col-lg-3" style="padding: 10px;">

<div class="dashboard-stats" style="padding: 30px 15px;" onclick="location.href=\''.$location.'\'">

<div class="text-center">

<span class="h1"><i class="fa fa-'.$icon.' p-2"></i></span>

<div class="h5">'.$title.'</div>

</div>

</div>

</div>

';

}

createSection2('address-card', 'new\_invoice.php', 'Create New Invoice');

createSection2('handshake', 'add\_customer.php', 'Add New Customer');

createSection2('shopping-bag', 'add\_medicine.php', 'Add New Medicine');

createSection2('group', 'add\_supplier.php', 'Add New Supplier');

createSection2('bar-chart', 'add\_purchase.php', 'Add New Purchase');

createSection2('book', 'sales\_report.php', 'Sales Report');

createSection2('book', 'purchase\_report.php', 'Purchase Report');

?>

</div>

<!-- form content end -->

<hr style=**"border-top: 2px solid #ff5252;"**>

</div>

</div>

</body>

</html>

**OUTPUT SCREEN LAYOUTS**

* To Run This Project

To run this project, you must have installed a virtual server i.e [XAMPP](https://www.apachefriends.org/download_success.html) on your PC (for Windows).

After Starting Apache and MySQL in XAMPP, follow the following steps

***1st Step***: Extract file  
**2nd Step:** Copy the main project folder  
**3rd Step:** Paste in xampp/htdocs/

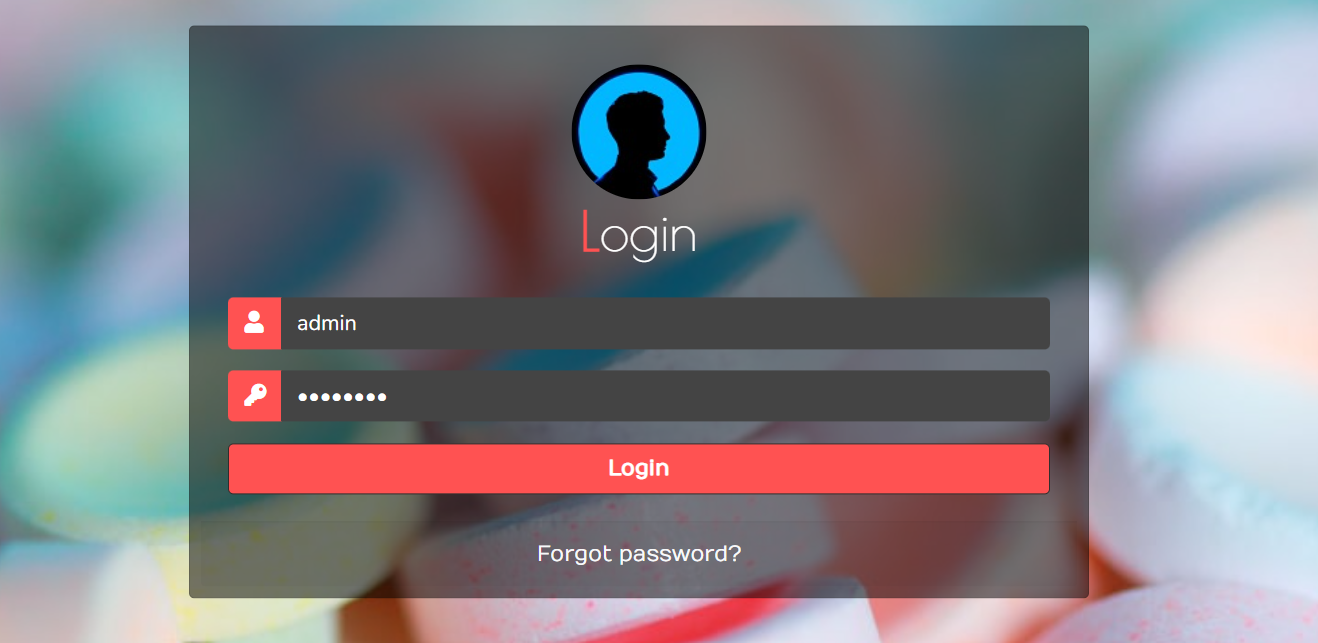
***Now Connecting Database***

**4th Step:** Open a browser and go to URL “http://localhost/phpmyadmin/”  
***5th Step:*** Then, click on the databases tab  
***6th Step:*** Create a database naming “pharmacy” and then click on the import tab  
**7th Step:** Click on browse file and select “pharmacy.sql” file  
**8th Step:** Click on go.

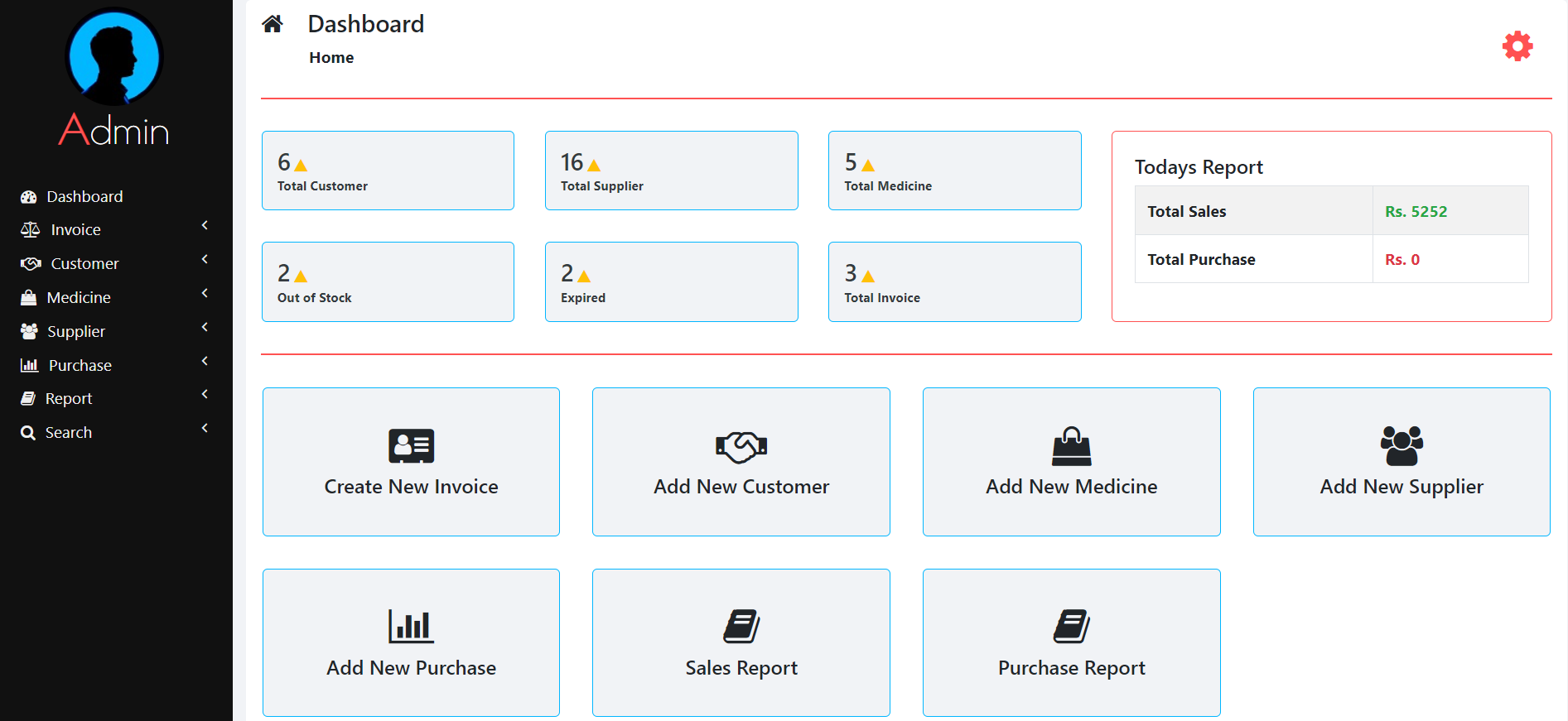
***After Creating Database***,

**9th Step:** Open a browser and go to URL “http://localhost/Pharmacy-Management/”

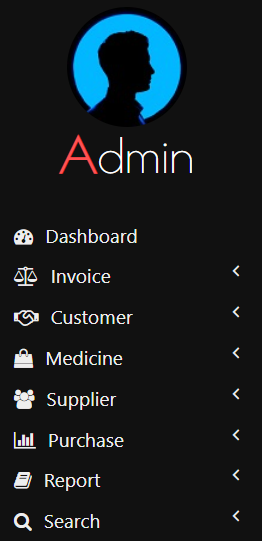
Login Page:



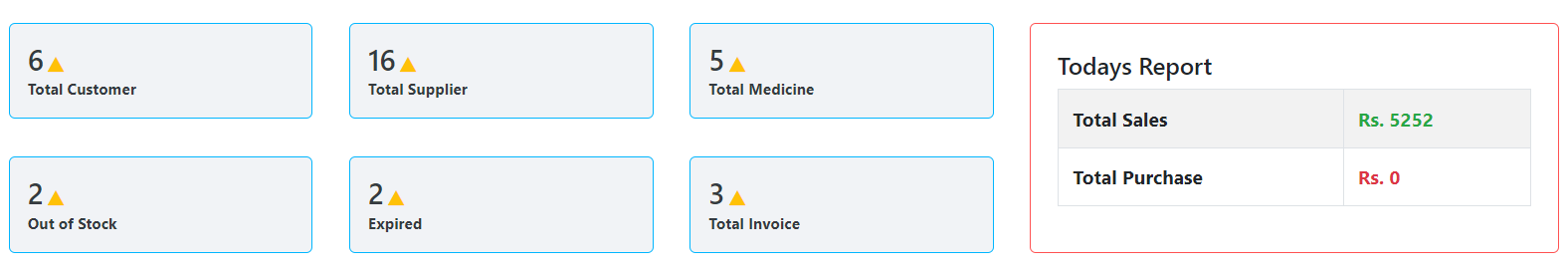
Dashboard /Home Page:



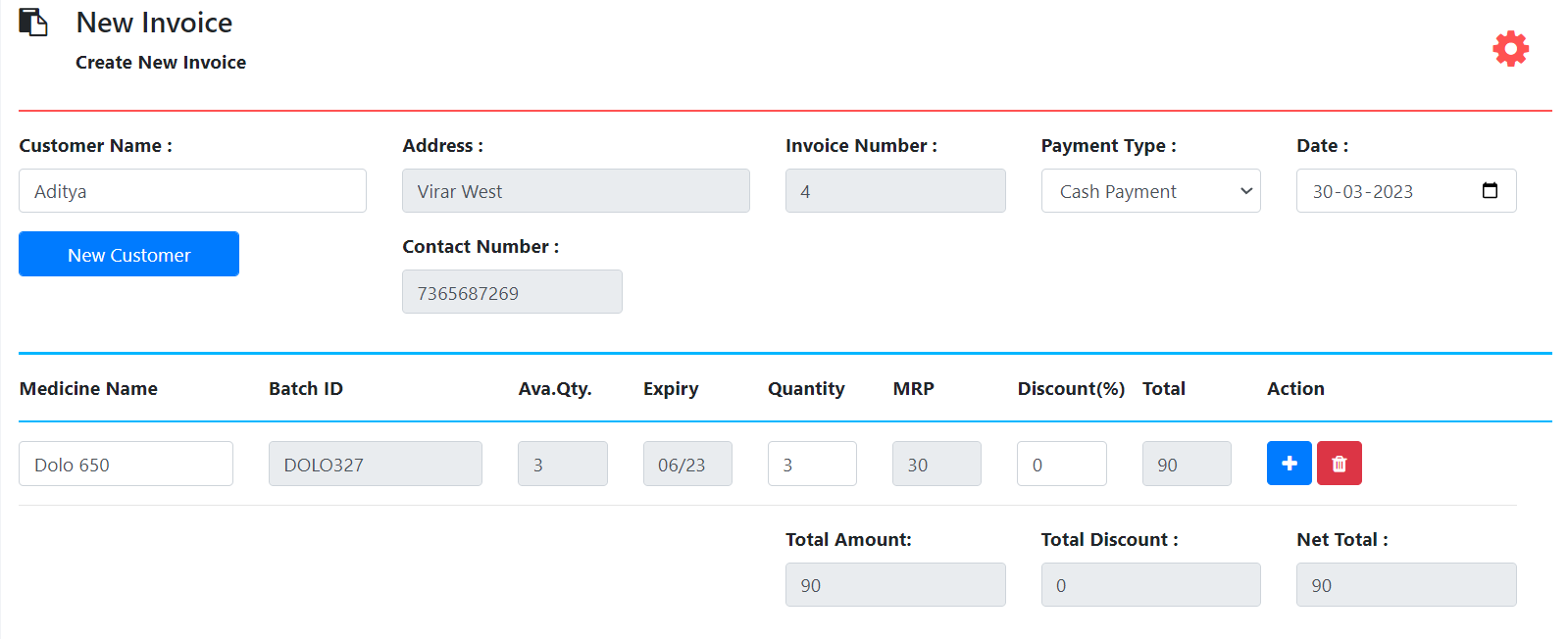
Sidebar Menu:



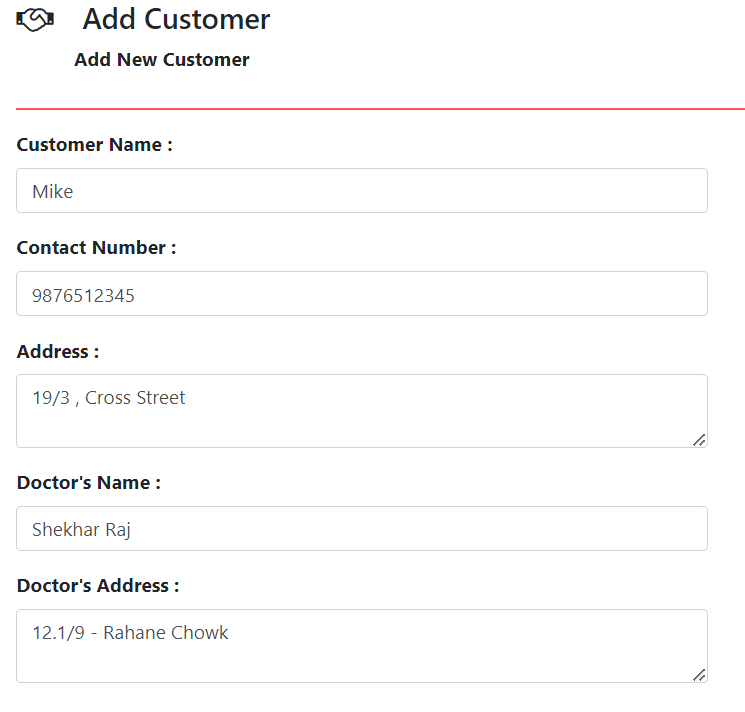
Quick Menu Section:



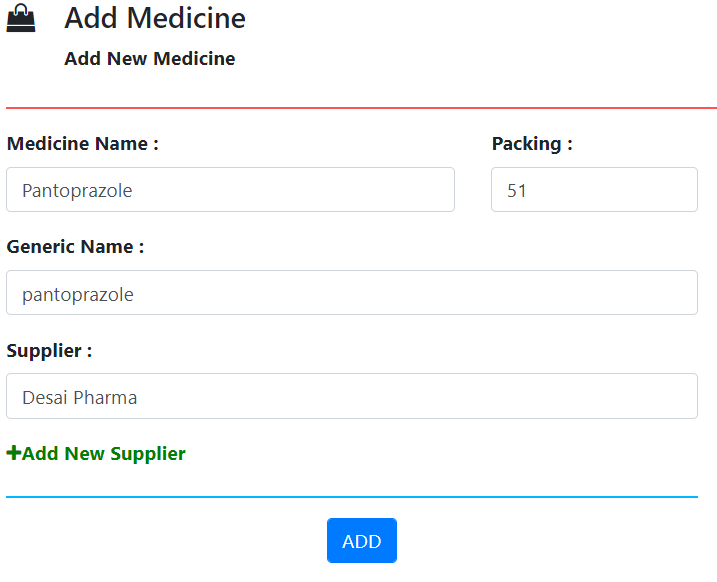
Invoice/Billing Section:



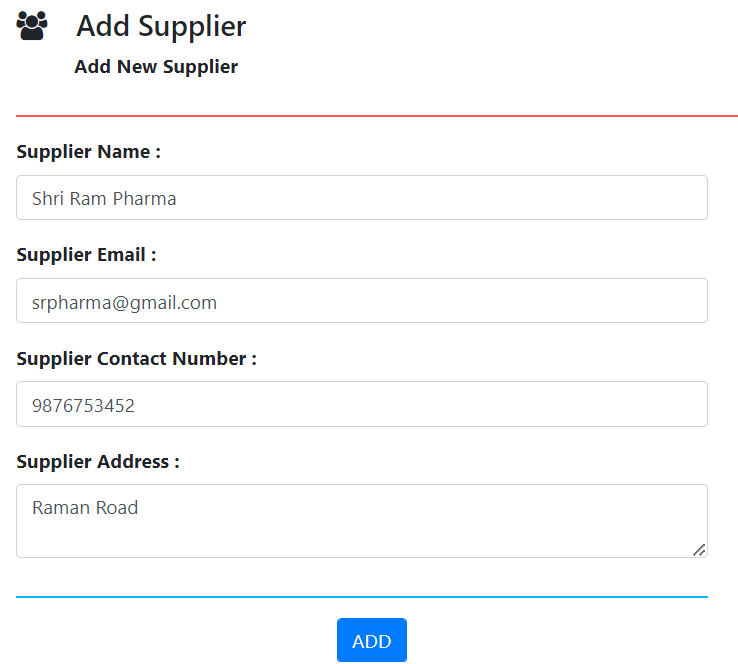
New Customer Adding Section:



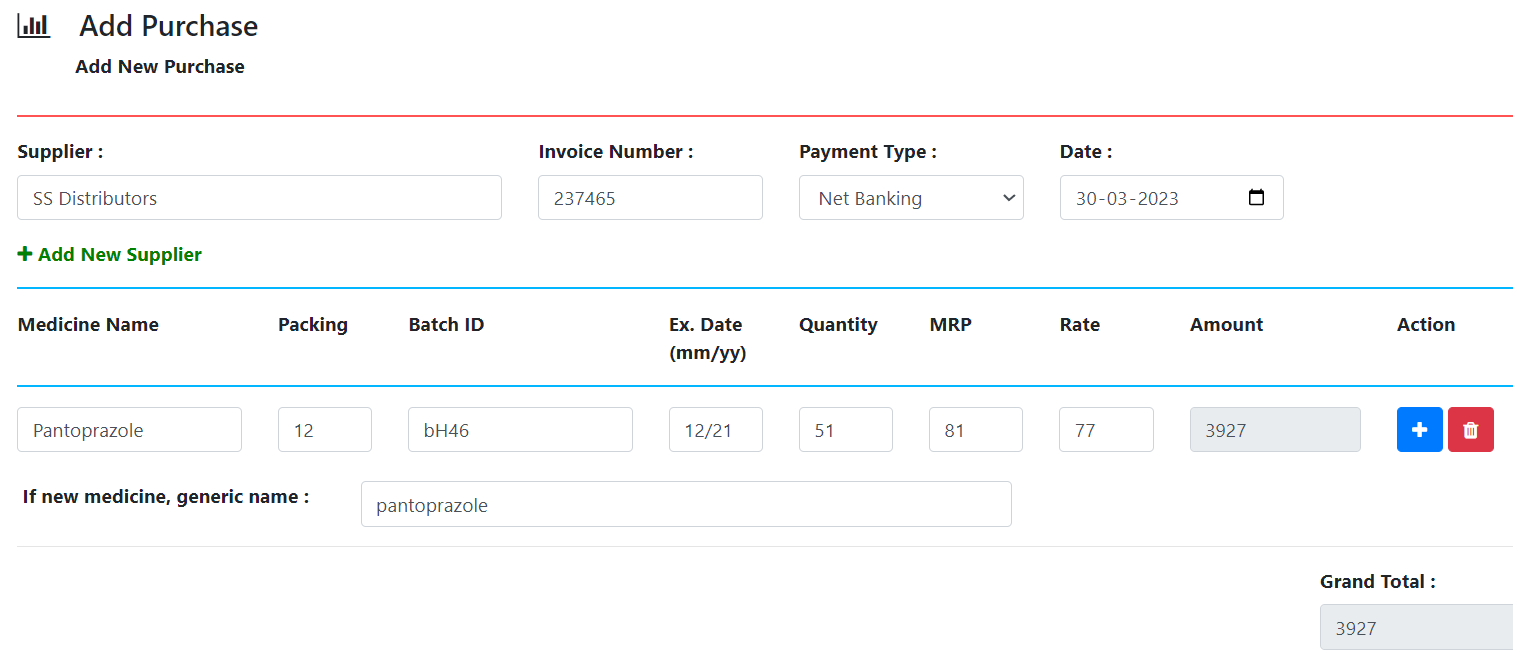
Medicine Section:



Supplier Section:



Purchase Order Section:



**CONCLUSION AND FUTURE WORK**

**Conclusion**

Effective implementation of this software will take care of the basic requirements of the pharmacy management system because it can provide easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the pharmacy management, whenever the idea arises.

The main objective of the project is to manage the administration of the pharmaceutical store and the database. This project is an analysis of the development and implementation of a pharmacy management system. This is done by creating a database of the available medicines in the shop.

The primary purpose of the pharmacy management system is to increase the accuracy and improve the safety and efficiency of the pharmacy store. The goal of this project is to develop software for the effective management of the store. This is a very useful application for the pharmacist, which reduces the work load and it will help you to manage all of the components of the pharmacy, such as Drug Administration, Invoicing, etc., etc. that is, the increase in the efficiency of processing. This will increase the clinical efficiency and patient convenience, since Ethiopia is in the direction of the pharmaceutical care of the patient. It automates tasks, and account management. In a pharmacy, and the bill inspection is an essential process. The pharmacy management system is easy- to use, so that the user can run a pharmacy without ambiguity.

**Future work:**

* We can connect it to a web server and can make it live for everyone.
* We can also design an app for users for better customer service.
* To create a backup mechanism for backing up data and information.
* We will host the platform on online servers, to make it accessible worldwide.

**BIBLIOGRAPHY**

**1. Documentation:** [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)

**2. Repository:** [**https://github.com/**](https://github.com/)

**3. Doubts:** [**https://stackoverflow.com/**](https://stackoverflow.com/)

**4. HTML -** [**https://developer.mozilla.org/en-US/docs/Web/HTML**](https://developer.mozilla.org/en-US/docs/Web/HTML)

**5. HTML -** [**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

**6. HTML -** [**https://en.wikipedia.org/wiki/HTML**](https://en.wikipedia.org/wiki/HTML)

**7. CSS - https://developer.mozilla.org/en-US/docs/Web/CSS**

**8. CSS -https://www.w3schools.com/css/**

**9. CSS -** [**https://en.wikipedia.org/wiki/CSS**](https://en.wikipedia.org/wiki/CSS)

**10. PHP- tutorialspoint.com/php/index.htm**

**11. Pharmacy Management System Source Code Github :**

**https://github.com/imybh/pharmacymanagementsystem**