

ABSTRACT

The Medical Shop Management System (MSMS) is a comprehensive software solution designed to streamline and automate the operations of a medical shop or pharmacy. This system leverages modern technologies to enhance efficiency, accuracy, and customer service within the pharmaceutical retail environment. By digitizing various aspects of the pharmacy's workflow, the MSMS aims to optimize inventory management, sales, and customer interactions.

The main aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer softer, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the client.

TABLE OF CONTENTS

TITLE	PAGE NO.
CHAPTER 1: INTRODUCTION	1-4
CHAPTER 2: REQUIREMENT ANALYSIS	5-19
CHAPTER 3: MODULE	20-21
CHAPTER 4: ER- DIAGRAM	22-23
CHAPTER 5: DATABASE DESIGN	24-26
CHAPTER 6: CODING	27-34
CHAPTER 7: TESTING	35-39
CHAPTER 8: SCREENSHOTS	40-44
CHAPTER 9: ANALYSIS AND DESIGN	45-46
CHAPTER 10: CONCLUSION	47-49
CHAPTER 11: FUTURE ENHANCEENT	50-52
CHAPTER 12: BIBLIOGRAPHY	53-54

CHAPTER 1

INTRODUCTION

INTRODUCTION

- In today's fast-paced world, managing a medical shop efficiently is crucial to provide timely and accurate healthcare services to the community. Introducing the Medical Shop Management System (MSMS), a cutting-edge software solution designed to streamline and enhance the operations of medical retail establishments.
- MSMS is the result of years of dedicated research and development, tailored specifically to meet the unique challenges faced by medical shops in handling inventory, sales, and customer interactions. This revolutionary system harnesses the power of technology to automate mundane tasks, allowing pharmacy owners and staff to focus on what matters most - delivering exceptional care to their customers.

➤ Objective:

- The primary objective of the Medical Shop Management System (MSMS) is to revolutionize the way pharmaceutical retail establishments operate, empowering them with modern technology to enhance efficiency, accuracy, and customer service. The system aims to achieve the following key objectives.
- It revolutionize pharmaceutical retail management by leveraging technology to optimize operations, enhance customer service, ensure compliance, and provide data-driven insights for better decision-making. MSMS empowers medical shops to thrive in the competitive healthcare landscape while making a positive impact on the well-being of their customers.

➤ Functionalities:

- Maintain, track, and automatically reorder pharmaceutical products to prevent stock outs and overstocking.
- The system provides insightful reporting and analytics for informed decision-making.
- Additionally, MSMS monitors product expiries, implements batch tracking, and offers a user-friendly interface for easy adoption. Regular data backup and recovery mechanisms safeguard information, while multi-store support facilitates the management of chain pharmacies efficiently.

➤ **Scope:**

The scope for a Medical Shop Management System is extensive, promising to transform pharmaceutical retail operations by automating processes, enhancing customer service, ensuring compliance, and providing valuable insights for business growth and success.

➤ **Existing System:**

- Content Management Systems (CMS): These systems allow for the creation and management of website content without the need for extensive coding knowledge. Popular CMS include WordPress, Drupal, and Joomla
- Server-side scripting languages: These languages allow for the creation of dynamic websites by generating HTML on the server-side, based on user requests. Examples of server-side scripting languages include PHP, Python, and Ruby.
- JavaScript frameworks: These are libraries of pre-written code that can be used to build complex dynamic websites. Examples of JavaScript frameworks include Angular, React.

➤ **Proposed System:**

The proposed Medical Shop Management System (MSMS) is a comprehensive software solution designed to revolutionize pharmaceutical retail operations, providing an efficient, secure, and user-friendly platform for medical shop owners and staff. The system encompasses a wide range of functionalities to streamline processes, enhance customer service, and enable data-driven decision-making. Here are the key components of the proposed system:

- User Authentication and Access Control: The system will feature robust user authentication mechanisms to ensure that only authorized personnel can access sensitive data and perform specific actions. Different user roles will be defined, granting appropriate access levels for owners, pharmacists, and other staff.
- Inventory Management: MSMS will maintain a centralized database of pharmaceutical products with details such as name, batch number, expiry date, supplier information, and cost price. Real-time tracking of stock levels will be implemented, triggering automated reorder alerts to prevent stockouts.
- Sales and Billing: The system will facilitate quick and accurate billing of customers, supporting multiple payment options and generating itemized invoices.

Integration with the Point of Sale (POS) system will ensure seamless transaction processing.

- **Prescription Management:** MSMS will securely store and retrieve prescription records, enabling easy refills and providing pharmacists with a comprehensive view of the customer's medication history.
- **Customer Relationship Management (CRM):** The system will include a CRM module to manage customer profiles, track purchase history, and offer personalized services. Automated reminders for prescription refills and health check-ups will enhance customer engagement.
- **Reporting and Analytics:** The system will provide comprehensive reports and data analytics, offering insights into sales trends, popular products, and financial performance. Visual representations such as graphs and charts will aid in data interpretation.
- **Expiry Management:** The system will monitor and manage product expiry dates, generating alerts for items approaching expiration to facilitate timely actions.
- **Scalability and Cloud Integration:** The system will be scalable to accommodate the needs of different-sized medical shops. Cloud integration will enable remote accessibility and data storage, enhancing flexibility and convenience.

CHAPTER 2

REQUIREMENT ANALYSIS

REQUIREMENT ANALYSIS

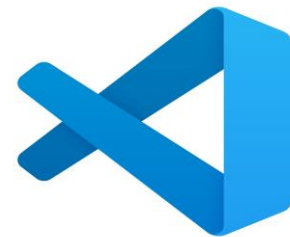
➤ **Problem definition:**

- In this section, we define problems. The problem is that, when results are to be declared, lot of money, physical space, man power is required. And there are so many candidates who needs to get their result, but due to medical illness or some emergency they are not able to visit the institute. Another problem is that, it depends on physical presence of the candidates to get their result.

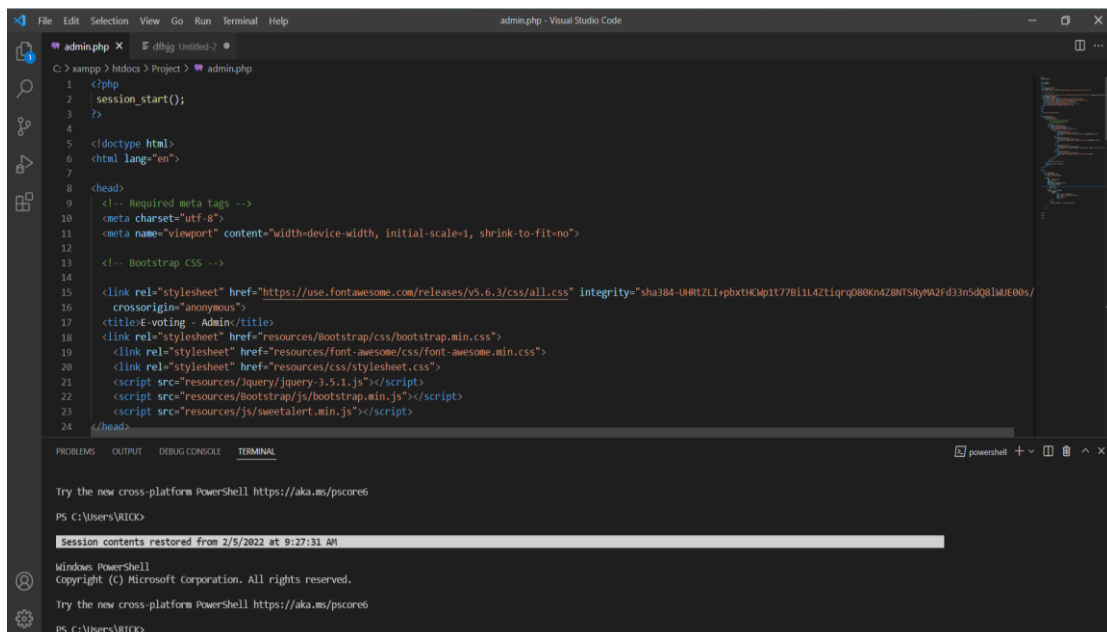
➤ **Applications requirements:**

1. Visual Studio Code:

- Visual Studio Code is an integrated development environment made by Microsoft for Windows, Linux and MACOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware.



- In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter. On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.



2. XAMPP:

- The full form of XAMPP is (X) stands for Cross-platform, (A) Apache server, (M) Maria DB, (P) PHP and (P) Perl. XAMPP is an open source free software developed by Apache friends. XAMPP software package contains Apache distributions for Apache server, Maria DB, PHP, and Perl. And it is basically a local host or a local server. This local server works on your own desktop or laptop computer. You can just install this software on your laptop or desktop and test the clients or your website before uploading it to the remote web server or computer. This XAMPP server software gives you suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer.
- The Cross-platform usually means that it can run on any computer with any operating system. Next Maria DB is the most famous database server and it is developed by MYSQL team. PHP usually provides a space for web development. PHP is a server-side scripting language. And the last Perl is a programming language and is used to develop a web application.

➤ **Main Tools of XAMPP and its Definition:**

- XAMPP contains tools such as Apache, MYSQL, PHP, and Perl. We will see these tools.

➤ **Apache:**

- Apache server is an open source free software which is initially developed by a group of software developers and now it is maintained by Apache software foundation. Apache HTTP is a remote server (computer) if someone request files, images or documents using their browser they will serve those files to clients using HTTP servers. Mainly hosting companies use this application to create a VPS server and shared hosting for their clients.

➤ **MYSQL:**

- MYSQL is open-source software. It is actually a relational database management system (RDBMS). This SQL stands for Structured Query Language. It is the most popular and best RDBMS used for developing a variety of web-based software applications. With the help of MYSQL, it is possible to organize the information, manage, retrieve and update the data whenever you wish to do.

➤ **PHP:**

The full form of PHP is Hypertext Preprocessor. It is a server-side scripting language that helps you to create dynamic websites. This language is mainly used to build web-based software applications. It is open-source software and works fine with MYSQL. What actually happens is, the PHP code will be executed on the server and at the browser side its HTML code will be displayed.

• **Programming Technology Requirements:**

HTML:

- HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript). "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.
- HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, , , and many others.
- An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case

insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

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 as well as enabling 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/css is 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.

JavaScript:

- JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions.
- Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client side page behavior and all major web browsers have a dedicated JavaScript engine to execute it.
- As a multi-paradigm language, JavaScript supports event driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.
- JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.
- Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

Bootstrap:

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
- Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is

to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

- Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an autocomplete function for input fields.
- Example of a webpage using Bootstrap framework rendered in Firefox. The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:
 - Smaller than 576 pixels
 - 576–768 pixels
 - 768–992 pixels
 - 992–1200 pixels
 - Larger than 1200 pixels
- Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.
- A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project. The raw form of Bootstrap,

however, enables developers to implement further customization and size optimizations. This raw form is modular, meaning that the developer can remove unneeded components, apply a theme and modify the uncompiled Sass files.

jQuery:

- JQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. [3] It –is free, open source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having at least 3 to 4 times more usage than any other JavaScript library.
- JQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, them able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications
- The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API. Later, this style has been enhanced with a deeper algorithm-data fusion in an heir of jQuery, the D3.js framework.
- Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

Ajax:

- Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from

the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page.[3] In practice, modern implementations commonly utilize JSON instead of XML.

- Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The webpage can then be modified by JavaScript to dynamically display—and allow the user to interact with—the new information. The built-in XMLHttpRequest object, or since 2017 the new "fetch ()" function within JavaScript, is commonly used to execute Ajax on webpages, allowing websites to load content onto the screen without refreshing the page. Ajax is not a new technology, or different language, just existing technologies used in new ways.
- The term Ajax has come to represent a broad group of Web technologies that can be used to implement a Web application that communicates with a server in the background, without interfering with the current state of the page. In the article that coined the term Ajax,[1][3] Jesse James Garrett explained that the following technologies are incorporated:
 1. HTML (or XHTML) and CSS for presentation
 2. The Document Object Model (DOM) for dynamic display of and interaction with data
 3. JSON or XML for the interchange of data, and XSLT for XML manipulation
 4. The XMLHttpRequest object for asynchronous communication
 5. JavaScript to bring these technologies together
- Since then, however, there have been a number of developments in the technologies used in an Ajax application, and in the definition of the term Ajax itself. XML is no longer required for data interchange and, therefore, XSLT is no longer required for the manipulation of data. JavaScript Object Notation (JSON) is often used as an alternative format for data interchange, although other formats such as preformatted HTML or plain text can also be used. A variety of popular JavaScript libraries, including JQuery, include abstractions to assist in executing Ajax requests.

PHP:

- Hypertext Preprocessor is a server side scripting language— designed for web development and also used as a general-purpose programming language. It was originally created by

Ramus Leadoff in 1994. The PHP reference is now produced by the PHP group. PHP originally stood for personal home page. But now it stands for recursive initialism PHP hypertext preprocessor.

- PHP code may be embedded into html code. It can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a common gateway interface executable. The web server combines the results of interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command line interface and can be used to implement standalone graphical applications.
- The standard PHP interpreter, powered by the Zend engine, is free software released under the PHP license. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.
- The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard.

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code.
- PHP code are executed on the server, and the result is returned to the browser as plain HTML.
- PHP files have extension ".PHP".

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access

- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.PHP.net
- PHP is easy to learn and runs efficiently on the server side
- PHP is an amazing and popular language!
- It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!
- It is deep enough to run the largest social network (Facebook)!
- It is also easy enough to be a beginner's first server-side language!

Advantages of PHP:

The reason behind the popularity of PHP is its several advantages. PHP is most suited for the purpose of web development. The advantages of PHP are discussed briefly below:

1. Cross Platform:

- All the PHP based applications can run on various types of platforms. PHP is supported by majority of Operating Systems, some of which includes Solaris, UNIX, Windows and Linux. The mentioned platforms can be used to write codes in PHP and also view web pages or run the PHP based applications.
- PHP easily interfaces with MySQL and Apache both. An effortless integration of PHP can be done with various other technologies like Java and there is no requirement of redevelopment. Therefore, saving both time and money, giving it an important advantage.

2. Easy database connection:

- A programming language like PHP is widely used on the internet and needs to connect to the database very often. Therefore, having a feature that could help PHP to connect to database easily is mandatory. Several websites such as the ecommerce websites require good database management system.
- PHP has a built-in module that helps it in connecting with database easily. Therefore, PHP has a great demand in the field of web development where a data driven website needs to be developed. PHP significantly reduces the time needed in developing the web application that needs an efficient database management system.

3. Easy to use:

- PHP is widely used because it is easy to use. In contrast with other programming languages that are complex, PHP is simple, fluent, clean and organized; hence it is a boon for the new users. PHP has a well-organized syntax which is logical at the same time.
- PHP does not require any intensive studying or manual to use it. Command functions of PHP are easily understood as the user can easily figure out from the name of the commands itself what it does. A person who is new to PHP can still code because the syntax is somewhat similar to C.
- A person who is new to PHP can still code because the syntax is somewhat similar to C. Hence, if a person who knows C can easily code in PHP. Hence, it is easier to create and optimize the application using PHP.

4. Speed:

- Speed is the primary need of web development. There are people who face the challenge of slow internet connection and slow data speed. Furthermore, a fast-loading website is always preferred by people across the globe. When compared to other programming languages, PHP is found to be the fastest programming language.
- In normal circumstances, it takes a lot of time to connect to the database, when you attempt to fetch certain data from the database. It takes a lot of time in connecting to the database, then executing the statement and finally getting the data. PHP performs these set

of tasks faster than other scripting languages. PHP is faster in both connecting to the database and in using other important applications.

- The high speed of PHP gives it an advantage over other scripting languages and gives it an application in important administrations such as the server administration and mail functionalities.

5. Open source:

- One of the important advantages of PHP is that it is Open-Source. Therefore, PHP is readily available and is entirely free. In contrast to other scripting languages used for web development which requires the user to pay for the support files, PHP is open to everyone, anytime and anywhere.
- A beginner in PHP need not worry about the support as PHP is maintained and developed by a large group of PHP developers which helps in creating support community of PHP that helps people in PHP implementation and manipulation.

- **System minimum requirements:**

Hardware System Configuration: -

- Processor - Dual Core/ Pentium/ i3 Intel Processor
- RAM - 2 GB
- Hard Disk - 512MB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

Software System Configuration: -

- Operating System - Windows XP/7/8/10
- Server-side Script – PHP
- Database - MySQL 6.0

Research Methodology:

- A research methodology is the elaboration of a clear strategy for gathering evidence, including the specific data collection methods to be used, the kinds of evidence to be collected, and the approach for analyzing the evidence (Darian-Smith & McCarthy, 2017). It is the path to solve a research problem. Hence it must be planned according to the objectives of the study.

Data Analysis:

- The classification and tabulation transfer the raw data collected into useful information by organizing and compiling the bits of data into graphically understandable manner, and in the current research it was done with the help of a UML (Unified Modelling Language) modelling tool.

System Analysis:

- Systems development is mainly done in two phases namely, system analysis and design. And this chapter focuses on analyzing the research data and describing a logical view of the whole process, by modelling the data analyzed in the form of diagrams to visualize the design and specifications of the system in an object-oriented manner. The analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used. During this phase, the research team investigates any current system(s), identifies opportunities for improvement, and develops a concept for the new system.

System Design:

- The purpose of the analysis phase is to figure out what the business needs and right after system analysis, started system design whose purpose is to decide how to build the same system. And according to Dennis et al system design is the determination of the overall system architecture consisting of a set of physical processing components, hardware, software, people, and the communication among them, that will satisfy the system's essential requirements. During the initial part of design, the business requirements for the system are converted into system requirements that describe the technical details for building the system.

CHAPTER 3

MODULE

MODULE

The entire project mainly consists of three modules:

1. Log In
 - ✓ Admin Login
 - ✓ Customer Login
2. Report Management
 - ✓ Admin Access to all
3. User Profile
 - ✓ All users have separate one
4. Manage Products
5. Customer Registration
6. Logout

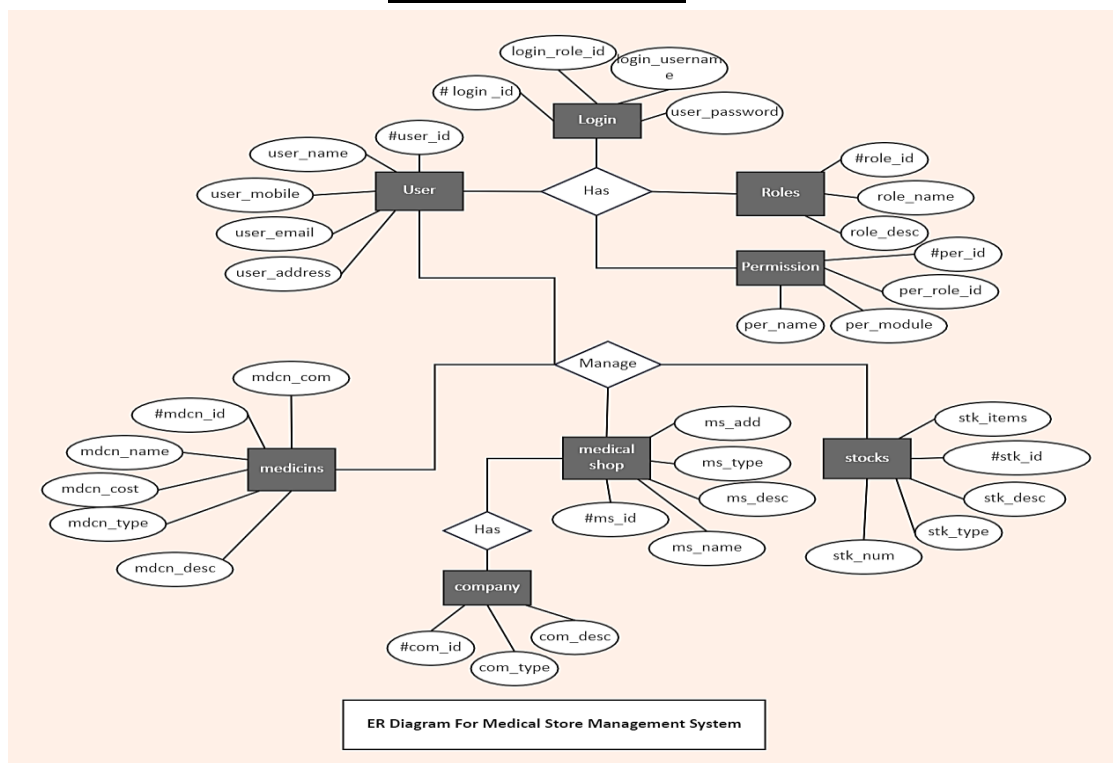
CHAPTER 4

ER- DIAGRAM

Entity Relationship Diagram:

- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.
- ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

ER- DIAGRAM



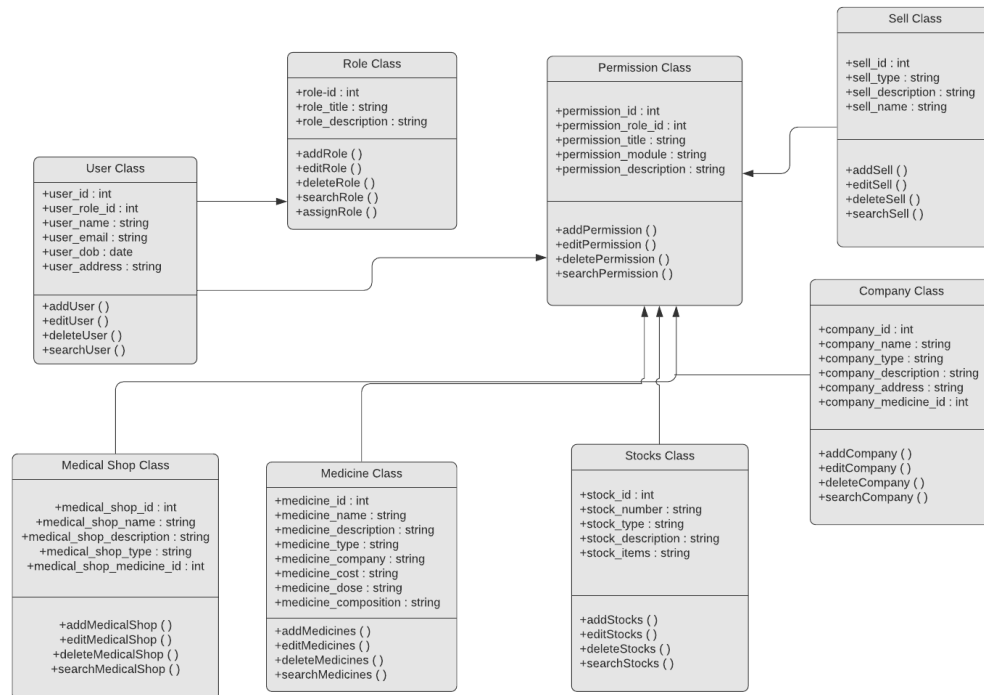
CHAPTER 5

DATABASE DESIGN

Database Design:

- From the analysis phase, a logical model of the system was created. And the following particular activity describes the proposed system's data organization, storage and management. Which is to be managed by the DBMS a system responsible for storing, retrieving and protecting the data. These issues are important as they affect the consistency and quality of the data. The data is structured in files or tables that interact in various ways. Each table contains data about students, professors, users and subjects. According to Dennis, relational database is the most popular kind of database for application development today. And it is based on collections of tables with each table having a primary key, which is a field or fields whose values are unique for every row of the table and are used to identify each row or record. The tables are related to one another by placing the primary key from one table into the related table as a foreign key. Most relational database management systems (RDBMS) support referential integrity, or the idea of ensuring that values linking the tables together through the primary and foreign keys are valid and correctly synchronized.

• Design



CHAPTER 6

CODING

SAMPLE CODE

Index.php:

```

<?php require_once('config.php'); ?>
<!DOCTYPE html>
<html lang="en">
<?php require_once('inc/header.php') ?>
<?php if($_settings->chk_flashdata('success')): ?>
<script>
$(function(){
alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success')
})
</script>
<?php endif;?>
<body>
<?php require_once('inc/topBarNav.php') ?>
<?php $page = isset($_GET['p']) ? $_GET['p'] : 'home'; ?>
<?php
if(!file_exists($page.".php") && !is_dir($page)){
include '404.html';
}else{
if(is_dir($page))
include $page.'/index.php';
else
include $page.'.php';
}
?>
<?php require_once('inc/footer.php') ?>

<div class="modal fade" id="uni_modal" role='dialog'>
<div class="modal-dialog rounded-0 modal-md modal-dialog-centered" role="document">
<div class="modal-content rounded-0">
<div class="modal-header">
<h5 class="modal-title"></h5>
</div>
<div class="modal-body">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-primary" id='submit' onclick="$('#uni_modal
form').submit()">Save</button>
<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
</div>
</div>
</div>
</div>
<div class="modal fade" id="uni_modal_right" role='dialog'>
<div class="modal-dialog rounded-0 modal-full-height modal-md" role="document">
<div class="modal-content rounded-0">

```

```

<div class="modal-header">
<h5 class="modal-title"></h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span class="fa fa-arrow-right"></span>
</button>
</div>
<div class="modal-body">
</div>
</div>
</div>
</div>
<div class="modal fade" id="viewer_modal" role='dialog'>
<div class="modal-dialog modal-md" role="document">
<div class="modal-content">
<button type="button" class="btn-close" data-dismiss="modal"><span class="fa fa-
times"></span></button>
<img src="" alt="">
</div>
</div>
</div>
<div class="modal fade" id="confirm_modal" role='dialog'>
<div class="modal-dialog modal-md modal-dialog-centered" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Confirmation</h5>
</div>
<div class="modal-body">
<div id="delete_content"></div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-primary" id='confirm' onclick="">Continue</button>
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
</body>
</html>

```

Register.php:

```

<?php require_once('./config.php') ?>
<!DOCTYPE html>
<html lang="en" class="" style="height: auto;">
<?php require_once('inc/header.php') ?>
<script>
    start_loader()
</script>
<style>

```

```

html, body{
  width:100%;
  height:100% !important;
}
body{
  background-image: url("<?php echo validate_image($_settings->info('cover')) ?>");
  background-size:cover;
  background-repeat:no-repeat;
  backdrop-filter: contrast(1);
  overflow-x:hidden
}
#page-title{
  text-shadow: 6px 4px 7px black;
  font-size: 3.5em;
  color: #fff4f4 !important;
  background: #8080801c;
}
img#cimg{
  height: 5em;
  width: 5em;
  object-fit: cover;
  border-radius: 100% 100%;
}
</style>
<body class="">
<div class="d-flex flex-column align-items-center justify-content-center h-100 w-100">
<h1 class="text-center text-white px-4 py-5" id="page-title"><b><?php echo $_settings-
>info('name') ?></b></h1>
<div class="col-lg-6 col-md-8 col-sm-12 col-xs-12">
<!-- /.login-logo -->
<div class="card card-navy my-2 rounded-0">
  <div class="card-header rounded-0">
    <h4 class="card-title">Registration</h4>
  </div>
  <div class="card-body rounded-0">
    <form id="register-form" action="" method="post">
      <input type="hidden" name="id">
      <div class="row">
        <div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
          <div class="form-group">
            <label for="firstname" class="control-label">First Name</label>
            <input type="text" class="form-control form-control-sm rounded-0" required=""
name="firstname" id="firstname">
          </div>
          <div class="form-group">
            <label for="middlename" class="control-label">Middle Name</label>
            <input type="text" class="form-control form-control-sm rounded-0"
name="middlename" id="middlename">
          </div>
          <div class="form-group">
            <label for="lastname" class="control-label">Last Name</label>

```



```

        <input type="text" class="form-control form-control-sm rounded-0" required=""
name="lastname" id="lastname">
    </div>
    <div class="form-group">
        <label for="gender" class="control-label">Gender</label>
        <select class="custom-select custom-select-sm rounded-0" required="" name="gender"
id="gender">
            <option>Male</option>
            <option>Female</option>
        </select>
    </div>
</div>
<div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
    <div class="form-group">
        <label for="email" class="control-label">Email</label>
        <input type="text" class="form-control form-control-sm rounded-0" required=""
name="email" id="email">
    </div>
    <div class="form-group">
        <label for="contact" class="control-label">Contact</label>
        <input type="text" class="form-control form-control-sm rounded-0" required=""
name="contact" id="contact">
    </div>
    <div class="form-group">
        <label for="password" class="control-label">Password</label>
        <div class="input-group input-group-sm">
            <input type="password" class="form-control form-control-sm rounded-0"
required="" name="password" id="password">
            <button tabindex="-1" class="btn btn-outline-secondary btn-sm rounded-0
pass_view" type="button"><i class="fa fa-eye-slash"></i></button>
        </div>
    </div>
    <div class="form-group">
        <label for="cpassword" class="control-label">Confirm Password</label>
        <div class="input-group input-group-sm">
            <input type="password" class="form-control form-control-sm rounded-0"
required="" id="cpassword">
            <button tabindex="-1" class="btn btn-outline-secondary btn-sm rounded-0
pass_view" type="button"><i class="fa fa-eye-slash"></i></button>
        </div>
    </div>
</div>
<div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
    <div class="form-group">
        <label for="" class="control-label">Avatar</label>
        <div class="custom-file">
            <input type="file" class="custom-file-input rounded-0" id="customFile" name="img"
onchange="displayImg(this,$(this))" accept="image/png, image/jpeg">
            <label class="custom-file-label rounded-0" for="customFile">Choose file</label>
        </div>
    </div>
</div>

```

```

        </div>
        <div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
            <div class="form-group d-flex justify-content-center">
                
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-8">
            <a href="/login.php">Already hava an Account</a>
        </div>
        <!-- /.col -->
        <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block">Create Account</button>
        </div>
        <!-- /.col -->
    </div>
</form>
<!-- /.social-auth-links -->

<!-- <p class="mb-1">
    <a href="forgot-password.html">I forgot my password</a>
</p> -->

</div>
<!-- /.card-body -->
</div>
<!-- /.card -->
</div>
<!-- jQuery -->
<script src="<?= base_url ?>plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="<?= base_url ?>plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="<?= base_url ?>dist/js/adminlte.min.js"></script>

<script>
function displayImg(input,_this) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            $('#cimg').attr('src', e.target.result);
        }

        reader.readAsDataURL(input.files[0]);
    }else{
        $('#cimg').attr('src', "<?php echo validate_image('') ?>");
    }
}

```

```

}
$(document).ready(function(){
    end_loader();
    $('.pass_view').click(function(){
        var input = $(this).siblings('input')
        var type = input.attr('type')
        if(type == 'password'){
            $(this).html('<i class="fa fa-eye"></i>')
            input.attr('type','text').focus()
        }else{
            $(this).html('<i class="fa fa-eye-slash"></i>')
            input.attr('type','password').focus()
        }
    })
    $('#register-form').submit(function(e){
        e.preventDefault()
        var _this = $(this)
        var el = $('<div>')
        el.addClass('alert alert-danger err_msg')
        el.hide()
        $('.err_msg').remove()
        if($('#password').val() != $('#cpassword').val()){
            el.text('Password does not match')
            _this.prepend(el)
            el.show('slow')
            $('html, body').scrollTop(0)
            return false;
        }
        if(_this[0].checkValidity() == false){
            _this[0].reportValidity();
            return false;
        }
        start_loader()
        $.ajax({
            url:_base_url_+"classes/Users.php?f=registration",
            method:'POST',
            type:'POST',
            data:new FormData($(this)[0]),
            dataType:'json',
            cache:false,
            processData:false,
            contentType: false,
            error:err=>{
                console.log(err)
                alert('An error occurred')
                end_loader()
            },
            success:function(resp){
                if(resp.status == 'success'){
                    location.href = ('./')
                }else if (!!resp.msg){

```

```
        el.html(resp.msg)
        el.show('slow')
        _this.prepend(el)
        $('html, body').scrollTop(0)
    }else{
        alert('An error occurred')
        console.log(resp)
    }
    end_loader()
}
})
})
</script>
</body>
</html>
```

CHAPTER 7

TESTING

TESTING

Software Testing:

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

Alpha Testing:

Alpha testing is a type of testing that is done on an application towards the end of a development process when the product is almost in a usable state. This type of testing does not involve functional testing on the application. Instead, it is a user testing on the application in order to understand the user behavior and experience on the application. Normally this test is performed by test engineers, employees and sometimes friends / family members with the aim of trying to emulate around 80% of the customers. While these users test and give their feedback, the development team observes the behavior to check for design issues in the application. Alpha testing is mainly conducted to unveil bugs that might arise due to abrupt errors created by the users, validate the quality state of the software in minimal time and finally propound a build that procures the specifications required. Once this test is executed properly, the software is ready for the next stage, i.e., the beta test. Alpha testing has two phases.

Black Box Testing:

Black box testing refers to any type of software test that examines an application without knowledge of the internal design, structure, or implementation of the software project. Black box testing can be performed at multiple levels, including unit testing, integration testing, system testing, or acceptance testing.

I. Unit Testing: -

- Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

II. Integration Testing: -

- All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

VALIDATION TESTING

- This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

SYSTEM TESTING

- This testing is a series of different tests whose primary is to fully exercise the computer-based system.
- This involves, implementing the system in a simulated production environment and testing it.

White Box Testing:

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, open box testing, transparent box testing, Code-based testing and Glass box testing.

Beta Testing:

Beta testing is one of the final steps in your software development lifecycle (SDLC) before a product goes live. Also referred to as user testing or customer validation, beta testing aims to ensure that end users are satisfied with a software product before you make it generally available (GA). While beta tests want to catch any software bugs and errors that have snuck through the testing process, it is more about understanding and improving the product's full end user experience before it becomes GA. That means thoroughly investigating the experience flow and understanding any pain points that will hinder enjoyment of the experience for your end user.

Positive Test Cases:

- The positive flow of the functionality must be considered
- Valid inputs must be used for testing
- Must have the positive perception to verify whether the requirements are justified.

Negative Test Cases:

- Must have negative perception
- Invalid inputs must be used for test.

Feasibility Study:

- The next step in analysis is to verify the feasibility of the proposed system. "All projects are feasible given unlimited resources and infinite time ". But in reality, both resources and time are scarce. Project should confirm to time bounce and should be optimal in their consumption of resources. This place a constant is approval of any project.

- Feasibility has applied to the following areas:

Technical feasibility

Operational feasibility

Economic feasibility

TECHNICAL FEASIBILITY:

- To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

OPERATIONAL FEASIBILITY:

- To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also, the system is very friendly and to use.

ECONOMIC FEASIBILITY:

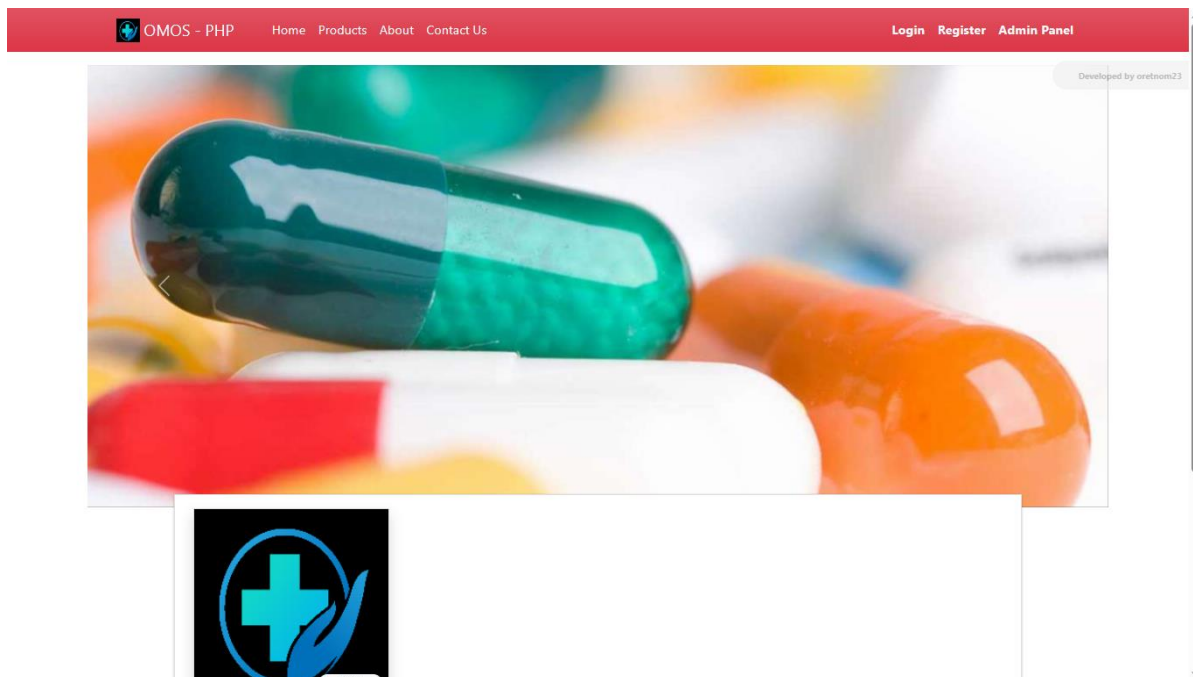
To decide whether a project is economically feasible, we have to consider various factors as:

- Cost benefit analysis
- Long-term returns
- Maintenance costs

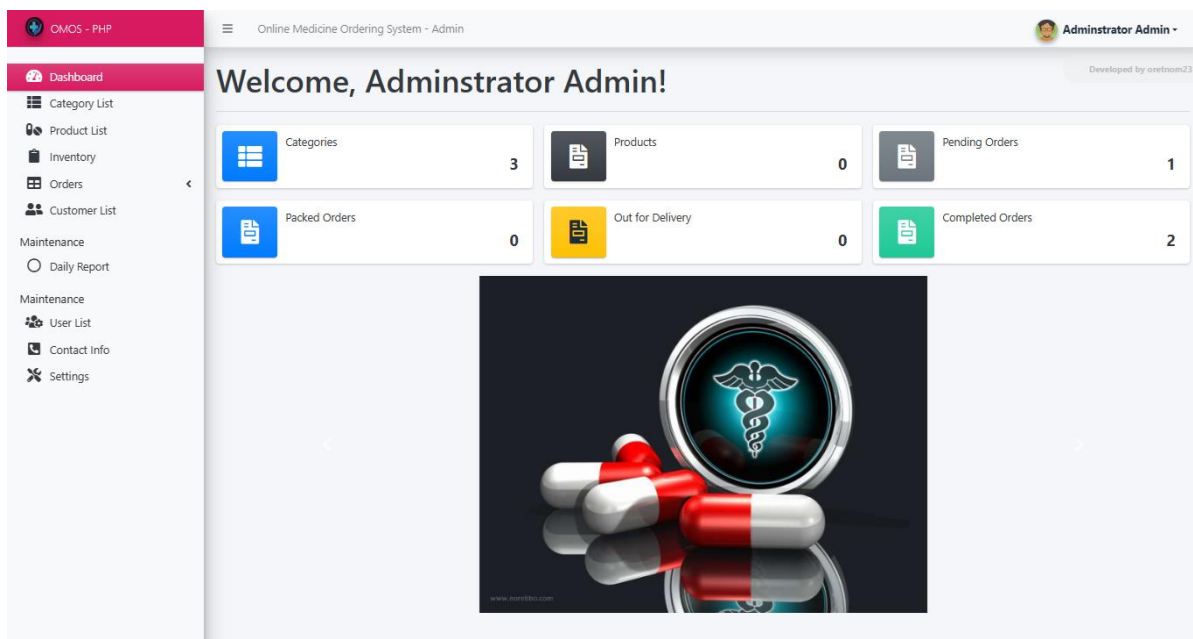
CHAPTER 8

SCREENSHOTS

Login Page:



Admin Dashboard:



Product's Page:

OMOS - PHP

Dashboard

Category List

Product List

Inventory

Orders

Customer List

Maintenance

Daily Report

Maintenance

User List

Contact Info

Settings

Online Medicine Ordering System - Admin

Administrator Admin






List of Products

Developed by oretnom23

+ Create New

Show 10 entries

Search:

#	Date Created	Image	Brand	Name	Price	Status	Action
1	2022-05-25 11:10		Brand 101	Amoxicillin 250	7.00	Active	Action
2	2022-05-25 11:06		Brand 101	Mefenamic 500mg	10.00	Active	Action
3	2022-05-25 11:11		Brand 102	Drug 101 25mg	25.00	Active	Action
4	2022-05-25 14:18		Brand 103	Drug 102 50mg	20.00	Active	Action
5	2022-05-26 10:59		Brand 103	Medicine #101 280ml	375.00	Active	Action

Showing 1 to 5 of 5 entries

Previous 1 Next

Order Page:

OMOS - PHP

Dashboard

Category List

Product List

Inventory

Orders

Customer List

Maintenance

Daily Report

Maintenance

User List

Contact Info

Settings

Online Medicine Ordering System - Admin

Administrator Admin

List of Completed Order

Developed by oretnom23

Show 10 entries

Search:

#	Date Ordered	Code	Customer	Total Amount	Status	Action
1	2023-07-30 23:37	2023073100001	shaw shaw	20.00	Paid	View
2	2022-05-26 10:56	2022052600002	Samantha Jane C Miller	105.00	Paid	View
3	2022-05-26 09:22	2022052600001	Mark D Cooper	50.00	Paid	View
4	2022-05-25 16:36	20212165468	Mark D Cooper	203.00	Paid	View

Showing 1 to 4 of 4 entries

Previous 1 Next

Copyright © 2023. All rights reserved.

OMOS - PHP (by: oretnom23) v1.0

inventory:

OMOS - PHP

Online Medicine Ordering System - Admin

Administrator Admin

Dashboard

Category List

Product List

Inventory

Orders

Customer List

Maintenance

Daily Report

Maintenance

User List

Contact Info

Settings

Inventory

Show 10 entries

Search:

#	Date Created	Image	Brand	Name	Available	Action
1	2022-05-25 11:10		Brand 101	Amoxicillin 250	-4	View
2	2022-05-25 11:06		Brand 101	Mefenamic 500mg	-3	View
3	2022-05-25 11:11		Brand 102	Drug 101 25mg	-8	View
4	2022-05-25 14:18		Brand 103	Drug 102 50mg	25	View
5	2022-05-26 10:59		Brand 103	Medicine #101 280ml	0	View

Showing 1 to 5 of 5 entries

[Previous](#)
[1](#)
[Next](#)

Copyright © 2023. All rights reserved.

OMOS - PHP (by: oretnom23) v1.0

Profile View:

OMOS - PHP

Online Medicine Ordering System - Admin

Administrator Admin

Dashboard

Category List

Product List

Inventory

Orders

Customer List

Maintenance

Daily Report

Maintenance

User List

Contact Info

Settings

Profile View

First Name

Administrator

Middle Name

Last Name

Admin

Username

admin

Password

Avatar

Choose file

Browse

Update

Receipt Print:

OMOS - PHP

Dashboard

Category List

Product List

Inventory

Orders

Customer List

Maintenance

Daily Report

Maintenance

User List

Contact Info

Settings

Online Medicine Ordering System - Admin

Administrator Admin

To exit full screen, move mouse to top of screen or press F11

Developed by oretnom23

Daily Report

Filter

Choose Data

30-07-2023

Filter

Print

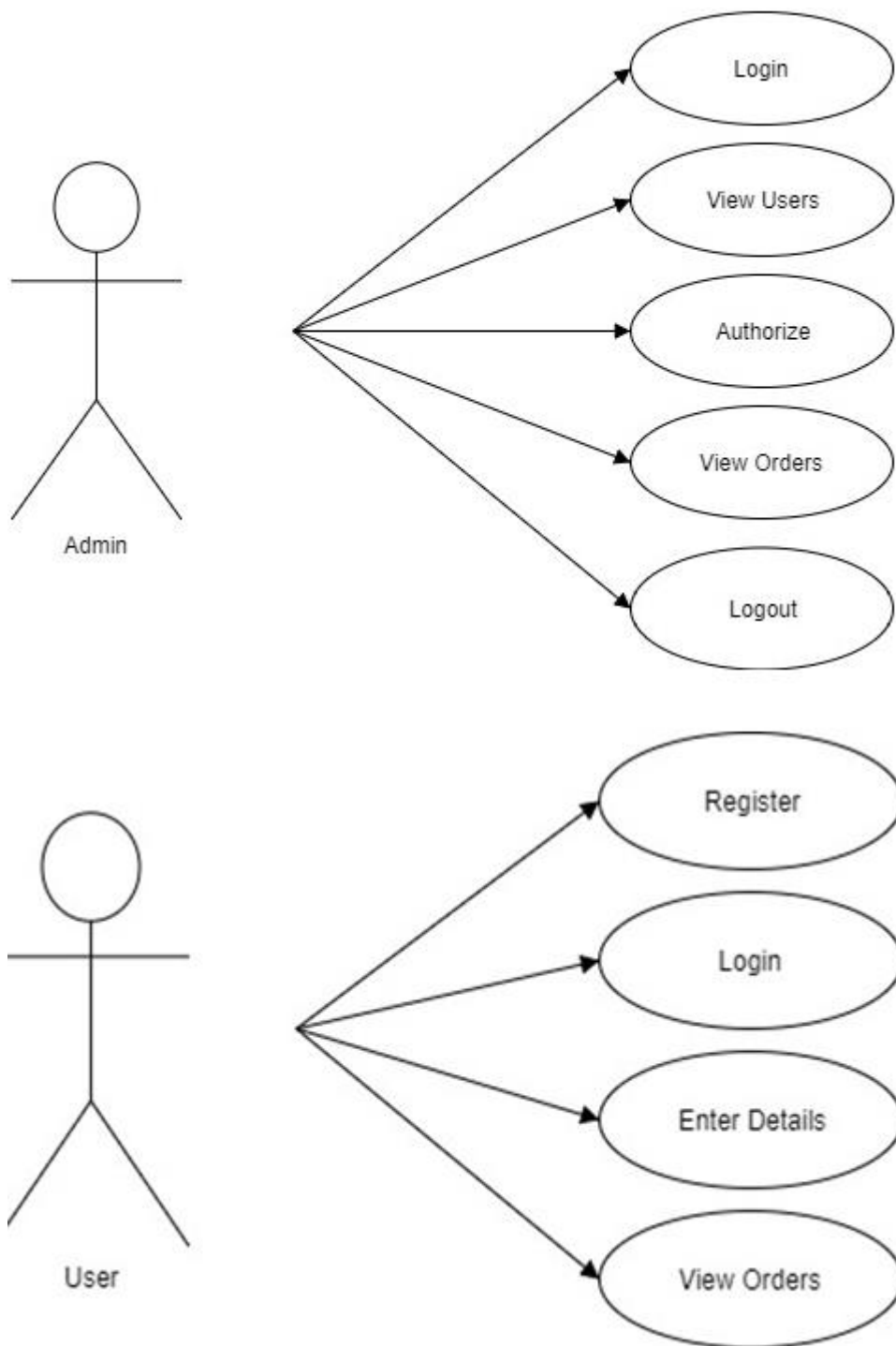
#	Order Code	Product	Unit Price	Quantity	Total
1	2023073100001	Brand 103 Drug 102 (50mg)	20.00	1	20.00
Total Sales					20.00

Copyright © 2023. All rights reserved.

OMOS - PHP (by: oretnom23) v1.0

CHAPTER 9

ANALYSIS AND DESIGN

Use Case Diagram:

CHAPTER 10

CONCLUSION

CONCLUSION

The Medical Shop Management System project is an essential and efficient software solution designed to streamline and optimize the operations of a medical shop or pharmacy. Throughout the development and implementation of this project, several key features and functionalities were integrated to enhance the overall management and productivity of the medical store.

The project's primary objectives were to improve the accuracy of inventory management, automate the sales process, enhance customer service, and provide valuable insights through data analytics. These goals have been successfully achieved, resulting in numerous benefits for both the pharmacy staff and customers.

The conclusion of the Medical Shop Management System project highlights the following key points:

- **Inventory Management:** The system offers a robust and user-friendly inventory management module that enables the pharmacy staff to efficiently track stock levels, monitor expiration dates, and automatically place orders for restocking. This helps to reduce stockouts and minimize wastage, ultimately leading to increased profitability.
- **Sales Automation:** The project's sales module automates the billing process, enabling faster and more accurate transactions. The inclusion of barcode scanning and integration with a point-of-sale (POS) system simplifies the checkout process, reducing waiting times and improving customer satisfaction.
- **Customer Service:** The system includes a customer database that stores essential information, such as purchase history and contact details. This allows the pharmacy staff to offer personalized services, such as medication reminders and special discounts, fostering better customer relationships and loyalty.
- **Data Analytics:** With the integration of data analytics tools, the project empowers the management to gain valuable insights into sales trends, popular products, and overall business performance. Informed decision-making becomes possible, leading to strategic planning and improved profitability.
- **User-Friendly Interface:** The system boasts a user-friendly interface, making it accessible to all levels of staff without extensive training. The intuitive design reduces the learning curve and ensures smooth adoption across the pharmacy.
- **Security and Privacy:** Emphasis was placed on data security and privacy to protect sensitive information and comply with relevant regulations. Access controls and

encryption measures were implemented to safeguard the system from potential threats.

In conclusion, the Medical Shop Management System project has successfully achieved its goals of optimizing pharmacy operations, improving customer service, and enhancing overall efficiency. Its user-friendly interface and comprehensive features make it an indispensable tool for managing a medical shop. As the system continues to evolve with the changing needs of the pharmacy, it is expected to contribute significantly to the growth and success of the business.

CHAPTER 11

FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

To ensure that the Medical Shop Management System remains relevant and competitive in the ever-evolving healthcare landscape, it is essential to plan for future enhancements. Here are some potential areas of improvement for the project:

- **E-Commerce Integration:** Incorporate an e-commerce platform that allows customers to place orders online and have them delivered to their doorstep. This enhancement will expand the customer base, provide greater convenience, and increase sale.
- **Mobile Application:** Develop a mobile app for the Medical Shop Management System, enabling customers to browse products, check stock availability, and place orders from their smartphones. Additionally, the app could provide medication reminders and offer exclusive discounts to app users, encouraging customer engagement and loyalty.
- **Prescription Management:** Implement a feature that allows the system to capture and manage prescriptions electronically. This enhancement will streamline the dispensing process, reduce errors, and provide a digital record of patient prescriptions and medication history.
- **Intelligent Analytics:** Enhance the data analytics capabilities of the system to provide more in-depth insights. Implement predictive analytics to forecast demand, optimize inventory levels, and identify trends, enabling better decision-making and resource planning.
- **Automated Supplier Integration:** Integrate the system with suppliers' databases, allowing for automated purchase orders based on real-time inventory levels. This integration will help maintain optimal stock levels, reduce manual intervention, and improve supplier relationships.
- **Patient Health Records:** Expand the system to include a patient health records module that securely stores medical histories, diagnoses, and prescribed medications. This integration will enable pharmacists to provide personalized medication advice and avoid potential drug interactions.
- **CRM Integration:** Integrate the Medical Shop Management System with a Customer Relationship Management (CRM) tool to manage customer interactions, track feedback, and address complaints promptly.

- **Staff Training and Certification Tracking:** Develop a module to track staff training and certification records. This will ensure that all employees are up-to-date with the latest regulations and best practices.
- **Cloud-Based Solution:** Offer a cloud-based version of the system to provide scalability, accessibility from any location, and automatic backups for data security.
- **By continuously enhancing the Medical Shop Management System** with these features and improvements, pharmacies can stay competitive, improve customer satisfaction, and adapt to the dynamic healthcare industry. It will ultimately enable them to deliver better services, streamline operations, and achieve greater success in the long term.

CHAPTER 12

BIBLIOGRAPHY

BIBLIOGRAPHY

Programming languages:

- <https://www.w3schools.com>
- <https://www.javapoint.com>
- <https://www.youtube.com>
- <https://www.wikipedia.com>
- <https://getbootstrap.com>

Software:

- www.apachefriends.org
- visualstudio.com

References:

- Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, "Implementing Customizable Medical Shop Management System (MSMS) Using Web Based Application", International Journal of Innovative Science, Engineering Technology(IJSET) 2015
- Patel Krishna, Patel Palak, Raj Nirali, Patel Lalit," Medical Shop Management System (MSMS)", International Journal of Engineering Research and Development (IJERD) 2015.
- Awojide, Simon, I. M. Omogbhemhe, O. S. Awe, and T. S. Babatope, "Towards the digitalization of Medical Shop Management System (MSMS) in Nigeria Private University: The Design Perspective. A Study of Samuel Adegboyega University Edo State Nigeria," Int. J. Sci. Res. Publ., vol. 8, no. 5, pp. 46–54, 2018
- mysql code [online] available at www.stackoverflow.com