**~~Title of the Project~~**

A project report submitted to Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore - 641 020 in partial fulfillment of the

requirements for the award of the Degree of

### MASTER OF COMPUTER APPLICATIONS

Submitted by

### NAME

**Reg. No.: …………..**

*Under the guidance of*

**Dr.M.CHANDRAN MCA.,M.Phil.,Ph.D.**

**Associate Professor**

****

### DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

**SRI RAMAKRISHNA MISSION VIDYALAYA COLLEGE OF ARTS AND SCIENCE**

*(An Autonomous Institution Affiliated to Bharathiar University, Re-Accredited by NAAC with A+ grade)*

### COIMBATORE-641 020 APRIL-2024

**CERTIFICATE**

This is to certify that the project entitled ~~“title of the project”~~ submitted to Sri Ramakrishna Mission Vidyalaya College of Arts and Science,Coimbatore-641 020, affiliated to Bharathiar University, in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a record of original project work done by **, Reg. No.: ,** during the academic year 2023-24 in the Department of Master of Computer Aplications at Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore- 20, under my supervision and guidance and the dissertation has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate of any university.

Place: Coimbatore-20

Date: Signature of the

Guide

Countersigned

Head of the Department Principal

Internal Examiner External Examiner

### DECLARATION

I, **Name of the candidate (Reg. No.),** hereby declare that the project entitled **“** **”** submitted in partial fulfillment of the requirements for the award of the Degree of **Master of Computer Applications** is a record of original project work done by me during the academic year 2023-2024 under the supervision and guidance of Dr.M.Chandran Associate Professor, Department of MCA, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore-20, and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate of any university.

Place: Coimbatore-20 Signature of the

Candidate

Date: **(Name of the**

**Candidate)**

|  |  |
| --- | --- |
| **CONTENTS** |  |
| **SYNOPSIS** | **………………….. 01** |
| **CHAPTER – I INTRODUCTION**  1.1 About the Project | **…………………... 03** |
| 1.2 About the language | **. 04** |
| **CHAPTER – II SYSTEM STUDY AND ANALYSIS** |  |
| 2.1. Existing system | **………………….. 09** |
| 2.2. Drawback of the existing system | **………………….. 09** |
| 2.3. System analysis | **………………….. 09** |
| 2.4. Proposed System | **………………….. 10** |
| 2.4.1. Advantages of the proposed system | **………………….. 11** |
| 2.5. Feasibility Study | **………………….. 11** |
| 2.5.1. Technical Feasibility | **………………….. 12** |
| 2.5.2. Economic Feasibility | **………………….. 12** |
| 2.5.3. Behavioral Feasibility | **………………….. 13** |
| 2.6. Hardware and software requirements | **………………….. 14** |

**CHAPTER – III SYSTEM DESIGN AND DEVELOPMENT PROCESS**

|  |  |
| --- | --- |
| 3.1. Introduction | **………………….. 15** |
| 3.2. Input Design | **………………….. 17** |
| 3.3. Output Design | **………………….. 18** |
| 3.4. Database Design | **………………….. 18** |
| 3.5. Table Design | **………………….. 23** |
| 3.6. Data Flow Diagram | **………………….. 31** |
| 3.7. ER Diagram | **………………….. 33** |

**CHAPTER – IV SYSTEM TESTING AND IMPLEMENTATION**

|  |  |
| --- | --- |
| 4.1. System Testing | **………………….. 40** |
| 4.1.1. Unit Testing | **………………….. 42** |
| 4.1.2. Integration testing | **………………….. 43** |
| 4.1.3. Validation Testing | **………………….. 44** |
| 4.1.4. Output Testing or User Acceptance Testing | **………………….. 45** |
| 4.2. System Implementation | **………………….. 45** |
| 4.3. Training | **………………….. 46** |
| 4.4. System Maintenance | **………………….. 46** |
| **CHAPTER – V APPENDIX** |  |
| 5.1. Sample Source Code | **………………….. 47** |
| 5.2. Screen Shots | **………………….. 51** |

**CHAPTER – VI CONCLUSION AND FUTURE ENHANCEMENT**

* 1. [Conclusion **75**](#_TOC_250001)
  2. Scope for Further Enhancement **75**

[BIBLIOGRAPHY 76](#_TOC_250000)

# SYNOPSIS

The **Salon Management System** is a web-based application developed using PHP and MySQL, aimed at streamlining and automating various day-to-day activities of a salon. In the traditional setup, salons manage appointments, staff, and customer records manually, which often leads to inefficiencies and customer dissatisfaction. This system addresses those challenges by providing a user-friendly digital platform for both salon administrators and customers.

The primary goal of this system is to provide a seamless experience for customers to book appointments online, choose from a list of services, and receive real-time updates on their booking status. By logging into the customer panel, users can view all available services along with pricing, select a preferred date and time, and confirm their appointments without needing to call or visit the salon in person. This ensures convenience for the customers and reduces the burden on salon staff.

For the admin, the system offers a centralized dashboard to manage all core activities. Admins can add, update, or remove services, view upcoming appointments, manage staff members, and analyze reports related to bookings and revenue. This helps the salon keep track of operations and make informed business decisions. Admin authentication ensures that only authorized users can access sensitive data and perform critical actions.

The system is built with a responsive interface to ensure usability across devices, including desktops, tablets, and smartphones. The use of MySQL for the backend database ensures secure and efficient data management, while PHP handles the logic and communication between the interface and the database. Additional features such as email notifications, appointment status tracking, and booking history enhance the overall functionality.

In the future, this system can be further expanded to support online payments, SMS reminders, loyalty programs, and mobile application integration. By embracing such features, the salon can offer a more professional and customer-centric experience. Overall, the Salon Management System is a scalable and practical solution for modernizing salon operations and improving service delivery.

### MODULE:

The module which performs all the main operations in the Saloon Management system. The major operations in the project are:

**Admin Panel:**

* Login Authentication
* Dashboard Overview
* Manage Services (Add/Update/Delete services)
* Manage Appointments (View/Approve/Cancel)
* Manage Staff
* View Customer Records
* View Reports (Daily/Weekly/Monthly appointments and earnings)

**2. Customer Panel:**

* User Registration and Login
* View Available Services
* Book Appointment (Choose service, date, and time)
* View Booking History and Status
* Edit Profile

The Admin Module is the core of the Salon Management System. It gives the administrator full control over the entire platform. Through this module, the admin can manage all essential features such as adding, updating, or deleting salon services, assigning or managing staff members, and monitoring customer appointments.

The Customer Module is designed for salon clients who want to view services and book appointments online. New users can register and create an account, while existing users can log in to access their booking history, profile, and upcoming appointments. This module makes it easy for users to choose services, select dates and time slots, and submit bookings, all from the convenience of their device.These modules collectively ensure a fully functional, user-friendly, and efficient system that caters to both customer convenience and administrative control.

# INTRODUCTION

## About the project:

The Salon Management System is a web-based application developed to simplify and automate the operations of a beauty salon or spa. Traditionally, salons rely on manual systems such as paper logs or phone calls to manage appointments and customer records, which often leads to scheduling conflicts, customer dissatisfaction, and inefficient time management. This project addresses these issues by offering a digital solution where all salon-related activities can be managed through a centralized and user-friendly interface.

The system allows customers to conveniently browse services, book appointments online, and receive updates on their bookings. It provides a streamlined experience that enhances customer satisfaction while reducing the workload for salon staff. For salon owners or administrators, the system offers powerful tools to manage appointments, services, staff details, and customer information, all from a secure admin panel.

Built using PHP for the backend and MySQL for the database, this project ensures dynamic interaction between users and the system. The frontend, built using HTML, CSS, JavaScript, and Bootstrap, ensures a clean, responsive design that works well on all devices. The system also includes user authentication and role-based access to ensure secure data handling and separation of admin and customer functionalities.

The project is ideal for small to medium-sized salons looking to digitize their operations without investing in expensive software. It is scalable, meaning additional features like payment gateways, SMS/email notifications, and mobile app support can be integrated in the future to enhance its functionality.

Overall, the Salon Management System is a smart and efficient solution that bridges the gap between technology and traditional salon services, providing a modern approach to managing appointments, services, and customer relationships.

## About the Language

### PHP:

PHP is a purpose server language originally designed for [Web development](http://en.wikipedia.org/wiki/Web_development) to produce [dynamic Web pages](http://en.wikipedia.org/wiki/Dynamic_Web_page). It is one of the first developed server-side scripting languages to be embedded into an [HTML](http://en.wikipedia.org/wiki/HTML) source document rather than calling an external file to process data. The code is [interpreted](http://en.wikipedia.org/wiki/Interpreter_%28computing%29) by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a [command-line interface](http://en.wikipedia.org/wiki/Command-line_interface) capability and can be used in standalone graphical.

PHP can be deployed on most Web servers and also as a standalone [shell](http://en.wikipedia.org/wiki/Shell_%28computing%29) on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Computing_platform) free of charge. A competitor to [Microsoft'](http://en.wikipedia.org/wiki/Microsoft)s [Active Server Pages](http://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million [Web servers.](http://en.wikipedia.org/wiki/Web_server) Software that uses PHP includes [Media Wiki](http://en.wikipedia.org/wiki/MediaWiki), [Joomla](http://en.wikipedia.org/wiki/Joomla), [Word press](http://en.wikipedia.org/wiki/Wordpress), [Concrete5,](http://en.wikipedia.org/wiki/Concrete5) [MyBB,](http://en.wikipedia.org/wiki/MyBB) and [Drupal](http://en.wikipedia.org/wiki/Drupal).

While PHP originally stood for *Personal Home Page*, it is now said to stand for

*PHP: Hypertext Preprocessor*, a [recursive acronym](http://en.wikipedia.org/wiki/Recursive_acronym).

#### History:

PHP development began in 1994 when the [Danish](http://en.wikipedia.org/wiki/Danish_people)/[Greenlandic](http://en.wikipedia.org/wiki/Greenland)/[Canadian](http://en.wikipedia.org/wiki/Canada) programmer [RasmusLerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) initially created a set of [Perl](http://en.wikipedia.org/wiki/Perl) scripts he called "Personal Home Page Tools" to maintain his [personal homepage.](http://en.wikipedia.org/wiki/Personal_homepage) The scripts performed tasks such as displaying his résumé and recording his web-page [traffic.](http://en.wikipedia.org/wiki/Web_traffic)

Lerdorf released PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" publicly on June 8, 1995.In 1997 they formed the base of PHP 3, changing the language's name to the [recursive initialism](http://en.wikipedia.org/wiki/Recursive_initialism)*PHP: Hypertext Preprocessor*.

As of August 2008 this branch is up to version 4.4.9. PHP 4 is no longer under development nor will any security updates be released.PHP 5 included new features such as improved support for [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming).Since version 5.4, PHP has native support for [Unicode](http://en.wikipedia.org/wiki/Unicode) or multibyte strings, allowing strings as well as class-, method-, and function-names to contain non-[ASCII](http://en.wikipedia.org/wiki/ASCII) characters.

PHP interpreters are available on both [32-bit](http://en.wikipedia.org/wiki/32-bit) and [64-bit](http://en.wikipedia.org/wiki/64-bit) operating systems, but on [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) the only official distribution is a 32-bit implementation, requiring Windows 32-bit compatibility mode while using [Internet Information Services](http://en.wikipedia.org/wiki/Internet_Information_Services) (IIS) on a 64-bit Windows platform. Experimental 64-bit versions of PHP 5.3.0 were briefly available for MS Windows, but have since been removed.

#### Security:

* There are advanced protection patches such as [Suhosin](http://en.wikipedia.org/wiki/Suhosin) and [Hardening](http://en.wikipedia.org/wiki/Hardening_%28computing%29)-Patch, especially designed for Web hosting environments.
* Recognizing that programmers make mistakes, some languages include [taint](http://en.wikipedia.org/wiki/Taint_checking) [checking](http://en.wikipedia.org/wiki/Taint_checking) to detect automatically the lack of [input validation](http://en.wikipedia.org/wiki/Data_validation) which induces many issues are done in PHP.

#### Syntax:

The PHP interpreter only executes PHP code within its [delimiters](http://en.wikipedia.org/wiki/Delimiter). Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to [control structures](http://en.wikipedia.org/wiki/Control_structure) described in PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <?or<?= (which is used to echo back a [string](http://en.wikipedia.org/wiki/String_%28computer_science%29) or [variable](http://en.wikipedia.org/wiki/Variable_%28programming%29)) and ?> as well as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages)-style short forms <% or <%= and %>. While short delimiters are used, they make script files less portable as support for them can be disabled in the [PHP configuration,](http://wiki.php.net/rfc/shortags) and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, <?php and ?>, in [XHTML](http://en.wikipedia.org/wiki/XHTML) and other [XML](http://en.wikipedia.org/wiki/XML) documents, creates correctly formed XML 'processing instructions'. This means that the

resulting mixture of PHP code and other markup in the server-side file is itself well- formed XML.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. if conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

<!DOCTYPE html>

<meta charset=utf-8>

<title>PHP Test</title>

<?php

echo 'Hello World';

?>

**INTRODUCTION TO JAVASCRIPT**

An explanation of exactly what JavaScript is has to begin with Java. Java is a new kind of Web programming language developed by Sun Microsystems. A Java program, or *applet*, can be loaded by an HTML page and executed by the Java Interpreter, which is embedded into the browser.

Java is a complex language, similar to C++. Java is object-oriented and has a wide variety of capabilities; it's also a bit confusing and requires an extensive development cycle. That's where JavaScript comes in.

You can program in JavaScript easily; no development tools or compilers are required. You can use the same editor you use to create HTML documents to create JavaScript, and it executes directly on the browser (currently, Netscape or Microsoft Internet Explorer).

JavaScript was originally called LiveScript, and was a proprietary feature of the Netscape browser. JavaScript has now been approved by Sun, the developer of Java, as a scripting language to complement Java. Support has also been announced by several other companies.

Although useful in working with Java, you'll find that JavaScript can be quite useful in its own right. It can work directly with HTML elements in a Web page, something Java can't handle. It is also simple to use, and you can do quite a bit with just a few JavaScript statements.

## Hyper Text Markup Language(HTML)

Hyper is the opposite of linear. It used to be that computer programs had to move in a linear fashion. This before this, this before this, and so on. HTML does not hold to that pattern and allows the person viewing the World Wide Web page to go anywhere, any time they want.

HTML is the code behind your webpage and is what your browser looks for to display a webpage, the way the webdesigner intended it to look, and is a series of tags <tags> that tells the browser where to display what. It is really a series of simple commands that you give to the browser, just like telling your dog to sit, and because it is in plain English it is easy to learn. For example, if you want your text to show in a bold type, you command it

<bold> to be bold text </bold>, it really is that easy.

# CASCADING STYLE SHEET

## What are Cascading Style Sheets?

CSS was first developed in 1997, as a way for Web developers to define the look and feel of their Web pages. It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on - the markup of content, without worry about the design and layout.

CSS didn’t gain in popularity until around 2000, when Web browsers began using more than the basic font and color aspects of CSS. And now, all modern browsers support all of CSS Level 1, most of CSS Level 2, and some aspects of CSS Level 3.

Web Designers that don’t use CSS for their design and development of Web sites are rapidly becoming a thing of the past. And it is arguably as important to understand CSS as it is to know HTML - and some would say it was more important to know CSS.

## Where is CSS Used?

CSS is used to style Web pages. But there is more to it than that. CSS is used to style XHTML and XML markup. This means that anywhere you have XML markup (including XHTML) you can use CSS to define how it will look.

CSS is also used to define how Web pages should look when viewed in other media than a Web browser. For example, you can create a print style sheet that will define how the Web page should print out and another style sheet to display the Web page on a projector for a slide show.

## Why is CSS Important?

CSS is one of the most powerful tools a Web designer can learn because with it you can affect the entire mood and tone of a Web site. Well written style sheets can be updated quickly and allow sites to change what is prioritized or valued without any changes to the underlying XHTML.

But because CSS can cascade, and combine and browsers interpret the directives differently, CSS is more difficult than plain HTML. But once you start using it, you’ll see that harnessing the power of CSS will give you more options and allow you to do more and more things with your Web sites.

# SYSTEM STUDY AND ANALYSIS

## Existing System

In the existing or traditional salon management system, most tasks such as appointment bookings, service records, and customer details are handled manually. Customers usually have to call or physically visit the salon to check service availability and book appointments. This often results in scheduling conflicts, forgotten bookings, and long waiting times, which negatively affect both customer experience and salon productivity.

Service details, staff assignments, and customer preferences are commonly maintained in registers or spreadsheets. This manual data handling not only increases the risk of errors and data loss but also makes it difficult to retrieve or analyze historical information for decision-making. Managing staff availability and keeping track of appointments in peak hours becomes challenging without a structured system in place.

## Drawbacks of the Existing System

The existing manual salon management system has several limitations that hinder the smooth and efficient operation of salon services. Some of the major drawbacks are outlined below:

**Time-Consuming Processes:**  
 Booking appointments and managing schedules manually takes up a significant amount of time for both salon staff and customers. This reduces productivity and leads to unnecessary delays.

**Risk of Human Error:**  
Manual entries are prone to mistakes such as double-booking, incorrect appointment times, or missing customer information. These errors can lead to customer dissatisfaction and service delays.

**Lack of Centralized Data Management:**  
Customer details, service history, and staff schedules are often scattered across paper files or spreadsheets. This makes it difficult to retrieve or update information quickly and accurately.

**No Real-Time Updates:**  
Customers have no way to check real-time service availability or receive updates about their bookings. Any changes in schedule must be communicated manually, which is inefficient and prone to miscommunication.

**No Reminders or Notifications:**  
The existing system lacks automated notifications or reminders for upcoming appointments. As a result, customers may forget their scheduled visits, and salons may face last-minute cancellations.

## System Analysis

The Salon Management System is designed to overcome the limitations of the existing manual system by introducing a digital, web-based solution that streamlines and automates salon operations. The system analysis focuses on understanding the core requirements of salon management and designing a system that caters to both administrative and customer needs in an efficient, secure, and user-friendly manner.

At the heart of the system is the appointment booking module, which allows customers to book salon services online, view available time slots, and receive confirmation and status updates. This minimizes scheduling conflicts and reduces customer wait times. The system also includes user authentication to ensure secure access for admins, customers, and staff, with role-based permissions to protect sensitive data and limit unauthorized actions.

The admin dashboard serves as the control panel where salon owners or managers can manage services, staff, and appointments. Through this dashboard, admins can add or update services, assign appointments to staff members, monitor booking activity, and generate business reports. These features greatly improve operational efficiency and decision-making.

From the customer’s perspective, the system provides a seamless interface to browse services, book appointments, and manage their profile. It also allows them to view their appointment history and receive updates on service status, which enhances their overall experience and engagement.

The system uses PHP as the backend scripting language and MySQL for the database, ensuring data integrity and fast access to records. The frontend, built using HTML, CSS, JavaScript, and Bootstrap, ensures the system is visually appealing and responsive across devices. Together, these technologies provide a robust and scalable platform that can be enhanced with additional features like online payments, notifications, and mobile app integration in the future.

## Proposed System

The Proposed Salon Management System is a fully web-based solution developed using PHP and MySQL, aimed at addressing the inefficiencies of the current manual system. It is designed to automate the entire process of salon operations, from appointment bookings to staff and service management. This system will provide an efficient, organized, and user-friendly environment for both customers and salon administrators.

#### 4. 1 Advantages of the Proposed System

### ****Automated Appointment Booking****

Customers can easily book appointments online without visiting or calling the salon. This reduces staff workload and minimizes errors caused by manual scheduling.

### ****Improved Customer Experience****

Users can view available services, choose time slots, and receive instant updates about their booking status. This convenience enhances customer satisfaction and encourages repeat visits.

### ****Efficient Staff and Service Management****

The admin can assign appointments to specific staff members, manage service listings, and adjust schedules with just a few clicks. This leads to better organization and service delivery.

### ****Centralized Data Handling****

All customer, staff, and service-related information is stored in a centralized MySQL database. This ensures data consistency, easy access, and better record-keeping for future use.

### ****Time and Cost Saving****

Automation significantly reduces the time spent on managing bookings, generating reports, and handling customer records, leading to increased productivity and reduced operational costs.

## Feasibility Study

A feasibility study is a crucial step in the project development life cycle. It helps determine whether the proposed Salon Management System is practical and viable in terms of technical, economic, operational, and time-related aspects. The goal is to ensure that the system can be successfully developed and implemented with the available resources and will provide benefits that justify the investment.

#### Technical Feasibility

The project is technically feasible as it uses reliable and well-supported technologies such as PHP for server-side scripting and MySQL for database management. These technologies are widely used, open-source, and compatible with various platforms. The system can be hosted on standard web servers and does not require high-end hardware or proprietary software, making it easy to implement and maintain. Frontend development using HTML, CSS, Bootstrap, and JavaScript ensures responsive and modern user interfaces.

#### Economic Feasibility

The system is economically feasible as it is developed using free and open-source tools, reducing software licensing costs. Since the system is web-based, there is no need for expensive infrastructure or installations. In the long run, the system will reduce the cost of manual operations, paper records, appointment handling, and communication, offering a high return on investment for salon owners.

#### Behavioral Feasibility

The proposed system is user-friendly and easy to operate for both salon staff and customers. Admins can manage services, staff, and appointments with minimal training. The intuitive user interface makes it simple for customers to browse services and book appointments. The system also improves operational workflow and reduces manual effort, which increases staff productivity and customer satisfaction.

## Hardware and Software Requirements

**HARDWARE REQUIREMENTS**

Processor : Intel Pentium IV

RAM : 512 MB or more

Hard disk : 20 GB or more

Monitor : VGA/SVGA

Keyboard : 104 Keys

Mouse : 2 buttons/ 3 buttons

**SOFTWARE REQUIREMENTS**

Operating System : Windows 2000/XP Front end : PHP

Back end : MYSQL

# SYSTEM DESIGN AND DEVELOPMENT PROCESS

## Introduction

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

#### Logical Design:

The logical flow of a system and define the boundaries of a system. It includes the following steps:

* Reviews the current physical system – its data flows, file content, volumes , frequencies etc.
* Prepares output specifications – that is, determines the format, content and frequency of reports.
* Prepares input specifications – format, content and most of the input functions.
* Prepares edit, security and control specifications.
* Specifies the implementation plan.
* Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
* Reviews benefits, costs, target dates and system constraints.

#### Physical Design:

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

* Design the physical system.
* Specify input and output media.
* Design the database and specify backup procedures.
* Design physical information flow through the system and a physical design Walk through.
* Plan system implementation.
* Prepare a conversion schedule and target date.
* Determine training procedures, courses and timetable.
* Devise a test and implementation plan and specify any new hardware/software.
* Update benefits , costs , conversion date and system constraints

#### Design/Specification activities:

* Concept formulation.
* Problem understanding.
* High level requirements proposals.
* Feasibility study.
* Requirements engineering.
* Architectural design.

## Input Design

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

#### Objectives

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in a maze of instant. Thus the objective of input design is to create an input layout that is easy to follow

## Output design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should:

* Identify the specific output that is needed to meet the requirements.
* Select methods for presenting information.
* Create document, report, or other formats that contain information produced by the system.

## Database design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called

Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

* Data Integrity
* Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

* To structure the data so that there is no repetition of data , this helps in saving.
* To permit simple retrieval of data in response to query and report request.
* To simplify the maintenance of the data through updates, insertions, deletions.
* To reduce the need to restructure or reorganize data which new application requirements arise.

### RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

#### Relations, Domains & Attributes:

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship

Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

#### Relationships:

Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys. Entity Integrity enforces that no Primary Key can have null values. Referential Integrity enforces that no Primary Key can have null values.

Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

#### Normalization:

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. This includes:

* Normalize the data.
* Choose proper names for the tables and columns.
* Choose the proper name for the data.

#### First Normal Form:

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

#### Second Normal Form:

According to Second Normal Form, For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependant on apart of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

#### Third Normal Form:

According to Third Normal Form, Relation should not have a nonkey attribute functionally determined by another nonkey attribute or by a set of nonkey attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the nonkey attributes that functionally determines other nonkey attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non key attribute.

## Table Design

Table: tbladmin

Primary Key: ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| AdminName | varchar(45) | Yes | NULL |
| UserName | varchar(45) | Yes | NULL |
| MobileNumber | bigint | Yes | NULL |
| Email | varchar(120) | Yes | NULL |
| Password | varchar(120) | Yes | NULL |
| AdminRegdate | timestamp | Yes | CURRENT\_TIMESTAMP |

Table: tblcategory

Primary Key: ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| CategoryName | varchar(120) | Yes | NULL |
| CreationDate | timestamp | Yes | CURRENT\_TIMESTAMP |

## 

Table:Tblfoodtracking

Primary Key:ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| CategoryName | varchar(120) | Yes | NULL |
| ItemName | varchar(120) | Yes | NULL |
| ItemPrice | varchar(120) | Yes | NULL |
| ItemDes | varchar(500) | Yes | NULL |
| Image | varchar(120) | Yes | NULL |
| ItemQty | varchar(120) | Yes | NULL |

Table:Tblfoodtracking

Primary Key:ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| OrderId | char(50) | Yes | NULL |
| remark | varchar(200) | Yes | NULL |
| status | char(50) | Yes | NULL |
| StatusDate | timestamp | Yes | CURRENT\_TIMESTAMP |
| OrderCanclledByUser | int | Yes | NULL |

## Table: Tblorderaddresses

## Primary Key:ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| UserId | char(100) | Yes | NULL |
| Ordernumber | char(100) | Yes | NULL |
| Flatnobuldngno | varchar(255) | Yes | NULL |
| StreetName | varchar(255) | Yes | NULL |
| Area | varchar(255) | Yes | NULL |
| Landmark | varchar(255) | Yes | NULL |
| City | varchar(255) | Yes | NULL |
| OrderTime | timestamp | No | CURRENT\_TIMESTAMP |
| OrderFinalStatus | varchar(255) | Yes | NULL |

## Table: Tblorders

## Primary Key:ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| UserId | char(10) | Yes | NULL |
| FoodId | char(10) | Yes | NULL |
| FoodQty | int | Yes | NULL |
| IsOrderPlaced | int | Yes | NULL |
| OrderNumber | char(100) | Yes | NULL |

## Table: Tbluser

## Primary Key:ID

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| **ID** | int | No |  |
| FirstName | varchar(45) | Yes | NULL |
| LastName | varchar(50) | Yes | NULL |
| Email | varchar(120) | Yes | NULL |
| MobileNumber | bigint | Yes | NULL |
| Password | varchar(120) | Yes | NULL |

* 1. **Data Flow Diagram:**

**Level 1:**

RMV Hospital

User

Admin

**Level 2:**

Login

Login\_db

User

Admin

**Level 3:**

Login

Check

admin\_register

Type

[supplier\_details](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=supplier_details&pos=0) [category\_master](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=category_master&pos=0)

`

Supplier Details

Category Master

Department Details

[department\_details](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=department_details&pos=0)

[purch\_medicine](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=purchase_medicine&pos=0) [doctor\_details](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=doctor_details&pos=0) [test\_master](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=test_master&pos=0)

Test Master

Doctor Details

[outpatient\_appointment](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=out_patient_doctor_appointment&pos=0)



Outpatient Appointment

[doctors\_prescription](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=doctors_prescription&pos=0)

Purchase of medicine

Doctor Prescription



[lab\_details](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=lab_details&pos=0)

Laboratory Details

[medical\_prescription](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=medical_prescription&pos=0)

Medical Pharmacy

Bill

Bill

Bill

#### 3.7 ER Diagram:

Admin Register

Address

phone

Gender

E-mail

Age

username

name

password

pk\_reg\_id

photo

admin\_register

Department Details

depart\_pk\_id

department\_name

department\_details

**Medicine Master**

med\_pk\_id

medicine\_name

medicine\_master

**Supplier Details**

Gender

Age

Address

supplier\_nam

mobile\_no

sup\_pk\_id

e\_mail

supplier\_details

**Category Master**

cat\_pk\_id

category\_name

category\_master

**Test Master**

test\_unit

test\_name

normal\_value

test\_cate\_name

test\_rate

test\_pk\_id

test\_master

avail\_start\_time avail\_day

doc\_desig qualification doc\_gender doc\_name

doc\_id

doc\_pk\_id

avail\_end\_time

mobile\_no

address1

address2

address3

State Pincode

Photo

doctor\_details

**Doctor Details**

**Outpatient Appointment**

outpat\_app\_mobile

outpat\_app\_date

outpat\_app\_address

outpat\_app\_depert

outpat\_app\_age

outpat\_app\_doc\_name

outpat\_app\_name

outpat\_app\_doc\_time

outpat\_app\_id

doc\_app\_time

outpat\_app\_pk\_id

outpat\_app\_token

**outpatient\_appointment**

**Doctor Prescription**

pre\_doc\_date

pre\_doc\_specialist

pre\_doc\_pat\_id

pre\_doc\_name

pre\_doc\_pat\_name

pre\_doc\_id

pre\_doc\_pat\_age

pre\_pk\_id

no\_of\_prescription

[doctors\_prescription](http://localhost:81/phpmyadmin/sql.php?db=rmv_hospital&token=545c2caea7cbd3225370543d41da56d7&table=doctors_prescription&pos=0)

**Medical Prescription**

no\_prescription

morning

prescription\_name

afternoon

med\_pre\_fk\_id

evening

med\_pre\_pk\_id

amount

medical\_prescription

**Purchase Medicine**

exp\_date

dept\_name

qty

medicine\_name

pur\_rate

pur\_fk\_id

sales\_rate

pur\_mas\_pk\_id

mrp\_rate

purchase\_medicine

**Lab Details**

lab\_range

test\_name

normal\_value

labdet\_fk\_id

unit

labdet\_pk\_id

rate

lab\_details

Outpatient Appointment

Purchase Medicine

Supplier Details

Supplier process

Medical process

Test process

Doctor Prescription process

Appointment process

Doctor Prescription

Doctor process

Purchase Medicine process

Admin Register

Doctor Details

Login proces

Medicine process

Lab Details

Medical Prescription

Category process

Category Master

Test Master

Medicine Master

# SYSTEM TESTING AND IMPLEMENTATION

## System Testing

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified?. Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation **:** Are we doing the right job? Verification **:** Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are

Testing is a process of executing a program with the intend of finding an error.A good test case is one that has high possibility of finding an undiscovered error.A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncovered errors in the software also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

* For correctness
* For implementation efficiency
* For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

#### Test Plan

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove

the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

#### Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate

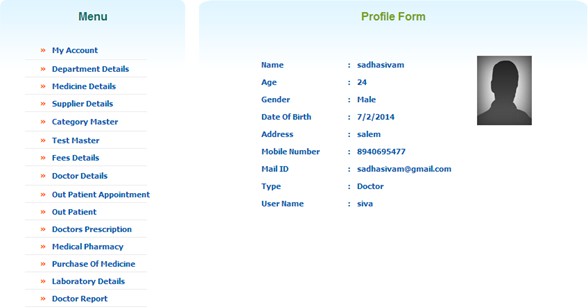
processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

#### Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing in Sell-Soft System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

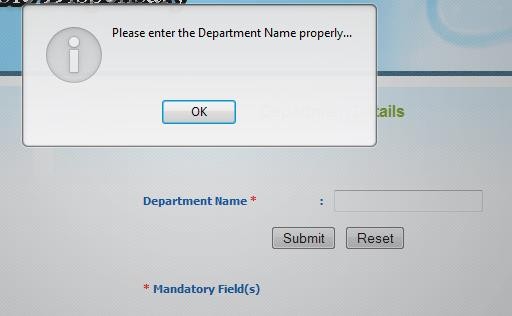


#### Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System testing.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.



#### Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points

* + - * Input Screen Designs,
      * Output Screen Designs,
      * Online message to guide the user and the like.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

## System Implementation:

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

Implementation is the state in the project where the theoretical design is turned into a working system. By this, the users get the confidence that the system will work effectively. The system can be implemented only after through testing.

## Training

Once the system is successfully developed the next important step is to ensure that the administrators are well trained to handle the system. This is because the success of a system invariably depends on how they are operated and used. The implementation depends upon the right people being at the right place at the right time. Education involves creating the right atmosphere and motivating the user. The administrators are familiarized with the run procedures of the system, working through the sequence of activities on an ongoing basis.

The systems personnel check the feasibility of the system. The actual data were inputted to the system and the working of the system was closely monitored. The master option was selected from the main menu and the actual data were input through the corresponding input screens. The data movement was studied and found to be correct queries option was then selected and this contains various reports. Utilities provide various data needed for inventory was input and the module was test run. Satisfactory results were obtained. Reports related to these processes were also successfully generated. Various input screen formats are listed in the appendix.

## System Maintenance:

The Last part of the system development lifecycle is the System maintenance, which is actually the implementation of the post implementation review planned. Maintenance means resorting to its original position. The Proposed System has been designed with effective tools and techniques. The System was designed such that the future changes can be made with minimum changes in the code. The system was also designed to be flexible and Adaptable, so that the maintenance cost in the future can be reduced as much as possible. It has been made easier to maintain the database. Only the authorized person of the institution has been allowed to access the database.

# APPENDIX

## Sample Source Code: Index Page:

<?php

error\_reporting(E\_ERROR | E\_WARNING | E\_PARSE); include("includes/connection.php");

db\_Connect();

?>

<html>

<head>

<title>RMV Hospital</title>

<link rel="stylesheet" type="text/css" href="style.css" media="screen" />

<style type="text/css">

<!--

.style2 {color: #333333}

-->

</style>

</head>

<body>

<div id="main\_container">

<div class="header">

<div id="logo"><a href="#"><img src="images/logo.png" alt="" width="162" height="100" border="0" /></a></div>

<div class="right\_header">

<div id="menu">

<ul>

<li><a class="current" href="Index.php">Home</a></li>

<li><a href="Aboutus.php">About Us</a></li>

<li><a href="Services.php">Services</a></li>

<li><a href="Contactus.php">Contact Us</a></li>

</ul>

</div>

</div>

</div>

<div class="pattern\_bg"></div>

<div id="middle\_box">

</div>

<div class="pattern\_bg"></div>

<div id="main\_content">

<div class="box\_content">

<div class="box\_title">

<h2 align="center"><span class="dark\_blue">Menu</span></h2>

</div>

<div class="box\_text\_content">

<table width="178" border="0" cellpadding="0" cellspacing="0" style="padding- bottom:12px;">

<tr>

<td height="25"><pre class="left"> <span class="style1"> &raquo;</span>

<a href="Index.php">Home</a></pre></td>

</tr>

<tr>

<td><img src="images/index\_15.gif" width="178" height="3" alt="" /></td>

</tr>

<tr>

<td height="25"><pre class="left"> <span class="style1"> &raquo;</span>

<a href="Doctor\_appointment.php">Doctor Appointment</a></pre></td>

</tr>

<tr>

<td><img src="images/index\_15.gif" width="178" height="3" alt="" /></td>

</tr>

<tr>

<td height="25"><pre class="left"> <span class="style1"> &raquo;</span>

<a href="doctor\_details.php">Doctor Details</a></pre></td>

</tr>

<tr>

<td><img src="images/index\_15.gif" width="178" height="3" alt="" /></td>

</tr>

<tr>

<td height="25"><pre class="left"> <span class="style1"> &raquo;</span>

<a href="Laboratory\_details.php">Laboratory Details </a></pre></td>

</tr>

<tr>

<td><img src="images/index\_15.gif" width="178" height="3" alt="" /></td>

</tr>

<tr>

<td height="25"><pre class="left"> <span class="style1"> &raquo;</span>

<a href="Medicine\_details.php">Medicine Details </a></pre></td>

</tr>

<tr>

<td><img src="images/index\_15.gif" width="178" height="3" alt="" /></td>

</tr>

</table>

</div>

</div>

<div class="box\_content1">

<div class="box\_title">

<h2 align="center">Home Pages </h2>

</div>

<div class="box\_text\_content1">

<table border="0" align="center" width="100%">

<tr>

<td align="center"><font color="#FF0000"><b>

<?php

if(isset($\_SESSION['msg']))

{

</tr>

}

?></b></font>

</td>

if($\_SESSION['msg']!="")

echo $\_SESSION['msg'];

$\_SESSION['msg']="";

</table>

<table border="0" align="center" width="100%">

<tr>

<td align="right"></font><div align="left" class="style2">

<div align="justify">Ramakrishna Mission Vidyalaya, Coimbatore, is a major educational centre of the well-known Ramakrishna Mission. It is 19 km from the Coimbatore City on the highway to Ootacamund,and situated in a serene environment spread over about 300 acres. The Institution was founded by Sri T.S. Avinashilingam in 1930 with an investment of just Rs. 5.75 and with one harijan boy on its roll.</div>

</div></td>

</tr>

</table><br />

</div>

</div>

<div class="clear"></div>

</div>

<div id="footer">

<div class="center\_footer">&copy; Copy Right 2014. All Rights Reserved</div>

</div>

</div>

</body>

</html>

## Screen Shots

#### Admin Login page:



**Description:**

This Screen is Admin Login page of the Online RMV Hospital, in which the user can Login with Username and Password the login profile, in which the user can give the personal information before the login.

#### Login Profile Page:



**Description:**

This Screen is user login home and profile page of the Online RMV Hospital.

#### Department Details:

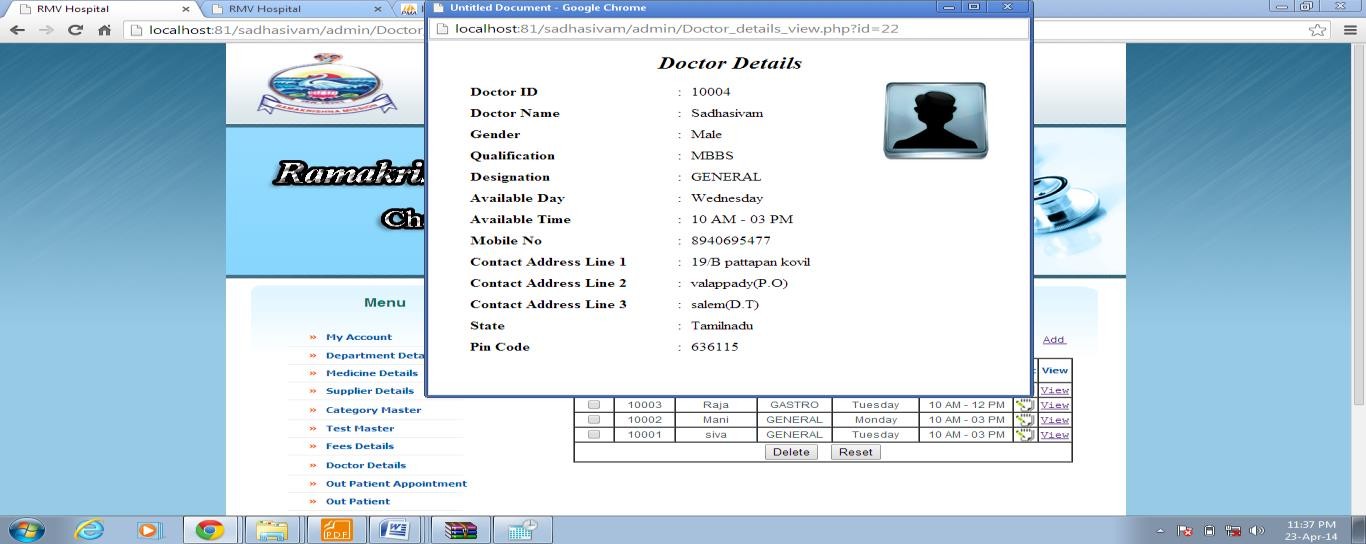
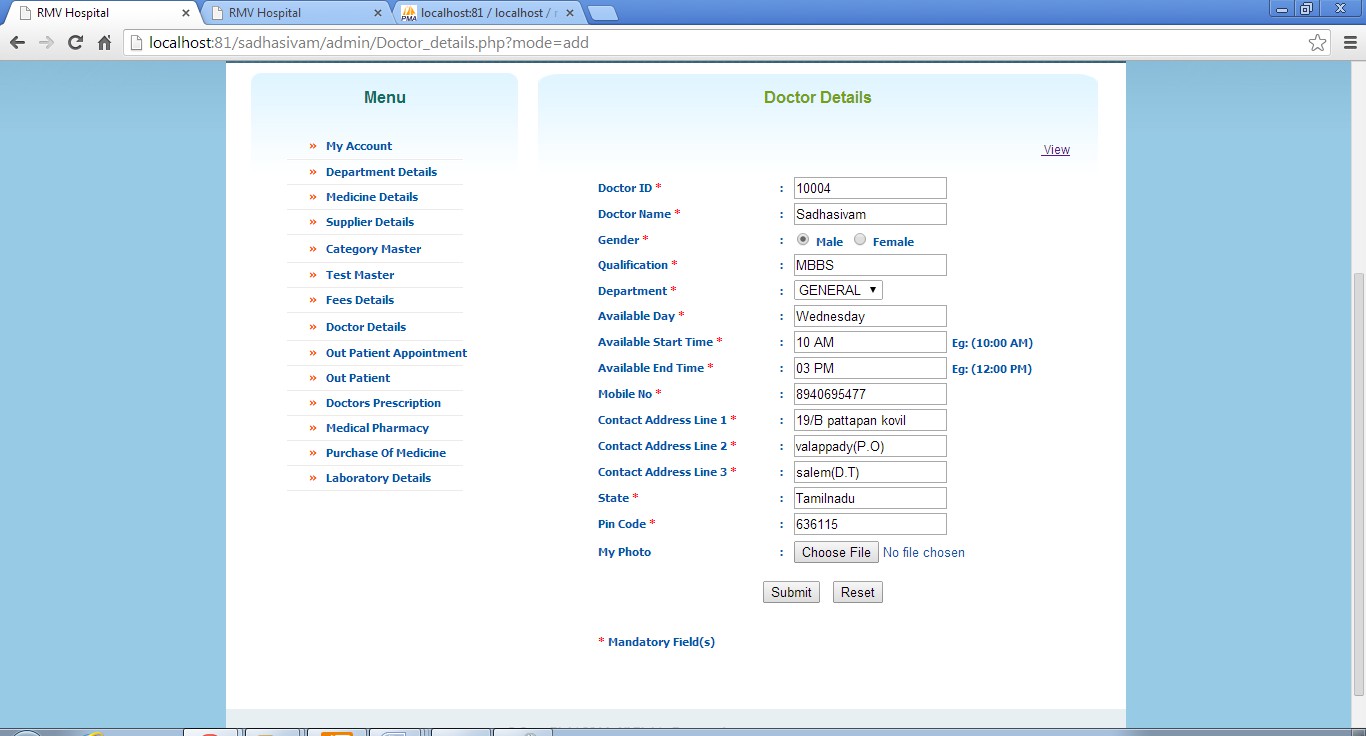


****

**Description:**

This Screen is department master of the Online RMV Hospital, in which the user can view the particular type of department name.

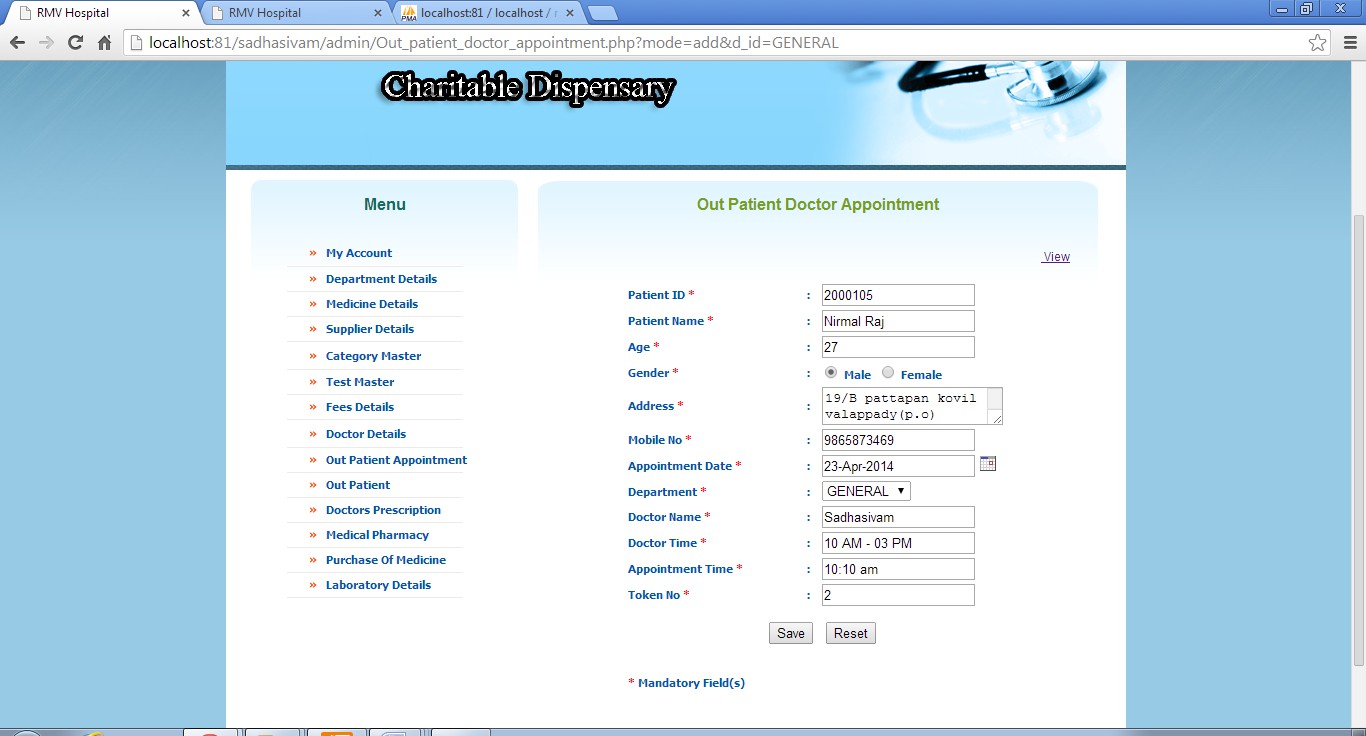
#### Doctor Details:



**Description:**

This Screen is Doctor Detail’s page of the Online RMV Hospital, in which the user can view the doctor details.

#### Doctor Appointment:

****

**Description:**

This Screen is doctor Appointment of the Online RMV Hospital, in which the user can view the particular type of Token Number.

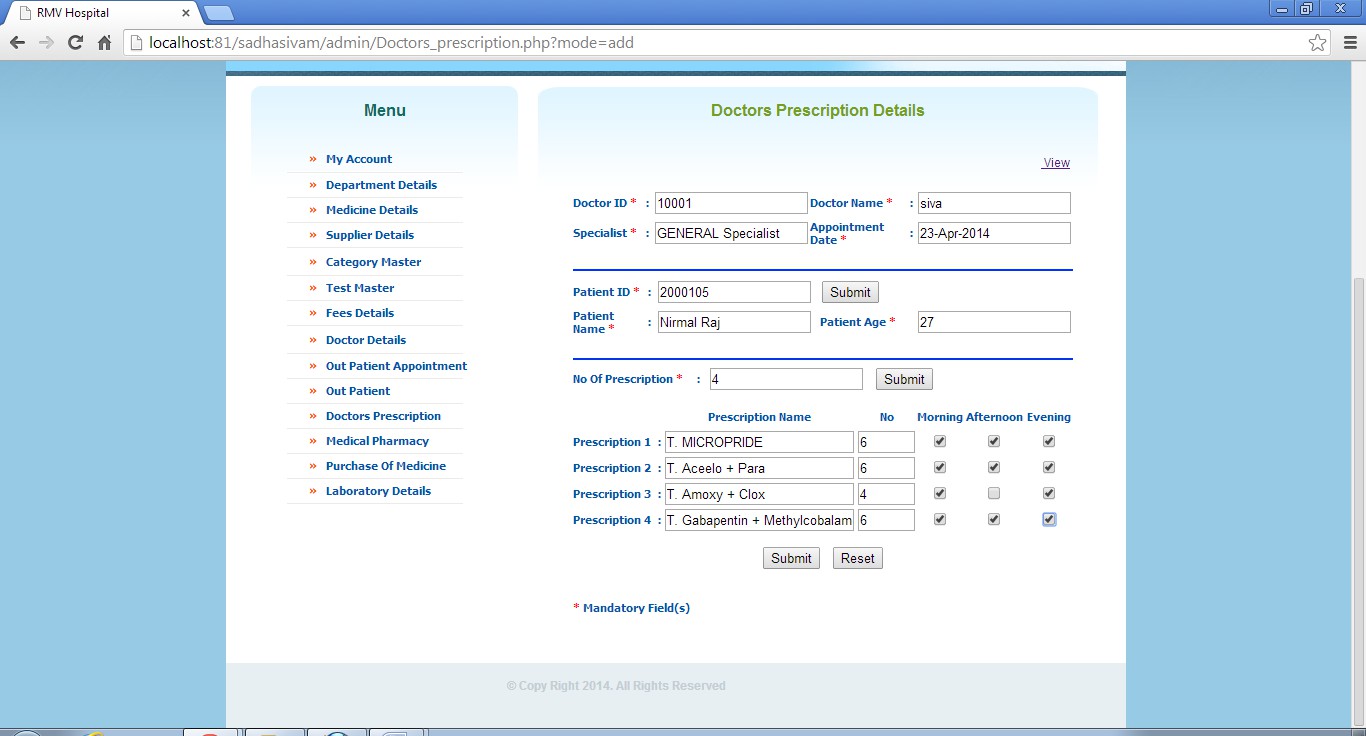
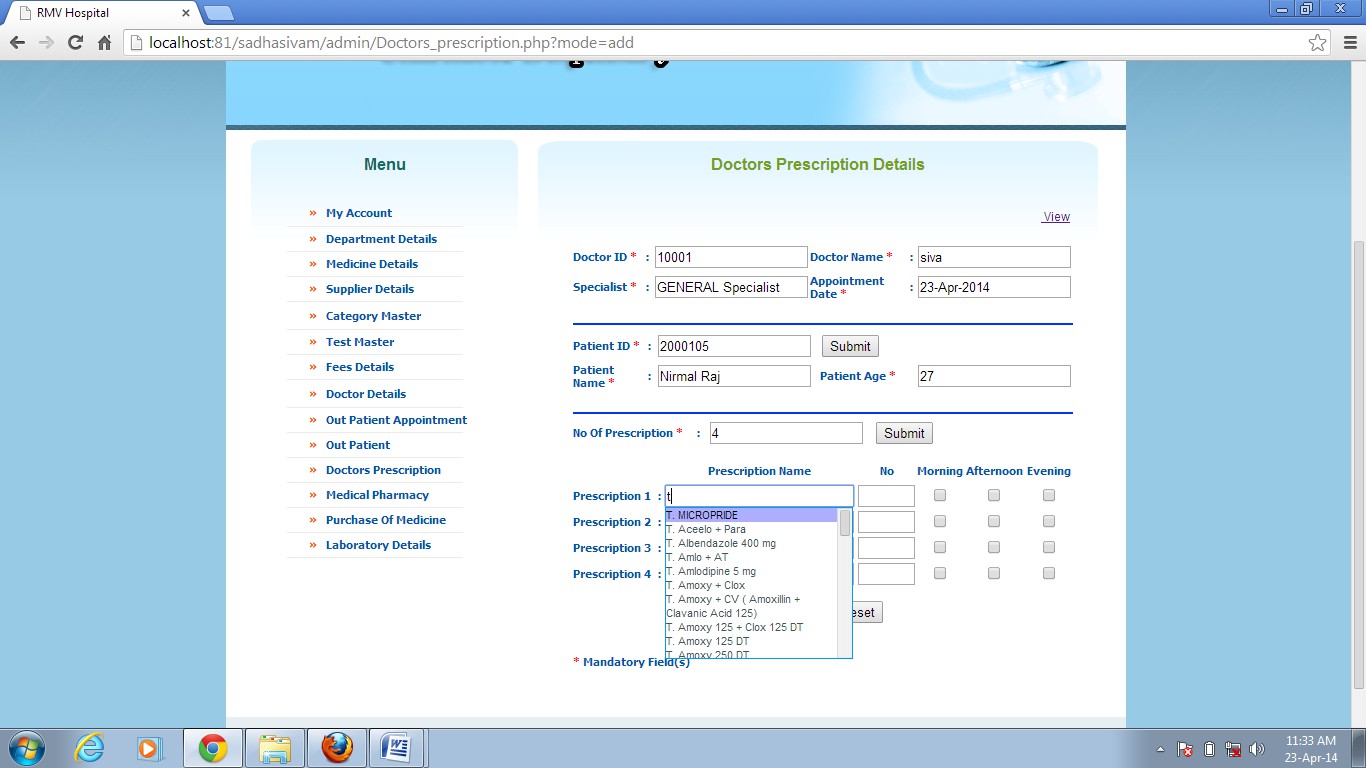
**Medicine Details:**

****

#### Description:

This Screen is Medicine Details page of the Online RMV Hospital, in which the user can view the particular type of medicine name.

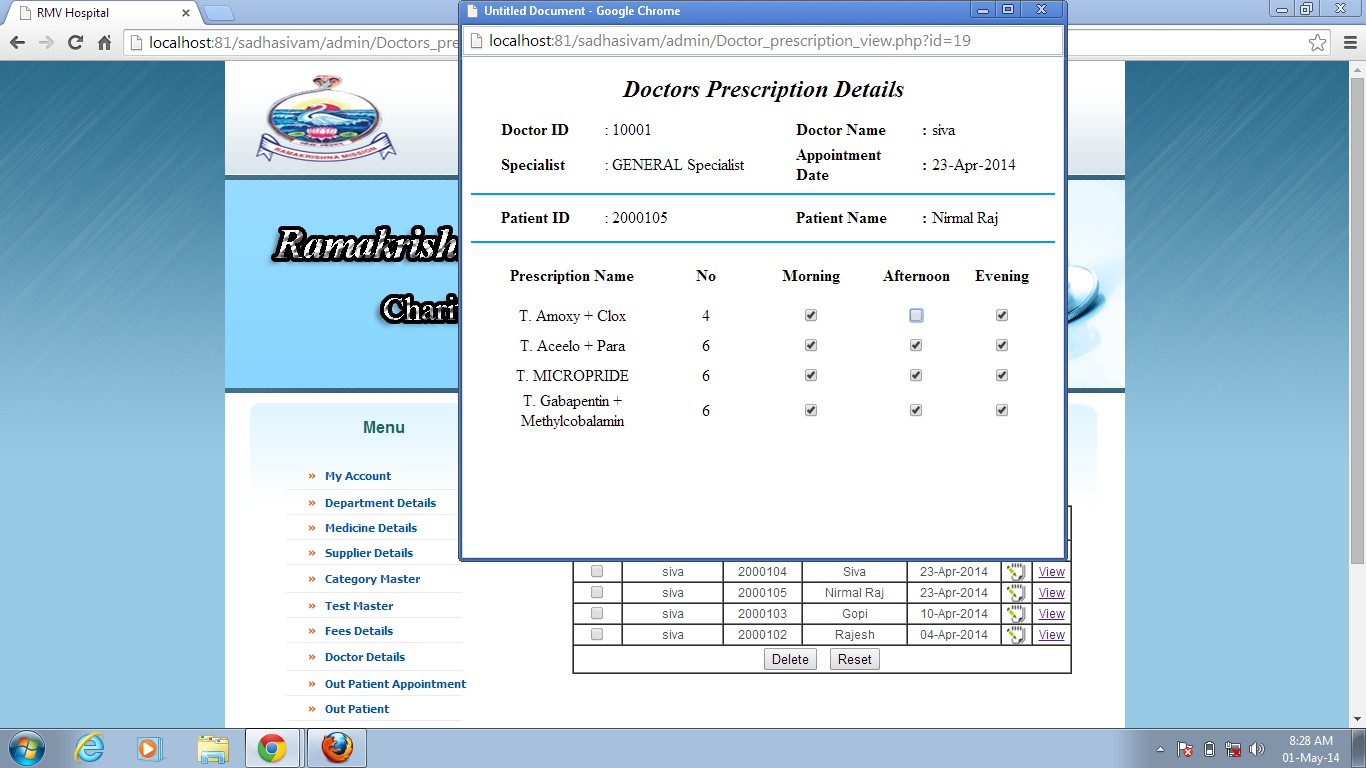
#### Doctor Prescription Add Details:

****

**Description:**

This Screen is doctor prescription page of the Online RMV Hospital, in which the user can add the information medicine details.

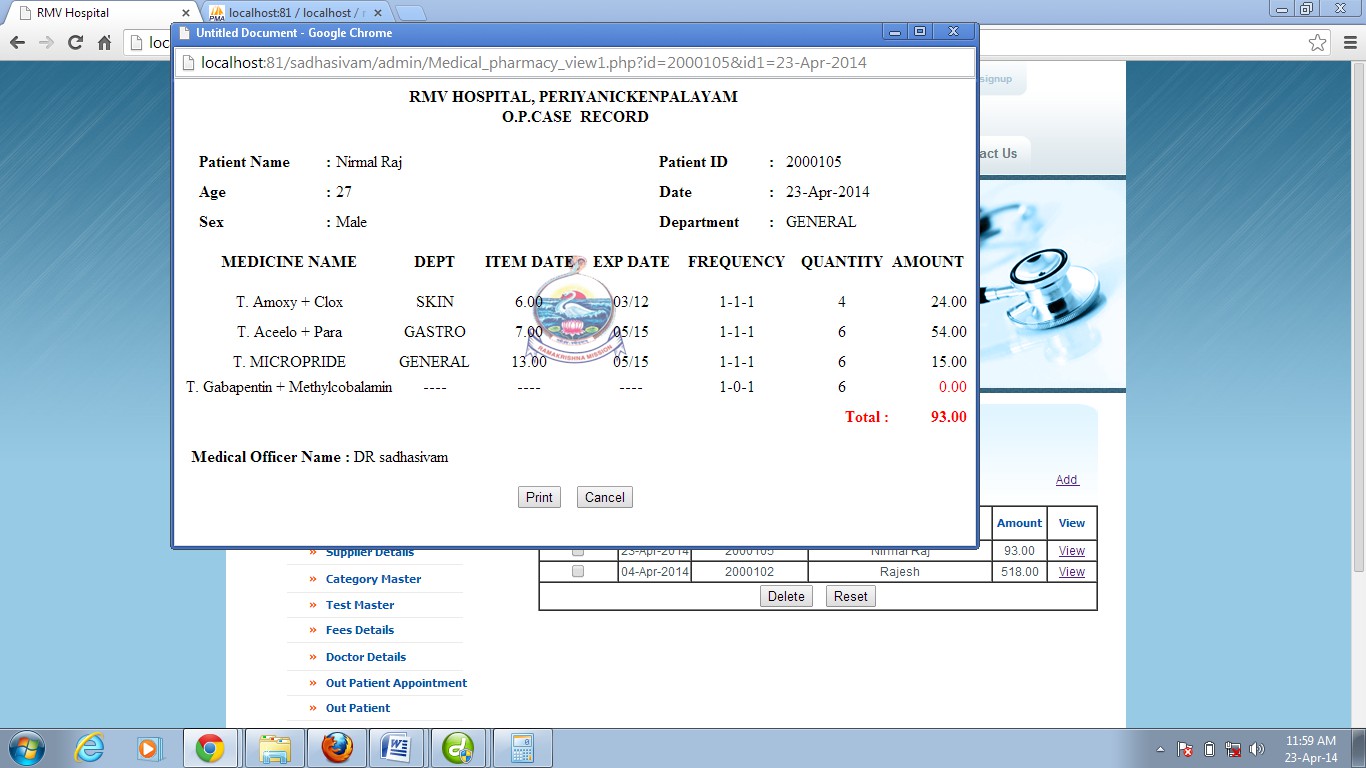
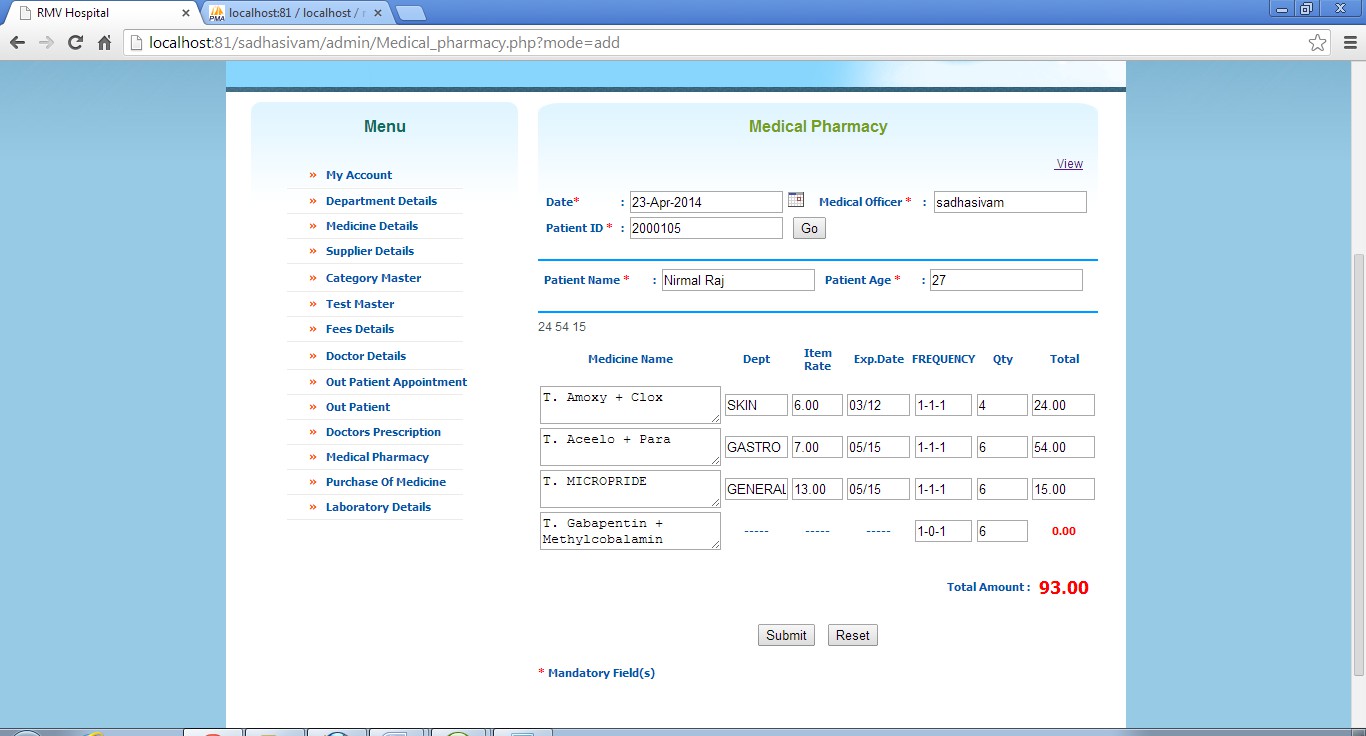
#### Doctor Prescription View Details:

****

**Description:**

This Screen is doctor prescription of the Online RMV Hospital, in which the user can view the entire information about the doctor prescription view details.

#### Medical Pharmacy:

****

**Description:**

This Screen is Medical Pharmacy page of the Online RMV Hospital, in which the user can view the entire information about the medical pharmacy details.

#### Supplier Details:

****

**Description:**

This Screen is Supplier Details of the Online RMV Hospital, in which the user can view the particular type of Supplier name.

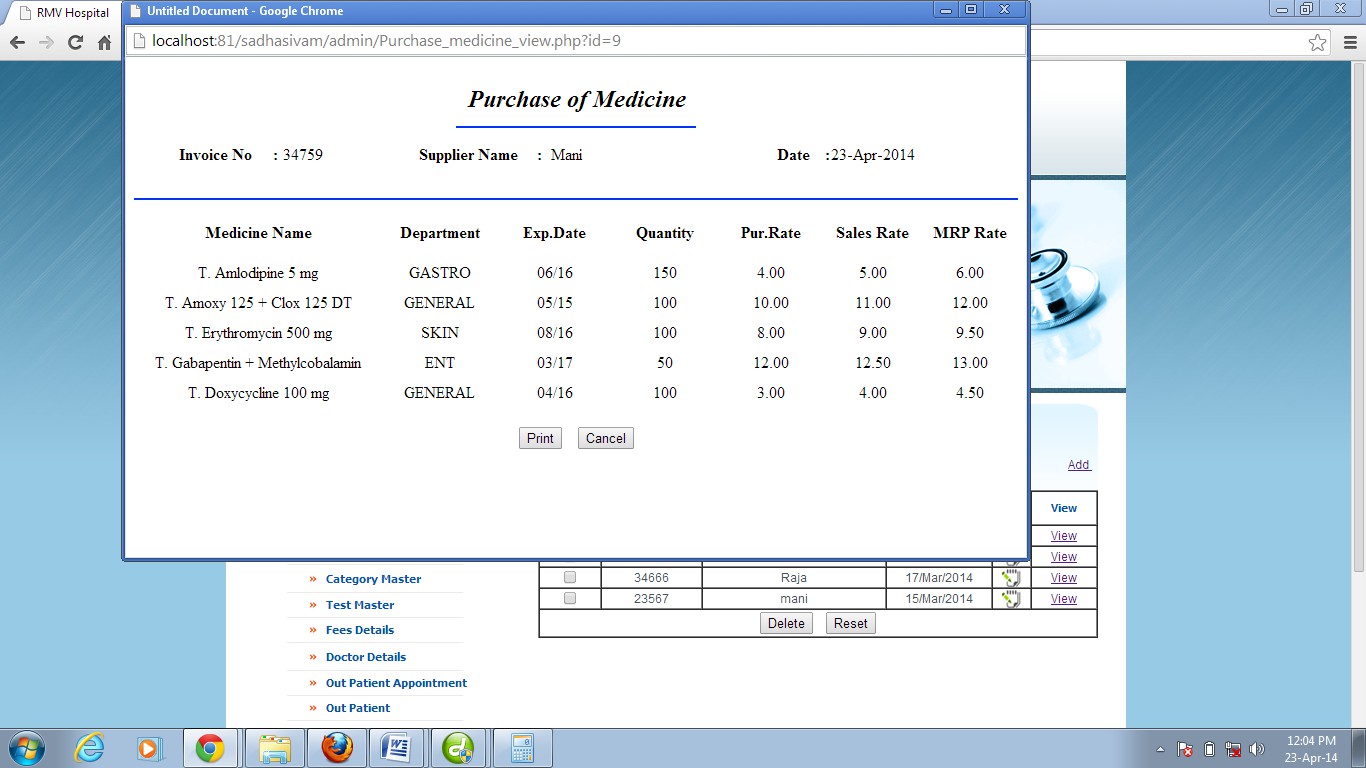
#### Purchase of Medicine Add Details:

****

**Description:**

This Screen is Purchase of Medicine page of the Online RMV Hospital, in which the user can add the information medicine details.

#### Purchase of Medicine View Details:

****

**Description:**

This Screen is Purchase of Medicine of the Online RMV Hospital, in which the user can view the entire information about the purchase of medicine view details.

#### Category Master:

****



**Description:**

This Screen is Category Details of the Online RMV Hospital, in which the user can view the particular type of Category name.

#### Test Details:

****



**Description:**

This Screen is Test Details page of the Online RMV Hospital, in which the user can view the entire information about the test details.

#### Laboratory Details:

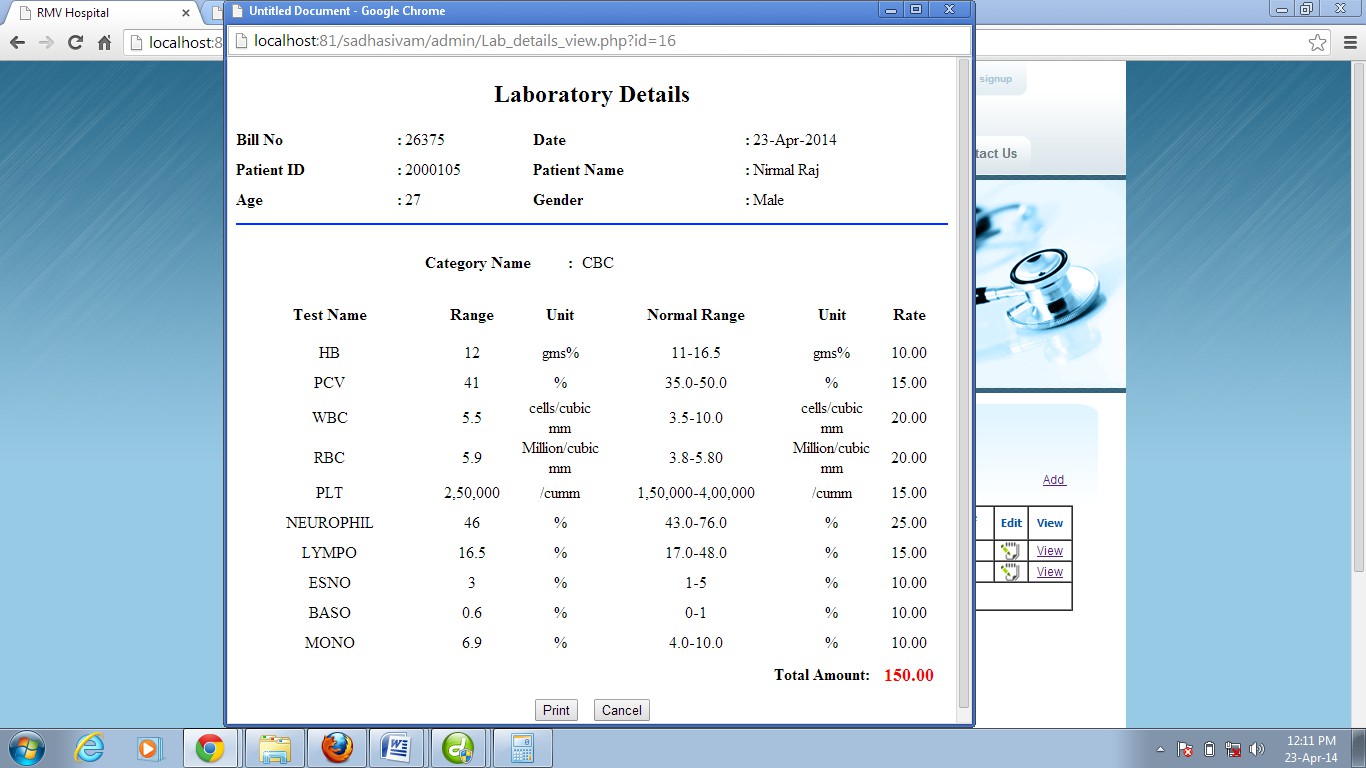
****



**Description:**

This Screen is Laboratory Details page of the Online RMV Hospital, in which the user can view the entire information about the Laboratory details.

#### Laboratory Billing Details:



**Description:**

This Screen is Laboratory Details page of the Online RMV Hospital, in which the user can view the entire information about the Laboratory Billing details.

#### Doctor Report:

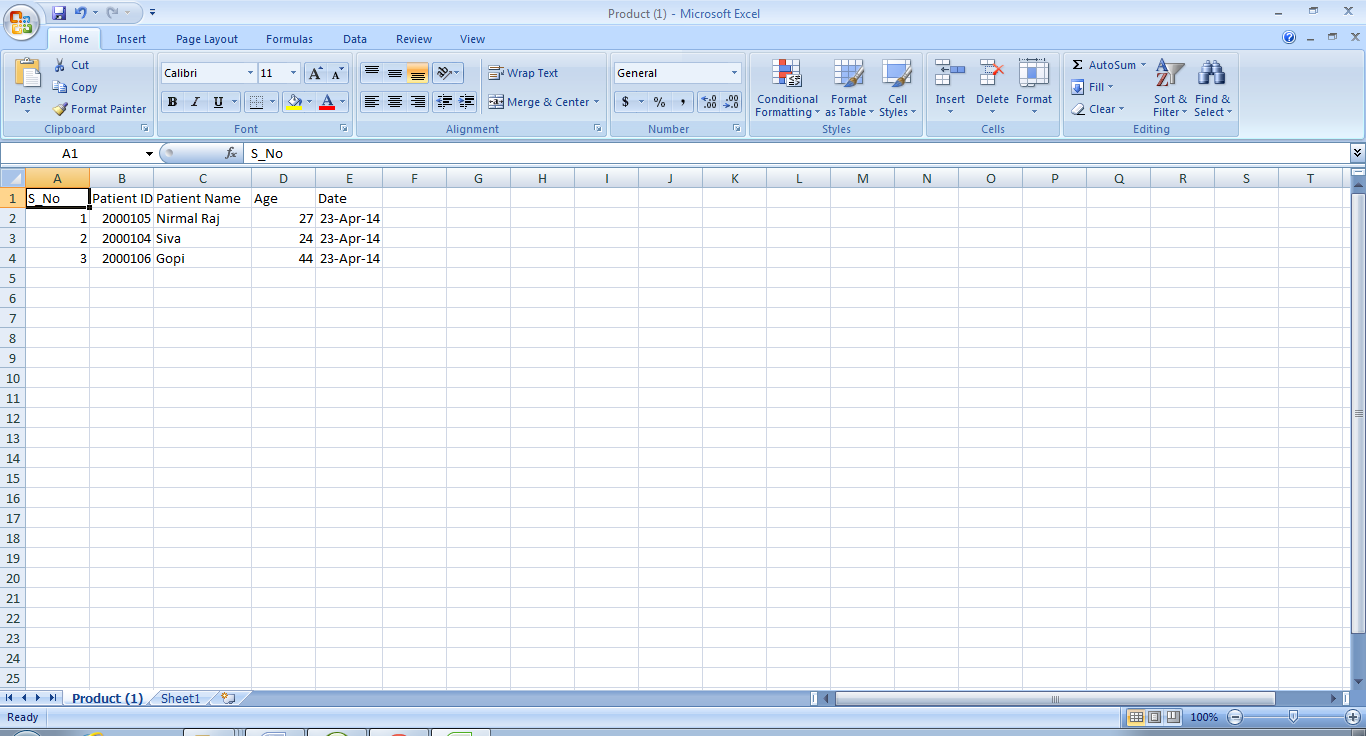
****



**Description:**

This Screen is Doctor Report Detail’s page of the Online RMV Hospital, in which the user can view the entire information about the doctor report details.

#### Doctor Report Excel Page:



**Description:**

This Screen is Doctor Report excel sheet page of the Online RMV Hospital.

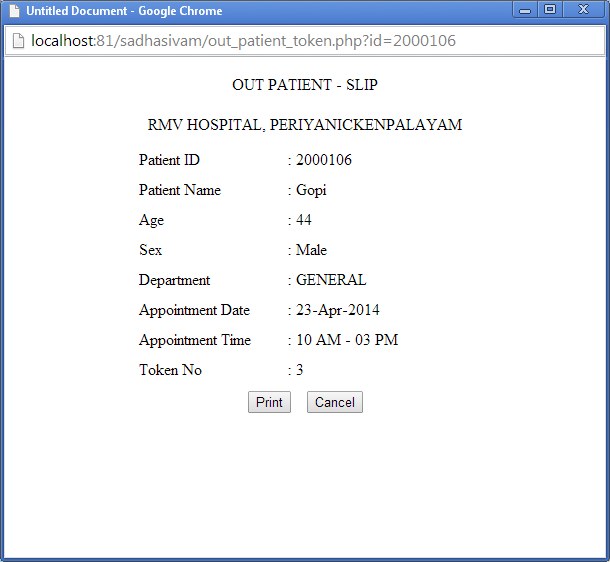
#### Home Page:



**Description:**

This Screen is Homes page of the Online RMV Hospital.

#### Doctor Appointment:

****



**Description:**

This Screen is doctor Appointment of the Online RMV Hospital, in which the user can view the particular type of Token Number and Billing Details.

#### Doctor View Details:

****



**Description:**

This Screen is doctor View details of the Online RMV Hospital, in which the user can view the entire information about the doctor details.

#### Doctor Information Details:



**Description:**

This Screen is doctor details of the Online RMV Hospital, in which the user can view the doctor full information display.

#### Laboratory Details:

****



**Description:**

This Screen is Laboratory Details page of the Online RMV Hospital, in which the user can view the entire information about the Laboratory details and download for patient test report.

.

#### Medicine Details:

****



**Description:**

This Screen is Medical Details page of the Online RMV Hospital, in which the user can view the entire information about the medical details and download for patient medicine report.

# CONCLUSION AND FUTURE ENHANCEMENT

## Conclusion

* Hospital Management System not only provides an opportunity to the hospital to enhance their patient care but also can increase the profitability of the organization
* Hospital Management System would enable hospitals or Nursing Homes to serve the rapidly growing number of health care consumers in a cost-effective manner
* Hospital Management System can also save extra money on your current computer hardware shopping. Check up with our executive to more on this
* Hospital administrators would be able to significantly improve the operational control and thus streamline operations
* This would enable to improve the response time to the demands of patient care because it automates the process of collecting, collating and retrieving patient information

Very important for some, the reduced cost of the manpower would pay for the cost of this product within a short time after its implementation

## Future Enhancement

The proposed system is Hospital Management System. We can enhance this system by including more facilities like billing system, inpatient room allotment for the admitted patients . Providing such features enable the users to include more comments into the system.

# BIBLIOGRAPHY

### BOOKS:

* PHP A Beginner’s Guide --- Dave Mercer Herbert Schildt (2000).
* HTML 4 Complete References.
* Wiley publications for the complete reference of PHP.

### ONLINE REFERENCE:

* [www.w3schools.com](http://www.w3schools.com/)
* [www.theserverside.com](http://www.theserverside.com/)
* [www.php.net](http://www.php.net/)