**INTRODUCTION**

**ABSTRACT:**

A coffee shop has always been more than a place to have a beverage. It is a place to socialize, ideate, have fun and meet new people. As starbucks puts it, it’s the third place, a home away from home or office. Also, due to competitive and busy lifestyle that people lead today, a coffee house near their work or home has become a major point of relaxation.

Coffee Shop management system is general web application software developed particularly for coffee shop transaction.

In big coffee shops or restaurants maintaining product details and billing will be complicated one, because in peak time like evening crowd will be more. It will be challenging task for both administration and preparation team. Many coffee shops face difficulty in billing and arranging tables for customers. Normally when customers visit coffed shop as family arranging table will be complicated which leads customer unhappy and service point will not be a good management. The project titled as “**COFFEE SHOP MANAGEMENT SYSTEM**” is a web based application. This software provides facility for adding new items, booking tables and billing facilities.

This application developed in such a way administrator or billing person can access. Items can be added dynamically through concern person login. Item details and billing information can see by single click.

The Coffee Shop Management System project is to provide all coffee shop solutions which are easily accessible for maintain items and billing. This application serves for administrator or cashier of the shop. Web based application is basically multiuser application, in distributed application environment gives efficient service. Admin can add new coffee names and details like name, type, price and descriptions including. Those details will be available for admin to edit and update as well delete. Admin can do billing by entering items what customer’s required and print bill instantly. And for special occasion, customers can book table for meeting. This application saves time and more efficient.

**Statement of Problem:**

A problem statement is a clear description of the issue(s), it includes a vision, issue statement, and method used to solve the problem. The 5’W’s – Who, What, Where, When and Why – is a great tool that helps get pertinent information out for discussion.

So in our application we solved the problem of adding, maintaining details and billing. Here “WHO” points admin or cashier, as normally in coffee shop. So their productivity is very important for company growth and service of the company. For booking tables’ manual methods will be time consuming methods which waste productivity and stressful method for both customer and admin. We developed a project which saves employee working time and increase productivity.

The Next main point is “WHAT”, because of this application what is benefit and what improvement we are going to get. We save manual paper work and ledgers of resources and online updating systems. Resources information’s are maintained and updates very properly by concern person. It avoids the problem inside shop and makes professional process of order and billing.

This application can be used in small shops to bigger shops. Because orders, billing, and maintaining details of all items create lot of problems and it leads to lot of clashes and waste of time. So there is no boundary size for usage. The application made process more professional and flexible. In future can extend this application to mobile, so that people get their bills from their tables itself. We can extend this in working place kind of tech parks, so that employees can save their time to reach cafes.

This is web based application and we used open source technologies here. In real time hosting cost and maintainability cost is very less and user friendly. Application is going to save lot of money in-terms of purchasing and maintain. Because of centralized application environment data are stored in the server, doesn’t affect local storage and avoid the unnecessary data loss.

**Objective:**

The main objective of the application is to automate the existing system of manually maintain the records of available items, table booking details, billing details information to be computerized. So the searching product, updating information, booking and cancelling tables all will be faster. The objective of this application is to maintain a centralized repository of information related to coffee shop such as items, table’s availability, billing details and etc. The purpose is to enable quickly access the information and take appropriate actions regarding sales. The items information are added by admin and managed by him. With the help of this admin can easily check the availability of tables and book tables for customer. So all the processes of booking, billing done very easily. Admin can search and view the item details and information. Admin can add new tables and manipulate easily.

* To build a system that can receive input and generate automatically output in easy way and short time.
* To build a monitoring system that is able to monitor and manage all booking and selling operations efficiently.
* Give an opportunity to management to reduce mistakes that always happen during manual method.
* To design a user friendly graphical user interface which suit the users so that it increase the usability.
* To build an application that can increase productivity and maintainability in company.

**Methodology:**



Waterfall model is one of a system development life cycle(SDLC) model. Users proceed to next phase if and only if current phase is complete. Users are not allowed go back to previous phases if there are any mistake so they named it waterfall model, just like the water is always fall down from the waterfall and not flow upward.

In Royce's original waterfall model, the waterfall model originally consists of 7 phases which is Requirement Specification, Design, Construction, Integration, Testing and Debugging, Installation and Maintenance.

Firstly, we collect the requirement for the booking system, and then we analyzed it. After analyzed the requirement, we proceed to design stage. In the design phase, we design not only the user interface, but also the database design. The next phase in waterfall model after design phase is Construction phase. Construction phase is an important phase in waterfall model and it is a time consuming phase depends on programmer’s ability. In Online Web Meeting Booking development, construction phase is using PHP coding to write the program. It is very time consuming if the programmer don’t understand the logic or still fresh to the coding. After done the coding phase, we will proceed to integration phase. In this phase, online web site and library system will share a same database to make integration between two applications. The next phase after integration is the testing and debugging phase. For testing module, it is separated into few types which are module testing, system testing, unit testing and user acceptance test. Once there is a bug founded, it will be solve immediately before the system is launched to ensure the application launched is bug free. Lastly, it is implementation and maintenance phase. In this phase, the application will be installed at user side. After deployed the application, maintenance is compulsory needed to ensure the application is always-on and up to date with latest technologies or latest business process. In my opinion, the time spent on earlier phases of SDLC can lead to greater economy in later stages. It is because in the earlier phase, a bug can be fixed in short time, less cost and less effort compared to later phases

#### ****Advantages of Waterfall Model:****

* Waterfall model is very simple and easy to understand and use a method that is why it is really beneficial for the beginner or novice developer
* It is easy to manage, because of the rigidity of the model. Moreover, each phase has specific deliverables and individual review process
* In this model phases are processed and completed are at once in a time thus it saves a significant amount of time
* This type of development model works more effectively in the smaller projects where requirements are very well understood
* The testing is easier as it can be done by reference to the scenarios defined in the earlier functional specification

#### **Disadvantages of Waterfall Model:**

* This model can only be used when very precise up-front requirements are available
* This model is not applicable for maintenance type of projects
* The main drawback of this method is that once an application is in the testing stage, it is not possible to go back and edit something
* There is no possibility to produce any working software until it reaches the last stage of the cycle
* In this model, there is no option to know the end result of the entire project
* This model is good for a small project but not ideally suitable for long and ongoing projects
* Not ideal for the projects where requirements are very moderates, and there is great scope for modification

**Limitation of the project:**

* The system requires very low system resources and the system will work in almost all configurations.
* It can be used for small tier as well as big level companies also.
* It is a web based application so can be used in internet or intranet environment.
* All are open source technologies, so it will not cost more for set up.
* This is responsive application so anybody can access this in mobile devices.
* If we host this in intranet only devices can access which all connected in network.
* With Internet only can access when we hosted in main server.

**Literature Survey:**

The general problem faced by shop administrator in shop is finding details and prices, booking tables for meeting and checking availability of tables. Unfortunately, in big shops, finding products details and prices, checking tables with concern person manually or over through phone is complicated task.

Mostly admin rely on problems to find the available tables and prices of items. This is time consuming and needs a lot of time and effort. In case of tables are not available for booking it leads to business loss or problems. Coffee shop management system is a user-friendly web portal which combines all the above said features to provide a on the go solution for searching, booking and updating, billing etc just by a search over the distributed internet.

The existing system is being done manually. Each process is entered by manual means. The process involved in creating a particular report consumes such time, since searching through the manual record involves more manual activity which is a time consuming process. The process can be prone to resulting in errors and in duplication of data. Apart from tedious process, there is a chance for missing data and reports.

**Limitations of Existing System:**

* Existing system is a Manual method.
* The time involved in retrieving any information takes long.
* Lots of papers has be maintained
* Time delay
* Wastage of human resources
* Record maintenance is difficult
* Possible of mistakes in calculating amount.

**PROPOSED SYSTEM:**

This project helps to maintain the details of the coffee items, tables, booking details and billing details in full fledged security. Unauthorized persons cannot access the data. The complete process is kept online, so that there is fast completion of the transaction process available.

* The current system is developed by web based application.
* No need for manual work.
* The retrieval time is quick and fast
* Account maintainability is very easy.
* User mistakes can avoid.

**This project contains the following Modules:**

* Home
* About
* Contact
* Admin Login
* Dashboard
* Add Coffee
* Add Sell
* Add Table
* Book Table
* Coffee Report.
* Sell Report.
* Booking Report.
* Change Password.
* Logout.

**SYSTEM ANALYSIS**

**SYSTEM ANALYSIS**

## Feasibility Study:

It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Months or years of effort, thousands and millions of dollars, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase.

Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. During product engineering, however, we concentrate our attention on four primary areas of interest.

* **Technical Feasibility:**

Technical feasibility is the need of hardware and software, which are needed to implement the proposed system in the organization. Technical requirements are to be fulfilled to make the proposed system work. This should be necessarily predetermined so as to make the system more competent.

* **Economical Feasibility:**

The Economical feasibility must satisfy the needs of the technical feasibility and the operational feasibility. It involves the economic feasibility of developing and implementing the proposed system.

* **Operational Feasibility:**

The proposed system should use the internet level then the different types of end users are involved in the system, so it solves the user’s needs and the organization needs. And it supports the all users environment.

**SYSTEM SPECIFICATIONS**

**HARDWARE SPECIFICATIONS:**

* + PROCESSOR : Intel Pentium IV
  + PROCESSOR SPEED : 2.66GHz
  + MAIN MEMORY : 252 MB RAM
  + DISPLAY DEVICE : Plug and Play Monitor
  + KEY BOARD : 108 KEYS
  + FLOPPY DISK : 1.44MB
  + MOUSE : PS/2 Compatible Mouse

**SOFTWARE SPECIFICATIONS:**

* Client Side Script : HTML, CSS, JavaScript.
* Server Side Script : PHP
* Server Side Database : MySql

**Technologies Used:**

**TOOLS AND TECHNOLOGIES**

* Client side Scripting : HTML5, CSS3, JS, JQUERY
* Server side Scripting : PHP
* Database : MySQL server
* Development Environment : Dreamweaver CS3
* Operating system : windows 7

**HTML**

HTML abbreviates hyper text markup language. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012. Recent version is HTML5.

* HTML stands for **H**yper **T**ext **M**arkup **L**anguage
* HTML is a **markup** language
* A markup language is a set of markup **tags**
* The tags **describe** document content
* HTML documents contain HTML **tags** and plain **text**
* HTML documents are also called **web pages**
* It runs on all browser
* It is used to construct a web page

It is not a programming language. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.

**CSS (Cascading style sheet)**

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[[1]](http://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-1) This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work
* External Style Sheets are stored in **CSS files**

**JavaScript**

Java Script is the scripting language of the Web. JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more.

JavaScript, despite the name, is essentially unrelated to the Java programming language even though the two do have superficial similarities. Both languages use syntaxes influenced by that of C syntax, and JavaScript copies many Java names and naming conventions. The language's name is the result of a co-marketing deal between Netscape and Sun, in exchange for Netscape bundling Sun's Java runtime with their then-dominant browser. [Citation needed] The key design principles within JavaScript are inherited from the self and Scheme programming languages.

"JavaScript" is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape Communications and current entities such as the Mozilla Foundation.

Due to the widespread success of JavaScript as a client-side scripting language for web pages, Microsoft developed a compatible dialect of the language, naming it JScript to avoid trademark issues. JScript added new date methods to fix the non-Y2K-friendly methods in JavaScript, which were based on java.util.Date.[5] JScript was included in Internet Explorer 3.0, released in August 1996. The dialects are perceived to be so similar that the terms "JavaScript" and "JScript" are often used interchangeably. Microsoft, however, notes dozens of ways in which JScript is not ECMA-compliant.

**What is JavaScript?**

* JavaScript was designed to add interactivity to HTML pages
* JavaScript is a scripting language
* A scripting language is a lightweight programming language
* JavaScript is usually embedded directly into HTML pages
* JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
* Everyone can use JavaScript without purchasing a license

**What can a JavaScript do?**

* JavaScript gives HTML designers a programming tool - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
* JavaScript can put dynamic text into an HTML page - A JavaScript statement like this: document. Write("<h1>" + name + "</h1>") can write a variable text into an HTML page
* JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
* JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element
* JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
* JavaScript can be used to detect the visitor's browser - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
* JavaScript can be used to create cookies - A JavaScript can be used to store and retrieve information on the visitor's computer.

## *The Real Name is ECMA Script*

* JavaScript's official name is ECMA Script.
* ECMA Script is developed and maintained by the ECMA organization.
* ECMA-262 is the official JavaScript standard.
* The language was invented by Brendan Erich at Netscape (with Navigator 2.0), and has appeared in all Netscape and Microsoft browsers since 1996.
* The development of ECMA-262 started in 1996, and the first edition of was adopted by the ECMA General Assembly in June 1997.
* The standard was approved as an international ISO (ISO/IEC 16262) standard in 1998.
* The development of the standard is still in progress.

# JavaScript How To:-

The HTML <script> tag is used to insert a JavaScript into an HTML page.

<Html>  
<body>  
<script type=”text/JavaScript”>

document.Write("HelloWorld!") ;  
</script>  
</body>  
</html>

To insert a JavaScript into an HTML page, we use the <script> tag. Inside the <script> tag we use the type attribute to define the scripting language.So, the <script type="text/JavaScript"> and </script> tells where the JavaScript starts and ends:

## *Browsers Support:*

* Browsers that do not support JavaScript will display JavaScript as page content.
* To prevent them from doing this, and as a part of the JavaScript standard, the HTML comment tag should be used to "hide" the JavaScript.
* Just add an HTML comment tag <! -- Before the first JavaScript statement, and a --> (end of comment) after the last JavaScript statement, like this:
* JavaScript in the body section will be executed WHILE the page loads.
* JavaScript in the head section will be executed when CALLED.

## *Where to Put the JavaScript?*

JavaScript in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

## *Scripts in <head>*

Scripts to be executed when they are called, or when an event is triggered, go in the head section. If you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

## *Scripts in <body>*

Scripts to be executed when the page loads go in the body section. If you place a script in the body section, it generates the content of a page.

## *Using an External JavaScript*

If you want to run the same JavaScript on several pages, without having to write the same script on every page, you can write a JavaScript in an external file. Save the external JavaScript file with a .js file extension. The external script cannot contain the <script> tag. To use the external script, point to the .js file in the "src" attribute of the <script> tag:

# Overview of PHP

**History**

The origins of PHP date back to 1995, when an independent software development contractor named Rasmus Lerdorf developed a Perl/CGI script that enabled him to know how many visitors were reading his online résumé. His script performed two tasks: logging visitor information, and displaying the count of visitors to the Web page. Because the Web as we know it today was still young at that time, tools such as these were nonexistent, and they prompted e-mails inquiring about Leadoff’s scripts. Lerdorf thus began giving away his toolset, dubbed Personal Home Page (PHP).

The clamor for the PHP toolset prompted Lerdorf to begin developing additions to PHP, one of which converted data entered in an HTML form into symbolic variables that allowed users to export them to other systems. To accomplish this, he opted to continue development in C code rather than Perl. Ongoing additions to the PHP toolset culminated in November 1997 with the release of PHP 2.0, or Personal Home Page — Form Interpreter (PHP-FI). As a result of PHP’s rising popularity, the 2.0 release was accompanied by a number of enhancements and improvements from programmers worldwide.

The new PHP release was extremely popular, and a core team of developers soon joined Lerdorf. They kept the original concept of incorporating code directly alongside HTML and rewrote the parsing engine, giving birth to PHP 3.0. By the June 1998 release of version 3.0, over 50,000 users were using PHP to enhance their Web pages.

Development continued at a hectic pace over the next two years, with hundreds of functions being added and the user count growing in leaps and bounds. At the beginning of 1999, Net craft reported a conservative estimate of a user base surpassing 1,000,000, making PHP one of the most popular scripting languages in the world. Its popularity surpassed even the greatest expectations of the developers, as it soon became apparent that users intended to use PHP to power far larger applications than was originally anticipated. Two core developers, Zeev Suraski and Andi Gutmans, took the initiative to spearhead a complete rethinking of the way PHP operated, culminating in a rewriting of the PHP parser, dubbed the Zend scripting engine. The result of this work was seen in the release of PHP 4.

# PHP - What is it?

Taken directly from PHP's home, “PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly."

This is generally a good definition of PHP. However, it does contain a lot of terms you may not be used to. Another way to think of PHP is a powerful, behind the scenes scripting language that your visitors won't see!

When someone visits your PHP webpage, your web server processes the PHP code. It then sees which parts it needs to show to visitors (content and pictures) and hides the other stuff (file operations, math calculations, etc.) then translates your PHP into HTML. After the translation into HTML, it sends the webpage to your visitor's web browser.

# PHP - What's it do?

It is also helpful to think of PHP in terms of what it can do for you. PHP will allow you to:

* Reduce the time to create large websites.
* Create a customized user experience for visitors based on information that you have gathered from them.
* Open up thousands of possibilities for online tools.
* Allow creation of shopping carts for e-commerce websites.
* HTML - Know the syntax and especially HTML Forms.
* Basic programming knowledge - This isn't required, but if you have any traditional programming experience it will make learning PHP a great deal easier.

**Usage of PHP**

In year 2000 there were only few hundred web sites developed in PHP. By year 2007 number increased to more than 2, 00, 000, 00 and number is increasing and increasing. Now a day’s most popular web site with heavy traffic has been developed in PHP.

PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems' Java Server Pages, and mod\_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symphony, Code Igniter, and Zend Framework, offering features similar to other web application frameworks.

The LAMP and WAMP architectures have become popular in the web industry as a way of deploying web applications. PHP is commonly used as the P in this bundle alongside Linux, Apache and MySQL, although the P may also refer to Python or Perl.

As of April 2007, over 20 million Internet domains were hosted on servers with PHP installed, and mod\_php was recorded as the most popular Apache module. Significant websites are written in PHP including the user-facing portion of Face book, Wikipedia (MediaWiki), Yahoo!, MyYearbook, Digg, Joomla, Word Press, You Tube, Drupal and Tagged.

# Features of PHP

**Opensource:**

PHP is a open source general purpose scripting language.

** Platform independence:**

The use of PHP adds versatility to a Web application by enabling its execution on any computer.

** Enhanced performance:**

The compilation process in PHP produces faster results or output.

**Fast:**

Because of built-in functions, developers can reduce the number of line coding. Through this PHP page execution become very fast.

 **Separation of logic from display:**

The use of PHP permits the HTML specific static content and a mixture of HTML, Javascript, and PHP specific dynamic content to be placed in separate files.

** Ease of administration:**

The use of PHP eliminates the need for high-level technical expertise, thereby helping Web developers, designers, content creators, and content managers to work together and develop Java-based applications in less time and with less effort.

** Ease of use:**

All PHP applications run on major Web servers and operating systems, including Microsoft IIS, Netscape Enterprise Server, iPlanet Web Server, and Apache Web Server. These applications are also available on Windows NT, Windows 2000, and Solaris 7.

**MySQL**

The Structured Query Language (SQL) is a very popular database language, and its standardization makes it quite easy to store, update and access data. One of the most powerful SQL servers out there is called MySQL and surprisingly enough, its free.

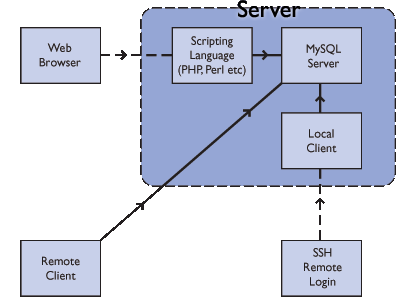
Some of the features of MySQL Include:

* Handles large databases, in the area of 50,000,000+ records.
* No memory leaks. Tested with a commercial memory leakage detector (purify).
* A privilege and password system which is very flexible and secure, and which allows host-based verification.
* Passwords are secure since all password traffic when connecting to a server is encrypted.

MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements. SQL is a database language that is used to retrieve, insert, delete and update stored data. This is achieved by constructing conditional statements that conform to a specific syntax (i.e. the strict order required of elements for a statement to work).

**How does MySQL work?**

MySQL is a database server program and as such is installed on one machine, but can 'serve' the database to a variety of locations. To explain look at the following diagram.



The MySQL Server is installed on a Server and can be accessed **directly** via various client interfaces, which send SQL statements to the server and then display the results to a user. Some of these are:

**A Local Client** – is a program on the same machine as the server. An example of this is the command line MySQL client software we will be using in the rest of the MySQL workshops (although there are other programs including graphical interfaces).

**A Scripting Language** - can pass SQL queries to the server and display the result.

**A Remote Client** – is a program on a different machine that can connect to the server and run SQL statements.

You can also use two more indirect methods.

**Remote Login** - You may be able to connect to the Server Machine to run one of its local clients.

**Web Browser** - you can use a web browser and scripts that someone has written (we're going to use this method for the rest of the workshop).

**Uses:**

Many web applications use MySQL as the database component of a LAMP software stack. Its popularity for use with web applications is closely tied to the popularity of PHP, which is often combined with MySQL. Several high-traffic web sites (including Flicker, Face book, Wikipedia, Google. (though not for searches), Nokia and You Tube) use MySQL for data storage and logging of user data.

**MySQL Architecture**

It will greatly aid your thinking about storage engines and the capabilities they bring to MySQL if you have a good mental picture of where they fit. Figure 2-1 provides a logical view of MySQL. It doesn’t necessarily reflect the low-level implementation, which is bound to be more complicated and less clear cut. However, it does serve as a guide that will help you understand how storage engines fit in to MySQL. (The NDB

storage engine was added to MySQL just before this book was printed. Watch for it in the second edition.)

**A logical view of MySQL architecture.**

The topmost layer is composed of the services that aren’t unique to MySQL. They’re services most network-based client/server tools or servers need: connection handling, authentication, security, etc.

The second layer is where things get interesting. Much of the brains inside MySQL live here, including query parsing, analysis, optimization, caching, and all the built-in functions (dates, times, math, encryption, etc.). Any functionality provided across storage engines lives at this level. Stored procedures, which will arrive in MySQL 5.0, also reside in this layer.

The third layer is made up of storage engines. They’re responsible for the storage and retrieval of all data stored “in” MySQL. Like the various file systems available for Linux, each storage engine has its own benefits and drawbacks. The good news is that many of the differences are transparent at the query layer.

The interface between the second and third layers is a single API not specific to any given storage engine. This API is made up of roughly 20 low-level functions that perform operations such as “begin a transaction” or “fetch the row that has this primary key” and so on. The storage engines don’t deal with SQL or communicate with each other; they simply respond to requests from the higher levels within MySQL.

* + **The Main Features of MySQL**

The following list describes some of the important characteristics of the MySQL Database Software.

**Internals and Portability**

* Written in C and C++.
* Tested with a broad range of different compilers.
* Works on many different platforms.
* Uses GNU Automake, Autoconf, and Libtool for portability.
* APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available.
* Fully multi-threaded using kernel threads. This means it can easily use multiple CPUs if they are available.
* Provides transactional and non-transactional storage engines.
* Uses very fast B-tree disk tables (MyISAM) with index compression.
* Relatively easy to add another storage engine. This is useful if we want to add an SQL interface to an in-house database.
* A very fast thread-based memory allocation system.
* Very fast joins using an optimized one-sweep multi-join.
* In-memory hash tables which are used as temporary tables.
* SQL functions are implemented using a highly optimized class library and should be as fast as possible. Usually there is no memory allocation at all after query initialization.
* The MySQL code is tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool.
* The server is available as a separate program for use in a client/server networked environment. It is also available as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

**SOFTWARE REQUIREMENT SPECIFICATION**

“**Coffee Shop Management System**” –has modules and sub modules in this application. This application inspired from few real time billing applications, but we included many more modules. Customized bill management and table booking features are included in this. In this application, we designed login page for Admin. Admin can add new items, add tables and will list details.

When the admin is login into this application by using unique username and password, he can see dashboard where we have quick links to reach important pages. After authentication only admin can login into his account and view items and book tables. Admin can login into account and can edit profile, change password in secure way. If the password is updated or not updated immediately admin account will logout for security reason. So that we can avoid the anonymous access of our application. Admin can view available coffee items and check the table availability on particular date. If the tables are available admin can book the table for any customer. The table booking information will be stored in database, and get the booking report. Admin can see the status and report through this account very easily; this saves a lot of time.

After login into account, admin can change his password. Admin can add new coffee items with follow details such as barcode, type of coffee, price and description. Admin can see coffee report and edit delete details from report page. Admin can add new table and admin can view table availability based on date. Admin can book table for customer based on required date. Admin can add billing for sales, which contains coffee name, price, quantity and total amount to pay. Complete sales details also stored in database, so that they can see the sales report.

**FUNCTIONAL REQUIREMENTS**

In Software development, a functional requirement declares and defines a function of an application or its component. A function is described as a set of inputs what need to given by end user, the behaviour which defines the functionality of each module, and outputs which describes the presentation what is the outcome. Functional requirements are like calculations, technical details, data manipulation and processing and other specific functionality that defines what an application or project is supposed to accomplish.

**Administrator**

* Login into Admin Account
* Add New coffee items
* View coffee report from Database
* Add New Tables to the shop
* Check table availability and booking table
* Add Sales details and generate bill
* View sales report
* Change account password

**NON-FUNCTIONAL REQUIREMENTS**

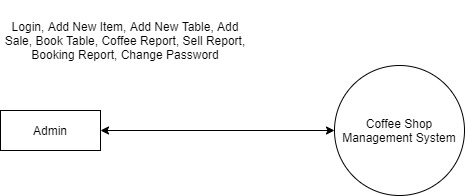
Basically web application projects runs on web server. Application speed and data rendering in user browser are decided by one of the common factor Internet connection in user machine. More over application is developed using one of the light weight and good responsive technology. And more over we are using html & css in this application which helps for responsive design which works in all devices and more user friendly. So it will be faster obviously. And data must be back-up on daily basis to prevent from un-excepted data corrupt and loss. And this application runs on Linux server basically cost less and virus free server.

The whole application doesn’t have any online payment options. All employee, Admin, Resource details are maintained very safe and secure from third party access through our application layer. More over all passwords are protected by built-in method of message digest. The application main server has the complete power backup. For large amount of size floor plans, master plan data and videos are segregated properly using drives and databases. And we are can use triggers to reduce the complexity of application layer and reduce the page traffic.

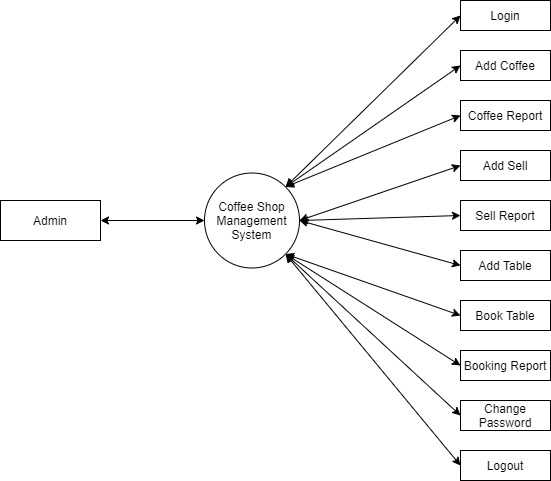
**SYSTEM DESIGN**

**DATA FLOW DIAGRAM:**

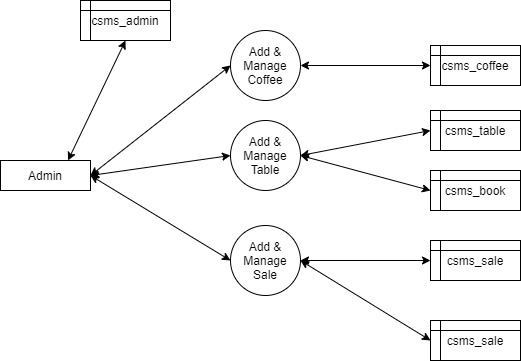
* **DFD LEVEL 0**



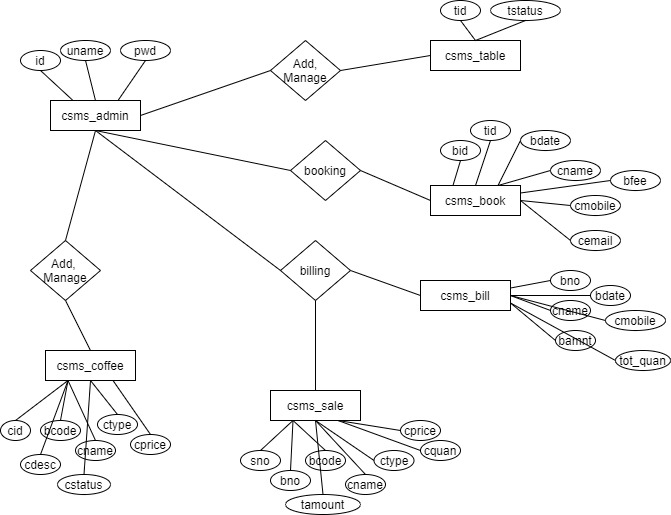
**DFD LEVEL 1:**

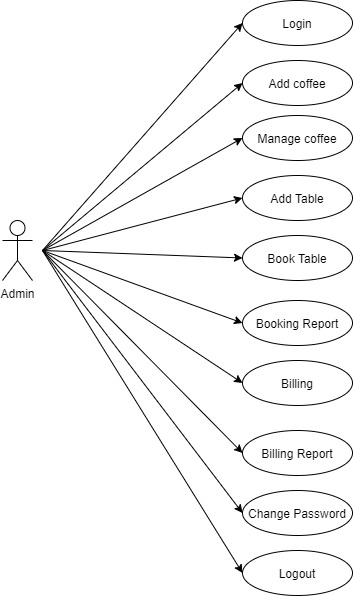


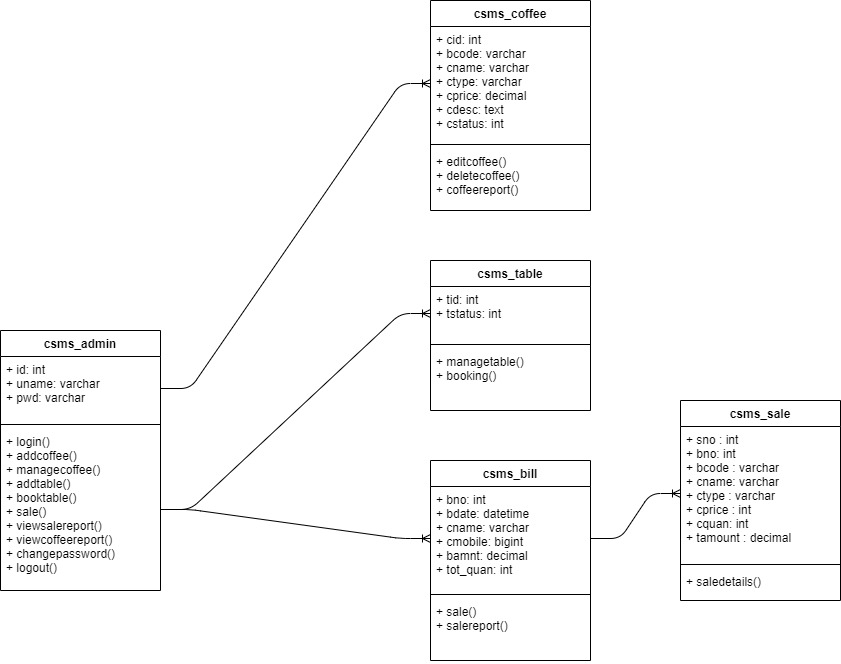
**DFD OF LEVEL 2:**

****

**ENTITY-RELATIONSHIP DIAGRAM:**

****





**DATABASE DESIGN**

**Table Design**

**csms\_admin**

|  |  |
| --- | --- |
| id | int primary\_key, auto\_increment |
| uname | varchar(30) unique |
| pwd | varchar(100) |

**csms\_coffee**

|  |  |
| --- | --- |
| cid | int primary\_key, auto\_increment |
| bcode | varchar(10) unique |
| cname | varchar(30) unique |
| ctype | varchar(30) |
| cprice | decimal(8,2) |
| cdesc | text |
| cstatus | int |

**csms\_table**

|  |  |
| --- | --- |
| tid | int primary\_key, auto\_increment |
| tstatus | Int |

**csms\_book**

|  |  |
| --- | --- |
| bid | int primary\_key, auto\_increment |
| tid | int |
| bdate | date |
| cname | varchar(30) |
| cmobile | bigint |
| cemail | varchar(30) |
| bfee | int |

**csms\_bill**

|  |  |
| --- | --- |
| bno | int primary\_key, auto\_increment |
| bdate | datetime |
| cname | varchar(30) |
| cmobile | bigint(20) |
| bamnt | decimal(8,2) |
| tot\_quan | int |

**csms\_sale**

|  |  |
| --- | --- |
| sno | int primary\_key, auto\_increment |
| bno | int |
| bcode | varchar(10) |
| cname | varchar(20) |
| ctype | varchar(30) |
| cprice | int |
| cquan | int |
| tamount | decimal(8,2) |

**INPUT AND OUTPUT DESIGN**

**Input Design:**

Input design is a part of overall system design. The main objective during the input designs the input designs is given bellow:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy
* To ensure that the input is acceptable and understood by the user.

**Input Stages:**

The main input stages can be listed below:

* Employee Details
* Client Details
* Bank Details
* Bank Criteria Details

**Input Types:**

It is necessary to determine the various types of inputs can be categorized as follows:

* External inputs, which are prime inputs for the system
* Internal inputs, which are user communication with the system.
* Operational, which are computer department’s communication to the system
* Interactive, which are inputs entered during a dialogue

**Input Media:**

At this stage choice has to be made the input media. To conclusion about the input media consideration has to be given to:

* Type of output
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**Error Avoidance**

As this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled

**Error Detection**

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors are always likely to occur, these types of errors can be discovered by using validation to check the input data

**Data Validation**

Procedures are designed to detect error in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with pop-up menus.

**User Interface design**

It is essential to consult the system users and discuss their needs while designing the user interface. The user interface system can be broadly classified as:

* User initiated interface the user is in charge. Controlling the computer-initiated interface, the computer selects the next stage in the interaction.
* Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and tJhe user response of the computer takes action or display further information.

**User Initiated Interface**

User initiated interface fall into two approximate classes:

* Command driven interfaces: in this type of interface the user inputs commands or queries which are interpreted by the computer.
* Forms oriented interface: the user calls up an image of the form to his/her screen and fills in the form. The forms oriented is chosen because it is the best choice.

**Computer-Initiated Interface**

The following computer-initiated were used:

* The menu system for the user is presented with a list of alternatives and the user chooses on: of alternatives.
* Question –answer type dialog where the computer asks question and takes action based on the basis of users reply.

Right from the start the system is going to be menu driven; the opening menu displays the available options. Choosing one option gives another popup menu with more option. In this way every option leads the users to data entry where the user can key in the data

**Error Message Design**

The design of the error massages is a important part of the user interface design. As user is bound to commit some error or other will design a system should be design to be helpful by providing the user information regarding the error he/she has committed. This application is able to produce the output at different modules of different inputs.

**Performance Requirement**

Performance is measured in terms of report generation weekly and monthly. Requirement specification is play important parts in the analysis of the system. Once the requirement specifications are properly given, it is possible to design the system, which will fit into the required environment. It rests largely in the par of the user of existing system to give the requirement specification because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

**Stage as given bellow:**

* The system to able to interface with the existing system
* The system is to be accurate
* The system to be better than the existing system

The existing system is completely dependent on the staff to perform all duties.

**Output Definition**

The output should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output

It is not always desirable to print to display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

**Output Media**

In the next stage it is to be decided that which medium is the most appropriate for the output. The main consideration when deciding about the output media are:

The suitability or the device to the particular application.

* The need for hard copy
* The response time required
* The location of the users
* The software and hardware available
* The cost

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

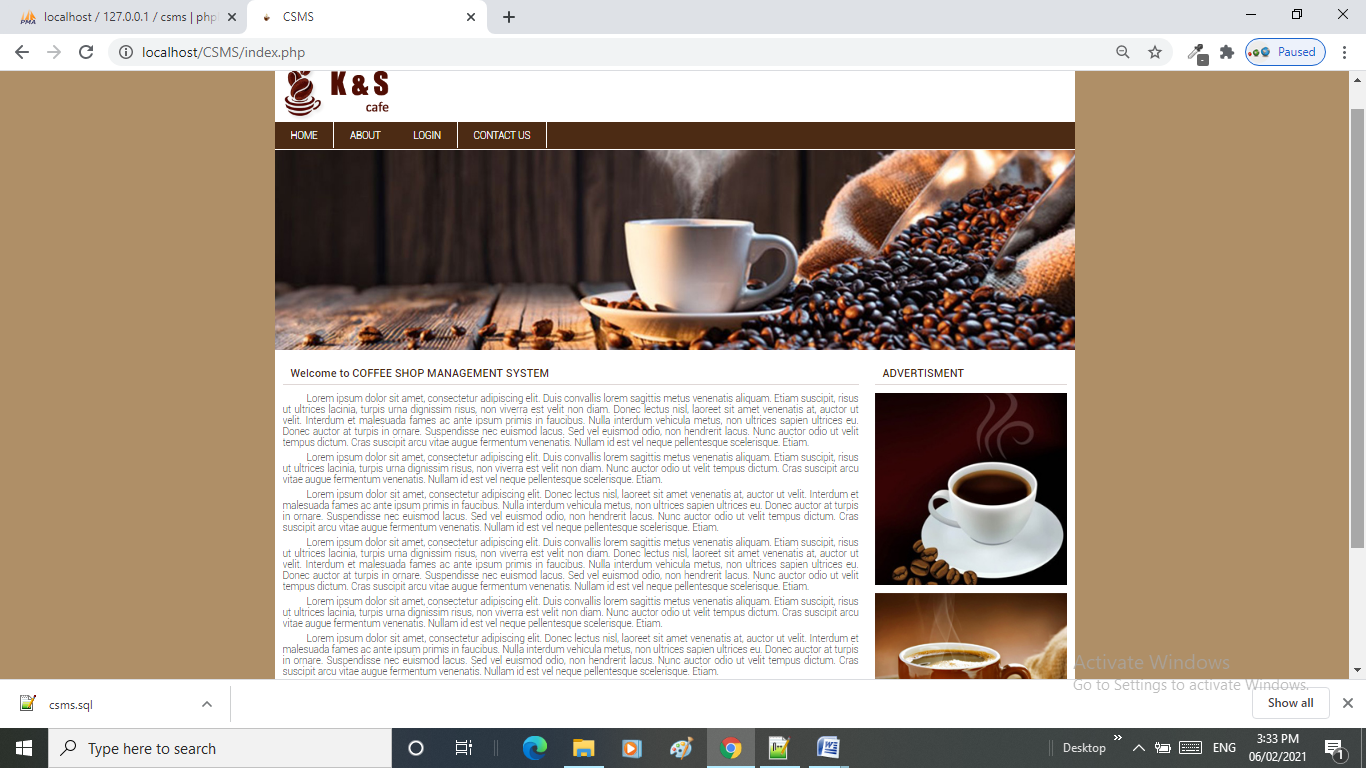
* The outputs were needed to be generated as a hot copy and as well as queries to be viewed on the screen.
* Keeping in view these outputs
* The format for the output is taken form the outputs
* Which are currently being obtained after manual processing
* The standard printer is to be used as output media for hard copies

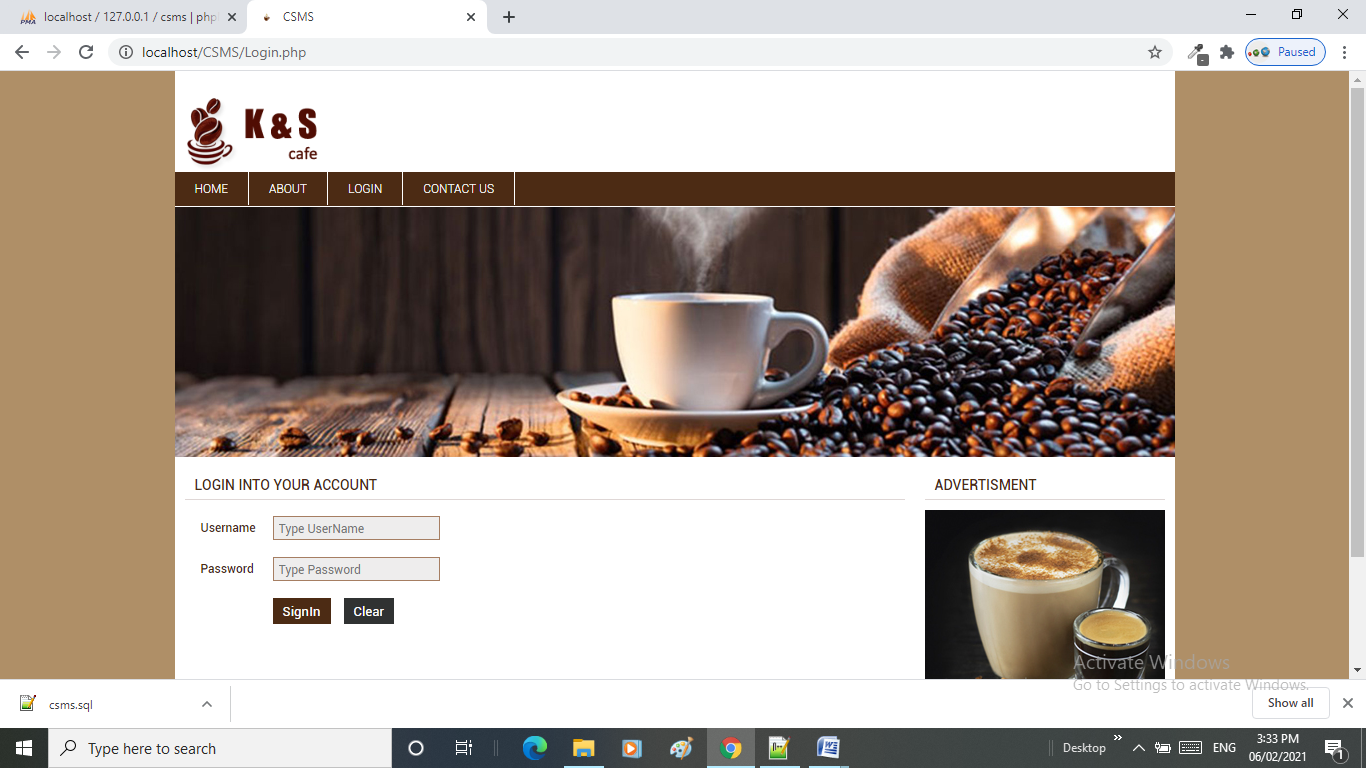
**Application of the Project :**

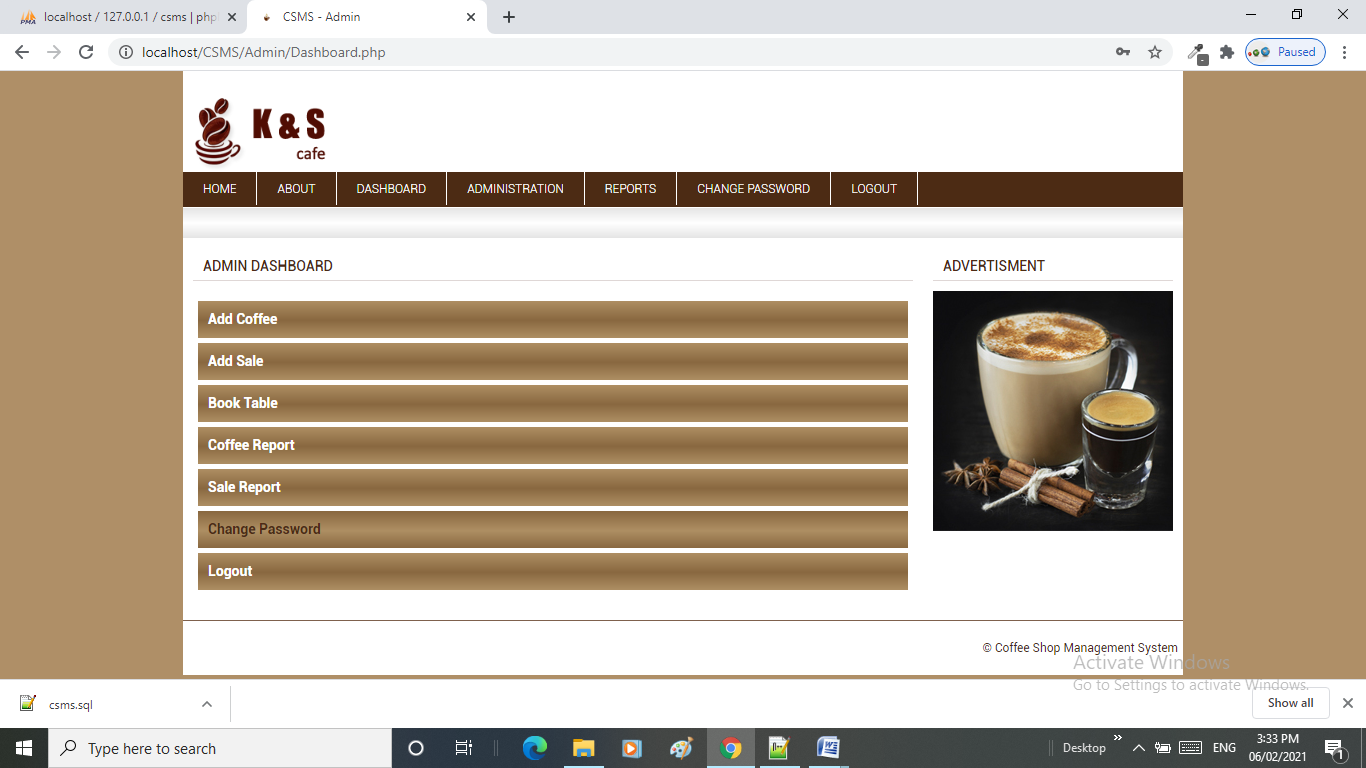
The system requires very low system resources and the system will work in almost all configurations. It has got following applications

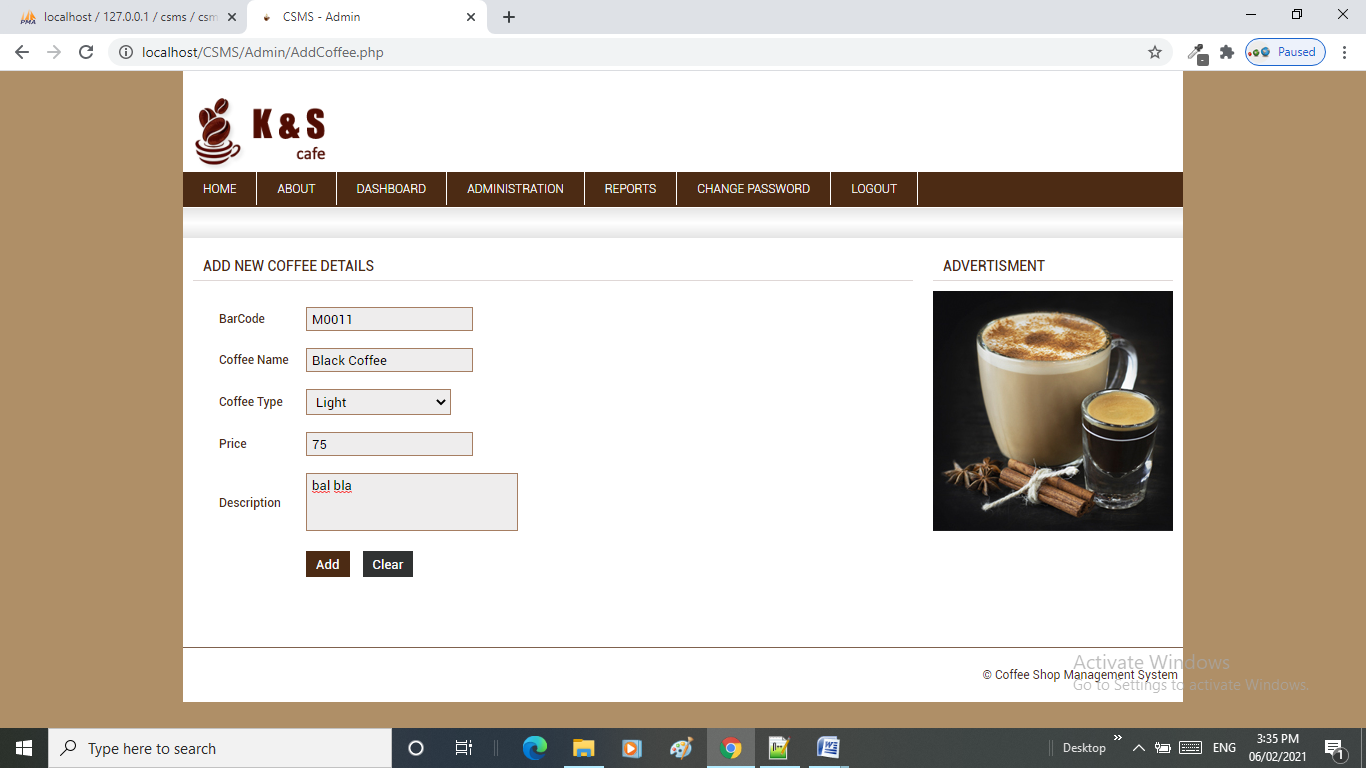
* Small and big level companies can handle data accurately
* Can use for commercial purpose with little modification
* Ensure data accuracy’s
* Avoiding redundancy
* Proper control of the higher officials
* User friendly and interactive

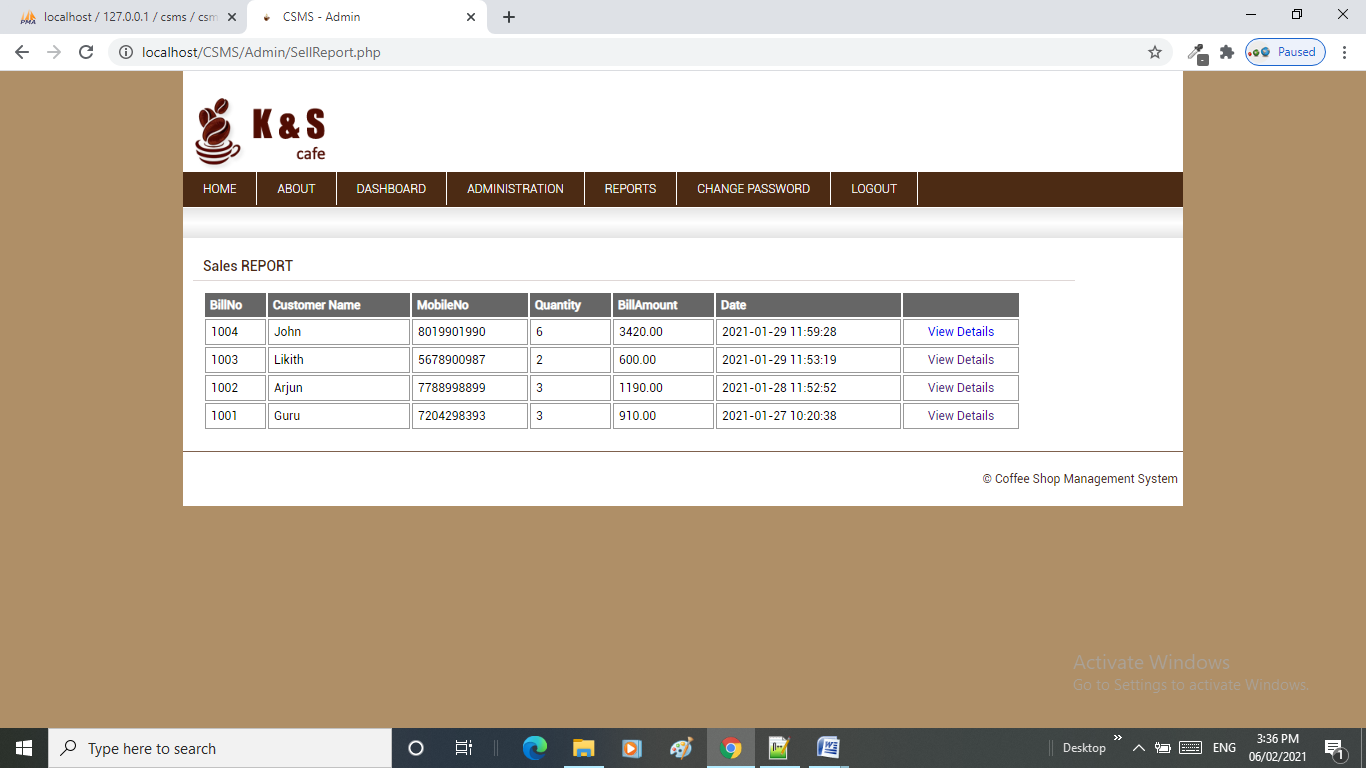
**SCREENSHOTS**

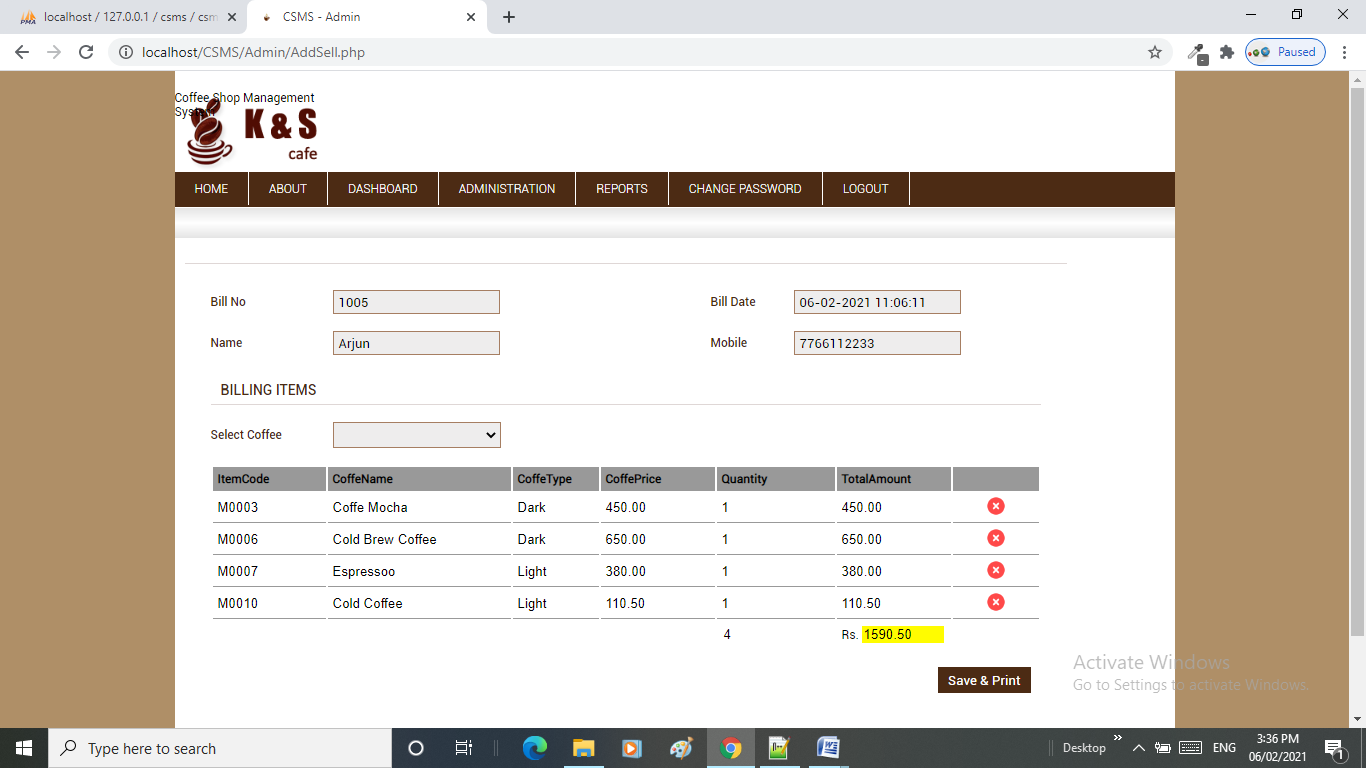
****

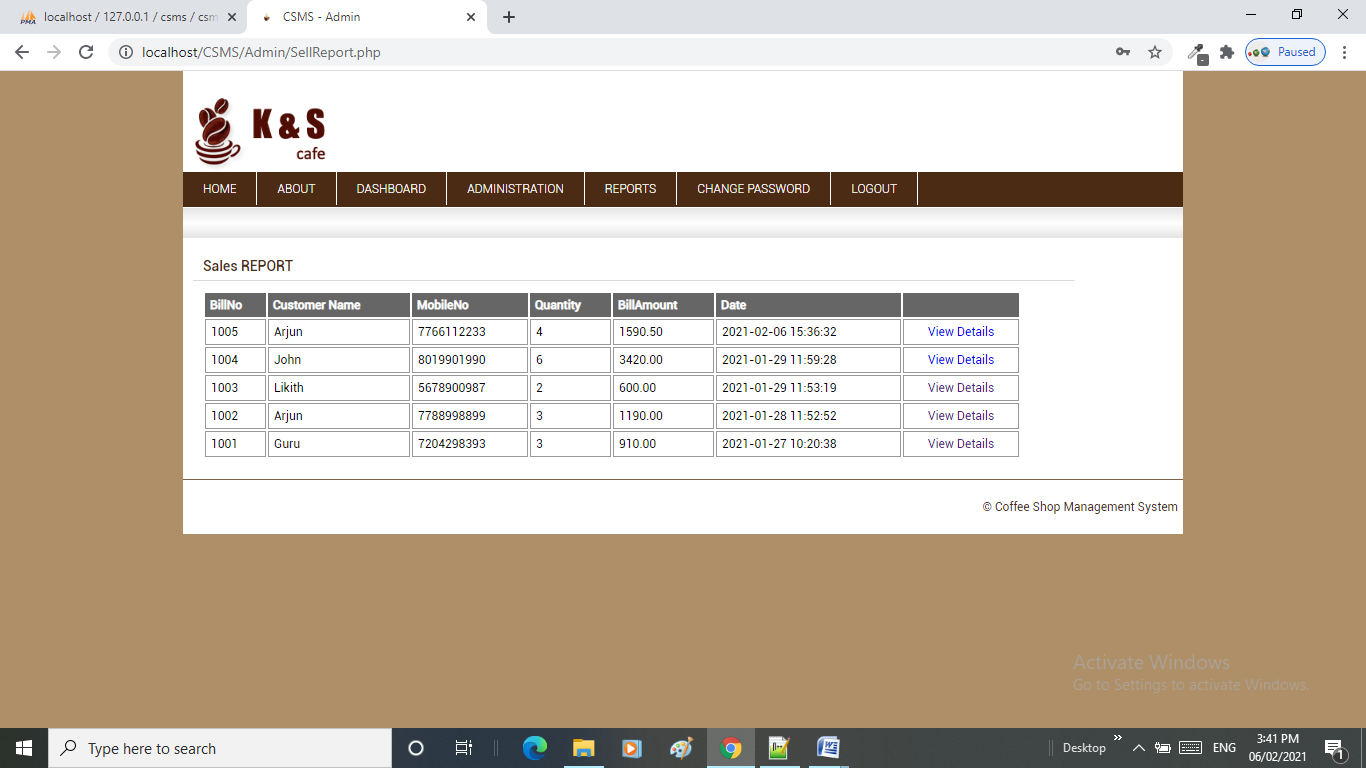
****

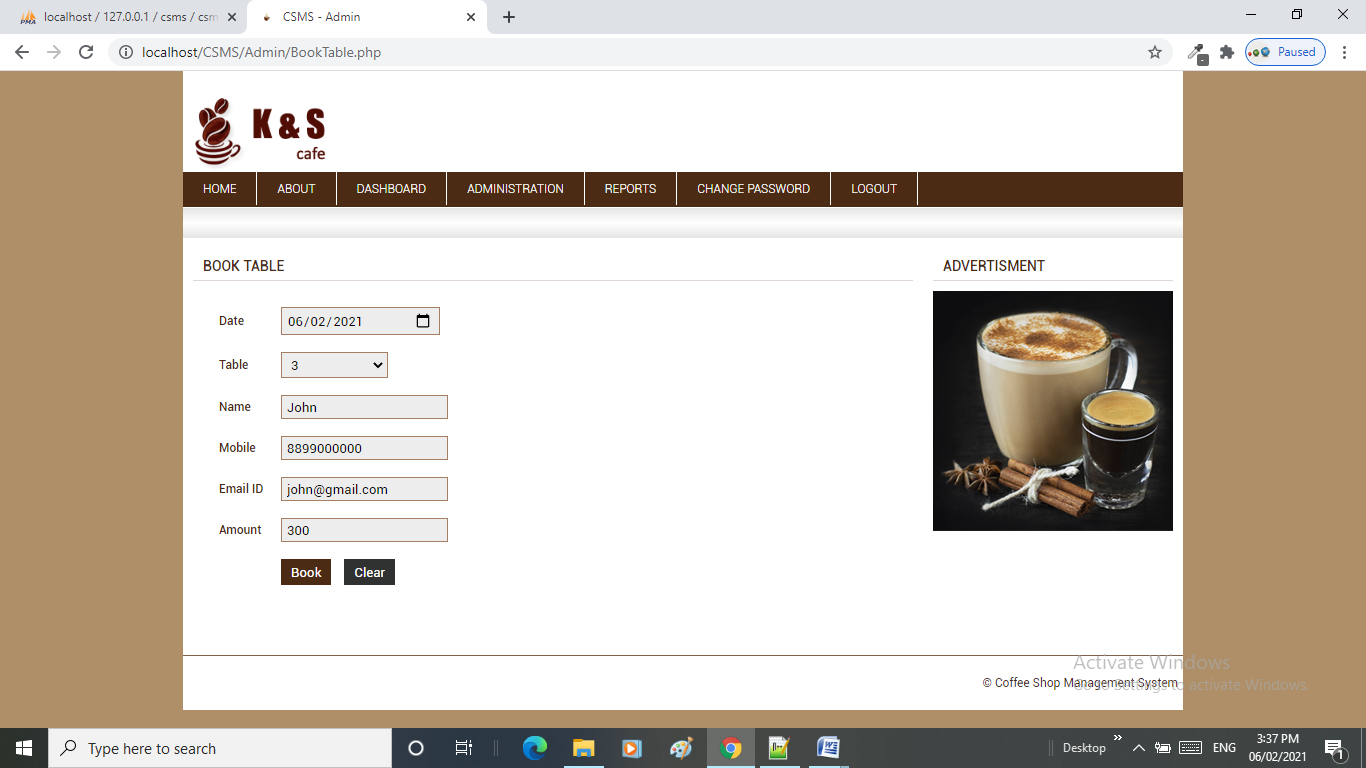
****

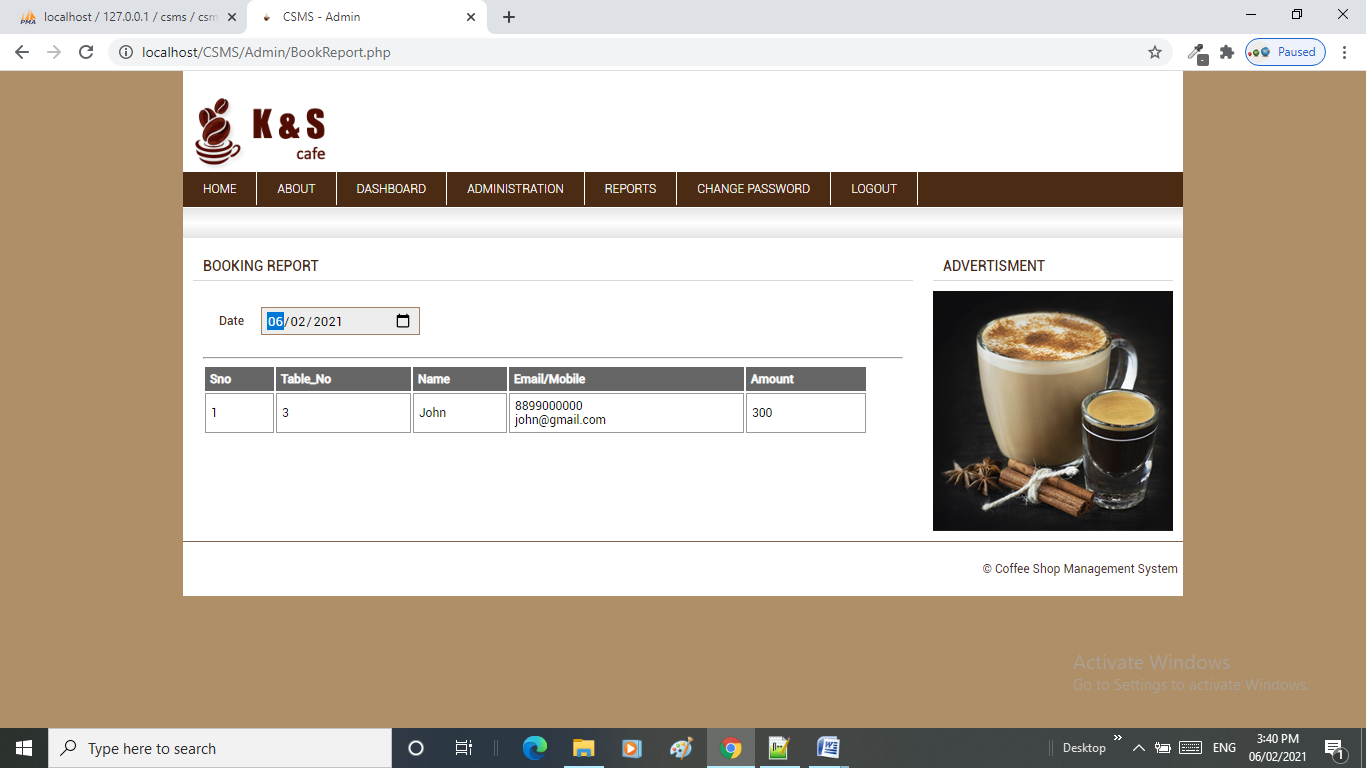
****

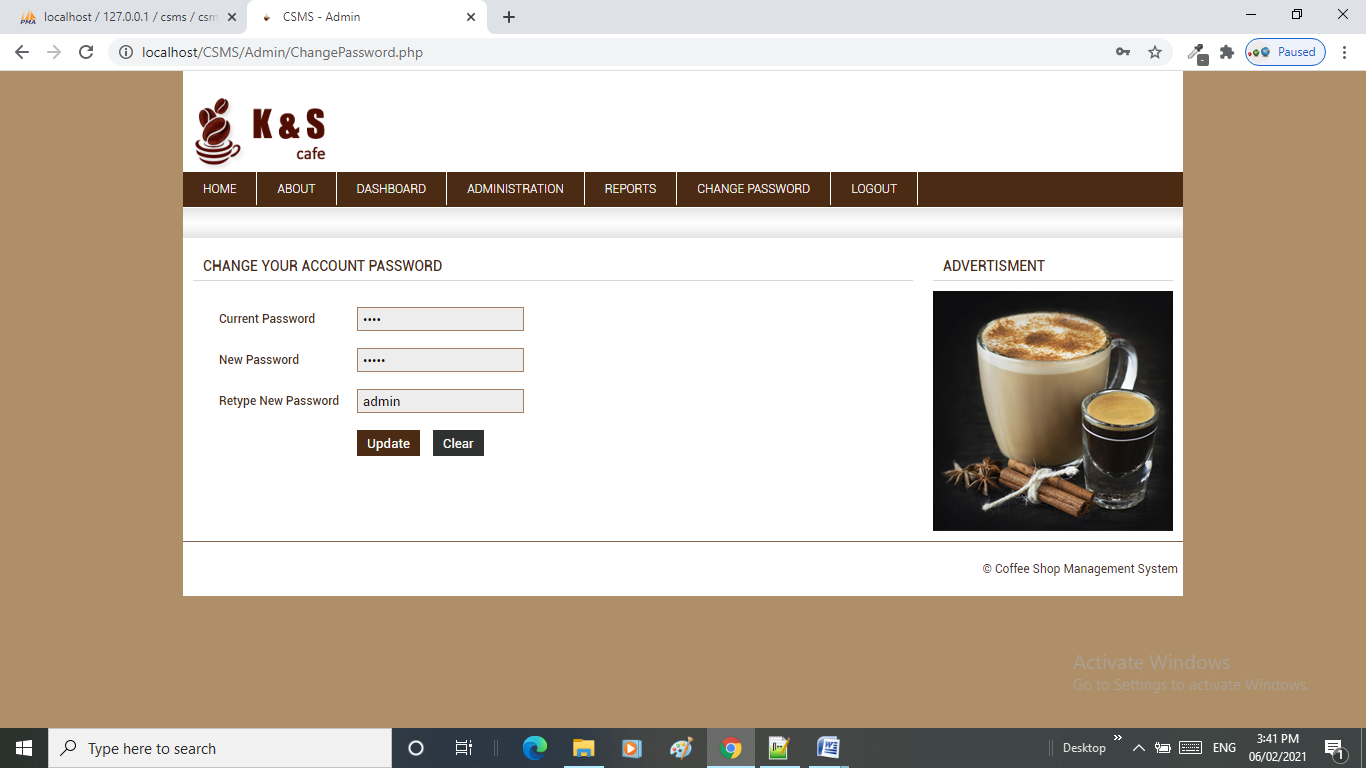
****

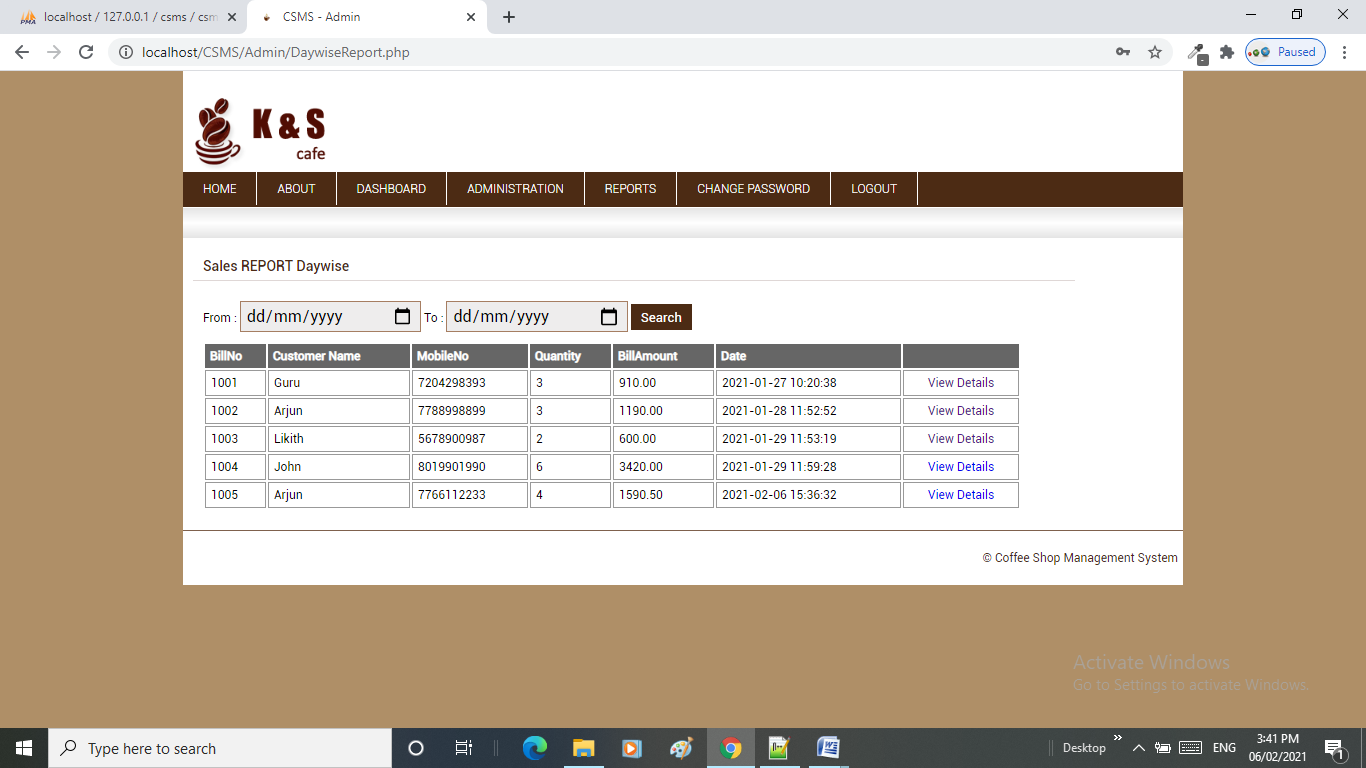
****

****

****

****

****

****

**CODINGS**

**config.php**

<?php

//session

session\_start();

if(isset($\_SESSION['adm\_sess']))

{

//header("location:Admin/");

}

//dbconnection

$con=mysqli\_connect("localhost","root","","csms") or die(mysqli\_error());

?>

**login.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS</title>

<link rel="shortcut icon" type="image/x-icon" href="Images/favicon.ico">

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

<script src="Script/new.js"></script>

</head>

<body>

<?php

include\_once("config.php");

if(isset($\_POST['sub']))

{

@extract($\_POST);

$sql=mysqli\_query($con,"select \* from csms\_admin where uname='$uname' and pwd=md5('$pwd')");

if(mysqli\_num\_rows($sql)==1)

{

$\_SESSION['adm\_sess']=$uname;

header("location:Admin/Dashboard.php");

}

else

{

$dbmsg="<div class='neg'>Login Info is not Correct</div>";

}

}

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left"></div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<div class="menu">

<div class="item"> <a href="index.php">HOME</a></div>

<div class="divider"> </div>

<div class="item"> <a href="About.php">ABOUT</a></div>

<div class="divider"> </div>

<div class="item"> <a href="Login.php">LOGIN</a></div>

<div class="divider"> </div>

<div class="item"> <a href="Contact.php">CONTACT US</a></div>

<div class="divider"> </div>

</div>

<div class="sep\_div1"></div>

<div id="banner">

<img src="Images/banner1.jpg"/>

</div>

<!--<div id="divider1">

</div>-->

<div id="main\_content">

<div id="cont\_left">

<div class="cont\_header txt1">LOGIN INTO YOUR ACCOUNT</div>

<div id="login\_form">

<form action="" method="post" name="admin\_login" id="admin\_login" autocomplete="off" onsubmit="return adm\_login()">

<table cellspacing="15px">

<tr>

<td class="label">Username</td>

<td><input type="text" name="uname" id="uname" class="inp1" placeholder="Type UserName" /></td>

</tr>

<tr>

<td class="label">Password</td>

<td><input type="password" name="pwd" id="pwd" class="inp1" placeholder="Type Password" /></td>

</tr>

<tr>

<td class="label"></td>

<td>

<input type="submit" name="sub" id="sub" class="sub" value="SignIn" />

<input type="reset" name="res" id="res" class="res" value="Clear" />

</td>

</tr>

<tr>

<td id="err" colspan="2"></td>

</tr>

<tr>

<td colspan="2"><?php if(isset($dbmsg)){echo $dbmsg;} ?></td>

</tr>

</table>

</form>

</div>

</div>

<div id="cont\_right">

<div class="cont\_header txt1">ADVERTISMENT</div>

<div class="space10"></div>

<div class="adv1"><img src="Images/coffee3.JPG"/></div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

</body>

</html>

**AddCoffee.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS - Admin</title>

<link rel="shortcut icon" type="image/x-icon" href="../Images/favicon.ico">

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

<script src="../Script/new.js"></script>

</head>

<body>

<?php

include\_once("config.php");

$msg="";

if(isset($\_POST['sub']))

{

@extract($\_POST);

$err=0;

$asql=mysqli\_query($con,"select bcode from csms\_coffee where bcode='$bcode'");

if(mysqli\_num\_rows($asql)!=0)

{

$err=1;

$msg.="Barcode Exist. ";

}

$bsql=mysqli\_query($con,"select cname from csms\_coffee where cname='$cname'");

if(mysqli\_num\_rows($bsql)!=0)

{

$err=1;

$msg.="Coffee Name Exist. ";

}

if($err==0)

{

$ins=mysqli\_query($con,"insert into csms\_coffee values('','$bcode','$cname','$ctype','$cprice','$cdesc',1)");

if(!$ins)

{

$dbmsg="<div class='pos'>Coffee Not Added.</div>";

}

else

{

$dbmsg="<div class='pos'>Coffee Added Successfully.</div>";

}

}

else

{

$dbmsg="<div class='neg'>$msg</div>";

}

}

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left"></div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<?php include\_once("menu.php"); ?>

<div class="sep\_div1"></div>

<div id="divider1">

</div>

<div id="main\_content">

<div id="cont\_left">

<div class="cont\_header txt1">ADD NEW COFFEE DETAILS</div>

<div id="admin\_main">

<form action="" method="post" name="add\_coffee" id="add\_coffee" autocomplete="off" onsubmit="return add\_cof()">

<table cellspacing="15px">

<tr>

<td class="label">BarCode</td>

<td><input type="text" name="bcode" id="bcode" class="inp1" placeholder="Type BarCode" /></td>

</tr>

<tr>

<td class="label">Coffee Name</td>

<td><input type="text" name="cname" id="cname" class="inp1" placeholder="Type CoffeeName" /></td>

</tr>

<tr>

<td class="label">Coffee Type</td>

<td>

<select name="ctype" id="ctype" class="inp1">

<option value="0">Select Coffee Type</option>

<option value="Light">Light</option>

<option value="Dark">Dark</option>

</select>

</td>

</tr>

<tr>

<td class="label">Price</td>

<td><input type="text" name="cprice" id="cprice" class="inp1" placeholder="Type Price" /></td>

</tr>

<tr>

<td class="label">Description</td>

<td><textarea name="cdesc" id="cdesc" class="inp1" placeholder="Type Description"></textarea></td>

</tr>

<tr>

<td class="label"></td>

<td>

<input type="submit" name="sub" id="sub" class="sub" value="Add" />

<input type="reset" name="res" id="res" class="res" value="Clear" />

</td>

</tr>

<tr>

<td id="err" colspan="2"></td>

</tr>

<tr>

<td colspan="2"><?php if(isset($dbmsg)){echo $dbmsg;} ?></td>

</tr>

</table>

</form>

</div>

</div>

<div id="cont\_right">

<div class="cont\_header txt1">ADVERTISMENT</div>

<div class="space10"></div>

<div class="adv1"><img src="../Images/coffee3.JPG"/></div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

</body>

</html>

**coffeeReport.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS - Admin</title>

<link rel="shortcut icon" type="image/x-icon" href="../Images/favicon.ico">

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

<script src="../Script/new.js"></script>

</head>

<body>

<?php

include\_once("config.php");

$msg="";

if(isset($\_POST['sub']))

{

}

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left"></div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<?php include\_once("menu.php"); ?>

<div class="sep\_div1"></div>

<div id="divider1">

</div>

<div id="main\_content">

<div id="cont\_left\_full">

<div class="cont\_header txt1">COFFEE REPORT</div>

<div id="admin\_main">

<form action="" method="post" name="searchcoffee" id="searchcoffee" style="margin:10px 0px;">

<input type="text" id="bcode" name="bcode" class="inp2" placeholder="BarCode" />

<input type="submit" name="sub" class="sub" value="Search" />

</form>

<table class="reporttab" align="center" width="95%" cellpadding="5px">

<tr bgcolor="#666666" style="color:white">

<th>BarCode</th>

<th>Coffe Name</th>

<th>Coffe Type</th>

<th>Coffe Price</th>

<th></th>

</tr>

<?php

if(isset($\_POST['sub']))

{

$bcode=$\_POST['bcode'];

$sql=mysqli\_query($con,"select \* from csms\_coffee where bcode='$bcode'");

}

else

{

$sql=mysqli\_query($con,"select \* from csms\_coffee");

}

if(mysqli\_num\_rows($sql)==0)

{

echo "<tr><td colspan='5'>Details Not Found.</td></tr>";

}

else

{

while($arr=mysqli\_fetch\_row($sql))

{

?>

<tr>

<td><?php echo $arr[1]; ?></td>

<td><?php echo $arr[2]; ?></td>

<td><?php echo $arr[3]; ?></td>

<td><?php echo $arr[4]; ?></td>

<td align="center"> <a href="EditCoffee.php?ce=<?php echo $arr[0]; ?>">Edit</a> | <a href="misc.php?cd=<?php echo $arr[0]; ?>" onclick="return cnfrm()">Delete</a> </td>

</tr>

<?php

}

}

?>

</table>

</div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

</body>

</html>

**AddSell.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS - Admin</title>

<link rel="shortcut icon" type="image/x-icon" href="../Images/favicon.ico">

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

<script src="../Script/new.js"></script>

<script src="Jquery.js"></script>

<style>

.hideit{display:none;}

.fbox{border:0px;outline:0px;}

#bill\_row\_wrapper input,.outline0 input{border:0px;outline:0px;}

#bill\_row\_wrapper tr td{padding:3px;border-bottom:1px solid #999999;}

#f\_amount{background:#fffc00;}

.width1{width:10%;}

.rmvbtn{background-image:url(../Images/remove.png);

height:20px;width:20px;background-size:100%;

background-color:rgba(0,0,0,0);}

</style>

</head>

<body>

<?php

include\_once("config.php");

$msg="";

if(isset($\_POST['sub']))

{

@extract($\_POST);

$ins=mysqli\_query($con,"insert into csms\_bill values($billno,now(),'$cus\_name','$cmobile',$f\_amount,$f\_quan)");

$insdetail="insert into csms\_sale values";

for($i=0;$i<sizeof($pid);$i++)

{

$bd\_pid=$pid[$i];

$bd\_bcode=$bcode[$i];

$bd\_pname=$pname[$i];

$bd\_ptype=$ptype[$i];

$bd\_rate=$rate[$i];

$bd\_quan=$quan[$i];

$bd\_amount=$amount[$i];

$insdetail.="('','$billno','$bd\_bcode','$bd\_pname','$bd\_ptype',$bd\_rate,$bd\_quan,$bd\_amount),";

}

$insdetail=substr($insdetail,0,strlen($insdetail)-1);

$sql\_a=mysqli\_query($con,$insdetail) or die(mysqli\_error($con));

if(!$sql\_a)

{

$dbmsg="<div class='pos'>Bill not generated.</div>";

}

else

{

$dbmsg="<div class='pos'>Bill Generated Successfully.</div>";

}

}

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left">Coffee Shop Management System</div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<?php include\_once("menu.php"); ?>

<div class="sep\_div1"></div>

<div id="divider1">

</div>

<div id="main\_content">

<div id="cont\_left\_full">

<div class="cont\_header txt1"></div>

<div id="admin\_main">

<form action="" method="post" name="add\_bill" id="add\_bill" autocomplete="off" onsubmit="return add\_bil()">

<table cellspacing="15px" width="100%">

<tr>

<td class="label">Bill No</td>

<td><input type="text" name="billno" id="billno" class="inp1" value="<?php echo getBillno(); ?>" readonly /></td>

<td width="100px"></td>

<td class="label">Bill Date</td>

<td><input type="text" name="bdate" id="bdate" class="inp1" value="<?php echo date("d-m-Y H:i:s"); ?>" readonly /></td>

</tr>

<tr>

<td class="label">Name</td>

<td><input type="text" name="cus\_name" id="cus\_name" class="inp1" placeholder="Customer Name" /></td>

<td width="100px"></td>

<td class="label">Mobile</td>

<td><input type="text" name="cmobile" id="cmobile" class="inp1" placeholder="Customer Mobile" /></td>

</tr>

<tr>

<td colspan="5"><div class="cont\_header txt1">BILLING ITEMS</div></td>

</tr>

<tr>

<td class="label">Select Coffee</td>

<td colspan="3">

<select name="bcode" id="bcode" class="inp1">

<option value="0">Select Coffee</option>

<?php

$csql=mysqli\_query($con,"select bcode,cname,ctype from csms\_coffee");

if(mysqli\_num\_rows($csql)==0)

{

echo "";

}

else

{

while($carr=mysqli\_fetch\_row($csql))

{

echo "<option value='$carr[0]'>$carr[1] $carr[2]</option>";

}

}

?>

</select>

<input type="text" name="cid" id="cid" class="hideit"/>

<input type="text" name="cbcode" id="cbcode" class="hideit" />

<input type="text" name="cname" id="cname" class="hideit"/>

<input type="text" name="ctype" id="ctype" class="hideit"/>

<input type="text" name="cprice" id="cprice" class="hideit"/>

</td>

</tr>

<tr>

<td colspan="5">

<table cellpadding="5px" width="100%">

<tr bgcolor="#999999">

<th>ItemCode</th>

<th>CoffeName</th>

<th>CoffeType</th>

<th>CoffePrice</th>

<th>Quantity</th>

<th>TotalAmount</th>

<th></th>

</tr>

<tbody id="bill\_row\_wrapper">

</tbody>

<tr class="outline0">

<td colspan='4'></td>

<td><input type='text' id='f\_quan' name='f\_quan' size="10" readonly/></td>

<td>Rs. <input type='text' id='f\_amount' name='f\_amount' size="7" readonly/></td>

<td></td>

</tr>

</table>

</td>

</tr>

<tr>

<td class="label" colspan="4"></td>

<td align="right">

<input type="submit" name="sub" id="sub" class="sub" value="Save & Print" />

</td>

</tr>

<tr>

<td colspan="5"><?php if(isset($dbmsg)){echo $dbmsg;} ?></td>

</tr>

<tr>

<td colspan="5" id="err"></td>

</tr>

</table>

</form>

</div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

<script>

var r=1;

$('#bcode').change(function(){

var bc=$("#bcode").val();

$.ajax({url:"Bill.php?bc="+bc,cache:false,success:function(result){

var pinfo=result;

if(pinfo=="\*\*\*\*\*")

{

$("#bcode").val("").focus();

return false;

}

var parr=pinfo.split("\*");

var pid=parr[0];var bcode=parr[1];var pname=parr[2];var ptype=parr[3];var pprice=parr[4];

$("#cid").val(pid);

$("#cbcode").val(bcode);

$("#cname").val(pname);

$("#ctype").val(ptype);

$("#cprice").val(pprice);

var rowid="row\_"+r;

var markup="<tr id='"+rowid+"'><td class='width1'><input readonly type='text' name='bcode[]' value='"+bcode+"' size='10'/></td><td class='width1'><input readonly type='text' name='pname[]' value='"+pname+"'/></td><td class='width1'><input readonly type='text' name='ptype[]' value='"+ptype+"' size='5'/></td><td class='width1'><input hidden type='text' class='pid' name='pid[]' value='"+pid+"' size='2'/><input readonly type='text' class='rate' name='rate[]' value='"+pprice+"' size='10'/></td><td class='width1'><input type='number' class='quan' name='quan[]' min='0' value='1' style='width:100px' /></td><td class='width1'><input readonly type='text' name='amount[]' class='amount' value='"+pprice+"' size='10'/></td><td class='width1' align='center'> <input type='button' class='rmvbtn' value='' onclick=\"rmv('"+rowid+"')\"/> </td></tr>";

$("#bill\_row\_wrapper").append(markup);

$("#bcode").val("").focus();

r++;

tot\_qua();

//tot\_gst();

//tot\_gst\_amnt(rowid);

//tot\_gst\_famnt();

calc\_famount();

}});

return false;

});

$(document).on("click", "input.rmvbtn" , function() {

$(this).closest('tr').remove();

var rowid=$(this).closest('tr').attr('id');

tot\_qua();

calc\_famount();

});

$(document).on("keyup", "input.quan" , function() {

var qu=$(this).val();

var qres=qu.replace(/\D+/,'');

$(this).val(qres);

});

//when we type quantity box - updation of amount + GSTamount

$(document).on("input", "input.quan" , function() {

add\_qua();

var cur\_row=$(this).closest('tr').attr('id');

var cur\_qua\_el="#"+cur\_row+" .quan";

var cur\_qua\_el\_val=$(cur\_qua\_el).val();

var cur\_rat\_el="#"+cur\_row+" .rate";

var cur\_rat\_el\_val=$(cur\_rat\_el).val();

var cur\_am\_el="#"+cur\_row+" .amount";

$(cur\_am\_el).val((cur\_qua\_el\_val\*cur\_rat\_el\_val).toFixed(2));

calc\_famount();

});

//When we type price

$(document).on("keyup", "input.rate" , function(){

var rowid=$(this).closest('tr').attr('id');

tot\_qua();

calc\_famount();

});

//updating final quantity

function tot\_qua()

{

var tq=0;

var q=document.getElementsByClassName('quan');

for(i=0;i<q.length;i++)

{

tq=tq+parseInt(q[i].value);

}

document.getElementById("f\_quan").value=tq;

}

//updating final quantity

function add\_qua()

{

var tq=0;

var q=document.getElementsByClassName('quan');

for(i=0;i<q.length;i++)

{

tq=tq+parseInt(q[i].value);

}

document.getElementById("f\_quan").value=tq;

}

function calc\_famount()

{

var famnt=0;

var fa=document.getElementsByClassName('amount');

for(i=0;i<fa.length;i++)

{

famnt=famnt+Number(fa[i].value);

}

var bam=0;

document.getElementById("f\_amount").value=famnt.toFixed(2);

}

</script>

</body>

</html>

**Changepassword.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS - Admin</title>

<link rel="shortcut icon" type="image/x-icon" href="../Images/favicon.ico">

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

<script src="../Script/new.js"></script>

</head>

<body>

<?php

include\_once("config.php");

if(isset($\_POST['sub']))

{

@extract($\_POST);

$chk=mysqli\_query($con,"select uname,pwd from csms\_admin where uname='$adm\_sess' and pwd=md5('$cpwd')");

if(mysqli\_num\_rows($chk)==1)

{

$upd=mysqli\_query($con,"update csms\_admin set pwd=md5('$npwd') where uname='$adm\_sess'");

if(!$upd)

{

$dbmsg="<div class='neg'>Password not updated</div>";

echo "<meta http-equiv='Refresh' content='2;url=Logout.php'>";

}

else

{

$dbmsg="<div class='pos'>Password updated successfully</div>";

echo "<meta http-equiv='Refresh' content='2;url=Logout.php'>";

}

}

else

{

$dbmsg="<div class='neg'>Current password is not correct</div>";

echo "<meta http-equiv='Refresh' content='2;url=Logout.php'>";

}

}

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left"></div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<?php include\_once("menu.php"); ?>

<div class="sep\_div1"></div>

<div id="divider1">

</div>

<div id="main\_content">

<div id="cont\_left">

<div class="cont\_header txt1">CHANGE YOUR ACCOUNT PASSWORD</div>

<div id="admin\_main">

<form action="" method="post" name="admin\_login" id="admin\_login" autocomplete="off" onsubmit="return chn\_pwd()">

<table cellspacing="15px">

<tr>

<td class="label">Current Password</td>

<td><input type="password" name="cpwd" id="cpwd" class="inp1" placeholder="Type Current Password" /></td>

</tr>

<tr>

<td class="label">New Password</td>

<td><input type="password" name="npwd" id="npwd" class="inp1" placeholder="Type New Password" /></td>

</tr>

<tr>

<td class="label">Retype New Password</td>

<td><input type="text" name="rpwd" id="rpwd" class="inp1" placeholder="Re-Type New Password" /></td>

</tr>

<tr>

<td class="label"></td>

<td>

<input type="submit" name="sub" id="sub" class="sub" value="Update" />

<input type="reset" name="res" id="res" class="res" value="Clear" />

</td>

</tr>

<tr>

<td id="err" colspan="2"></td>

</tr>

<tr>

<td colspan="2"><?php if(isset($dbmsg)){echo $dbmsg;} ?></td>

</tr>

</table>

</form>

</div>

</div>

<div id="cont\_right">

<div class="cont\_header txt1">ADVERTISMENT</div>

<div class="space10"></div>

<div class="adv1"><img src="../Images/coffee3.JPG"/></div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

</body>

</html>

**SellReport.php**

<!DOCTYPE html>

<html>

<head>

<title>CSMS - Admin</title>

<link rel="shortcut icon" type="image/x-icon" href="../Images/favicon.ico">

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

<script src="../Script/new.js"></script>

</head>

<body>

<?php

include\_once("config.php");

$msg="";

?>

<div id="main\_wrap">

<div id="header">

<div id="header\_left"></div>

<div id="header\_right">

info@csms.com <br/>

+91999 890 9999

</div>

</div>

<div class="sep\_div1"></div>

<?php include\_once("menu.php"); ?>

<div class="sep\_div1"></div>

<div id="divider1">

</div>

<div id="main\_content">

<div id="cont\_left\_full">

<div class="cont\_header txt1">Sales REPORT</div>

<div id="admin\_main">

<table class="reporttab" align="center" width="95%" cellpadding="5px">

<tr bgcolor="#666666" style="color:white">

<th>BillNo</th>

<th>Customer Name</th>

<th>MobileNo</th>

<th>Quantity</th>

<th>BillAmount</th>

<th>Date</th>

<th></th>

</tr>

<?php

$sql=mysqli\_query($con,"select \* from csms\_bill order by bno desc");

if(mysqli\_num\_rows($sql)==0)

{

echo "<tr><td colspan='7'>Details Not Found.</td></tr>";

}

else

{

while($arr=mysqli\_fetch\_row($sql))

{

?>

<tr>

<td><?php echo $arr[0]; ?></td>

<td><?php echo $arr[2]; ?></td>

<td><?php echo $arr[3]; ?></td>

<td><?php echo $arr[5]; ?></td>

<td><?php echo $arr[4]; ?></td>

<td><?php echo $arr[1]; ?></td>

<td align="center"> <a href="ViewBill.php?bno=<?php echo $arr[0]; ?>">View Details</a> </td>

</tr>

<?php

}

}

?>

</table>

</div>

</div>

</div>

<div class="sep\_div2"></div>

<!-- footer -->

<div id="footer"> &copy; Coffee Shop Management System</div>

</div>

</body>

</html>

**TESTING**

**SYSTEM TESTING**

**Project Testing:**

Testing is the most important phases in the software development activity. In software development life cycle (SDLC), the main aim of the testing process in the quality, the developed software is tested against attaining the required functionality and performance.

During the testing process the software is worked with some particular test case and the output of the test cases are analyzed whether the software is working according to the expectations or not.

The success of the testing process in determining the error is mostly depends upon the test case criteria, for testing any software we need to have a description of the expected behavior of the system and method of determining whether the observed behavior confirmed to the expected behavior.

**Level of testing:**

Since the error in the software can be injured at any stage. so, we have carry out the testing process at different levels during the development. The basic levels of testing are

* + - Unit Testing
    - Integration Testing
    - System Testing
    - Acceptance Testing.

**Unit Testing:**

Unit testing mainly focused first in the smallest and low level modules, proceeding one at a time. Bottom-up testing was performed on each module. As developing a driver program, the test modules by developed or used. But for the purpose of testing modules themselves were used as stubs, to print verification of the actions performed. After the lower level modules were tested, the modules that the next higher level those make use of the lower modules were tested

Each module was tested against required functionality and test cases were developed to test boundary values.

The unit testing has been tested with sample data and adequate corrections were made as per the user requirement, such that login module was tested with user id and code and appropriate error messages are provided for errors like data entry error, id error,etc.

**Integration Testing:**

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover error associated with interfacing. As the system consists of the number the edges of the two modules. The software tested under this incremental bottom-up approach.

Bottom-up approach integration strategy was implemented with the following steps.

* Low modules were combined into clusters that perform specific software sub function
* The clusters were tested

**Testing Process:**

A number of activities must be performed for testing software. Testing start with test plan. Test plan identifies all testing related activities that need to be performed along with the schedule and guide lines for testing. The plan also specified the levels of testing that need to be done, by identifying the different testing units. For each unit specified in the plan first the test cases and reports are produced. These report are analyzed.

**Test Plan:**

Test plan is a general document for entire project, which defines the scope, approach to be taken and the personal responsible for different activities of testing. The inputs for forming test plan are

* Project Plan
* Requirements Document System Design

There are two basic approaches for testing. They are

* + - Functional Testing
    - Structural Testing

**System testing:**

Series of different tests whose primary purpose is to fully exercise the computer-based system. It also tested to find discrepancies between the system and its original objective, current specifications.10011

**Execution Testing:**

This program was successfully loaded and executed. Due to programming there were no execution errors as fall as possible

**QUALITY ASSURANCE:**

Quality assurance defines the objectives of a project and reviews the overall activities so that error are corrected early in the development process.

Levels of Quality Assurance

Quality Assurance comes in three main levels namely

* + - * + Testing
        + Validation
        + Certification

**Testing:**

In system testing a common view is to eliminate program errors. This is extremely difficult and time consuming. Since designers cannot prove 100% accuracy. A successful test, then, is one that find errors.

Validation:

It checks the quality of the software in both simulated and live environments. In the Simulated approach the developers test the product on their workplace to make the products meet its requirements. In the Live Environment phase the product is given to the customer to evaluate the product’s functionality.

Validation refers to the different set of activities that ensure that software correctly implements a specific function and the software that been built is traceable to customer requirements. Verification and validation can be defined in a way like:

* Verification
* Validation

Software validation is achieved through a series of black-box test that demonstrate conformity with requirement. After each validation check a test has been conducted, one of the two possible condition exists

* The function or performance characteristics conform to specification and are expected
* A deviation from specification is uncovered and a deficiency list is created

**Alpha and Beta testing:**

The alpha testing is conducted at the developer’s site by the customer. the software is used in the natural setting with the developer”looking over the developer” and recording errors and usages problems. Alpha test is conducted in controlled environment

The beta testing is conducted at one or more customer site by the end user of the software. Unlike software testing the developer is generally not present. Therefore beta test is live application of the software in an environment that cannot be controlled by the developer

**Validation check applied in the project**

* The files entered in the project must only have the doc extension
* The data entered must have only one format
* One should not make a enter into without checking the password
* The qc must enter into the status column only accepted or not accepted
* File which are already be registered should not be registered once again or other time
* Qc can’t enter the file which are not entered by the dc
* Qc should enter the file into corresponding filenames registered by the dc
* Files which are registered and not accepted for the first time should not be registered again but the reentry of date and status should be done only

Certification:

Certification is to certify that the program or software package is correct and confirms to standards. With growing trend towards purchasing ready to use software, certification has become more important.

**SYSTEM SECURITY:**

Software integrity has become increasingly important in the age of hackers and firewalls. This attributes measures a system ability to withstand attacks (both accidental and intentional) to its security. Attacks can be made on all three components of software program, data, and documents

To measure integrity, two additional attributes must be defined

* + Thread
  + Security

**THREAD:**

Threat is the probability (which can be derived or estimated from empirical evidence) that an attack of specific type occur with in a specific time.

**Security:**

Security is the probability (which can be estimated or derived from empirical evidence) that attack on the specific type will be repelled.

**Security Testing:**

Any computer based system that manages sensitive information or causes action that can improperly harm(or benefit) individuals is the target for improper or illegal penetration. Penetration spans a board range of activities; hackers who penetrate system for sport; disgruntled employee who attempt to penetrate for revenge; dishonest individual who penetrate for illicit personnel gains.

Security testing to verify that protection mechanism built into a system will in fact provide proper protection form improper penetration. During system testing, the tester plays the role of the individual who desires to penetrate the system. Anything goes! The tester may attempt to acquire password through external clerical means; may attack the system with custom software designed to break down any defenses that have been constructed may overwhelm the system thereby denying the service to other; may purposely cause system errors.

**CONCLUSION**

**CONCLUSION**

**BIBLIOGRAPHY**

**1.Achyat. S.Godbole andAtulKahate,** “Web Technologies”, Tata McGrawHill, 2003.

**2. Dave Mercer,** “PHP5.0 with updation”, Tata McGraw-Hill Publications, 2001.

**3. Liberty** “PHP beginner” , Prentice Hall, 2001