**CHAPTER-1**

**INTRODUCTION**

Employee Attendance Management System is a software developed for daily employee attendance in collages and institutes**.** If facilitates to access the attendance information of a particular employee in a particular dept. The information **is** sorted by the operators, which will be provided by the hod for a particular dept.

This system will also help in evaluating attendance eligibility criteria of a employee.the employee attendance record in day by day they can manage to the employee attendance.by perfomed to the employee into the human by can manage to the employee attendance.

Every employee attendance & salary can manage to the user by satisfaction to the employee that are performing to the employee. New employee are registered that are performed to the data by which can handle to the user .this management system has performed by the user to that can manage by the salary to the employee.

The salary will assure delivery of the request from the citizen to the specified field employee of the concerned department and provide the acknowledgment of its receipt to the employee from the for the successful registered to the employee

**CHAPTER-2**

**SCOPE OF WORK**

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular employee. But later on the project can be modified to operate it employee.

The work of project that can satisfy to the employee by which can perform to manage the employee attendance. this project is useful to the attendance and salary can manage to the employee.

The system has working about to the user by which can satisfaction to the employee by handle the access of the database. By handle the operation to the salary can manage to the particular employee.

**CHAPTER- 3**

**EXISTING SYSTEM AND NEED FOR SYSTEM**

**EXISTING SYSTEM**

The existing system is a manual system.  Here the employees needs to save the information in the form of excel sheets or Disk Drives. There is no sharing is possible if the data is in the form of paper or Disk drives. The manual system gives us very less security for saving data; some data may be lost due to this management. It’s a limited system and fewer users friendly. Searching of particular information is very critical it takes lot of time. Gathering information of different sources is not an easy job, data will be mismanaged. Calculating different bills manually going to be as a process of mistake. There is no interface to provide various for providing centralized messages in the existing system.

The following drawbacks of the existing system emphasize the need for computerization of the existing System:

* Conventional System makes use of huge amounts of paper for recording transactions.
* Difficulty in tracking and retrieving data from the abundant papers is quite difficult.
* Existing system was not user friendly.
* System was not well organized and precise.
* It was time consuming.
* Information was redundant and inconsistent
* It didn’t integrated all the modules.
* Decision making was difficult.

**Department 4**

**Department 1**

**Department 6**

**Department 5**

**Department 3**

**Department 2**

**Front-end 2**

**Front-end 1**

**NEED FOR SYSTEM**

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

* Minimizing the manual records kept at different locations.
* There will be more data integrity.
* Facilitating desired information display, very quickly, by retrieving information from users.
* Facilitating various statistical information which helps in decision-making?
* To reduce manual efforts in activities that involved repetitive work.
* Updating and deletion of such a huge amount of data will become easier.

**CHAPTER-4**

**OPERATING ENVIRONMENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**HARDWARE REQUIRMENTS**

CPU type : PIV 2.5 GHz Processor and Above

Memory : 1 GB RAM

Hard disk : 80 GB HDD

Key board : 110 keys

**SOFTWARE REQUIREMENTS**

Operating System : Windows (xp) and Above

Web Technologies : HTML, PHP, VBScript, CSS,

Database (Back End) : My SQL

Language (Front End) : PHP, HTML, VBScript

Platform Environment : Xampp Server OR Apacthe Tomcat

### **CHAPTER- 5**

### **INTRODUCTION TO TOOLS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**5.1 Front End:**

* **HTML** –It is used to generate web page. HTML, an initialism of Hypertext Markup Language, is the predominant [markup language](http://en.wikipedia.org/wiki/Markup_language) for [web pages](http://en.wikipedia.org/wiki/Web_page). It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on.

* **VBSCRIPT** – It is used for checking User information before sending to JavaScript is a [scripting language](http://en.wikipedia.org/wiki/Scripting_language) most often used for [client-side](http://en.wikipedia.org/wiki/Client-side) web development. It is a [dynamic](http://en.wikipedia.org/wiki/Dynamic_language), [weakly typed](http://en.wikipedia.org/wiki/Weak_typing), [prototype-based](http://en.wikipedia.org/wiki/Prototype-based_programming) language with [first-class functions](http://en.wikipedia.org/wiki/First-class_function). Currently, "VBScript" is an implementation of the [ECMAScript](http://en.wikipedia.org/wiki/ECMAScript) standard.
* **PHP-** Php is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same.

**5.2 Backend:**

* **My SQL** is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) which has more than 6 million installations. MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases.
* **IIS SERVER** /WAMP- Apache is a [web container](http://en.wikipedia.org/wiki/Web_container), or [application server](http://en.wikipedia.org/wiki/Application_server) developed at the [Apache Software Foundation](http://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF).It adds tools for configuration and management but can also be configured by editing configuration files that are normally [XML](http://en.wikipedia.org/wiki/XML)-formatted. Apache includes its own internal [HTTP](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) server.

**Why PHP?**

**PHP** is a widely used, general-purpose [scripting language](http://en.wikipedia.org/wiki/Scripting_language) that was 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 can be embedded into [HTML](http://en.wikipedia.org/wiki/HTML) and generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server), which needs to be configured to process PHP code and create [web page](http://en.wikipedia.org/wiki/Web_page) content from it. It can be deployed on most web servers and on almost every [operating system](http://en.wikipedia.org/wiki/Operating_system) and [platform](http://en.wikipedia.org/wiki/Platform_%28computing%29) free of charge. PHP is installed on over 20 million websites and 1 million [web servers](http://en.wikipedia.org/wiki/Web_server).

PHP was originally created by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) in [1994](http://en.wikipedia.org/wiki/1995) and has been in continuous development ever since. The main implementation of PHP is now produced by **The PHP Group** and serves as the [*de facto* standard](http://en.wikipedia.org/wiki/De_facto_standard) for PHP as there is no [formal specification](http://en.wikipedia.org/wiki/Formal_specification). PHP is [free software](http://en.wikipedia.org/wiki/Free_software) released under the [PHP License](http://en.wikipedia.org/wiki/PHP_License), which is incompatible with the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL) because of restrictions on the use of the term *PHP*.

PHP has evolved to include a [command line interface](http://en.wikipedia.org/wiki/Command_line_interface) capability and can also be used in [standalone](http://en.wikipedia.org/wiki/Standalone_software) [graphical applications](http://en.wikipedia.org/wiki/Graphical_user_interface).

### **HISTORY**

*PHP* originally stood for Personal Home Page. It began in 1994 as a set of [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) [binaries](http://en.wikipedia.org/wiki/Binary_file) written in the [C programming language](http://en.wikipedia.org/wiki/C_programming_language) by the [Danish](http://en.wikipedia.org/wiki/Danish_people)/[Greenlandic](http://en.wikipedia.org/wiki/Greenland) programmer [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf). Lerdorf initially created these Personal Home Page Tools to replace a small set of [Perl](http://en.wikipedia.org/wiki/Perl) scripts he had been using to maintain his [personal homepage](http://en.wikipedia.org/wiki/Personal_homepage). The tools were used to perform tasks such as displaying his résumé and recording how much [traffic](http://en.wikipedia.org/wiki/Web_traffic) his page was receiving. He combined these binaries with his Form Interpreter to create PHP/FI, which had more functionality. PHP/FI included a larger implementation for the C programming language and could communicate with [databases](http://en.wikipedia.org/wiki/Database), enabling the building of simple, dynamic [web applications](http://en.wikipedia.org/wiki/Web_application). Lerdorf released PHP publicly on June 8, 1995 to accelerate [bug](http://en.wikipedia.org/wiki/Software_bug) location and improve the code. This release was named PHP version 2 and already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax was similar to Perl but was more limited, simpler, and less consistent.

[Zeev Suraski](http://en.wikipedia.org/wiki/Zeev_Suraski) and [Andi Gutmans](http://en.wikipedia.org/wiki/Andi_Gutmans), two developers at the [Technion IIT](http://en.wikipedia.org/wiki/Technion_IIT), rewrote the [parser](http://en.wikipedia.org/wiki/Parser) in 1997 and 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*. The development team officially released PHP/FI 2 in November 1997 after months of [beta](http://en.wikipedia.org/wiki/Development_stage) testing. Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new [rewrite](http://en.wikipedia.org/wiki/Rewrite_%28programming%29) of PHP's core, producing the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine) in 1999. They also founded [Zend Technologies](http://en.wikipedia.org/wiki/Zend_Technologies) in [Ramat Gan](http://en.wikipedia.org/wiki/Ramat_Gan), Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released. 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. On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), the PHP Data Objects extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. In 2008, PHP 5 became the only stable version under development. [Late static binding](http://en.wikipedia.org/wiki/Late_static_binding) has been missing from PHP and has been added in version 5.3. PHP 6 is under development alongside PHP 5. Major changes include the removal of register\_globals [magic quotes](http://en.wikipedia.org/wiki/Magic_quotes), and [safe mode](http://en.wikipedia.org/wiki/Safe_mode). The reason for the removals was that register\_globals had given way to security holes, and magic quotes had an unpredictable nature, and was best avoided. Instead, to escape characters, magic quotes may be substituted with the addslashes () function, or more appropriately an escape mechanism specific to the database vendor itself like mysql\_real\_escape\_string () for [MySQL](http://en.wikipedia.org/wiki/MySQL). Functions that will be removed in PHP 6 have been deprecated in PHP 5.3 and will produce a warning if used.

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

PHP currently does not have native support for [Unicode](http://en.wikipedia.org/wiki/Unicode) or multibyte strings; Unicode support will be included in PHP 6 and will allow strings as well as class, method and function names to contain non-[ASCII](http://en.wikipedia.org/wiki/ASCII) characters.

It runs in both [32-bit](http://en.wikipedia.org/wiki/32-bit) and [64-bit](http://en.wikipedia.org/wiki/64-bit) environments, but on Windows the only official distribution is 32-bit, requiring Windows 32-bit compatibility mode to be enabled while using [IIS](http://en.wikipedia.org/wiki/Internet_Information_Services) in a 64-bit Windows environment. As of PHP 5.3.0, experimental x64 bit versions are available.

**Usage**

PHP is a general-purpose scripting language that is especially suited for [web development](http://en.wikipedia.org/wiki/Web_development). PHP generally runs on a [web server](http://en.wikipedia.org/wiki/Web_server). Any PHP code in a requested file is [executed](http://en.wikipedia.org/wiki/Execution_%28computing%29) by the PHP runtime, usually to create [dynamic web page](http://en.wikipedia.org/wiki/Dynamic_web_page) content. It can also be used for [command-line](http://en.wikipedia.org/wiki/Command-line) scripting and [client-side](http://en.wikipedia.org/wiki/Client-side) [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) applications. PHP can be deployed on most [web servers](http://en.wikipedia.org/wiki/Web_server), many [operating systems](http://en.wikipedia.org/wiki/Operating_system) and [platforms](http://en.wikipedia.org/wiki/Platform_%28computing%29), and can be used with many [relational database management systems](http://en.wikipedia.org/wiki/Relational_database_management_system). 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](http://en.wikipedia.org/wiki/Filter_%28software%29), 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](http://en.wikipedia.org/wiki/Parser) [compiles](http://en.wikipedia.org/wiki/Compiler) input to produce [bytecode](http://en.wikipedia.org/wiki/Bytecode) for processing by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine), giving improved performance over its [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29) predecessor.Originally designed to create dynamic web pages, PHP now focuses mainly on [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting), and it is similar to other server-side scripting languages that provide dynamic content from a web server to a [client](http://en.wikipedia.org/wiki/Client_%28computing%29), such as [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s [Active Server Pages](http://en.wikipedia.org/wiki/Active_Server_Pages), [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems)' [JavaServer Pages](http://en.wikipedia.org/wiki/JavaServer_Pages), and [mod\_perl](http://en.wikipedia.org/wiki/Mod_perl). PHP has also attracted the development of many [frameworks](http://en.wikipedia.org/wiki/Software_framework) that provide building blocks and a design structure to promote [rapid application development](http://en.wikipedia.org/wiki/Rapid_application_development) (RAD). Some of these include [CakePHP](http://en.wikipedia.org/wiki/CakePHP), [Symphony](http://en.wikipedia.org/wiki/Symfony), [CodeIgniter](http://en.wikipedia.org/wiki/CodeIgniter), and [Zend Framework](http://en.wikipedia.org/wiki/Zend_Framework), offering features similar to other [web application frameworks](http://en.wikipedia.org/wiki/List_of_web_application_frameworks).

The [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) and [WAMP](http://en.wikipedia.org/wiki/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](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server) and [MySQL](http://en.wikipedia.org/wiki/MySQL), although the *P* may also refer to [Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29) or [Perl](http://en.wikipedia.org/wiki/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 [Facebook](http://en.wikipedia.org/wiki/Facebook), [Wikipedia](http://en.wikipedia.org/wiki/Wikipedia) (Media Wiki), [Yahoo!](http://en.wikipedia.org/wiki/Yahoo%21), [My Yearbook](http://en.wikipedia.org/wiki/MyYearbook), [Digg](http://en.wikipedia.org/wiki/Digg), [Joomla](http://en.wikipedia.org/wiki/Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress), [YouTube](http://en.wikipedia.org/wiki/YouTube), [Drupal](http://en.wikipedia.org/wiki/Drupal) and [Tagged](http://en.wikipedia.org/wiki/Tagged).

**Syntax**

[Syntax-highlighted](http://en.wikipedia.org/wiki/Syntax_highlighting) PHP code embedded within [HTML](http://en.wikipedia.org/wiki/HTML)

PHP only parses code within its [delimiters](http://en.wikipedia.org/wiki/Delimiter). Anything outside its delimiters is sent directly to the output and is not processed by PHP. 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 their purpose 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 well-formed XML.

Variables are prefixed with a [dollar symbol](http://en.wikipedia.org/wiki/Dollar_sign) and a [type](http://en.wikipedia.org/wiki/Primitive_type) does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted ("") and [heredoc](http://en.wikipedia.org/wiki/Heredoc) strings allow the ability to embed a variable's value into the string. PHP treats [newlines](http://en.wikipedia.org/wiki/Newline) as [whitespace](http://en.wikipedia.org/wiki/Whitespace_%28computer_science%29) in the manner of a [free-form language](http://en.wikipedia.org/wiki/Free-form_language) (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of [comment syntax](http://en.wikipedia.org/wiki/Comparison_of_programming_languages_%28syntax%29): /\* \*/ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text (e.g. to a web browser).

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.

**Data types**

PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit [signed](http://en.wikipedia.org/wiki/Signed_number_representations) [integers](http://en.wikipedia.org/wiki/Integer_%28computer_science%29). Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), [octal](http://en.wikipedia.org/wiki/Octal), and [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal) notations. [Floating point](http://en.wikipedia.org/wiki/Floating_point) numbers are also stored in a platform-specific range. They can be specified using [floating point](http://en.wikipedia.org/wiki/Floating_point) notation, or two forms of [scientific notation](http://en.wikipedia.org/wiki/Scientific_notation). PHP has a native [Boolean](http://en.wikipedia.org/wiki/Boolean_datatype) type that is similar to the native Boolean types in [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is *NULL*. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in [hashes](http://en.wikipedia.org/wiki/Hash_table) with both keys and values, and the two can be intermingled. PHP also supports [strings](http://en.wikipedia.org/wiki/String_%28computing%29), which can be used with single quotes, double quotes, or [heredoc syntax](http://en.wikipedia.org/wiki/Heredoc).

The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

**Example**

PHP gained support for [closures](http://en.wikipedia.org/wiki/Closure_%28computer_science%29). True [anonymous functions](http://en.wikipedia.org/wiki/Anonymous_function) are supported using the following syntax:

function getAdder($x)

{

return function ($y) use ($x) {

return $x + $y;

};

}

$adder = getAdder(8);

echo $adder(2); // prints "10"

Here, getAdder() function creates a closure using parameter $x (keyword "use" forces getting variable from context), which takes additional argument $y and returns it to the caller. Such a function can be stored, given as the parameter to other functions, etc. For more details see [Lambda functions and closures RFC](http://wiki.php.net/rfc/closures).

### **5.4 MYSQL**

**MySQL** is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) which has more than 6 million installations. MySQL stands for "My Structured Query Language". The program runs as a server providing multi-user access to a number of databases.

The project's [source code](http://en.wikipedia.org/wiki/Source_code) is available under terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL is owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB), now a [subsidiary](http://en.wikipedia.org/wiki/Subsidiary) of [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems), which holds the copyright to most of the codebase.

MySQL is commonly used by [free software](http://en.wikipedia.org/wiki/Free_software) projects which require a full-featured database management system, such as [WordPress](http://en.wikipedia.org/wiki/WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB) and other software built on the [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) software stack. It is also used in very high-scale [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web) products including [Google](http://en.wikipedia.org/wiki/Google) and [Facebook](http://en.wikipedia.org/wiki/Facebook).

**Uses** MySQL is used in [web applications](http://en.wikipedia.org/wiki/Web_application) and acts as the database component of the [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) software stack. Its popularity for use with web applications is closely tied to the popularity of [PHP](http://en.wikipedia.org/wiki/PHP), which is often combined with MySQL. Several high-traffic web sites (including [Flickr](http://en.wikipedia.org/wiki/Flickr), [Facebook](http://en.wikipedia.org/wiki/Facebook), [Wikipedia](http://en.wikipedia.org/wiki/Wikipedia), [Google](http://en.wikipedia.org/wiki/Google) (though not for searches), [Nokia](http://en.wikipedia.org/wiki/Nokia), Auctionmarts and [YouTube](http://en.wikipedia.org/wiki/YouTube)) use MySQL for data storage and logging of user data.

**Microsoft Front Page**

**Microsoft Front Page**, code-named *Quartz*, is a [WYSIWYG](http://en.wikipedia.org/wiki/WYSIWYG) [HTML editor](http://en.wikipedia.org/wiki/HTML_editor) and general [web design](http://en.wikipedia.org/wiki/Web_design) program by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). It is part of the [Expression Studio](http://en.wikipedia.org/wiki/Expression_Studio) suite. Expression Web edits web pages using [XML](http://en.wikipedia.org/wiki/XML), [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) 2.1, [ASP.NET](http://en.wikipedia.org/wiki/ASP.NET) 2.0, [XHTML](http://en.wikipedia.org/wiki/XHTML), [XSLT](http://en.wikipedia.org/wiki/XSLT) and [JavaScript](http://en.wikipedia.org/wiki/JavaScript). It requires the [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework) 2.0 to operate. [Microsoft SharePoint Designer](http://en.wikipedia.org/wiki/Microsoft_SharePoint_Designer) provides related Microsoft technology. It offers Visual Studio support for [ASP.NET](http://en.wikipedia.org/wiki/ASP.NET) applications. It has some legacy support for [classic ASP](http://en.wikipedia.org/wiki/Classic_ASP). Expression Web uses its own standards-based rendering engine which is different from Internet Explorer's [Trident](http://en.wikipedia.org/wiki/Trident_%28layout_engine%29) engine. Microsoft Expression Web provides the ability to install add-ins from third-party developers, extending its capabilities.

**CHAPTER- 6**

**PROPOSED SYSTEM**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PROPOSED SYSTEM**

This project covers the brief introduction of **EMPLOYEE ATTENDANCE & SALARY MANAGEMENT SYSTEM** and its aims to make all employee of accessible to the common man in his locality, through Internet Device OR System This project focuses on employee That how these services will be developed and implemented so that attendance are provided with employee where they can access the services under a single interface mechanism in the form of the his salary management. It includes the objective of the salary management of help of the any employee, This project also includes the enqury system of the any problem related to employee attendance & salary management.

**6.1 Objective to be fulfilled**

* Providing easy, anywhere and anytime access to candidate Services (both Information & Output).
* Publishing the static data and all information of the State departments in line with guidelines for necessary integration with GGU CSIT Employee Website.

**6.2 USER REQUIREMENT**

* Admin can manage user.
* Admin can be Modify , Upadate, Delete the Information of Any employee.
* Candidate itselt Online Exam Form fillup any time any places.
* Candidate first Manual guide then alfer filling the form carefully
* Candidate Must be remember Registation No. and Password.
* Candidate also download the Hall ticket of Semester Exam Time
* Each Candidate must be remember in our Registration No. And Password then Your Login the print Application.
* Must be required of the each field of Exam Form

**6.3 Requirements Determination Techniques and Systems Analysis Methods**

**DETERMINATION TECHNIQUES**

* **Fact Finding Analysis**

Fact finding is process of collection of data and information based on techniques which contain sampling of existing documents, research, observation, questionnaires, interviews, prototyping and joint requirements planning. System analyst uses suitable fact-finding techniques to develop and implement the current existing system. Collecting required facts are very important to apply tools in System Development Life Cycle because tools cannot be used efficiently and effectively without proper extracting from facts. Fact-finding techniques are used in the early stage of System Development Life Cycle including system analysis phase, design and post implementation review. Facts included in any information system can be tested based on three steps: data- facts used to create useful information, process- functions to perform the objectives and interface- designs to interact with users

A Fact-Finding Strategy

1. Learn from existing documents, forms, reports, and files.

2. If appropriate, observe the system in action.

3. Given all the facts that already collected, design and distribute questionnaires to clear up things that aren’t fully understood.

4. Conduct interviews (or group work sessions).

5. (Optional). Build discovery prototypes for any functional requirements that are not understood or for requirements that need to be validated.

6. Follow up to verify facts.

**Decision Analysis**

Decision analysis is a process that allows the decision maker to select at least and at most one option from a set of possible decision alternatives. There must be uncertainty regarding the future along with the objective of optimizing the resulting payoff (return) in terms of some numerical decision criterion.

The elements of decision analysis problems are as follow:

1. A sole individual is designated as the decision-maker. For example, the CEO of a company, who is accountable to the shareholders?
2. A finite number of possible (future) events called the 'States of Nature' (a set of possible scenarios).
3. A finite number of possible decision alternatives (i.e., actions) is available to the decision-maker. Only one action may be taken.

4. Payoff is the return of a decision.

**Data Flow Analysis**

Data-flow analysis is a technique for gathering information about the possible set of values calculated at various points in a computer program. A program's control flow graph (CFG) is used to determine those parts of a program to which a particular value assigned to a variable might propagate.

**SYSTEM ANALYSIS**

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or analyzer begins a study of the program using existing system.

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, ER-Diagram etc.

A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

* System analysis can be categorized into four parts.
* System planning
* Information Gathering
* Applying analysis tools for structured analysis
* Feasibility study
* Cost/ Benefit analysis

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility

**ECONOMIC FEASIBILITY**

A systems financial benefit must exceed the cost of developing that system. i.e. a new system being developed should be a good investment for the organization. Economic feasibility considers the following

i. The cost to conduct a full system investigation.

ii. The cost of hardware and software for the class of application.

iii. The benefits in the form of reduced cost or fewer costly errors.

iv. The cost if nothing changes (i.e. the proposed system is not developed).

The proposed system is economically feasible because

1. The system requires very less time factors.

ii. The system will provide fast and efficient automated environment instead of slow and error prone manual system, thus reducing both time and man power spent in running the system.

iii. The system will have GUI interface and very less user-training is required to learn it.

iv. The system will provide service to view various information for proper managerial decision making.

**TECHNICAL FEASIBILITY**

Technical feasibility centers around the existing computer system ((hardware and software) whether it can support the addition of proposed system, if not, to what extent it can support and the organization’s capacity to acquire additional components.

Our proposed system is technically feasible because –

* The hardware and software required are easy to install and handle.
* The necessary hardware configuration and software platform is already there

Technical feasibility centers around the existing computer system (hardware and software) whether it can support the addition of proposed system, if not to what extend it can support the organization’s capacity to acquire required additional components.

Proposed system is technically feasible because of the following reasons:-

1. It’s required less system resources.
2. Expandability will be maintained in the new system. New modules can be added later on the application, if required in the future.

The application will have User-friendly Forms and Screens, all validation checks. So the new system guarantees accuracy, reliability, ease of access and data security.

**ANALYSIS MODEL**

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

* Requirement Analysis
* Project Planning
* System design
* Detail design
* Coding
* Unit testing
* System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

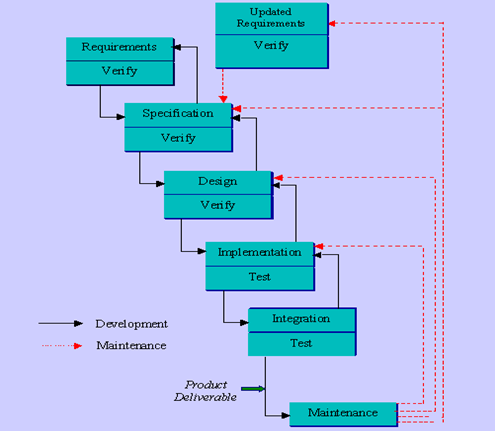
****

Figure 6.3.0 Waterfall Model

**Requirement:-**

In the requirement phase the need to create the application is specified. What is the need of the system is defined. What information to be feeder to create the application will come under the requirement phase?

**Design:**

After the requirement phase the next phase is the Design phase where the application is designed according to the forms and other modules created. This phase is much important phase because it will structure the layout of your application.

**Implementation:**

Implementation is the process of having a system personnel phase check out and put new equipment into use, train users, install new application and construct any file of data need to use it.

**Verification:**

After the whole application is being the developed the main phase is the verification phase where the whole application tested and verified to check the whole application.

**Maintenance:**

After the successful verification of the application the main phase is the maintenance phase where the application needs to be maintained for its successful operation in future.

**Disadvantages:**

* It is difficult for the customers to state the requirements clearly at the beginning. There is always certain degree of natural uncertainty at beginning of each project.
* Difficult and costlier to change when the changes occur at later stages.
* Customer can see the working version only at the end. Thus any changes suggested here are not only difficult to incorporate but also expensive. This may result in disaster if any undetected problems are precipitated to this stage

**6.4 SOFTWARE PROTOTYPING**

**Software prototyping** is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. It is an activity that can occur in software development and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing.

* Prototyping is the rapid development of a system
* In the past, the developed system was normally thought of as inferior in some way to the required system so further development was required
* Now, the boundary between prototyping and normal system development is blurred and many systems are developed using an evolutionary approach.
* Misunderstandings between software users and developers are exposed
* Missing services may be detected and confusing services may be identified
* A working system is available early in the process
* The prototype may serve as a basis for deriving a system specification
* The system can support user training and system testing.

**Prototyping Process:**

**Figure 6.4.1:- Prototyping Process**

**THE BENEFITS OF DEVELOPING A PROTOTYPE IN THE SOFTWARE PROCESS ARE:**

(1) Misunderstandings between software developers and users may be identified as the system functions are demonstrated.

(2) Missing user services may be detected.

(3) Difficult-to-use or confusing user services may be identified and refined.

(4) Software development staff may find incomplete and/or inconsistent requirements as the prototype is developed.

(5) A working, albeit limited, system is available quickly to demonstrate the feasibility and usefulness of the application to management.

(6) The prototype serves as a basis for writing the specification for a production quality system.

**SOFTWARE PROTOTYPES ALSO HAVE OTHER USES:**

(1) User training

(2) System testing

1. The final stage of the process is prototype evaluation.

2. There are non-technical managerial problems which may make it difficult to use prototyping in some organizations.

**PROTOTYPING IN THE SOFTWARE PROCESS**

1. One way of tackling this difficulty is to use an evolutionary approach to systems development.
2. Alternatively, a delivered decision might be made to build a 'throw-away' prototype to help requirements analysis and validation.
3. There is an important difference between the objectives of evolutionary and throw-away programming:

(1) The objective of evolutionary prototyping is to deliver a working system to end-users.

(2) The objective of throw-away prototyping is to validate or derive the system requirements.

**Approaches of Prototyping:**

****

**Figure 6.4.2:- Prototyping Approaches**

1. **Evolutionary Prototyping**

****

**Figure 6.4.3:- Evolutionary Prototyping**

1. **Incremental Process**

****

**Figure 6.4.4:- Incremental Prototyping**

1. **Throw-away Prototyping**

**8.5 Throw-away-prototyping.eps                                 00002F99Docs                           B1931E2B:**

**Figure 6.4.5:- Throw-away Prototyping**

**6.5 SYSTEM FEATURES**

**INPUT DESIGN**

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

* 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 as below:

* Data recording
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**INPUT TYPES:**

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

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

**INPUT MEDIA:**

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

* Type of input
* 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.

**OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization.
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely with in the computer department.
* Interface outputs, which involve the user in communicating directly with

**OUTPUT DEFINITION**

# The outputs 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
    - Volume of the output
    - Sequence of the output

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

For Example

* + - Will decimal points need to be inserted
    - Should leading zeros be suppressed.

**OUTPUT MEDIA:**

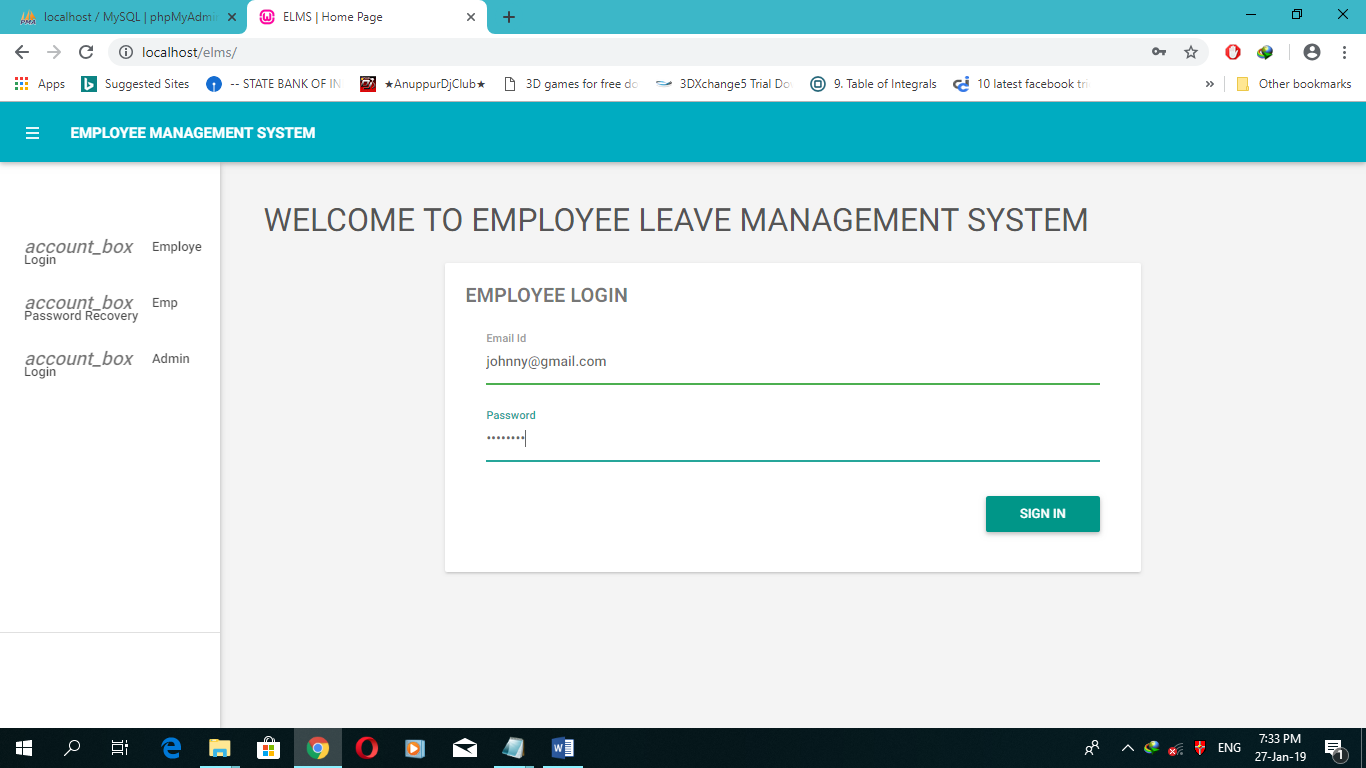
In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

* The suitability for the device to the particular application.
* The need for a hard copy.
* The response time required.
* The location of the users
* The software and hardware available.

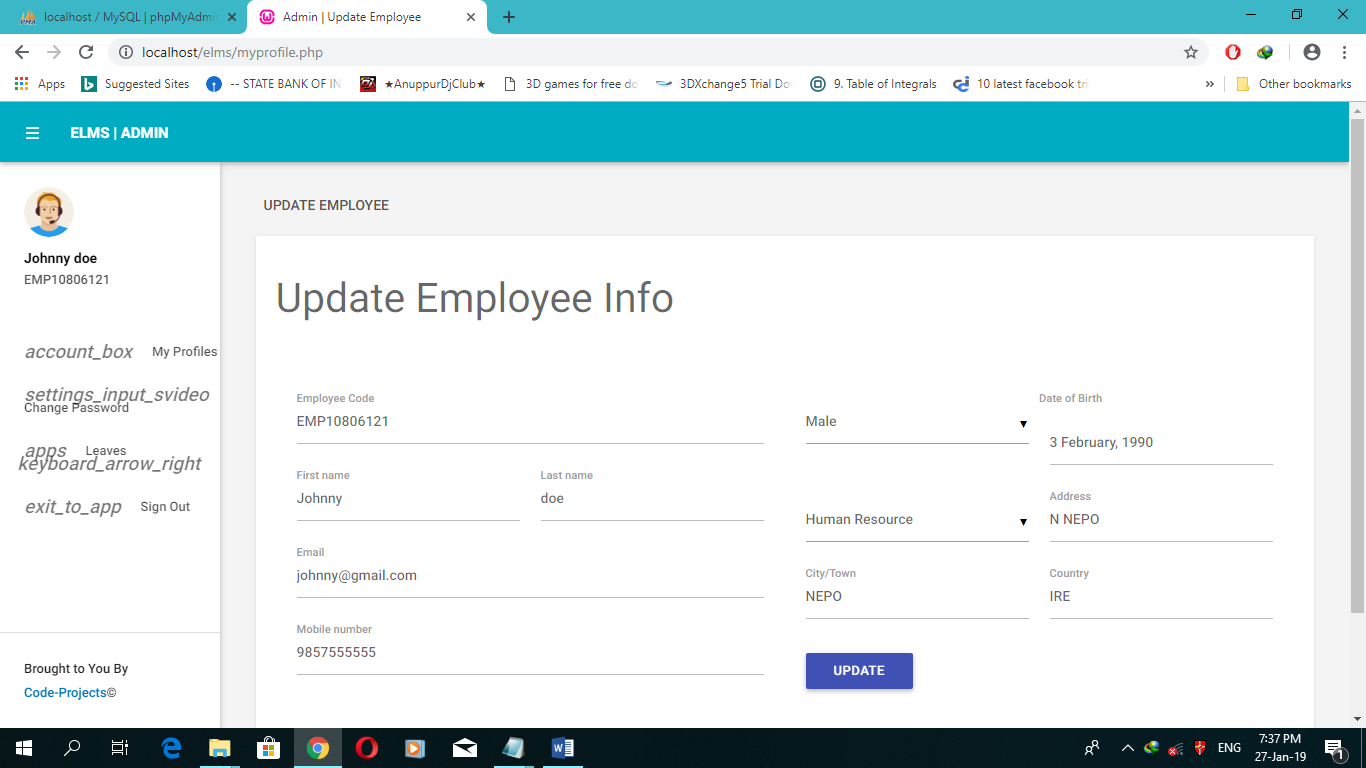
Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are: The outputs were needed to be generated as a hot copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

**OUTPUT SCREEN**

**ADMIN LOGIN PAGE**

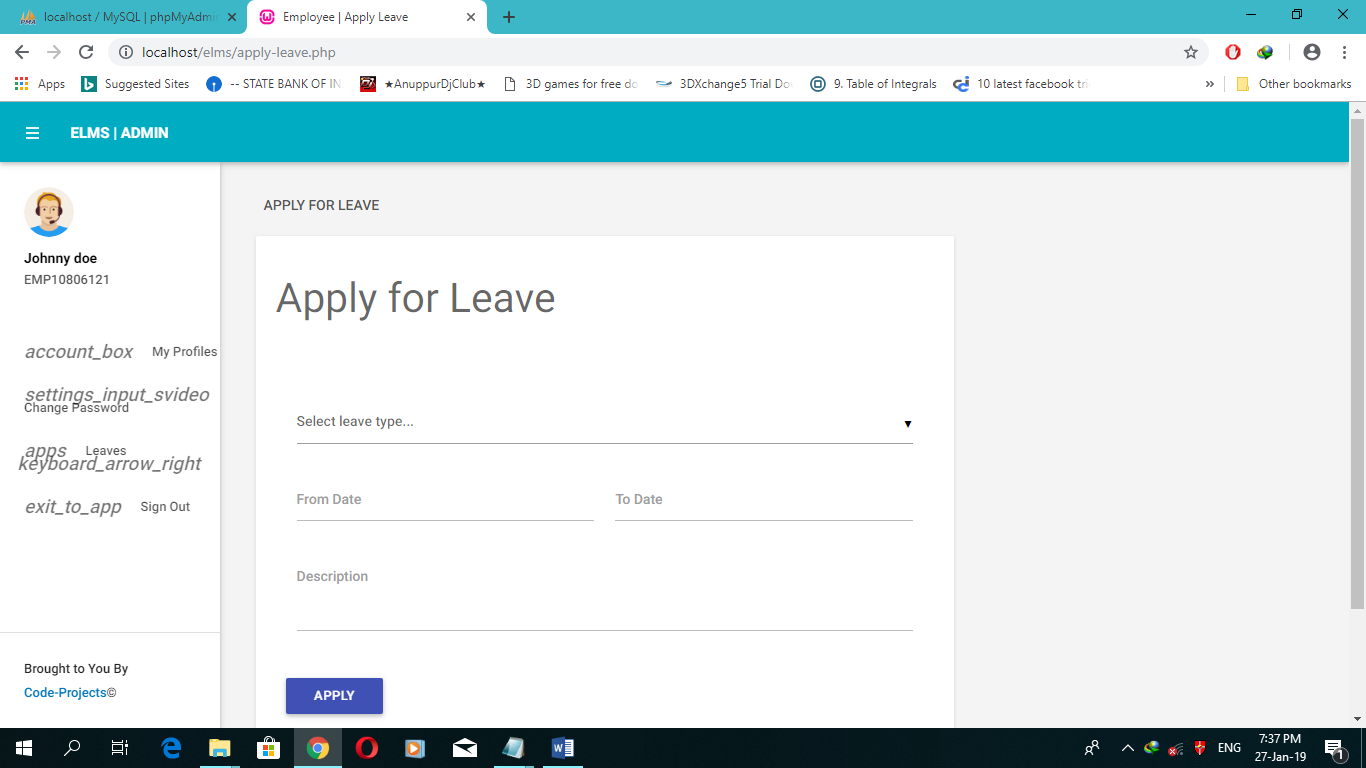


**Admin Homepage:**

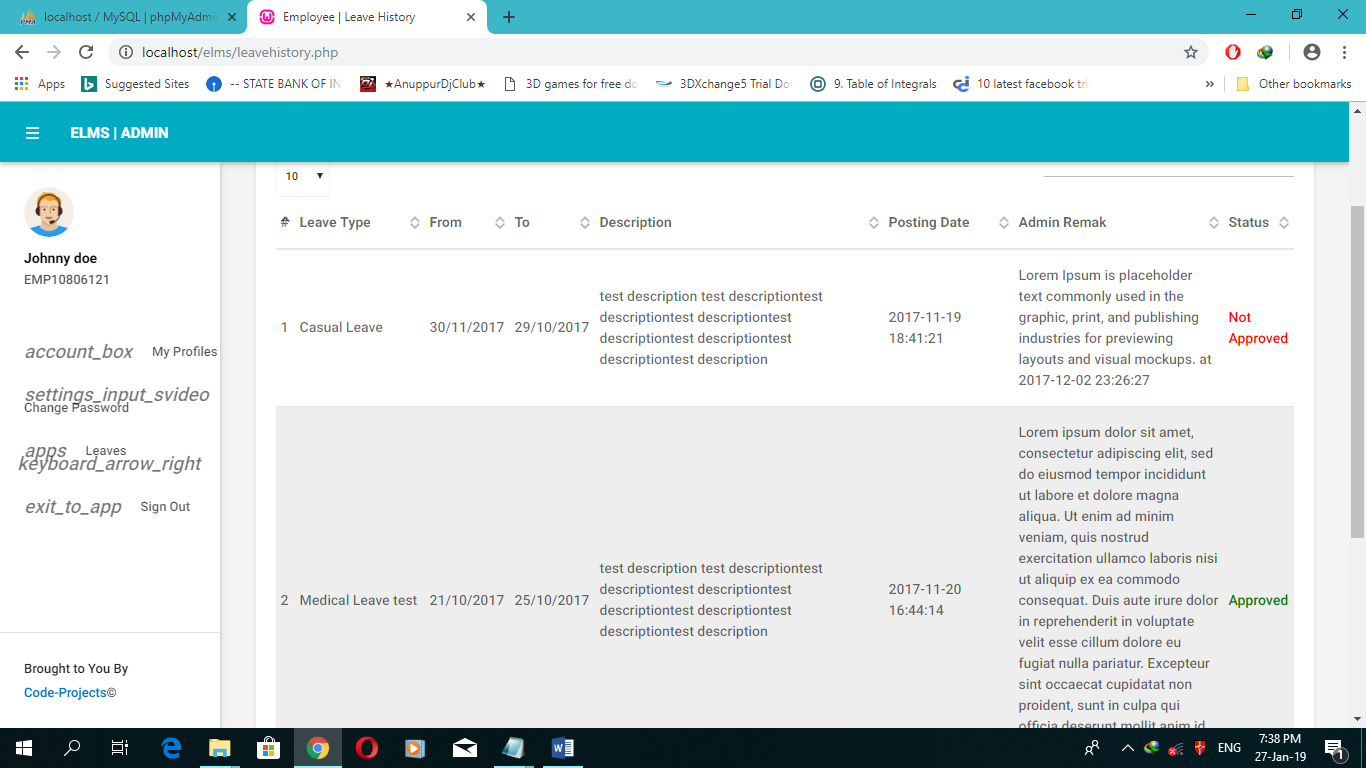


**Purpose** The purpose of this screen is to provide all the option to the admin.

**Administrator Manage**

**Purpose:** The purpose of this control of whole Database

**DISPLAY ATTENDANCE RECORD**



**MODULE SPECIFICATIONS**

Admin

1. Administrator

Department Employee

2) Employee

3) View Report

Department Employee Attendance

1. Latest News
2. Attendance
3. Application Record
4. Notice
5. Date by Date attendance can manage

Employee

1. New Registration
2. Show Employee Details

**E-R DIAGRAMS**

An entity-relationship model is a systematic way of describing and defining a business process. The process is modeled as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building*.* Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a [database](http://en.wikipedia.org/wiki/Database). In the case of a [relational database](http://en.wikipedia.org/wiki/Relational_database), which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationships.

**E-R MODEL**

M

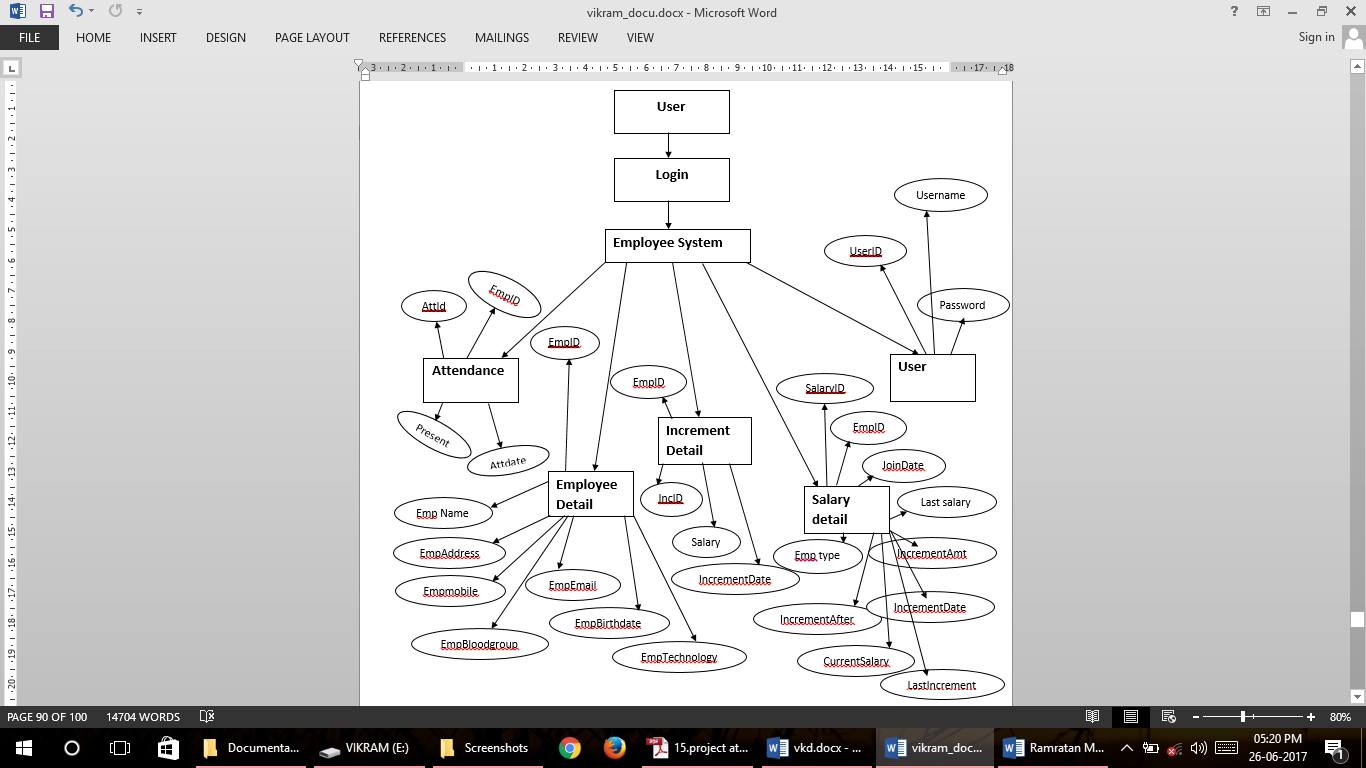
**The**[**three schema approach**](http://en.wikipedia.org/wiki/Three_schema_approach)**to**[**software engineering**](http://en.wikipedia.org/wiki/Software_engineering)**uses three levels of ER models that may be developed.**

[**Conceptual data model**](http://en.wikipedia.org/wiki/Conceptual_data_model)

This is the highest level ER model in that it contains the least granular detail but establishes the overall scope of what is to be included within the model set. The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing an enterprise-wide conceptual ER model is useful to support documenting the [data architecture](http://en.wikipedia.org/wiki/Data_architecture) for an organization.

A conceptual ER model may be used as the foundation for one or more *logical data models* (see below). The purpose of the conceptual ER model is then to establish structural [metadata](http://en.wikipedia.org/wiki/Metadata) commonality for the [master data](http://en.wikipedia.org/wiki/Master_data) entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration.

**ER DIAGRAM**



[**Logical data model**](http://en.wikipedia.org/wiki/Logical_data_model)

A logical ER model does not require a conceptual ER model, especially if the scope of the logical ER model includes only the development of a distinct information system. The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the entity relationships between these data entities are established. The logical ER model is however developed independent of technology into which it is implemented.

[**Physical data model**](http://en.wikipedia.org/wiki/Physical_data_model)

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each [database management system](http://en.wikipedia.org/wiki/Database_management_system) is somewhat different.

The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as [database tables](http://en.wikipedia.org/wiki/Database_table), [database indexes](http://en.wikipedia.org/wiki/Database_index) such as [unique key](http://en.wikipedia.org/wiki/Unique_key) indexes, and database constraints such as a [foreign key constraint](http://en.wikipedia.org/wiki/Foreign_key_constraint) or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database

**DFD (Data Flow Diagram)**

HOSTEL

STUDENT

VACATING

**0-level DFD:**

**Administrator**

**System**

**Output**

1-level DFD:

**Login**

**Process**

**Get**

**Result**

**Valid login**

**Page**

**New**

**Registration**

**Login**

**Process**

**Get**

**Info**

**Password**

**Page**

**User ID**

**1-level DFD:**

**2 -LEVEL DFD :-**

**2.1**

**\_\_\_\_\_Student**

**Login**

**After login**

**Database**

**Data Stored**

**Data Stored**

**After login**

**Data Entry**

**Data Entry**

**Administrator**

**Login**

**Process**

**Add record**

**Emp**

**Update**

**Delete**

**Execute**

**Administrator**

**2 -LEVEL DFD :-**

**2.2**

**Student**

**Login**

**After login**

**Database**

**Fetch data**

**Show Data**

**After login**

**Data Entry**

**Data Entry**

**Employee**

**Registraion**

**Process**

**Login Process**

**Entry**

**New Reg.**

**Administrator**

**DATA DICTIONARY**

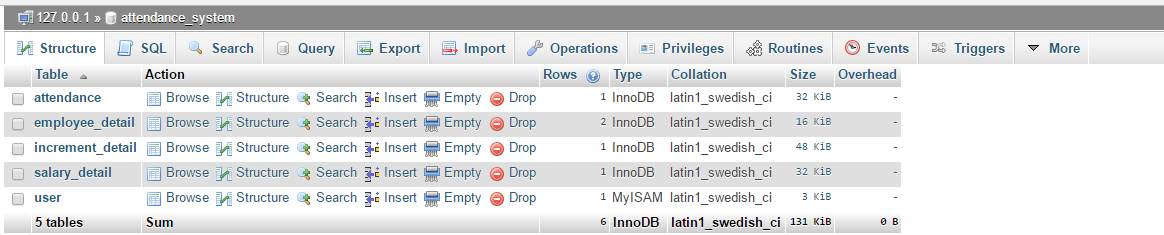
A data dictionary contains:

* The definitions of all schema objects in the database (tables, views, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, and so on)
* How much space has been allocated for, and is currently used by, the schema objects
* Default values for columns
* Integrity constraint information
* The names of Oracle users
* Privileges and roles each user has been granted
* Auditing information, such as who has accessed or updated various schema objects
* Other general database information

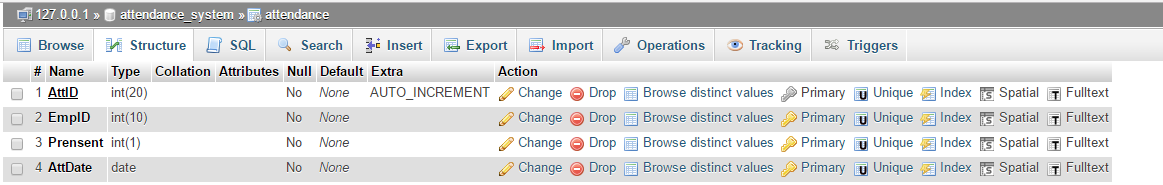
The data dictionary is structured in tables and views, just like other database data. All the data dictionary tables and views for a given database are stored in that database's SYSTEM table space.

**Tables in project :**

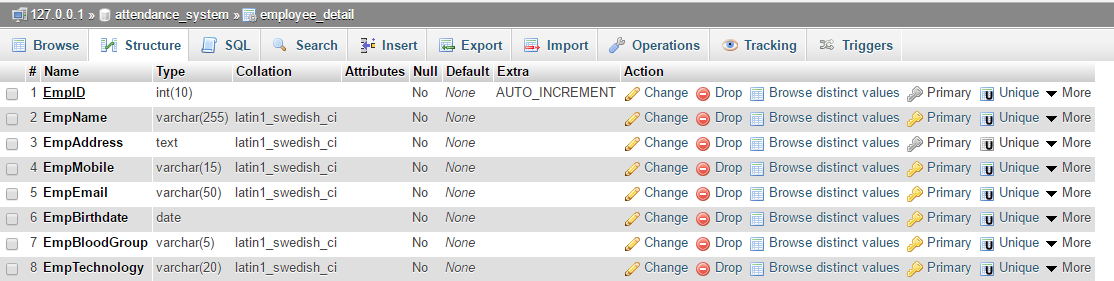
**Figure 6.7.1** TABLE ATTENDANCE



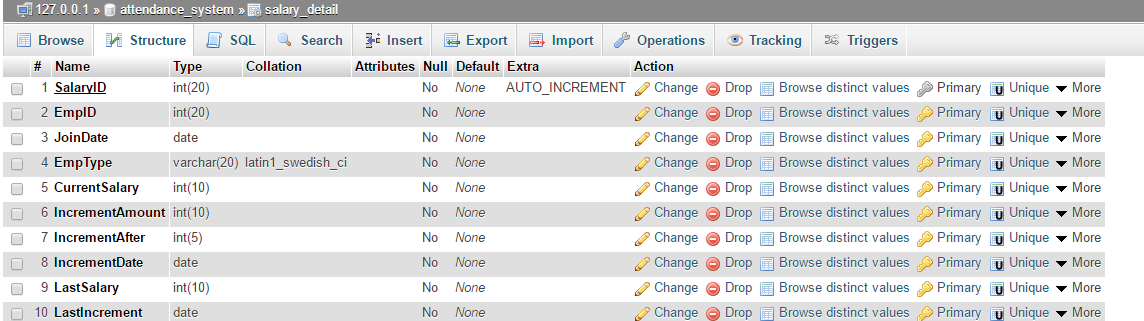
**Figure 6.7.2** TABLE ATTENDANCE DETAIL



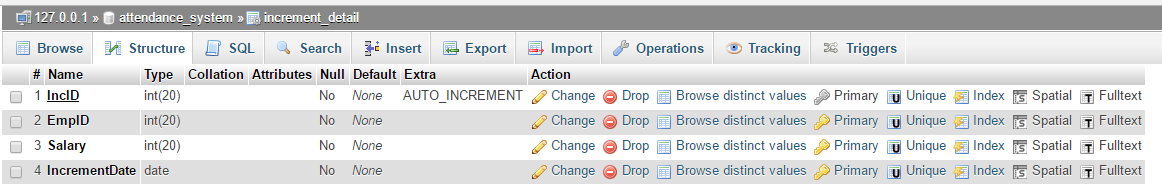
**Figure 6.7.3** TABLE EMPLOYEE DETAIL



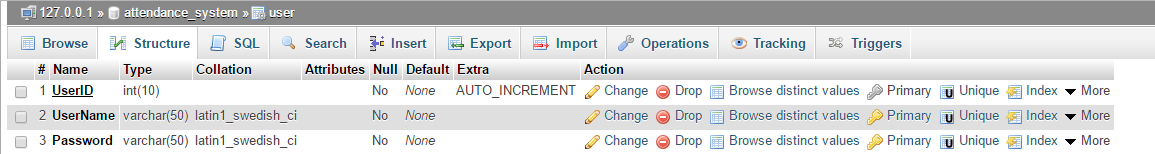
**Figure 6.7.4** TABLE SALARY DETAIL



**Figure 6.7.2** TABLE SALARY INCREMENT DETAIL

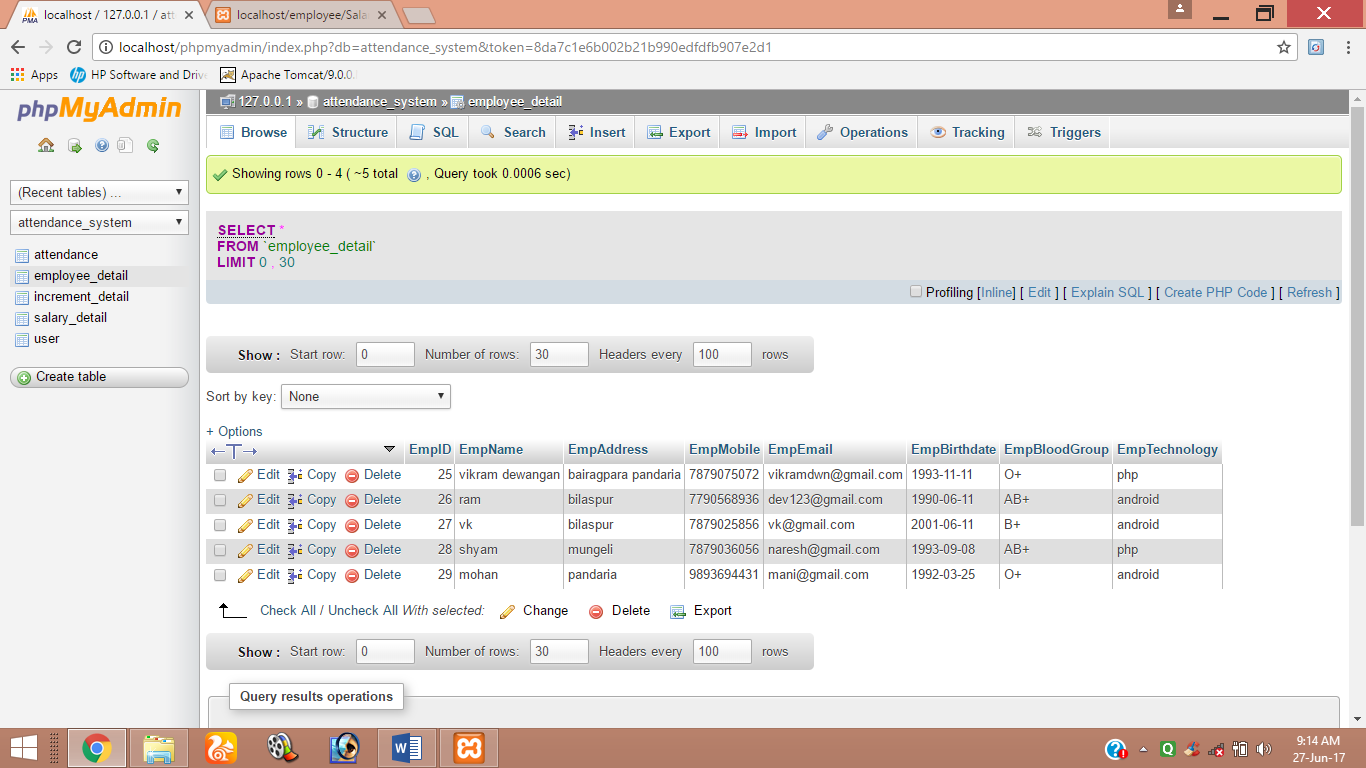


**Figure 6.7.2** TABLE USER DETAIL

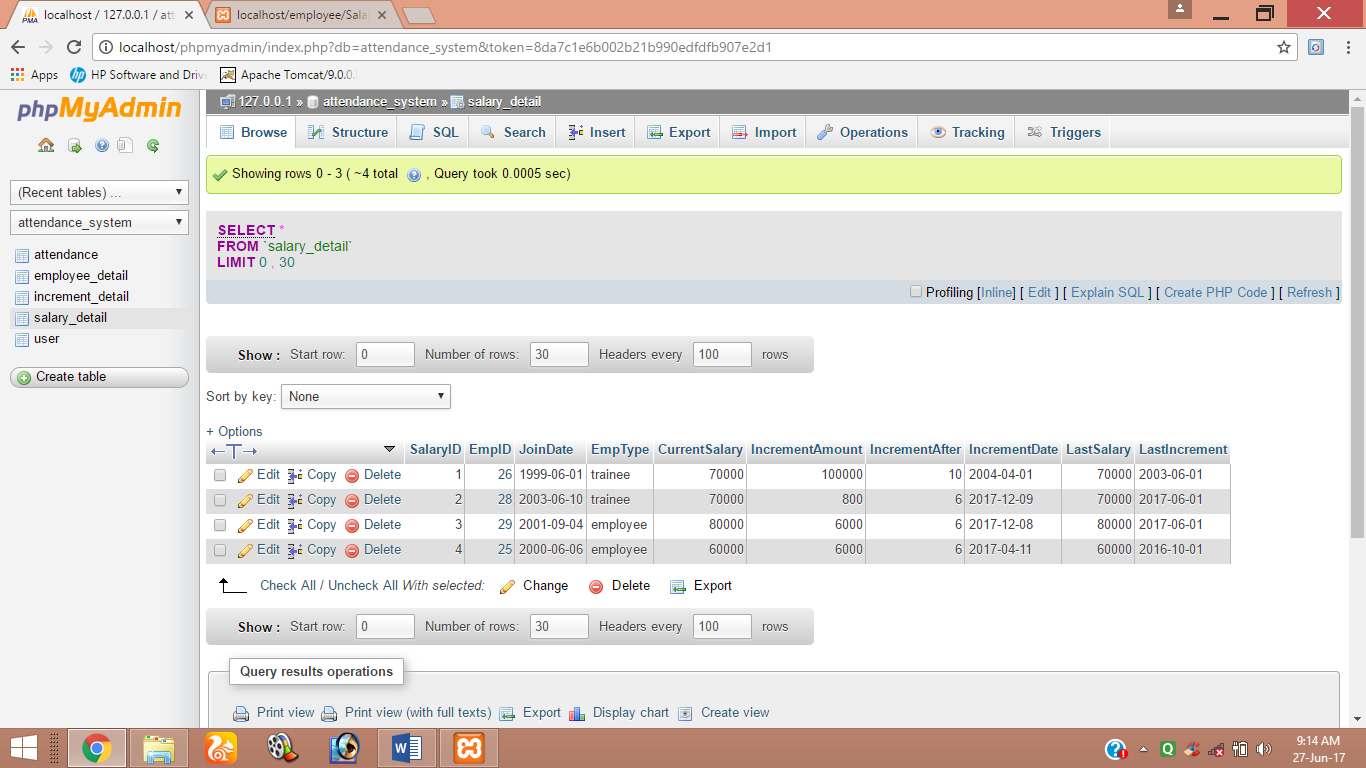
****

**DATABASE TABLE**

**EMPLOYEE RECORD**

****

**SALARY RECORD**

****

**STRUCTURE CHART**

A **Structure Chart** (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them. The lines represent the connection and or ownership between activities and sub-activities as they are used in organization charts.

In structured analysis structure charts, are used to specify the high-level design, or architecture, of a computer program. As a design tool, they aid the programmer in dividing and conquering a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain. The process is called top-down design, or functional decomposition. Programmers use a structure chart to build a program in a manner similar to how an architect uses a blueprint to build a house. In the design stage, the chart is drawn and used as a way for the client and the various software designers to communicate. During the actual building of the program (implementation), the chart is continually referred to as "the master-plan".

A structure chart depicts

* the size and complexity of the system, and
* number of readily identifiable functions and modules within each function and
* Whether each identifiable function is a manageable entity or should be broken down into smaller components.

Structure chart is a chart derived from Data Flow Diagram. It represents the system in more detail than DFD. It breaks down the entire system into lowest functional modules, describes functions and sub-functions of each module of the system to a greater detail than DFD.

Structure chart represents hierarchical structure of modules. At each layer a specific task is performed.

Here are the symbols used in construction of structure charts –

* **Module** - It represents process or subroutine or task. A control module branches to more than one sub-module. Library Modules are re-usable and invokable from any module.
* **Condition** - It is represented by small diamond at the base of module. It depicts that control module can select any of sub-routine based on some condition.
* **Jump** - An arrow is shown pointing inside the module to depict that the control will jump in the middle of the sub-module.
* **Loop** - A curved arrow represents loop in the module. All sub-modules covered by loop repeat execution of module.
* **Data flow** - A directed arrow with empty circle at the end represents data flow.
* **Control flow** - A directed arrow with filled circle at the end represents control flow.

**USER INTERFACE**

It is essential to consult the system users and discuss their needs while designing the user interface:

**USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:**

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
2. Computer initiated interfaces

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

**USER\_INITIATED INTERGFACES**

User initiated interfaces fall into two approximate classes:

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

**COMPUTER\_INITIATED INTERFACES**

The following computer\_ initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the 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 options. In this way every option leads the users to data entry form where the user can key in the data.

**DESIGN OF CONTROL PROCEDURES**

A new graphical representation, the Control Structure Diagram (CSD), has been created to visualize software at both the source code and program design language (PDL) level. The primary impetus for creation of the CSD was to improve the comprehension efficiency of software and, as a result, improve reliability and reduce costs. The CSD has the potential to replace traditional pretty printed source code.

As part of the GRASP (Graphical Representations of Algorithms, Structures, and Processes) research project at Auburn University, the GRASP software engineering tool has been successfully developed. GRASP automatically generates CSDs from source code written in Ada, C, C++, Java, and VHDL. The emphasis to this point has been on the automatic generation of the CSD to support development, maintenance, reverse engineering and reengineering through the use of GRASP. GRASP has been applied successfully to numerous programs ranging in size from several hundred to several thousand lines of source code and is efficient and sufficiently flexible for use in a production setting.

To demonstrate the potential benefits of the CSD and its automatic generation using GRASP, a series of empirical studies has been planned and initiated. First, as reported in this article, the perceived usefulness of the CSD was evaluated using a preference instrument based on eleven performance characteristics in which a comparison was made with other well-known graphical representations for algorithms. Statistical analysis indicated numerous significant differences with a clear preference for the CSD in seven of the eleven performance characteristics. Further empirical studies, currently being implemented, will examine the effect of the CSD and GRASP on objective measures such as comprehension efficiency and effectiveness.

**DESIGN OF EXCEPTION HANDLING**

Exception handling is the method of building a system to detect and recover from exceptional conditions. Exceptional conditions are any unexpected occurrences that are not accounted for in a system's normal operation. It is difficult to protect a system from the effects of exceptional conditions because, by nature, all unusual occurrences cannot be anticipated when the system is designed. Some examples of exceptional conditions are incorrect inputs from the user, bit level memory or data corruption, software design defects that cause a system to enter an undefined state, and environmental anomalies. If these exceptional conditions are not properly caught and handled, they can cause an error or failure in the system. Failures due to exceptions are estimated to account for two thirds of system crashes and fifty percent of system security vulnerabilities.

Exception handling is different from fault tolerance. Fault tolerance focuses on keeping known error states from causing system failures. Exception handling deals with the undefined and unanticipated conditions that, if left unchecked, can propagate through the system and cause a fault. Exception handling is more like fault avoidance or fault containment. I submit that exception handling is more difficult than fault tolerance because it must deal with all the unpredictability of the system.

When designing an embedded system, exception handling is usually focused on software. In fact, more than two thirds of code written for systems is devoted to properly detecting and handling exceptions. However, most software testing efforts focus on exercising the correct operation of code, and not determining how robust it is to exceptional conditions .Therefore, exception handling code is the least tested and most susceptible to bugs.

**Categories:**

Exception handling techniques can be separated into two broad categories: programmed exception handling and default exception handling.

In some cases programmed exception handling is capable of doing forward error recovery, but both programmed and default exception handling methods can perform backward error recovery. Forward error recovery can mask any exceptional occurrences and continue normal operation. Backward error recovery must halt normal system execution and attempt to return to a previous normal state to continue execution and retry the operation. Check pointing and recovery is a technique of backward error recovery for tolerating transient or intermittent conditions.

**Programmed Exception Handling**

Programmed exception handling modules are mechanisms built into software for specific exceptional cases that are known are likely to occur. Since these occurrences are relatively well understood, protection for them can be incorporated into the system. When a program is executing, if one of the exceptional conditions is detected, control is passed from the main process block to the special exception handling block. This code will deviate from normal execution to compensate for the exceptional condition and will attempt to mask it to prevent propagating an error condition to higher levels in the software hierarchy.

If the condition cannot be recovered, the exception handler may call check pointing recovery code to return the system to a known state before the exception occurrence and retry the operation.

**Default Exception Handling**

For all the exceptional conditions that are not anticipated by the system designers, default exception handlers must be built. The default handlers may be within the programming language or operating environment itself, transparent to the application developer. They must be a catch-all for any unexpected exceptions, and must also be responsible for containing exceptions due to design defects.

Exceptional conditions due to design defects are especially dangerous because they will always be present. If you knew about all design defects in a system a priori, they would have been eliminated before building the system. Since we have not yet learned how to design perfect systems, it is important that exception handlers can reduce the impact of design defects as much as possible.

**CHAPTER- 7.**

**TESTING PROCEDURE AND IMPLEMENTATION PHASES**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**7.1 INTRODUCTION**

**System Testing:**

Software testing is a critical element of software quality assurance and represents the review of specification, design and code generation. Once the source code has been generated the program should be executed before the customer gets it with the specific intent of finding and removing all errors, tests must be conducted systematically and tests must be signed using disciplined techniques.

Testing techniques provides systematic guidance for designing tests. To uncover the errors in program in behavior function and performance. The following steps have to be followed.

* Execute the internal logic of the software components.
* Execute the input and output domains of the program to uncover errors.

During testing the system is used experimentally to ensure that the software does not fail, i.e. it will run according to the speciation and in the way the users expects. Preparation of the data plays a vital role in the system testing. Different set of data are generated and the system under study is tested using data. While testing using data errors are again uncovered and corrected using different testing techniques.

**7.2 STRATEGIC APPROACH TO SOFTWARE TESTING**

The following are the different types of testing

* Unit Testing
* Integration Testing
* Validation Testing
* Output Testing
* User Acceptance Testing

1. **Unit Testing**

Unit testing focuses verification efforts on smallest unit of design and the module. This is known as module testing. Each module is tested separately. Different modules of the project such as remote desktop connecting, spy mode activity, client system controls are tested separately. Each module, like remote desktop connecting is to be tested as they were connected with their required clients. Similarly the spy mode activity module and client system controls modules should be tested. Each and every function of every module is tested and is found to be working satisfactorily as regard to the expected output from the module.

1. **Integration Testing**

Integration testing is for the design and construction of the software architecture. Data can be lost across the interfaces; one module may have an adverse effect on other. Thus integration testing is a systematic testing for constructing tests to uncover errors within the interface. Integration testing enables to decide whether the IP address has been displayed and whether the desktop is connected. Again it checks for the remote client for the proper continues signals. In this project all the modules are combined and the program is tested as a whole.

1. **Validation Testing**

Validation testing is where requirements established as a part of software requirements analysis is validated against the software that has been constructed. The validation that has been identified such as project team and bug administration has been tested. This test provides the final assurance that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are corrected during this phase.

1. **Output testing**

No system could be useful it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested along the users about the format required by them. Output testing does not result in any correction the system.

1. **User Acceptance Testing**

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

UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

Component Testing

Integration Testing

User Testing

**Figure** **7.1:- Testing Process**

**7.3 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**1. WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

**2. CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**3. DATA FLOW TESTING**

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

**4. LOOP TESTING**

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

* All the loops were tested at their limits, just above them and just below them.
* All the loops were skipped at least once.
* For nested loops test the inner most loop first and then work outwards.
* For concatenated loops the values of dependent loops were set with the help of connected loop.
* Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

**The Hardware and System Requirements**

To run the project, certain hardware and software has to be installed on the system. The system requirements include

* Microsoft Windows NT workstation.
* 80486 of higher microprocessor.
* A hard disk with a minimum of 50 megabytes available space for a full installation.
* SVGA or higher-resolution screen supported by Microsoft Windows.
* 32 MB RAM.
* 1.44’’ floppies drive.
* A mouse or other suitable pointing device.

**CHAPTER- 8**

**ACCEPTANCE PROCEDURE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

This section provides an overview of the acceptance test procedure that has been chosen.

# Testing Requirements

## Test Program

This section describes in detail the steps of the acceptance test procedure to be used.

# Code Path Test Procedure

# System Integration Test Procedure

This section lists the acceptance test procedure steps for verifying that the software meets the system integration requirements.

# Use-Case Requirements Test Procedure

This section lists the acceptance test procedure steps for verifying that the software meets the applicable functional requirements.

# Performance Requirements Test Procedure

This section lists the acceptance test procedure steps for verifying that the software meets the applicable performance requirements.

# Cross Reference Index

This section contains a list of any related documents. These can be documents, books, customer documents, newspaper cartoons, or whatever. Hyperlink pointers to online documents should be used.

**CHAPTER- 9**

**POST-IMPLEMENTATION REVIEW**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The Post-Implementation Review is used to evaluate the effectiveness of the system development after the system has been in production for a period of time (normally 6 months). The objectives are to determine if the system does what it is designed to do: Does it support the user as required in an effective and efficient manner? The review should assess how successful the system is in terms of functionality, performance, and cost versus benefits, as well as assess the effectiveness of the life-cycle development activities that produced the system. The review results can be used to strengthen the system as well as system development procedures.

The review is scheduled to follow the release of a system or system revision by an appropriate amount of time to allow determination of the effectiveness of the system. A representative from the functional development group or other member of the major user organization participates in the review. The System Proponent ensures that all documentation and all personnel needed to participate in the review are accessible.

The reviewer and an assigned team collect the information needed for the Post-Implementation Review by interviewing end users and their managers, system administrators, and computer operations personnel. The report is then prepared and provided to the user organization that requested it and the information systems organization, which may jointly use the findings to initiate other actions.

The Post-Implementation Review is a free-form report, and not all sections are relevant or necessary to the final product. A description of the Post-Implementation Review Report is attached.

# INTRODUCTION

## Project Identification

Provide the identifying information associated with the project, including the applicable project control code, system acronym, and system title.

## System Proponent

Provide the name of the System Proponent.

## History of the System

Briefly describe the system’s history and predecessor, if any. State the mission needs and information requirements, including how the system is expected to help users.

**Functional System Description and Data Usage**

Briefly describe what the system does functionally and how the data are used by the system.

## General Satisfaction with the System

Describe the users’ experience with the implemented system. Comments should address the following:

* The level of user satisfaction
* The strengths of the system, including specific areas of success
* Any problems
* Frequently used features
* Infrequently used features
* Features not used at all
* Suggested improvements

## Needed Changes or Enhancements

Gauge the magnitude of effort needed to change or improve the system. Describe the nature and priority of the suggested changes~ more detail will be provided in other sections. Comments should address the following:

* The suggested changes
* The scope of the changes
* The resource requirements to effect the changes

# ANALYSIS AND IMPLEMENTATION

The purpose of this section is to gauge the completeness of the functional requirements and implementation according to the study.

## Purpose and Objectives

Evaluate the adequacy of the original definition of purpose and objectives presented in the functional requirements document and if the objectives were achieved during implementation. Evaluate if any objectives have changed or should have changed. Comments should address the following:

* Extent to which goals were met
* The level of the objective definition
* Extent to which objectives were met
* Possible changes to the objectives

## Scope

Analyze if proper limits were established in the feasibility study and if they were

Maintained during implementation. Comments should address the following:

* Variations from the scope definition as agreed to in the concept development
* The extent to which the scope was followed
* Any possible future changes to the scope

## Benefits

Analyze if the benefits anticipated in the concept development and requirements definition analyses were realized. Detail all benefits, quantifiable or non-quantifiable, and any quantifiable resources associated with each. Comments should address the following:

* The adequacy of the benefit definition
* The level of benefits realized
* The anticipated benefits that can be realized
* The reason for the variance between planned and realized benefits

# OUTPUTS

The purpose of this section is to evaluate the adequacy and usefulness of the outputs from the system. Care must be taken to ensure that all reports are evaluated.

## Usefulness

Measure the extent to which the users need the output of the system. Comments should address identification of the level of need, such as the following:

* Utility
* Absolutely essential
* Important and highly desirable
* Interesting - proves what is already known
* Incomplete - does not provide all the necessary information
* Unnecessary
* Identification of information/reports needed but not currently generated by the system or unable to be obtained
* Demonstration of the ability to do without the reports
* Alternatives for obtaining the information where improvements can be achieved

## Timeliness

Determine if output production performance meets user needs. Comments should address the frequency with which output arrives on time, early, and late; and the amount of follow-up needed to obtain the output.

## Data Quality

Assess the need to provide for effective use of shared data to enhance performance and system interoperability. Comments should address data accuracy and data reliability.

# Security

The purpose of this section is to determine if the system provides adequate security of data and programs. In addition to access security, procedures for backup, recovery, and restart should be reviewed.

## Data Protection

Determine if the security, backup, recovery, and restart capabilities adequately safeguard data, including master, transaction and source. Online systems naturally require special techniques (such as, transaction logging). Comments should address the following:

* The adequacy of the security, backup, recovery, and restart procedures
* The suggested changes
* The effort required to make the changes

## Disaster Recovery

Determine if appropriate files, programs, and procedures are established to enable recovery from a disaster resulting in the loss of data. Comments should address the following:

* The adequacy and currency of off site storage procedures
* The extent that procedures cover the following:
* Master data
* Transaction data
* Source programs
* Object programs
* Documentation (such as, systems, operations, user manuals)
* The results of any adequacy-of-recovery test

## Controls

Evaluate the adequacy of the controls on the database, source documents, transactions, and outputs of the system. Review each area thoroughly for financial controls and file control counts. Comments should address the following:

* The level of controls present in the entire system and on each component (such as, transaction and batch, and file)
* The adequacy of the controls, including the strengths and possible areas for improvement
* The amount of resources required, if any, to obtain improvements

## Allowed Access

Evaluate the adherence to restriction of access to data. State desired privacy criteria for the system then evaluate how the criteria have been followed up to this point. Comments should address the following:

* Established privacy criteria
* Recommended privacy criteria
* Adherence to and violations of privacy
* The cost of providing this level of privacy
* The potential effect on individuals if the privacy criteria are not followed

# COMPUTER OPERATIONS

The purpose of this section is to ascertain the current level of operational activities. Although the user point of view is primary to the Post-Implementation Review Report, the computer operations view is also important to investigate.

## Scheduling

Determine the ability of computer operations to schedule according to user needs and to complete scheduled tasks. Comments should address the following:

* Any problems in accomplishing the work
* The frequency and extent of the problems
* Suggested changes
* The effort required to make changes

## User Interface

Analyze the usability of the system. The transaction throughput and error rate are included in this analysis. Comments should address the following:

* Number of errors made
* Frequency of problems with the interface
* Suggested changes
* Effort required to make the changes

## Computer Processing

Analyze computer processing issues and problems. Some areas to review are as follows:

* The correct or incorrect use of forms and off line files
* The adequacy of instructions (such as, forms lineup and proper responses on the console)
* The extent of reruns, if any

# MAINTENANCE ACTIVITIES

The purpose of this section is to evaluate maintenance activity involving the system.

## Maintenance Review

Review completed and pending changes to the system. Provide conclusions regarding the benefits to be achieved by completing recommended changes. Provide conclusions about the amount of maintenance required based on activity that has occurred to date.

## System Maintenance

Discuss the system maintenance based on the design, types of changes required, documentation, and knowledge about the system (both user and technical personnel).

**CHAPTER- 10**

**BENEFITS AND LIMITATIONS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**BENEFITS:**

* The main objective of the project is Any employee can manage to the employee attendance in particular way by thinking about a employee. This system has also perform on employee salary can manage to the employee.
* All the data are store on Server computer, But only allow the delete and modification by Administrator, and employee can be access around the globe.
* Employee record and Future enhancement download the and easy to learn. Can provide instruction/notification to in this system.

**LIMITATIONS:**

* **Not User Friendly:** The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.
* **Difficulty in report generating:** We require more calculations to generate the report so it is generated at the end of the session. And the student not get a single chance to improve their attendance
* **Manual control**: All calculations to generate report is done manually so there is greater chance of errors.
* **Lots of paperwork**: Existing system requires lot of paper work. Loss of even a single register/record led to difficult situation because all the papers are needed to generate the reports.
* **Time consuming**: Every work is done manually so we cannot generate report in the middle of the session or as per the requirement because it is very time consuming.

**CHAPTER- 11**

**CONCLUSION**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

The Attendance Management System is developed using PHP fully meets the objectives of the system which it has been developed. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the teachers and user associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification.

**CHAPTER- 12**

**FUTURE ENHANCEMENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* This System being web-based and an undertaking of Employee Attendance & Salary Management System , needs to be thoroughly tested to find out any security gaps.
* A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.
* Internet bandwidth can be upgraded from 60mbps to 100mbps.
* IP v4 to IP v6 conversion can be done for better performance.
* In future we can develop a mobile version of our system so that every person can access our site from anywhere while walking also.

* We also like to make our site more secure.

**CHAPTER- 13**

**REFERENCES & BIBLIOGRAPHY**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* Web Enabled Commercial Appplication Developmen -Ivan Bayross
* Head First Servlets and PHP

-Bryan Basham,Kathy Sierra and Bert Bates

* Wep search in [www.wikipedia.com](http://www.wikipedia.com) and PHP tutorial in help us.
* Website in help of [www.w3school.com](http://www.w3school.com) of the create a webpage
* **FOR PHP**
* [www.w3schools.com/php](http://www.w3schools.com/php)
* php.net/manual/en/intro-whatis.ph
* [www.tutorialspoint.com/php/](http://www.tutorialspoint.com/php/)
* [www.javatpoint.com/php-tutorial](http://www.javatpoint.com/php-tutorial)
* **WEB REFERENCES:**
  + - * [www.google.co.in](http://www.google.co.in)
      * [www.phptutorials.com](http://www.erenthub.com/)
      * [www.ggu.ac.in](http://www.ggu.ac.in)
      * [www.w3cschools.com](http://www.w3cschools.com/)
      * [www.phpbuddy.com](http://www.phpbuddy.com/)
      * [www.mysql.com](http://www.mysql.com/)
      * [www.phpeasystep.com](http://www.phpeasystep.com)
      * [www.w3schools.com/xml](http://www.w3schools.com/)