CHAPTER 1.0

INTRODUCTION

In the fast changing world, information technology and information management are going to play an important role. We are living in the computer age during past some year .The computer has gaining popularity. Computer revolution found its way into almost every aspect of human life and living. A computer is admirably suited to handle any information and hence is an information processor that is, it can receive data, perform some basic operations on that data and produces results according to a predetermined program.

The main objective of the proposed system is to decrease the paper work and easier record maintenance by having a particular website for leaves and notice maintenance. Initially leaves are taken by using the application form. Staff needs to submit their leaves manually to their respected higher authorities. This increases the paper work at the same time maintaining the records is more tedious. Maintaining the notices in the records also increases the paper work. But by using this system all the paper and manual work can be removed and things would become much simpler.

PROBLEM IDENTIFICATION & FEASIBILITY STUDY

PROBLEM IDENTIFICATION

We have got more than 250 Employee in our university; all employees’ leaves are managed manually by the Admin office. The Registration form verification to the different data processing are done manually. Thus there are a lot of repetitions which can be easily avoided. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a Leave and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

FEASIBILITY STUDY  
  
All projects are feasible given unlimited resources and infinite time! Unfortunately, the development of computer based system is more likely to be plagued by a scarcity of resources and difficult delivery dates. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Months or years of effort, Money loss and untold professional embarrassment can be averted I few better understand the project at its study time.

This type of study determines if an application can and should be developed. Once it has been determining that, application is feasible. After that analyst can go ahead and prepares the project specification, which finalizes project requirements. Feasibility studies are undertaken within tight time constraints.

1. **Technical Feasibility**
2. **Operational Feasibility**
3. **Economic Feasibility**
4. **Legal Feasibility**

**2.2 Technical Feasibility**

As we know the technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

* The facility to produce outputs of advertisements, shopping and mailing in a given time for ease of use.
* Response time under certain condition is minimal.
* Ability to process a certain volume of transaction at a particular speed.
* Facility to communicate data to distinct location.

In examining the technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system’s requirements- how many workstations are required, how these units are interconnected so that they could operate and communicate smoothly.

**2.3 Operational Feasibility**

Proposed projects are beneficial only if they can be turned into information system that will meet the financial management requirements of the business/organization. This test of feasibility asks if the system will work when it developed and installed. Are there major barriers to implementation?

Some of the important questions that are useful to test the operational feasibility of a project are given below:

* + Is there sufficient support for the project from the implementation? From user? If the present system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
  + Are current business methods acceptable to the user? If they are not, user may welcome a change that will bring about a more operational and useful system
  + Have the user been involved in the planning and development of the Project? If they are involved at the earliest stage of project development, the chances of resistance can be possibly reduced.
  + Will the proposed system cause harm? Will it produce poorer result in any case or area?
  + Will the performance of staff member fall down after implementation? Issue that
  + Appears to be quite minor at the early stage can grow into major problem after Implementation.

**Economical Feasibility**

Economic analysis is the most frequently used technique for evaluating the effectiveness of the proposed system. More commonly known as cost/benefits analysis, the procedure is to determine the benefits and savings that are expected from the purposed system and compared with costs.

If benefits outweigh cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative of the proposed system will have to be made if it has a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle. The analysis part also clears the doubt of economic problems which could be possible in developing the system. As already mentioned that the company has to just pay the developed software cost and not other investment is needed at the time of implementation of the new system as the preliminary requirements already exist in the company.

**Legal Feasibility**

In the legal feasibility is necessary to check that the software we are going to develop is legally correct which means that the ideas which we have taken for the proposed system will be legally implemented or not so , it is also an important step in feasibility study.

REQUIREMENT ANALYSIS

Requirements Analysis is the process of understanding the customer needs and expectations from a proposed system or application and is a well-defined stage in the Software Development Life Cycle model. Requirements are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of ‘what’ an application is expected to do. Given the multiple levels of interaction between users, business processes and devices in global corporations today, there are simultaneous and complex requirements from a single application, from various levels within an organization and outside it as well. The Software Requirements Analysis Process covers the complex task of eliciting and documenting the requirements of all these users, modeling and analyzing these requirements and documenting them as a basis for system design. Software Requirements Analysis and Documentation Processes are critical to software project success. Requirements engineering is an emerging field which deals with the systematic handling of requirements.

EXISTING SYSTEM

Analysis begins when a user or admin begins a study of the program using existing system. 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, interviews, etc. training, experience and common sense are required for collection of relevant information needed to develop the system. The Existing system comprises of a system in which details are to be manually handled this is not user friendly.

PROBLEM EVALUVATION AND SYNTHESIS

The project deals with the problems on managing employees’ leave applications and avoid the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly. We can improve the efficiency of the system, thus overcome the following drawbacks of the existing system.

REVIEW OF PREVIOUS WORK

PROPOSED WORK

Required software is for maintaining all the activities in managing a Leave Automation System. There are mainly Four modules in the proposed system:

MODULE DESCRIPTION

# Number of Modules

After careful analysis the system has been identified to have the following modules:

1. Admin Module
2. Employee Module

1. Admin Module  
 Admin will maintain all the employee details, employee leave record details, Hod leave record details in the database.

2. EmployeeModule  
  
In this module faculty will register their daily attendance information like incoming time and outgoing time and other information like leave approvalsand new notifications. Members should log in to their account and updateinformation. If there are any notifications related to previous day it is displayed in the form of pop up. Members should respond to that notification by sending replay to higher authorities.

**PROJECT PLANNING AND SCHEDULING:**

**PERT CHART:**

A PERT chart is a project management tools used to schedule, organize, and coordinate tasks within a project. PERT stands for Program Evaluation Review Technique. A PERT chart presents a graphic illustration of a project as network diagram consisting of numbered nodes(either circles or rectangles) representing events, or milestones in the project linked by labeled vectors (directional lines) representing tasks in the project.

The direction of the arrows on the lines indicates the sequence of tasks.

**Programming**

**Beta Testing**

**SRS And Design**

**Alpha Testing**

10

8

5

3

10 15 30 10 10

1

2

11

9

7

6

4

3 10 10 10 5

**Writing Manuals**

**Buy Hardware**

**User Requirement And Analysis**

**User Test**

**Training**

**Installation**

**GANTT CHART**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task No**. | **Task** | **Jul-Aug** | **Aug-Oct** | **Oct-Nov** | **Nov-Dec** | **Jan-Feb** | **Feb-Mar** | **Mar-April** |
| 1 | Requirement Analysis |  |  |  |  |  |  |  |
| 2 | Design |  |  |  |  |  |  |  |
| 3 | Coding |  |  |  |  |  |  |  |
| 4 | Testing |  |  |  |  |  |  |  |
| 5 | Documentation |  |  |  |  |  |  |  |
| 6 | Implementation |  |  |  |  |  |  |  |
| 7 | Final Review |  |  |  |  |  |  |  |

HARDWARE & SOFTWARE SPECIFICATION

**Hardware Resources :-**

The required hardware combination for this project will be as following :-

* Intel Pentium or Celeron class processor.
* Processor Speed **–** 1.7GHz or above.
* RAM **-** 256 MB or above.
* HDD - 40 GB or above.
* VGA Monitor.
* Scroll mouse, Multimedia Keyboard, CD-Drive.

**Software Resources :-**

**Front-end Tool:** - PHP, HTML, JAVASCRIPT.

* + - * + User friendly
        + Low Cost Solution
        + GUI feature
        + Better designing aspects.

**Back-end Tool:** - MySQL

**Features are**:

* + - The graphical management tools make it extremely easy to manage the server.
    - Security
    - Portability
    - Performance
    - Scalability
    - Reliability

**SOFTWARE ENGINEERING PARADIGM APPLIED**

This is the most common and classic of life cycle models, also referred to as a linear sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed in its entirety before the next phase can begin. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. Unlike what I mentioned in general model, phase does not overlap in a waterfall model.

**Waterfall Model Life cycle**

**Requirements**

**Design**

**Implementation and Unit Testing**

**Integration and System Testing**

**Operation**

Choice of Tools and Technology

**Introduction**

**PHP** as it's known today is actually the successor to a product named PHP/FI. Created in 1994 by Rasmus Lerdorf, the very first incarnation of PHP was a simple set of Common Gateway Interface (CGI) binaries written in the C programming language. Originally used for tracking visits to his online resume, he named the suite of scripts "Personal Home Page Tools," more frequently referenced as "PHP Tools." Over time, more functionality was desired, and Rasmus rewrote PHP Tools, producing a much larger and richer implementation. This new model was capable of database interaction and more, providing a framework upon which users could develop simple dynamic web applications such as guest books. In June of 1995, Rasmus [» released](http://groups.google.com/group/comp.infosystems.www.authoring.cgi/msg/cc7d43454d64d133) the source code for PHP Tools to the public, which allowed developers to use it as they saw fit. These also permitted - and encouraged - users to provide fixes for bugs in the code, and to generally improve upon it.

In September of that year, Rasmus expanded upon PHP and - for a short time - actually dropped the PHP name. Now referring to the tools as FI (short for "Forms Interpreter"), the new implementation included some of the basic functionality of PHP as we know it today. It had Perl-like variables, automatic interpretation of form variables, and HTML embedded syntax. The syntax itself was similar to that of Perl, albeit much more limited, simple, and somewhat inconsistent. In fact, to embed the code into an HTML file, developers had to use HTML comments. Though this method was not entirely well-received, FI continued to enjoy growth and acceptance as a CGI tool --- but still not quite as a language. However, this began to change the following month; in October, 1995, Rasmus released a complete rewrite of the code. Bringing back the PHP name, it was now (briefly) named "Personal Home Page Construction Kit," and was the first release to boast what was, at the time, considered an advanced scripting interface. The language was deliberately designed to resemble C in structure, making it an easy adoption for developers familiar with C, Perl, and similar languages. Having been thus far limited to UNIX and POSIX-compliant systems, the potential for a Windows NT implementation was being explored.

The code got another complete makeover, and in April of 1996, combining the names of past releases, Rasmus introduced PHP/FI. This second-generation implementation began to truly evolve PHP from a suite of tools into a programming language in its own right. It included built-in support for DBM, mSQL, and Postgres95 databases, cookies, user-defined function support, and much more. That June, PHP/FI was given a version 2.0 status. An interesting fact about this, however, is that there was only one single full version of PHP 2.0. When it finally graduated from beta status in November, 1997, the underlying parsing engine was already being entirely rewritten.

Though it lived a short development life, it continued to enjoy a growing popularity in still-young world of web development. In 1997 and 1998, PHP/FI had a cult of several thousand users around the world. A Net craft survey as of May, 1998, indicated that nearly 60,000 domains reported having headers containing "PHP", indicating that the host server did indeed have it installed. This number equated to approximately 1% of all domains on the Internet at the time. Despite these impressive figures, the maturation of PHP/FI was doomed to limitations; while there were several minor contributors, it was still primarily developed by an individual.

Example #1 Example PHP/FI Code

<!--include /text/header.html-->

<!--getenv HTTP\_USER\_AGENT-->

<!--ifsubstr $exec result Mozilla-->

Hey, you are using Netscape!<p>

<!--endif-->

<!--sql database select \* from table where user='$username'-->

<!--ifless $numentries 1-->

Sorry, that record does not exist<p>

<!--endif exit-->

Welcome <!--$user-->!<p>

You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->

### PHP 3

PHP 3.0 was the first version that closely resembles PHP as it exists today. Finding PHP/FI 2.0 still inefficient and lacking features they needed to power an eCommerce application they were developing for a university project, Andi Gutmans and Zeev Suraski of Tel Aviv, Israel, began yet another complete rewrite of the underlying parser in 1997. Approaching Rasmus online, they discussed various aspects of the current implementation and their redevelopment of PHP. In an effort to improve the engine and start building upon PHP/FI's existing user base, Andi, Rasmus, and Zeev decided to collaborate in the development of a new, independent programming language. This entirely new language was released under a new name, that removed the implication of limited personal use that the PHP/FI 2.0 name held. It was renamed simply 'PHP', with the meaning becoming a recursive acronym - PHP: Hypertext Preprocessor.

One of the biggest strengths of PHP 3.0 was its strong extensibility features. In addition to providing end users with a mature interface for multiple databases, protocols, and APIs, the ease of extending the language itself attracted dozens of developers who submitted a variety of modules. Arguably, this was the key to PHP 3.0's tremendous success. Other key features introduced in PHP 3.0 included object-oriented programming support and a far more powerful and consistent language syntax.

In June, 1998, with many new developers from around the world joining the effort, PHP 3.0 was announced by the new PHP Development Team as the official successor to PHP/FI 2.0. Active development of PHP/FI 2.0, which had all-but ceased as of November of the previous year, was now officially ended. After roughly nine months of open public testing, when the announcement of the official release of PHP 3.0 came, it was already installed on over 70,000 domains around the world, and was no longer limited to POSIX-compliant operating systems. A relatively small share of the domains reporting PHP as installed were hosted on servers running Windows 95, 98, and NT, and Macintosh. At its peak, PHP 3.0 was installed on approximately 10% of the web servers on the Internet.

### PHP 4

By the winter of 1998, shortly after PHP 3.0 was officially released, Andi Gutmans and Zeev Suraski had begun working on a rewrite of PHP's core. The design goals were to improve performance of complex applications, and improve the modularity of PHP's code base. Such applications were made possible by PHP 3.0's new features and support for a wide variety of third party databases and APIs, but PHP 3.0 was not designed to handle such complex applications efficiently.

The new engine, dubbed 'Zend Engine' (comprised of their first names, Zeev and Andi), met these design goals successfully, and was first introduced in mid 1999. PHP 4.0, based on this engine, and coupled with a wide range of additional new features, was officially released in May 2000, almost two years after its predecessor. In addition to the highly improved performance of this version, PHP 4.0 included other key features such as support for many more web servers, HTTP sessions, output buffering, more secure ways of handling user input and several new language constructs.

### PHP 5

PHP 5 was released in July 2004 after long development and several pre-releases. It is mainly driven by its core, the *Zend Engine 2.0* with a new object model and dozens of other new features.

PHP's development team includes dozens of developers, as well as dozens others working on PHP-related and supporting projects, such as PEAR, PECL, and documentation, and an underlying network infrastructure of well over one-hundred individual web servers on six of the seven continents of the world. Though only an estimate based upon statistics from previous years, it is safe to presume PHP is now installed on tens or even perhaps hundreds of millions of domains around the world.

**MySQL:-**

MySQL ("My S-Q-L" officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used relational database management system (RDBMS) and most widely used open-source RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: [TYPO3](http://en.wikipedia.org/wiki/TYPO3), [MODx](http://en.wikipedia.org/wiki/MODx" \o "MODx), [Joomla](http://en.wikipedia.org/wiki/Joomla" \o "Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress" \o "WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB" \o "PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB" \o "MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal) and other software. MySQL is also used in many high-profile, large-scale [websites](http://en.wikipedia.org/wiki/Website), including [Google](http://en.wikipedia.org/wiki/Google) (though not for searches), [Facebook](http://en.wikipedia.org/wiki/Facebook), [Twitter](http://en.wikipedia.org/wiki/Twitter), [Flickr](http://en.wikipedia.org/wiki/Flickr) and [YouTube](http://en.wikipedia.org/wiki/YouTube).

**Advantages of MySQL**

* It's easy to use: While a basic knowledge of SQL is required—and most relational databases require the same knowledge—MySQL is very easy to use. With only a few simple SQL statements, you can build and interact with MySQL.
* It's secure: MySQL includes solid data security layers that protect sensitive data from intruders. Rights can be set to allow some or all privileges to individuals. Passwords are encrypted.
* It's inexpensive: MySQL is included for free with NetWare® 6.5 and available by free download from [MySQL Web site](http://www.mysql.com).
* It's fast: In the interest of speed, MySQL designers made the decision to offer fewer features than other major database competitors, such as Sybase\* and Oracle\*. However, despite having fewer features than the other commercial database products, MySQL still offers all of the features required by most database developers.
* It's scalable: MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
* It manages memory very well: MySQL server has been thoroughly tested to prevent memory leaks.
* It supports Novell Cluster Services: MySQL on NetWare runs effectively with Novell® Cluster Services™, letting you add your database solution to a Novell cluster. If one server goes down, MySQL on an alternate server takes over and your customers won't know that anything happened.
* It runs on many operating systems: MySQL runs on many operating systems, including Novell NetWare, Windows\* Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others.
* It supports several development interfaces: Development interfaces include JDBC, ODBC, and scripting (PHP and Perl), letting you create database solutions that run not only in your NetWare 6.5 environment, but across all major platforms, including Linux, UNIX, and Windows.

DESIGN

**Data Flow Diagram**

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It’s also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD:-

In the DFD, four symbols are used and they are as follows.

1. A square defines a source (originator) or destination of system data.

1. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.

1. A circle or a “bubble “(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

1. An open rectangle is a data store-data at rest, or a temporary

repository of data.

i-) Context Level Diagram

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Online book store is shown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1st level DFD comes into play.

Add, Update Details of personals, Apply Leave, Manage Users & Approve Leave.

Staff Users

Get Queries.

Get Register, View Reply & Schedules,

Non Staff Users

Add Queries, Upload Documents

**0 Level DFD**

ADMIN

Get Leave Status

Add New Leave, Update Existing Leave Approve/Disapprove Faculty, HOD

Employee

Apply For Leave

Get Information about New Registered User

& Download Documents

**1st Level DFD**

Process Login

Enter Username & Password

ADMIN

ADMIN USER

I F LOGIN TRUE THEN ENTER TO ADMIN SECTION

Add update, delete Employee Record/ Leave

employeee

Add update, delete TO Record/Leave

Add, Update, Delete Leave

Approve/Disapprove Faculty A/C

Get Information About New Registered User

Leave

Employee A/C

Document

**2nd Level DFD (ADMIN):**

iii-) ER-DIAGRAMS

Entity-relationship model:-

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams. Such models are typically used in the first stage of information-system design; they are used, for example, to describe information needs and/or the type of information that is to be stored in the database during the requirement analysis. The data modeling technique, however, can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e. area of interest).

In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design.

Note that sometimes, both of these phases are referred to as "physical design".

There are a number of conventions for entity-relationship diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly relates to conceptual modeling. There are a range of notations more typically employed in logical and physical database design.

Leave\_detail

**Add**

Login

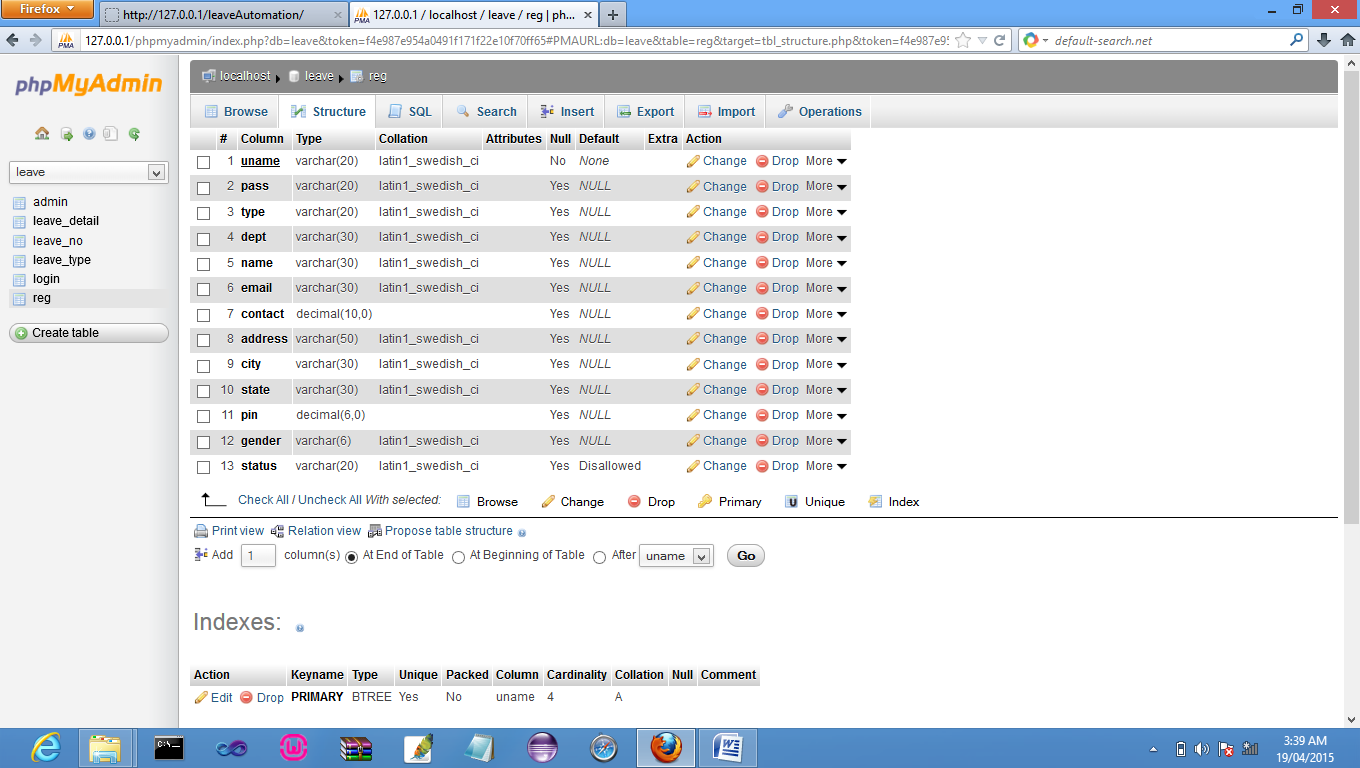
Admin

**Has**

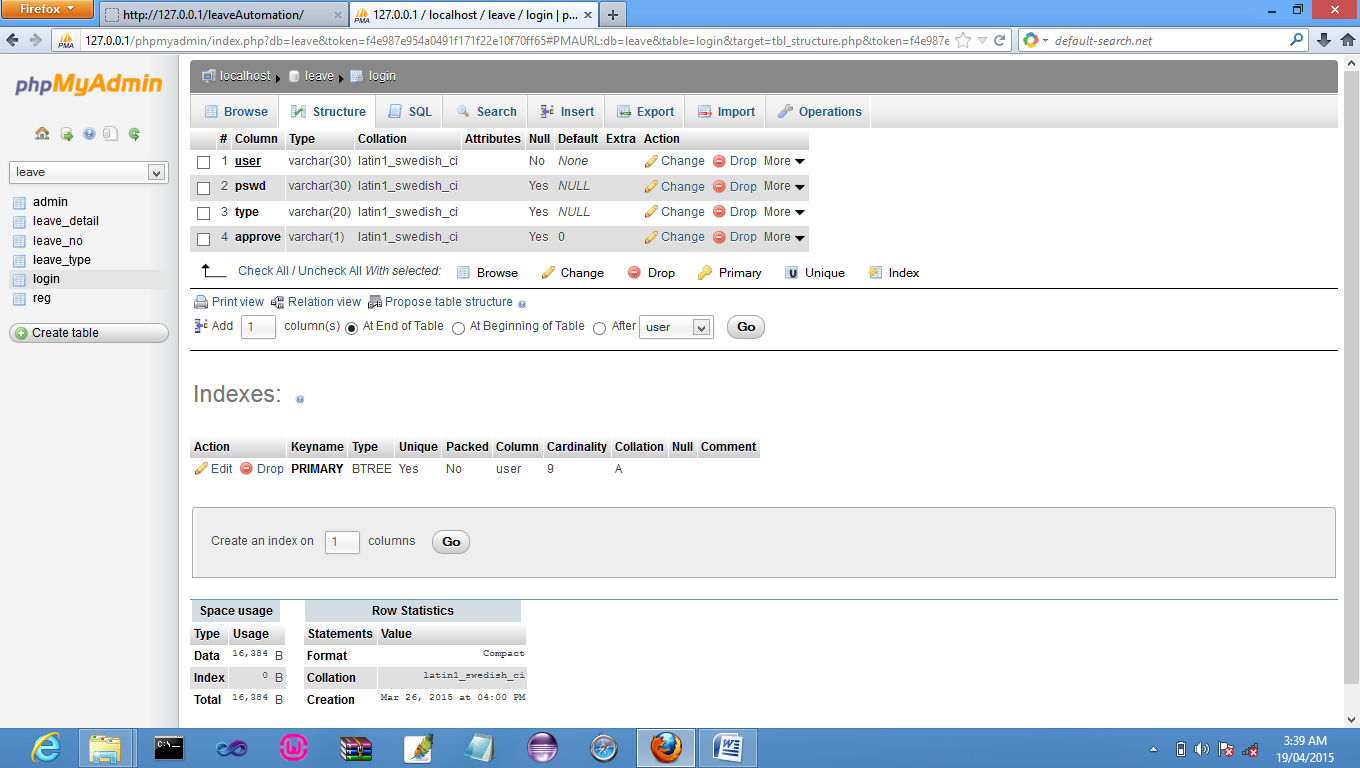
employee

DATABASE DESIGN

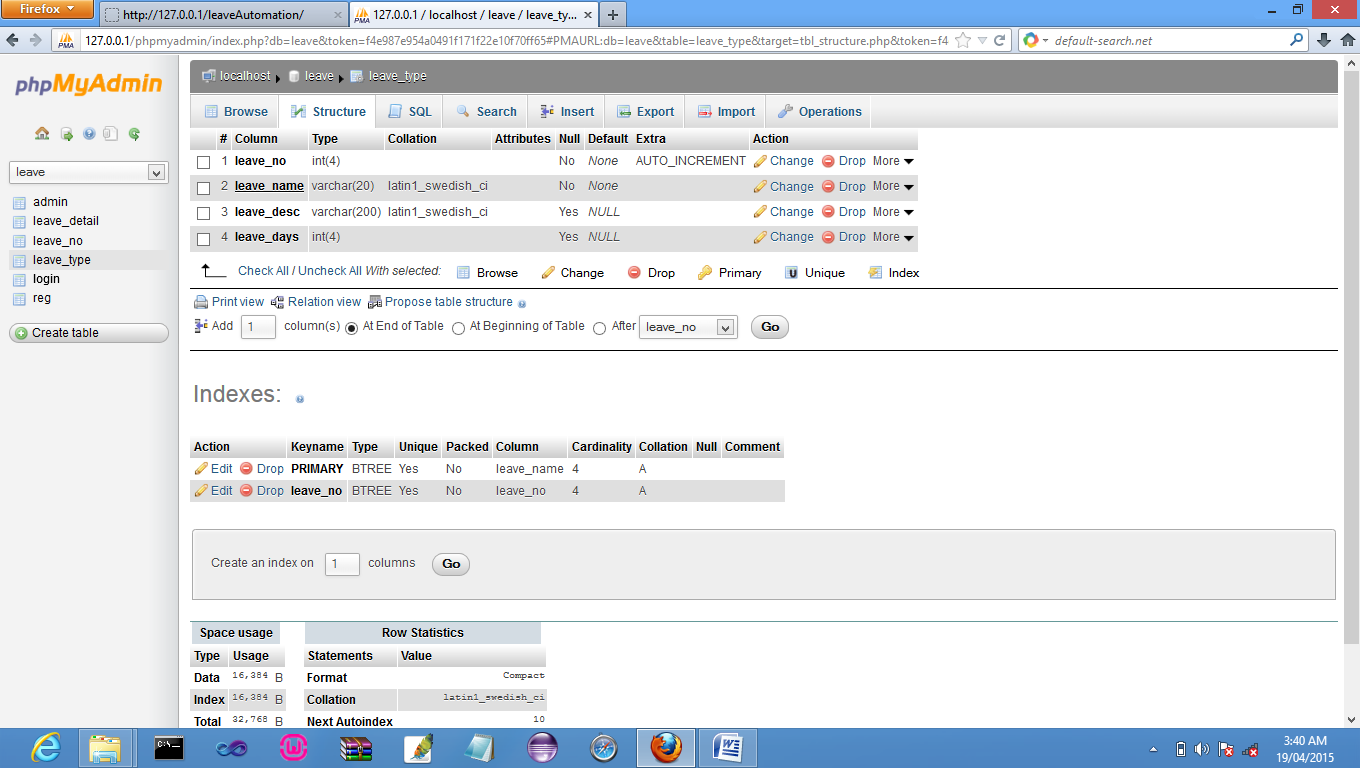
Registration table:



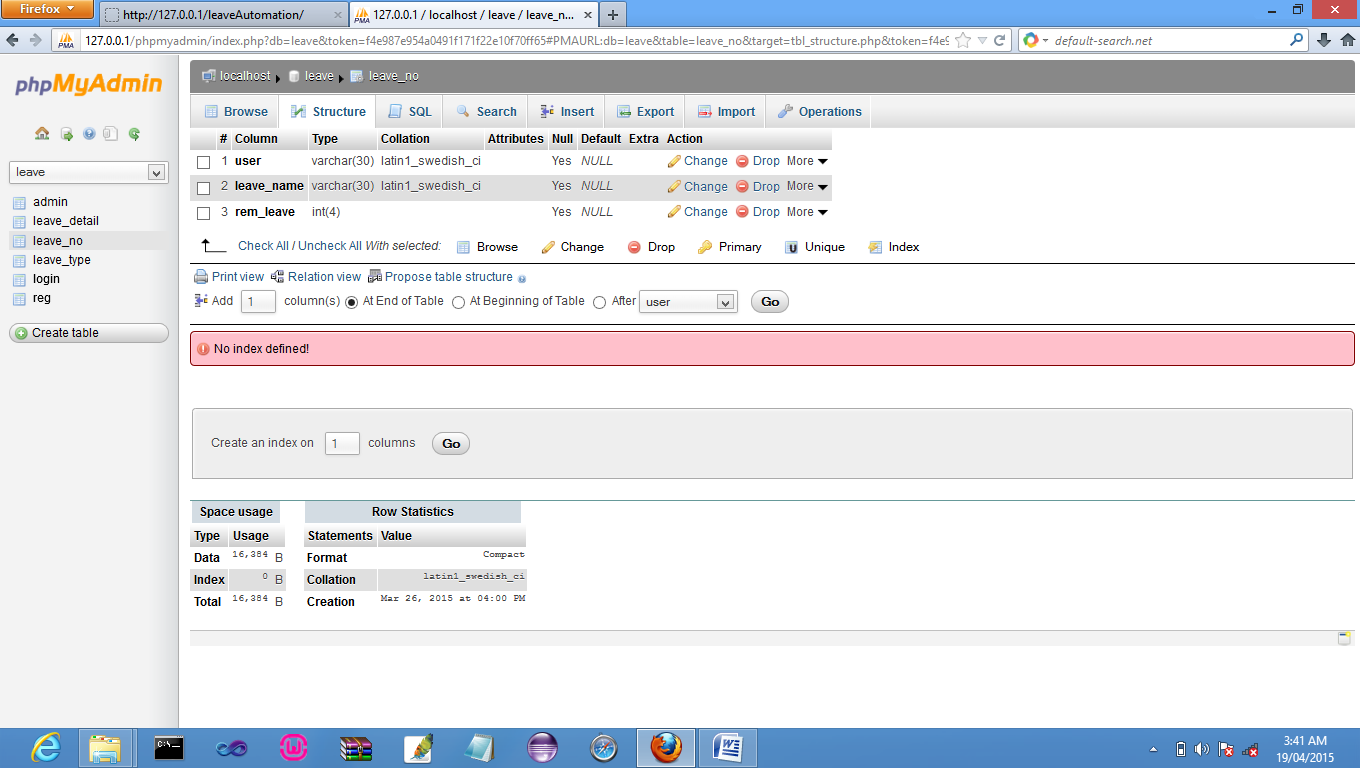
login table:



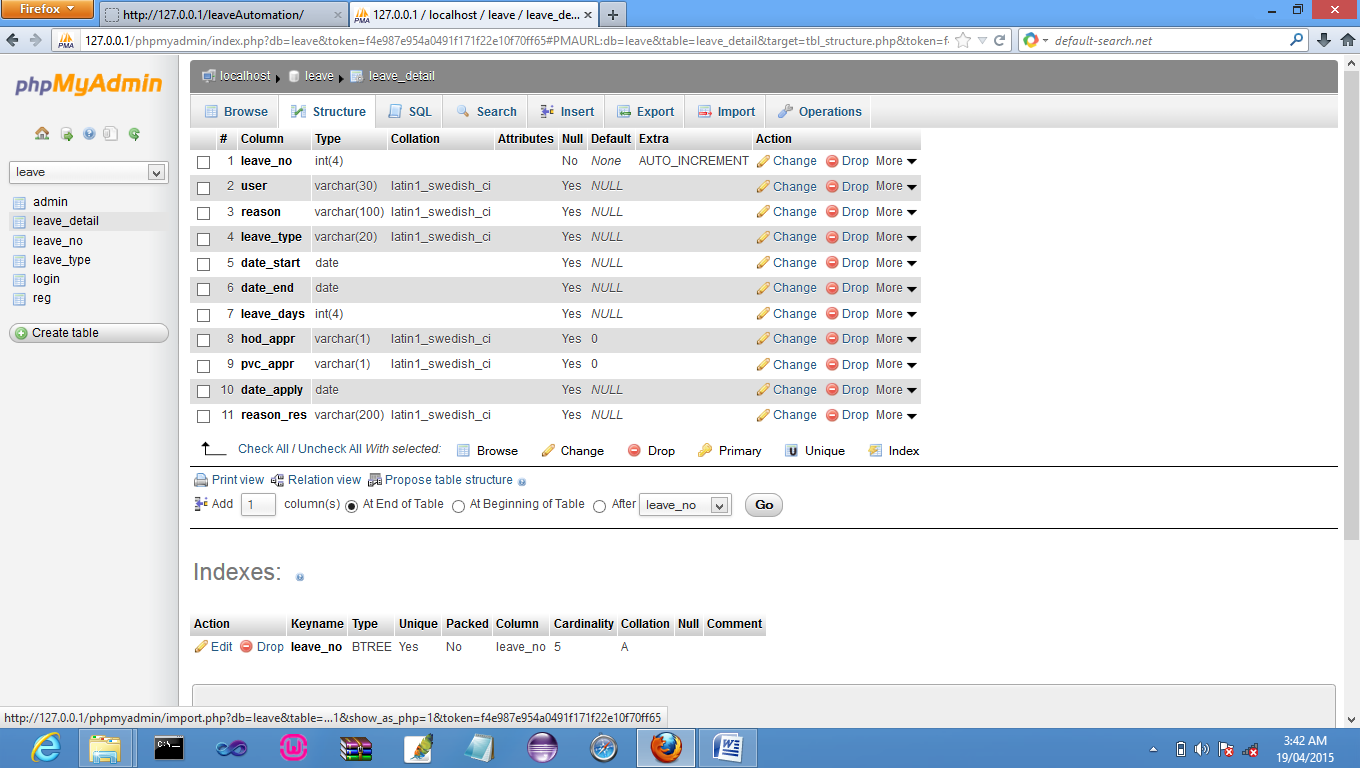
Leave\_type Table:



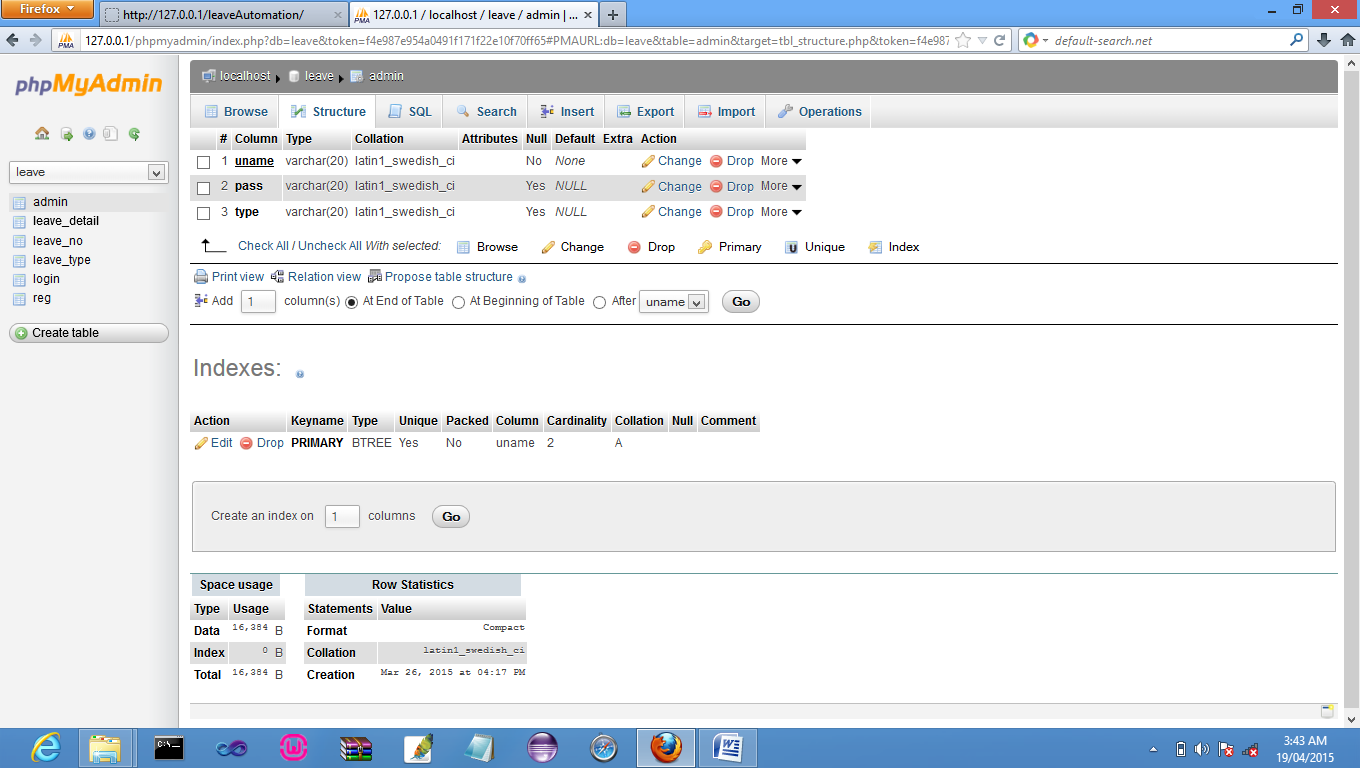
Leave\_no table:



Leave\_Details table :



Admin table:



**FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT:**

* Further enhancements to this project will be to Message Alert on Leave Approval
* Another enhancement would be allow Voice Calling conferencing.

CONCLUSION

We have successfully designed a project on Web Based Leave Automation. This project is designed for “Integral University”. This project is designed in PHP (Hypertext Preprocessor) which runs on Apache Web Server.

Keeping in view these facts we will develop successfully. Developing the project will help us some experience on website development.

REFERENCES

Designing and implementation phase: -

1. Software engineering: a practitioners approach by roger s pressman.
2. System analysis and design by Elias m. Ewad.
3. DBMS : Bipin C Desai

**Coding phase: -**

1. PHP (BIBLE)
2. PHP (BLACK Book)
3. PHP (Complete Reference)

Referenced Sites:

[www.w3school. com](http://www.w3school.microsoft.com)

[www.php.net](http://www.php.net)