

# **HOSTEL MANAGEMENT SYSTEM**

## **A PROJECT REPORT**

*Submitted by*

**Shreshth Sahay – 21BCS5409**

**B. Venu Gopal Reddy – 21BCS5361**

*in partial fulfillment for the award of the degree of*

## **BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCES AND ENGINEERING**  
with specialization in **ARTIFICIAL INTELLIGENCE** and  
**MACHINE LEARNING**



**Chandigarh University**

**DECEMBER 2023**



## **BONAFIDE CERTIFICATE**

Certified that this project report “**Hotel Management System**” is the bonafide work of “**Shreshth Sahay, Venu Gopal Reddy**” who carried out the project work under my/our supervision.

### **SIGNATURE**

Er. Aman Kaushik

**HEAD OF THE DEPARTMENT**  
**COMPUTER SCIENCE &**  
**ENGINEERING**

### **SIGNATURE**

Dr. Kalpana Singh

**SUPERVISOR**  
**COMPUTER SCIENCE &**  
**ENGINEERING**

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **Acknowledgment**

We would like to express our sincere gratitude to Dr. Kalpana Singh, Assistant Professor, Department of Computer Science and Engineering, Chandigarh University, for leading us through this project with unmatched expertise and tremendous encouragement.

We acknowledge Aman Kaushik, Head of the Computer Science and Engineering Department, for providing us with the tools we needed to finish the project.

We are incredibly appreciative of the Principal and Management at Chandigarh University, for their support and cooperation in getting this work done.

We thank Dr. Kalpana Singh, the project coordinator, for her ongoing encouragement and support.

We thank the entire teaching staff at the Department of CSE whose recommendations during reviews assisted us in finishing our project. We would like to express our gratitude to the Head of the Department of CSE, for his significant contribution to the success of our project.

# TABLE OF CONTENTS

|    |  |    |
|----|--|----|
| 1. | INTRODUCTION. ....                     | 01 |
|    | 1.1 Problem Definition. ....           | 01 |
|    | 1.2 User panel. ....                   | 01 |
|    | 1.2.1 User login. ....                 | 01 |
|    | 1.2.1.1 Home. ....                     | 01 |
|    | 1.2.1.2 Registration form. ....        | 02 |
|    | 1.2.1.3 Notice Board. ....             | 02 |
|    | 1.3 Admin panel. ....                  | 02 |
|    | 1.3.1 Administrator Login. ....        | 02 |
|    | 1.3.1.1 Home. ....                     | 02 |
|    | 1.3.1.1.1 Allotment of the hostels ... | 02 |
|    | 1.3.1.1.2 Vacating the rooms ...       | 02 |
| 2. | LITERATURE SURVEY .....                | 03 |
| 3. | SYSTEM ENVIRONMENT .....               | 03 |
|    | 3.1 Hardware Configuration. ....       | 04 |
|    | 3.2 Software Configuration. ....       | 04 |
|    | 3.3 Software Features. ....            | 04 |
|    | 3.3.1 php triad. ....                  | 04 |
|    | 2.3.1.1 php. ....                      | 04 |
|    | 2.3.1.1.1 Speed optimization. ....     | 07 |
|    | 2.3.1.1.2 Security .....               | 08 |
|    | 2.3.1.1.3 Syntax. ....                 | 08 |
|    | 2.3.1.1.4 Data types. ....             | 09 |
|    | 2.3.1.1.5 Functions. ....              | 10 |
|    | 2.3.1.1.6 Objects. ....                | 11 |

|   |    |
|---|----|
| 2.3.1.1.7 Resources.....                          | 12 |
| 2.3.1.2.1 Database tables. ....                   | 13 |
| 2.3.1.2.2 Queries. ....                           | 13 |
| 2.3.1.2.2.1 Create a Connection to Database. .... | 14 |
| 2.3.1.2.2.2 Closing a connection. ....            | 14 |
| 2.3.1.2.2.3 Create a database. ....               | 15 |
| 2.3.1.2.2.4 Create a table.....                   | 15 |
| 2.3.1.2.3 MySql functions. ....                   | 16 |
| 2.3.1.2 Macromedia Dreamweaver 8.....             | 17 |
| 2.3.1.2.1 Dreamweaver & accessibility .....       | 18 |
| 2.3.1.3 phpMyAdmin. ....                          | 19 |
| 2.3.1.3.1 Requirements.....                       | 21 |
| 2.3.1.4 Apache web server .....                   | 22 |
| 4. RESULT ANALYSIS.....                           | 24 |
| 4.1 Existing System.....                          | 25 |
| 5. SYSTEM DESIGN.....                             | 26 |
| 5.1 Input Design.....                             | 27 |
| 5.1.1 Administrator.....                          | 27 |
| 5.2 Process Design. ....                          | 28 |
| 5.2.1 Data Flow Diagram. ....                     | 29 |
| 5.3 Database Design.....                          | 31 |
| 5.4 Output Design. ....                           | 35 |
| 6. CONCLUSION. ....                               | 48 |
| 7. APPENDIX - I. ....                             | 53 |
| APPENDIX - II. ....                               | 64 |
| APPENDIX - III. ....                              | 66 |

## **LIST OF TABLES**

|                                     |    |
|-------------------------------------|----|
| 4.1. Student account creation. .... | 29 |
| 4.2. Administrator Login .....      | 29 |
| 4.3. Application form .....         | 30 |
| 4.4. Allotment. ....                | 31 |
| 4.5. Vacating and editing.....      | 31 |
| 4.6. Notice Board .....             | 32 |

## **LIST OF FIGURES**

|                               |    |
|-------------------------------|----|
| 4.1 Context Level DFD.....    | 25 |
| 4.2 Student Module.....       | 25 |
| 4.3 Registration Process..... | 26 |
| 4.4 Administrator Module..... | 32 |
| 4.5 Allotment Process. ....   | 32 |
| 4.6 Vacating Process.....     | 33 |

## **LIST OF SYMBOLS**

|                            |    |
|----------------------------|----|
| 4.2.1 Rectangle.....       | 26 |
| 4.2.2 Arrow .....          | 26 |
| 4.2.3 Circle. ....         | 26 |
| 4.2.4 Open rectangle. .... | 26 |

# ABSTRACT

As the name specifies “HOSTEL MANAGEMENT SYSTEM” is a software developed for managing various activities in the hostel. For the past few years the number of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. 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 hostel 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.

- Less human error
- Strength and strain of manual labour can be reduced
- High security
- Data redundancy can be avoided to some extent
- Data consistency
- Easy to handle
- Easy data updating
- Easy record keeping
- Backup data can be easily generated

## **ABBREVIATIONS**

1. **HMS:** Hostel Management System
2. **CMS:** Content Management System (used for managing website content)
3. **CRUD:** Create, Read, Update, Delete (common operations in database management)
4. **ERP:** Enterprise Resource Planning (integrating various aspects of hostel management)
5. **GUI:** Graphical User Interface (the visual aspect of the website)
6. **API:** Application Programming Interface (enabling communication between different



# **CHAPTER 1: INTRODUCTION**

## **1.1 Problem definition**

We have got nine hostels in our university, which consist of four boy's hostel and five girl's hostel. All these hostels at present are managed manually by the hostel 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 hostel 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.

## **1.2 User Panel**

### **1.2.1 User Login**

#### **1.2.1.1 Home**

1. This consist of the different pop-up menus showing the details of the different hostels
2. It contain a link to the CUSAT official web-site.
3. It allows the different users to access the registration forms.
4. He can view the Student administration division of the different hostels and also view the notice boards.

### **1.2.1.2 Registration Form**

This section provides an online form to the students which can be filled by them, and a copy of the filled page can be taken in the printed form. This is later submitted to the Hostel authorities which can be verified by them before allotting them to the respective hostels.

### **1.2.1.3 Notice Board**

All the 9 hostels have their Notice boards. Any change in the Hostel fee, mess fee will be shown in this. It can be also used for different notifications.

## **1.3 Admin Panel**

### **1.3.1 Administrator Login**

#### **1.3.2.1 Home**

The Administrator can :

1. Allot different students to the different hostels.
2. Vacate the students for the hostels.
3. Control the status of the fee payment.
4. Edit the details of the students & modify the student records.

#### **1.3.2.1.1 Allotment of the hostels**

There will be pre-defined criteria for the admission to the hostels. He checks the attested application forms of the students obtained from the internet and verifies it with the student database. If the students are found eligible then they are allotted to the hostel.

#### **1.3.2.1.2 Vacating the rooms**

As the student's course is over they will vacate their rooms. So it is required for the administrator to remove their records from the database tables. This section includes the option for the room vacation and the deletion of the particular record from the database.

## **CHAPTER – 2 LITERATURE SURVEY**

### **2.1 Timeline of the Reported Problem as Investigated Throughout the World**

Understanding the historical context and evolution of hostel management systems is crucial for identifying the gaps and challenges that researchers have encountered over time. The timeline of the reported problem showcases the progression from manual record-keeping to the integration of digital solutions. Initially, hostels relied on traditional methods for tasks such as room allocation, fee collection, and attendance tracking. As technology advanced, researchers started exploring computerized systems to streamline these processes.

Throughout the years, studies have delved into the issues faced by hostel administrators and how technology can be leveraged to address these challenges. The timeline provides insights into the chronological development of solutions and the emergence of new problems with evolving technologies. It highlights the dynamic nature of hostel management and the need for adaptive systems.

### **2.2 Bibliometric Analysis**

A bibliometric analysis involves evaluating the existing literature on hostel management systems to identify key trends, influential authors, and frequently addressed issues. By analyzing the body of research, we gain a comprehensive overview of the domain's landscape. This analysis will include an examination of the most cited papers, prevalent methodologies, and recurring themes, offering valuable insights into the prevailing concerns and favored approaches within the research community.

### **2.3 Proposed Solutions by Different Researchers**

Researchers worldwide have proposed various solutions to enhance hostel management systems. These solutions range from introducing user-friendly interfaces and mobile applications for real-time updates to implementing advanced algorithms for efficient room allocation and resource optimization. Exploring these proposed solutions provides a foundation for understanding the diverse

approaches and technologies used to tackle similar issues.

## **2.4 Summary Linking Literature Review with the Project**

The literature review culminates in a summary that draws connections between the existing body of knowledge and the project at hand. It identifies gaps in the current solutions, justifies the need for a new hostel management system, and lays the groundwork for the subsequent chapters.

## **2.5 Problem Definition, Goals, and Objectives**

The literature survey sets the stage for clearly defining the problem, establishing the goals, and outlining the objectives of the proposed hostel management system. By synthesizing insights from existing research, the project's problem statement becomes more nuanced, and the objectives are refined based on the shortcomings identified in the literature. This chapter serves as a bridge between the theoretical foundation provided by the literature review and the practical implementation to be detailed in the subsequent chapters.

## CHAPTER 3

### SYSTEM ENVIRONMENT

#### **3.1 Hardware Configuration**

- Pentium IV Processor
- 512 MB RAM
- 40GB HDD
- 1024 \* 768 Resolution Color Monitor

**Note: This is not the “System Requirements”.**

#### **3.2 Software Configuration**

- OS : Windows XP
- PHPTriad (PHP, MySQL, Apache, and PHPMyAdmin)

#### **3.3 Software Features**

##### **2.3.1 PHP TRIAD**

PHPTriad installs a complete working PHP/MySQL server environment on Windows platforms (9x/NT). Installs PHP, MySQL, Apache, and PHPMyAdmin.

##### **2.3.1.1 PHP**

**PHP** is a scripting language originally designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications. While PHP was originally created by Rasmus Lerdorf in 1995, the main implementation of PHP is now produced by **The PHP Group** and serves as the *de facto* standard for PHP as there is no formal specification. PHP is free software released under the PHP License, however it is incompatible with the GNU General Public License

(GPL), due to restrictions on the usage of the term *PHP*. It is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can be deployed on most web servers and on almost every operating system and platform free of charge. PHP is installed on more than 20 million websites and 1 million web servers.

*PHP* originally stood for Personal Home Page. It began in 1994 as a set of Common Gateway Interface binaries written in the C programming language by the Danish/Greenlandic programmer Rasmus Lerdorf. Lerdorf initially created these Personal Home Page Tools to replace a small set of Perl scripts he had been using to maintain his personal homepage. The tools were used to perform tasks such as displaying his résumé and recording how much 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, enabling the building of simple, dynamic web applications.

Lerdorf released PHP publicly on June 8, 1995 to accelerate 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 and Andi Gutmans, two Israeli developers at the Technion IIT, rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive initialism *PHP: Hypertext Preprocessor*. The development team officially released PHP/FI 2 in November 1997 after months of beta testing. Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999. They also founded Zend Technologies in Ramat Gan, Israel.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was 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, the PHP Data Objects extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. The most recent update released by The PHP Group is for the older PHP version 4 code branch.

In 2008, PHP 5 became the only stable version under development. Late static binding has been missing from PHP and will be added in version 5.3. PHP 6 is under development alongside PHP 5. Major changes include the removal of `register_globals`, magic quotes, and safe mode. The reason for the removals was because `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.

PHP does not have complete native support for Unicode or multibyte strings; Unicode support will be included in PHP 6. Many high profile open source projects ceased to support PHP 4 in new code as of February 5, 2008, due to the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5. It runs in both 32-bit and 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 in a 64-bit Windows environment. There is a third-party distribution available for 64-bit Windows.

## **Usage**

PHP is a general-purpose scripting language that is especially suited for web development. PHP generally runs on a web server, taking PHP code as its input and creating web pages as output. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP's principal focus is server-side scripting, and it is similar to other server-side scripting

languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems' JavaServer Pages, and mod\_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, and Zend Framework, offering features similar to other web application frameworks.

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

As of April 2007, over 20 million Internet domains were hosted on servers with PHP installed, and PHP was recorded as the most popular Apache module. Significant websites are written in PHP including the user-facing portion of Facebook, Wikipedia (MediaWiki), Yahoo!, MyYearbook, , Digg, Wordpress and Tagged. In addition to server-side scripting, PHP can be used to create stand-alone, compiled applications and libraries, it can be used for shell scripting, and the PHP binaries can be called from the command line.

#### **2.3.1.1.1 Speed optimization**

As with many scripting languages, PHP scripts are normally kept as human-readable source code, even on production web servers. In this case, PHP scripts will be compiled at runtime by the PHP engine, which increases their execution time. PHP scripts are able to be compiled before runtime using PHP compilers as with other programming languages such as C (the language PHP and its extensions are written in). Code optimizers aim to reduce the computational complexity of the compiled code by reducing its size and making other changes that can reduce the execution time with the overall goal of improving performance. The nature of the PHP compiler is such that there are often opportunities for code optimization, and an example of a code optimizer is the Zend Optimizer PHP extension.

Another approach for reducing overhead for high load PHP servers is using PHP accelerators. These can offer significant performance gains by caching the compiled form of a PHP script in shared memory to avoid the overhead of parsing and compiling the code every time the script runs.



### 2.3.1.1.2 Security

The National Vulnerability Database stores all vulnerabilities found in computer software. The overall proportion of PHP-related vulnerabilities on the database amounted to: 12% in 2003, 20% in 2004, 28% in 2005, 43% in 2006, 36% in 2007, and 35% in 2008. Most of these PHP-related vulnerabilities can be exploited remotely: they allow hackers to steal or destroy data from data sources linked to the webserver (such as an SQL database), send spam or contribute to DOS attacks using malware, which itself can be installed on the vulnerable servers.

These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent. Recognizing that programmers cannot be trusted, some languages include taint checking to detect automatically the lack of input validation which induces many issues. However, such a feature is being developed for PHP

Hosting PHP applications on a server requires a careful and constant attention to deal with these security risks. There are advanced protection patches such as Suhosin and Hardening-Patch, especially designed for web hosting environments. Installing PHP as a CGI binary rather than as an Apache module is the preferred method for added security. With respect to securing the code itself, PHP code can be obfuscated to make it difficult to read while remaining functional.

### 2.3.1.1.3 Syntax

```
<html>
<head>
    <title>PHP Test </title>
</head>
<body>
    <?php echo "<p> Hello World </p>"; ?>
</body></html>
```

Note : - Code in bold letters shows the PHP code embedded within HTML

PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not parsed by PHP. The most common delimiters are `<?php` and `?>`, which are open and close delimiters respectively. `<script language="php">` and `</script>` delimiters are also available. Short tags can be used to start PHP code, `<?` or `<?=` (which is used to echo back a string or variable) and the tag to end PHP code, `?>`. These tags are commonly used, but like ASP-style tags (`<%` or `<%=` and `%>`), they are less portable as they can be disabled in the PHP configuration. For this reason, the use of short tags and ASP-style tags is discouraged. The purpose of these delimiters is to separate PHP code from non-PHP code, including HTML.

Variables are prefixed with a dollar symbol and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted ("" ) and heredoc strings allow the ability to embed a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: `/** */` serves as block comments, and `//` as well as `#` are used for inline 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.

#### **2.3.1.1.4 Data types**

PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit signed integers. 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, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. 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 with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, or heredoc syntax. The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

#### **2.3.1.1.5 Functions**

PHP has hundreds of base functions and thousands more from extensions. These functions are well documented on the PHP site, but unfortunately, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming.

#### **Version 5.2 and earlier**

Functions are not first-class functions and can only be referenced by their name—directly or dynamically by a variable containing the name of the function. User-defined functions can be created at any time without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero argument class constructor functions called with the PHP new operator, where parentheses are optional. PHP supports quasi-anonymous functions through the `create_function()` function, although they are not true anonymous functions because anonymous functions are nameless, but functions can only be referenced by name, or indirectly through a variable `$function_name()`, in PHP.

#### **Version 5.3 and newer**

PHP gained support for first-class functions and closures. True anonymous functions are supported function `getAdder($x)` 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 another functions, etc. For more details see [Lambda functions and closures RFC](#).

#### **2.3.1.1.6 Objects**

Basic object-oriented programming functionality was added in PHP 3. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like primitive types. The drawback of this method was that the whole object was copied when a variable was assigned or passed as a parameter to a method. In the new approach, objects are referenced by handle, and not by value. PHP 5 introduced private and protected member variables and methods, along with abstract classes and final classes as well as abstract methods and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model.

Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. Objects implementing `ArrayAccess` can be used with array syntax and objects implementing `Iterator` or `IteratorAggregate` can be used with the `foreach` language construct. There is no virtual table feature in the engine, so static variables are bound with a name instead of a reference at compile time.

If the developer creates a copy of an object using the reserved word *clone*, the Zend engine will check if a `__clone()` method has been defined or not. If not, it will call a default `__clone()` which will copy the object's properties. If a `__clone()` method is defined, then it will be responsible for setting the necessary properties in the created object. For convenience, the engine will supply a function that imports the properties of the source object, so that the programmer can start with a by-value replica of the source object and only override properties that need to be changed.

#### **2.3.1.1.7 Resources**

PHP includes free and open source libraries with the core build. PHP is a fundamentally Internet-aware system with modules built in for accessing FTP servers, many database servers, embedded SQL libraries such as embedded PostgreSQL, MySQL and SQLite, LDAP servers, and others. Many functions familiar to C programmers such as those in the stdio family are available in the standard PHP build. PHP has traditionally used features such as “`magic_quotes_gpc`” and “`magic_quotes_runtime`” which attempt to escape apostrophes (') and quotes (") in strings in the assumption that they will be used in databases, to prevent SQL injection attacks. This leads to confusion over which data is escaped and which is not, and to problems when data is not in fact used as input to a database and when the escaping used is not completely correct. To make code portable between servers which do and do not use magic quotes, developers can preface their code with a script to reverse the effect of magic quotes when it is applied.

PHP allows developers to write extensions in C to add functionality to the PHP language. These can then be compiled into PHP or loaded dynamically at runtime. Extensions have been written to add support for the Windows API, process management on Unix-like operating systems, multibyte strings (Unicode), cURL, and several popular compression formats. Some more unusual features include integration with Internet Relay Chat, dynamic generation of images and Adobe Flash content, and even speech synthesis. The PHP Extension Community Library (PECL) project is a repository for extensions to the PHP language. Zend provides a certification exam for programmers to become certified PHP developers.

### **2.3.1.2 MY SQL**

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

MySQL is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQLAB now a subsidiary of Sun Microsystems, which holds the copyright to most of the codebase. The project's source code is available under terms of the GNU General Public Licence, as well as under a variety of proprietary agreements.

MySQL is a database. The data in MySQL is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows. Databases are useful when storing information categorically. A company may have a database with the following tables: "Employees", "Products", "Customers" and "Orders".

#### **2.3.1.2.1 Database Tables**

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

#### **2.3.1.2.2 Queries**

A query is a question or a request. With MySQL, we can query a database for specific information and have a recordset returned.

### 2.3.1.2.2.1 Create a connection to a database

Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the `mysql_connect()` function.

#### Syntax

```
mysql_connect (servername,username,password) ;
```

| Parameter  | Description   |
|------------|---|
| servername | Optional. Specifies the server to connect to. Default value is "localhost:3306"                                     |
| username   | Optional. Specifies the username to log in with. Default value is the name of the user that owns the server process |
| password   | Optional. Specifies the password to log in with. Default is ""  |

#### Example

In the following example we store the connection in a variable (`$con`) for later use in the script. The “die” part will be executed if the connection fails:

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// some code
?>
```

### 2.3.1.2.2.2 Closing a Connection

The connection will be closed automatically when the script ends. To close the connection before, use the `mysql_close()` function:

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// some code
mysql_close($con);
?>
```

#### 2.3.1.2.2.3 Create a Database

The CREATE DATABASE statement is used to create a database in MySQL.

##### **Syntax**

CREATE DATABASE database\_name

To get PHP to execute the statement above we must use the mysql\_query() function. This function is used to send a query or command to a MySQL connection.

#### 2.3.1.2.2.4 Create a Table

The CREATE TABLE statement is used to create a table in MySQL

##### **Syntax**

```
CREATE TABLE table_name
(
    column_name1 data_type,
    column_name2 data_type,
    column_name3 data_type,
    ....
)
```



### 2.3.1.2.3 MySQL Functions

`mysql_affected_rows` — Get number of affected rows in previous MySQL operation

`mysql_change_user` — Change logged in user of the active connection

`mysql_client_encoding` — Returns the name of the character set

`mysql_close` — Close MySQL connection

`mysql_connect` — Open a connection to a MySQL Server

`mysql_create_db` — Create a MySQL database

`mysql_data_seek` — Move internal result pointer

`mysql_db_name` — Get result data

`mysql_db_query` — Send a MySQL query

`mysql_drop_db` — Drop (delete) a MySQL database

`mysql_errno` — Returns the numerical value of the error message from previous MySQL operation

`mysql_error` — Returns the text of the error message from previous MySQL operation

`mysql_escape_string` — Escapes a string for use in a `mysql_query`

`mysql_fetch_array` — Fetch a result row as an associative array, a numeric array, or both

`mysql_fetch_assoc` — Fetch a result row as an associative array

`mysql_fetch_field` — Get column information from a result and return as an object

`mysql_fetch_lengths` — Get the length of each output in a result

`mysql_fetch_object` — Fetch a result row as an object

wsmysql\_num\_rows — Get number of rows in result

mysql\_pconnect — Open a persistent connection to a MySQL server

mysql\_ping — Ping a server connection or reconnect if there is no connection

mysql\_query — Send a MySQL query

mysql\_result — Get result data

mysql\_select\_db — Select a MySQL database

mysql\_set\_charset — Sets the client character set

mysql\_stat — Get current system status

mysql\_tablename — Get table name of field

mysql\_thread\_id — Return the current thread ID

mysql\_unbuffered\_query — Send an SQL query to MySQL, without fetching and buffering the result

*(See Appendix 2 for more My\_SQL Functions.)*

### **2.3.1.3 Macromedia Dreamweaver 8**

Is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Whether you enjoy the control of hand-coding HTML or prefer to work in a visual editing environment, Dreamweaver provides you with helpful tools to enhance your web creation experience.

The visual editing features in Dreamweaver let you quickly create pages without writing a line of code. You can view all your site elements or assets and drag them from an easy-to-use panel directly into a document. You can streamline your development workflow by creating and editing images in Macromedia Fireworks or another graphics application, then importing them directly into Dreamweaver, or by adding Macromedia Flash objects.

Dreamweaver also provides a full-featured coding environment that includes code-editing tools (such as code coloring and tag completion) and language reference material on Cascading Style Sheets (CSS), JavaScript, and ColdFusion Markup Language (CFML), among others. Macromedia Roundtrip HTML technology imports your hand-coded HTML documents without reformatting the code; you can then reformat code with your preferred formatting style. Dreamweaver also enables you to build dynamic database-backed web applications using server technologies such as CFML, ASP.NET, ASP, JSP, and PHP.

#### **2.3.1.3.1 Dreamweaver and accessibility**

Accessibility refers to making websites and web products usable for people with visual, auditory, motor, and other disabilities. Examples of accessibility features for software products and websites include screen reader support, text equivalents for graphics, keyboard shortcuts, change of display colors to high contrast, and so on. Dreamweaver provides tools that make the product accessible and tools that help you author accessible content:

**Using Dreamweaver accessibility features** For Dreamweaver web designers who need to use accessibility features, Dreamweaver offers screen reader support, keyboard navigation, and operating system accessibility support. For more information, see [Using Dreamweaver accessibility features](#).

**Authoring for accessibility** For Dreamweaver web designers who need to create accessible content, Dreamweaver assists you in creating accessible pages that contain useful content for screen readers and comply with government guidelines.

Dreamweaver provides dialog boxes that prompt you to enter accessibility attributes when you insert page elements (see [Optimizing the workspace for accessible page design](#)). For example, the accessibility dialog box for images reminds you to add text equivalents for graphics. Then, when the image appears on a page for a user with visual disabilities, the screen reader reads the description.

### 2.3.1.3.2 Laying Out Pages with CSS

In Macromedia Dreamweaver 8, you can use CSS styles to lay out your page. You can either insert div tags manually and apply CSS positioning styles to them, or you can use Dreamweaver layers to create your layout. A layer in Dreamweaver is an HTML page element—specifically, a div tag, or any other tag—that has an absolute position assigned to it. Whether you use CSS, tables, or frames to lay out your pages, Dreamweaver has rulers and grids for visual guidance in your layout. Dreamweaver also has a tracing image feature, which you can use to re-create a page design that was created in a graphics application.

### Client-side role of forms

Forms support the client side of the client-server relationship. When a visitor enters information into a form displayed in a web browser (the client) and clicks the submit button, the information is sent to the server where a server-side script or application processes it. Common server-side technologies used for processing form data include Macromedia ColdFusion, Microsoft Active Server Pages (ASP), and PHP. The server responds by sending requested information back to the user (or client), or performing some action based on the form's contents.

*( Note : - See Appendix 1 for more about Macromedia Dreamweaver 8 and phpMyAdmin )*

### 2.3.1.4 phpMAdmin

**phpMyAdmin** is an open source tool written in PHP intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. Currently it can create and drop databases, create/drop/alter tables, delete/edit/add fields, execute any SQL statement, manage users and permissions, and manage keys on fields. While you still have the ability to directly execute any SQL statement, phpMyAdmin can manage a whole MySQL server (needs a super-user) as well as a single database. To accomplish the latter you'll need a properly set up MySQL user who can read/write only the desired database. It's up to you to look up the appropriate part in the MySQL manual.

## phpMyAdmin can:

- browse and drop databases, tables, views, fields and indexes
- create, copy, drop, rename and alter databases, tables, fields and indexes
- maintenance server, databases and tables, with proposals on server configuration
- execute, edit and bookmark any SQL-statement, even batch-queries
- load text files into tables
- create and read dumps of tables
- export data to various formats: CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel and L<sup>A</sup>T<sub>E</sub>X formats
- administer multiple servers
- manage MySQL users and privileges
- check referential integrity in MyISAM tables
- using Query-by-example (QBE), create complex queries automatically connecting required tables
- create PDF graphics of your Database layout
- search globally in a database or a subset of it
- transform stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
- support InnoDB tables and foreign keys
- support mysqli, the improved MySQL extension

## **A word about users:**

Many people have difficulty understanding the concept of user management with regards to phpMyAdmin. When a user logs in to phpMyAdmin, that username and password are passed directly to MySQL. phpMyAdmin does no account management on its own (other than allowing one to manipulate the MySQL user account information); all users must be valid MySQL users.

- 1) phpMyAdmin can compress (Zip, GZip -RFC 1952- or Bzip2 formats) dumps and CSV exports if you use PHP with Zlib support (`--with-zlib`) and/or Bzip2 support (`--with-bz2`). Proper support may also need changes in `php.ini`. a phpMyAdmin screen appears as shown below.

### **2.3.1.4. 1 Requirements**

- o **PHP**
- o You need PHP 5.2.0 or newer, with session support and the Standard PHP Library (SPL) extension.
- o To support uploading of ZIP files, you need the PHP zip extension.
- o For proper support of multibyte strings (eg. UTF-8, which is currently default), you should install mbstring and ctype extensions.
- o You need GD2 support in PHP to display inline thumbnails of JPEGs (`"image/jpeg: inline"`) with their original aspect ratio
- o When using the "cookie" authentication method, the mcrypt extension is strongly suggested for most users and is required for 64-bit machines. Not using mcrypt will cause phpMyAdmin to load pages significantly slower.

### 2.3.1.5 Apache Web server

Often referred to as simply *Apache*, a public-domain open source Web server developed by a loosely-knit group of programmers. The first version of Apache, based on the NCSA httpd Web server, was developed in 1995.

Core development of the Apache Web server is performed by a group of about 20 volunteer programmers, called the *Apache Group*. However, because the source code is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. In many respects, development of Apache is similar to development of the Linux operating system.

The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms. The name is a tribute to the Native American Apache Indian tribe, a tribe well known for its endurance and skill in warfare. A common misunderstanding is that it was called Apache because it was developed from existing NCSA code plus various patches, hence the name *a patchy server*, or Apache server.

Apache consistently rates as the world's most popular Web server according to analyst surveys. Apache has attracted so much interest because it is full-featured, reliable, and free. Originally developed for UNIX™ operating systems, Apache has been updated to run on Windows, OS/2, and other platforms. One aspect of Apache that some site administrators find confusing — especially those unfamiliar with UNIX-style software — is its configuration scheme. Instead of using a point-and-click graphic user interface (GUI) or Windows Registry keys as most other modern software packages, Apache generally relies on simple text files for its configuration settings.

## Configuration Files

Apache uses a system of three text files for managing its configuration data. All three of these files (almost always) appear in Apache's `/conf` directory and are designed to be edited by system administrators:

1. `httpd.conf` for general settings
2. `srn.conf` for resource settings
3. `access.conf` for security settings

When Apache first starts, these files are processed in the order shown above. Originally, the initial installation of Apache included default entries within each of the three files. In the most recent versions of Apache, however, the default installation has changed. Now `httpd.conf` is treated as the “master” configuration file and it contains all of the settings. Both `srn.conf` and `access.conf` still exist in the installation, but they contain no settings and are empty except for some comments.

### Inside `Httpd.conf`

Traditionally `httpd.conf` contained general settings such as the `ServerName` and `Port` number. These entries appear as follows in the file: `ServerName compnetworking.about.com Port 80` The term “httpd” stands for *HTTP Daemon*. Recall that in a UNIX environment, the term *daemon* refers to a type of process designed to launch at system boot and continue running for very long periods of time. This file contains a number of other entries (technically called directives), but for most of these, modifications are optional. Probably the most useful of these entries is `ServerAdmin`.

### Access and Security Settings

It is recommended practice now for Apache administrators to manage their resource and security settings from `httpd.conf`. Administrators of older versions of Apache can simply cut their entries from `srn.conf` and `access.conf` and paste them into the master file. If an administrator wants to go one step further and delete the two empty files, they should also place the following entries in `httpd.conf` to prevent Apache from attempting to access them.



# CHAPTER 3

## SYSTEM DESIGN

### **3.1 Input Design**

The system design is divided into two portions. The Administrator section and the User(student's) section.

#### **3.1.1 Administrator**

1. The Administrator can allot different students to the different hostels.
2. He can vacate the students for the hostels.
3. He can control the status of the fee payment.
4. He can edit the details of the students. He can change their rooms, edit and delete the student records.

A process of converting user originated inputs to a computer-based format. Input design is an important part of development process since inaccurate input data are the most common cause of errors in data processing. Erroneous entries can be controlled by input design. It consists of developing specifications and procedures for entering data into a system and must be in simple format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. In input data design, we design the source document that capture the data and then select the media used to enter them into the computer.

There are two major approaches for entering data into the computer. They are

- Menus.
- Dialog Boxes.

#### **Menus**

A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user chooses from a list of options. A menu limits a user choice of response but reduce the chances for error in data entry.

## **Dialog Box**

Dialog boxes are windows and these windows are mainly popup, which appear in response to certain conditions that occur when a program is run. It allows the display of bitmaps and pictures. It can have various controls like buttons, text boxes, list boxes and combo boxes. Using these controls we can make a 'dialog' with the program.

The proposed system has three major inputs. They are Machine Registration, Machine Scheduling and Request Form.

### **3.2 Process Design**

Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow Diagram and System Flow chart are the tools used for process design.

System Flow Chart is a graphical representation of the system showing the overall flow of control in processing at the job level; specifies what activities must be done to convert from a physical to logical model.

Data Flow Diagram is the logical representation of the data flow of the project. The DFD is drawn using various symbols. It has a source and a destination. The process is represented using circles and source and destination are represented using squares. The data flow is represented using arrows.

One reader can easily get the idea about the project through Data Flow Diagram.

## SYMBOLS USED IN DATA FLOW DIAGRAM

3.2.1



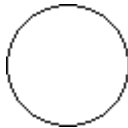
- source rectangle, which defines or destination

3.2.2



-Arrow, which shows dataflow.

3.2.3



-Circle, which represent a process that transforms incoming data into outgoing flow.

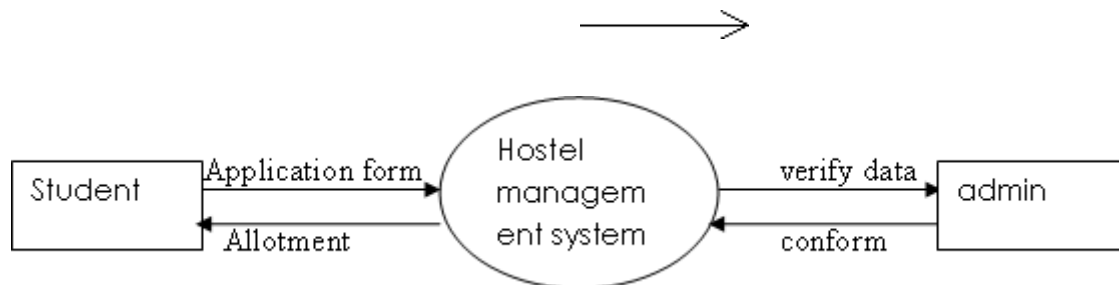
3.2.4



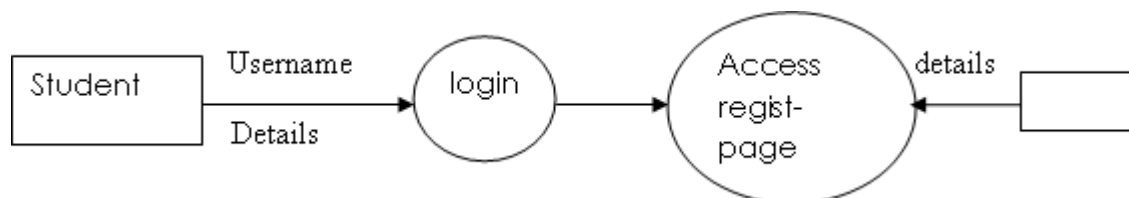
-Open rectangle, which shows a data store.

## 4.2.1 Data Flow Diagram

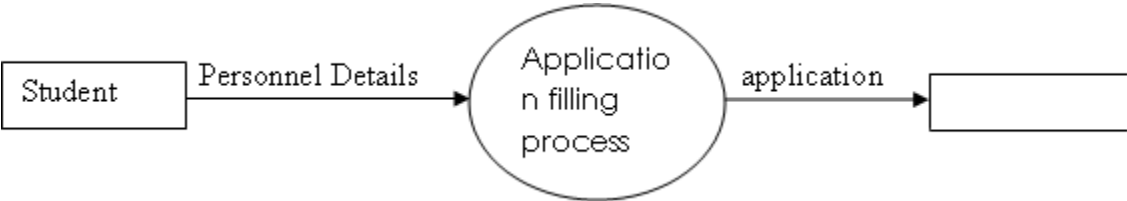
### 4.1 Context level DFD



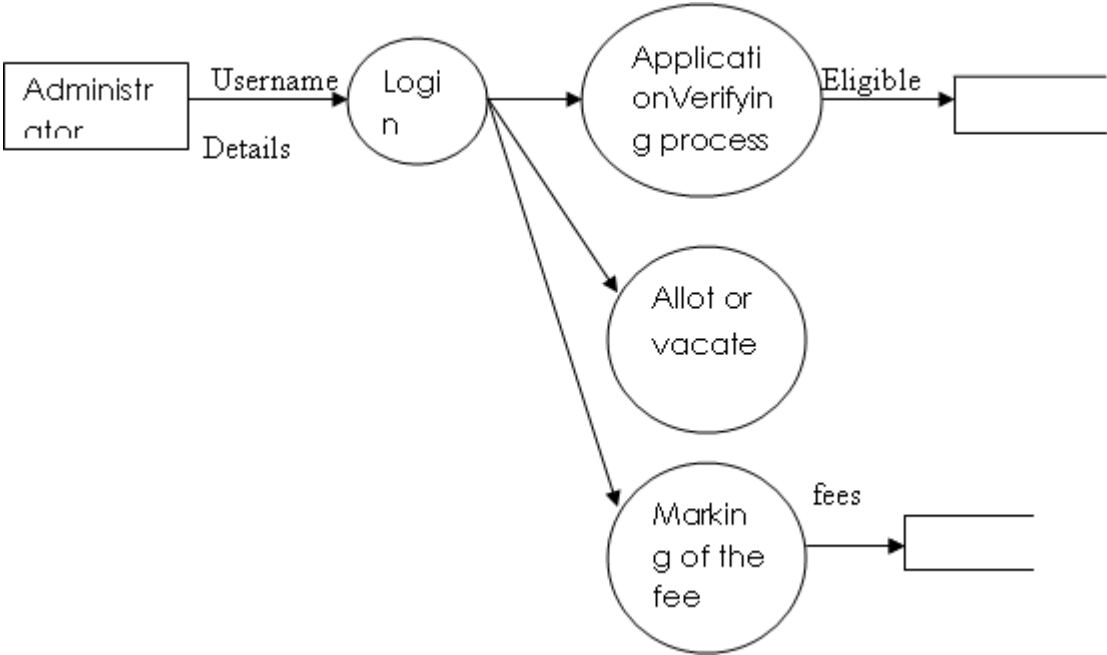
### 4.2 Student Module



### 4.3 Registration process



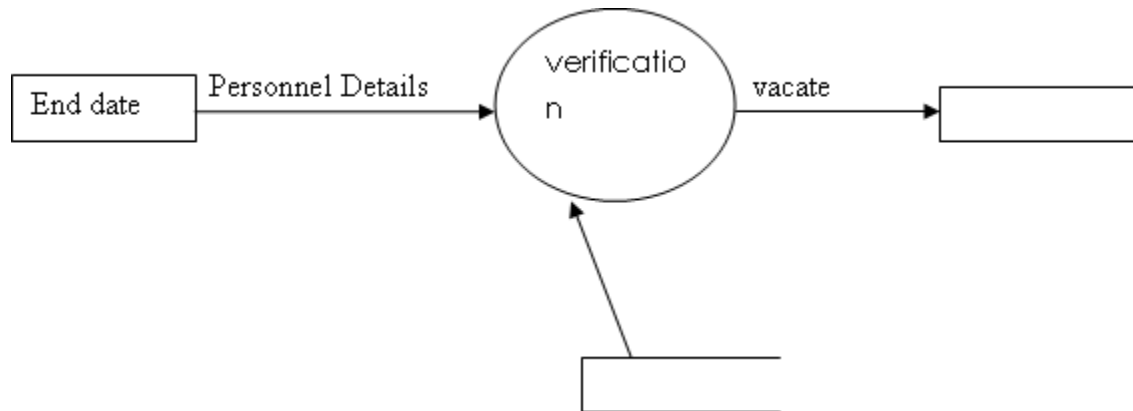
### 4.4 Admin module



### 4.5 Allotment process



## 4.6 Vacating process



## **4.3 Database Design**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

The following are the tables that are involved in the proposed system

### 1. Student account creation

| Field Name       | Data Type | Description           |
|------------------|-----------|-----------------------|
| Name             | Varchar   | Name of the student   |
| Branch           | Varchar   | Branch of the student |
| Userid           | Int       | Userid of the student |
| Password         | Varchar   | Password to use       |
| Re type password | Varchar   | Repeating it          |

### 2. Administrator Login

| Field Name | Data Type | Description             |
|------------|-----------|-------------------------|
| Username   | Int       | Username of the student |
| Pass       | Varchar   | Password of the student |

### 3. Application form

| Field Name  | Data Type | Description                  |
|-------------|-----------|------------------------------|
| Index no    | Int       | A unique no given to student |
| Name        | Varchar   | Name of the student.         |
| Age         | Int       | Age of the student           |
| dd_birth    | Date      | Date of birth of the student |
| Sex         | Varchar   | Sex of the student           |
| Reservation | Varchar   | Reservation if any           |
| Dept        | Varchar   | department                   |

|                   |          |                          |
|-------------------|----------|--------------------------|
| Course            | Varchar  | Course of study          |
| Semester          | Int      | Semester of study        |
| Course_nature     | Varchar  | Type of course           |
| Date_admit        | Date     | Date of course admission |
| Date_end          | Date     | End of course            |
| Date_host_admit   | Date     | Date of hostel admission |
| Local Address     | Longtext | Address of student       |
| Permanent Address | Longtext | Address of guardian      |
| Distance          | Int      | Distance from home       |

#### 4. Allotment

| Field Name     | Data Type | Description                  |
|----------------|-----------|------------------------------|
| Room no.       | Int       | A unique no given to student |
| Name           | Varchar   | Name of the student.         |
| Index no.      | Int       | Age of the student           |
| Date_admission | Date      | Date of birth of the student |
| Sex            | Varchar   | Sex of the student           |
| Reservation    | Varchar   | Caste of the student         |

#### 5. Vacating and editing

| Field Name     | Data Type | Description                  |
|----------------|-----------|------------------------------|
| Room no.       | Int       | A unique no given to student |
| Name           | Varchar   | Name of the student.         |
| Index no.      | int       | Age of the student           |
| Date_admission | date      | Date of birth of the student |
| Sex            | Varchar   | Sex of the student           |
| Reservation    | Varchar   | Caste of the student         |



## 6. Notice Board

| Field Name | Data Type | Description                                    |
|------------|-----------|--|
| Title      | varchar   | The title of the notice                        |
| Contents   | varchar   | It includes the different notice descriptions. |

### **4.4 Output Design**

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design an output we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output.

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the require output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports. The output must be concerned to the overall performance and the system's working, as it should. It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, ie maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system's working, as it should.

Output design consists of displaying specifications and procedures as data presentation. User never left with the confusion as to what is happening without appropriate error and acknowledges message being received. Even an unknown person can operate the system without knowing anything about the system.

## **Chapter -4**

### **RESULT ANALYSIS**

#### **4.1 Existing System**

For the past few years the number of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. 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 hostel 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 following drawbacks of the existing system.

- more human error.
- more strength and strain of manual labour needed
- Repetition of the same procedures.
- low security
- Data redundancy
- difficult to handle
- difficult to update data
- record keeping is difficult
- Backup data can be easily generated

## **CHAPTER 5**

# **CONCLUSION**

To conclude the description about the project : The project, developed using PHP and MySQL is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years the number of educational institutions are increasing rapidly.

Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. 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 hostel 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.

## References

- [1] Ayanlowo, K., Shoewu, O., Olatinwo, S.O., Omitola, O.O. and Babalola, D.D., 2014. Development of an automated hostel facility management system. *Journal of Science and Engineering*, 5(1), pp.01-10. C. Chen, Q. Wang, and Z. Guo, "Plant disease detection and classification using convolutional neural networks," in Proceedings of the International Conference on Artificial Intelligence and Industrial Engineering, pp. 138–142, ACM, 2019.
- [2] Baffoe, E.A., 2015. Hostel Management System. *Saf. Sci*, 33(3), pp.1-6. L. Zhang, Y. Zhang, and S. Song, "Faster r-cnn for plant disease detection and classification," IEEE Access, vol. 8, pp. 111051–111060, 2020.
- [3] Magar, S., Jadhav, R., Said, S. and Jadhav, S., 2021. Hostel Management System and Aggregation. *Journal of Emerging Technologies and Innovative Research*, 8(10), pp.234-238.
- [4] Radhakrishnan, R., RINSHA, P. and ROOPASREE, R., 2014. Online hostel management system. A. Smith and B. Johnson, "Deep learning-based tomato disease classification using mobilenet and densenet models," Journal of Agricultural Technology, vol. 16, no. 5, pp. 1201–1212, 2020.
- [5] Wong, P.L. and Mahdin, H., 2022. Design and Development of Online Hostel Management System to Improve Application and Management Process. *Applied Information Technology And Computer Science*, 3(2), pp.356-369. L. Zhang, Y. Zhang, and S. Song, "Faster r-cnn for plant disease detection and classification," IEEE Access, vol. 8, pp. 111051–111060, 2020. [10] H. Wang, L. Sun, and Y. Wang, "Leaf disease detection based on convolutional neural networks with mixup regularization," Computers and Electronics in Agriculture, vol. 189, p. 106334, 2021.
- [6] Jha, L. and Patil, H., 2022. A comprehensive study of and possible solutions for a hostel management system. *Computing and Communications Engineering in Real-Time Application Development*, pp.47-54.
- [7] Ma'radzi, A.A., Zakaria, M.A.A., Ishak, M.A. and Azman, A.M., 2021. RFID Hostel Management System. *Multidisciplinary Applied Research and Innovation*, 2(1), pp.330-334.
- [8] JUNE, E.A.B., 2016. *HOSTEL MANAGEMENT SYSTEM* (Doctoral dissertation).
- [9] Shoewu, O., Braimah, S.A. and Duduyemi, O., 2016. Design and Implementation of Hostel Management System (HOMASY): LASU as Case Study. *The Pacific Journal of Science and Technology*, 17(2), pp.189-196.
- [10] Batra, P., Goel, N., Sangwan, S. and Dixit, H., 2020. Design and implementation of hostel management system using Java and MySQL. *LC International Journal of STEM (ISSN: 2708-7123)*, 1(4), pp.63-74.

## APPENDIX - I

### Dreamweaver 8 & phpMyAdmin

Dreamweaver 8 includes various new features that improve usability and help you to build pages whether you're working in the design or the coding environment. First, Dreamweaver 8 provides support for best practices and industry standards, including support for advanced CSS use, XML and RSS feeds, and accessibility requirements.

| Work with best practices                                   |   |
|--|---|
| Visual authoring with XML data                             | Get up to speed with XML using powerful, visual tools to integrate feeds into work and remove the mystery from XML to HTML translation. Integrate XML-based data, such as RSS feeds, into web pages using a simple drag-and-drop workflow. Jump to Code view to customize the transformation, using improved code hinting for XML and XSLT. For more information, see <a href="#">Displaying XML Data in Web Pages</a> .  |
| New, unified CSS panel                                     | The new, unified CSS panel provides a central location for learning, understanding, and working with the CSS styles applied to pages in a visual way. All the CSS functionality is consolidated into one panel set and enhanced to make working with CSS styles easier and more productive. The new interface makes it easier to see the cascade of styles applied to a specific element so that you can easily identify where attributes are defined. A property grid allows for quick edits. For more information, see <a href="#">Using the CSS Styles panel</a> , |
| CSS layout visualization                                   | Apply visual aides at design time to outline CSS layout borders or color CSS layouts. Applying visual aides reveals complex nesting schemes and improves selection. Click the CSS layout for valuable tooltips that help you understand the elements that are controlling the design. See <a href="#">Laying Out Pages with CSS</a> .   |
| Style Rendering toolbar                                    | View content the same way users will see it no matter what the delivery mechanism with new support for CSS media types. Use the Style Rendering toolbar to toggle to Design view and see how it will look in print, on a handheld, or onscreen. See <a href="#">The Style Rendering toolbar</a> .   |
| CSS rendering improvements                                 | Match how complex CSS layouts will render in most browsers with substantial improvements in Design view accuracy. Dreamweaver now fully supports advanced CSS techniques, such as overflow, pseudo-elements, and form elements.   |
| Accessibility: Support for WCAG/W3C priority 2 checkpoints | In addition to the integrated accessibility evaluation tool for Section 508 and WCAG Priority 1 checkpoints, Dreamweaver now supports both CSS and accessibility with an updated evaluation tool that includes WCAG Priority 2 checkpoints.   |
| Improved WebDAV  | WebDAV in Dreamweaver 8 now supports digest authentication and SSL for secure file transfer, and offers improved connectivity with a wider array of servers. See <a href="#">Using WebDAV</a> to check in and check out files.  |

Get more done in less time with optimized user workflows that reduce the time required to complete common tasks. Dreamweaver 8 takes the hassle out of the little things so you can spend more time designing and developing engaging websites and applications.

| Get more done            |      |   |
|--------------------------|------|---|
| Background transfer      | file | Keep working while Dreamweaver 8 uploads files to the server. For more information, see <a href="#">Managing file transfers</a> .   |
| Zoom                     |      | Get greater control over your design with zoom. Zoom in and inspect an image or work with a complex nested table layout. Zoom out to preview how a page will look. For more information, see <a href="#">Zooming in and out</a> .                                     |
| Guides                   |      | Compare the page layout to page mockups with pixel-perfect accuracy using guides to measure page layouts. Visual feedback helps measure distances accurately and supports intelligent snapping. For more information, see <a href="#">Using guides</a> .              |
| Coding toolbar           |      | The new Coding toolbar provides buttons for common coding features in a gutter bar along the side of Code view. For more information, see <a href="#">Inserting code quickly with the Coding toolbar</a> .  |
| Code collapse            |      | Focus only on the code you want to see by hiding and expanding blocks of code. For more information, see <a href="#">Collapsing and expanding code fragments</a> .  |
| Workspace layouts        |      | Customize and save workspace configurations. Dreamweaver 8 ships with four different configurations tailored to the needs of designers and coders. You can also build a custom workspace. For more information, see <a href="#">Saving custom workspace layouts</a> . |
| Site-relative references |      | Work seamlessly with server-side includes at design time and runtime by ensuring that references are relative to sites instead of local files. For more information, see <a href="#">Setting the relative path of new links</a> .                                     |
| Code-editing             |      | Gain greater control over how Dreamweaver provides code hints   |

|  |  |
|--|--|
| Tabbed documents for the Mac                                     | New document tabs on the Mac help simplify the user interface and make it easier to select documents. For more information, see <a href="#">Displaying tabbed documents (Macintosh)</a> .  |
| New starter pages  | New layouts and designs let you to create sites quickly.   |
| Improved site synchronize and check-in/check-out                 | Manage sites with increased reliability and confidence. Improved site synchronization features help ensure that the file in use is the latest version. Prevent accidental overwriting of others' work with improved check-in/check-out functionality. For more information, see <a href="#">Synchronizing the files on your local and remote sites</a> .   |
| Compare files  | Quickly compare files to identify what has changed. You can compare two local files, a file on the local computer and one on a remote computer, or two files on the remote computer. Use your favorite file comparison tool with Dreamweaver on both the Macintosh and Windows platforms. For more information, see <a href="#">Comparing files for differences</a> .  |
| Paste Special  | With the new pasting options in Dreamweaver, you can retain all the source formatting created in Microsoft Word, or just paste the text. For more information, see <a href="#">Adding text to a document</a> .   |
| <b>Integrates with the latest technologies and standards</b>     |  |
| Support for ColdFusion MX 7                                      | Updated support for ColdFusion MX 7 includes new server behaviors and code hinting. To match the code hinting and debugging with the correct version of ColdFusion, Dreamweaver automatically detects the server version the first time it connects to the site. The tight integration between Dreamweaver and ColdFusion lets you add and remove databases directly from the Databases panel, and view only ColdFusion components defined in the current site. For more information, see <a href="#">Enabling the ColdFusion enhancements</a> . |
| Support for PHP 5  | Take advantage of updated support for PHP 5, including server behaviors and code hinting.  |
| Flash Video  | Quickly and easily insert a Flash Video file in a web page. For more information, see <a href="#">Inserting Flash Video content</a> .  |
| Macromedia Web Publishing System: notification and event logging | Keep track of everything that is going on within your site. Events in Dreamweaver notify the Macromedia Web Publishing System server so that all changes to a website in the WPS system are recorded.  |
| Updated reference material from O'Reilly                         | Consult new reference content for XML, XSLT, and XPath, and updated content for ASP and JSP.   |

Dreamweaver 8 supports efforts to learn and take advantage of new technologies, including PHP 5, Flash Video, ColdFusion MX 7, and the Macromedia Web Publishing System.

## **Dreamweaver and accessibility**

Accessibility refers to making websites and web products usable for people with visual, auditory, motor, and other disabilities. Examples of accessibility features for software products and websites include screen reader support, text equivalents for graphics, keyboard shortcuts, change of display colors to high contrast, and so on. Dreamweaver provides tools that make the product accessible and tools that help you author accessible content:

**Using Dreamweaver accessibility features** For Dreamweaver web designers who need to use accessibility features, Dreamweaver offers screen reader support, keyboard navigation, and operating system accessibility support. For more information, see [Using Dreamweaver accessibility features](#).

**Authoring for accessibility** For Dreamweaver web designers who need to create accessible content, Dreamweaver assists you in creating accessible pages that contain useful content for screen readers and comply with government guidelines.

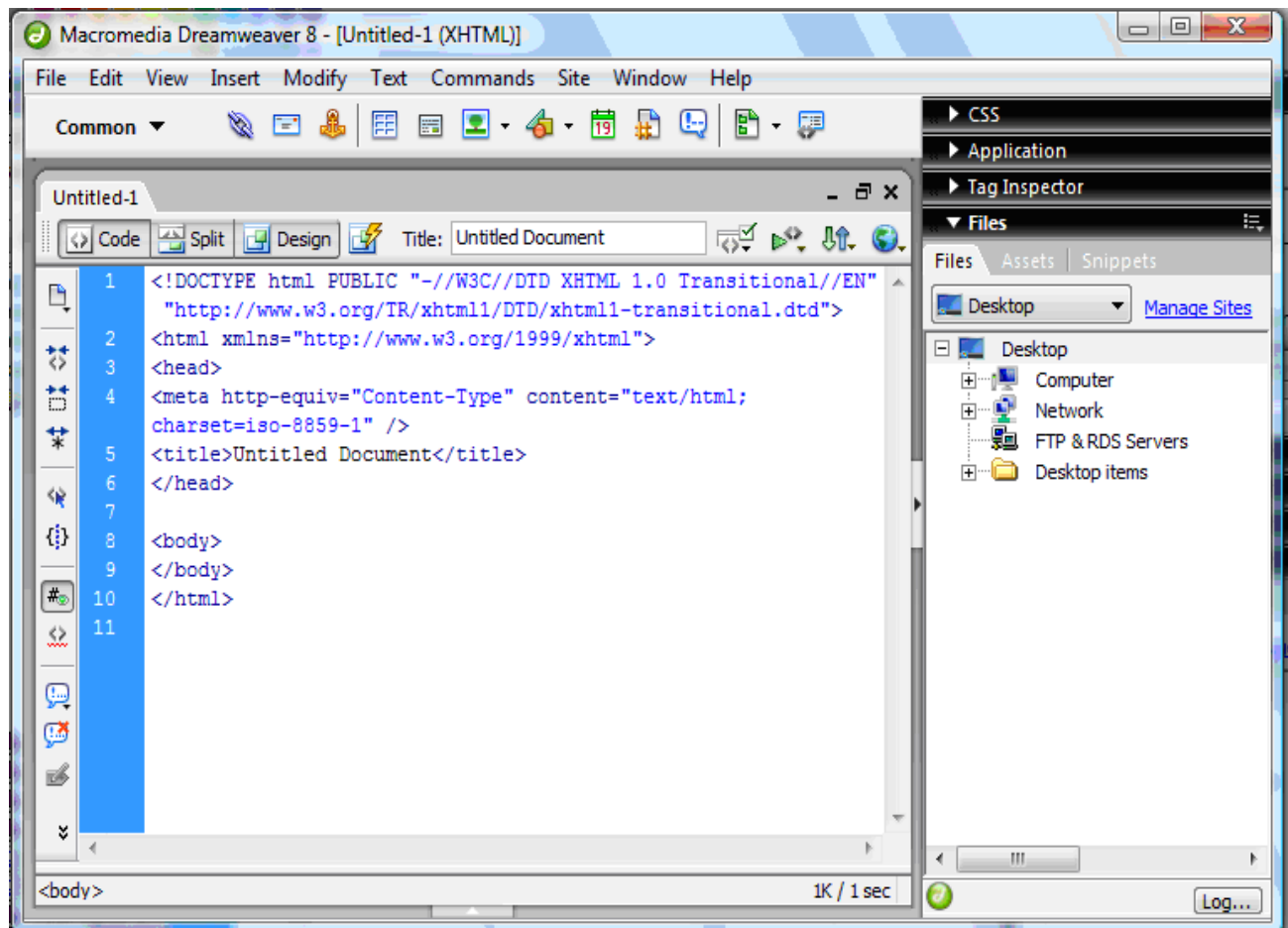
Dreamweaver provides dialog boxes that prompt you to enter accessibility attributes when you insert page elements (see [Optimizing the workspace for accessible page design](#)). For example, the accessibility dialog box for images reminds you to add text equivalents for graphics. Then, when the image appears on a page for a user with visual disabilities, the screen reader reads the description.

Dreamweaver also provides sample web pages that were designed for accessibility (see [Creating a document based on a Dreamweaver design file](#)) and an accessibility report that you can run to test your page or site against the Section 508 accessibility guidelines.

## **The workspace layout**

In Windows, Dreamweaver provides an all-in-one-window integrated layout. In the integrated workspace, all windows and panels are integrated into a single larger application window.





The Windows workspace also has a Coder option, which docks the panel groups on the left side and displays the Document window in Code view by default. For more information, see *Using the coder-oriented workspace (Windows only)*. To use this option, see *Choosing the workspace layout (Windows only)*.

On the Macintosh, Dreamweaver can display multiple documents in a single window with tabs that identify each document. Dreamweaver can also display a floating workspace in which each document appears in its own individual window. Panel groups are initially docked together, but can be undocked into their own windows. Windows “snap” automatically to each other, to the sides of the screen, and to the Document window as you drag or resize them.

## Dreamweaver workspace elements

This section briefly describes some elements of the Dreamweaver workspace.

Dreamweaver provides many other panels, inspectors, and windows. To open Dreamweaver panels, inspectors, and windows, use the Window menu. If you can't find a panel, inspector, or window that's marked as open, select Window > Arrange Panels to neatly lay out all open panels.

The Windows workspace also has a Coder option, which docks the panel groups on the left side and displays the Document window in Code view by default. For more information, see [Using the coder-oriented workspace \(Windows only\)](#). To use this option, see [Choosing the workspace layout \(Windows only\)](#).

**The Start page** enables you to open a recent document or create a new document. From the Start page you can also learn more about Dreamweaver by taking a product tour or a tutorial.

**The Insert bar** contains buttons for inserting various types of “objects,” such as images, tables, and layers, into a document. Each object is a piece of HTML code that enables you to set various attributes as you insert it. For example, you can insert a table by clicking the Table button in the Insert bar. If you prefer, you can insert objects using the Insert menu instead of the Insert bar.

**The Document toolbar** contains buttons that provide options for different views of the Document window (such as Design view and Code view), various viewing options, and some common operations such as previewing in a browser.

**The Standard toolbar** (not displayed in the default workspace layout) contains buttons for common operations from the File and Edit menus: New, Open, Save, Save All, Cut, Copy, Paste, Undo, and Redo. To display the Standard toolbar, select View > Toolbars > Standard.

**The Coding toolbar** (displayed in Code view only) contains buttons that let you perform many standard coding operations.

**The Style Rendering toolbar** (hidden by default) contains buttons that let you see how your design would look in different media types if you used media-dependent style sheets. It also contains a button that lets you enable or disable CSS styles. **The Document window** displays the current document as you create and edit it.

**The Property inspector** lets you view and change a variety of properties for the selected object or text. Each kind of object has different properties. The Property inspector is not expanded by default in the Coder workspace layout. **The tag selector** in the status bar at the bottom of the Document window shows the hierarchy of tags surrounding the current selection. Click any tag in the hierarchy to select that tag and all its contents. **Panel groups** are sets of related panels grouped together under one heading. To expand a panel group, click the expander arrow at the left of the group's name; to undock a panel group, drag the gripper at the left edge of the group's title bar. **The Files panel** enables you to manage your files and folders, whether they are part of a Dreamweaver site or on a remote server. The Files panel also enables you to access all the files on your local disk, much like Windows Explorer (Windows) or the Finder (Macintosh).

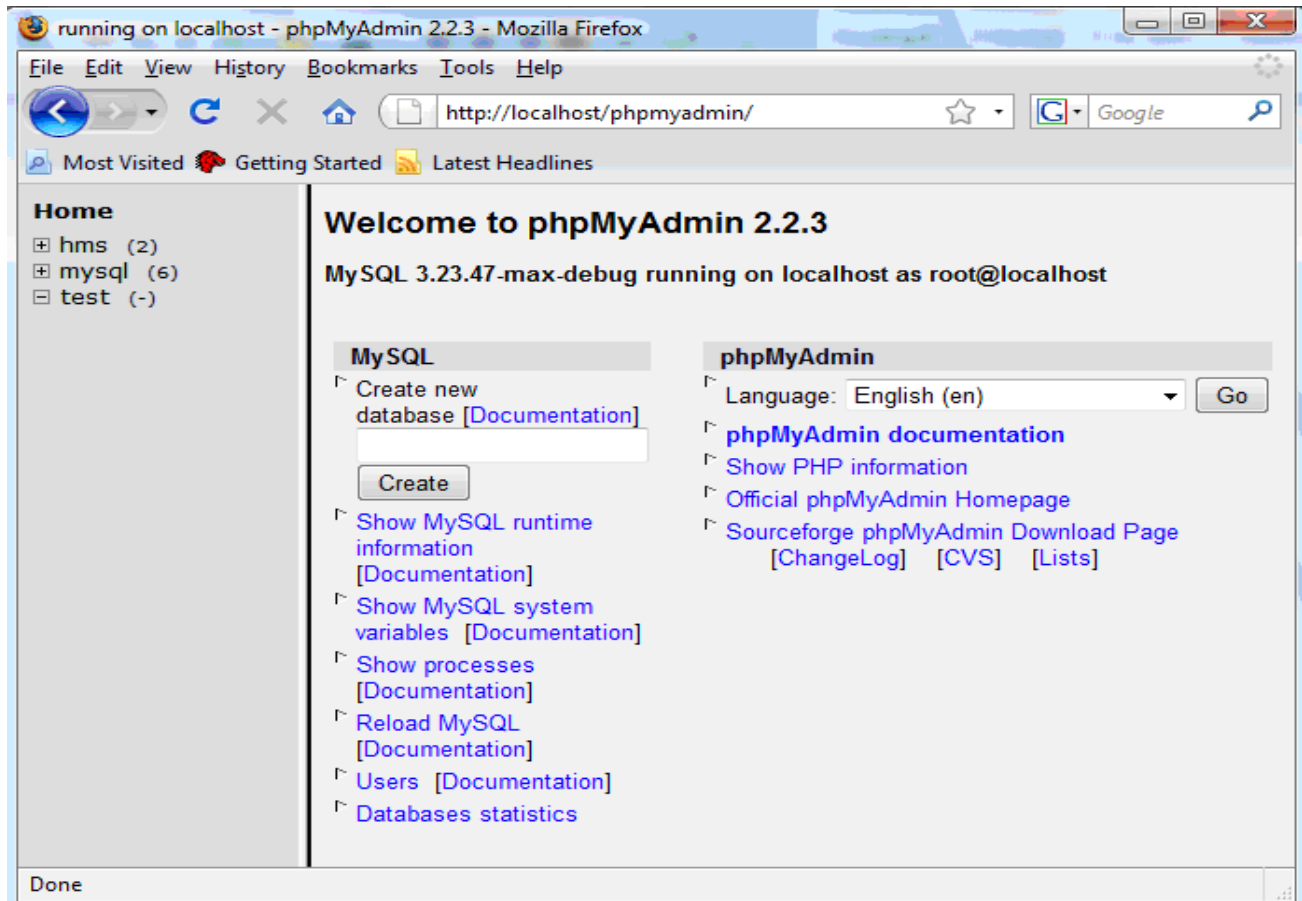
## Laying Out Pages with CSS

In Macromedia Dreamweaver 8, you can use CSS styles to lay out your page. You can either insert div tags manually and apply CSS positioning styles to them, or you can use Dreamweaver layers to create your layout. A layer in Dreamweaver is an HTML page element—specifically, a div tag, or any other tag—that has an absolute position assigned to it. Dreamweaver also has a tracing image feature, which you can use to re-create a page design that was created in a graphics application.

### Client-side role of forms

Forms support the client side of the client-server relationship. When a visitor enters information into a form displayed in a web browser (the client) and clicks the submit button, the information is sent to the server where a server-side script or application processes it. Common server-side technologies used for processing form data include Macromedia ColdFusion, Microsoft Active Server Pages (ASP), and PHP.

## phpMyAdmin



### Databases statistics

**Host :**

**localhost**

**Generation Time :**

**Mar 19, 2009 at 09:45 PM**

|                          | Database ▾ | Table(s) | Data      | Indexes | Total   |
|--------------------------|------------|----------|-----------|---------|---------|
| <input type="checkbox"/> | hms        | 2        | 85 Bytes  | 2.0 KB  | 2.1 KB  |
| <input type="checkbox"/> | mysql      | 6        | 579 Bytes | 9.0 KB  | 9.6 KB  |
| <input type="checkbox"/> | test       | 0        | 0 Bytes   | 0 Bytes | 0 Bytes |
|                          | Sum: 3     | 8        | 664 Bytes | 11.0 KB | 11.6 KB |



With selected:

Drop

hms.application running on localhost - phpMyAdmin 2.2.3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmyadmin/

Most Visited Getting Started Latest Headlines

hms.application running... x mysql running on localho... x mysql running on localho... x mysql running on localho... x phpMyAdmin

### Database *hms* - table *application running on localhost*

| Field | Type    | Length/Values* | Attributes | Null     | Default | Extra | Primary                  | Index                    | Unique                   | Fulltext                 |
|-------|---------|----------------|------------|----------|---------|-------|--------------------------|--------------------------|--------------------------|--------------------------|
|       | TINYINT |                |            | not null |         |       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|       | TINYINT |                |            | not null |         |       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|       | TINYINT |                |            | not null |         |       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|       | TINYINT |                |            | not null |         |       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|       | TINYINT |                |            | not null |         |       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Table comments :

Table type :

\* If field type is "enum" or "set", please enter the values using this format: 'a','b','c'...  
If you ever need to put a backslash ("\") or a single quote (') amongst those values, backslashes it (for example '\\xyz' or 'a\\b').

[\[Documentation\]](#)

## Transformations

To enable transformations, you have to setup the `column_info` table and the proper directives. Please see the Configuration section on how to do so. You can apply different transformations to the contents of each field. The transformation will take the content of each field and transform it with certain rules defined in the selected transformation. Say you have a field 'filename' which contains a filename. Normally you would see in phpMyAdmin only this filename. Using transformations you can transform that filename into a HTML link, so you can click inside of the phpMyAdmin structure on the field's link and will see the file displayed in a new browser window. Using transformation options you can also specify strings to append/prepend to a string or the format you want the output stored in. For a general overview of all available transformations and their options, you can consult your `<www.your-host.com>/<your-install-dir>/transformation_overview.php` installation. For a tutorial on how to effectively use transformations, see our Link section on the official phpMyAdmin homepage.

## Usage

Go to your *tbl\_structure.php* page (i.e. reached through clicking on the 'Structure' link for a table). There click on "Change" (or change icon) and there you will see three new fields at the end of the line. They are called 'MIME-type', 'Browser transformation' and 'Transformation options'.

- The field 'MIME-type' is a drop-down field. Select the MIME-type that corresponds to the column's contents. Please note that transformations are inactive as long as no MIME-type is selected.
- The field 'Browser transformation' is a drop-down field. You can choose from a hopefully growing amount of pre-defined transformations. See below for information on how to build your own transformation. There are global transformations and mimetype-bound transformations. Global transformations can be used for any mimetype. They will take the mimetype, if necessary, into regard. Mimetype-bound transformations usually only operate on a certain mimetype. There are transformations which operate on the main mimetype (like 'image'), which will most likely take the subtype into regard, and those who only operate on a specific subtype (like 'image/jpeg'). You can use transformations on mimetypes for which the function was not defined for. There is no security check for you selected the right transformation, so take care of what the output will be like.
- The field 'Transformation options' is a free-type textfield. You have to enter transform-function specific options here. Usually the transforms can operate with default options, but it is generally a good idea to look up the overview to see which options are necessary. Much like the ENUM/SET-Fields, you have to split up several options using the format 'a','b','c',...(NOTE THE MISSING BLANKS). This is because internally the options will be parsed as an array, leaving the first value the first element in the array, and so forth. If you want to specify a MIME character set you can define it in the *transformation\_options*.

- . You have to put that outside of the pre-defined options of the specific mime-transform, as the last value of the set. Use the format “; charset=XXX”. If you use a transform, for which you can specify 2 options and you want to append a character set, enter “‘first parameter’,’second parameter’,’charset=us-ascii””. You can, however use the defaults for the parameters: “‘’,’,’charset=us-ascii””.

## **File structure**

All mimetypes and their transformations are defined through single files in the directory ‘libraries/transformations/’. They are stored in files to ease up customization and easy adding of new transformations.

Because the user cannot enter own mimetypes, it is kept sure that transformations always work. It makes no sense to apply a transformation to a mimetype, the transform-function doesn’t know to handle.

One can, however, use empty mime-types and global transformations which should work for many mimetypes. You can also use transforms on a different mimetype they where built for, but pay attention to option usage as well as what the transformation does to your field. There is a basic file called ‘*global.inc.php*’. This function can be included by any other transform function and provides some basic functions.

There are 5 possible file names:

1.A mimetype+subtype transform:[mimetype]\_[subtype]\_[transform].inc.php. Please note that mimetype and subtype are separated via ‘\_’, which shall not be contained in their names. The transform function filename may contain only characters which cause no problems in the file system as well as the PHP function naming convention. The transform function will be called PMA\_transform\_[mimetype]\_[subtype]\_[transform]()’.

## **APPENDIX - II**

### **My\_SQL Functions**

`mysql_field_len` — Returns the length of the specified field

`mysql_field_name` — Get the name of the specified field in a result

`mysql_field_seek` — Set result pointer to a specified field offset

`mysql_field_table` — Get name of the table the specified field is in

`mysql_field_type` — Get the type of the specified field in a result

`mysql_free_result` — Free result memory

`mysql_get_client_info` — Get MySQL client info

`mysql_get_host_info` — Get MySQL host info

`mysql_get_proto_info` — Get MySQL protocol info

`mysql_get_server_info` — Get MySQL server info

`mysql_info` — Get information about the most recent query

`mysql_insert_id` — Get the ID generated from the previous INSERT operation

`mysql_list_dbs` — List databases available on a MySQL server

`mysql_list_fields` — List MySQL table fields

`mysql_list_processes` — List MySQL processes

`mysql_list_tables` — List tables in a MySQL database

`mysql_num_fields` — Get number of fields in result

`wsmysql_num_rows` — Get number of rows in result

`mysql_pconnect` — Open a persistent connection to a MySQL server



mysql\_fetch\_row — Get a result row as an enumerated array

mysql\_field\_flags — Get the flags associated with the specified field in a result

mysql\_field\_len — Returns the length of the specified field

mysql\_field\_name — Get the name of the specified field in a result

mysql\_field\_seek — Set result pointer to a specified field offset

mysql\_field\_table — Get name of the table the specified field is in

mysql\_field\_type — Get the type of the specified field in a result

mysql\_free\_result — Free result memory

mysql\_get\_client\_info — Get MySQL client info

mysql\_get\_host\_info — Get MySQL host info

mysql\_get\_proto\_info — Get MySQL protocol info

mysql\_get\_server\_info — Get MySQL server info

mysql\_info — Get information about the most recent query

mysql\_insert\_id — Get the ID generated from the previous INSERT operation

mysql\_list\_dbs — List databases available on a MySQL server

mysql\_list\_fields — List MySQL table fields

mysql\_list\_processes — List MySQL processes

mysql\_list\_tables — List tables in a MySQL database

mysql\_num\_fields — Get number of fields in result



User Database

Home

hostel (17)

adm\_account

aiswarya

anaswara

athulya

news

obc girls

postmetric(boys)

postmetric(girls)

reg\_dbm

reg\_det

sanatana

sarovar

siberia

std\_account

std\_login

stud\_host

user

mysql (6)

test (-)

Database *hostel* - table *user* running on localhost

Showing records 0 - 4 (4 total)

SQL-query : [\[Edit\]](#)  
SELECT \* FROM `user` LIMIT 0, 30

Show : 30

 rows starting from 

0

in 

horizontal

 mode and repeat headers after 

100

 cells

|                      |                        | id | user   | pass   |
|----------------------|------------------------|----|--------|--------|
| <a href="#">Edit</a> | <a href="#">Delete</a> | 1  | hostel | hostel |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 2  | hostel | hostel |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 3  | admins | admins |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 4  | admins | admins |

Student List

|                      |                        | id  | room_no | index_no | name | dept | yr |
|----------------------|------------------------|-----|---------|----------|------|------|----|
| <a href="#">Edit</a> | <a href="#">Delete</a> | 101 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 102 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 103 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 104 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 105 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 106 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 107 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 108 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 109 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 110 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 111 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 112 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 113 | 0       | 0        |      |      | 0  |
| <a href="#">Edit</a> | <a href="#">Delete</a> | 115 | 0       | 0        |      |      | 0  |

