**Chapter 5**

**IMPLEMENTATION**

**5.1 Selection of the platform**

The platform selected for the project is windows which supports LAMP Server 1.6.5 as it is compatible with Php editor in the front end, MYSQL in the back end and is supported by the Microsoft SQL Server.

**5.2 Selection of the programming language**

Web applications are an extension of a web server (Armstrong et al, 2004). Web applications are either service oriented or presentation oriented. A presentation oriented web application produces interactive web pages containing mark up languages like (XML and HTML), server side scripting has been done in PHP and the validation has been implemented in JavaScript.

**5.2.1 Front End: HTML WITH PHP (PHP editor)**

The programming language used for the development work is Visual HTML (with PHP). The reason for selection of this language includes among many others the following few.

* The object oriented structure of PHP programming language is very simple, and HTML has vast capabilities of a good and featured display.
* The libraries provided by the HTML contain many standard functions which make the life of developer easy.
* **Php editor** is an integrated, Interactive Development Environment ("IDE") which provides an easy development environment for Php and HTML applications and tools for automated testing and optimization.
* The **Php editor** has been highly optimized to support rapid application development. It is particularly easy to develop graphical user interfaces and to connect them to handler functions provided by the application.
* The graphical user interface of the **HTML** provides intuitively appealing views for the management of the program structure in the large and the various types of entities (classes, modules, procedures, forms…).
* **Php** provides a comprehensive interactive and context-sensitive online help system.

**5.2.2 BACK END: MYSQL(LAMP server)**

**MySQL** is a relational data base management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

* MySQL is an open source tool.
* MySQL is a popular choice of database for use in web applications, and is a central component of the widely-used LAMP web application software stack — LAMP is an acronym for "Linux, [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [PHP](http://en.wikipedia.org/wiki/PHP)".
* MySQL is primarily an [RDBMS](http://en.wikipedia.org/wiki/RDBMS) and therefore ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within.
* LAMPserver provides an ease in creating tables by a graphical as well as query based interface.

MySQL implements the following features, which some other [RDBMS](http://en.wikipedia.org/wiki/RDBMS) systems may not:

* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
* Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

**5.2.3 Server: LAMP server**

* LAMP Server features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interface(GUIs).
* It is auto configured with Php editor,so there is a ease of accessing database and creating reports and views.
* The application is available for a wide variety of operating systems, including Unix,GNU,FreeBSD,Linux,Solaris, Novell Netware, Mac OS X,Microsoft Windows,OS/2,TPF, and eComStation.
* These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl,Python,Tcl, and PHP. Popular authentication modules include mod\_access, mod\_auth, mod\_digest, and mod\_auth\_digest, the successor to mod\_digest.

**5.3 Programming Coding Guidelines**

### 5.3.1 HTML With PHP Coding Guidelines

### 5.3.1.1 Naming Conventions

* The user does not have to change the name of objects created by the designer tools to make them fit the guidelines.
* Each table and attribute is full name and self explanatory.

#### 5.3.1.2 Procedures Usage for Database access

* Use procedures rather than direct queries making error free and small code in front end
* Procedures name starts from ‘prc’ followed by input or update type ‘in’ or ‘up’ followed by table name for eg prcin\_con,prcup\_reg…

#### 5.3.1.3 Dynamic Pages

* The Use of dynamic pages makes it easy in a server round trip hence facilitating features like checking if a username exists or not before creating a new user.
* For User searches the data bound with the views depends on the keyword in the search textbox hence accomplished by Dynamic pages
* Display of password in password recovery page without using any other page, making buttons visible or hidden depending on the end user choice comes along with dynamic pages.

#### 5.3.1.4 Button click coding

* As soon as the button is clicked a set of code are executed before forwarding the user to the next page.
* For eg;- in saving the information of a form a database round trip occurs in the same page before continuing to next page.

**5.3.2 CODE LAYOUT**

**5.3.2.1 Including Braces**

Braces should **always** be included when writing code using if, for, while etc. blocks. There are **no exceptions** to this rule, even if the braces could be omitted. Leaving out braces makes code harder to maintain in the future and can also cause bugs that are very difficult to track down.

### 5.3.2.2 Where to put the braces

Braces should always be placed on a line on their own; again there are no exceptions to this rule. Braces should also align properly (use tabs to achieve this) so a closing brace is always in the same column as the corresponding opening brace.

**5.3.2.3 Spaces between Tokens**

There should always be one space on either side of a token in expressions, statements etc. The only exceptions are commas (which should have one space after, but none before), semi-colons (which should not have spaces on either side if they are at the end of a line, and one space after otherwise). Functions should follow the rules laid out already, i.e. no spaces between the function name and the opening bracket and no space between the brackets and the arguments, but one space between each argument.