# Chapter 6

# Implementation

## 6.1 Introduction

Implementation of a software project should be done in methodological way applying applicable technical methods with proper guidance. In this regards technology adopted, developer approach on the project, analysis of the system discussed in the earlier chapters are very important. This chapter describes how I have implemented this software project.

## 6.2 Hardware and Software Used

This proposed Activity management and Monitoring System was designed and developed using Web technology for the purpose of storing, manipulating and summarizing the records related process of product certification scheme. The implementation of the project was done Apache, MySQL and PHP (WAMP) technologies.

### 6.2.1 Software Installation and Configuration



Installation and configuration of several open source software were done as follows:

#### 6.2.1.1 Apache and PHP Installation

The WAMP stack provides the four key elements namely an operating system, web server, database and web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS); Apache is the Web server, while being PHP dynamic scripting language.

Step 1 – to start apache server 'start all service' command was executed in command prompt.

Step 2 – Launched browser window and enter this URL http://localhost/ and see the result page and found that installation was success.

All the development files like html, php, css, javascripts are saved in a folder called 'AMMSdev' in the 'htdocs' folder of apache installation path.

#### 6.2.1.2 MySQL

MySQL version 5.0.51a was installed as the database management system software. To data definition and data manipulation purpose MySQL Administrator and Query Browser tool version 1.2.12 software was used to enhance the development progress and to meet dead line requirement of this project.

#### 6.2.1.3 Microsoft Dream Weaver 8.0

For Graphical User Interface (GUI) Designing Macromedia Dream Weaver 8.0 was used. All the web interfaces were designed using Hyper Text Markup Language (HTML).

#### 6.2.1.4 Java Script Language

This is used as the client side scripting language. This is heavily used for the data entry validation purposes.

### 6.2.2 Hardware Used

Personal computer with Pentium-iv 1.6 GHz processor was used to install the software and develop the project.



## 6.3 Testing of Database Connectivity

Each time when there is need of connecting to the database, a connection must be established in PHP script via three controlled classes. These classes are included to the PHP script using *"Include"* statement and this include statement includes and evaluates the file.

1 **'config.php'** class is used to initialize connection variables

```
<?php
//connetion variables
$dbhost='localhost'; $dbuser='root'; $dbpass=''; $dbname='AMMSdev';
?>
```

2. **'opendb.php'** class is used to connect to the data base using variables defined in config.php

```
<?php

$conn=mysql_connect($dbhost,$dbuser,$dbpass) or die('I can not
connect to the database because:'.mysql_error()); mysql_select_db
($dbname);

?>
```

3. **'closedb.php'** is used to close the database connection once it is not required.

```
<?php

//mysql_free_result()

mysql_close($conn);

?>
```

Given below is the way that these classes are included in to php files.

Include 'library/config.php'; include 'library/opendb.php';



When there is no need of having database connectivity, data based is closed using

include 'library/closedb.php';

## 6.4 Implementation of Security Measures

### 6.4.1 Password Encryption

To protect form passwords being accessed by authorized people, all the pass words are encrypted. When creating a new user, the password provided will be encrypted and stored in "user account" table. When user enters the password to log in to the system, the login page encrypts the password and compares with the encrypted password in "user" table and if both passwords are matching then grant access to the system.

### 6.4.2 Sessions

When a user is granted access privileges to the system, a session variable is created.

```
session_start();
  $_SESSION['au_userid'] = $row['user_id'] ;
```

The user will be checked in every page and if not successful the user will be re-directed to the log-in page. Session variable 'au_userid' is introduced in the 'login_process.php'. In every page, this session variable is checked as following.

```
<?php
session_start();
if (!isset($_SESSION['au_userid'])){
  print "<meta http-equiv=\"refresh\" content=\"0;URL=login.php\">";
exit;
}?>
```

### 6.4.3 Password Protection and System Login

A user has to log-in to the system at the first place before use any system facility. All the users of the system are provided with a Username and a Password created by System Administrator. All the Usernames and the Passwords created by the system Administrator are stored in a separate database table called "user account" and password is encrypted. To encrypt the password md5 () php function is used. This ensures protection of reading of password.



When a user submits Username and the Password, the system establishes a connection with the database and checks the validity of the entered Username and Password. If both Username and Password are correct then the system grants the permission to access the home page.

### 6.4.4 System User Management

Only system administrator has been granted the right privileges to manage user account. System administrator can define categories of users. When creating a user account, administrator should specify the user category. All the users which belong to a particular user category have the same access privileges. The System maintains a list

of system facilities which reflects the systems units designed in the system. System administrator can assign any system facility to user category by using 'assign facilities' unit.

When user is logged into the system the tree navigation structure is initialized according to the system facilities granted to the user category. 'Treeprocess.php' is used to initialize tree navigation structure. If a particular unit is not allowed to use then hyperlink facility is disabled to the menu item. Please refer to Appendix E – Important Codes Used for Implementation

### 6.5 Implementation of the System

There are 5 sub systems and 4 modules as per Architectural Design.

#### 6.5.1 Sub Systems

Product certification activities, auditors' profile, project officers' profile, system administration profile, administration profiles are the subsystem in this projects. Administration profile is designed linking some tables such as; employees, epm_section, division and section tables from another software project design for administration activities in Sri Lanka Standards Institution. Implementation of system administration process has discussed already above this chapter. Project officer profile and auditors profile have been designed to add, edit and delete relevant details.



Product certification activities have divided to modules as 'SLS' mark holders' handling, application handling, audit handling and annual fee handling. In addition to these modules report generation of relevant important fields have been introduced in implementation.

### 6.6 Summery

Software installation, database table preparation and coding using PHP, java script, HTML, have been described in this chapter. As the examples, some repetitive main code has been given here and security measures have been discussed at the last section of this chapter.