PHP & MySQL Simple Project Management Application

Vedat Shaqiri

Follow this and additional works at: https://knowledgecenter.ubt-uni.net/etd

Part of the Computer Sciences Commons
PHP & MySQL Simple Project Management Application

Bachelor Degree

Vedat Shaqiri

September/2010
Prishtinë
Student: Vedat Shaqiri

**PHP & MySQL Simple Project Management Application**

Supervisor: Univ. Prof. Dr. Werner Retschitzegger

September/2010

This thesis is submitted in partial fulfillment of the requirements for a Bachelor Degree
Acknowledgements

First and foremost I would like to thank the Almighty God for giving me the opportunity to study and come to the situation of writing this Thesis, for which I also try to be thankful.

For the last three months I have periodically been working in my application. Difficulties and challenges appeared through this process but the end stage proved satisfaction for what I had been doing during this period of time. It was more or less a new experience developing such an application using those programming languages. From the start to the end I challenged myself to learn something new, which I hope to have achieved.

I owe my deepest gratitude to my supervisor, Univ. Prof. Dr. Werner Retschitzegger, whose encouragement, guidance and support from the initial to the final level enabled me write this Bachelor Thesis.

Lastly, I offer my respect and regards to all those who supported me in any way during the completion of the Thesis writing.
Abstract

Starting building an application is not that easy at all, especially when people are not experienced enough with the tools needed to cook it. This thesis contains a brief introduction to an PHP & MySQL application. Just as said some lines before, when you get into a thing that you might not be familiar with, difficulties appear, at this time patience and dedication come in to play realizing and making it become true.

PHP and MySQL offer a great opportunity building dynamic web sites and application. It is not PHP that makes a web site or an application to become dynamic, it is the MySQL database that contributed for the Simple Project Management to be dynamic. Nevertheless, PHP and MySQL know very well how to communicate with each other, in order things to go the right direction.

Next chapters of this thesis try to explain how a Structured Query Language (SQL) database looks like, when it communicates with an Graphical User Interface (GUI) for the data to managed. The application is called ‘Simple Project Management’, which intention is to help companies manage customers, in an easier way. It integrates some functional features for handling information about customers and projects in general.

Adobe Dreamweaver software was used to write the whole code of the Simple Project Management Application. I have been also using the ‘phpMyAdmin’ open source software, which is written in PHP, and helped me administering the MySQL database during the development of this application. The application can be used by those who have permission. Permission requires a valid username and a password corresponding to the username, and all that information should match in the database in order to proceed forward to the application.
# Table of Contents

1. Introduction .............................................................................................................. 1

2. Application Review .................................................................................................... 3
   2.1 SPM (Simple Project Management) application database ........................................ 3
   2.2 GRAPHICAL User Interface ................................................................................... 4
       2.2.1 Index.php file ................................................................................................. 6
       2.2.2 Customers Menu .......................................................................................... 8
       2.2.3 Projects Menu ............................................................................................... 11
       2.2.4 ToDos Menu ................................................................................................. 12
       2.2.5 Users Menu .................................................................................................. 13

3. Problem Statement .................................................................................................... 16

4. Methodology ............................................................................................................. 17
   4.1 Database Design ................................................................................................... 17
       4.1.1 Using PHPMyAdmin .................................................................................. 17
       4.1.2 WampServer .............................................................................................. 18
   4.2 Graphical User Interface ...................................................................................... 19
       4.2.1 Adobe Dreamweaver CS IV ....................................................................... 19
   4.3 Overview of the PHP Structure and Syntax ........................................................... 20
       4.3.1 How PHP Fits with MySQL ....................................................................... 21
       4.3.2 Passing Variables ....................................................................................... 22

5. Critical Discussion .................................................................................................... 24
   5.1 Pros of SPM ......................................................................................................... 24
   5.2 Cons of SPM ......................................................................................................... 24
   5.3 Software Engineering Models .............................................................................. 25
       5.3.1 Analysis of Feasibility ................................................................................ 25
       5.3.2 Analysis of Requirements .......................................................................... 25
       5.3.3 Software Design ......................................................................................... 25
       5.3.4 Testing ......................................................................................................... 26
       5.3.5 Software Deployment ................................................................................. 26
## Table of Contents

5.3.6 Maintainability and Usability ........................................................................................................... 26  

6. Conclusion ................................................................................................................................................. 27  

7. Reference / Bibliography ......................................................................................................................... 28  

8. Abbreviations ............................................................................................................................................ 29  

9. Appendices ................................................................................................................................................. 30  
   9.1 Appendix A ........................................................................................................................................... 30  
   9.1.1 mysql_close.php ............................................................................................................................... 30  
   9.2 Appendix B ........................................................................................................................................... 30  
   9.3 Appendix C ........................................................................................................................................... 31  
   9.4 Appendix D ........................................................................................................................................... 31  
   9.5 Appendix E ........................................................................................................................................... 32  
   9.6 Appendix F ........................................................................................................................................... 32  
   9.7 Appendix F ........................................................................................................................................... 33
List of Figures:

Figure 2.1: SPM Database Relationship Diagram ................................................................. 3
Figure 2.2: Login Form .............................................................................................................. 4
Figure 2.3: MD5 Encrypted Password ....................................................................................... 5
Figure 2.4: Login Information Error ......................................................................................... 6
Figure 2.5: SPM Main Window ................................................................................................. 7
Figure 2.6: Customer’s Table ................................................................................................. 8
Figure 2.7: Edit Customer Form .............................................................................................. 9
Figure 2.8: New Customer Being Created .............................................................................. 10
Figure 2.9: Project’s Table ..................................................................................................... 11
Figure 2.10: Todo’s Table ..................................................................................................... 12
Figure 2.11: User’s Table ...................................................................................................... 13
Figure 2.12: New User Being Created .................................................................................... 14
Figure 2.13: Warning Message Box ....................................................................................... 14
Figure 2.14: User Deleted Successfully .................................................................................. 14
Figure 2.15: Edit User Form .................................................................................................. 15
Figure 2.16: E-mail address not valid ..................................................................................... 15
Figure 2.17: Successfully Modified User Message ................................................................. 15
Figure 4.1: Wamp Server ..................................................................................................... 19
Figure 4.2: Adobe Dreamweaver ......................................................................................... 20
1. **INTRODUCTION**

In order for different companies to achieve their goals in accomplishing their tasks on schedule, software project managements are developed. Today we are living in an era of Internet where online software serve as a great opportunity and tool to complete and manage our work in the easiest way known so far.

This thesis also tries to alleviate someone’s daily work, when it is a question of managing different projects. It can also be taken as an example of how a simple project management would look like in general. The steps how this application was developed will be presented in the next chapters, in a more detailed way for each part of the application. Even that I will try to explain in a concise way the development of this application, there will still be questions for some of the components of the application. That is because I will be mainly focused in the way of how PHP and MySQL programming languages play their role in making such applications happen. On the other side components like HTML, CSS, XHTML, etc will not be in the focus of this thesis, even that they play a big role in the entire job. I would strongly recommend inexperienced readers who want to step in the world of such applications to make a basic step into HTML, CSS, XHTML in order for him/her to face fewer challenges during different phases in the settlement of the application. Some of the most necessary components needed to build such an application would include PHP, Apache, MySQL, etc.

PHP is a server – side scripting language that allows your Web site to be truly dynamic. PHP stands for PHP: Hypertext Preprocessor.

Apache acts as your Web server. Its main job is to parse any file requested by a browser and display the results according to the code within that file. Apache is quite powerful and can accomplish virtually any task that a Webmaster may require.

MySQL is the database constructor that enables PHP and Apache to work together to access and display data in a readable format to a browser. It is a Structured Query Language server designed for heavy loads and processing of complex queries. As a relational database system, MySQL allows many different tables to be joint together for maximum efficiency and speed. [5]

PHP, Apache, and MySQL are all part of the open source group of software programs. The open source movement is a collaboration of some of the finest minds in computer programming. By allowing the open exchange of information, programmers from all over the world contribute to make a truly powerful and efficient piece of software available to
everyone. Through the contributions of many people to the publicly available source code, bugs get fixed, improvements are made, and a good software program becomes a great one over time. [5]

PHPMyAdmin is another wonderful open source project that enables you to access your MySQL database through a GUI. It’s easy to set up and manage, and it makes administering your databases, tables, and data a breeze. It does have some limitations, but for most part, it will make you a lot more efficient. [5] All these features mentioned above were used during the time I developed the application. I will be focused explaining each of these software programs in more detail in the next chapters of this thesis. Furthermore, as told earlier the ‘Simple Project Management’ application was developed using PHP and MySQL programming languages, which will cover most part of this thesis, their intercommunion to each other and so on to the end of this thesis.

The overall of this thesis include nine chapters excluding the Acknowledgements and Abstract chapters.

The ‘Application Review’ is the second chapter whose intent is to describe the application, its features, mainly the database and the graphical user interface. Also, the menus of the ‘Simple Project Management Application’ will be described in details.

The ‘Problem Statement’ chapter will provide us with the challenges that occurred during the time this application was developed, mainly the technical problems.

In sequence the ‘Methodology’ chapter describes the methods and phases used in order to achieve the goal of finishing the application.

‘Critical Discussion’ chapter focuses on the advantages and the disadvantages of the application. The waterfall software engineering model was used to analyze different phases like: feasibility, requirements, design, testing, deployment, maintainability and usability phase.

The sixth chapter is a conclusion of the whole thesis. Furthermore, there will be concluded whether the goals of the thesis were attained or not.

The ‘Reference / Bibliography’ chapter offer the sources that have been used in this thesis. Chapters ‘Abbreviations’ and ‘Appendices’ portray some abbreviated terms found in the previous chapters, whereas in the ‘Appendices’ some essential programming code of the application is provided.
2. APPLICATION REVIEW

2.1 SPM (SIMPLE PROJECT MANAGEMENT) APPLICATION DATABASE

The database of the “Simple Project Management” application is made using MySQL (Structured Query Language) programming language.

A database refers to data organized and stored on a computer, which can be searched, retrieved and managed by another computer program. It is really important the way how the database is organized, its inside relations are crucial, in order to have a relatively good performance when it is used by an application or anything else. Any changes that might be needed to be done in the future would cost a lot of time, and money. So, it is preferable to pay more attention during the design phase than healing it by a later on intervention. Database is the feature that makes a web site or an application to be truly dynamic or not, and not because of animation or real time applications.

SPM’s database consists of six (6) tables. The tables and their relationship are shown in the relationship diagram below, in Figure 2.1.

As you might see from the Entity Relationship diagram, there is a ‘customers’ table which contains the information for the customers, another table is for documents, one for projects,
one for the project work details ‘project_work’, one for the things that have to be done, called ‘todos’ table, and the last one is the ‘user’ table, which holds the information about different users.

The creation of the database opens the way for further development of the application. So, after the database is made, we can go on with next step of building the application, which would be the use of the other programming languages such as HTML, CSS, PHP & MySQL, etc.

2.2 GRAPHICAL USER INTERFACE

The entire code written for this application was realized using Adobe Dreamweaver CS4 software. Next, I will be presenting some of parts of the application, starting from the Login Form, which can be seen in the Figure 2.2 below.

As you can see from the above figure, this login form requires an authentication for the user that wants to enter the SPM application. This form contains two input fields. Input fields are the areas where the user types in the information, as required from that specific input field, in order to proceed forward.
**Username input field** – the user is required to give his/her username that matches with the username located in the database.

**Password input field** – after a valid username has been entered, the Password input field has to be fulfilled with a valid password that corresponds to the Username input field given above. If the username and the password match to each other this form would lead us to the main window of the application. Verifying whether or not the information given in the form is correct, a process of authentication and verification will be done in the ‘user’ database table where the data is stored. Below is provided a figure that shows the username and password given for that Username and Password field. Notice that the password is always encrypted with the Message Digest 5 (Md5) encrypting algorithm for security issues. This can be seen in the password field provided below in Figure 2.3.

<table>
<thead>
<tr>
<th>u_id</th>
<th>username</th>
<th>spent_time</th>
<th>password</th>
<th>email</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Administrator</td>
<td>0</td>
<td>b1c21019a8d435194216c37f14ecce</td>
<td><a href="mailto:test@admin.test">test@admin.test</a></td>
<td>2</td>
</tr>
<tr>
<td>☐</td>
<td>Administrator</td>
<td>0</td>
<td>7bd5252ec4d79533262799bc1368ca6</td>
<td><a href="mailto:admin1@mini.com">admin1@mini.com</a></td>
<td>2</td>
</tr>
<tr>
<td>☐</td>
<td>Adem Jashari</td>
<td>0</td>
<td>b57a856237436257040da8dc3d7b6b9</td>
<td><a href="mailto:adem@gmail.com">adem@gmail.com</a></td>
<td>1</td>
</tr>
</tbody>
</table>

**FIGURE 2.3: MD5 ENCRYPTED PASSWORD**

The code of the form above (Figure 2.2) can be found at Appendix F, where you could have a look how it was programmed. It is not so difficult to be understood.

If the information given in the input fields is not correct or better to say does not match the one in the database, the access for that user is denied. This is illustrated with the example of Figure 2.4, where user “Letra Musliu” and the password typed did not match.
Even if you press the Log In button without having written your username or password, an error message will be shown, that would require entering the username, password or both of them at the same time.

2.2.1 INDEX.php file

If the login information entered in the previous form matched with the information in the database, the user continues to the main page of the application, to the index.php file.

The user who has logged in successfully will have the opportunity to play with the menus. The logged in user’s name and e-mail address will be shown all time at all forms during the use of the application. This can be found at the right side of the forms (example: see Figure 2.6). Next to the e-mail address are: ‘Edit account’ and ‘Log out?’ links where the account of that specific user can be directed to the edit form, whereas hitting the ‘Log out’ link, the user will be logged off.

There are four menus: Customers, Projects, Todos, and the Users menu.

Customers menu offers some general information about customers, such as their name and so on.
Projects menu handles the projects being processed. The time spent on the projects is counted in hours, as you can see in Figure 2.9.

The Todos menu shows the work that has to be done next. The time when working on a project has started is shown and also when it was last edited (see Figure 2.10).

Users menu and the menus mentioned above will be explained in more detail later on during this chapter one by one.

The main page can be seen in the Figure 2.5, as shown below:

As can be seen from the above figure, it includes some fancy features, providing some links that deal with the application use in general. There is photo gallery, where pictures can be added. The ‘todo list’ links to the todo menu, where new todo can be created or the one being processed can be viewed. The calendar shows the correct date and is automatically updated from the operating system by the computer used to view the application. The ‘Add Event’ button enables to highlight the dates where something important is to happen, deadlines or whatever it might help.
2.2.2 Customers Menu

‘Customers’ menu offers information about customers. Starting from name, their companies, a description, e-mail addresses, country information, and also some tools to manipulate and manage customers with.

Below is Figure 2.6, which provides us with the customers table.

![Customers Table](image)

**FIGURE 2.6: CUSTOMER’S TABLE**

Using the tools, ‘Delete’ and ‘Edit’, are pretty much the same as the one found at the users form. The process of deleting a customer is completely the same as deleting a user, which will be explained in more detail at the ‘Users’ menu later on.

Updating or editing customers differs a little bit because of the different information the two different forms offer. So, we will be describing customers editing below supported by Figure 2.7,
The customer editing form integrates all the information of the customer that is to be edited. In that way it makes editing the customer much easier, because the information that you do not want to change is already there, and the part that you must change, just have to delete and overwrite it. At the right end, click the ‘Edit Customer’ button to execute the query that updates the data in the database, exactly in the customers table of the database.

If the query was submitted correctly a message box is applied to confirm that the data was modified successfully.

Creating a new customer is also not so difficult, since it is pretty much the same as creating a user. The form for creating a new customer would look like Figure 2.8, which can be found right below:
As you might have already noticed, creating a new customer differs from editing a customer. Just like we said, when we edited a customer the information that was in the fields remained there, otherwise when creating a totally new one, the input fields are empty. So we have to fill them in, with the information needed, as required, see Figure 2.8, for a better understanding.
2.2.3 **PROJECTS MENU**

The ‘Project’ table offers some opportunities to handle some part of data. As the Project’s table name implies, it deals with the current projects being handled and processed of people managing them.

It includes a field where there is a description for the customer and the project related to that specific customer. The time being spent for the project is calculated in hours, as can be seen from the Figure 2.9, below. There are some tools to manipulate with, the edit and delete functions.

![Figure 2.9: Project's Table](image)

**FIGURE 2.9: PROJECT'S TABLE**
2.2.4 **Todos Menu**

This menu lists all the work that needs to be done. Just like the other menus, it offers some options to conduct different functions offered.

The ‘todo’s short statement’, is part of this menu, which describes the todo shortly. The ‘Created’ field tells the date when that work was created, just like the edit option, which offers us know when it was last updated, or edited. The tools for deleting and editing are shown in the Figure 2.10, provided below:

![Figure 2.10: TODO’S Table](image)

**FIGURE 2.10: TODO’S TABLE**
2.2.5 Users Menu

After having logged in successfully to the application, and the ‘Users’ menu was chosen this window is opened, as in Figure 2.11.

![User Table](image)

**FIGURE 2.11: USER’S TABLE**

The user table lists all users registered and some information for each of them, as well as some managing tools. From this place users can be edited, deleted or even created by hitting the ‘Create User’ link given on the upper side of the window. When that button is clicked, Figure 2.12 opens.
When the ‘Delete’ button is pressed a warning message is shown to notice the final decision of deleting or canceling the process of deletion. A warning message box would look like the one presented in Figure 2.13.

Deletion will take place when the ‘OK’ button is confirmed. At that moment the data for that user will be deleted in the database. Maybe it is valuable to mention that the programming background of this Message Box is JavaScript. When the user is successfully deleted a message like the one in Figure 2.14 will appear.

This message appears on top of the users list, after deletion has taken place successfully.
On the other side when the ‘Edit’ button is clicked which is located next to the ‘Delete’ button (see Figure 2.11), another window is opened. For example if I want to edit the user with the name Letra Musliu, hit the ‘Edit’ button, automatically this window opens, refer Figure 2.15 below.

As we can see from the above figure, information such as name, password, e-mail and user level can be edited by this form. There are some requirements set for some fields, such as the password must be longer than 5 characters, e-mail specifications (see Figure 2.16, email is NOT VALID), etc. After the input fields of information are filled the ‘Update User’ button should take place, in order for the information to be updated in the database.

Figure 2.17, above shows that the user has been modified successfully
Problem Statement

3. PROBLEM STATEMENT

As probably mentioned earlier this was the first time for me, developing such an application with such programming languages. Gaining knowledge was one of mine secondary goals of this Thesis. Unfortunately, some barriers appeared to stay on my way from the time I decided to work on my application. After I had decided to do something like this, I started gathering different materials that dealt with PHP and MySQL, but it was not that easy to find the one that would fulfill my needs. The application was developed during a period of time where I had been intensively working on some other projects. This drawback of time caused some problems of managing time where I could fully be concentrated on this Thesis. Even though there were some attempts to make SPM application more functional, for the time speaking, this couldn’t be done. But anyway, in general I could say of being satisfied for some of its parts.

The friendly graphical user interface offers a great way to handle the information about the customers, instead of communicating with a harshly looking database. However, just like any other application it has its advantages and disadvantages which I will try to improve in the future when an application like this is needed.

Always, when somebody wants to start creating something, he/she has to look for a model where could start, be motivated and serve as a guide for a further evolution in that specific field.

For those who want to take a step forward in applications programmed in PHP and MySQL, this one can be taken as a simple sample. It is preferable to have a basic knowledge in HTML or XHTML. Inexperienced students or whoever they might be can start building such an application by thinking of a simple database design, implement it. After the database has been created, start cooking some HTML code, in building the Graphical User Interface (GUI), where you really can make some fancy forms. The code can be written in different editors, but I would recommend Adobe Dreamweaver as your place to write your code. After the forms are created you can start clothing HTML with PHP and MySQL in order to be connected to the database and be able to manipulate with the data.
4. METHODOLOGY

This chapter will examine the methods used in this study to illuminate its central questions about how the ‘PHP Simple Project Management’ application was developed, through its phases and through the necessary tools needed for it. This approach enabled the exploration of understanding how PHP and MySQL programming languages work and intersect together in making such an application be created. It also provides us with the cooperation of these two programming languages with the Hypertext Markup Language.

4.1 DATABASE DESIGN

The first step made when started to make an application like this or like any other application, is the database concept. Before anything else one must have a clue how the database would look like, than continue further. The database for this application was made in MySQL. Since those databases are relational, they allow you to separate information into different tables, whereas in non-relational systems, all the information is stored in one big area, which makes it much more difficult to sort and extract the data you want. I have been using the so called ‘phpMyAdmin’ software to access the data. Next, a more detailed explanation of this software will be presented.

4.1.1 USING PHPMYADMIN

PHPMyAdmin is considered to be the best friend for the people who have dealt with databases related to application like SPM.

PhpMyAdmin is an open source software that enables us to access MySQL database through a Graphical User Interface (GUI). It is easy to set up and manage, and makes administering your databases, tables much easier.

With this software, you can do the following:

- Drop and create databases
- Create, edit, and delete tables
Methodology

- Create, edit, and delete fields
- Enter any MySQL statements
- View and print table structure
- Generate PHP code
- View data in table format [1]

The reason for using PHPMyAdmin was its friendly user interface and the way it makes administering data easier. I could start and type a lot of code lines in programming the tables and fields, and the other different features of the database, but I decided to create the database by using PHPMyAdmin. At the end I was really thankful for the time it had saved me.

This wonderful open source project comes as part of the ‘WampServer’ software, which will be illustrated next.

4.1.2 WAMP SERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache, PHP and the MySQL database. It also comes with PHPMyAdmin to manage your databases. [2]

Some of the functionalities that WampServer offers are:

- Manage your Apache and MySQL services
- Switch online/offline (give access to everyone or only localhost)
- Install and switch Apache, MySQL and PHP releases
- Manage your servers settings
- Access your logs
- Access your settings files
- Create alias [2]
This is a screenshot of the WampServer that I have been using all the time. As you might have noticed this is a 5.1.36 version. It includes Apache 2.2.11, MySQL 5.1.36, PHP 5.3.0. [3] The files/web pages or whatever other formats that are hosted on WampServer can be accessed by typing http://localhost or http://127.0.0.1/ in the address bar of the browser. WAMP must be running in order to access either of the above addresses.

After the files have been created and hosted on WampServer, you can test them to see your work. It is important to specify the pathway in a correct manner in order to access and the result to be displayed.

4.2 GRAPHICAL USER INTERFACE

4.2.1 ADOBE DREAMWEAVER CS IV

As probably mentioned earlier, I have been using Adobe Dreamweaver CS IV as my editor. It is the best editor that I have ever tried so far, this opinion can be taken for most of the people that I work with. It is definitely friendly. I used the version IV of Adobe Dreamweaver, lately Adobe Systems developed Adobe Dreamweaver CS V which is the final product, so far for this kind of editor.

Dreamweaver is available for both Mac and Windows operating systems. Supporting web technologies such as CSS, JavaScript, and various server – side scripting languages and frameworks including ASP, ColdFusion, and PHP. [4]
From the figure above the design of this software can be seen. You can see some parts of it, some lines of code and a half window of design. The first half presents the code written, some of that code is PHP, some CSS, XHTML, JavaScript, MySQL, etc. It is important to mention that the PHP programming lines (opening and closing tags) in Adobe Dreamweaver are red colored. Inside the PHP code MySQL code is embedded. The second half is the design or better to say the output of the code written above of the design. This is a good opportunity, because when code is typed you immediately see the output of it. Otherwise, it is also possible to have the whole window covering the code or the design. In general, this is a software of many functionalities.

4.3 OVERVIEW OF THE PHP STRUCTURE AND SYNTAX

PHP programs nowadays are mostly written in some more sophisticated software, just like I did with Adobe Dreamweaver. But, it can also be done using a text editor, such as Notepad or WordPad, just like HTML pages. For beginners, from my experience it is preferable to use simple notepads, because of the experience that they absorb during the time they have to write the code from scratch. The PHP pages or files for the most part end
in a ‘.php’ extension. This extension signifies to the server that it needs to parse the PHP code before sending the resulting HTML code to the viewer’s Web browser.

When people go to a restaurant, they see plates of beautiful food served for them. But they do not see where the food comes from, or nor how and who prepared it. In a similar way, PHP fits into HTML code and is invisible to the people who visit the site. HTML can also be written inside PHP, this enables text formatting. Visitors of a site see only the output of the HTML, not the PHP, this gives a more security of the PHP code.

However, PHP can function without any HTML at all, as a standalone program. I usually used only PHP code when stored the connection variables and some other functions where the need of HTML code is not necessary.

4.3.1 HOW PHP FITS WITH MYSQL

In order for PHP and MySQL to be able to communicate to each other, it is necessary to enable MySQL in the ‘php.ini’ file. MySQL commands can be used within the PHP code all the time. Different functions of PHP work specifically with MySQL to make our programming much easier.

Some of the more commonly used functions are:

- mysql_connect ("hostname", "user", "pass"): Connects to the MySQL server.
- mysql_select_db ("database name"): Equivalent to the MySQL command USE; makes the selected database the active one.
- mysql_query ("query"): Used to send any type of MySQL command to the server.
- mysql_fetch_rows ("results variable from query"): Used to return a row of the entire results of a database query.
- mysql_fetch_array("results variable from query"): Used to return several rows of the entire results of a database query.
- mysql_error (): Shows the error message that has been returned directly from the MySQL server.
MySQL commands can also be sent to the server through PHP and the ‘mysql_query’ command. This can be done by sending the text through PHP either a variable or through the ‘mysql_query’ command directly, like this:

```php
$query = "SELECT * from TABLE";
$results = mysql_query ($query);
```

You can also do it like this:

```php
$results = mysql_query ("SELECT * from TABLE");
```

The results of the query are then put into a temporary array known as $results. [5]

### 4.3.2 Passing Variables

There are primarily four ways available to accomplish this task:

- Through URL
- Through a session
- Via a cookie
- With an HTML form

Suppose, that you want something you have posted on the front page to be present to all, or to some of the other pages of the application. This can be done by using the methods mentioned earlier. The method is chosen based on the situation and what best fits for the needs. I have been using cookies for this application, which will be explained next.

#### 4.3.2.1 Passing Variables with Cookies

The main purpose of cookies is to identify users. When you enter a Web site using cookies, you may be asked to fill out a form of some information as your name and some other information. This information is packaged into a cookie and sent to the Web browser which stores it for later use. Next time, there is no need to enter the information that you had to give last time. This time the browser will send the cookie to the Web server.

Cookies are tiny bits of information stored on the Web site visitor’s computer. Some people hesitate about the use of cookies, there are many people that decide to disable this feature in
their web browsers. In theory, cookies can be intercepted to gain information such as a person’s IP address and operating system, but they are primarily used for storing information only.

Someone might ask why developers use cookies, despite of the security issues?!

The advantage of storing information in a cookie versus a session is longevity. Session can store information alone only until the browser window is open. Cookies on the other hand, can live on the computer until the developer has decided it’s been long enough and they automatically “die”. The longevity makes them be such fabulous.

To set a cookie, the ‘setcookie()’ function is used. Some information set to a cookie are mandatory, some are not. The following information can be determined along with the cookie:

- Cookie name (this is mandatory)
- Value of the cookie (such as the person’s username)
- Time in seconds when the cookie will expire. (This time is based on a UNIX timestamp, but you can set it using the syntax time ()+60*60*24*365, which keeps the cookie alive for a year. This is optional, but if it is not set, the cookie will expire when the browser is closed.)
- Path (the directory where the cookie will be saved – the default is usually sufficient, this is optional)
- Domain (domains that may access this cookie – this is optional)
- Whether a cookie must have a secure connection to be set (defaults to 0, to enable this feature set this to 1)

The structure how to set each of these attributes to a cookie is as follows:

```python
setcookie('cookiename', 'value', 'expiration time', 'path', 'domain', 'secure connection'). [5][7]
```

These were some of the main parts that contributed to establish the development of the Simple Project Management Application. Absolutely, it would be not fair if we would narrow ourselves and say that there is nothing else included in this application. Some of the programming code will be provided at the Appendices chapter, which could help anyone who might be interested about the way how this application was programmed. This is not the only way, there are several methods and several opportunities. Just be creative and work hard, this is all that is required from computer scientists.
5. CRITICAL DISCUSSION

This chapter will provide us with some of the advantages – pros, disadvantages – cons, and the encountered results from different phases of the Software Engineering Model for the Simple Project Management Application.

5.1 PROS OF SPM

Simple Project Management application is a web based application which offers an easy – to – use online teamwork. It helps the people working in a company work together more productively. People working on different projects will be noticed. Everyone will know who is working on what, when the deadlines are, how much time has been spent etc. It also enables to see the work that has to be done, have the list of the customers, and so on. It can be accessed from any type of computer without installing software on user’s computer, it is also ease of access and control (the password and the username are required).

Nevertheless, the Application Review chapter offers a more detailed explanation about SPM and its uses, where you could refer for more details.

5.2 CONS OF SPM

Despite of the advantages that this application offers, some cons are encountered:

- It responds slower than a desktop application
- The information about the projects are not available when the user or server is offline
- In case of a forgotten password or username there is no solution at this moment, even that I will do it in the future
- There is not search option within the application, where users could search for specific things of their interests.
- No help menu
5.3 SOFTWARE ENGINEERING MODELS

I have been using the Waterfall Model software development to establish the requirements for this application. This model incorporates in itself some phases which will be described below.

5.3.1 ANALYSIS OF FEASIBILITY [6]

This project may help companies to accomplish their regularly given tasks. SPM was developed by one person (the foregoing student) and there is no need for much of personnel. Despite of that, there are some prerequisites that someone needs, in order to make such a project happen: a web server (Apache or even WAMP server which integrates Apache server in it), a place where you can write the code (preferable software like Adobe Dreamweaver CS IV or newer version) or just a simple notepad which would gain beginners with experience. If one decides to take WAMP as his/her server, than most of the things are completed, because there is also the PHPMyAdmin open source project through which the MySQL database can easily be handled. A web browser would help to test the recent work, make any improvements if needed before launching.

This project was done as a requirement for my Bachelor Degree by the Faculty of Computer and Engineering at the University for Business and Technology, and brings me no predefined profit, except the gained knowledge and the degree as mentioned earlier.

5.3.2 ANALYSIS OF REQUIREMENTS [6]

There was no direct request for making such an application, but being aware of what companies need to have in order to perform their daily work, I decided to start developing it. I also needed to do something like that, so I could learn from this field of web development. The main purpose is to help people manage their work, by having under control projects and customers in general. No need of software is required, a web browser should be enough to use it.

5.3.3 SOFTWARE DESIGN [6]

The Graphical User Interface is considered to be friendly. It was made using Adobe Dreamweaver CS IV software, populating it with HTML, CSS, PHP, MySQL and some secondary programming languages for this application. The design can be more understandable if you go back and refer to the Application Review chapter.
5.3.4 Testing [6]

There have been several tests during this phase. No problems were encountered in the database neither the forms nor the Graphical User Interface. Some web browsers, such as Google Chrome, Internet Explorer, Mozilla Firefox were used to test the application if it is able to work on all of the previous browsers above, and the results told that the application has been implemented successfully.

5.3.5 Software Deployment [6]

Everybody has a web browser, and therefore all that is needed to be done is to enter the URL for accessing the tools the application offers. When I talk about deploying this software over the web, I mean that the actual application is available to be used in real-time over the web. The URL address will be available to those who are part of the company using this application. It runs on server and users access it through a web browser remotely.

5.3.6 Maintainability and Usability [6]

SPM application is available for use by anyone working in those companies who have accepted this application as a tool to work with. Users can not change anything without any professional help, especially by the original developer. Also, in case of any data loss, proper help should be required, mostly the author should intervene. From time to time it is recommended for the company to ask for professional help in inspecting the application for any security and malicious issues that might appear.
6. CONCLUSION

After having tested the project by some volunteers, and based on the results of the application performance I would likely think that the application has fulfilled the main goal of being able to help people manage their capacities at work.

Except that I always had in my mind the opportunity of trying to learn something new, which I think have achieved. The phases that the project had to go through made me gain enough knowledge in order to develop a more sophisticated application a near future. It would be fair enough to say that maybe this thesis may help inexperienced people of all ages to start doing such things.

With some modification I will be starting building an application like this one, but a more expanded one in the way of offering more options to the users. This application will serve the cadets of the Centre of University Studies (CUS) at Kosova Security Force (KSF). Probably, being part of it, I might know what the needs are more accurately, than the design of the database would be much easier to be imagined, and so on.

Obviously, SPM application would have been more powerful if more time was in disposal and if no interruption occurred during different phases of the thesis development.

Indeed, as secondary, this period of time provided me with the opportunity of being able see how PHP & MySQL applications are developed, both, their advantages and disadvantages. The wide use of them has made me aware of dedicating more space to those two programming languages, PHP and MySQL.

In conclusion, I hope to have written something, where somebody could read and learn with satisfaction.
7. REFERENCE / BIBLIOGRAPHY


[8] Janet Valade, Tricia Ballad and Bill Ballad PHP & MySQL® Web Development All-in-One Desk Reference For Dummies®


8. Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>KSF</td>
<td>Kosova Security Force</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>PHP</td>
<td>Preprocessor Hypertext</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SPM</td>
<td>Simple Project Management</td>
</tr>
<tr>
<td>MD5</td>
<td>Message Digest 5 Algorithm</td>
</tr>
<tr>
<td>CUS</td>
<td>Center for University Studies</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>XHTML</td>
<td>Extensible HyperText Markup Language</td>
</tr>
</tbody>
</table>
9. APPENDICES

This chapter includes some programming code of the application explained before.

At this moment, it would be almost impossible to provide every single line of code used to build the application, because of the huge space it occupies. So, I decided to show some parts (functions), where I thought are more important and require less space.

9.1 APPENDIX A

Connection.php, file

```php
<?php
    error_reporting(E_ERROR);
    $connect = mysql_connect("localhost", "root", 
"
")or die("Connection not established 
".mysql_error());

    mysql_select_db("simpleprojectmanagement")or die("Connection not established to the database 
".mysql_error());
?
```

9.1.1 MYSQL_CLOSE.PHP

```php
<?php
    mysql_close($connect)or die(mysql_error());
?
```

9.2 APPENDIX B

Check_login.php

```php
<?php
    if(!$_COOKIE['logged_id'])
        header("Location: login.php");
?
```
9.3 APPENDIX C

Get_customer_info.php

```php
<?php
    function getcustomer_info($customer_id)
    {
        $result = mysql_query("SELECT * FROM customers WHERE c_id = $customer_id");
        $fetch = mysql_fetch_array($result);
        return $fetch;
    }
?>
```

9.4 APPENDIX D

Get_user_info.php

```php
<?php
    function getUserInfo($user_id)
    {
        $getuser = mysql_query("SELECT * FROM user WHERE u_id = $user_id");
        $fetch = mysql_fetch_array($getuser);
        return $fetch;
    }
?>
```
Appendices

9.5 APPENDIX E

Logout_checker.php

```php
<?php
if($_GET['logout']=="true")
{
    setcookie('logged_id',false, time());
    header("Location: login.php");
}
?>
```

9.6 APPENDIX F

User_check.php

```php
<?php
function userExist($username, $password)
{
    $encrypted_password = md5($password);
    $query = "SELECT * FROM user WHERE username = \\
    '$username' AND password = \\
    '$encrypted_password' ";
    $result= mysql_query($query);
    if(mysql_num_rows($result))
    {
        return true;
    }
    return false;
}
?>
```
9.7 Appendix F

Login_form.php

```php
<?php

if(isset($_COOKIE['logged_id']))
{
    header("Location: index.php");
}

//Avoid error messages that are not so crucial in the way of effecting the code work properly
error_reporting(E_ERROR);
include("mysql_open.php");
include("functions/usercheck.php");

$username = $_POST['username'];
$password = $_POST['password'];
$login = $_POST['login'];

if( isset($login) )
{
    if(empty($username))
    {
        $message= "Please enter your username.";
    }
    elseif(empty($password))
    {
        $message="Please enter your password .";
    }
```
elseif(!userExist($username, $password))
{
    $message="Username or Password doesn't match !";
}
else
{
    $encrypted_password = md5($password);
    $get_id = mysql_query("SELECT * FROM user WHERE username = '$username' AND password = '$encrypted_password'");
    $fetch = mysql_fetch_array($get_id);
    setcookie('logged_id', $fetch['u_id'], time()+7200);
    header("Location: index.php");
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /
<title>SimpleProjectManagement - Ultimate Admin Panel Solution</title>

<meta name="description" content="" />
<meta name="keywords" content="" />
<meta name="robots" content="index,follow" />

<!--[if IE]><link rel="stylesheet" href="css/ie.css" type="text/css" media="screen, projection" />
<![endif]-->
<?php if($message){
    <div id="fail" class="info_div"><span class="ico_cancel"><?php echo $message; ?></span></div> <?php } ?>

<form name="loginform" id="loginform" action="" method="post">
    <label><strong>Username</strong></label>
    <input name="username" type="text" class="input" id="user_login" value="<?php echo $username; ?>" size="28"/>
    <br />
    <label><strong>Password</strong></label>
    <input type="password" name="password" id="user_pass" size="28" class="input"/>
    <br />
    <input name="login" type="submit" class="loginbutton" id="save" value="Log In" />
</form>

<a href="#" id="passwordrecoverylink">Forgot your username or password?</a></div>

<!-- end container -->

</body>
</html>

<?php include("mysql_close.php"); ?>