

NEAR EAST UNIVERSITY

Faculty Of Engineering

Department of Computer Engineering

**FARM AND MILK PRODUCER SYSTEM
PROGRAMMING USING PHP and MYSQL**

**Graduation Project
COM-400**

Student: Hikmet DAŞÇI

Supervisor: Mr.Ümit Soyer

Nicosia - 2006

ACKNOWLEDGEMENTS

It is my pleasure to take this opportunity to express my greatest gratitude to individuals who have given me a lot of supports during my four-year Undergraduation program in the **Near East University**. Without them, my Graduation Project would not have been successfully compelled on time.

First of all, I would like to express my thanks to my supervisor **Mr. Ümit Soyer** for supervising my project. Under the guidance of him successfully overcome many difficulties and I learned a lot about web designing. He always helped me a lot and I felt remarkable progress during his supervisor. Also I thank for giving his time during the my study and my advising.

I also want to thank all my friend and specially **Emre Selim Şahin, Mahsun Akbaş, Firat Silinir, Mahfuz Daşçi, Fadıl Doğan, Farah Diba Kartal, Süheyla Yıldırım, Gülnaz Süzen, Özgür Umut Vurgun, Mehmet Deniz Özyüksel**, who supported and helped me all the time.

Finally, special thanks for my family, especially my brothers **Mehmet Emin Daşçi** and **Mehmet Eta Daşçi** for being patientfull during my undergraduate degree study. I could never have completed my study without their encouragement and endless support.

ABSTRACT

The aim of this Project to help the Milk Producer, Veterinarian Association and KTSUTEK Association, by this system the association can register the farm and the farmer.

Using this system Milk Producer can control their production, and association can send information about the result of the milk. Also Milk Producer can get information easily and quickly about the announcement and the price of the milk.

The producer can easily pursuit the amount of milk that Association accepted or not, and also he can see how many animal in the farm by the Animal ID Number. The association can add, search and delete all information about the Milk Producer, and create the password for the farmer or for the other Administrator.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
CHAPTER ONE : ABOUT MY PROJECT	2
1.1 Farm and Milk Producer System Programming	2
1.1.1 User Interface Area	4
1.1.2 Admin Interface Area	10
CHAPTER TWO : WHAT IS THE HTML	18
2.1 What is the Html	18
2.1.1 HTML Tags	19
2.1.2 What is IIS?	20
2.1.3 World-Wide Web	21
2.1.4. SLIP and PPP	21
2.1.5 Advanced Conferencing Software	22
2.1.6 Proprietary Software vs. Open Standards	22
2.2 HTTP Authentication With PHP	22
CHAPTER THREE : WHAT IS MYSQL	25
3.1 Understanding MySQL	25
3.1.1 What Is the Enterprise?	27
3.1.2. How hard is it to change?	28
3.1.3. What Is a Relational Database?	29

3.2 Designing Databases	29
3.2.1 Connecting to Database	30
3.2.2 Encrypted Storage Model	30
3.2.3 The Client/Server Paradigm	31
3.2.4 History of Mysql	32
3.2.5 How Large MySQL Tables Can Be	33
3.2.6 Features of MySQL	33
3.3 Installation MySQL for Windows	34
3.3.1. File Contents	34
3.3.2. Changing Passwords	34
3.3.2. Changing Passwords	36
3.3.3 Starting and Stopping the Server	36
3.3.4 Using MySQL with Windows	37
3.3.5 Using MySQL the Command Line	38
3.2.6. Use MySQL	40
3.3.6 The Built-in Database	41
3.4 Database Security	41
3.4.1 SQL Injection	42
3.4.2 Avoiding Techniques	43
CHAPTER FOUR : WHAT IS APACHE SERVER	45
4.1 What is Apache	45
4.1.1 The Apache Server	45
4.1.2. Apache's architecture	45
4.1.3. More recent history	46

4.1.4. The Future of Apache	47
4.1.5 Obtaining and Installing Apache	47
4.1.6. Hardware / Software Requirements	47
4.1.7. Configuring	47
4.1.8. Configuration Files	48
4.1.9. Starting, Stopping, Restarting	48
CHAPTER FIVE : WHAT IS PHP	49
5.1 What is PHP?	49
5.2 What can PHP do?	49
5.3 General Installation Considerations	51
5.4 Installation on Windows Systems	52
5.4.1. Windows InstallShield	53
5.4.2 Building From Source	53
5.4.3 Putting It All Together	53
5.4.4 Compiling	54
5.5 Installation of Windows Extensions	55
5.5.1 Windows and PWS/IIS 3	55
5.5.2 Windows and PWS 4 or newer	56
5.6 The Configuration File	57
5.7 Security	58
5.8 General Considerations	59
5.8.1 Installed as an Apache module	60
5.8.2 Installed as CGI binary	60
CHAPTER SIX : APPLICATION CODES	62
6.1 User Interface Area Codes	62
6.1.1. Index.php	62

6.1.2. Farm.php	64
6.1.3. Farm info.php	68
6.1.4. Milk Info.php	72
6.1.5. Animal Info.php	77
6.1.6. About.php	82
6.1.7. Help.php	85
6.1.9. User_Function.php	89
6.1.8. Database.php	89
6.1.10. All_Function.php	89
6.2 Aplication Code For Admin Side	90
6.2.1. Index.php	90
6.2.2 Admin.php	91
6.2.3 Admin Manage.php	92
6.2.4 City.php	95
6.2.5 Farm.php	97
6.2.6 Farmer.php	100
6.2.7 Farmer Report.php	105
6.2.8 Milk Accept.php	107
6.2.9 Region.php	110
6.2.10 Milk Farm.php	113
6.2.11 Animal.php	115
6.2.12 Milk Price	118
6.2.13 Function.php	120
6.2.14 Logout.php	121
CONCLUSION	122
REFERENCES	123

INTRODUCTION

In the early days of the World Wide Web, information was stored in simple files with only HTML markup. Today, the web server often processes scripts within web files that e.g. call other files, stylesheets, etc. Furthermore, the scripts may also acquire information from a relational database management system. Such a modern system gets data from database tables and includes them in the web page that is delivered to the user.

To make database-backed system of web pages like Evolutionary Economics. we need a system of tools. Thus we leave the user's viewpoint that there is just a web server that is capable of sending web pages.

In the First Chapter I explained Farm and Milk Producer System Programming, and determined some word that it was mentioned in the system. I have given some example from the program code and also some page from the project

In the Second Chapter I explained about the HTML. What is html and how we use it that code.

In the Third Chapter I write about the Database, I have used MySQL for my project so I gave some information about the MySQL database,

In the Fourth Chapter I mentioned about the Apache server, how is work, the apache architecture, the history of apache.

In the Fifth Chapter I write about my fundamental program that it is known PHP (Personal Home Page) and again I write about php program and give some example about this language.

In the Last Chapter I write the Project code in two parts, one of them is User Interface Area, the other one is Admin Interface Area

Given that an adequate software system is related to the web server, the web site developer has a radically increased set of possibilities. A first glimpse of these possibilities may be obtained by inspecting one of the files that are used to produce a web page on such a system. The server-side file will probably have some ordinary text and HTML markup, but it will also have smaller or larger amounts of commands written in a programming ("scripting") language.

CHAPTER ONE ABOUT MY PROJECT

1.1 Farm and Milk Producer System Programming

In this chapter; I will explain the contents of project and also the goals of project by the some word that used in the Project.

The main goal of the Project is that, to registers the farmer and farmer production in the Internet environment where in the TRNC. And also to control farmer production step by step by this system.

Farmers sent their production to the KTSUTEK Association and after that this production is tested. These tests do everyday by KTSUTEK and Veteriner Dairesi (Veterinarian Association) the main of these tests to determine whether the milk is healthy or not. If the results of tests are not positive the milk is cancel. This result will send the farmer by the e-mail system.

There are two types of tests, one of them is tested by the KTSUTEK and the other one is tested by the Veterinarian Association. In the KTSUTEK Association they are tested Percent of oil, Protein, Lactose, Casein, water, and Frozen degree, Dirty, Pestisit etc. In the Veterinarian Association they check more sensitive tests. These tests are doing for whether are diseases or not, like Brucella, Antibiyotik, Salmonella, E.Coli, Stafilococ tests.

There are necessary for to find positive result in tests. If there is one negative test result, in that time the Veterinarian Association is contributed and send a Veterinarian Doctor to the farms to take blood from every animals for which animal is disease.

There are two type of ID number one of them is for animals. The animals ID number is determined by Veterinarian Association. The other one is for Farms and this ID is determined by the KTSUTEK Association.

And when the animal born they give an ID Number. There are two aims of ID Numbers, one is for take blood or take milk to test. Other one is for how many animals have in the farmer and how many kind of animal in the Country. That ID Numbers are created by 11 digits. The beginning of this ID started by NC lather and the remaining is should be numerical values. For example; like NC0000000000. In this number system

the first 0 is determines the kind of animals, like (Cow, Sheep, Goat) the other two 00 are determine the year and the other five 00000 are determine ears ID number, and the last 0 is using for control the ID Number that this ID Numbers is correct or not.

For example NC 3 06 00021 6 is determine that the number 3 is cow, 06 is determine the year of 2006, 00021 is determine ears ID, the last number of 6 gives opportunity for determine control.

Now this ID number system is doing by offices of Veterinarian Association in the eight different region in TRNC. This ID system is rearranged, before they used such type of system as VD000000000.

The farm and the milk producer system programming is created by two steps. One of this is for farmer and the other one is for admin (KTSUTEK). Farmer must get password from the KTSUTEK to use the system. Farmer can enter his production and after that he can get report about whether his production is accepted or not.

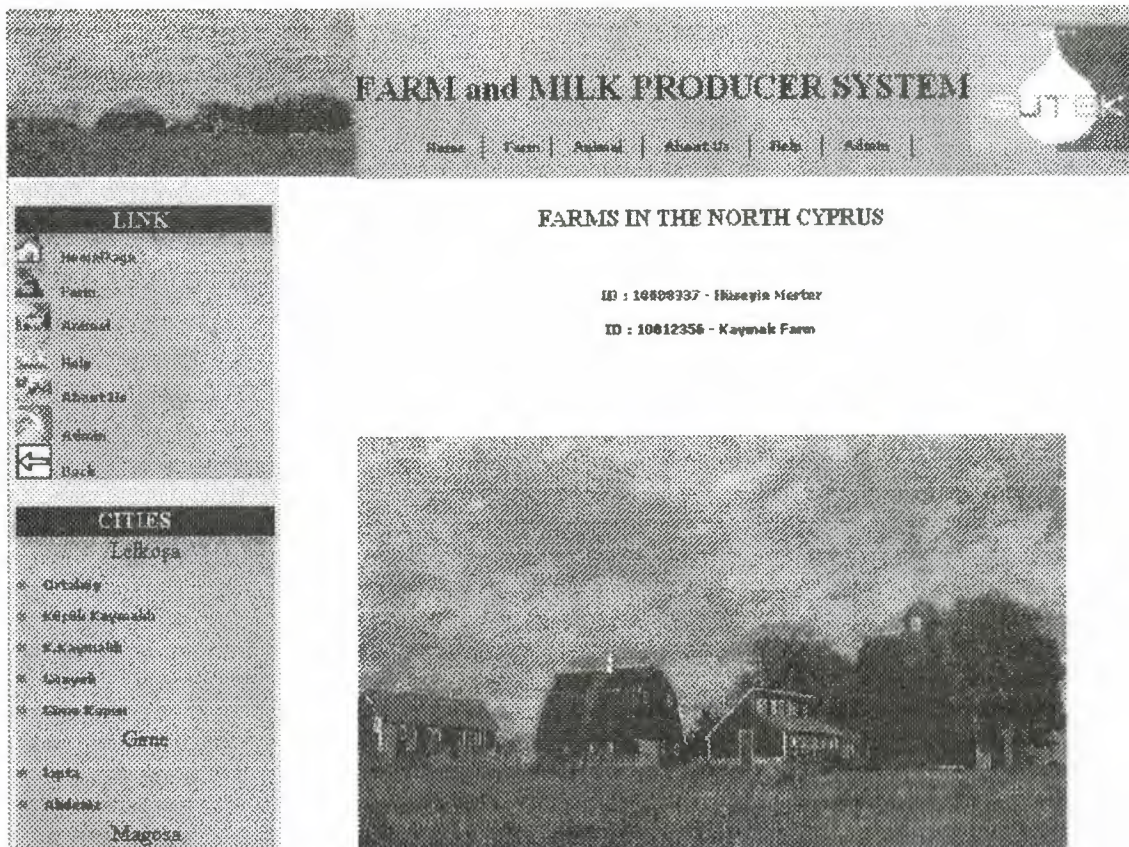
1.1.1 User Interface Area

When we enter the system we will see a page like follow figure 1.1



Figure 1.1

We will see 5 cities left of the page under **CITIES** word. Farmer can click a region name where is his farmer location in that city. For example; when he click the **ortaköy** like the figure 1.2



In this page (Figure 1.2) the Farms ID Numbers and the Farm name seen together, Farmer can check his information by click his farm name. for example when he click the **ID:10008337 – Hüseyin Merter**, he can see his information like; name, surname farmer name, kind of animal, how many animals have in the farm and total milk, as seen in the Figure 1.3

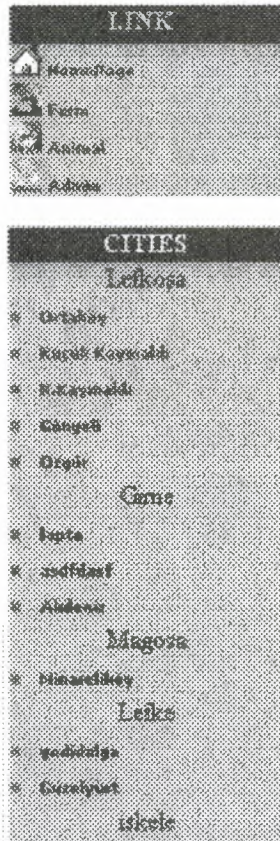


Figure 1.3

And in this page (Figure 1.3) for seen ID Numbers of animal in details, there is created a link. For get detail information we can click (**Cow, Sheep, Goat**) when we click that words we can see the following Figure 1.4

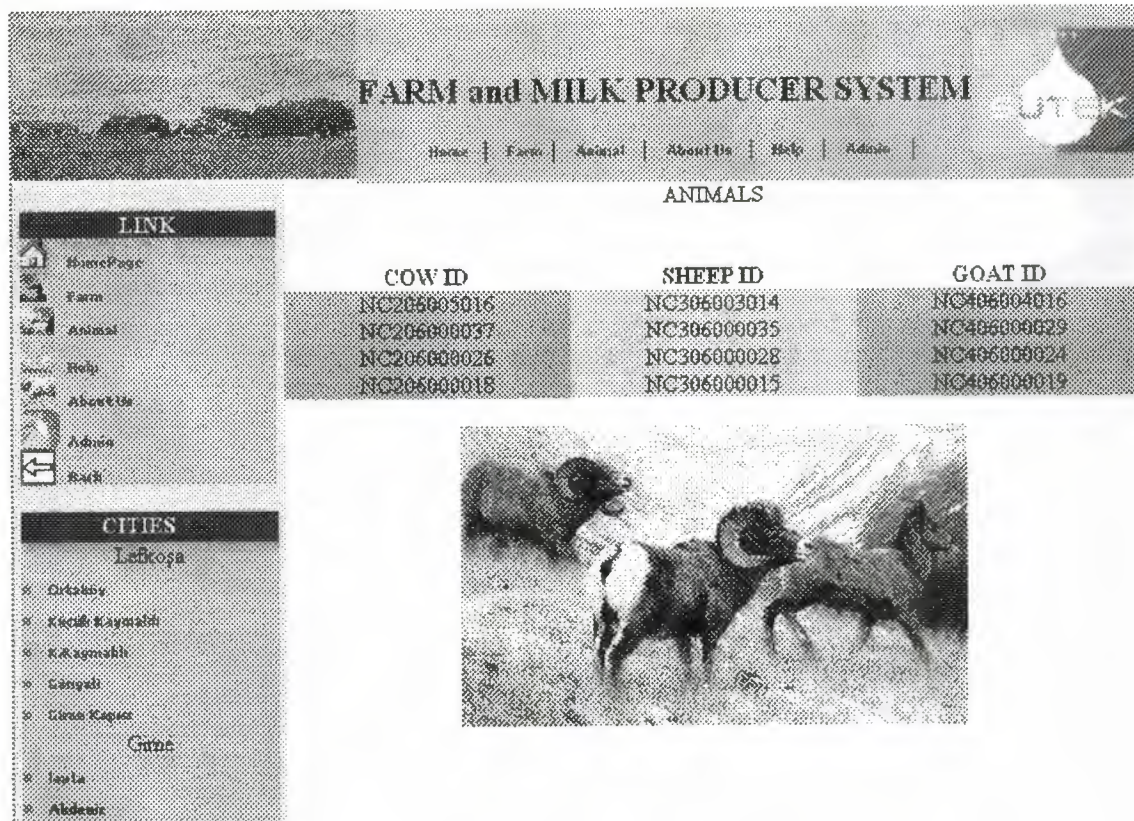


Figure 1.4

For see the milk information there is another link (Figure 1.4) when we click **Milk Information** we can see two parts, one of them is Show Incoming Milks to Accept. In other part show to us accepted amount of milk. Like the figure 1.5

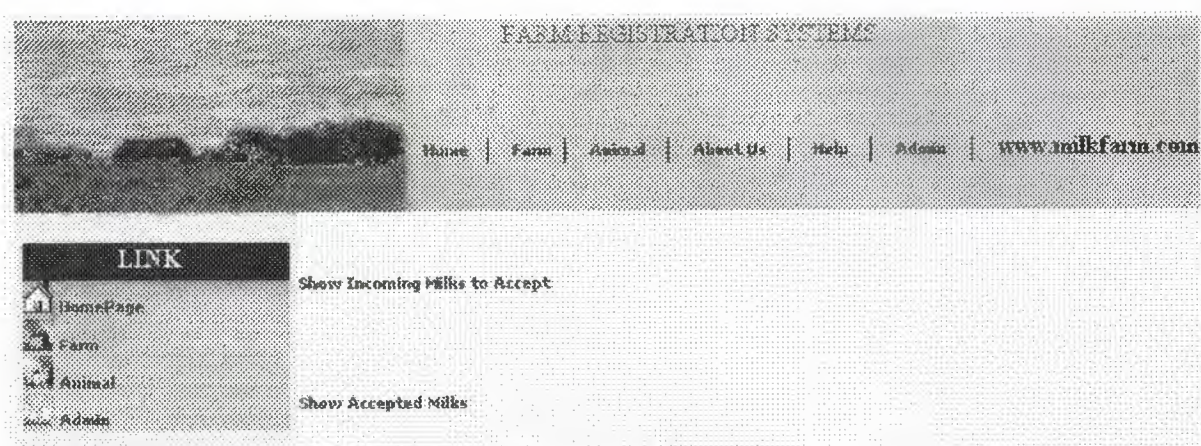


Figure 1.5

When we Enter Show Incoming Milk Accept we can see the report about the milk that is waiting to accept, like the Figure 1.6

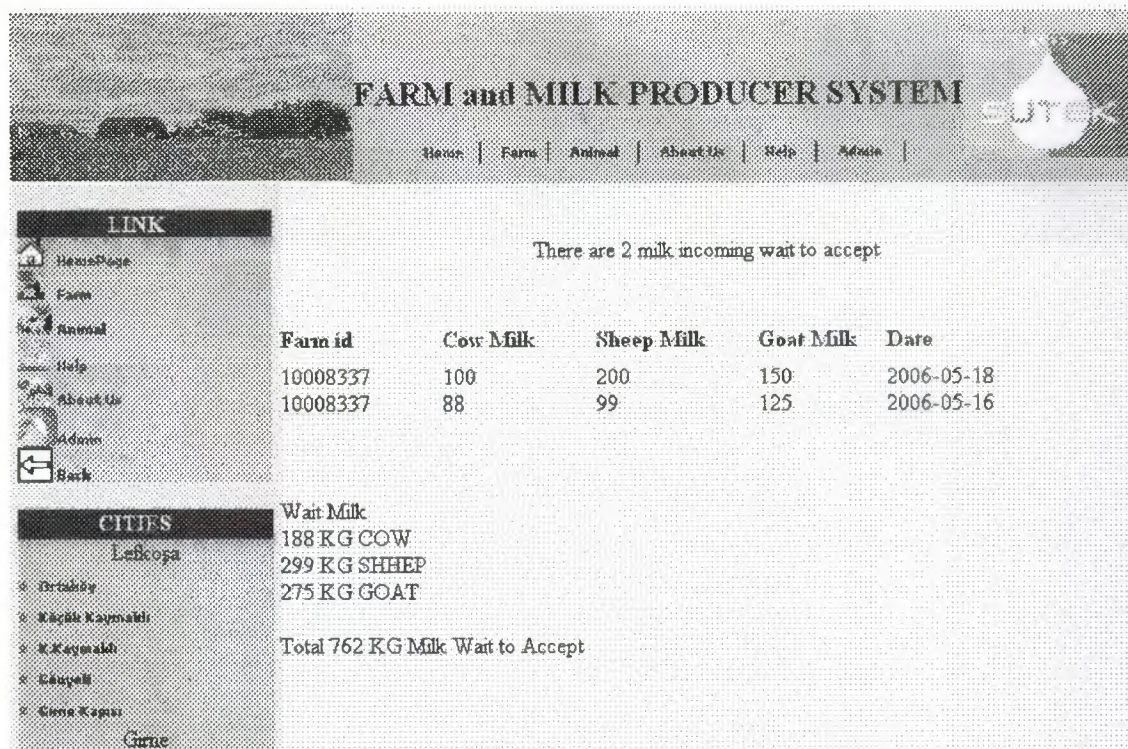


Figure 1.6

If we enter Show Accepted Milk we can see how many KG milk accepted by the Farmer ID number, and the date like the figure 1.7

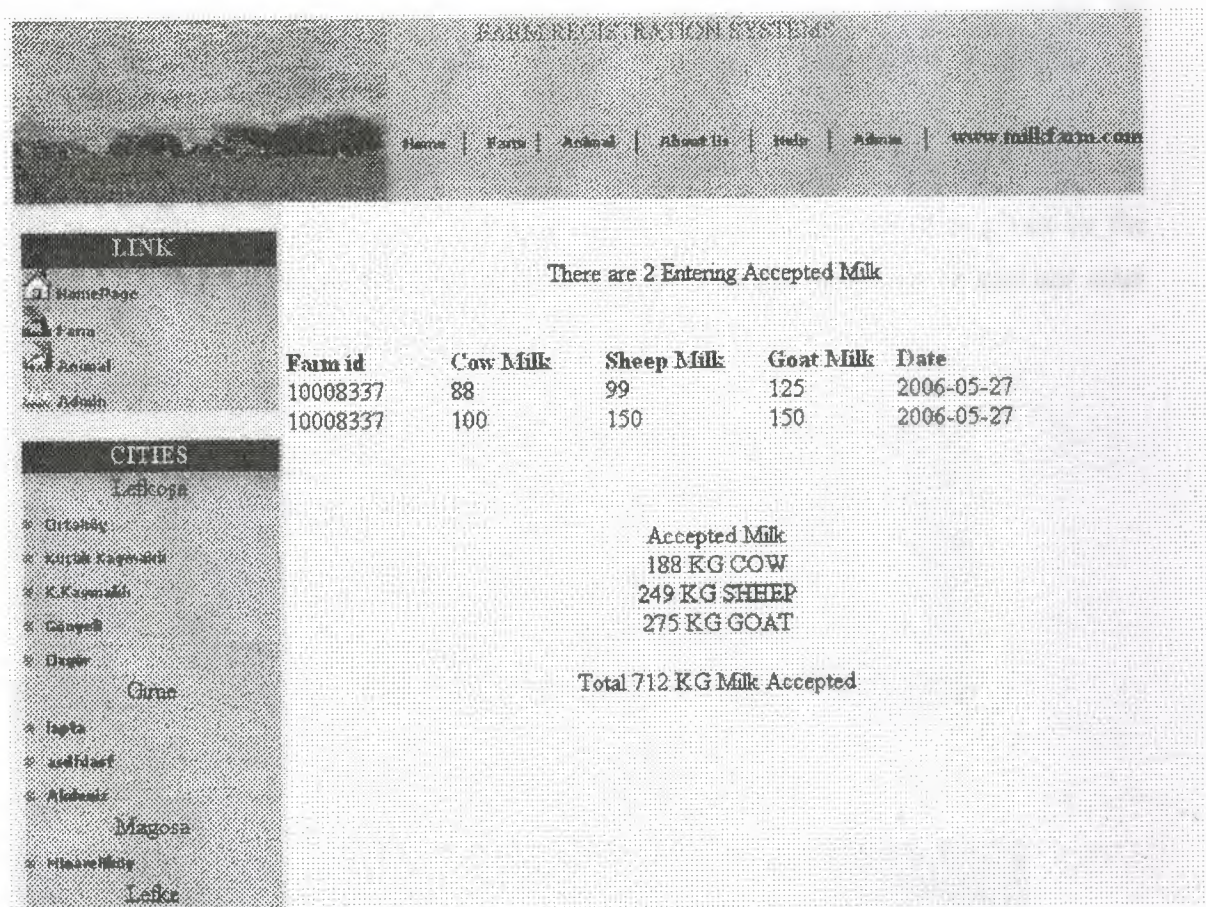
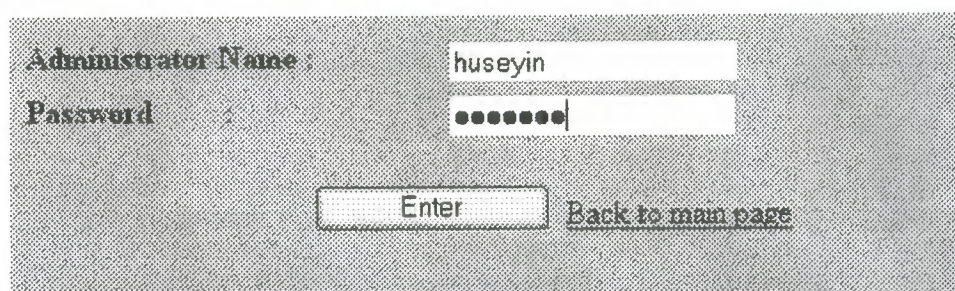


Figure 1.7

1.1.2 Admin Interface Area

Farmer could surf until this part. After this part there is **Admin** part. We can enter the admin part by two types of password. One of them is **Admin** password, the other one is **Farmer** password. When the user click the **Admin**, he will see figure 1.8. If the farmer enter the system by his password he can change just two things, as seen in the Figure 1.9 For example ID-10008337 – Hüseyin Merter want to enter the system, for this he must write his password to the admin part, this password is given by the Admin, I mean by the KTSUTEK Association, without the password he can not enter the system.

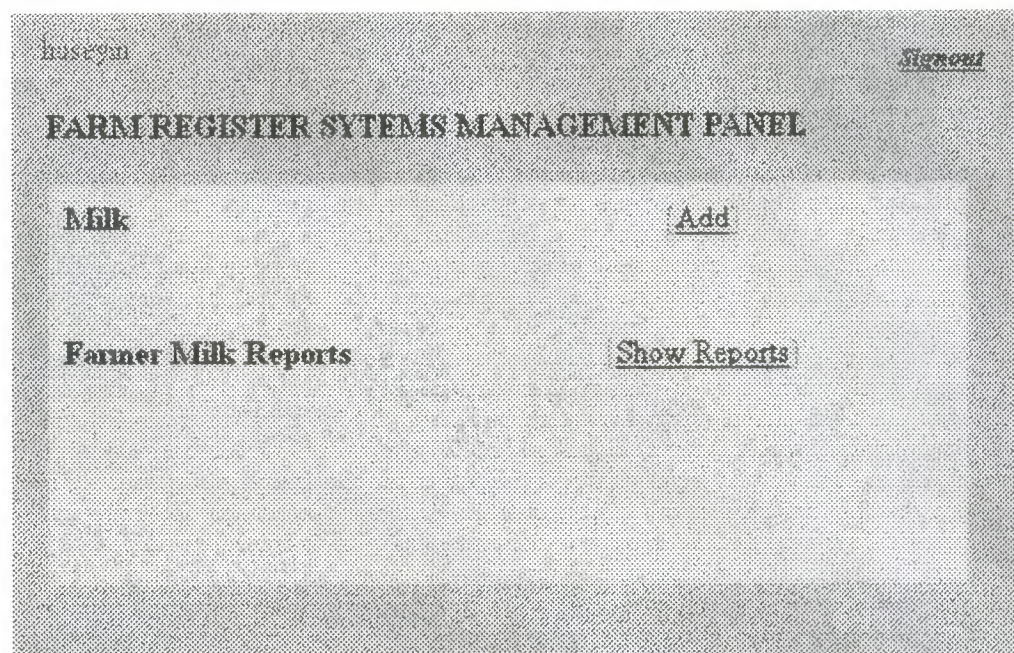


Administrator Name : huseyin

Password :

Enter Back to main page

Figure 1.8



huseyin Logout

FARM REGISTER SYTEMS MANAGEMENT PANEL

Milk Add

Farmer Milk Reports Show Reports

Figure 1.9

In the Figure 9, farmer can enter his production amount and until that time he can see the accepted milk amount by KTSUTEK, in here the farmer can enter the

amount of the milk by the kind of animal, and the price of the milk, these price is determined by the **Admin** part (KTSUTEK), if he click **Add** he can see Figure 10.

MILK PART [Logout](#)

Admin Name : huseyin

Farm Name:

Cow Milk: Enter

Sheep Milk: Enter

Goat Milk: Enter

Milk Price List of This Month

Cow :	4 YTL
Sheep :	3 YTL
Goat :	2 YTL

<< Main Page (admin)

Figure 1.10

When you enter in this page as **Admin**, you will have permission to modified Producer System Programming data. And you will see a page like the Figure 1.11 for this part you are admin and you can add, edit, search or delete, and also he created password for farmer or for his self, if he create a password for admin, he will see the Figure 1.12, to create the password for the farmer he must choose the farm name, as seen in Figure 1.13

FARM REGISTER SYTEMS MANAGEMENT PANEL

Region	Add	Edit	Delete
Farms	Add	Edit	Delete
Farmer	Add	Edit	Delete
Animal	Add	Search	Delete
Milk	Add		
Admin Manager	Add		
Milk Prices	Add		
Milk Accept	Accept		
Reports	Reports		

Figure 1.11**Admin Manage**

admin

Administrator Type	<input type="text" value="Select Admin Type"/>
Farm Name	<input type="text" value="Select a Farm"/> (if admin type is farmer)
Administrator Name	<input type="text"/>
Administrator Password	<input type="text"/>
Administrator Mail	<input type="text"/>
	<input type="button" value="Enter"/>

[<< Main Page \(admin\)](#)

Figure 1.12

The screenshot shows a web page titled "Admin Manage" with a "Logout" link in the top right corner. On the left, there is a vertical menu with the word "Admin" visible. The main content area contains a form with the following fields: "Administrator Type" (a dropdown menu currently showing "Administrator"), "Administrator Name" (a text input field), "Administrator Password" (a text input field), and "Administrator Mail" (a text input field). Below these fields is an "Enter" button. In the bottom right corner, there is a link that says "<< Main Page (admin)".

Figure1.13

If we click **Add** for the **Region** we can select the region and then we can add the location, as seen in the Figure 1.14

The screenshot shows a web page titled "REGION" with a "Logout" link in the top right corner. On the left, there is a vertical menu with the word "Region" visible. The main content area contains a form with the following fields: "Admin Name : admin" (a text input field), "Region" (a dropdown menu currently showing "Select a Region"), and "Location" (a text input field). Below these fields is an "Enter" button. In the bottom right corner, there is a link that says "<< Main Page (admin)".

Figure 1.14

If we click **Edit** for the **Region** we can select the region and then we will see the figure 15,

Then we can click any farm to update, for example if we click Ortaköy farm we will see the figure 16

The screenshot shows a web interface titled "REGION" with a "Logout" link in the top right corner. Below the title, it displays "Admin Name : admin". A "Region" label is followed by a dropdown menu currently showing "Lefkoşa". Below the dropdown, there is a list of five regions: 1 - Ortaköy, 2 - Küçük Kaymaklı, 3 - K. Kaymaklı, 4 - Gönyeli, and 5 - Ozgür. At the bottom right, there is a link that says "<< Main Page (admin)".

Figure 1.15

The screenshot shows the same "REGION" interface. The "Region" dropdown is now replaced by a text input field containing "Ortaköy". Below this input field is a button labeled "enter". The "Admin Name : admin" and "Logout" link remain in the same positions. At the bottom right, the link "<< Main Page (admin)" is still present.

Figure 1.16

If we click **Delete** for the **Region** we can select the region and then we will see the figure 1.15, in here we can delete any farm just by clicking the farm name.

The screenshot shows a web interface titled "FARM" in the top center. In the top right corner, there is a link labeled "Logout". Below the title, it says "Administraor : admin". The form contains the following elements:

- Region**: A dropdown menu with the text "Select a City" and a downward arrow.
- Location**: A dropdown menu with the text "Select a Location" and a downward arrow.
- Farm ID**: A text input field.
- Farm Name**: A text input field.
- Farm Address**: A text input field.
- Enter**: A button located below the Farm Address field.
- Navigation**: A link labeled "<< Main Page (admin)" in the bottom right corner.

Figure 1.17

If we click **Add** for the **Farms** we can select the region and the location then we can enter the Farm ID, Farm Name and Farm Address as seen in the Figure 1.17 in here we can **Edit** the farms information and also we can **Delete**.

FARMER[Logout](#)

Admin Name : admin

Farm Name

Select a Farm

Farmer ID

Farmer Name

Surname

Address

Telephone

Mobil Phone

Fax

E-mail

Enter

[<< Main Page \(admin\)](#)

Figure 1.18

If we click **Add** for the **Farmer** we must select the farm name then we can enter the information about the farmer, seen in the Figure 1.18. In here we can **Edit** the farmer information and also we can **Delete**.

The screenshot shows a web application titled "ANIMAL" with a "Logout" link in the top right corner. Below the title, the word "admin" is displayed. The main area contains four search filters: "Farm Name" with a dropdown menu showing "Select a Farm", "Cow ID" with a text input field, "Sheep ID" with a text input field, and "Goat ID" with a text input field. An "Enter" button is positioned below these fields. In the bottom right corner, there is a link that says "<< Main Page (admin)".

Figure 1.19

If we click **Add** for the **Animal** again we must select the farm name then we can enter the animal ID Number, seen in the Figure 1.19. In here we can **Search** the Animal by the ID number, and also we can **Delete** the animal by the ID.

CHAPTER TWO : WHAT IS THE HTML

2.1 What Is The Html

The basic language of the Internet is Hyper Text Markup Language (HTML). Unlike a true programming language, HTML doesn't work like an operating system and run your computer. Instead, it allows the author to "mark up" the contents of a document in order to change its visual appearance in a web browser. A browser takes the content as written in the HTML file and represents it on the screen of your computer.

HTML is designed to specify the logical organization of a document, with important hypertext extensions. It is not designed to be the language of a WYSIWYG word processor such as Word or WordPerfect. This choice was made because the same HTML document may be viewed by many different "browsers", of very different abilities. Thus, for example, HTML allows you to mark selections of text as titles or paragraphs, and then leaves the interpretation of these marked elements up to the browser. For example one browser may indent the beginning of a paragraph, while another may only leave a blank line.

HTML instructions divide the text of a document into blocks called elements. These can be divided into two broad categories -- those that define how the BODY of the document is to be displayed by the browser and those that define information 'about' the document, such as the title or relationships to other documents. The detailed rules for HTML (the names of the tags/elements, how they can be used) are defined using another language known as the standard generalized markup language, or SGML. SGML is wickedly difficult, and was designed for massive document collections, such as repair manuals for F-16 fighters, or maintenance plans for nuclear submarines. Fortunately, HTML is much simpler!

However, SGML has useful features that HTML lacks. For this reason, markup language and software experts have developed a new language, called XML (the extensible markup language) which has most of the most useful features of HTML and SGML.

2.1.1 HTML Tags

- HTML Tag `<html> </html>` : This tells the browser that the document is in HTML format.
- Head Tag `<head> </head>` : This section contains information about your site but is not visible in a browser window. This information is helpful when you want to use search engines to promote your site. The one tag you should always include is the `<title>` tag.
- Title Tag `<title> </title>` : This will display the title of your site in the top bar of the browser. You want to include this or your page will appear as an untitled document in the browser and in lists of bookmarks.
- Body tag `<body> </body>` : The visible content of the page must be placed between the `<body>` and `</body>` tags. In our example in above, an attribute of `<bgcolor="#ffffff">` was added to this tag to ensure that the page would have a white background.
- Paragraph tag `<p> </p>` : This tag tells the browser to skip a line and begin a new paragraph. This tag often appears without a corresponding closing tag. Although browsers will still represent your paragraphs correctly, including that closing tag will make possible transitions to future technologies easier.
- Linebreak tag `
` : Content following the `
` tag will appear on the next line.
- Heading tag `<h1> </h1>` : This tag indicates to the browser that you are using a headline. The range of values from `<h1>` to `<h6>` (largest to smallest) sets the size of the headline. Good practice indicates that you should use an `<h1>` tag only once on a web page, just as a newspaper uses the largest headline size only once on the front page. Text formatted with this tag will also be bold. However, do not use this tag if you simply want bold text. Use the `` tag instead.
- Bold tag ` ` : This will make the content between the tags appear bold.
- Italic tag `<i> </i>` : This italicizes the content between the tags.
- Image tag `` : This tag allows you to place graphics onto a web page. You will have to indicate the location of the graphic you want to use. This is done through the attribute `<src="filename">`. You will also want to indicate the

height and width of the graphic. This will allow the browser to continue loading other content on your page as the graphic is downloaded. Finally, use the `<alt>` attribute to describe the graphic. This will be helpful for those surfing with graphics turned off or for persons with visual disabilities using a text reader.

- Anchor tag `<a> ` : The anchor tag creates links to additional content elsewhere on the Web or elsewhere in your site.
- Font tag ` ` : This tag will allow you to change the size, color, and font of the text on your page. You can modify any of these properties by altering these attributes. If you wish to modify the size of the font to be one size larger than normal, you would use ``. While it is possible to set an absolute size, it is better to set a relative size instead such as `"-1"` or `"+2"`. This way you can make your page increasable for someone with a visual problem to read your content.
- Tables : Tables allows for better organization of larger amounts of information into a more cohesive structure.
- Table tag : The `<table>` and `</table>` tags go at the beginning and very end of the table.

2.1.2 What is IIS?

IIS (Microsoft Internet Information Services or Server) is a set of Internet based services for Windows machines. Originally supplied as part of the Option Pack for Windows NT, they were subsequently integrated with Windows 2000 and Windows Server 2003. The current (Windows 2003) version is IIS 6.0 and includes servers for FTP, SMTP, NNTP and HTTP/HTTPS. Earlier versions also included a Gopher server.

The web server itself can not directly perform server side processing but can delegate the task to ISAPI applications on the server. Microsoft provides a number of these, including ones for Active Server Pages and ASP.NET. Third parties have provided support for PHP and Perl languages in the same way.

Internet Information Services is designed to run on Windows server operating systems. A restricted version that supports one web site and a limited number of connections is also supplied with Windows XP Professional.

IIS has been attributed with a number of security exploits, most of which were in fact issues within the lesser used ISAPI handlers. With Windows Server 2003 Microsoft finally elected to turn off all ISAPI handlers by default thereby giving the web server a much more secure "out of the box" configuration.

Microsoft has also changed the server account that IIS runs on. In versions of IIS before 6.0, all the features were run on the System account, allowing exploits to run wild on the system. Under 6.0 many of the processes have been brought under a Network Services account which has fewer privileges. In particular this means that if there is an exploit on that feature, it wouldn't necessarily compromise the entire system.

Apache is the dominant software in the web server market and IIS's main competitor. Solaris Operating Environment/J2EE also competes in the enterprise web services arena.

2.1.3 World-Wide Web

The Web incorporated the capabilities of most earlier tools, and added the ability to handle various media types in a much more usable graphical hypertext environment. Interfaces for the Web exist on all computer platforms, and are easy to use so much so that Web usage has exploded in the past year, and the amount of academic and commercial information available is increasing at an almost exponential rate. The amount of information available from a desktop computer provides incredible support for research on a variety of topics, including distance education.

2.1.4 SLIP and PPP

These acronyms refer to two standard protocols for utilizing a graphical interface to the Internet through a regular telephone line and modem. Previously, users had to travel to a central location if they wanted to use tools which relied on graphics technologies much like the similar need required to utilize satellite conferencing technology.

2.1.5 Advanced Conferencing Software

Conferencing software is still very much in the early stages on the Internet. NetPhone from Electric Magic permits real-time transfer of audio across the Internet, much like a telephone call. Some users in Hong Kong are now experimenting to see what other components they would need in order to listen to the U.S. National Public Radio broadcasts through the Internet, as they are not available locally. CU-SeeMe is an experimental one-many and/or many-many video repeater. It is an experimental tool using standard network protocols to provide video feeds to anyone who wants one. Videoconferencing from the desktop will be feasible once more network lines have been upgraded to higher signal capacity. Where fiber-optic cable has been installed, the capacity is already far beyond that provided by satellite. Another drawback to computer solutions is the limited image resolution currently available. Now the hardware and software need to take the next step into utility and viewability.

2.1.6 Proprietary Software vs. Open Standards

Many information service companies are offering purpose-built proprietary software for specialized applications on the Internet. A common example is Ameritech's Advanced Video Service, which provides full-motion videoconferencing over standard dial-up lines. Those which fail to incorporate the ability to utilize standard network communication protocols will achieve very limited market penetration. Tremendous amounts of information on every subject imaginable are being provided on servers throughout the Internet. They are following standard protocols, which are constantly under creation or revision. Software which is not able to speak to or query these resources is crippled when it comes to actual utility.

2.2 HTTP Authentication With PHP

The HTTP Authentication hooks in PHP are only available when it is running as an Apache module and is hence not available in the CGI version. In an Apache module PHP script, it is possible to use the `header()` function to send an "Authentication Required" message to the client browser causing it to pop up a Username/Password input window. Once the user has filled in a username and a password, the URL containing the PHP script will be called again with the variables, `$PHP_AUTH_USER`,

`$PHP_AUTH_PW` and `$PHP_AUTH_TYPE` set to the user name, password and authentication type respectively. Only "Basic" authentication is supported at this point. See the `header()` function for more information. An example script fragment which would force client authentication on a page would be the following:

Example 2.2.1. HTTP Authentication example

```
<?php
if (!isset($PHP_AUTH_USER)) {
    header("WWW-Authenticate: Basic realm=\"My Realm\"");
    header("HTTP/1.0 401 Unauthorized");
    echo "Text to send if user hits Cancel button\n";
    exit;
} else {
    echo "<p>Hello $PHP_AUTH_USER.</p>";
    echo "<p>You entered $PHP_AUTH_PW as your password.</p>";
}
?>
```

Instead of simply printing out the `$PHP_AUTH_USER` and `$PHP_AUTH_PW`, you would probably want to check the username and password for validity. Perhaps by sending a query to a database, or by looking up the user in a dbm file.

In order to prevent someone from writing a script which reveals the password for a page that was authenticated through a traditional external mechanism, the `PHP_AUTH` variables will not be set if external authentication is enabled for that particular page. In this case, the `$REMOTE_USER` variable can be used to identify the externally-authenticated user.

Configuration Note: PHP uses the presence of an `AuthType` directive to determine whether external authentication is in effect. Remember to avoid this directive for the context where you want to use PHP authentication (otherwise each authentication attempt will fail).

Note, however, that the above does not prevent someone who controls a non-authenticated URL from stealing passwords from authenticated URLs on the same server.

Both Netscape Navigator and Internet Explorer will clear the local browser window's authentication cache for the realm upon receiving a server response of 401. This can

effectively "log out" a user, forcing them to re-enter their username and password.

Some

people use this to "time out" logins, or provide a "log-out" button.

Example 2.8.2 HTTP Authentication example forcing a new name/password

```
<?php
function authenticate() {
    header( "WWW-Authenticate: Basic realm=\"Test Authentication System\"");
    header( "HTTP/1.0 401 Unauthorized");
    echo "You must enter a valid login ID and password to access this
resource\n";
    exit;
}
if (!isset($PHP_AUTH_USER) || ($SeenBefore == 1 && !strcmp($OldAuth,
$PHP_AUTH_USER))) {
    authenticate();
}
else {
    echo "<p>Welcome: $PHP_AUTH_USER<br>";
    echo "Old: $OldAuth";
    echo "<form action='$PHP_SELF' METHOD='POST'>\n";
    echo "<input type='hidden' name='SeenBefore' value='1'>\n";
    echo "<input type='hidden' name='OldAuth' value='$PHP_AUTH_USER'>\n";
    echo "<input type='submit' value='Re Authenticate'>\n";
    echo "</form></p>\n";
}
?>
```

This behavior is not required by the HTTP Basic authentication standard, so you should never depend on this. Testing with Lynx has shown that Lynx does not clear the authentication credentials with a 401 server response, so pressing back and then forward again will open the resource as long as the credential requirements haven't changed. The user can press the '_' key to clear their authentication information, however.

Also note that this does not work using Microsoft's IIS server and the CGI version of PHP due to a limitation of IIS.

CHAPTER THREE : WHAT IS MYSQL

3.1 Understanding MySQL

MySQL, pronounced "my Ess Que El," is an open source, Enterprise-level, multi-threaded, relational database management system. That sounds like a lot of sales or marketing hype, but it truly defines MySQL. You may not be familiar with some of these terms but, by the end of today, you will be.

MySQL was developed by a consulting firm in Sweden called TcX. They were in need of a database system that was extremely fast and flexible. Unfortunately (or fortunately, depending on your point of view), they could not find anything on the market that could do what they wanted. So, they created MySQL, which is loosely based on another database management system called mSQL. The product they created is fast, reliable, and extremely flexible. It is used in many places throughout the world.

Universities, Internet service providers and nonprofit organizations are the main users of MySQL, mainly because of its price (it is mostly free). Lately, however, it has begun to permeate the business world as a reliable and fast database system. Some examples of commercial use are available on the CD-ROM that accompanies this book.

The reason for the growth of MySQL's popularity is the advent of the Open Source Movement in the computer industry. The Open Source Movement, in case you haven't heard about it, is the result of several computer software vendors providing not only a product but the source code as well. This allows consumers to see how their program operates and modify it where they see fit. This, and the popularity of Linux, has given rise the use of open source products in the business world. Because of Linux's skyrocketing popularity, users are looking for products that will run on this platform. MySQL is one of those products.

MySQL is often confused with SQL, the structured query language developed by IBM. It is not a form of this language but a database system that uses SQL to manipulate, create, and show data. MySQL is a program that manages databases, much like Microsoft's Excel manages spreadsheets. SQL is a programming language that is used by MySQL to accomplish tasks within a database, just as Excel uses VBA (Visual Basic for Applications) to handle tasks with spreadsheets and workbooks. Other programs that manage databases include Microsoft's SQL Server, Sybase Adaptive Server, and DB2.

Now that you know where MySQL came from, look at what it is. To begin with, start with the term database. What is a database? You have probably used one in your lifetime. If you've ever bought anything over the Internet or have a driver's license, you can be assured that you have used one.

A database is a series of structured files on a computer that are organized in a highly efficient manner. These files can store tons of information that can be manipulated and called on when needed. A database is organized in the following hierarchical manner, from the top down. You start with a database that contains a number of tables. Each table is made up of a series of columns.

Data is stored in rows, and the place where each row intersects a column is known as a field. **Figure 3.1** depicts this breakdown. For example, at your favorite online book store there is a database. This database is made up of many tables. Each table contains specific, common data. You would probably see an Authors table or a Books table. These tables are made up of named columns that tell what data is contained in them. When a record is inserted into a table, a row of data has been created. Where a row and a column intersect, a field is created. This how databases are broken down.

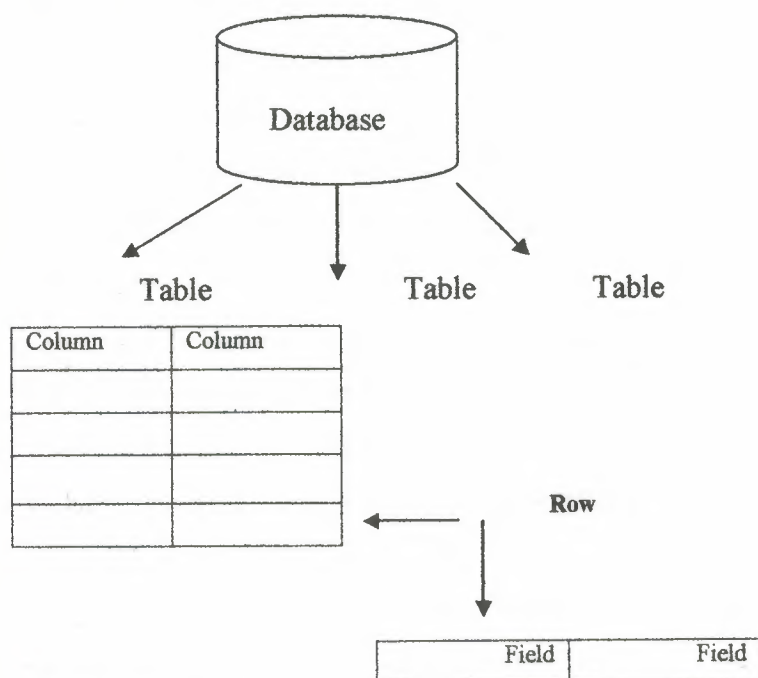


Figure 3.1 The anatomy of a database.

MySQL is more than just a database. It is a system that manages databases. It controls who can use them and how they are manipulated. It logs actions and runs

continuously in the background. This is different from what you may be used to. Most people think about Microsoft Access or Lotus Approach when they think about databases. These are databases, but they are not management systems.

A DBMS can contain many databases. Users connect to the database server and issue requests. The database server queries its databases and returns the requests to the issuers. Databases, such as Approach and Access, are a step down from this type of system. They share their files with multiple users, but there is no interface controlling the connections or answering requests.

There are many uses for a DBMS such as MySQL. Uses can range from help desk systems to Web site applications. The important thing to remember is that MySQL is large enough and quick enough to function in almost any situation. Where it finds itself most comfortable is the Enterprise.

3.1.1 What Is the Enterprise?

The Enterprise is not a starship or a space shuttle. The *Enterprise* is the area in the business world where many large systems interact with one another to accomplish a common goal. Some applications that are at this level of business include SAP, Microsoft SQL Server, Oracle 8i, and Sybase Adaptive Server.

The computer applications that exist at this level of business tend to have certain characteristics. They are usually multiuser in nature—many people can use the same application at the same time. Another characteristic is that they provide some sort of security mechanism. The final characteristic is that applications at this level have to be very flexible.

The first characteristic of an Enterprise-level application is that it can be used by more than one person at a time. This is a requirement at this level of business. More than one person may need to have access to business information at a given time. This is critical for the business to function successfully.

MySQL meets this requirement. It can have up to 101 simultaneous connections. This doesn't mean that only 101 people can use this application. It means it can have 101 connections going on at the same time—which is a little different. A connection is the time it takes for a user to receive the data that he or she has requested. In the case of MySQL, this is hardly any time at all. Most database systems in the same

class as MySQL allow fewer simultaneous connections. Currently, the only DBMS to offer more connections is Microsoft SQL Server.

The next characteristic that an Enterprise-level application must have is security. When dealing with mission-critical information, only people with the need to know should be allowed to view it. Security keeps malicious people at bay; without it, disasters can happen. MySQL meets this requirement.

The security in MySQL is unparalleled. Access to a MySQL database can be determined from the remote machine that can control which user can view a table. The database can be locked down even further by having the operating system play a role in security as well. Very few databases in the same class as MySQL can compare to the level of security that MySQL provides.

3.1.2. How hard is it to change?

MySQL answers these questions very well. It is extremely flexible and easy to use. MySQL can run on almost any platform. If a new CIO wants to change from Windows NT to Linux, fine—MySQL can adapt. MySQL also comes with the source code. If there are any deep-level changes that you need to make, you can edit the source and make these changes yourself.

If MySQL is missing a feature that you can't live without, just add it yourself. No other database on the market can offer you that kind of flexibility. MySQL also has several application-level interfaces in a variety of languages. If yours is mainly a Microsoft shop, you can use ODBC to interact with MySQL. If your company is a UNIX shop, you can use C, Perl, or JDBC. There is no end to the flexibility that MySQL has to offer.

In addition to the previously discussed characteristics, databases at the Enterprise level must be able to work together. Data warehousing is a technique that combines all the data in a business. Because of the flexibility and speed that MySQL has to offer, it can work well in any situation.

The Internet has also become a piece of the Enterprise pie. No large corporation is without an Internet presence. These corporations need databases to sell and compete at this level of business.

MySQL works well as an Internet-based database server. It has been proven in this arena and is the preferred database of many Internet service providers. Because of its speed and multiple application interfaces, MySQL is an ideal choice.

Enterprise applications are the crucial component to a business's decision-making power. Information must be timely and accurate for a business to perform effectively. To do this, applications must work quickly. An application is much like a car. It can look pretty on the outside, but the engine is what gives it its power. The same applies to an application; If its database engine is weak, so is the application. MySQL is clearly the choice for the Enterprise.

3.1.3. What Is a Relational Database?

A *relational database*, simply defined, is a database that is made up of tables and columns that relate to one another. These relationships are based on a key value that is contained in a column. For example, you could have a table called Orders that contains all the information that is required to process an order, such as the order number, date the item was ordered, and the date the item was shipped. You could also have a table called Customers that contains all the data that pertains to customers, such as a name and address. These two tables could be related to each other.

The relational database model was developed by E.F. Codd back in the early 1970s. He proposed that a database should consist of data stored in columns and tables that could be related to each other. This kind of thinking was very different from the hierarchical file system that was used at the time. His thinking truly revolutionized the way databases are created and used.

A relational database is very intuitive. It mimics the way people think. People tend to group similar objects together and break down complex objects into simpler ones. Relational databases are true to this nature. Because they mimic the way you think, they are easy to use and learn. In later days, you will discover how easy a relational database is to design and learn.

Most modern databases use a relational model to accomplish their tasks. MySQL is no different. It truly conforms to the relational model. This further adds to the ease of use of MySQL.

3.2 Designing Databases

The first step is always to create the database, unless you want to use an existing third party's one. When a database is created, it is assigned to an owner, who executed the creation statement. Usually, only the owner (or a superuser) can do anything with the objects in that database, and in order to allow other users to use it,

privileges must be granted. Applications should never connect to the database as its owner or a superuser, because these users can execute any query at will, for example, modifying the schema (e.g. dropping tables) or deleting its entire content.

You may create different database users for every aspect of your application with very limited rights to database objects. The most required privileges should be granted only, and avoid that the same user can interact with the database in different use cases. This means that if intruders gain access to your database using one of these credentials, they can only effect as many changes as your application can.

You are encouraged not to implement all the business logic in the web application (i.e. your script), instead to do it in the database schema using views, triggers or rules. If the system evolves, new ports will be intended to open to the database, and you have to reimplement the logic in each separate database client. Over and above, triggers can be used to transparently and automatically handle fields, which often provides insight when debugging problems with your application or tracing back transactions.

3.2.1 Connecting to Database

You may want to establish the connections over SSL to encrypt client/server communications for increased security, or you can use ssh to encrypt the network connection between clients and the database server. If either of them is done, then monitoring your traffic and gaining informations in this way will be a hard work.

3.2.2 Encrypted Storage Model

SSL/SSH protects data travelling from the client to the server, SSL/SSH does not protect the persistent data stored in a database. SSL is an on-the-wire protocol.

Once an attacker gains access to your database directly (bypassing the webserver), the stored sensitive data may be exposed or misused, unless the information is protected by the database itself. Encrypting the data is a good way to mitigate this threat, but very few databases offer this type of data encryption.

The easiest way to work around this problem is to first create your own encryption package, and then use it from within your PHP scripts. PHP can assist you in this case with its several extensions, such as Mcrypt and Mhash, covering a wide variety of encryption algorithms. The script encrypts the data be stored first, and decrypts it when retrieving. See the references for further examples how encryption works.

In case of truly hidden data, if its raw representation is not needed (i.e. not be displayed), hashing may be also taken into consideration. The well-known example for the hashing is storing the MD5 hash of a password in a database, instead of the password itself. See also `crypt()` and `md5()`.

3.2.3 The Client/Server Paradigm

The client/server paradigm or model has been around a lot longer than most people think. If you look back to the early days of programming, you remember or have heard or read about the large mainframe computer with many smaller "dumb" terminals. These terminals were called dumb for a reason. No logic or processing was done at the terminals. They were just receptacles for the output of the mainframe. This was the dawn of the client/server age, but the term client/server wasn't the buzzword it is today.

As the personal computer became more prevalent, giving rise to the local area network (LAN), the client/server model evolved. Now processing could be done at the client. Clients started sharing data. This data was stored in sharable computers called file servers. Now, instead of all the processing being done at the server, it was all being done at the client. The server or centralized computer was just a large storage device. It did little or no processing—a complete reversal of earlier thinking. After a couple of years, desktop applications became more powerful. People needed to share more information more quickly. This gave rise to the more powerful server machines. These machines answered requests from clients and processed them. These servers are what you know today as database servers, Web servers, and file servers. This is when people started calling it client/server computing. It is basically a two-tier design; a client issues requests, and a server answers them. All the business logic is at the application level on the client. Two-tier design is still very prevalent today. This is also known as a *fat client* because all the application processing is done at the client level. After a couple of years, servers became the powerhouses of business organizations because of their duties. They were usually top-of-the-line systems with the best hardware and were tweaked for speed. So, it was just a matter of time before someone came up with the idea of moving the guts of their programs to the server. The client would just be a graphical user interface (GUI) and the main application or business logic would be processed on the server. The server would then make the necessary calls to other servers, such as database servers or file servers, as needed. This gave birth to the three-tier or *thin client* design. In this design, all processing of the business logic is done at the server level. This allows the

more powerful machine to handle the logic and the slower machines to display the output. Does this sound familiar? It should—we've come full circle. The heavy processing is again done on the more powerful, centralized machines, while all the client machines do is display the output.

The Internet is a prime example of thin client architecture. A very thin client—the browser—sends requests to a Web server, which sends a response back to the browser. The browser then displays the requested information—completely full circle.

Again, we are on the verge of a new era in computing. Applications are becoming more balanced across the network. Because of a decline in computer prices, very good machines are showing up on the desktop as clients. This allows applications to pick up the slack and perform some processing. Server applications are becoming more advanced as well. You can now run functions remotely and accomplish distributed computing fairly easily. These advancements allow your applications to be more robust in nature and more useful to your business.

3.2.4 History of Mysql

We started out with the intention of using the mSQL database system to connect to our tables using our own fast low-level (ISAM) routines. However, after some testing, we came to the conclusion that mSQL was not fast enough or flexible enough for our needs. This resulted in a new SQL interface to our database but with almost the same API interface as mSQL. This API was designed to allow third-party code that was written for use with mSQL to be ported easily for use with MySQL.

The derivation of the name MySQL is not clear. Our base directory and a large number of our libraries and tools have had the prefix “my” for well over 10 years. However, co-founder Monty Widenius's daughter is also named My. Which of the two gave its name to MySQL is still a mystery, even for us.

The name of the MySQL Dolphin (our logo) is “Sakila,” which was chosen by the founders of MySQL AB from a huge list of names suggested by users in our “Name the Dolphin” contest. The winning name was submitted by Ambrose Twebaze, an Open Source software developer from Swaziland, Africa. According to Ambrose, the feminine name Sakila has its roots in SiSwati, the local language of Swaziland. Sakila is

also the name of a town in Arusha, Tanzania, near Ambrose's country of origin, Uganda.

3.2.5 How Large MySQL Tables Can Be

MySQL 3.22 had a 4GB (4 gigabyte) limit on table size. With the MyISAM storage engine in MySQL 3.23, the maximum table size was increased to 65536 terabytes (2567 – 1 bytes). With this larger allowed table size, the maximum effective table size for MySQL databases is usually determined by operating system constraints on file sizes, not by MySQL internal limits.

The InnoDB storage engine maintains InnoDB tables within a tablespace that can be created from several files. This allows a table to exceed the maximum individual file size. The tablespace can include raw disk partitions, which allows extremely large tables. The maximum tablespace size is 64TB.

3.2.6 Features of MySQL

MySQL is a full-featured relational database management system. It is very stable and has proven itself over time. MySQL has been in production for over 10 years. MySQL is a multithreaded server. *Multithreaded* means that every time someone establishes a connection with the server, the server program creates a thread or process to handle that client's requests. This makes for an extremely fast server. In effect, every client who connects to a MySQL server gets his or her own thread.

MySQL is also fully ANSI SQL92-compliant. It adheres to all the standards set forth by the American National Standards Institute. The developers at TcX take these standards seriously and have carefully adhered to them.

Note: ANSI SQL92 is a set of standards for the Structured Query Language that was agreed on in 1992 by the American National Standards Institute. Another valuable feature of MySQL is its online help system. All commands for MySQL are given at a command prompt. To see which arguments the commands take or what the utility or command does, all you have to do is type the command and include the -help or -? switch. This will display a slew of information about the command.

Yet another feature of MySQL is its portability—it has been ported to almost every platform. This means that you don't have to change your main platform to take advantage of MySQL. And if you do want to switch, there is probably a MySQL port for your new platform.

MySQL also has many different application programming interfaces (APIs). They include APIs for Perl, TCL, Python, C/C++, Java (JDBC), and ODBC. So no matter what your company's expertise is, MySQL has a way for you to access it.

MySQL is also very cheap. For an unlicensed, full version of MySQL, the cost is nothing. To license your copy will currently cost you \$200. This is an incredible deal, considering what you are getting for your money. Database systems that provide half the features that MySQL has can cost tens of thousands of dollars. MySQL can do what they do better and for less.

3.3 Installation MySQL for Windows

The Windows is very similar to a Linux installation. Select a mirror site from which to download your Windows binary. The Windows executable is a self-installing WinZip file. After the download has finished, double-clicking the zipped file will begin the extraction/installation routine. After it has installed and performed cleanup, you will be returned to your desktop. To see the new files, open Windows Explorer and navigate to the C:\ directory. You should see something similar to Figure 3.2.

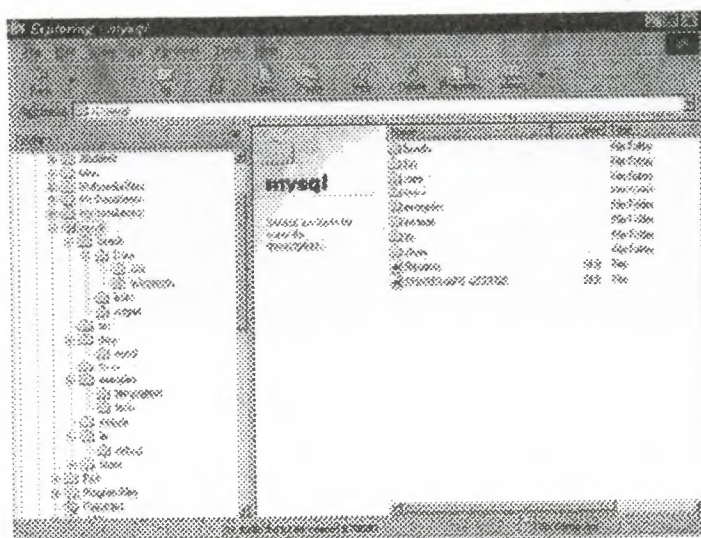


Figure 3.2 New My SQL Windows Installation

3.3.1. File Contents

Before continuing, examine what you have. The ChangeLog is a file containing all the changes and fixes for that particular release. INSTALL-BINARY is an information file explaining how to install MySQL and release notes specific to the

platform that is using this binary distribution. PUBLIC is the copyright and right to use license. The README file contains useful information on setting up your MySQL server. The support-files directory contains files to aid in configuring your MySQL server. It also contains a specification file that explains why TcX created MySQL.

The bin directory contains all the MySQL commands, such as mysqladmin, mysqlaccess, and several other important commands. You will study this in depth in later chapters. In the Windows version, there are a couple of extra commands. The MySQLManager (see Figure 3.3) is the GUI (graphical user interface) of MySQL. It shows the current databases and table structure and allows the user to run queries, all within a friendly graphical environment. As with any GUI, some power has been sacrificed for looks. Not all of the features available from the command line are represented. GUIs are nice, but to know the tool, you must know how to use the command line.

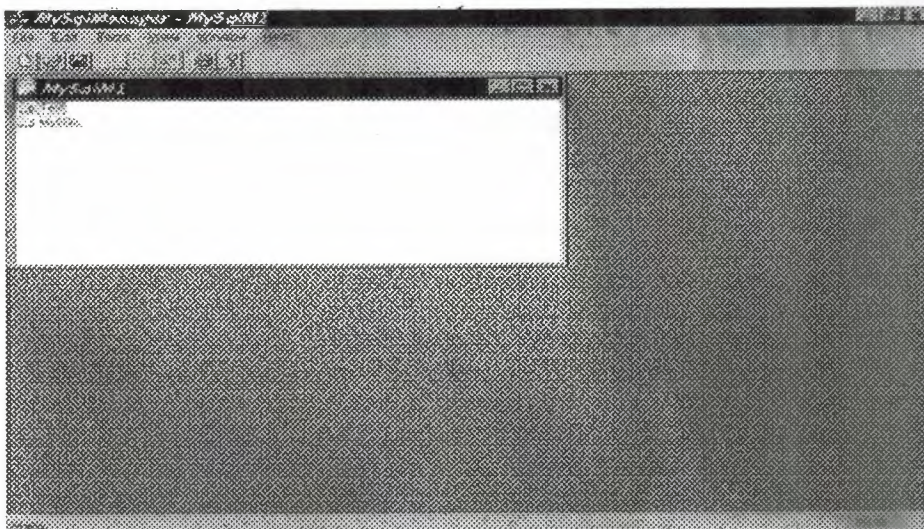


Figure 3.3 MySQLManager

The configure file contains a script that sets up the grant tables in your MySQL server. The data directory is where all the data for the databases is stored. Each database has its corresponding directory, as well as files that store the data, configure the tables, and provide a way to access the data stored in the files.

The include directory contains all the C header files. These are used with the C/C++ API. These files are also used in the MyODBC driver. The tests directory contains several Perl scripts to test your MySQL server. The lib directory contains the libraries used in the C++ API. The manual.txt, manual.htm, and manual_toc.htm are

probably the most important tools for a MySQL administrator/developer after the data files. They provide a wealth of information that is invaluable. The `mysql_for_dummies` file is a good starting place for the MySQL newbie. The `scripts` directory contains the install script for MySQL. It is called by the `configure` command.

The `share` directory contains the error logs and messages. The `mysql_bench` directory contains the `crash_me` tool. This tool is used to generate comparisons between database systems. The MySQL Web site also contains the benchmarks and comparison information.

3.3.2. Changing Passwords

Now that the directory structure is created and the files have been extracted, you can begin configuring and running MySQL. To create the grant tables, make sure you're in the `/usr/local/mysql` directory and type the following from the command line:

scripts/mysql_install_db

You will see a flurry of screen output. The script is creating the grant table for MySQL. This determines who can connect to the database. It is a good idea to change the root password of your MySQL database now. The database privileges and the file system privileges are two different things. This means that if you have a system user, named Mike, you do not have a database user named Mike unless you create one. MySQL's security operates independently of the system's security. now, choose a password for root.

From the command line, type the following—where *newpassword* is your new password: **bin/mysqladmin -password *newpassword*** This changes the current password (which is empty) to the new password. Security is a high priority in any environment, especially when dealing with the Internet. Make sure that you change your password; if you don't, you are opening the door for anyone to have his or her way with your data.

3.3.3 Starting and Stopping the Server

Like most Database Management Systems (DBMS) in its class, MySQL runs as a service or daemon. A *service* or *daemon* is a program that runs continuously in the background. Generally it doesn't have a user interface and cannot be seen unless you do a `ps` in Linux or look in the Task Manager of Windows (see Figure 3.4). `mysqld` is a

server program, which means that its entire purpose is to wait for someone to connect to it and issue a request, and then it responds to that request.

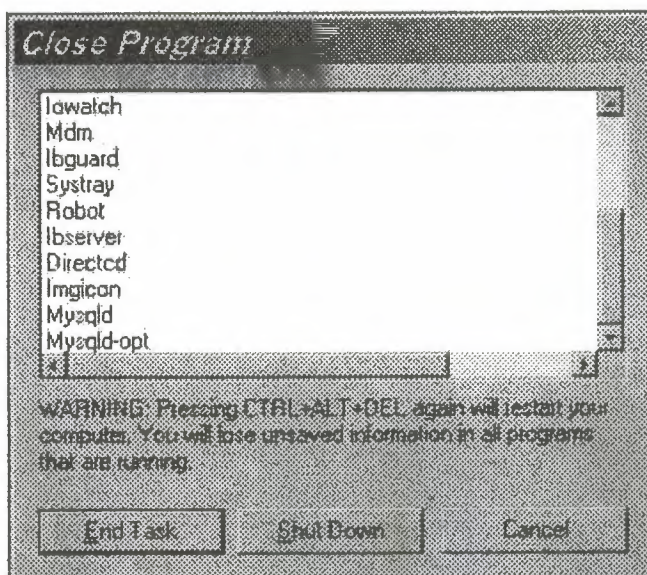


Figure 3.4 `mysqld` running as a background process.

You can think of the server program as an information desk in a mall. It sits there and has no other purpose but to answer questions from inquisitive shoppers. The people who work at the desk and respond to the shopper's requests don't store all the answers in their heads. They look up information from available resources. This is much like the `mysqld` daemon. It sits there—after it is started—waiting for requests. Depending on the type of request, it will either answer it directly or use the database that was asked for in the request. This is what makes an enterprise-level database different from a desktop application.

3.3.4 Using MySQL with Windows

MySQL server runs as a background process. To start the server on a Windows platform, double-click the `mysqld.exe` file in the `\mysql\bin` directory. This will start the process.

To stop the service, run the `mysqladmin shutdown` from the DOS prompt. This will take the server process down gracefully. The more drastic, possibly harmful way is to use the Task Manager to shut the process down. This could cause data corruption and is not recommended. It is also worth mentioning that, unlike the Linux daemon, the Windows process has a small memory leak. After time, this small leak will cause

problems. The system will lock up and programs will not run. To fix this problem, reboot the machine every couple of weeks. According to the documentation, the TcX development team has fixed this problem and it will be corrected in the next release. To have Windows NT start this process automatically, go to the Control Panel and click Services. You should see the mysqld service. Check it to start automatically. On Windows 95/98, you must place the mysqld.exe command in the Startup directory.

3.3.5 Using MySQL—the Command Line

Now that you have MySQL up and running, it is time to take your database engine for a spin. The command line interface of MySQL can be daunting the first time, especially if you're used to the GUIs that the other databases in the same class of MySQL offer.

To start the command line interface in Linux or in Windows, make sure you are in the /mysql directory. In Windows, you must use a DOS prompt. At the command line, type the following: **bin/mysql -p**

You should see the following:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 3 to server version : 3.22.23

Type **help** for help.

After the output is displayed, you are left with an empty prompt (see Figure 3.5). This is where it can get a little scary for the first time MySQL user. For the most part, all commands to manipulate your data are entered here. A good command of the Structured Query Language (SQL) is a must to get around. This can be a bane or a blessing, depending on how you look at it.

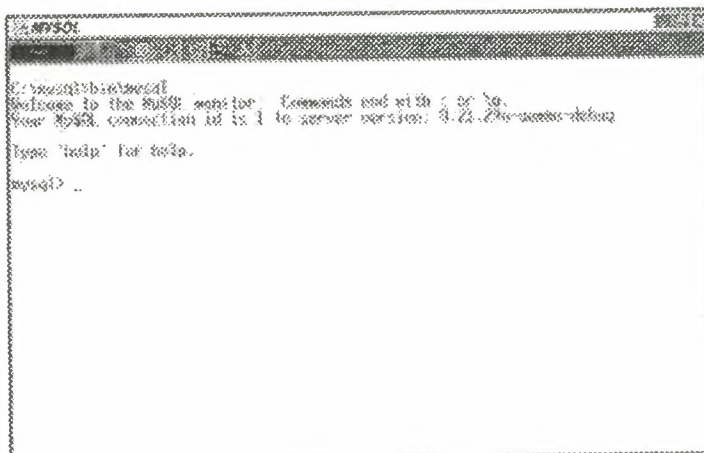


Figure 3.5 The MySQL monitor.

With the command line prompt, you don't need a GUI. You may argue that it is nice to have a GUI; it makes things easier. You can, with a few clicks of the mouse, see all your tables and how they relate to each other. With a GUI, you can see permissions and active connections. You can do a lot with a GUI, but what you can't do is remotely administer your database quickly and efficiently. Most Windows administrators use PC AnyWhere or some other similar product to administer the servers under their control. Though these programs are nice, they leave a lot to be desired, especially over slow dial-up connections. With the command prompt, these inefficiencies are no longer a problem. With a simple Telnet session, you can remotely administer you database quickly and efficiently. With a command prompt, you can create, drop, and populate a database as if you were there. It only takes a few late night calls to fix a problem to fully realize and understand the power that is afforded with this simple feature.

Assuming that you logged in as root, take a look at the existing databases in the default installation. To see what databases exist on this server, type the following:

show databases; You should see output similar to that shown in **Figure 3.6**.

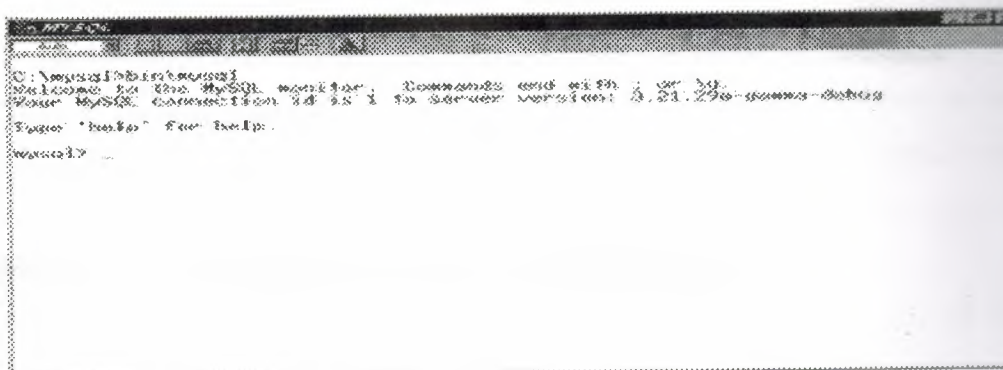


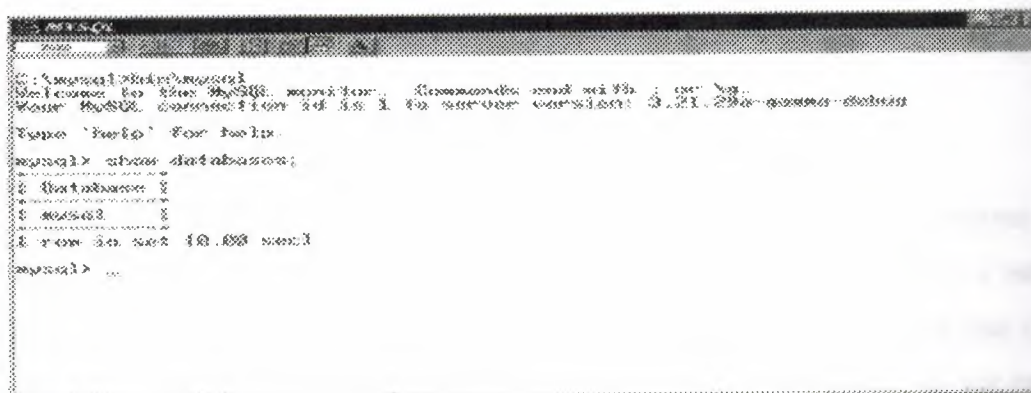
Figure 3.6 Viewing existing databases in the MySQL monitor.

To commit the action or SQL statement you have typed, you must end your line with either a semicolon (;) or a \g. This tells the MySQL monitor that you have finished your command and are ready to execute it. Simply pressing the Return or Enter key causes a line feed. This allows you to enter a long SQL string legibly. Another nice feature is the recall button. For Linux users, the up arrow will recall the last lines you have typed, just like the system. It uses a different history file than the operating system, so the only commands that are recalled are the commands that were typed at the MySQL prompt. For Windows users, well, we're just out of luck. There is no history recall key (not even the F3 key—the normal recall key for DOS). The recall key is

extremely convenient, especially when you make an error in a long SQL query. There isn't a lot of re-typing. To work with a listed database, you must tell the MySQL monitor which one to use. The command is simple enough. Type the following to use the mysql database:

3.2.6. Use MySQL

You should see output similar to that shown in **Figure 3.7**



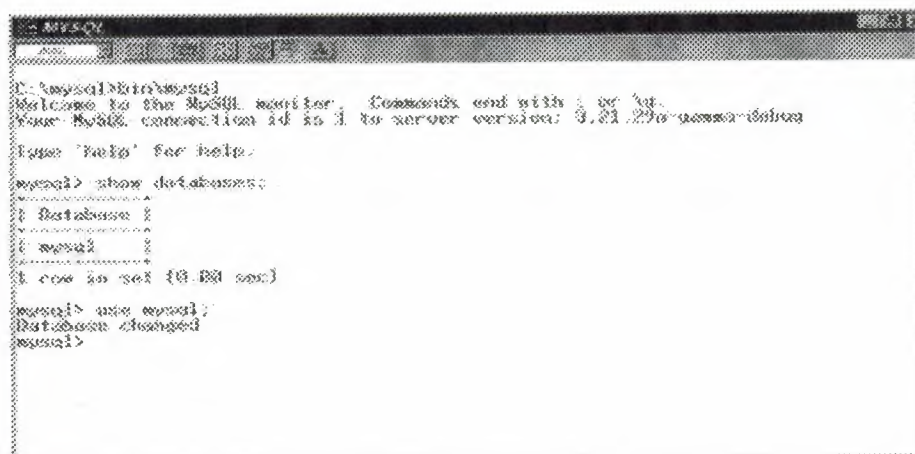
```
C:\mysql> mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.24a-secure
Type 'help;' for help;
mysql> use mysql;
Database
mysql
1 row in set (0.00 sec)
mysql>
```

Figure 3.7 Results of the USE command.

To see the structure or schema of a database, issue the following command:

show tables from mysql;

The output should resemble **Figure 3.8**.



```
C:\mysql> mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.24a-secure
Type 'help;' for help;
mysql> show databases;
Database
mysql
1 row in set (0.00 sec)
mysql> use mysql;
Database changed
mysql>
```

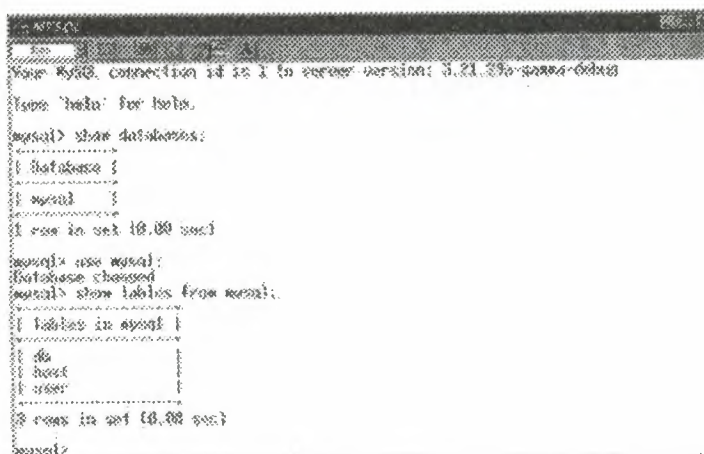
Figure 3.8

This simple command provides a listing of all the tables of the selected database. The following

command will show a detailed listing of the columns of your database.

Show Columns From User;

Again, your output should look like **Figure 3.9**.



```
mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
+-----+
1 row in set (0.00 sec)

mysql> use mysql;
Database changed
mysql> show tables;
+-----+
| Tables in mysql |
+-----+
| db               |
| host             |
| user             |
+-----+
3 rows in set (0.00 sec)
```

Figure 3.9 Listing the columns of a selected database.

This command displays the descriptions of the fields in the database table. It shows the type, default value, null or not null, and any key fields. This is a very useful command and is used quite extensively. As you can see, the command line is a great tool when working with your database. It may take some getting used to, but in the end, it is fast, powerful, and reliable.

3.3.6 The Built-in Database

With the conclusion of your brief tour of MySQL, you probably noticed that there are already two databases within MySQL. You are probably asking yourself what these tables are and how they are used. The following section describes this built-in database.

The default database is the `mysql` database. This database stores all the privilege information. For now, just know that the `mysql` database stores all the user, database, and host privileges. Altering or dropping any of these tables will cause problems within MySQL. It is therefore recommended that the tables within this database keep their default structure. Do not change the default types.

3.4 Database Security

Nowadays, databases are cardinal components of any web based application by enabling websites to provide varying dynamic content. Since very sensitive or secret informations can be stored in such database, you should strongly consider to protect them somehow.

To retrieve or to store any information you need to connect to the database, send a legitimate query, fetch the result, and close the connection. Nowadays, the commonly used query language in this interaction is the Structured Query Language (SQL). See how an attacker can tamper with an SQL query.

As you can realize, PHP cannot protect your database by itself. The following sections aim to be an introduction into the very basics of how to access and manipulate databases within PHP scripts.

Keep in mind this simple rule: defence in depth. In the more place you take the more action to increase the protection of your database, the less probability of that an attacker succeeds, and exposes or abuse any stored secret information. Good design of the database schema and the application deals with your greatest fears.

3.4.1 SQL Injection

Many web developers are unaware of how SQL queries can be tampered with, and assume that an SQL query is a trusted command. It means that SQL queries are able to circumvent access controls, thereby bypassing standard authentication and authorization checks, and sometimes SQL queries even may allow access to host operating system level commands.

Direct SQL Command Injection is a technique where an attacker creates or alters existing SQL commands to expose hidden data, or to override valuable ones, or even to execute dangerous system level commands on the database host. This is accomplished by the application taking user input and combining it with static parameters to build a SQL query. The following examples are based on true stories, unfortunately.

Owing to the lack of input validation and connecting to the database on behalf of a superuser or the one who can create users, the attacker may create a superuser in your database.

Normal users click on the 'next', 'prev' links where the \$offset is encoded into the URL. The script expects that the incoming \$offset is decimal number. However, someone tries to break in with appending urlencode()'d form of the following to the URL. If it happened, then the script would present a superuser access to him. Note that 0; is to supply a valid offset to the original query and to terminate it.

A feasible way to gain passwords is to circumvent your search result pages. What the attacker needs only is to try if there is any submitted variable used in SQL

statement which is not handled properly. These filters can be set commonly in a preceding form to customize WHERE, ORDER BY, LIMIT and OFFSET clauses in SELECT statements. If your database supports the UNION construct, the attacker may try to append an entire query to the original one to list passwords from an arbitrary table. Using encrypted password fields is strongly encouraged.

The static part of the query can be combined with another SELECT statement which reveals all passwords: If this query (playing with the ' and --) were assigned to one of the variables used in \$query, the query beast awakened.

SQL UPDATES are also subject to attacking your database. These queries are also threatened by chopping and appending an entirely new query to it. But the attacker might fiddle with the SET clause. In this case some schema information must be possessed to manipulate the query successfully. This can be acquired by examining the form variable names, or just simply brute forcing. There are not so many naming convention for fields storing passwords or usernames.

But a malicious user submits the value ' or uid like'%admin%'; -- to \$uid to change the admin's password, or simply sets \$pwd to "hehehe", admin='yes', trusted=100 " (with a trailing space) to gain more privileges. Then, the query will be twisted: If attacker submits the value a%' exec master..xp_cmdshell 'net user test testpass /ADD' -- to \$prod, then the \$query will be:

MSSQL Server executes the SQL statements in the batch including a command to add a new user to the local accounts database. If this application were running as sa and the MSSQLSERVER service is running with sufficient privileges, the attacker would now have an account with which to access this machine.

3.4.2 Avoiding Techniques

You may plead that the attacker must possess a piece of information about the database schema in most examples. You are right, but you never know when and how it can be taken out, and if it happens, your database may be exposed. If you are using an open source, or publicly available database handling package, which may belong to a content management system or forum, the intruders easily produce a copy of a piece of your code. It may be also a security risk if it is a poorly designed one.

These attacks are mainly based on exploiting the code not being written with security in mind. Never trust on any kind of input, especially which comes from the

client side, even though it comes from a select box, a hidden input field or a cookie. The first example shows that such a blameless query can cause disasters.

- Never connect to the database as a superuser or as the database owner. Use always customized users with very limited privileges.
- Check if the given input has the expected data type. PHP has a wide range of input validating functions, from the simplest ones found in Variable Functions and in Character Type Functions (e.g. `is_numeric()`, `ctype_digit()` respectively) onwards the Perl compatible Regular Expressions support.
- If the application waits for numerical input, consider to verify data with `is_numeric()`, or silently change its type using `settype()`, or use its numeric representation by `sprintf()`.
- Quote each non numeric user input which is passed to the database with `addslashes()` or `addcslashes()`. See the first example. As the examples shows, quotes burnt into the static part of the query is not enough, and can be easily hacked.
- Do not print out any database specific information, especially about the schema, by fair means or foul. See also Error Reporting and Error Handling and Logging Functions.
- You may use stored procedures and previously defined cursors to abstract data access so that users do not directly access tables or views, but this solution has another impacts.

Besides these, you benefit from logging queries either within your script or by the database itself, if it supports. Obviously, the logging is unable to prevent any harmful attempt, but it can be helpful to trace back which application has been circumvented. The log is not useful by itself, but through the information it contains. The more detail is generally better.

CHAPTER FOUR WHAT IS APACHE SERVER?

4.1 What is Apache

The Apache web server project is more than just a piece of software. The Apache web server is the best, and most preferred, HTTP server software in use on the Internet today, and it was written entirely as a volunteer project, by volunteer programmers, in their spare time. That, in itself, is astonishing. That is, it is to people that are not familiar with the Open Source methodology, and Open Source projects like Linux, Perl, Sendmail, and a variety of others. The interesting thing about these volunteer-written, free software packages is that most of us, and our businesses, rely heavily on them, whether we are aware of it or not.

Before diving directly into talking about what Apache is, it is useful to talk about where Apache came from, and how it came to be.

4.1.1 The Apache Server

When Rob left the project, it left a problem. There were still a lot of people using his code, and actively making patches to the code, but there was no longer anyone collecting those patches.

In 1995, Brian Behlendorf and a small group of other developers started collecting these patches in a central repository. Brian got some space donated on a server, and set up a CVS tree so that developers could check in patches. And in April of 1995, they released the first official release (Version 0.6.2), which was given the name Apache, because it was "a patchy server".

The Apache Group, as they were known at that time, had no formal organizational structure, never met, communicated only over email, and worked entirely in their free time, on a volunteer basis. Early the next year, Apache passed NCSA as the most widely used server on the Internet, and is now used on more than 60% of all web servers on the Internet.

4.1.2. Apache's architecture

Since the 1.0 release of Apache (December 1, 1995) Apache has has a modular design. The core of the server is very light-weight, and all other functions are implemented as modules that plug in to the core. This means that you can keep the size of the executable down by leaving out functionality that you don't need. It also means

that if there is some functionality missing that you do need, you can write your own custom module to plug into the core.

4.1.3. More recent history

In the last few years, Open Source has been getting a lot of press, because of Linux, Perl, and Apache. In 1998, IBM decided to abandon development of a web server engine to go into WebSphere - an application server for the web - and use Apache instead. This decision, along with Netscape's decision to release the source code for the Netscape browser, earlier that same year, showed the business world that Open Source was more than just a lot of long-haired anti-establishment types out to bring down the software industry, but that it was actually a good business model. It produces code more quickly, and that code is more reliable, because, in the words of Eric Raymond, with enough eyes, all bugs are shallow.

In June of 1999, The Apache Software Foundation was officially incorporated in the state of Delaware. The ASF has a much broader mission than just the Apache HTTP server, and has several other projects that exist under the larger umbrella of the ASF

The stated goals of the ASF are:

- provide a foundation for open, collaborative software development projects by supplying hardware, communication, and business infrastructure;
- create an independent legal entity to which companies and individuals can donate resources and be assured that those resources will be used for the public benefit;
- provide a means for individual volunteers to be sheltered from legal suits directed at the Foundation's projects; and,
- protect the 'Apache' brand, as applied to its software products, from being abused by other organizations.

Some of the better-known projects under the ASF are the Apache web server, mod_perl, mod_php, and Jakarta.

4.1.4. The Future of Apache

At ApacheCon in Orlando, back in March, Apache 2.0 was released. This is largely a rewrite from earlier versions, and uses a threading model that will increase

performance substantially on most platforms. As of this writing, version Alpha 6 of the 2.0 server has been released.

The Apache Group, as mentioned above, has become the Apache Software Foundation, and continues to take on new projects that seem to fit the larger vision that the ASF has for the future. Open Source, and open standards, produce better software. In the end, this makes life better for all of us, and we should support the ASF in all its endeavors, if only for purely selfish reasons.

4.1.5 Obtaining and Installing Apache

Apache is available as source code, and is probably available as a binary installation for your operating system, unless you are running something truly arcane and rare. And, of course, if you are, you can still get the source code, and compile it yourself.

4.1.6. Hardware / Software Requirements

Apache runs on anything. Almost. It will almost certainly run on whatever you have. I've run Apache on a 386 with 4 MB of RAM. And I've run it on a 4-processor machine with 1GB of RAM. It was happy both places. The Apache.org web site does not list any hardware requirements. It will run on any hardware that runs the supported operating systems. Apache will run on any flavor of *nix, and also on Microsoft Windows (95, 98, NT, 2000), Mac, and OS/2.

4.1.7. Configuring

Once you've compiled and installed your server, you need to configure it for your particular environment. Many of the configuration directives got set when you ran configure (or Configure), and so the server should work correctly immediately. However, you will probably want to change some things, since the default installation is very generic, and not precisely suited to your needs.

Apache, unlike most of its competitors in the web server market, lets you configure everything, down to the smallest detail. And if there's really something that you want to configure that you can't, you have the source code, so you can change it if you are so inclined.

4.1.8. Configuration Files

The configuration for your Apache server is located in a file called `httpd.conf`, which is usually located at `/usr/local/apache/conf/httpd.conf`.

Note that if you installed Apache with a RPM (don't do that!) then the files will be in bizarre places that have no relation to logic. Uninstall the RPM, and install from source. It's a simple process, and reduces your pain in the long run.

The format of `httpd.conf` is very simple.

There are comments, which consist of a hash sign (`#`) at the beginning of a line: Based upon the NCSA server configuration files originally by Rob McCool.

There are directives, which look like a name, followed by a value:

```
ServerAdmin webmaster@rcbowen.com
```

There are sections, or containers, which look rather like HTML tags:

```
<Directory /usr/local/apache/cgi-bin>
```

```
    AllowOverride None
```

```
</Directory>
```

Sections can contain directives, and those directives apply to the resources defined by the container definition. In the above example, the `AllowOverride` directive will apply to files located in the specified directory.

You can edit these configuration files with your favorite text editor. You need to restart the server when you are done editing the configuration files in order for the new configuration to take effect.

```
/usr/local/apache/bin/apachectl configtest.
```

4.1.9. Starting, Stopping, Restarting

There are a variety of ways to control your Apache server. We'll focus on a script that ships with Apache, called `apachectl`, which does a few other things in addition to just starting, stopping, and restarting.

start

Starts the server.

stop

Stops the server.

restart

Restarts the server, if running, by sending a `SIGHUP`. If the server is not running, starts it.

CHAPTER FIVE : WHAT IS PHP

5.1 What is PHP?

PHP is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

Simple answer, but what does that mean? An example:

Example 5-1

```
<html> <head>
    <title>Example</title>
</head> <body>
    <?php
    echo "Hi, I'm a PHP script!";
    ?>
</body></html>
```

Notice how this is different from a script written in other languages like Perl or C++ instead of writing a program with lots of commands to output HTML, you write an HTML script with some embedded code to do something. The PHP code is enclosed in special start and end tags that allow you to jump into and out of "PHP mode".

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

5.2 What can PHP do?

Anything. PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main fields where PHP scripts are used.

Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a webserver and a web browser. You need to run the webserver, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. See the installation instructions section for more information.

Command line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks. See the section about Command line usage of PHP for more information.

Writing client-side GUI applications. PHP is probably not the very best language to write windowing applications, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution. If you are interested in PHP-GTK, visit it's own website.

PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor.

So with PHP, you have the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object oriented programming, or a mixture of them. Although not every standard OOP feature is realized in the current version of PHP, many code libraries and large applications (including the PEAR library) are written only using OOP code.

With PHP you are not limited to output HTML. PHP's abilities includes outputting images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. You can also output easily any text, such as XHTML and any other XML file. PHP can autogenerate these files, and save them in the file system, instead of printing it out, forming a server-side cache for your dynamic content.

We also have a DBX database abstraction extension allowing you to transparently use any database supported by that extension. Additionally PHP supports ODBC, the Open Database Connection standard, so you can connect to any other database supporting this world standard.

PHP also has support for talking to other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (on Windows) and countless others. You can also open raw network sockets and interact using any other protocol. PHP has support for the WDDX complex data exchange between virtually all Web programming languages. Talking about interconnection, PHP has support for instantiation of Java objects and using them transparently as PHP objects. You can also use our CORBA extension to access remote objects.

PHP has extremely useful text processing features, from the POSIX Extended or Perl regular expressions to parsing XML documents. For parsing and accessing XML documents, we support the SAX and DOM standards. You can use our XSLT extension to transform XML documents.

While using PHP in the ecommerce field, you'll find the Cybercash payment, CyberMUT, VeriSign Payflow Pro and CCVS functions useful for your online payment programs.

At last but not least, we have many other interesting extensions, the mnoGoSearch search engine functions, the IRC Gateway functions, many compression utilities (gzip, bz2), calendar conversion, translation...

As you can see this page is not enough to list all the features and benefits PHP can offer. Read on in the sections about installing PHP, and see the function reference part for explanation of the extensions mentioned here.

5.3 General Installation Considerations

Before installing first, you need to know what do you want to use PHP for. There are three main fields you can use PHP, as described in the What can PHP do? section:

- Server-side scripting

- Command line scripting

- Client-side GUI applications

For the first and most common form, you need three things: PHP itself, a web server and a web browser. You probably already have a web browser, and depending on

your operating system setup, you may also have a web server (eg. Apache on Linux or IIS on Windows). You may also rent webspace at a company. This way, you don't need to set up anything on your own, only write your PHP scripts, upload it to the server you rent, and see the results in your browser.

While setting up the server and PHP on your own, you have two choices for the method of connecting PHP to the server. For many servers PHP has a direct module interface (also called SAPI). These servers include Apache, Microsoft Internet Information Server, Netscape and iPlanet servers. Many other servers have support for ISAPI, the Microsoft module interface (OmniHTTPd for example). If PHP has no module support for your web server, you can always use it as a CGI processor. This means you set up your server to use the command line executable of PHP (php.exe on Windows) to process all PHP file requests on the server.

If you are also interested to use PHP for command line scripting (eg. write scripts autogenerating some images for you offline, or processing text files depending on some arguments you pass to them), you always need the command line executable. For more information, read the section about writing command line PHP applications. In this case, you need no server and no browser.

With PHP you can also write client side GUI applications using the PHP-GTK extension. This is a completely different approach than writing web pages, as you do not output any HTML, but manage windows and objects within them.

From now on, this section deals with setting up PHP for web servers on Unix and Windows with server module interfaces and CGI executables.

5.4 Installation on Windows Systems

This section applies to Windows 95/98/Me and Windows NT/2000/XP. Do not expect PHP to work on 16 bit platforms such as Windows 3.1. Sometimes we refer to the supported Windows platforms as Win32.

There are two main ways to install PHP for Windows: either manually or by using the InstallShield installer. If you have Microsoft Visual Studio, you can also build PHP from the original source code. Once you have PHP installed on your Windows system, you may also want to load various extensions for added functionality.

5.4.1. Windows InstallShield

The Windows PHP installer available from the downloads page at <http://www.php.net/>, this installs the CGI version of PHP and, for IIS, PWS, and Xitami, configures the web server as well. Also note, that while the InstallShield installer is an easy way to make PHP work, it is restricted in many aspects, as automatic setup of extensions for example is not supported.

Install your selected HTTP server on your system and make sure that it works.

Run the executable installer and follow the instructions provided by the installation wizard. Two types of installation are supported - standard, which provides sensible defaults for all the settings it can, and advanced, which asks questions as it goes along.

The installation wizard gathers enough information to set up the php.ini file and configure the web server to use PHP. For IIS and also PWS on NT Workstation, a list of all the nodes on the server with script map settings is displayed, and you can choose those nodes to which you wish to add the PHP script mappings.

Once the installation has completed the installer will inform you if you need to restart your system, restart the server, or just start using PHP.

5.4.2 Building From Source

Before getting started, it is worthwhile answering the question: "Why is building on Windows so hard?" Two reasons come to mind:

1. Windows does not (yet) enjoy a large community of developers who are willing to freely share their source. As a direct result, the necessary investment in infrastructure required to support such development hasn't been made.
2. Pretty much all of the instructions that follow are of the "set and forget" the.

5.4.3 Putting It All Together

Follow the instructions for installing the unzip utility of your choosing.

Execute setup.exe and follow the installation instructions. If you choose to install to a path other than c:\cygnus, let the build process know by setting the Cygwin environment variable. On Windows 95/98 setting an environment variable can be done by placing a line in your autoexec.bat. On Windows NT, go to My Computer => Control Panel => System and select the environment tab.

Make a directory and unzip win32build.zip into it.

Launch Microsoft Visual C++, and from the menu select Tools => Options. In the dialog, select the directories tab. Sequentially change the dropdown to Executables, Includes, and Library files, and ensure that cygwin\bin, win32build\include, and win32build\lib are in each list, respectively. (To add an entry, select a blank line at the end of the list and begin typing). Typical entries will look like this:

c:\cygnus\bin

c:\php-win32build\include

c:\php-win32build\lib

Press OK, and exit out of Visual C++.

Make another directory and unzip bindlib_w32.zip into it. Decide whether you want to have debug symbols available (bindlib - Win32 Debug) or not (bindlib - Win32 Release). Build the appropriate configuration:

For GUI users, launch VC++, and then select File => Open Workspace and select bindlib. Then select Build=>Set Active Configuration and select the desired configuration. Finally select Build=>Rebuild All.

For command line users, make sure that you either have the C++ environment variables registered, or have run vcvars.bat, and then execute one of the following:

msdev bindlib.dsp /MAKE "bindlib - Win32 Debug"

msdev bindlib.dsp /MAKE "bindlib - Win32 Release"

At this point, you should have a usable resolv.lib in either your Debug or Release subdirectories. Copy this file into your win32build\lib directory over the file by the same name found in there.

5.4.4 Compiling

The best way to get started is to build the standalone/CGI version.

For GUI users, launch VC++, and then select File => Open Workspace and select php4ts. Then select Build=>Set Active Configuration and select the desired configuration. Finally select Build=>Rebuild All.

For command line users, make sure that you either have the C++ environment variables registered, or have run vcvars.bat, and then execute one of the following:

msdev php4ts.dsp /MAKE "php4ts - Win32 Debug_TS"

msdev php4ts.dsp /MAKE "php4ts - Win32 Release_TS"

At this point, you should have a usable php.exe in either your Debug_TS or Release_TS subdirectories.

Repeat the above steps with `php4isapi.dsp` (which can be found in `sapi\isapi`) in order to build the code necessary for integrating PHP with Microsoft IIS.

5.5 Installation of Windows Extensions

After installing PHP and a webserver on Windows, you will probably want to install some extensions for added functionality. The following table describes some of the extensions available. You can choose which extensions you would like to load when PHP starts by uncommenting the: `'extension=php_*.dll'` lines in `php.ini`. You can also load a module dynamically in your script using `dl()`.

The DLLs for PHP extensions are prefixed with `'php_'` in PHP 4 (and `'php3_'` in PHP 3). This prevents confusion between PHP extensions and their supporting libraries.

Note: In PHP 4.0.6 BCMath, Calendar, COM, FTP, MySQL, ODBC, PCRE, Session, WDDX and XML support is built in. You don't need to load any additional extensions in order to use these functions. See your distributions `README.txt` or `install.txt` for a list of built in modules.

Note: Some of these extensions need extra DLLs to work. Couple of them can be found in the distribution package, in the `'dlls'` folder but some, for example Oracle (`php_oci8.dll`) require DLLs which are not bundled with the distribution package.

Copy the bundled DLLs from `'DLLs'` folder to your Windows PATH, safe places are:

- `c:\windows\system` for Windows 9x/Me
- `c:\winnt\system32` for Windows NT/2000
- `c:\windows\system32` for Windows XP

If you have them already installed on your system, overwrite them only if something doesn't work correctly (Before overwriting them, it is a good idea to make a backup of them, or move them to another folder - just in case something goes wrong).

5.5.1 Windows and PWS/IIS 3

The recommended method for configuring these servers is to use the REG file included with the distribution (`pws-php4cgi.reg`). You may want to edit this file and make sure the extensions and PHP install directories match your configuration. Or you can follow the steps below to do it manually.

Run Regedit.

Navigate to: HKEY_LOCAL_MACHINE /System /CurrentControlSet /Services /W3Svc /Parameters /ScriptMap.

On the edit menu select: New->String Value.

Type in the extension you wish to use for your php scripts. For example .php

Double click on the new string value and enter the path to php.exe in the value data field. ex: c:\php\php.exe.

Repeat these steps for each extension you wish to associate with PHP scripts.

The following steps do not affect the web server installation and only apply if you want your php scripts to be executed when they are run from the command line (ex. run c:\myscripts\test.php) or by double clicking on them in a directory viewer window. You may wish to skip these steps as you might prefer the PHP files to load into a text editor when you double click on them.

Navigate to: HKEY_CLASSES_ROOT

On the edit menu select: New->Key.

Name the key to the extension you setup in the previous section. ex: .php

Highlight the new key and in the right side pane, double click the "default value" and enter phpfile.

Repeat the last step for each extension you set up in the previous section.

Now create another New->Key under HKEY_CLASSES_ROOT and name it phpfile.

Highlight the new key phpfile and in the right side pane, double click the "default value" and enter PHP Script.

Right click on the phpfile key and select New->Key, name it Shell.

Right click on the Shell key and select New->Key, name it open.

Right click on the open key and select New->Key, name it command.

Highlight the new key command and in the right side pane, double click the "default value" and enter the path to php.exe. ex: c:\php\php.exe -q %1. (don't forget the %1).

Exit Regedit.

If using PWS on Windows, reboot to reload the registry.

PWS and IIS 3 users now have a fully operational system. IIS 3 users can use a nifty tool from Steven Genusa to configure their script maps.

5.5.2 Windows and PWS 4 or newer

When installing PHP on Windows with PWS 4 or newer version, you have two options. One to set up the PHP CGI binary, the other is to use the ISAPI module DLL.

If you choose the CGI binary, do the following:

Edit the enclosed pws-php4cgi.reg file (look into the SAPI dir) to reflect the location of your php.exe. Forward slashes should be escaped, for example:
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="c:\\php\\php.exe"

In the PWS Manager, right click on a given directory you want to add PHP support to, and select Properties. Check the 'Execute' checkbox, and confirm.

If you choose the ISAPI module, do the following:

Edit the enclosed pws-php4isapi.reg file (look into the SAPI dir) to reflect the location of your php4isapi.dll. Forward slashes should be escaped, for example:
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="c:\\php\\sapi\\php4isapi.dll"

In the PWS Manager, right click on a given directory you want to add PHP support to, and select Properties. Check the 'Execute' checkbox, and confirm.

5.6 The Configuration File

The configuration file (called php3.ini in PHP 3.0, and simply php.ini as of PHP 4.0) is read when PHP starts up. For the server module versions of PHP, this happens only once when the web server is started. For the CGI version, it happens on every invocation.

```
include_path = ".:c:\php\lib"
```

When using PHP as an Apache module, you can also change the configuration settings using directives in Apache configuration files and .htaccess files (You will need "AllowOverride Options" or "AllowOverride All" privileges).

With PHP 3.0, there are Apache directives that correspond to each configuration setting in the php3.ini name, except the name is prefixed by "php3_".

With PHP 4.0, there are several Apache directives that allow you to change the PHP configuration from within the Apache configuration file itself.

```
php_value name value
```

This sets the value of the specified variable.

```
php_flag name on|off
```

This is used to set a Boolean configuration option.

`php_admin_value name value`

This sets the value of the specified variable. "Admin" configuration settings can only be set from within the main Apache configuration files, and not from .htaccess files.

`php_admin_flag name on|off`

This is used to set a Boolean configuration option.

Example 5-2. Apache configuration example

```
<IfModule mod_php4.c>
```

```
    php_value include_path ".:usr/local/lib/php"
```

```
    php_flag safe_mode on
```

```
</IfModule>
```

```
<IfModule mod_php3.c>
```

```
    php3_include_path ".:usr/local/lib/php"
```

```
    php3_safe_mode on </IfModule>
```

5.7 Security

PHP is a powerful language and the interpreter, whether included in a web server as a module or executed as a separate CGI binary, is able to access files, execute commands and open network connections on the server. These properties make anything run on a web server insecure by default. PHP is designed specifically to be a more secure language for writing CGI programs than Perl or C, and with correct selection of compile-time and runtime configuration options, and proper coding practices, it can give you exactly the combination of freedom and security you need.

As there are many different ways of utilizing PHP, there are many configuration options controlling its behaviour. A large selection of options guarantees you can use PHP for a lot of purposes, but it also means there are combinations of these options and server configurations that result in an insecure setup.

The configuration flexibility of PHP is equally rivalled by the code flexibility. PHP can be used to build complete server applications, with all the power of a shell user, or it can be used for simple server-side includes with little risk in a tightly controlled environment. How you build that environment, and how secure it is, is largely up to the PHP developer.

This chapter starts with some general security advice, explains the different configuration option combinations and the situations they can be safely used, and describes different considerations in coding for different levels of security.

5.8 General Considerations

A completely secure system is a virtual impossibility, so an approach often used in the security profession is one of balancing risk and usability. If every variable submitted by a user required two forms of biometric validation (such as a retinal scan and a fingerprint), you would have an extremely high level of accountability. It would also take half an hour to fill out a fairly complex form, which would tend to encourage users to find ways of bypassing the security.

The best security is often inobtrusive enough to suit the requirements without the user being prevented from accomplishing their work, or over-burdening the code author with excessive complexity. Indeed, some security attacks are merely exploits of this kind of overly built security, which tends to erode over time.

A phrase worth remembering: A system is only as good as the weakest link in a chain. If all transactions are heavily logged based on time, location, transaction type, etc. but the user is only verified based on a single cookie, the validity of tying the users to the transaction log is severely weakened.

When testing, keep in mind that you will not be able to test all possibilities for even the simplest of pages. The input you may expect will be completely unrelated to the input given by a disgruntled employee, a cracker with months of time on their hands, or a housecat walking across the keyboard. This is why it's best to look at the code from a logical perspective, to discern where unexpected data can be introduced, and then follow how it is modified, reduced, or amplified.

The Internet is filled with people trying to make a name for themselves by breaking your code, crashing your site, posting inappropriate content, and otherwise making your day interesting. It doesn't matter if you have a small or large site, you are a target by simply being online, by having a server that can be connected to. Many cracking programs do not discern by size, they simply trawl massive IP blocks looking for victims. Try not to become one.

5.8.1 Installed as an Apache module

When PHP is used as an Apache module it inherits Apache's user permissions (typically those of the "nobody" user). This has several impacts on security and authorization. For example, if you are using PHP to access a database, unless that database has built-in access control, you will have to make the database accessible to the "nobody" user. This means a malicious script could access and modify the database, even without a username and password. It's entirely possible that a web spider could stumble across a database administrator's web page, and drop all of your databases. You can protect against this with Apache authorization, or you can design your own access model using LDAP, .htaccess files, etc. and include that code as part of your PHP scripts.

Often, once security is established to the point where the PHP user (in this case, the apache user) has very little risk attached to it, it is discovered that PHP is now prevented from writing any files to user directories. Or perhaps it has been prevented from accessing or changing databases. It has equally been secured from writing good and bad files, or entering good and bad database transactions.

A frequent security mistake made at this point is to allow apache root permissions, or to escalate apache's abilities in some other way.

Escalating the Apache user's permissions to root is extremely dangerous and may compromise the entire system, so sudo'ing, chroot'ing, or otherwise running as root should not be considered by those who are not security professionals.

There are some simpler solutions. By using `open_basedir` you can control and restrict what directories are allowed to be used for PHP. You can also set up apache-only areas, to restrict all web based activity to non-user, or non-system, files.

5.8.2 Installed as CGI binary

Using PHP as a CGI binary is an option for setups that for some reason do not wish to integrate PHP as a module into server software (like Apache), or will use PHP with different kinds of CGI wrappers to create safe chroot and setuid environments for scripts. This setup usually involves installing executable PHP binary to the web server `cgi-bin` directory. CERT advisory CA-96.11 recommends against placing any interpreters into `cgi-bin`. The query information in a url after the question mark (?) is passed as command line arguments to the interpreter by the CGI interface. Usually interpreters open and execute the file specified as the first argument on the command

line. When invoked as a CGI binary, PHP refuses to interpret the command line arguments.

Accessing any web document on server: `http://my.host/cgi-bin/php/secret/doc.html`. The path information part of the url after the PHP binary name, `/secret/doc.html` is conventionally used to specify the name of the file to be opened and interpreted by the CGI program. Usually some web server configuration directives (Apache: Action) With this setup, the web server first checks the access permissions to the directory `/secret`, and after that creates the redirected request `http://my.host/cgi-bin/php/secret/script.php`. Unfortunately, if the request is originally given in this form, no access checks are made by web server for file `/secret/script.php`, but only for the `/cgi-bin/php` file. This way any user able to access `/cgi-bin/php` is able to access any protected document on the web server.

CHAPTER SIX : APPLICATION CODES

6.1 User Interface Area Codes

6.1.1. Index.php

```
<?
require_once("all_function.php");
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--
.style2 { color: #000000}
.style8 {
    font-family: Georgia, "Times New Roman", Times, serif;
    color: #336699;
    font-size: 16px; }
.style17 { font-size: 12px}
.style20 { font-size: 16pt}
.style21 { font-size: 18px}
.style22 {
    font-size: 18pt;
    font-weight: bold; }
-->
</style> </head>
<body bgcolor="#999999">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0"><style type="text/css">
a:link { text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt; font-weight: bold; }
a:visited { text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt; font-weight: bold; }
a:active { font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt; color: #000000; text-decoration:none; font-weight: bold; }
a:hover { font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt; color: #ffffff; text-decoration:none; font-weight: bold;
}</style> <tr align="left" valign="top">
    <td height="97" colspan="2" ><table width="100%" height="119" border="0"
cellpadding="0" cellspacing="0"> <tr>
        <td width="30%" rowspan="2" align="left" valign="top"
bgcolor="#FFFFFF"></td>
```



```

<td width="55%" height="36" align="center" valign="top" nowrap
bgcolor="#FFB64A"><p></p>
<p><span class="style17"><span class="style20"><span class="style21"><span
class="style22">FARM and MILK PRODUCER
SYSTEM</span></span></span></span> </p>
<td width="15%" rowspan="2" align="left" bgcolor="#FFB64A"></td> </tr> <tr>
<td height="53" align="center" valign="middle" bgcolor="#ffb64a"><p><a
href="index.php" class="style2">Home</a> |<a href="farm.php?op=10"
class="style2">Farm</a><span class="style2"></span>|<a
href="animal_info.php?op=10" class="style2">Animal</a>|<a href="about.php"
class="style2">About Us</a>|<a href="help.php" class="style2">Help</a>|<a
href="admin" class="style2"><strong>Admin</strong></a>|</p> </td> </tr>
</table> </td> </tr> <tr align="left" valign="top">
<td width="24%" height="216" bgcolor="#FFFFFF"><table width="100%"
height="214" border="0" cellpadding="0" cellspacing="0"><tr>
<td width="24%" height="214" align="center" valign="top"
bgcolor="#FBEAD2"> <br>
<table width="94%" height="116" border="0" cellpadding="0" cellspacing="0"
bordercolor="#9B0000"> <tr>
<td width="18%" align="left" valign="baseline" bgcolor="#9B0000"><div
align="center"></div></td>
<td width="82%" align="left" valign="baseline" bgcolor="#9B0000"><div
align="left"><font color="#FFFFFF"><strong>LINK</strong></font></div></td>
</tr> <tr> <td align="left" valign="baseline" bgcolor="#ffb64a">
<span class="style2"><a
href="index.php"></a></span></td>
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="index.php" class="style2">HomePage</a></td> </tr> <tr>
<td align="left" valign="baseline" bgcolor="#ffb64a"><strong><a
href="farm.php?op=10"></a></strong></td>
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="farm.php?op=10" class="style2">Farm</a></td>
</tr> <tr>
<td align="left" valign="baseline" bgcolor="#ffb64a"><strong><a
href="animal_info.php?op=10"></a></strong></td>
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="animal_info.php?op=10" class="style2">Animal</a></td>
</tr> <tr>
<td align="left" valign="baseline" bgcolor="#ffb64a"></td>
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="help.php" class="style2">Help</a></td>
</tr> <tr>
<td height="22" align="left" valign="baseline" bgcolor="#ffb64a"></td>
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="about.php" class="style2">About Us</a></td>

```

```

</tr> <tr>
    <td height="27" align="left" valign="baseline" bgcolor="#ffb64a"></td>
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="admin" class="style2">Admin</a></td></tr><tr><td align="left"
valign="baseline" bgcolor="#ffb64a"><a href="javascript:history.back(1)"></a></td>
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><div
align="left"><a href="javascript:history.back(1)">Back</a></div></td>
</table> <br>
<table width="95%" height="67" border="0" cellpadding="0" cellspacing="0"
bordercolor="#9B0000" bgcolor="#ffb64a">
    <tr> <td width="135" height="21" align="left" valign="baseline"
bgcolor="#9B0000">
        <div align="center"><strong><font
color="#FFFFFF">CITIES</font></strong></div></td>
    </tr> <?
$сор_region=mysql_query("select * from region where city = town ");
$num_region=mysql_num_rows($сор_region);
for($i=1; $i<=$num_region; $i++)
{ $рег=mysql_fetch_array($сор_region);
$city_name= $рег['city'];
?> <tr> <td height="22" align="center" valign="top"><? echo "$city_name";
?></td></tr> <?
$сор_town=mysql_query("select * from region where city = '$city_name' AND city !=
town ");
$num_town=mysql_num_rows($сор_town);
for($t=1; $t<=$num_town; $t++) {
$town=mysql_fetch_array($сор_town);
$town_name = $town['town'];
$с_no = $town['city_no'];
?> <tr> <td height="22" align="left" valign="top"> <a href="farm.php?city_no=<? echo "$с_no"; ?>"><? echo
"$town_name"; ?></a>
    </td> </tr> <? }
} ?> </table> <br> <br></td>
    <td align="left" valign="top" bgcolor="#FEF7ED"><table width="100%"
border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="4%" height="248"></td>
            <td width="95%"><br>
                <br><br></td>
            <td width="1%"></td> </tr> </table></td> </tr> </table></td> </tr> <tr >
    <td bgcolor="#9B0000"><FONT face="Arial, Helvetica"
color=#0000ff size=1> </FONT>
    <HR>
    <font face="Arial, Helvetica"
color=#FFFFFF size=1>All rights reserved. www.milkfarm.com &copy; 2006
Cyprus</font>
</td> </tr> </table></body> </html>

```


6.1.2. Farm.php

```
<?
require_once("all_function.php");
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--
.style1 {
    font-size: 9px;
    font-weight: bold; }
.style22 {font-size: 18pt;
    font-weight: bold; }
-->
</style> </head>
<body bgcolor="808080">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0">
<style type="text/css">
a:link {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    font-weight: bold; }
a:visited {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    font-weight: bold; }
a:active {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    color: #000000;
    text-decoration:none;
    font-weight: bold; }
a:hover {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    color: #FF0000;
    text-decoration:none;
    font-weight: bold; }</style>
<tr align="left" valign="top">
<td height="97" colspan="2" ><table width="100%" height="121" border="0"
cellpadding="0" cellspacing="0">
<tr>
<td width="33%" rowspan="2" align="left" valign="top"
bgcolor="#FFFFFF"></td>
```

```

        <td width="48%" height="36" align="center" valign="top" nowrap
bgcolor="#FFB64A"><p class="style22"></p>
        <p class="style22">FARM and MILK PRODUCER SYSTEM</p></td>
        <td width="19%" rowspan="2" align="center" valign="middle"
bgcolor="#FFB64A"></td>
    </tr>
    <tr>
        <td height="44" align="center" valign="middle" bgcolor="#ffb64a"><p><a
href="index.php" class="style2">Home</a> |<a href="farm.php?op=10"
class="style2">Farm</a><span class="style2"></span>|<a
href="animal_info.php?op=10" class="style2">Animal</a>|<a href="about.php"
class="style2">About Us</a>|<a href="help.php" class="style2">Help</a>|<a
href="admin" class="style2"><strong>Admin</strong></a>|</p>
        </tr>
    </table>
    <td align="left" valign="top">
        <td width="24%" height="216" bgcolor="#FFFFFF"><table width="100%"
height="214" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td width="24%" height="214" align="center" valign="top"
bgcolor="#FBEAD2">
                    <br>
                    <table width="94%" height="185" border="0" cellpadding="0" cellspacing="0"
bordercolor="#9B0000">
                        <tr>
                            <td width="18%" align="left" valign="baseline" bgcolor="#9B0000"><div
align="center"></div></td>
                            <td width="82%" align="left" valign="baseline" bgcolor="#9B0000"><div
align="left"><font color="#FFFFFF"><strong>LINK</strong></font></div></td>
                        </tr>
                        <tr>
                            <td align="left" valign="baseline"
bgcolor="#ffb64a"><span
class="style2"><a href="index.php"></a></span></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="index.php" class="style2">HomePage</a></td>
                        </tr>
                        <tr>
                            <td align="left" valign="baseline" bgcolor="#ffb64a"><strong><a
href="farm.php?op=10"></a></strong></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="farm.php?op=10" class="style2">Farm</a></td>
                        </tr>
                        <tr>
                            <td align="left" valign="baseline" bgcolor="#ffb64a"><strong><a
href="animal_info.php?op=10"></a></strong></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="animal_info.php?op=10" class="style2">Animal</a></td>
                        </tr>
                        <tr>
                            <td align="left" valign="baseline" bgcolor="#ffb64a"></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="help.php" class="style2">Help</a></td>
                        </tr>
                        <tr>
                            <td height="22" align="left" valign="baseline" bgcolor="#ffb64a"></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="about.php" class="style2">About Us</a></td>
                        </tr>
                        <tr>
                            <td height="27" align="left" valign="baseline" bgcolor="#ffb64a"></td>
                            <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a
href="admin" class="style2">Admin</a></td>
                        </tr>
                    </table>
                </td>
            </tr>
        </table>
    </td>

```



```

<tr><td align="left" valign="baseline" bgcolor="#ffb64a"><a
href="javascript:history.back(1)"></a></td>
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><div
align="left"><a href="javascript:history.back(1)">Back</a></div></td>
</table>    <br>
    <table width="95%" height="67" border="0" cellpadding="0" cellspacing="0"
bordercolor="#9B0000" bgcolor="#ffb64a">    <tr>
        <td width="135" height="21" align="left" valign="baseline"
bgcolor="#9B0000">
            <div align="center"><strong><font
color="#FFFFFF">CITIES</font></strong></div></td>    <tr>    <?
$SOR_region=mysql_query("select * from region where city = town ");
$num_region=mysql_num_rows($SOR_region);
for($i=1; $i<=$num_region; $i++)
{ $reg=mysql_fetch_array($SOR_region);
$city_name= $reg['city'];
?><tr> <td height="22" align="center" valign="top"><? echo "$city_name"; ?></td>
    </tr>    <?
$SOR_town=mysql_query("select * from region where city = '$city_name' AND city !=
town ");
$num_town=mysql_num_rows($SOR_town);
for($t=1; $t<=$num_town; $t++){
$town=mysql_fetch_array($SOR_town);
$town_name = $town['town'];
$C_no = $town['city_no'];?>    <tr>
        <td height="22" align="left" valign="top"> <a href="farm.php?city_no=<? echo "$C_no"; ?>"><? echo
"$town_name"; ?></a>    </td>    </tr>    <? } }?> </table> </td>
        <td width="76%" align="center" valign="top"><strong><br>FARMS IN THE
NORTH CYPRUS</strong><br> <br> <br> <? if(!$op) {
$SOR_farm=mysql_query("select * from farm where city_no = $city_no ");
$num_farm=mysql_num_rows($SOR_farm);
for($f=1; $f<=$num_farm; $f++) {
$far=mysql_fetch_array($SOR_farm);
$farm_n= $far['farm_name'];
$f_id= $far['farm_id'];
echo" <a href='farm_info.php?farm_id=$f_id'>ID : $f_id - $farm_n <br><br></a>";
} } else if($op==10) { ?>
<table width="75%" height="98" border="0" cellpadding="0" cellspacing="0">
    <tr>    <td width="171" height="16" align="left" valign="top"><div
align="center"><span class="style1"><font>LOCATIONS </font></span></div></td>
        <td width="239" align="left" valign="top"><span
class="style1"><font>CITIES</font></span></td>    <tr>    <tr>
            <td colspan="2" ><div align="center"><font
color="#FFFFFF"><strong><br>    <br> </strong></font></div></td>    <tr>    <?
$SOR_region=mysql_query("select * from region where city = town ");
$num_region=mysql_num_rows($SOR_region);
for($i=1; $i<=$num_region; $i++) {
$reg=mysql_fetch_array($SOR_region);

```

```

$city_name= $reg['city']; ?>      <tr>
      <td height="22" colspan="2" align="center" valign="top"><? echo
"$city_name"; ?></td> </tr>      <?
$sor_town=mysql_query("select * from region where city = '$city_name' AND city !=
town "); $num_town=mysql_num_rows($sor_town);
for($t=1; $t<=$num_town; $t++) {
$town=mysql_fetch_array($sor_town);
$town_name = $town['town'];
$c_no = $town['city_no']; ?>      <tr>
      <td height="22" colspan="2" align="left" valign="top"> <a href="farm.php?city_no=<? echo
"$c_no"; ?>"><? echo "$town_name"; ?></a>      </td>
      </tr> <? } } ?> </table> ? } ?>      <br><br><br> </td> </tr>
      </table></td> </tr>      <tr align="left" valign="top">
      <td height="34" bgcolor="#FFFFFF"><FONT face="Arial, Helvetica"
color=#0000ff size=1> </FONT>      <HR>
      <font face="Arial, Helvetica"
color=#0000ff size=1>All rights reserved. www.milk.com &copy; 2006
Cyprus</font></td> </tr> </table> </body> </html>

```

6.1.3. Farm_info.php

```

<?
require_once("all_function.php");
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--
.style22 { font-size: 18pt;
           font-weight: bold; }
--> </style> </head>
<body bgcolor="808080">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr align="left" valign="top">
    <td height="97" colspan="2" ><table width="100%" height="97" border="0"
cellpadding="0" cellspacing="0">
      <style type="text/css">
a:link { text-decoration: none;
         font-family: Verdana, Arial, Helvetica, sans-serif;
         font-size: 7pt;
         font-weight: bold;}
a:visited { text-decoration: none;
            font-family: Verdana, Arial, Helvetica, sans-serif;
            font-size: 7pt;
            font-weight: bold;}

```



```
a:hover { font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #ffffff;
text-decoration: none;
font-weight: bold; }</style>
```

[illegible]


```

<td width="76%" align="center" valign="top"><strong>Farm
Information</strong><br><br><?
$sor_farm=mysql_query("select
farm.farm_id,farm.farm_name,farm.farm_address,farmer.farmer_name from
farm,farmer where farm.farm_id='$farm_id' OR farmer.farm_id = '$farm_id' ");
$far=mysql_fetch_array($sor_farm);
$farm_n= $far['farm_name'];
$f_no= $far[farm_no];
$farm_a= $far['farm_address'];
$farmer_n= $far['farmer_name'];
$f_id= $far['farm_id'];
$num_cow=0;
$num_sheep=0;
$num_goat=0;
$sor_an=mysql_query("select * from animal where farm_id = '$farm_id' ");
$num_animal=mysql_num_rows($sor_an);
for($num=1; $num<=$num_animal; $num++)
{
$animal=mysql_fetch_array($sor_an);
$cow = $animal['cow_id'];
$sheep = $animal['sheep_id'];
$goat = $animal['goat_id'];
if($cow != "")
$num_cow = $num_cow+1;
if($sheep != "")
$num_sheep = $num_sheep+1;
if($goat != "")
$num_goat = $num_goat+1;
}
echo "Farm id : $f_id<br><br><br>";
echo "Name : $farm_n<br><br><br>";
echo "Owner : $farmer_n<br><br><br>";
echo "Address : $farm_a<br><br>";
echo "<a href='animal_info.php?farm_id=$farm_id'>";
echo "Cow : $num_cow <br> Sheep : $num_sheep <br> Goat : $num_goat <br><br> ";
echo "</a>";
$total = $num_cow + $num_sheep + $num_goat ;
echo "Number of total animal : $total";
$sor_milk=mysql_query("select * from milk where farm_id = '$farm_id' ");
$num_milk=mysql_num_rows($sor_milk);
$cow_milk = 0;
$sheep_milk = 0;
$goat_milk = 0;
for($milk=1; $milk<=$num_milk; $milk++)
{
$milk=mysql_fetch_array($sor_milk);
$cow=$milk[cow_milk];
$sheep=$milk[sheep_milk];
$goat=$milk[goat_milk];
$cow_milk = $cow_milk + $cow;

```

```

$sheep_milk = $sheep_milk + $sheep;
$goat_milk = $goat_milk + $goat;
}
$total_milk = $cow_milk + $sheep_milk + $goat_milk;
echo "<br><br><br><br>Milk Informations<br>";
echo "<a href='milk_info.php?farm_id=$farm_id'><br>Cow : $cow_milk KG <br>
Sheep : $sheep_milk KG <br> Goat : $goat_milk KG<br></a>";
echo "<br>total milk : $total_milk KG";
?></td> </tr> </table></td> </tr>
<tr align="left" valign="top">
<td height="34" bgcolor="#FFFFFF"><FONT face="Arial, Helvetica"
color=#0000ff size=1>&nbsp; </FONT> <HR>
<font face="Arial, Helvetica"
color=#0000ff size=1>All rights reserved. www. milkfarm.com &copy; 2006
Cyprus</font> </td></tr></table></body></html>

```

6.1.4. Milk_Info.php

```

<?
require_once("all_function.php");
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--
.style22 { font-size: 18pt;
font-weight: bold;
}
-->
</style>
</head>
<body bgcolor="#FEF7ED">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0">
<style type="text/css">
a:link {text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold; }
a:visited { text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold; }
a:active {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #000000;

```



```

$sor_town=mysql_query("select * from region where city = '$city_name' AND city !=
town ");
$num_town=mysql_num_rows($sor_town);
for($t=1; $t<=$num_town; $t++){
$town=mysql_fetch_array($sor_town);
$town_name = $town['town'];
$sc_no = $town['city_no'];
?>
<tr>
<td height="22" align="left" valign="top"> &nbsp;&nbsp;&nbsp;<a href="farm.php?city_no=? echo "$c_no";
?>"><? echo "$town_name"; ?</a>
</td> </tr> <?
} } ?> </table> <br>
<br></td>
<td align="left" valign="top" bgcolor="#FEF7ED"><?
if(!$op)
{ ?> <br> <br>
<a href="milk_info.php?op=1&farm_id=? echo "$farm_id"; ?>">Show Incoming
Milks to Accept</a><br>
<br> <br> <br>
<a href="milk_info.php?op=2&farm_id=? echo "$farm_id"; ?>">Show Accepted
Milks</a>
<font color="#BDE6EA">
<? }
else if($op==1)
{
$milk_acc=mysql_query("select * from milk where verify ='0' and farm_id = '$farm_id'
order by tarih desc");
$num_acc=mysql_num_rows($milk_acc);
echo "<br><br><center>There are $num_acc milk incoming wait to accept</center>
<br><br>";
$sum_c=0;
$sum_s=0;
$sum_g=0; ?>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="19%" height="32"><strong>Farm id </strong></td>
<td width="18%"><strong>Cow Milk </strong></td>
<td width="19%"><strong>Sheep Milk </strong></td>
<td width="15%"><strong>Goat Milk </strong></td>
<td width="29%"><strong>Date</strong></td>
</tr><?
for($ma=1; $ma<=$num_acc; $ma++)
{
$milk=mysql_fetch_array($milk_acc);
$f_id = $milk['farm_id'];
$p_no = $milk['produce_no'];
$cow = $milk['cow_milk'];
$sheep = $milk['sheep_milk'];
$goat = $milk['goat_milk'];

```

```

$trh= $milk[tarih];
$sac_date=substr("$trh",0,10);
$sum_c=$sum_c+$cow;
$sum_s=$sum_s+$sheep;
$sum_g=$sum_g+$goat;
?> <tr>
    <td><? echo "$f_id"; ?></td>
    <td><? echo "$cow"; ?></td>
    <td><? echo "$sheep"; ?></td>
    <td><? echo "$goat"; ?></td>
    <td><? echo "$ac_date"; ?></td> </tr><?
}
echo " </table>";
echo "<br><br><br>";
$totalll=$sum_c+$sum_s+$sum_g;
echo "Wait Milk<br> $sum_c KG COW <br> $sum_s KG SHHEP <br> $sum_g KG
GOAT<br><br>";
echo "Total $totalll KG Milk Wait to Accept"; }
    else if($op==2) {
        $milk_accepted=mysql_query("select * from milk_accepted where
farm_id='$farm_id' order by date desc");
$num_accepted=mysql_num_rows($milk_accepted);
echo "<br><br><center>There are $num_accepted Entering Accepted
Milk<center><br><br>";
$sum_ca=0;
$sum_sa=0;
$sum_ga=0;?>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td width="19%"><strong>Farm id </strong></td>
        <td width="18%"><strong>Cow Milk </strong></td>
        <td width="19%"><strong>Sheep Milk </strong></td>
        <td width="15%"><strong>Goat Milk </strong></td>
        <td width="29%"><strong>Date</strong></td>
    </tr> <?
for($mac=1; $mac<=$num_accepted; $mac++)
{
$milk=mysql_fetch_array($milk_accepted);
$f_id = $milk['farm_id'];
$p_no = $milk[produce_no];
$cow = $milk[cow_milk];
$sheep = $milk[sheep_milk];
$goat = $milk[goat_milk];
$trh= $milk[date];
$sac_date=substr("$trh",0,10);
$sum_ca=$sum_ca+$cow;
$sum_sa=$sum_sa+$sheep;
$sum_ga=$sum_ga+$goat;
?> <tr> <td><? echo "$f_id"; ?></td>
    <td><? echo "$cow"; ?></td>
    <td><? echo "$sheep"; ?></td>

```



```

        <td><? echo "$goat"; ?></td>
        <td><? echo "$ac_date"; ?></td> </tr> <?
    }
    echo "</table>";
    echo "<br><br><br>";
    $total_accepted=$sum_ca+$sum_sa+$sum_ga;
    echo "Accepted Milk<br> $sum_ca KG COW <br> $sum_sa KG SHHEP <br>
    $sum_ga KG GOAT<br><br>";
    echo "Total $total_accepted KG Milk Accepted";
        }      ?> </td>    </tr>
    </table></td>
</tr> <tr >
    <td bgcolor="#9B0000"><FONT face="Arial, Helvetica"
        color=#0000ff size=1>&nbsp;   </FONT>    <HR>
        <font face="Arial, Helvetica"
        color=#FFFFFF size=1>All rights reserved. www.milkfarm.com &copy; 2006
    Cyprus</font>    </td>    </tr> </table> </body> </html>

```

6.1.5. Animal_Info.php

```

<?
    require_once("all_function.php");
    $conn=db_connect();
    ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--
.style22 {font-size: 18pt;
        font-weight: bold;
    }
-->
</style>
</head>
<body bgcolor="808080">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0">
    <style type="text/css">
a:link {      text-decoration: none;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 7pt;
        font-weight: bold;}
a:visited {   text-decoration: none;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 7pt;
        font-weight: bold;}
a:active {    font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 7pt;

```



```

$town_name = $town['town'];
$city_no = $town['city_no'];
?> <tr>
    <td height="22" align="left" valign="top"> &nbsp;&nbsp;&nbsp;<a href="farm.php?city_no=<? echo "$city_no"; ?>"><? echo
"$town_name"; ?></a> </td> </tr> <?
}
?> </table> </td>
<td width="76%" align="center" valign="top">
    <? if(!$op) { ?>
        ANIMALS<br> <br> <br> <table width="100%" height="37"
border="0" cellpadding="0" cellspacing="0">
    <tr align="left" valign="top">
        <td height="19" align="center"> <strong>COW ID </strong></td>
        <td align="center"><strong>SHEEP ID </strong></td>
        <td align="center"><strong>GOAT ID </strong></td>
    </tr>
    <tr align="left" valign="top">
        <td><table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#999999">
            <?
            $sor_animal=mysql_query("select * from animal where farm_id = '$farm_id' ");
            $num_animal=mysql_num_rows($sor_animal);
            for($an=1; $an<=$num_animal; $an++)
            {
                $animal=mysql_fetch_array($sor_animal);
                $coww = $animal['cow_id'];
            }
            <tr>
                <td align="center" valign="top"><? echo "$coww"; ?></td>
            </tr>
        </table></td>
    </tr>
    <td><table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#CCCCCC">
            <?
            $sor_animal2=mysql_query("select * from animal where farm_id = '$farm_id' ");
            $num_animal2=mysql_num_rows($sor_animal2);
            for($ko=1; $ko<=$num_animal2; $ko++)
            {
                $animal2=mysql_fetch_array($sor_animal2);
                $sheep = $animal2['sheep_id'];
            }
            <tr>
                <td align="center" valign="top"><? echo "$sheep"; ?></td>
            </tr>
        </table></td>
    </tr>
    <td><table width="100%" border="0" cellpadding="0" cellspacing="0"
bgcolor="#999999">
            <?
            $sor_animal3=mysql_query("select * from animal where farm_id = '$farm_id' ");
            $num_animal3=mysql_num_rows($sor_animal3);
            for($ko=1; $ko<=$num_animal3; $ko++)
            {
                $animal3=mysql_fetch_array($sor_animal3);

```



```

$goatt = $animal3['goat_id'];
?>
    <tr>
        <td align="center" valign="top"><? echo "$goatt"; ?></td>
    </tr>
    <? }

    ?>    </table></td>    </tr>    </table>
    <?>    else if($sop==10) { ?>
        <table width="50%" height="100" border="0" cellpadding="0"
cellspacing="0">
            <tr>
                <td width="135" height="22" align="left" valign="top"><div
align="left"><br> <br> <strong>LOCATIO<font
color="#000000">NS</font></strong></div></td> </tr> <tr>
                <td ><div align="center"><font
color="#000000"><strong>CITIES</strong></font><font
color="#FFFFFF"><strong><br>
                    <br> </strong></font></div></td></tr> <?
$sor_region=mysql_query("select * from region where city = town ");
$num_region=mysql_num_rows($sor_region);
for($i=1; $i<=$num_region; $i++)
{
    $reg=mysql_fetch_array($sor_region);
    $city_name= $reg['city'];
    ?>    <tr>
        <td height="22" align="center" valign="top"><? echo "$city_name"; ?></td>
    </tr>    <?

$sor_town=mysql_query("select * from region where city = '$city_name' AND city !=
town");
$num_town=mysql_num_rows($sor_town);
for($t=1; $t<=$num_town; $t++)
{
    $town=mysql_fetch_array($sor_town);
    $town_name = $town['town'];
    $c_no = $town['city_no'];
    ?>    <tr>
        <td height="22" align="left" valign="top"> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="farm.php?city_no=<? echo "$c_no"; ?>"><? echo
"$town_name"; ?></a>
        </td>    </tr>    <?
}
} ?>    </table>    <? }
    ?>    <br>
        
<br></td>    </tr> </table></td> </tr>
    <tr align="left" valign="top">
        <td height="34" bgcolor="#FFFFFF"><FONT face="Arial, Helvetica"
color=#0000ff size=1>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</FONT>    <HR>
        <font face="Arial, Helvetica" color=#0000ff size=1>All rights reserved.
www.milkfarm.com &copy; 2006 Cyprus</font>    </td></tr></table></body></html>

```

6.1.6. About.php

```
<?
    require_once("all_function.php");
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Milk Produce Systems</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
<!--.style22 { font-size: 18pt;
        font-weight: bold;
    }--></style></head>
<body bgcolor="808080">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0"><style type="text/css">
<!--.style2 { color: #000000}
.style8 {
    font-family: Georgia, "Times New Roman", Times, serif;
    color: #336699;
    font-size: 16px;}
.style17 { font-size: 12px}
.style20 { font-size: 16pt}
.style21 { font-size: 18px}
.style22 {    font-size: 18pt;
    font-weight: bold;}
--></style></head>
<body bgcolor="#FEF7ED">
<table width="75%" height="366" border="0" align="center" cellpadding="0"
cellspacing="0"><style type="text/css">
a:link {    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    font-weight: bold;}
a:visited {    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    font-weight: bold;}
a:active {    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 7pt;
    color: #000000;
    text-decoration:none;
    font-weight: bold;}
a:hover {    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size:7pt;
    color: #ffffff;
    text-decoration:none;
    font-weight: bold;}</style>
```



```
<td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a href="animal_info.php?op=10" class="style2">Animal</a></td> <tr>  
    <td align="left" valign="baseline" bgcolor="#ffb64a">&nbsp;</td>  
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a href="help.php" class="style2">Help</a></td> <tr>  
    <td height="22" align="left" valign="baseline" bgcolor="#ffb64a"></td>  
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a href="about.php" class="style2">>About Us</a></td> <tr>  
    <td height="27" align="left" valign="baseline" bgcolor="#ffb64a"></td>  
    <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><a href="admin" class="style2">Admin</a></td> <tr>  
        <tr><td align="left" valign="baseline" bgcolor="#ffb64a"><a href="javascript:history.back(1)"></a></td>  
        <td align="left" valign="baseline" bgcolor="#ffb64a" class="style2"><div align="left"><a href="javascript:history.back(1)">Back</a></div></td>  
    </table>    <br>  
    <table width="95%" height="67" border="0" cellpadding="0" cellspacing="0" bordercolor="#9B0000" bgcolor="#ffb64a">    <tr>  
        <td width="135" height="21" align="left" valign="baseline" bgcolor="#9B0000">    <div align="center"><strong><font color="#FFFFFF">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&CITIES</font></strong></div></td>    <tr>    <?  
$sor_region=mysql_query("select * from region where city = town ");  
$num_region=mysql_num_rows($sor_region);  
for($i=1; $i<=$num_region; $i++)  
{  
$reg=mysql_fetch_array($sor_region);  
$city_name= $reg['city'];  
?>    <tr>  
        <td height="22" align="center" valign="top"><? echo "$city_name"; ?></td>  
    </tr>    <?  
$sor_town=mysql_query("select * from region where city = '$city_name' AND city != town");  
$num_town=mysql_num_rows($sor_town);  
for($t=1; $t<=$num_town; $t++)  
{  
$town=mysql_fetch_array($sor_town);  
$town_name = $town['town'];  
$c_no = $town[city_no];  
?>    <tr>  
        <td height="22" align="left" valign="top"> <a href="farm.php?city_no=<? echo "$c_no"; ?>"><? echo "$town_name"; ?></a>  
        </td>    </tr>    <?  
} } ?> </table>    </td>  
    <td width="76%" align="left" valign="top"><p>&nbsp;</p>
```



```

        color=#FFFFFF size=1>All rights reserved. www.milk.com &copy; 2006
Cyprus</font>
    </td> </tr></table></body></html>

```

6.1.8. Database.php

```

<? function db_connect()
{
    $result = mysql_pconnect("localhost", "hdasci", "db21neu");
    if (!$result) return false;
    if (!mysql_select_db("farm"))
        return false; return $result; }
?>

```

6.1.9. User_Function.php

```

<?
    require_once("database.php");
    function haberler()
    {
        $conn=db_connect();
        $sor=mysql_query("select * from haberler order by tarih asc ");
        for($i=0; $i<4; $i++)
        {
            $row=mysql_fetch_array($sor);
            $mm= nl2br(stripslashes($row['haber_baslik']));
            echo "<a href='haber.php?haber_no=$row[haber_no]' >";
            echo "$mm";
            echo "</a>";
            echo "<br>";
            echo "<br>";
            echo "<br>"; } }
?>

```

6.1.10. All_Function.php

```

<?
    require_once("database.php");
    require_once("user_function.php");
?>

```

6.2 Application Code For Admin Side

6.2.1. Index.php

[illegible]


```

        <td> <input name="admin_pass" type="password" maxlength="16"></td>
    </tr> <tr align="center" valign="top"> <td colspan="2"><br>
    ;<input type="submit" name="Submit" value="    Enter    ">&nbsp; <a
href=" ../index.php" class="style23 style24">Back to Main Page</a> </td>
    </tr> <td> </table></form></center></body></html>

```

6.2.2 Admin.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();
if ($admin_name && $admin_pass)
{ if (admin_login($admin_name, $admin_pass))
  { $administrator=$admin_name;
    session_register("administrator"); }
  else {
    echo "<font color='red'>ERROR!!!.</font><br>";
    echo "wrong name - password..please try again"; ?>
    <meta http-equiv="refresh" content="3;URL=index.php">
    <? exit; } }
check_administrator();
$qu_permission=mysql_query("select * from admin where admin_name =
'$administrator' ");
$permis=mysql_fetch_array($qu_permission);
$farm_id = $permis['farm_id']; ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <style type="text/css">
<!--.style1 { color: #000000;
font-weight: bold;}
.style8 {font-size: 18; font-weight: bold; }
--></style><head>
<title>farm Manager</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css"></head>
<body bgcolor="#ffb64a">
<table width="780" border="0" align="center" cellpadding="0" cellspacing="0">
<style type="text/css">
a:link {text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold;}
a:visited {text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold;}
a:active {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #000000;
text-decoration:none;
font-weight: bold;}

```

```

a:hover {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #ffffff;
text-decoration: none;
font-weight: bold;}</style>
<tr align="left" valign="top">
<td width="636" height="52"><font color="#0080C0"><? echo $administrator,
?></font><br> <br> <div align="center"><strong><br> FARM REGISTER SYTEMS
MANAGEMENT PANEL <br><br></strong></div></td>
<td width="144" align="center" valign="middle"><a href="logout.php"><font
color="#FF0000" size="2"><strong><em>Signout</em></strong></font></a>
</td> </tr>
<tr align="center" valign="middle">
<td height="186" colspan="2">
<table width="59%" height="364" border="0" cellpadding="4" cellspacing="4"
bgcolor="#D1D1D1">
<?      if($farm_id == ")
{ ?>      <tr align="left" valign="top">
<td width="38%" height="23" ><span class="style1">Region</span></td>
<td width="21%" align="center" ><span class="style8"><font
color="#999999"><[<a href="region.php">Add</a>]</font></span></td>
<td width="19%" align="center" ><span class="style8"><font
color="#999999"><[<a href="region.php?op=2">Edit</a>]</font></span></td>
<td width="22%" align="center" ><span class="style8"><font
color="#999999"><[<a href="region.php?op=5">Delete</a>]</font></span></td>
</tr>      <tr align="left" valign="top">
<td height="25" ><span class="style1">Farms</span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farm.php">Add</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farm.php?op=2">Edit</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farm.php?op=5">Delete</a>]</font></span></td>
</tr>      <tr align="left" valign="top">
<td height="25" ><span class="style1">Farmer</span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farmer.php">Add</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farmer.php?op=2">Edit</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="farmer.php?op=5">Delete</a>]</font></span></td>
</tr>      <tr align="left" valign="top">
<td height="25" ><span class="style1">Animal</span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="animal.php">Add</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="animal.php?op=2">Search</a>]</font></span></td>
<td align="center" ><span class="style8"><font color="#999999"><[<a
href="animal.php?op=4">Delete</a>]</font></span></td>
</tr>      <tr align="left" valign="top">

```



```

<td height="25" ><span class="style1">Milk</span></td>
<td align="center" ><span class="style8"><font color="#999999">[<a
href="milk.php">Add</a>]</font></span></td>
<td align="center" >&nbsp;</td>
<td align="center" >&nbsp;</td>
</tr>
<tr align="left" valign="top">
<td height="25" ><span class="style1">Admin Manager</span></td>
<td align="center" ><span class="style8"><font color="#999999">[<a
href="admin_manage.php">Add</a>]</font></span></td>
<td align="center" >&nbsp;</td>
<td align="center" >&nbsp;</td>
</tr>
<tr align="left" valign="top">
<td height="25" ><span class="style1">Milk Prices</span></td>
<td align="center" ><span class="style8"><font color="#999999">[<a
href="milk_price.php">Add</a>]</font></span></td>
<td align="center">&nbsp;</td>
<td align="center" >&nbsp;</td>
</tr>
<tr align="left" valign="top">
<td height="25" ><span class="style1">Milk Accept</span></td>
<td align="center"><span class="style8"><font color="#999999">[<a
href="milk_accept.php">Accept</a>]</font></span></td>
<td align="center"></td>
<td align="center"></td>
</tr>
<tr align="left" valign="top">
<td height="25" ><span class="style1">Reports</span></td>
<td align="center"><span class="style8"><font color="#999999">[<a
href="report.php">Reports</a>]</font></span></td>
<td align="center"></td>
<td align="center"></td>
</tr>
<!-- supervisor end -->
<?
} else {
?> <!-- farmer admin start -->
<tr align="left" valign="top">
<td height="52" ><span class="style1">Milk</span></td>
<td align="center" ><span class="style8"><font color="#999999">[<a
href="milk_farm.php">Add</a>]</font></span></td>
<td align="center" >&nbsp;</td>
<td align="center" >&nbsp;</td>
</tr>
<tr align="left" valign="top">
<td height="52" ><span class="style1">Farmer Milk Reports</span></td>
<td align="center" ><span class="style8"><font color="#999999">[<a
href="farmer_report.php">Show Reports</a>]</font></span></td>
<td align="center" >&nbsp;</td>
<td align="center" >&nbsp;</td>
</tr>
<!-- farmer admin end -->
<?
}
?> </table> </td> </tr></table>
<div align="center"><a href="logout.php"><font color="#FF0000"
size="2"><strong><em><br> <br> </em></strong></font></a><br>

```



```

session_start();
if ($admin_name && $admin_pass)
{
    if (admin_login($admin_name, $admin_pass))
    {
        $administrator=$admin_name;
        session_register("administrator");
    }
    else {
        echo "<font color='red'>ERROR!!!.</font><br>";
        echo "wrong name - password..please try again";
        exit;
    }
}
check_administrator();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>PM Yönetici Paneli</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--.style1 {color: #0000FF}
.style2 {
    color: #FF0000;
    font-weight: bold;}
--></style></head>
<body bgcolor="#ffb64a">
<table width="545" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
<tr align="left" valign="top">
<td width="418" height="33"><strong></strong>
<div align="center" class="style2"><font size="4">CITY</font></div>
<strong></strong><strong><font color="#BDE6EA" size="4"><br>
<br>
<br><font size="4"><span class="style1">Welcome:</span></font></strong>
<? echo $administrator; ?></td>
<td width="56" align="center" valign="top">
<a href="logout.php"><font color="#FF0000"
size="2"><strong><em>Logout</em></strong></font></a></td>
</tr>
<tr align="left" valign="top">
<td height="156" colspan="2">
<form action="" method="post" name="form1" class="style1">
<p>
<label> new city :
<input type="text" name="textfield">
</label>
<p>&nbsp;
<input name="button" type="button" value="Submit">
</p>
</form>
<form action="" method="post" name="form2" class="style1">
<p>
<label> new town :
<input type="text" name="textfield2">
</label>
<p>
<label>&nbsp;
<input type="submit" name="Submit" value="Submit">
</label>
</p>
</form>

```



```

    $add_farm=mysql_query("insert into farm values (
'$farm_id','$farm_name',$city_no,'$farm_address')");
    if ( $add_farm)
echo "add new farm";
else echo "ERROR!!! not added";
    } else if($op==2) { ?>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr align="left" valign="top">
    <br>    <td width="11%"><span
class="style2"><strong>&nbsp;&nbsp;&nbsp;Region</strong></span></td>
    <td width="89%"> <select name="select"
onChange="window.open(this.options[this.selectedIndex].value,'_parent') ">
    <option disabled>Select a City</option>
    <option <? if($city=='Lefkoşa') echo"selected"; ?>
value="farm.php?op=2&city=Lefkoşa">Lefkoşa</option>
    <option <? if($city=='Lefke') echo"selected"; ?>
value="farm.php?op=2&city=Lefke">Lefke</option>
    <option <? if($city=='Girne') echo"selected"; ?>
value="farm.php?op=2&city=Girne">Girne</option>
    <option <? if($city=='Magosa') echo"selected"; ?>
value="farm.php?op=2&city=Magosa">Magosa</option>
    <option <? if($city=='iskele') echo"selected"; ?>
value="farm.php?op=2&city=iskele">iskele</option>
    </select>    <br>    <br> </td>    </tr>
<tr>    <td>&nbsp;&nbsp;&nbsp;</td>    </tr> </table>    <?
    $sor_farm=mysql_query("select farm.farm_id,farm.farm_name from
farm,region where region.city = '$city' AND region.city_no=farm.city_no");
$num_farm=mysql_num_rows($sor_farm);
for($f=1; $f<=$num_farm; $f++)
{
$far=mysql_fetch_array($sor_farm);
$farm_n= $far['farm_name'];
$f_id= $far['farm_id'];
    echo "&nbsp;&nbsp;&nbsp; $f - <a href='farm.php?op=3&farm_id=$f_id'>ID : $f_id - $farm_n
farm <br><br></a>";
} }    else if($op==3)    {
    $farm_e=mysql_query("select * from farm where farm_id='$farm_id'");
    $far_e=mysql_fetch_array($farm_e);
    $f_name= $far_e['farm_name'];
    $f_adr= $far_e['farm_address'];    ?>
    <form name="form1" method="post" action="farm.php?op=4&farm_id=<? echo
"$farm_id"; ?>">    <br>
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr align="left" valign="top">
    <td width="25%" height="62"><span
class="style2"><strong>&nbsp;&nbsp;&nbsp;Farm Name</strong></span></td>
    <td width="75%">
    <input name="farm_name" type="text" value="<? echo " $f_name"; ?>"
size="45">    <br><br>    </td>    <tr align="left" valign="top">

```

```

        <td><span class="style2"><strong>&nbsp;&nbsp;&nbsp;Farm
Address</strong></span></td>        <td>
        <input name="farm_address" type="text" value="<? echo " $f_adr"; ?>"
size="45"> <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td>&nbsp;&nbsp;&nbsp;</td>        <td>
        <input name="submit" type="submit" value="  Update  "></td>
</tr>        <tr align="left" valign="top">
        <td>&nbsp;&nbsp;&nbsp;</td> <td>&nbsp;&nbsp;&nbsp;</td>        </tr>
        <tr align="left" valign="top">
        <td>&nbsp;&nbsp;&nbsp;</td>
        <td>&nbsp;&nbsp;&nbsp;</td> </tr> </table> </form>
        <?
        }
        else if($op==4)
        {
            $farm_upd=mysql_query("update farm set
farm_name='$farm_name',farm_address='$farm_address' where farm_id='$farm_id' ");
if( $farm_upd)
echo "neew info updated";
else
echo "Error..no update"; }
        else if($op==5){
            $farm_del=mysql_query("select * from farm");
$num_farm_d=mysql_num_rows($farm_del);
for($f_d=1; $f_d<=$num_farm_d; $f_d++)
{
    $far_d=mysql_fetch_array($farm_del);
    $farm_name= $far_d['farm_name'];
    $farm_id= $far_d['farm_id'];
    echo "&nbsp;&nbsp;&nbsp; $f_d - <a href='farm.php?op=6&farm_id=$farm_id'>$farm_name
<br><br></a>";
}
        }
        else if($op==6) {
            $del_f=mysql_query("delete from farm where farm_id = '$farm_id' ");
            if( $del_f)
echo "deleted farm ";
else
echo "Error..no delete";
        }
        else {
            echo "ERROR : general error....."; }?>
        <div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a>
</div></tr></table><div align="left"></div></body></html>

```

6.2.6 Farmer.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();

```



```

        <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Farmer
ID</strong></span></td>
        <td><input name="farmer_id" type="text" id="farmer_id" size="45"> <br>
        <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Farmer
Name</strong></span></td>
        <td><input name="farmer_name2" type="text" id="farmer_name2" size="45">
        <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span
class="style1"><strong>&nbsp;&nbsp;&nbsp;Surname</strong></span></td>
        <td><input name="farmer_surname2" type="text" id="farmer_surname2"
size="45"> <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span
class="style1"><strong>&nbsp;&nbsp;&nbsp;Address</strong></span></td>
        <td><input name="farmer_address2" type="text" id="farmer_address2"
size="45"> <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span
class="style1"><strong>&nbsp;&nbsp;&nbsp;Telephone</strong></span></td>
        <td><input name="farmer_tel2" type="text" id="farmer_tel2" size="45">
        <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Mobil
Phone</strong></span></td>
        <td><input name="farmer_mobil_tel2" type="text" id="farmer_mobil_tel2"
size="45"> <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Fax </strong></span></td>
        <td><input name="farmer_fax2" type="text" id="farmer_fax2" size="45">
        <br> <br> </td> </tr>
        <tr align="left" valign="top">
        <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;E-mail</strong></span></td>
        <td><input name="farmer_mail2" type="text" id="farmer_mail2" size="45">
        <br> </td> </tr>
        <tr align="left" valign="top">
        <td>&nbsp;&nbsp;&nbsp;</td>
        <td><br> <input name="submit2" type="submit" value=" Enter "></td>
</tr> </table>
<form name="form1" method="post" action="farmer.php?op=1">
</form>
<?
    } else if($op==1)    {
$new_farmer=mysql_query("insert into farmer values
('$farmer_id','$farm_id','$farmer_name','$farmer_surname','$farmer_address','$farmer_t
el','$farmer_mobil_tel','$farmer_fax','$farmer_mail')");
    if ($new_farmer)
echo "added new farmer";

```



```

else echo "ERROR!!! not added";
    } else if($op==2) {
        $all_farmer=mysql_query("select * from farmer");
        $num_farmer=mysql_num_rows($all_farmer);
        for($ff=1; $ff<=$num_farmer; $ff++)
        {
            $farmer=mysql_fetch_array($all_farmer);
            $f_id = $farmer['farmer_id'];
            $f_name = $farmer['farmer_name'];
            $f_sname = $farmer['farmer_surname'];
            echo "$ff - <a href='farmer.php?op=3&farmer_id=$f_id'>ID : $f_id - $f_name
            $f_sname</a> <br><br>";
        } } else if($op==3) {
            $farmer_ed=mysql_query("select * from farmer where
            farmer_id='$farmer_id'");
            $f=mysql_fetch_array($farmer_ed);
            $farmer_name = $f['farmer_name'];
            $farmer_surname = $f['farmer_surname'];
            $farmer_address = $f['farmer_address'];
            $farmer_tel = $f['farmer_tel'];
            $farmer_mobil_tel = $f['farmer_mobil_tel'];
            $farmer_fax = $f['farmer_fax'];
            $farmer_mail = $f['farmer_mail'];

            ?>
            <form name="form1" method="post" action="farmer.php?op=4&farmer_id=<?
            echo "$farmer_id"; ?>">
            <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr align="left" valign="top">
            <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Name</strong></span></td>
            <td><input name="farmer_name" type="text" id="farmer_name" value="<?
            echo "$farmer_name"; ?>" size="45"> <br> <br> </td> </tr>
            <tr align="left" valign="top"> <td><span
            class="style1"><strong>&nbsp;&nbsp;&nbsp;Surname</strong></span></td>
            <td><input name="farmer_surname" type="text" id="farmer_surname"
            value="<? echo "$farmer_surname"; ?>" size="45">
            <br> <br> </td> </tr>
            <tr align="left" valign="top">
            <td><span
            class="style1"><strong>&nbsp;&nbsp;&nbsp;Address</strong></span></td>
            <td><input name="farmer_address" type="text" id="farmer_address" value="<?
            echo "$farmer_address"; ?>" size="45">
            <br> <br> </td> </tr>
            <tr align="left" valign="top">
            <td><span
            class="style1"><strong>&nbsp;&nbsp;&nbsp;Telephone</strong></span></td>
            <td><input name="farmer_tel" type="text" id="farmer_tel" value="<? echo
            "$farmer_tel"; ?>" size="45">
            <br> <br> </td> </tr> <tr align="left" valign="top">
            <td><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Mobil</strong></span></td>

```


6.2.7 Farmer_report.php

```
<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--.style1 {color: #000000}
.style2 {color: #000000; font-weight: bold; }
--></style></head>
<body bgcolor="#ffb64a">
<table width="650" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
<tr align="left" valign="top">
<td width="634" height="33"><strong><font color="#FFFFFF" size="4"><div
align="center">REPORT</div></font><font color="#BDE6EA" size="4"><br>
<br> <br> <br>
</font><font size="4"><span class="style1">Admin Name :</span>
</font></strong><? echo $administrator; ?></td>
<td width="146" align="center" valign="top"> <a href="logout.php"><font
size="2"><strong><em><font
color="#FF0000">Logout</font></em></strong></font></a><font
color="#FF0000"><br>
</font> </td> </tr>
<tr align="left" valign="top">
<td height="156" colspan="2">
<? $milk_admin_no=mysql_query("select * from admin where admin_name =
'administrator' ");
$adm_no=mysql_fetch_array($milk_admin_no);
$f_id = $adm_no['farm_id'];
if(!$sop)
{
?><br> <strong><br>
<a href="farmer_report.php?op=1"><span class="style1">Show Incoming Milks to
Accept</span></a></strong><font color="#BDE6EA"><br>
</font> <br> <br> <br>
<a href="farmer_report.php?op=2" class="style2">Show Accepted Milks</a>
<font color="#BDE6EA">
<? } else if($sop==1){
$milk_acc=mysql_query("select * from milk where verify ='0' and farm_id='$f_id'
order by tarih desc");
$num_acc=mysql_num_rows($milk_acc);
echo "<br><br>There are $num_acc milk incoming wait to accept <br><br>";
$sum_c=0;
```



```

echo "$mac - Farm ID : $f_id - Cow Milk : $cow Sheep Milk : $sheep Goat Milk :
$goat date:$ac_date <br><br>";
}
$total_accepted=$sum_ca+$sum_sa+$sum_ga;
echo "Accepted Milk<br> $sum_ca KG COW <br> $sum_sa KG SHHEP <br>
$sum_ga KG GOAT<br><br>";
echo "Total $total_accepted KG Milk Accepted";
}?? </font>
<div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a> </div>
</tr></table><div align="left"></div></body></html>

```

6.2.8 Milk_accept.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<style type="text/css">
a:link {text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold;}
a:visited {text-decoration: none;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
font-weight: bold;}
a:active {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #000000;
text-decoration:none;
font-weight: bold;}
a:hover {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 7pt;
color: #ffffff;
text-decoration:none;
font-weight: bold;}</style>
<body bgcolor="#ffb64a">
<table width="545" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
<tr align="left" valign="top">
<td width="609" height="33">
<div align="center" class="style2"><font size="4">MILK ACCEPT</font></div>
<strong><font color="#BDE6EA" size="4"><br> <br> <br>

```



```

        <td align="center"><span class="style1"><strong>Cow
Milk</strong></span></td>
        <td align="center"><span class="style1"><strong>Sheep
Milk</strong></span></td>
        <td align="center"><span class="style1"><strong>Goat Milk<br>
        <br> </strong></span></td> </tr>
<tr align="left" valign="top">
    <td><u><? echo "$f_id"; ?></u></td>
    <td align="center"><span class="style1"><? echo "$cow"; ?> KG</span></td>
    <td align="center"><span class="style1"><? echo "$sheep"; ?>
KG</span></td>
    <td align="center"><span class="style1"><? echo "$goat"; ?> KG<br>
    <br> </span></td> </tr>
<tr align="left" valign="top">
    <td><span class="style1"><em>Accept Value :</em></span></td>
    <td align="center"><input name="cow_milk" type="text" id="cow_milk"
size="15"> </td>
    <td align="center"><input name="sheep_milk" type="text" id="sheep_milk"
size="15"> </td>
    <td align="center"><input name="goat_milk" type="text" id="goat_milk"
size="15"> <br> <br> </td> </tr>
<tr align="left" valign="top">
    <td><span class="style1"><em>Milk Prices
    <input name="farm_id" type="hidden" value="<? echo "$f_id"; ?>"
id="farm_id">
    </em></span></td>
    <td align="center"><span class="style1"><? echo "$cow_p"; ?>
YTL</span></td>
    <td align="center"><span class="style1"><? echo "$sheep_p"; ?>
YTL</span></td>
    <td align="center"><span class="style1"><? echo "$goat_p"; ?>
YTL</span></td> </tr>
<tr align="left" valign="top">
    <td>&nbsp;</td>
    <td align="right"><br> <br>
    <input type="submit" name="Submit" value=" Accept "> </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td> </tr> </table> </form>
<?
}
else if($op==2)
{
    $accepted_milk=mysql_query("insert into milk_accepted values (
NULL,$produce_no,'$farm_id',$cow_milk,$sheep_milk,$goat_milk,now())");
    $accepted_verify=mysql_query("update milk set verify='1' where
produce_no=$produce_no ");
    if($accepted_milk)
    { echo "Accepted New Milk Incomings";
    echo "<br><br>";
}

```

```

        $milk_price=mysql_query("select * from milk_price limit 0,1");
        $price=mysql_fetch_array($milk_price);
$cow_p = $price[cow_price];
$sheep_p = $price[sheep_price];
$goat_p = $price[goat_price];
$sc_p=$cow_milk*$cow_p;
$ss_p=$sheep_milk*$sheep_p;
$g_p=$goat_milk*$goat_p;
$total_p = $sc_p+$ss_p+$g_p;
echo "you will get $sc_p YTL for COW milk <br><br>";
echo "you will get $ss_p YTL for SHEEP milk <br><br>";
echo "you will get $g_p YTL for GOAT milk <br><br>";
echo "you will get $total_p YTL TOTALLY <br><br>";
        } else {
            echo "ERROR";
            echo "<br><br>";
        }
    }
?>
<div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a> </div>
</tr></table><div align="left"></div></body></html>

```

6.2.9 Region.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--.style1 {color: #0000FF}
.style3 {color: #0000FF; font-weight: bold; }
.style4 {color: #FF0000;
        font-weight: bold;}
.style5 {color: #000000}
--></style></head>
<body bgcolor="#ffb64a">
<table width="545" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
<tr align="left" valign="top">
<td width="605" height="33"><strong><font color="#FFFFFF"
size="4"></font></strong>
<div align="center" class="style4 style5"><font size="4">REGION</font></div>
<strong></strong><strong><font color="#BDE6EA" size="4"><br>
<br> <br> <br>

```



```
</font><font size="4"><span class="style1">&nbsp;&nbsp;&nbsp;Admin Name :</span>
</font></strong><? echo $administrator; ?></td>
<td width="175" align="center" valign="top"> <a href="logout.php"><font
size="2"><strong><em><font
color="#FF0000">Logout</font></em></strong></font></a><font
color="#FF0000"><br> </font> </td> </tr>
<tr align="left" valign="top">
<td height="156" colspan="2">
<? if(!$op){      ?>
<form name="form1" method="post" action="region.php?op=1">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <br>    <tr align="left" valign="top">
        <td width="13%"><span class="style1"><strong>&nbsp;&nbsp;&nbsp;Region
</strong></span></td>
        <td width="87%">
            <select name="city">
                <option value="">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~Select a
Region&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</option> <?
                $sor_region=mysql_query("select * from region where city = town ");
                $num_region=mysql_num_rows($sor_region);
                for($i=1; $i<=$num_region; $i++)
                {
                    $reg=mysql_fetch_array($sor_region);
                    $city_name= $reg['city'];
                    echo "<option value='$city_name'>$city_name</option>";
                } ?> </select> <br> <br> </td> </tr>
                <tr align="left" valign="top"> <td><span
class="style1"><strong>&nbsp;&nbsp;&nbsp;Location</strong></span></td>
                <td><input type="text" name="town"></td>
            </tr> <tr align="left" valign="top">
                <td>&nbsp;&nbsp;&~</td> <td><br>
                <input name="submit" type="submit" value=" Enter " ></td>
            </tr> </table> </form> <? }
            else if($op==1)
            {
                if (($town != "") && ($city != ""))
                {
                    $add_city=mysql_query("insert into region values ( NULL,'$city','$town')");
                    if ($add_city)
                    echo "add new city";
                    else echo "ERROR!!! not added";
                }
                else
                echo "Error empty location or City";
            }
            else if($op==2)
            {?><br>
                <font color="#BDE6EA">&nbsp;&nbsp;&~</font><span
class="style1"><strong>&nbsp;&nbsp;&~Region</strong> :</span>
                <select name="city"
```

```

        onChange="window.open(this.options[this.selectedIndex].value,'_parent') ">
        <option disabled>Select a Region</option>
        <option <? if($city=='Lefkoşa') echo "selected"; ?>
value="region.php?op=2&city=Lefkoşa">Lefkoşa</option>
        <option <? if($city=='Lefke') echo "selected"; ?>
value="region.php?op=2&city=Lefke">Lefke</option>
        <option <? if($city=='Girne') echo "selected"; ?>
value="region.php?op=2&city=Girne">Girne</option>
        <option <? if($city=='Magosa') echo "selected"; ?>
value="region.php?op=2&city=Magosa">Magosa</option>
        <option <? if($city=='iskele') echo "selected"; ?>
value="region.php?op=2&city=iskele">iskele</option>
    </select>
    <br><br><?
$towns=mysql_query("select * from region where city = '$city' AND town != '$city'");
$num_town=mysql_num_rows($towns);
for($t=1; $t<=$num_town; $t++)
{
    $town=mysql_fetch_array($towns);
    $t_name = $town['town'];
    $c_no = $town['city_no'];
    echo "$t - <a href='region.php?op=3&city_no=$c_no'> $t_name</a> <br>";
} } else if($op==3) {
    $towns2=mysql_query("select * from region where city_no = $city_no ");
    $town2=mysql_fetch_array($towns2);
    $town_name = $town2['town']; <?
        <form name="form1" method="post" action="region.php?op=4&city_no=<?
echo "$city_no"; ?> ">        <br><br> <br>
        <span class="style1"><strong>&nbsp;&nbsp;&nbsp;Location</strong></span>
        <input type="text" name="town" value="<? echo "$town_name" ?>"><br><br>
        <input type="submit" value=" enter "> </form>
<? } else if($op==4) {
    $supt=mysql_query("update region set town='$town' where city_no=$city_no ");
    if($supt){ echo "<br>";
    echo "town name is updated";
    } else { echo "ERROR...no updt"; } } else if($op==5) { ?> <br>
        <span class="style3">&nbsp;&nbsp;&nbsp;Region : </span>
        <select name="city"
            onChange="window.open(this.options[this.selectedIndex].value,'_parent') ">
            <option disabled>Select a Region</option>
            <option <? if($city=='Lefkoşa') echo "selected"; ?>
value="region.php?op=5&city=Lefkoşa">Lefkoşa</option>
            <option <? if($city=='Lefke') echo "selected"; ?>
value="region.php?op=5&city=Lefke">Lefke</option>
            <option <? if($city=='Girne') echo "selected"; ?>
value="region.php?op=5&city=Girne">Girne</option>
            <option <? if($city=='Magosa') echo "selected"; ?>
value="region.php?op=5&city=Magosa">Magosa</option>

```



```

        <option <? if($city=='iskele') echo "selected"; ?>
value="region.php?op=5&city=iskele">iskele</option>
    </select> <br><br>
    <?
$towns3=mysql_query("select * from region where city = '$city' AND town != '$city'");
$num_town3=mysql_num_rows($towns3);
for($t=1; $t<=$num_town3; $t++)
{
    $town3=mysql_fetch_array($towns3);
    $t_n = $town3['town'];
    $ci_no = $town3['city_no'];
    echo "$t - <a href='region.php?op=6&city_no=$ci_no' $t_n</a> <br>";
} } else if($op==6) {
    $delete_c=mysql_query("delete from region where city_no=$city_no");
    if($delete_c)
    {echo "<br>";
    echo "town name is deleted";
    }
    else {
    echo "ERROR...no delete";
    } }
    else {
    echo "error";
    }
    ?>
    <div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a> </div>
</tr></table><div align="left"></div></body></html>

```

6.2.10 Milk_farm.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!-- style1 {color: #000000}
.style2 {color: #000000;
        font-weight: bold;}
--></style></head>
<body bgcolor="#ffb64a">
<table width="780" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
    <tr align="left" valign="top">

```

```

<td width="589" height="33"><strong><font color="#FFFFFF" size="4"><div
align="center">MILK FARM</div></font><font color="#BDE6EA" size="4"><br>
<br> <br>
</font><font size="4"><span class="style1">Admin Name :</span>
</font></strong><? echo $administrator; ?><br><br></td>
<td width="191" align="center" valign="top"> <a href="logout.php"><font
size="2"><strong><em><font
color="#FF0000">Logout</font></em></strong></font></a><font
color="#FF0000"><br>
</font> </td> </tr>
<tr align="left" valign="top">
<td height="156" colspan="2">
<?
    if(!$op)
    {
        ?>
        <form name="form1" method="post" action="milk_farm.php?op=1">
        <table width="100%" border="0" cellpadding="0" cellspacing="0">
        <tr align="left" valign="top">
            <td width="21%"><span class="style1"><strong>Farm Name
</strong></span></td>
            <td width="38%"><select name="farm_id">
                <option disabled>Select a Farm</option>
                <?
                    $sor_f=mysql_query("select * from farm ");
                    $num_f=mysql_num_rows($sor_f);
                    for($t=1; $t<=$num_f; $t++)
                    {
                        $farm=mysql_fetch_array($sor_f);
                        $f_name = $farm['farm_name'];
                        $f_id = $farm['farm_id'];
                        echo "<option value='$f_id'>ID : $f_id - $f_name</option>";
                        $milk_price=mysql_query("select * from milk_price limit
0,1");
                        $price=mysql_fetch_array($milk_price);
                        $cow_p = $price[cow_price];
                        $sheep_p = $price[sheep_price];
                        $goat_p = $price[goat_price]; ?>
                    }
                </select> <br> <br> </td>
                <td colspan="2" align="center"><strong>Milk Price List of This Month
</strong></td> </tr>
                <tr align="left" valign="top">
                    <td><span class="style2">Cow Milk </span></td>
                    <td><input name="cow_milk" type="text" > <br> <br> </td>
                    <td width="11%"><strong>Cow : </strong></td>
                    <td width="30%"><? echo "$cow_p"; ?> YTL</td>
                </tr>
                <tr align="left" valign="top">
                    <td><span class="style1"><strong>Sheep Milk</strong></span></td>
                    <td><input name="sheep_milk" type="text" > <br> <br> </td>
                    <td><strong>Sheep : </strong></td>
                    <td><? echo "$sheep_p"; ?> YTL</td>
                </tr>
            </table>
        </form>
    }
    ?>
</td>
</tr>

```



```

</tr>      <tr align="left" valign="top">
  <td><span class="style1"><strong>Goat Milk</strong></span></td>
  <td><input name="goat_milk" type="text"> <br> <br> </td>
  <td><strong>Goat : </strong></td>
  <td><? echo "$goat_p"; ?> YTL</td>
</tr>      <tr align="left" valign="top">
  <td>&nbsp;</td>
  <td><input name="submit" type="submit" value=" Enter "></td>
  <td>&nbsp;</td> </tr> </table> </form>
  <?
  }
  else if($op==1)
  {
    $add_to_milk=mysql_query("insert into milk values
(NULL,$farm_id,$cow_milk,$sheep_milk,$goat_milk,now(), '0')");
    if ($add_to_milk)
    {
      echo "add new milk<br><br>"; }
  }
  else
  {
    echo "ERROR!!! not added";
  }
}

?>
<div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a> </div>
</tr></table><div align="left"></div></body></html>

```

6.2.11 Animal.php

```

<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="stil/style_admin.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--.style1 { color: #0000FF;
      font-weight: bold;}
.style2 {color: #000000;
      font-weight: bold; }
--></style></head>
<body bgcolor="#ffb64a">
<table width="545" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
  <tr align="left" valign="top">

```

```
<td width="418" height="33"><div align="center"><span  
class="style2">ANIMAL</span><br>  
<br></div>  
<? echo $administrator; ?></td>  
<td width="127" align="center" valign="top"><a href="logout.php"><font  
color="#FF0000" size="2"><strong><em>Logout</em></strong></font></a><br>  
<br></td> </tr>  
<tr align="left" valign="top">  
<td height="156" colspan="2">  
<? if(!$op)  
{?  
<br> <br>  
<form name="form1" method="post" action="animal.php?op=1">  
<table width="100%" border="0" cellpadding="0" cellspacing="0">  
<tr align="left" valign="top">  
<td width="19%"><span class="style1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Farm Name  
</span></td>  
<td width="81%"><select name="farm_id">  
<option disabled>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Select a  
Farm&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&,</option>  
<?  
$sor_f=mysql_query("select * from farm ");  
$num_f=mysql_num_rows($sor_f);  
for($t=1; $t<=$num_f; $t++)  
{  
$farm=mysql_fetch_array($sor_f);  
$f_name = $farm['farm_name'];  
$f_id = $farm['farm_id'];  
echo "<option value='$f_id'>ID : $f_id - $f_name</option>";  
} ?>  
</select> <br> <br> </td>  
</tr> <tr align="left" valign="top">  
<td><span class="style1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Cow ID</span></td>  
<td><input name="cow_id" type="text" id="cow_id"> <br> <br> </td>  
</tr> <tr align="left" valign="top">  
<td><span class="style1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Sheep ID</span></td>  
<td><input name="sheep_id" type="text" id="sheep_id"> <br> <br> </td>  
</tr> <tr align="left" valign="top">  
<td><span class="style1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&Goat ID</span></td>  
<td><input name="goat_id" type="text" id="goat_id"> <br> <br> </td>  
</tr> <tr align="left" valign="top">  
<td>&nbsp;&nbsp;&nbsp;&,</td>  
<td><br><input name="submit" type="submit" value=" Enter " ></td>  
</tr> </table> </form> <?  
<? else if($op==1)  
{  
$add_animal=mysql_query("insert into animal values (  
'$farm_id','$cow_id','$sheep_id','$goat_id')");  
if ($add_animal)
```



```

echo "add new aninamls"; else
echo "ERROR!!! not added"; }
    else if($op==2) { ?> <br> <br>
<form name="form2" method="post" action="animal.php?op=3">
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="18%"><span class="style1">Animal ID</span></td>
            <td width="82%"><input name="animal_id" type="text" id="animal_id"></td>
        </tr> <tr> <td>&nbsp;</td> <td><br>
            <input type="submit" name="Submit" value=" Search "></td>
        </tr> </table> </form>
    <? > } else if($op==3) {
        $src_anml=mysql_query("select * from animal where cow_id = '$animal_id'
OR sheep_id = '$animal_id' OR goat_id = '$animal_id' order by farm_id desc");
$num_anml=mysql_num_rows($src_anml);
if($num_anml<1)
{ echo "there is no animal at ID $animal_id";
echo "<br><br>"; }
else {
echo "Founded $num_anml animal at ID $animal_id <br> These Are";
echo "<br><br>"; }
for($t=1; $t<=$num_anml; $t++) {
$anml=mysql_fetch_array($src_anml);
$f_id = $anml['farm_id'];
$cow_id = $anml['cow_id'];
$sheep_id = $anml['sheep_id'];
$goat_id = $anml['goat_id'];
echo "$t - farm : $f_id ";
if($animal_id == $cow_id)
echo "Cow : $animal_id ";
if($animal_id == $sheep_id)
echo " Sheep : $animal_id ";
if($animal_id == $goat_id)
echo " Goat : $animal_id ";
echo "<br><br> ";
} } else if($op==4) { ?>
    <br> <br>
<form name="form2" method="post" action="animal.php?op=5">
    <table width="100%" border="0" cellspacing="0" cellpadding="0"> <tr>
        <td width="18%"><span class="style1">Animal ID</span></td>
        <td width="82%"><input name="animal_id" type="text" id="animal_id"></td>
    </tr> <tr> <td>&nbsp;</td> <td><br>
        <input type="submit" name="Submit" value=" Delete Animal "></td>
    </tr> </table> </form> <? > }
    else if($op==5) {
        $src_anml=mysql_query("select * from animal where cow_id = '$animal_id'
OR sheep_id = '$animal_id' OR goat_id = '$animal_id' order by farm_id desc");
$num_anml=mysql_num_rows($src_anml);
if($num_anml<1)
{ echo "there is no animal at ID $animal_id";

```

```

echo "<br><br>"; } else {
echo "deleted $num_anml animal at ID $animal_id <br> ";
echo "<br><br>"; }
for($t=1; $t<=$num_anml; $t++)
{
$anml=mysql_fetch_array($src_anml);
$f_id = $anml['farm_id'];
$cow_id = $anml['cow_id'];
$sheep_id = $anml['sheep_id'];
$goat_id = $anml['goat_id'];
echo "$t - farm : $f_id ";
if($animal_id == $cow_id)
$delete_cow=mysql_query("update animal set cow_id=" where cow_id = '$animal_id'
");
if($animal_id == $sheep_id)
$delete_sheep=mysql_query("update animal set sheep_id=" where sheep_id =
'$animal_id' ");
if($animal_id == $goat_id)
$delete_goat=mysql_query("update animal set goat_id=" where goat_id = '$animal_id'
"); echo "<br><br> ";
} } ?>
<div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a>
</div></tr></table></body></html>

```

6.2.12 Milk_price

```

<?
require_once("function.php");
require_once("database.php");
session_start();
check_administrator();
$conn=db_connect();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Farm management</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link href="file:///I:/hikmet_Graduation/hikmet/admin/stil/style_admin.css"
rel="stylesheet" type="text/css">
<style type="text/css">
<!--.style1 {color: #0000FF}
.style2 {color: #FF0000}
--></style></head>
<body bgcolor="#ffb64a">
<table width="545" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#D1D1D1">
<tr align="left" valign="top">
<td width="614" height="33"><strong><font color="#FFFFFF" size="4"><div
align="center" class="style2">MILK PRICE</div></font><font color="#BDE6EA"
size="4"><br> <br><br>

```



```

</font><font size="4"><span class="style1">Admin Name :
</span></font></strong><? echo $administrator; ?></td>
<td width="166" align="left" valign="top"> <a href="logout.php"><font
size="2"><strong><em><font color="#FF0000"><font size="2"><strong><em><font
color="#FF0000">Logout</font></em></strong></font></a><font
color="#FF0000"><br>
</font> </td>
</tr> <tr align="left" valign="top">
<td height="156" colspan="2">
<?
if(!$op)
{
?>
<form name="form1" method="post"
action="file:///I:/hikmet_Graduation/hikmet/admin/milk_price.php?op=1">
<br>
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr align="left" valign="top"> <br>
<td width="29%"><span class="style1"><strong>&nbsp;  Cow Milk
Price</strong></span></td>
<td width="71%"><input name="cow_price" type="text">
<span class="style1">YTL</span> <br>
<br> </td> </tr>
<tr align="left" valign="top">
<td><span class="style1"><strong>&nbsp;  Sheep Milk
Price</strong></span></td>
<td><input name="sheep_price" type="text">
<span class="style1"> YTL</span> <br>
<br> </td> </tr>
<tr align="left" valign="top">
<td><span class="style1"><strong>&nbsp;  Goat Milk
Price</strong></span></td>
<td><input name="goat_price" type="text">
<span class="style1"> YTL</span> <br>
<br> </td> </tr>
<tr align="left" valign="top">
<td>&nbsp;  </td>
<td><input name="submit" type="submit" value=" Enter "></td>
</tr> </table> </form><? }
else if($op==1)
{
$sorgulama=mysql_query("insert into milk_price values (
NULL,$cow_price,$sheep_price,$goat_price,now())");
if ($sorgulama)
echo "added new price new limits";
else
echo "ERROR!!! not added";
} ?>
<div align="right"><a href="admin.php">&lt;&lt; Main Page (admin)</a> </div></tr>
</table> <div align="left"></div> </body> </html>

```

6.2.13 Function.php

```
<?
require_once("database.php");
$conn=db_connect();
function admin_login($admin_name, $admin_pass)
{
    $conn = db_connect();
    if (!$conn) return 0;
    $result = mysql_query("select * from admin
        where admin_name='$admin_name'
        and admin_pass='$admin_pass'");

    if (!$result)
        return 0;
    if (mysql_num_rows($result)>0)
        return 1;
    else return 0;
}
function check_administrator()
{
    global $administrator;
    if (session_is_registered("administrator"))
    {
    } else {
        echo "<font size='5' color='red'>ERROR</font><br>";
        echo "Please enter name and password !.<br>";
    }
    ?>
<meta http-equiv="refresh" content="3;URL=index.php">
<? exit; } }
function add_milk($farm_no,$cow,$goat,$sheep)
{
    $sorgulama=mysql_query("insert into milk values
        (NULL,$farm_no,$cow,$goat,$sheep,now())");

    if (!$sorgulama)
        return false;
    return true; }
function
add_milk_price($cow_price,$goat_price,$sheep_price,$cow_limit,$goat_limit,$sheep_l
imit) {
    $sorgulama=mysql_query("insert into milk_price values
        (NULL,$cow_price,$goat_price,$sheep_price,$cow_limit,$goat_limit,$sheep_limit)");
    if (!$sorgulama)
        return false;
    return true; }?>
```

6.2.14 Logout.php

```
<?
require_once("function.php");
session_start();
$eski_user = $administrator;
$result_delete = session_unregister("old_user");
$result_dest = session_destroy();
if (!empty($old_user))
```



```

{
if ($result_delete && $result_dest)
{
echo "You are Exit the System Successfully...<br>";
?>
<meta http-equiv="refresh" content="3;URL=../index.php">
<? } else {
echo "ERROR!!! You are not exit!!!.<br>";
} } else {
echo "<font size='6' color='red'>Attention!!! You didn't enter the system so the
logoff is unsuccessful.<br></font>";
} ?>

```

REFERENCES

1 – <http://www.php.net>

2 – <http://www.phpnuke.org>

3 – <http://www.apache.org>

4 – <http://www.mysql.com>

5 – MySQL in 21 Days Mark Maslakowski