NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Student Registration
Though Interactive Webpage

Graduation Project
COM – 400

Prepared by: Anas Mahmoud Hijazi

Supervised by: Mr. Umit Ilhan

Nicosia - 2004

Acknowledgements

First of all, I would like to thank my family who gave me the opportunity to proceed in my education, and never stopped supporting me when I grew week, and kept encouraging me when I became strong.

I would to say, that I'm very grateful to my supervisor Mr. Umit Ilhan, where he was of a great help to me complete my project, he was of great interest, continuous guidance and cooperation in each step of my project.

Finally I would like to thank all my friends who helped me through my education career, and special thanks to Cem and Alaa these people who taught me PHP programming.

Table of contents

Introduction

In our modern world the need of reliable and fast WebPages became more important day after day, the role of interactive web pages is becoming wider and more useful, not only for companies and official organizations but also for normal internet users.

As the world is evolving and technology is becoming more sophisticated and powerful, the need of creating databases that can deal with numerous data types and tabulating the data in forms to be kept and saved for the purpose of usage by others, this technology applies for many organizations and companies.

The aim of this project is the creation of a website interacting with a database, where the website will be interactive. Using an online student registration example would be a good example for this technology.

Chapter 1. Interactive webpage

1.1 What is an interactive webpage?

An interactive web page can be defined as a dynamic (Database-driven) web page which is opposite to the static webpage, the website is built upon data stored in the servers data base, it is easy to be updated, including that it gives the user the opportunity to change the data in the data base which might be a thousand miles away from the user.

1.2 Why do we need to use an interactive webpage?

There are various types of database-driven (interactive) sites. An E-shop is one of them, and a community portal is another. There are also various online collaboration and online enterprise management solutions, forums, chats (these applications often become parts of portal sites or enterprise applications), and many more. The most remarkable example of a database-driven web application is the search engine.

E-commerce solutions have now become widespread. Presenting goods online, allowing customers to complete their shopping carts and to pay for their goods using a standard gateway such as World Pay or Pay Pal, all this is convenient and saves time, so more and more people are seeing the benefits of online shopping. All this requires implementation of a complicated functionality and high reliability, especially when processing payment transactions. So this work is not to be done by amateurs.

However, many ready-made solutions are available for free that facilitate the task of developing an e-shop. Most of the time, the only thing that needs to be done is to customize an existing script to meet the needs of the client. All the commonly used e-commerce functionality is usually already included and is therefore thoroughly tested by the system supervisor.

So where do the databases come in? The goods, their properties, descriptions, prices, discounts and other information is likely to be stored in one table or several closely related tables. Customers, their personal details, their histories, personal discounts and other things will be in another table (or tables). Databases are capable of storing graphic information, so even the pictures presenting various sorts of goods are here, too.

1.3 Advantages of interactive webpage

The development of such a helpful webpage have a lot of advantages which we can list some of them below;

- Far distance data base access
- Ability of editing database contents
- Applies for a variety of organizations
- Data base graphical interface
- Easy to update
- Fast and reliable

1.4 How an interactive webpage works

In static WebPages when a user requests a document from the server, the server returns an html document, which is processed by the users' computer. The web site hardly provides files that are requested and the user browser makes the decisions about rendering them. figure 1.1 illustrates the operation.

On the other hand when a user requests a file from the server the server processes the document (which is a PHP script in our case) and sends the processed script to the user, this operation consumes less time and it's more secure and reliable. Figure 1.2 illustrates the operation.

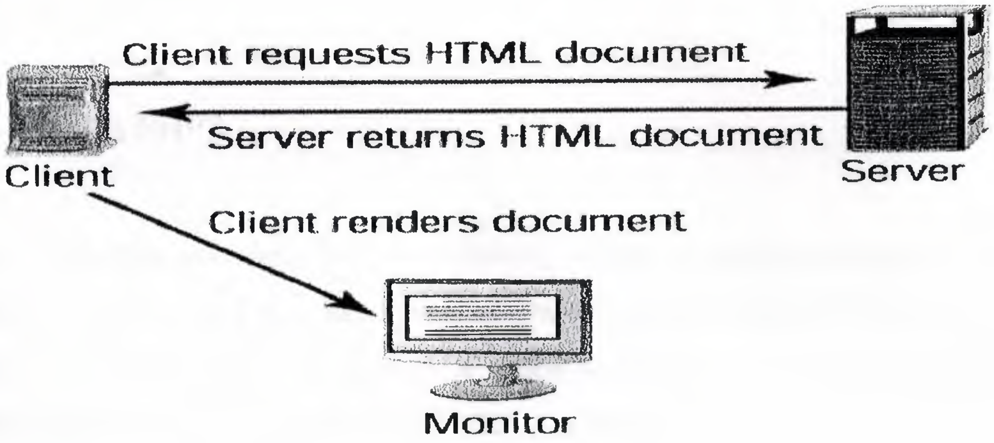
Client requests HTML document
Server returns HTML document
Client
Server
Client renders document
Monitor

Fig-1.1

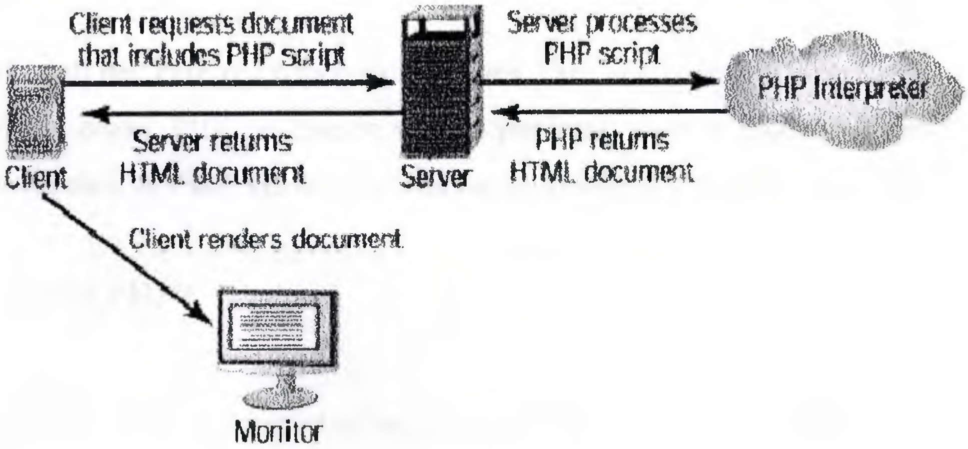
Client requests document that includes PHP script
Server processes PHP script
PHP Interpreter
Server returns HTML document
PHP returns HTML document
Client
Server
Client renders document
Monitor

Fig-1.2

Chapter 2. PHP

2.1 What is PHP?

PHP is simply a web application development tool that is capable of interacting with a variety of databases, it acts like a translator which translates the PHP code to a form that can be displayed at the client side when requesting a page that implements a certain operation, we can also define PHP as a purely a programming language that became very famous by the end of 1998, it is very similar to C++ in a way that if the user knows C++, he will be able to learn PHP in just a few weeks. We can also say that PHP is a server side handled document application, because the entire script is processed by the server and then the server returns a processed HTML document.

You can very well run PHP on Windows machines. PHP is a cross-platform support language, which means, PHP can run on various platforms such as Linux, Windows etc. Web servers such as PWS, IIS and Apache can be configured to work with PHP.

2.1.1 Who wrote PHP?

Rasmus wrote PHP. PHP was started off accidentally, Rasmus was designing his web site, and he had planed to put a hit counter for his site. He decided to write a combination of Perl. There were many enquires about the script which he used. Likewise it slowly improved on, and won the hearts of many programmers.

2.1.2 What is a web application?

A web application, in general, is an application that leverages the ubiquity and ease of communication the Internet provides. A more restricted definition of web application the one that is widely used is an application that uses a web browser as the client. There are many client-side technologies available to most web browsers. In general, the most far-reaching and easily accessed web applications are those that use simple and elegant Hypertext Markup Language (HTML). A few examples that strictly fit the term web application are Web-based banking systems, auctions, and news sites.

2.2 What type of scripting is PHP client or a server side?

PHP is a server-side scripting language. A server-side scripting language is one, which has the capability of executing the script on the server and serving the output as an HTML file. Server-side scripts have the main advantage of interacting with databases, as well as performing all types of server manipulations directly.

Server-side scripting languages can manipulate the data which is received from web forms, anywhere on the Net. But this is not the case with client-side scripts such as JavaScript and VBScript.

However, there are exceptions where, client-side scripts can also be used for server-side scripting, but this is limited to some extent.
Like the other scripting languages we can do so but it is not a compulsory act for us to use PHP with HTML. We can very well get the output with PHP alone, but the expected out put which is exhibited by PHP is not that attractive for display, attractive meaning the formatting which is done by PHP.
The figure below shows us how dose a PHP script works when it's requested by a use on the internet.

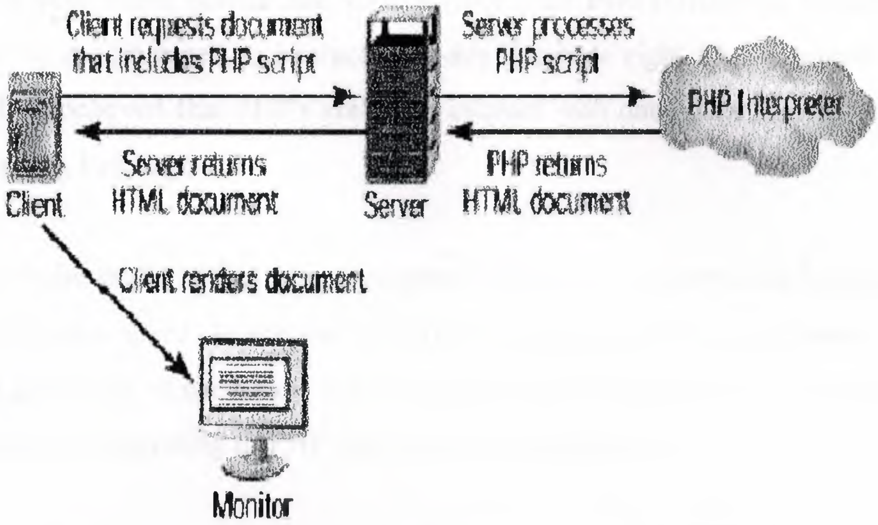
Client requests document that includes PHP script
Server processes PHP script
PHP Interpreter
Server returns HTML document
PHP returns HTML document
Client
Server
Client renders document
Monitor

Fig 2.1

2.3 Advantages of PHP.

- PHP is a cross-platform support language, which means, PHP can run on various platforms such as Linux, Windows etc. Web servers such as PWS, IIS and Apache can be configured to work with PHP.

- Like the other scripting languages you can do so but it is not a compulsory act for you to use PHP with HTML. You can very well get the output with PHP alone, but the expected out put which is exhibited by PHP is not that attractive for display, attractive meaning the formatting which is done by PHP.

- If someone is aware of programming concepts and the application of the logic, then, he/she will find it relatively easy to code any programming language by understanding the syntax. PHP's functions are similar to other scripting languages. The only thing that matters is his/her requirement, how he/she is going to apply the logic and code PHP to his/her logic. The main consideration when migrating from a different language lies with the syntax. PHP syntax is similar to that of the C language. People with a basic understanding of C will find it very easy to migrate to PHP.

- PHP is very much compatible with MySQL and POSTGRESQL Databases. PHP also can be programmed to interact with any database right from a simple text file to DB2. It is believed that PHP's ability to interact with databases is greater than any other scripting languages.

- PHP is believed to be fast when compared with other programming languages, Because of execution speed. In the case of large interactions with the database, PHP really plays a good role of execution in performing the interaction. Now a day major web sites' have been migrating to PHP, because of the performance.

- PHP is the only scripting language, which has won the hearts of many web servers and has been moving on enormously day-by-day, The main advantages of open source software is that, if there is a problem, that particular problem is shared by millions of programmers across the world, but in the case of licensed version scripting languages, it is very limited to a set of a small community. Because innovative ideas are studied by many programmers across the world, new ideas are implemented then and there, making the language strong. There is no problem in using PHP, which is open source.

2.4 The main disadvantages of PHP

Every language/scripting language has its own advantages and disadvantages and PHP also has some disadvantages. Some are direct disadvantages and some are indirect regarding functionality.

When it comes to error handling, it is believed that PHP has a very poor ability but even this disadvantage can be over come using a feasible advantage solution.

Chapter 3. MySQL

3.1 What is MySQL?

MySQL is a type of SQL database management featured in Thelix hosting plans. A database is an organized collection of information that a computer uses to select and display data. Databases can help organize and enhance our site content. Sites with dynamic pages and/or shopping cart software often need an underlying database structure.

MySQL is also a popular database server which is available for Linux, FreeBSD and other flavors of UNIX, and also Win32 platforms. MySQL is often used as a database back-end to PHP web applications or CGIs invoked by Perl database modules.

MySQL has highly configurable user/permissions model as well as network access permissions configuration. MySQL is typically accessed via the client software at the command prompt on a Linux or FreeBSD server. MySQL is much faster than Oracle or Microsoft Access, it's also considered to be the fastest data base server available.

We can also define MySQL as a relational database management system, which means it stores data in separate tables rather than putting all the data in one big area. This adds flexibility, as well as speed. The SQL part of MySQL stands for "Structured Query Language," which is the most common language used to access databases. The

MySQL database server is the most popular open source database in the world. It is extremely fast and easy to customize, due to its architecture. Extensive reuse of code within the software, along with a minimalist approach to producing features with lots of functionality, gives MySQL unmatched speed, compactness, stability, and ease of deployment. Their unique separation of the core server from the storage engine makes it possible to run with very strict control, or with ultra fast disk access, whichever is more appropriate for the situation.

3.2 MySQL features

- Security

A privilege and password system that is very flexible and secure, and allows Host-based verification. Passwords are secure because all passwords are encrypted when we connect to a server.

- Scalability and limits

Handles large databases. We use MySQL Server with databases that contain 50 million records. We also know of users that use MySQL Server with 60,000 tables and about 5,000,000,000 rows.

Up to 32 indexes per table are allowed. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 500 bytes (this may be changed when compiling MySQL Server). An index may use a prefix of a CHAR or VARCHAR column.

- Connectivity

Clients may connect to the MySQL server using TCP/IP sockets on any platform. On Windows systems in the NT family (NT, 2000, or XP), clients may connect using named pipes. On UNIX systems, clients may connect using UNIX domain socket files.

The Connector/ODBC interface provides MySQL support for client programs that use ODBC (Open-Database-Connectivity) connections. For example, we can use MS Access to connect to the MySQL server. Clients may be run on Windows or UNIX. Connector/ODBC source is available. All ODBC 2.5 functions are supported, as are many others.

The Connector/JDBC interface provides MySQL support for Java client programs that use JDBC connections. Clients may be run on Windows or Unix. Connector/JDBC source is available.

- localization

The server can provide error messages to clients in many languages. Full support for several different character sets, including ISO-8859-1 (Latin 1), German, big5, ujis, and more. For example, the Scandinavian characters '^a', 'Äa' and 'Äo' are allowed in table and column names. Unicode support is available for MySQL.

All data is saved in the chosen character set. All comparisons for normal string columns are case-insensitive.

Sorting is done according to the chosen character set (the Swedish way by default). It is possible to change this when the MySQL server is started. To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many deferent character sets that can be specified at compile and runtime.

3.3 MySQL support for large tables

Mysql can support a data base that reaches the size of 8 million terabytes. With this larger allowed table size, the maximum effective table size for MySQL databases now normally are determined by operating system constraints on file sizes, not by MySQL internal limits. The InnoDB storage engine maintains InnoDB tables within a table space that can be created from several files. This allows a table to exceed the maximum individual file size. The table space can include raw disk partitions, which allows extremely large tables. The maximum table space size is 64 TB.

The following table lists some examples of operating system file-size limits:

Operating system	File size limit
Linux-Intel 32-bit	2 GB, much more when using LFS
Linux-Alpha	8 TB
Solaris 2.5.1	2 GB (4 GB possible with patch)
Solaris 2.6	4 GB (can be changed with flag)
Solaris 2.7 Intel	4 GB
Solaris 2.7 UltraSPARC	512 GB

3.4 MySQL log files

MySQL has several different log files that can help US find out what's going on inside mysqld:

Log File	Types of Information Logged to File
The error log	Logs problems encountered starting, running, or stopping MySQLD.
The ISAM log	Logs all changes to the ISAM tables. Used only for debugging the ISAM code.
The query log	Logs established client connections and executed statements.
The update log	Logs statements that change data. This log is deprecated.
The binary log	Logs all statements that change data. Also used for replication.
The slow log	Logs all queries that took more than long_query_time seconds to execute or didn't use indexes.

By default, all logs are created in the mysqld data directory. WE can force mysqld to close and reopen the log files (or in some cases switch to a new log) by flushing the logs. Log flushing occurs when we issue a FLUSH LOGS statement or execute mysqladmin flush-logs or mysqladmin refresh

Chapter 4. HTML

4.1 What is HTML?

HTML (Hypertext Markup Language) the authoring language used to create documents on the World Wide Web. HTML is similar to SGML, although it is not a strict subset.

HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with (enter here what document is about) and ends with. All the information you'd like to include in your Web page fits in between the tags, and is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end. HTML is a formal Recommendation by the World Wide Web Consortium (W3C) and is generally adhered to by the major browsers, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes. The current version of HTML is HTML 4.0. However, both Internet Explorer and Netscape implement some features differently and provide non-standard extensions. Web developers using the more advanced features of HTML 4 may have to design pages for both browsers and send out the appropriate version to a user.

Significant features in HTML 4 are sometimes described in general as dynamic HTML. What is sometimes referred to as HTML 5 is an extensible form of HTML called Extensible Hypertext Markup Language (XHTML).

4.2 What dose HTML do?

HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with (enter here what document is about) and ends with. All the information that we like to include in our Web page fits in between the tags. There are hundreds of other tags used to format and layout the information in a Web page. Tags are also used to specify hypertext links. These allow Web developers to direct users to other Web pages with only a click of the mouse on either an image or words.

4.3 Dynamic HTML

Dynamic HTML is a collective term for a combination of new Hypertext Markup Language (HTML) tags and options, which will let us create Web pages more animated and more responsive to user interaction than previous versions of HTML. Much of dynamic HTML is specified in HTML 4.0. Simple examples of dynamic HTML pages would include:

- Having the color of a text heading change when a user passes a mouse over it.
- Allowing a user to "drag and drop" an image to another place on a Web page.

Dynamic HTML can allow Web documents to look and act like desktop applications or multimedia productions. The features that constitute dynamic HTML are included in Netscape Communications' latest Web browser, Navigator 4.0 (part of Netscape's Communicator suite), and by Microsoft's browser, Internet Explorer 4.0. While HTML 4.0 is supported by both Netscape and Microsoft browsers, some additional capabilities are supported by only one of the browsers. The biggest obstacle to the use of dynamic HTML is that, since many users are still using older browsers, a Web site must create two versions of each site and serve the pages appropriate to each user's browser version. The Concepts and Features in Dynamic HTML
Both Netscape and Microsoft support:

- An object-oriented view of a Web page and its elements Cascading style sheets and the layering of content Programming that can address all or most page elements Dynamic fonts An Object-Oriented View of Page Elements

Each page element (division or section, heading, paragraph, image, list, and so forth) is viewed as an "object." (Microsoft calls this the "Dynamic HTML Object Model."

Netscape calls it the "HTML Object Model." W3C calls it the "Document Object Model.") For example, each heading on a page can be named, given attributes of text style and color, and addressed by name in a small program or "script" included on the page. This heading or any other element on the page can be changed as the result of a specified event such a mouse passing over or being clicked or a time elapsing or an image can be moved from one place to another by "dragging and dropping" the image object with the mouse. (These event possibilities can be viewed as the reaction capabilities of the element or object.) Any change takes place immediately (since all variations of all elements or objects have been sent as part of the same page from the Web server that sent the page). Thus, variations can be thought of as different properties of the object. Not only can element variations change text wording or color, but everything contained within a heading object can be replaced with new content that includes different or additional HTML as well as different text. Microsoft calls this the "Text Range technology."

4.3.1 Style Sheets and Layering

These describes the default style characteristics (including the page layout and font type style and size for text elements such as headings and body text) of a document or a portion of a document. For Web pages, a style sheet also describes the default background color or image, hypertext link colors, and possibly the content of page. Style sheets help ensure consistency across all or a group of pages in a document or a Web site.

Dynamic HTML includes the capability to specify style sheets in a "cascading style sheet" fashion (that is, linking to or specifying different style sheets or style statements with predefined levels of precedence within the same or a set of related pages). As the result of user interaction, a new style sheet can be made applicable and result in a change of appearance of the Web page. You can have multiple layers of style sheet within a page, a style sheet within a style sheet within a style sheet. A new style sheet may only vary one element from the style sheet above it.

Layering is the use of alternate style sheets or other approaches to vary the content of a page by providing content layers that can overlay (and replace or superimpose on) existing content sections. Layers can be programmed to appear as part of a timed presentation or as the result of user interaction. In Internet Explorer 4.0, Microsoft implements layers through style sheets. Netscape supports the style sheet approach but also offers a new HTML ... tag set (that Microsoft does not support). Both approaches are being considered by the W3C Working Committee and both companies say they will support whatever W3C decides will be the recommended approach.

4.3.2 Programming

Although JavaScript, Java applet, and ActiveX controls were present in previous levels of Web pages, dynamic HTML implies an increased amount of programming in Web pages since more elements of a page can be addressed by a program.

Dynamic Fonts

Netscape includes dynamic fonts as part of dynamic HTML. This feature of Netscape's Navigator browser in its Communicator suite lets Web page designers include font files containing specific font styles, sizes, and colors as part of a Web page and to have the fonts downloaded with the page. That is, the font choice no longer is dependent on what the browser provides.

4.4 The work done using HTML

Using HTML in this project gave us a lot of good results starting from the first page till the last one, we will be listing the pages made using HTML and describing the use of each page.

4.4.1 The main page

Here in the main page is the instructors' login session. The instructor will have to insert his username and password that should be provided by the page admin.

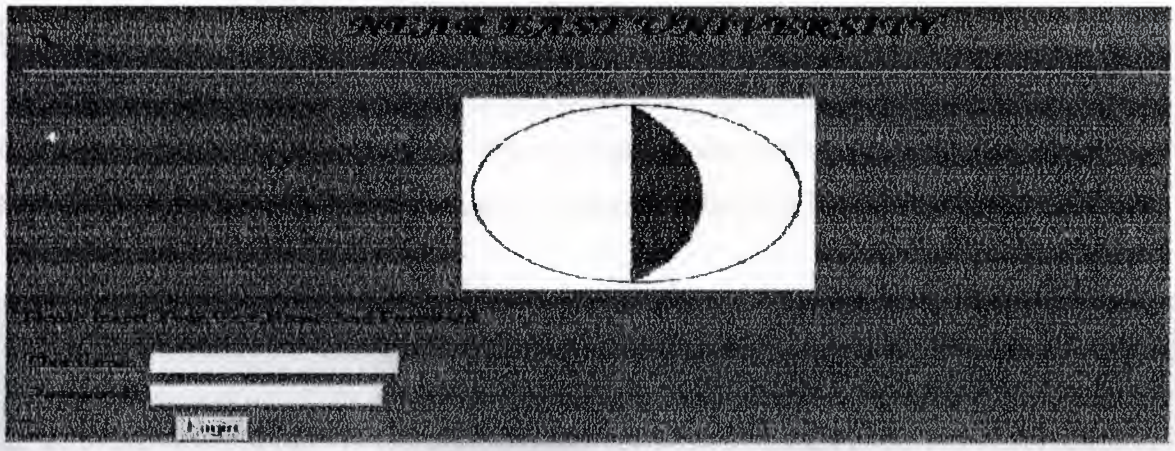

Fig-4.1

4.4.2 Instructors' work area

In this session the instructor will have the ability to do a variety of jobs which we can list as follows:

- Register a new student or delete a registered student.
- Register student coursed.
- Insert/edit student marks.
- Delete registered courses.
- Calculate the student CGBA and list his courses.

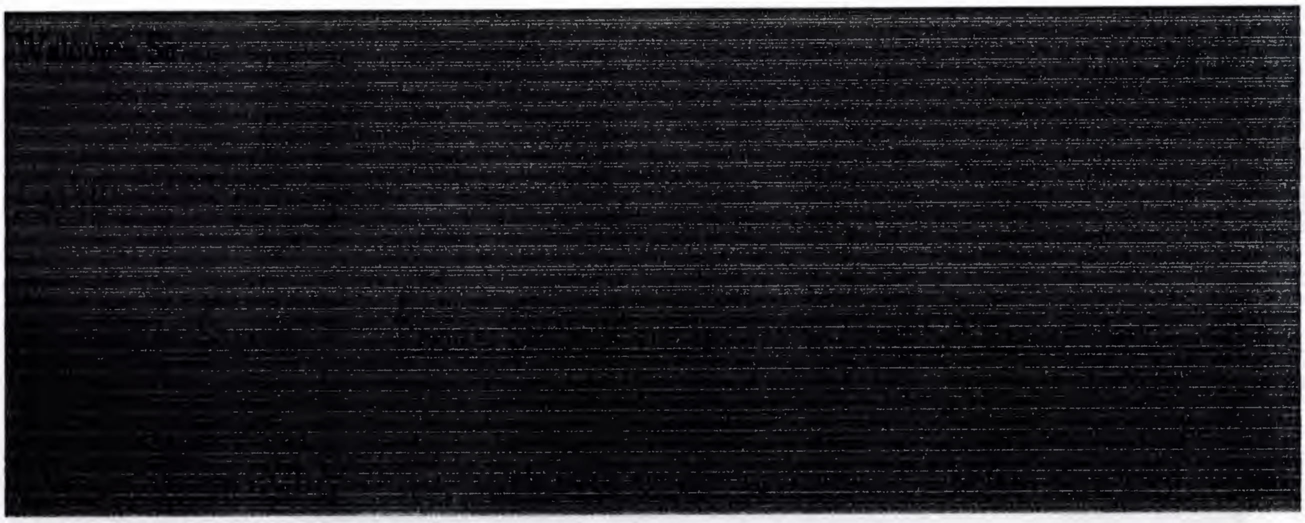

Fig-4.2

4.4.3 Registering a new student

in this session the instructor will be able to register new student, and that is simply done by filling the student name, surname, student number and his/her class.

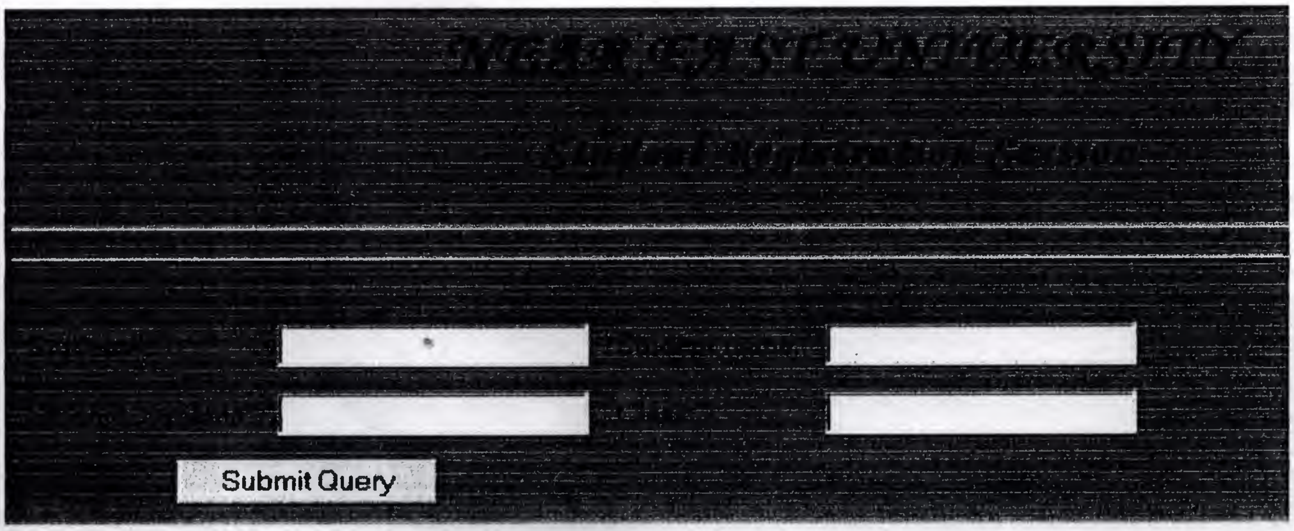

Fig- 4.3

4.4.4 Student deletion session

Here the instructor can delete students that he registered in any registration mistake takes place, he /she can do that simply by inserting the student number.

Fig-4.4

4.4.5 Course registration session

In this session the student courses will be registered it's simply done by, inserting the course code, course credit, student number and course grade if available.

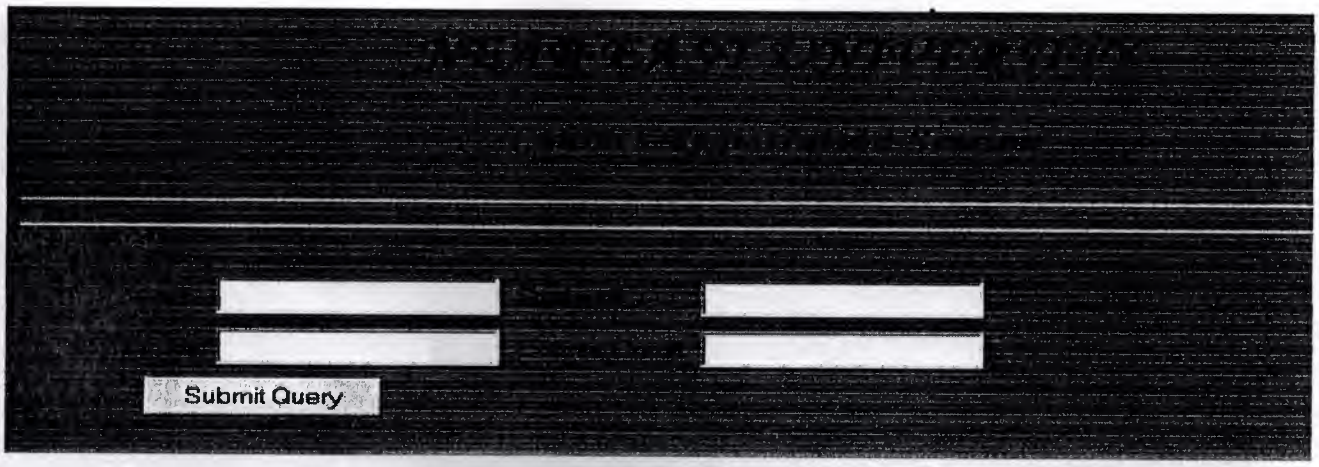

Fig-4.5

4.4.6 Grade insertion session

Here the instructor will be able to insert the grad or modify it in case he need to make a modification.

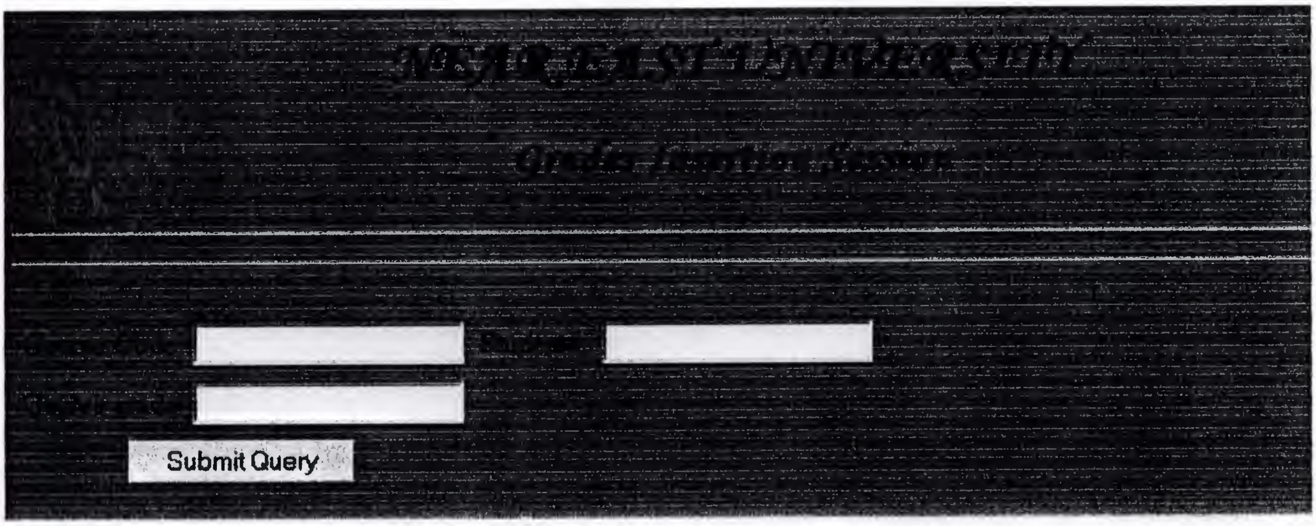

Fig-4.6

4.4.7 Courses deletion session

In this part of the page the instructor will be able to delete a registered course in case the student needs to drop or withdraw the course he/she already registered.

Fig-4.7

4.4.8 CGBA calculation session and course listing

Finally in this session the instructor will be able to easy calculate the students CGBA and view the students registered courses and display their results, and that is simply done by inserting the students' number.

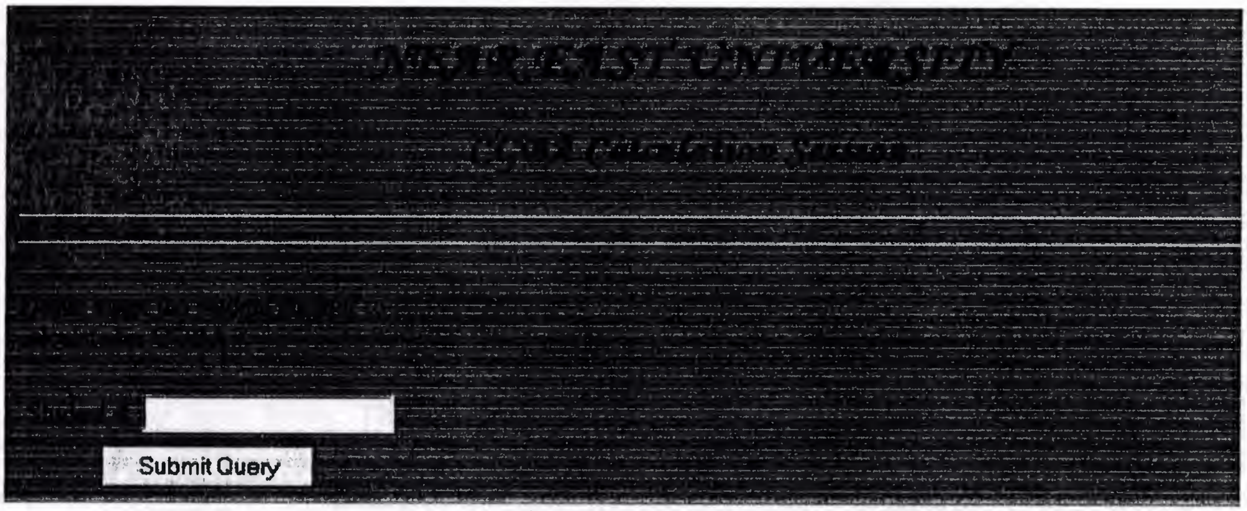

Fig-4.8

The figure below illustrates a sample result for the operation of calculating the student CGBA and viewing the student registered coursed and their results.

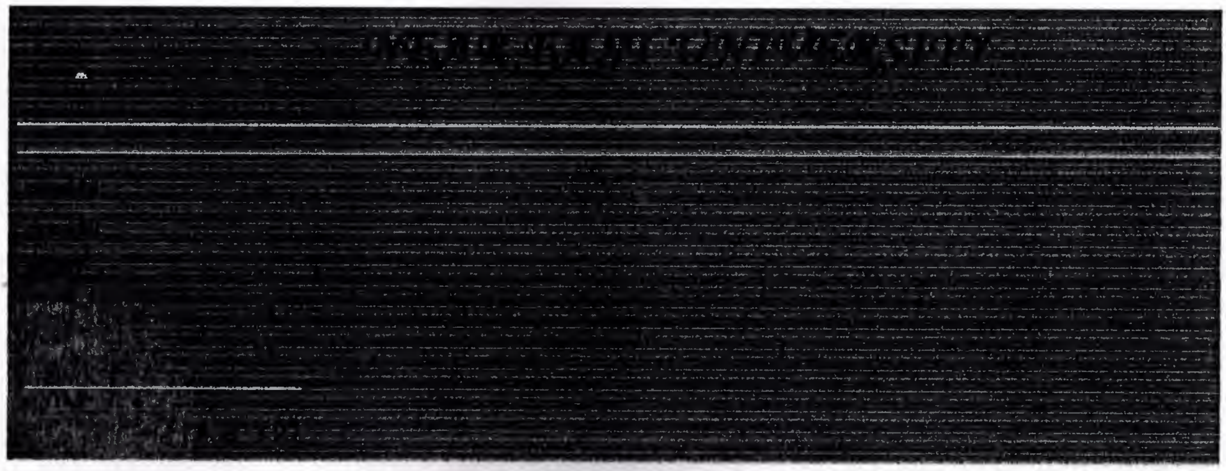

Fig-4.9

Chapter 5. Work accomplished

In this project we have been using three applications, PHP, HTML and MySQL, to create an interactive webpage for university student registration. The application was developed in a way that the instructors will log into a page where they can register students, insert/edit their marks, register/un-register students, and register/drop or withdraw courses. Our codes were written and tested using realistic data. The outputs were examined and according to the project planning no error was found.

5.1 The HTML codes

These are the html codes used in some parts of our application. Mostly HTML was used to make the visual interface for our application, including the implementation of most of our input forms. The codes were inserted into PHP scripts to provide the factor of interactivity.

5.1.1 The main page (login_neu.php)

This is the main page code. It is implemented to allow the instructors login to the session where they can do the operations listed in session 5.

```
<html>


<body bgcolor="#008080">
<p align="center">	<font face="Monotype Corsiva" size="6"><b> NEAR EAST
UNIVERSITY </b></font></p>
<hr>
<p align="center"><img border="0" src="NEU.BMP" width="219"
height="191"></p>
</body>
</html>

Please Insert Your User Name And Password<form action=logger.php method=post>
<table cellpadding=3 height="102">
```

```
<tr><td height="22">Username: <td height="22"><input name=user size="20"></tr>
<tr><td height="22">Password: <td height="22">
 <input name=pass type=password size="20"></tr>
<tr><td align=center colspan=2 height="34"><input type = submit
value="Login"></tr>
</table>
</form>
```

5.1.2 The instructor session page (user.php)

In this session the instructor will be able to execute the operations listed in session 5.

```
<html>
<body bgcolor="#008080">
<p align="center"><img border="0" src="NEU.BMP" width="219"
height="191"></p>
<pre> <font face="Monotype Corsiva" size="5"><b> NEAR
EAST UNIVERSITY </b></font></pre>
<hr>
</body>
</html>

<p>Welcome Sir<br>&nbsp;</p>

<p><a href=neu_logoff.php>Log off</a></p>
<p align="left"><br><a href=regstud.php>Register New
Student</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp; <a href=stud_marks.php>
Register Student
Courses</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp; <a href=change_mark.php>Insert/Edit Student
Mark</a></p><p><br><a href=cgba_get.php>Calculate Student CGBA</a></p>
```

5.1.3 Student registration session (regstud.php)

Here the instructor will be able to register a new student and insert the student name, surname, number, and class into the database.

```
<html>
<body bgcolor="#008080">
<p  align=center>        <font  face="Monotype  Corsiva"  size="5"><b>
Student Registration Session</b></font></p>
<hr>
<hr>
</body>
</html>

<?
//>
echo"<form action=regstuda.php method=post>
<table cellpadding=5>
<tr><td>Student#: <td><input name= studno> <td>Student name:<td><input name=
st_name> </tr>
<tr><td>Student  surname:  <td><input  name=  st_surname  ><td>Class:  <td><input
name= st_class></tr>
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.1.4 Courses registration session (stud_marks.php)

In this session the instructor can register the student courses. The student number, course ID, course credit, and if available the course grade.

```
<html>
<body bgcolor="#008080">
<pre>                <font    face="Monotype    Corsiva"    size="5"><b>
Student Course Registration </b></font></pre>
<hr>
<hr>
</body>
</html>
<?
//>
echo"<form action=stud_mark_add.php method=post>
<table cellpadding=3>
<tr><td>Course  Code  <td><input  name=  cs_id>  <td>Student#:<td><input  name=
studno> </tr>
<tr><td>Course  Grade  <td><input  name=  csgrad><td>Course  Credit  <td><input
name= cs_crid>
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.1.5 Course Grade registration (change_mark.php)

This is where the instructor can insert the student grades or edit them. Simply the course code, course grade and student number will be entered.

```
<html>
<body bgcolor="#008080">
<pre>                <font    face="Monotype    Corsiva"    size="5"><b>
Student Marks Insertion </b></font></pre>
<hr>
```

```
<hr>
</body>
</html>
<?
//>
echo"<form action=change_marka.php method=post>
<table cellpadding=3>
<tr><td>Course Code <td><input name= cs_id> <td>Student#:<td><input name=
studno> </tr>
<tr><td>Course grade <td><input name= csgrad >
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.1.6 The CGBA calculation session (cgba_get.php)

In this session the instructor will simply enter the student number to get a list of his courses, course grades and calculate his CGBA.

```
<html>
<body bgcolor="#008080">
<pre>                    <font face="Monotype Corsiva" size="4"><b>
Student CGBA Calculation</b></font></pre>
<hr>
<hr>
</body>
</html>
<?
//>
echo"<form action=cgba.php method=post>
<table cellpadding=3>
```

```
<p>Plesae inster the student number to calculate his CGBA</p>
<tr><td>Student #<td><input name= stno> <td>
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.1.7 Course deletion session (delete_mark.php)

As we said the instructor may need to delete a registered course if he/she had inserted it by mistake or if the student wants to withdraw the course. To do such an operation the instructor only need to enter the student number and course code to delete the course from the database.

```
<?
//>
echo"Deleting Student Marks";
echo"<form action=delet_marka.php method=post>
<table cellpadding=3>
<tr><td>Course Code <td><input name= cs_id> <td>Student#:<td><input name= studno> </tr>
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.1.8 Student deletion session (delete_stud.php)

In case the instructor needed to delete a student for some reason like wrong registration, he/she will be able of doing that in this session he/she only need to insert the student number and click submit.

```
<html>
<body bgcolor="#008080">
<p align="center">      <font face="Monotype Corsiva" size="6"><b> NEAR EAST
UNIVERSITY </b></font></p>
<p align="center">       <font face="Monotype Corsiva" size="5"><b> Student
Delition Session </b></font></p>
<hr>
<hr>
</body>
</html>
<?
//>
echo"Deleting Students ";
echo"<form action=delete_studa.php method=post>
<table cellpadding=3>
<tr>Student#:<td><input name= studno> </tr>
<tr><td align=center colspan=2><input type = submit></tr>
</table>
</form>
";
?>
```

5.2 The PHP scripts

As we know previously that we also used PHP scripts to make our page interactive, and we said that PHP is a pure programming language in root, so most our programming work is done using PHP. In this session we will present all the PHP scripts codes and explain what are the operations that will be carried out using each script.

5.2.1 Connecting to the database (connect_neu.php)

This is our first and most important PHP script. This script is used to connect the webpage to the database.

```
<?php
mysql_connect("localhost","root","");
mysql_query("use neu");
?>
```

5.2.2 The logging script (logger.php)

This is the script that allows the instructor to login into the session where he /she can execute the operations that we listed earlier. The script will connect to the database using (connect_neu.php) and will take the username and password posted using (login_neu.php), and then it will check if the username and password matches those in the database tables, if they didn't match the script will issu an error message indicating the error field.

```
<?php
//>
        include "neu_connect.php";
        $result=mysql_query("select pass from inst where user='$_POST[user]'");
        echo mysql_error();
        if(@mysql_num_rows($result)==0)
                die("Cannot find user");
        $p=mysql_result($result,0);
        if($p!=$_POST['pass'])
                die("Password is not correct");
        mysql_query("update inst set ip='$_SERVER[REMOTE_ADDR]'");
//      echo "<p>Your login is successful</p>";
      echo "<script>window.navigate('user.php');</script>"; ?>
```

5.2.3 Student registration script (regstuda.php)

This script is used to register student into the database, first the script will take the posted data from (regstud.php) the script will then select the table which is made to store the student registration information, then it will identify the data column for each type of data, if there is a mismatch in data types the row will be redundant.

```
<?
include "neu_connect.php";

import_request_variables("p","p_");

$sql = "INSERT into student (stnum, st_name, st_surname, st_class) VALUES
('$p_studno','$p_st_name','$p_st_surname','$p_st_class');";

mysql_query($sql);

echo"student insertion successfull";
?>
```

5.2.4 Courses registration script (stud_mark_add.php)

This script is used to register the student courses into the database, it will get the information posted through the (regstud.php) form, the open the database using (connect_neu.php) and then select the table designed for the student courses select the data columns that the data should be placed in then insert the data.

```
<?
include "neu_connect.php";

import_request_variables("p","p_");
```

```
$sql = "INSERT into courses (csid, stnum, cs_grd, credit, date) VALUES ('$p_cs_id','$p_studno','$p_csgrad','$p_cs_crid',now() );";

mysql_query($sql);

echo"Marks insertion successfull";
?>
```

5.2.5 Grades insert/edit script (change_marka.php)

This script will connect to the database through (connect_neu.php) the it will get the data posted through (change_mark.php) and then select the table that is designed for the grads and courses. It will then select the columns that the data should be inserted into, and then insert the data.

```
<?
include "neu_connect.php";

import_request_variables("p","p_");

$sql = "INSERT into courses (csid, stnum, cs_grd, credit, date) VALUES ('$p_cs_id','$p_studno','$p_csgrad','$p_cs_crid',now() );";

mysql_query($sql);

echo"Marks insertion successfull";
?>
```

5.2.6 The CGBA calculation script (cgba.php)

This is one amazing script that shows how powerful an interactive webpage can be. This script will connect to the database using (connect_neu.php) and takes the data posted through (cgba_get.php) then select the courses grads and course credits and will calculate the student CGBA and then it will select the course codes and course grads and display all of them.

```
<?php

include "neu_connect.php";

$result= mysql_query("select cs_grd, credit from courses where stnum=$p_[stno]");

$cnt=(@mysql_num_rows($result));

$sum=0; $sumc=1; $cgba=0;

for ($i=0; $i<$cnt; $i++)

{

$res = mysql_fetch_row($result);

switch($res[0])

{

case 'aa':
$sum+= 4 * $res[1];
break;

case 'cb':
$sum+= 3.5 * $res[1];
break;

case 'bb':
$sum+= 3 * $res[1];
break;
case 'cb':
$sum+= 2.5 * $res[1];
break;

case 'cc':
$sum+= 2.0 * $res[1];
break;

case 'dc':
$sum+= 1.5 *$res[1];
break;

case 'dd':
$sum+= 1 * $res[1];
break;

case 'fd':
$sum+= 0.5 * $res[1];
break;

case 'ff':
$sum+= 0.0 * $res[1];
break;

}

$sumc+= $res[1];

}

$cgba= $sum/$sumc;

?>
```

5.2.7 Course deletion script (delete_marka.php)

This is the script used for deleting the courses, the script will connect to the data base using (connect_neu.php), then it will take the data posted by the (delete_mark.php). the script will then select the table designed for the courses. It will select the data fields checking if it matches those posted by the corresponding form and the will delete the student grad.

```
<?
include "neu_connect.php";

import_request_variables("p","p_");

mysql_query("delete from courses where csid='$p_cs_id' && stnum='$p_studno'");

echo"Marks Deletion successfull";
?>
```

5.2.8 Student deletion script (delete_studa.php)

like all the other scripts this script will use (connect_neu.php) to connect to the database, and then it will take its data from the corresponding html form (delete_stud.php), now the script will search the student table for the data field that is entered so it will be able to delete it from the table.

```
<?
include "neu_connect.php";

import_request_variables("p","p_");

mysql_query("delete from student where stnum='$p_studno'");

echo"Student Deletion successfull"; ?>
```

5.2.9 The logoff script (neu_logoff.php)

This script is a good example for security. The script will connect to the database using (neu_connect.php) and then it will set the IP address which is stored in the database to NULL and send the user back to the login page.

```
<?php
        include "neu_connect.php";
        mysql_query("update inst set ip='' where
ip='$_SERVER[REMOTE_ADDR]'");
        echo "Log off successful";
        echo "<script>window.navigate('login_neu.php')</script>";
?>
```

5.3 MySQL creation and insertion codes

As we said that we are using MySQL for our webpage. We mainly used MySQL to store data on the server and of course because of its fast interactivity with PHP.

The codes listed below were used to create our database, create the database tables and insert some data into the tables.

5.3.1 Database and tables creation codes

Unlike some other databases in MySQL we have to create the database and inside the database we have to create its tables.

```
create database neu;              \\ create a data base
create table inst(                 \\ create the instructor table
user varchar(12) not null,
pass varchar(12),
ip text,
primary key (user);
```

```
create table student (                \\ create students table
Stnum varchar (12) not null,
St_name varchar (12),
St_surname varchar (12),
St_class char(1),
Primary key (stnum));

Create table courses (                \\ create courses table
Csid varchar (6) not null,
Stnum varchar (12) not null,
Cs_grd char (2),
Primary key (csid, stnum));
```

5.3.2 Insertion codes

As the instructor cannot create an account to him/her self, and the user ID and password will be given to him by the system administrator there must be a way that the administrator can insert the data into the tables.

A sample code of inserting the data into the tables is shown below:

```
Insert into inst values (
("username","password"," "); \\ inserting into table inst
```

Conclusion

I tried to meet the requirements of the project within the given deadline. The system is operating well and has been validated by tests. It is successful as it performs the necessary operations. The application was tested and used on the university's network to give an insight into the network resources and components as this may influence network related decisions of modification and maintenance.

References

1. www.chipmunkphpscripts.com
2. www.phpgeek.com
3. www.phpfreaks.com
4. www.php/mysql.com
5. Patrick N. Smith , Client/Server Computing, SHL Systemhouse, Inc., United States of America, January 1994.
6. Buzzard, James. The Client-Server Paradigm: Making Sense Out of the Claims, Data Based Advisor, August 1990.