NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

THE LIBRARY WEB PAGE DESIGN

Graduation Project
COM- 400

Student: Hamzeh Shatnawi (20021263)

Supervisor: Assist. Prof. Dr Firudin Muradov

Nicosia - 2006

ACKNOWLEDGMENTS

First of all I am happy to complete the task which I had given with blessing of God and also I am grateful to all the people in my life who have, supported me, advised me, taught me and who have always encouraged me to follow my dreams and ambitions. My dearest parents my brothers and sisters my friends and my tutors. They have taught me that no dream is unachievable. As in the words of Walt Disney "If you can dream it, you can do it."

I wish to thank my advisor, Assist.Prof Dr. Firudin Muradov, for intellectual support, encouragement, and enthusiasm, which made this project possible.

My sincerest thanks must go to all my friends, and all friends who shared their suggestions and evaluations throughout the completion of my project. The comments from these friends enabled me to present this project successfully.

Finally, I wish by this project to be useful for all students, especially Computer Engineering to support our improvement.

And above, I thank God for giving me stamina and courage to achieve my objectives.

To all of them, my love and respect

ABSTRACT

The Web is a hypertext system that runs over the internet as one of its services. Users can work on any computer and browse documents that exist any where in the world. The aim of the project is to design a website to search for books.

When the first page opened the user can connect to the search page. From this page the user can make search for all books, by entering the name of the book, the author, ISBN or felid of book. There are options that allow administrator updating, adding and deleting book or member in the database.

Similarly, many pages based on PHP that make the project on website attractive and easier.

TABLE OF CONTANTS

AKNOWLEDGMENT i
ABSTRACT ii
TABLE OF CONTENTS iii
INTRODUCTION vi

CHAPTER ONE: HYPER TEXT MARKUP LANGUAGE 1
1.1 Introduction 1
1.1.1 What is an HTML File? 1
1.2 HTML And The World Wide Web 1
1.2.1 What is the World Wide Web? 1
1.2.2 How does the WWW work? 1
1.2.3 How does the browser fetch the pages? 2
1.2.4 How does the browser display the pages? 2
1.3 HTML Elements 2
1.3.1 HTML Tags 2
1.3.2 Tag Attributes 2
1.4 Basic HTML Tags 3
1.4.1 Headings 3
1.4.2 Paragraphs 4
1.4.3 Line Breaks 4
1.4.4 Comments in HTML 4
1.5 HTML Links 4
1.5.1 The Href Attribute 5
1.5.2 The Target Attribute 5
1.6 HTML Tables 5
1.6.1 Table attribute 5
1.6.1.1 The Border Attribute 6
1.6.1.2 The align attribute 6
1.6.1.3 The width attribute 6
1.7 HTML Forms and Input 7
1.7.1 Forms 7
1.7.1.1 Type of method 7
1.7.1.2 Action 7
1.7.2 Input 7
1.7.2.1 Text Fields 7
1.7.2.2 Radio Buttons 8
1.7.2.3 Submit Button 9
1.8 Summary 9

CHAPTER TWO PERSONALIZED HOME PAGE (PHP) 10
2.1 Introduction 10
2.1.1 What Is PHP? 10

2.1.2 Why People Use PHP? 11

2.2 PHP Processing 11

2.3 Significance 12

2.4 File I/O And The File System 12

2.5 Characteristics Of PHP 12

2.6 Alternative Enclosure Bracketing 14

2.6.1 If Statement 14

2.6.2 If Else Statement 14

2.6.3 While Loop Statement 15

2.7 PHP Connection With MySQL: 16

2.8 PHP Forms And User Input 16

2.8.1 User Input 16

2.8.2 Request 17

2.9 Summary 17

CHAPTER THREE JAVA SCRIPT 19

3.1 Introduction 19

3.1.1 What is JavaScript? 19

3.1.2 Are Java and JavaScript the same? 19

3.1.3 What can a JavaScript Do? 19

3.2 JavaScript How TO? 20

3.2.1 How to Put a JavaScript Into an HTML Page 20

3.2.2 How to Handle Older Browsers 21

3.3 JavaScript Where To 21

3.3.1 Where to put the JavaScript 21

3.3.1.1 Scripts in the head section 22

3.3.1.2 Scripts in the body section 22

3.3.1.3 Scripts in both the body and the head section 22

3.4 JavaScript Variables 23

3.4.1 Declare a Variable 23

3.4.2 Assign a Value to a Variable 23

3.5 JavaScript Functions 24

3.5.1 How to Define a Function 24

3.5.2 How to Call a Function 25

3.5.3 The return Statement 25

3.6 JavaScript Conditional Statements 25

3.6.1 If and If...else Statement 25

3.6.1.1 Without ELSE 25

3.6.1.2 With ELSE 26

3.7 JavaScript Looping 27

3.7.1 for looping 28

3.8 Summary 28

CHAPTER FOUR STRUCTURED QUERY LANGUAGE (MySQL) 29

4.1 Introduction 29

2.1.1 What Is MySQL? 29
4.1.2 Why Use The MySQL Database Server? 30
4.2 The MySQL Interface 30
4.3 Basic MySQL Administration Tools 31
4.4 MySQL Table Types 32
4.5 MySQL Database Structure 33
4.5.1 Create A MySQL Database 33
4.5.2 Users, Passwords And Privileges 33
4.5.3 Account Settings: .my.cnf 34
4.6 MySQL Operations 35
4.6.1 Adding data to MySQL (INSERT) 35
4.6.2 Reading MySQL Databases: (SELECT) 35
4.6.3 Deleting Data From MySQL (DELETE) 36
4.6.4 Changing Data In MySQL (UPDATE) 36

CHAPTER FIVE THE LIBRARY WEB PAGE DESIGN 38
5.1 Introduction 38
5.2 Home Page 40
5.2.1 BookSearch.php 40
5.2.1.1 AdvanceSearch.php 42
5.2.1.2 SearchAvailability.php 44
5.3 Administrator Site 45
5.3.1 Admin.php 45
5.3.1.1 Cataloging (AdminSearch.php) 47
5.3.1.2 Member Info (StInfo.php) 50
5.3.1.3 Return (Return.php) 56
5.3.2 Searching Procedure 58
5.3.2.1 Fin.php 58
5.3.3 Collecting Of Registers 61
5.3.3.1 Add New Member (StMember.php) 61
5.3.3.2 Add New Book (BookRegistry.php) 63
5.3.4 Updating Procedure 64
5.3.4.1 Updating Member Information (EditMember.php) 64
5.3.4.2 Updating Book Information (BookEdit.php) 66
5.3.5 Deleting Procedure 67
5.3.5.1 Deleting Member 67
5.3.5.2 Deleting Book 68

CONCLUSION 69

REFERENCES 70

INTRODUCTION

The Hyper Text Markup Language (HTML), Personalized Home Page (PHP), and Structured Query Language (MySQL) are the main tools to design web pages.

JavaScript was designed to add interactivity to HTML page, and the JavaScript is used in the web page to improve the design, validate forms or data, and much more. JavaScript was developed by Netscape and is the most popular scripting language on the Internet.

Chapter One describes what is the World Wide Web, the difference between a Web browser and a Web server, also what HTML, how HTML "markup tags" are used to format a Web page and presents some basic guidelines for using HTML tags such as paragraph, heading, image, forms, and tables, ect....

Chapter Two describes Personalized Home Page (PHP) fundamentals such as PHP files and how PHP works, also the relationship between PHP and HTML, brief guide about Session variables and Application variables and the connection between PHP pages and MySQL.

Chapter Three introduces elements of JavaScript describes the fundamental concepts of JavaScript, How to insert a JavaScript code into an HTML Page.

Chapter Four describes the MySQL and how to create a table and how to make queries selection, updating, deleting, ect....

Chapter Five describes the web page design application techniques, and how we can run PHP on the computer (machine), for a library. It presents the pages used for this purpose, and how to run each page.

CHAPTER 1

HYPER TEXT MARKUP LANGUAGE (HTML)

1.1 Introduction

Hyper Text Markup Language (HTML) used to create web pages. Every web page has a basic structure; there are titles, headings, paragraphs, graphics and maybe even an itemized list of entries.

HTML uses a defined set of standards. All you need to do is use your favorite web editor and create a document with these labeled elements.

1.1.1 What is an HTML File?

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension
- An HTML file can be created using a simple text editor

1.2 HTML And The World Wide Web

1.2.1 What is the World Wide Web?

- The World Wide Web (WWW) is most often called the Web.
- The Web is a network of computers all over the world.
- All the computers in the Web can communicate with each other.
- All the computers use a communication standard called HTTP.

1.2.2 How does the WWW work?

- Web information is stored in documents called Web pages.
- Web pages are files stored on computers called Web servers.
- Computers reading the Web pages are called Web clients.
- Web clients view the pages with a program called a Web browser.

1.2.3 How does the browser fetch the pages?

- A browser fetches a Web page from a server by a request.
- A request is a standard HTTP request containing a page address.

1.2.4 How does the browser display the pages?

- All Web pages contain instructions for display
- The browser displays the page by reading these instructions.
- The most common display instructions are called HTML tags.
- HTML tags look like this <p>This is a Paragraph</p>.

1.3 HTML Elements

HTML documents are text files made up of HTML elements and HTML elements are defined using HTML tags.

1.3.1 HTML Tags

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like <b> and </b>
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, <b> means the same as <B>

1.3.2 Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page.

This tag defines the body element of your HTML page: <body>. With an added bgcolor attribute, you can tell the browser that the background color of your page should be red, like this: <body bgcolor="red">.

This tag defines an HTML table: <table>. With an added border attribute, you can tell the browser that the table should have no borders: <table border="0">

- Attributes always come in name/value pairs like this: name="value".
- Attributes are always added to the start tag of an HTML element.

1.4 Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Tag	**Description**
<html>	Defines an HTML document
<body>	Defines the document's body
<h1>To <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
<br>	Inserts a single line break
<hr>	Defines a horizontal rule
<! -- -- >	Defines a comment

1.4.1 Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading. <h6> defines the smallest heading.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
<h5>This is a heading</h5>
<h6>This is a heading</h6>
```

HTML automatically adds an extra blank line before and after a heading.

1.4.2 Paragraphs

Paragraphs are defined with the <p> tag.

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

HTML automatically adds an extra blank line before and after a paragraph.

1.4.3 Line Breaks

The <br> tag is used when you want to end a line, but don't want to start a new paragraph. The <br> tag forces a line break wherever you place it. The <br> tag is an empty tag. It has no closing tag.

1.4.4 Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment will be ignored by the browser. You can use comments to explain your code, which can help you when you edit the source code at a later date.

```
<!-- This is a comment -->
```

Note that you need an exclamation point after the opening bracket, but not before the closing bracket.

.1.5 HTML Links

HTML uses a hyperlink to link to another document on the Web.HTML Links uses the <a> (anchor) tag to create a link to another document.

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

1.5.1 The Href Attribute

The href attribute is used to address the document to link to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

1.5.2 The Target Attribute

With the target attribute, you can define where the linked document will be opened.(new page, main frame ,ect).

The line below will open the document in a new browser window:

```
<a href="url" arget="_blank"> Text to be displayed</a>
```

1.6 HTML Tables

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag).

The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table>
<tr>
<td>row 1, cell 1</td>
</tr>
<tr>
<td>row 2, cell 1</td>
</tr>
</table>
```

1.6.1 Table attribute

There are many attribute in the table, the Most Common is align, width ,border ,cellpadding ,cellspacing ect.

1.6.1.1 The Border Attribute

If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

To display a table with borders, you will have to use the border attribute:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

1.6.1.2 The align attribute

The align attribute is used as Center, Left, or Right to place the table on the page relative to browser.

```
<table align=”center”>
```

Here, will place a table on the center.

1.6.1.3 The width attribute

The table width is number of pixels or a percentage of the total page width.

```
<table width=”100”>
                    OR
<table width=”50%”>
```

1.7 HTML Forms and Input

1.7.1 Forms

Form elements are elements that allow the user to enter information (like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the <form> tag.

```
<form method=”hot to send” action=”URL of script”>
  <input>
  <input>
</form>
```

1.7.1.1 Type of method

There are two types of Methods used :

- Post: Not very popular but used for large quantity data transfers.
- Get : More popular but used to transfer a single line of text.

1.7.1.2 Action

Action is the attribute that defines the receiving URL.The receiving page must contain a script that can collect the incoming data from the sender.

1.7.2 Input

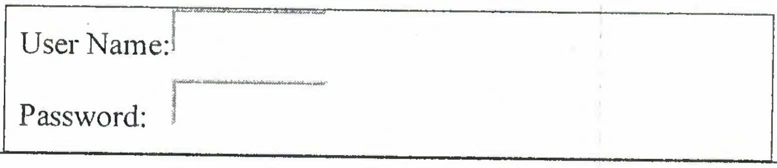
The most used form tag is the <input> tag. The type of input is specified with the type attribute. The most commonly used input types are explained below.

1.7.2.1 Text Fields

Text fields are used when you want the user to type letters, numbers, etc. in a form.

```
<form>
First name: <input type="text" name="User Name">
<br>
Last name: <input type="text" name="Password">
</form>
```

How it looks in a browser:



User Name:

Password:

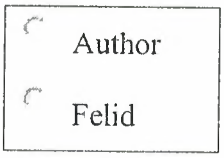
Note that the form itself is not visible. Also note that in most browsers, the width of the text field is 20 characters by default.

1.7.2.2 Radio Buttons

Radio Buttons are used when you want the user to select one of a limited number of choices.

```
<form>
<input type="radio name="chose"value="Author">Author
<br>
<input type="radio" name="chose" value="felid "> felid
</form>
```

How it looks in a browser:



○ Author

○ Felid

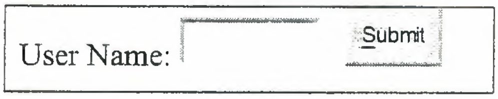
Note that only one option can be chosen.

1.7.2.3 Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to another file. The form's action attribute defines the name of the file to send the content to. The file defined in the action attribute usually does something with the received input.

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user">
          <input type="submit" value="Submit">
</form>
```

How it looks in a browser



If you type some characters in the text field above, and click the "Submit" button, you

will send your input to a page called "html_form_action.asp". That page will show you the received input.

1.8 Summary

The web pages (HTML) are the mediums of communicating across internet.They provide information in plain text as well as Graphics, Sound, Tables, Links, ect...

The web pages are published on host system called **web server,** which have IP Address and Domain names and are connected to the **www**

CHAPTER 2

PERSONALIZED HOME PAGE (PHP)

2.1 Introduction

Nowadays not only e-commerce but also various kinds of industries and educational institutions seem to seek to build dynamic websites that can handle database and can be customized for each visitor. The technology that helps the implementation of such dynamic features has been making progress.

ASP, PHP, JSP, J2EE, and ColdFusion are considered to be the most typical technology for this purpose. Among them, I chose PHP as my research topic. The goal of my research is to provide useful information for people who need to choose Web application server, Web server, database, OS, and platform for their application system. Before designing websites, people (system architects, system engineers, and web designers whether or not they are professional) often face a situation in which they need to decide which software to use for the implementation of their requirements and specifications.

2.1.1 What Is PHP?

PHP is one of the projects of the Apache Software Foundation. PHP (recursive acronym for "Hypertext Preprocessor") is free and offered under an open source license. (This means that you can use it as you wish. The "license" really is only to cover commercial use of the code. For your own use, you don't have to worry about “license” at all, so you can download it and use it without any problem.) It is a widely used general-purpose **scripting language** and **interpreter** that is especially used to create dynamic Web pages. It is commonly used to extract data out of a database and present it on the Web pages. It can be embedded into HTML. PHP interpreter (or often called ‘parser’) generates HTML code within the Web server. Therefore, it does not consume CPU time or memory whereas, in CGI, another process is created and executed and it consumes a great deal of CPU time and memory. Before the page is sent to a user that has requested it, the Web server calls the PHP interpreter to parse the script and perform the operations called for in the PHP script. An HTML page that includes a PHP script is typically given a file name suffix of ".php" ".php3," or ".phtml". In an HTML document, PHP script (similar syntax to that of Perl or C) is enclosed within special PHP tags. Therefore, the author can jump in and out of PHP mode in an HTML file instead of having to rely on heavy amounts of code to output HTML. You can think of it as a normal HTML file which happens to have a set of special tags available to you that do a lot of interesting things. PHP can perform any task that any CGI program can do, but its strength lies in its compatibility with many types of databases.

2.1.2 Why People Use PHP?

1.The old fashioned approach to adding interactivity was to use CGI mechanism.

2.Another reason to use PHP is because it's free.

3.PHP is easy. The language is a mixture of C and Perl, taking the best features from both.

4.One of the strongest and most significant feature of PHP is its native database support for a wide range of databases (MySQL, mSQL, Oracle, dBase, which allows access to the databases directly through SQL statements

5.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. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others.

2.2 PHP Processing

PHP is mainly focused on server-side scripting, so you can do anything that any other CGI program can do, such as collect form data, generate dynamic Web page content, or send and receive cookies. PHP is also good at database access, disk access, networking, and text manipulation.

1. Server-side scripting.
2. Command line scripting
3. Writing client-side GUI applications.

4. With PHP, you are not limited to output HTML. PHP's abilities include outputting images, PDF files, and even flash movies

5. 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.

2.3 Significance

PHP is still very popular among web application developers. One reason is that it is public domain software and free. However, there are some other features that support this popularity.

2.4 File I/O And The File System

(Input/output), as you can imagine, data input and output flows are put to considerable use in the developing of Web applications. Not limited to simple reading and writing of files, PHP provides support for viewing and modifying server information, in addition to executing third-party programs. These features are the subject of this chapter.

2.5 Characteristics Of PHP

As you may have realized, the PHP language revolves around the central theme of practicality. PHP is about providing the programmer with the necessary tools to get the job done in a quick and efficient fashion. Five important characteristics make PHP's practical nature possible:

- Familiarity
- Simplicity
- Efficiency
- Security
- Flexibility

One final characteristic makes PHP particularly interesting: it's free!

1.Familiarity:

Programmers from many backgrounds will find themselves already accustomed to the PHP language. Many of the language's constructs are borrowed from C and Perl, and in many cases PHP code is almost indistinguishable from that found in the typical C or Pascal program. This minimizes the learning curve considerably.

2.Simplicity

A PHP script can consist of 10,000 lines or one line: whatever you need to get the job done. There is no need to include libraries, special compilation directives, or anything of the sort. The PHP engine simply begins executing the code after the first escape sequence (<?) and continues until it passes the closing escape sequence (?>). If the code is syntactically correct, it will be executed exactly as it is displayed.

3.Efficiency

Efficiency is an extremely important consideration for working in a multi-user environment such as the WWW. PHP 4.0 introduced resource allocation mechanisms and more pronounced support for object-oriented programming, in addition to session management features. Reference counting has also been introduced in the latest version, eliminating unnecessary memory allocation.

4.Security

PHP provides developers and administrators with a flexible and efficient set of security safeguards. These safeguards can be divided into two frames of reference: system level and application level.

5.Flexibility

Because PHP is an embedded language, it is extremely flexible towards meeting the needs of the developer. Although PHP is generally touted as being used in conjunction solely with HTML, it can also be integrated alongside languages like JavaScript, WML, XML, and many others. Additionally, as with most other mainstream languages, wisely planned PHP applications can be easily expanded as needed. Browser dependency is not an issue because PHP scripts are compiled entirely on the server side before being sent to the user. In fact, PHP scripts can be sent to just about any kind of device containing a browser, including cell phones, personal digital assistant (PDA) devices, pagers, laptops, not to mention the traditional PC. People who want to develop shell-based applications can also execute PHP from the command line. Since PHP contains no server-specific code, users are not limited to a specific and perhaps unfamiliar Web server.

2.6 Alternative Enclosure Bracketing

Control structures are enclosed in a set of brackets to clearly signify the various statements making up the structure. Curly brackets ({ }) were introduced earlier.

As a convenience for programmers, an alternative format for enclosing control structures exists, as demonstrated here:

2.6.1 If Statement

```
if (expression) :
Statement block
```

Example:

```
if($numRows>0)
{$data=mysql_fetch_array($result);
$_SESSION["userid"]=$data["id"];
if(isset($_SESSION["userid"]))
{header ("Location:
adminsearch.php");}}
```

2.6.2 If Else Statement:

```
if (expression) :
```

```
Statement block
Else
Statement block
endif;
```

Example:

```
if($numRows>0)
{$data=mysql_fetch_array($result);
$_SESSION["userid"]=$data["id"];
if(isset($_SESSION["userid"]))
{header ("Location:
adminsearch.php");}}

else
{print "your Name and Password
are Uncorrect";}
```

2.6.3 While Loop Statement:

```
while (expression):
Statement block
endwhile;
```

Example:

```
while($getinfon2=mysql_fetch_array($sel2))
{$isbnbck2=$getinfon2["stid"];}
```

2.7 PHP Connection With MySQL:

In my project we can use MySQL and make a connection with page by using PHP to active the information from database in MySQL and show it in the screen or insert data to the database or do compare between the entry data and the data in the database the connection code in my project is:

```
<?php
session_start();
$sqlink=mysql_connect('localhost','root');
mysql_select_db("mysql");
if (! $sqlink){
die('could not connect to server..!!'.mysql_error());
}?>
```

2.8 PHP Forms And User Input

The Request commands may be used to retrieve information from forms, like user input.

2.8.1 User Input

The Request object may be used to retrieve user information from forms.

For example:

```
<form action="<?php print $_SERVER['PHP_SELF']?>" method="post"
name="regestry" onSubmit="return DoSubmit(this)">
<table align="center" cellpadding="0" cellspacing="0">
<td>Search ISBN availability :</td><td><input type="text"
name="isbn"></td>
```

```
<td><input type="submit" name="submit" value="Search"></td>
</table></form>
```

2.8.2 Request

The Request command is used to collect values in a form with method="post". Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

When the user send data as you see before it will received by request like this example:

```
if ($_SERVER['REQUEST_METHOD']=="POST")
{$username=$_POST["User"];
$passname=$_POST["pass"];
$result=mysql_query("SELECT * FROM admin WHERE
username='".$username."' and password='".$passname."' ") or
die(mysql_error());
$numRows = mysql_num_rows($result);
if($numRows>0)
{$data=mysql_fetch_array($result);
$_SESSION["userid"]=$data["id"];
if(isset($_SESSION["userid"]))
{header ("Location: adminsearch.php");}}
else
{print "your Name and Password are Uncorrect";}}?>
```

2.9 Summary

PHP is a free, open source scripting language. It is easy to develop and provides powerful control. It provides strong support for a wide range of databases, too. PHP is also available for a wide variety of platforms. It can be used on all major operating systems, including Linux, many UNIX variants (including HP-UX, Solaris and Open

BSD), Microsoft Windows, Mac OS X, RISC OS, etc. It 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, and many others. With PHP, you will have 'freedom of choice' regarding a platform, an operating system, a Web server, and a database. It is intriguing to think that we can easily build up a database and a dynamic Web site by using these kinds of free, public domain software. The Internet liberated knowledge and information from the monopoly by a small group of people to the mass. Now knowledge and information are open to everyone. People can obtain knowledge and information much more easily than before and can organize them in a way they like by utilizing new technology that costs you less expensive or even free.

CHAPTER 3

JAVA SCRIPT

3.1 Introduction

JavaScript is used in millions of Web pages to improve the design, validate forms, and much more. JavaScript was developed by Netscape and is the most popular scripting language on the Internet.

3.1.1 What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language - a scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript is usually embedded directly in HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license
- JavaScript is supported by all major browsers, like Netscape and Internet Explorer

3.1.2 Are Java and JavaScript the same?

NO!

Java and JavaScript are two completely different languages!

Java (developed by Sun Microsystems) is a powerful and very complex programming language - in the same category as C and C++.

3.1.3 what can a JavaScript Do?

- JavaScript gives HTML designers a programming tool - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages

- JavaScript can put dynamic text into an HTML page - A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page

- JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element

- JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element

- JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing

3.2 JavaScript How TO?

The HTML <script> tag is used to insert a JavaScript into an HTML page.

3.2.1 How to Put a JavaScript Into an HTML Page

```
<html>
<body>
<script type="text/JavaScript">
document.write("Hello World!")
</script>
</body>
</html>
```

The code above will produce this output on an HTML page: Hello World!

To insert a script in an HTML page, we use the <script> tag. Use the type attribute to define the scripting language **<script type="text/JavaScript">**

Then the JavaScript starts: The JavaScript command for writing some output to a page is document.write. **document.write("Hello World!")**

Then the <script> tag has to be closed. **</script>**

3.2.2 How to Handle Older Browsers

Browsers that do not support scripts will display the script as page content. To prevent them from doing this, we may use the HTML comment tag:

```
<script type="text/javascript">
<!--
   some statements
//-->
</script>
```

The two forward slashes at the end of comment line (//) are a JavaScript comment symbol. This prevents the JavaScript compiler from compiling the line.

Note: You cannot put // in front of the first comment line (like //<!--), because older browsers will display it.

3.3 JavaScript Where To

- Scripts in the body section will be executed WHILE the page loads.
- Scripts in the head section will be executed when CALLED.

3.3.1 Where to put the JavaScript

Scripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

3.3.1.1 Scripts in the head section:

Scripts to be executed when they are called, or when an event is triggered, go in the head section. When you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

```
<html>
<head>
<script type="text/javascript">
    Some statements
</script>
</head>
```

3.3.1.2 Scripts in the body section:

Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
    Some statements
</script>
</body>
</html>
```

3.3.1.3 Scripts in both the body and the head section:

We can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/javascript">
    some statements
</script>
</head>
<body>
<script type="text/javascript">
    some statements
</script>
</body>
```

3.4 JavaScript Variables

A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.

Rules for Variable names:

- Variable names are case sensitive
- They must begin with a letter or the underscore character

3.4.1 Declare a Variable

- We can create a variable with the var statement: var strname = some value
- We can also create a variable without the var statement: strname = some value

3.4.2 Assign a Value to a Variable

We can assign a value to a variable like this: var strname = "Hege"
The variable name is on the left side of the expression and the value you want to assign to the variable is on the right. Now the variable "strname" has the value "Hege".

3.5 JavaScript Functions

A function contains some code that will be executed by an event or a call to that function. A function is a set of statements. You can reuse functions within the same script, or in other documents. You define functions at the beginning of a file (in the head section), and call them later in the document.

3.5.1 How to Define a Function

To create a function you define its name, any values ("arguments"), and some statements:

```
function myfunction(argument1,argument2,etc)
{
some statements
}
```

A function with no arguments must include the parentheses:

```
function myfunction()
{
some statements
}
```

Arguments are variables used in the function. The variable values are values passed on by the function call.

By placing functions in the head section of the document, you make sure that all the code in the function has been loaded before the function is called.

Some functions return a value to the calling expression

```
function result(a,b)
{
c=a+b
return c
}
```

3.5.2 How to Call a Function

A function is not executed before it is called,We can call a function containing arguments:

myfunction(argument1,argument2,etc) Or without arguments: **myfunction()**

3.5.3 The return Statement

Functions that will return a result must use the "return" statement. This statement specifies the value, which will be returned to where the function was called from.

3.6 JavaScript Conditional Statements

Conditional statements in JavaScript are used to perform different actions based on different conditions.

In JavaScript we have three conditional statements:

- if statement - use this statement if you want to execute a set of code when a condition is true
- if...else statement - use this statement if you want to select one of two sets of lines to execute

3.6.1 If and If...else Statement

3.6.1.1 Without ELSE

You should use the if statement if you want to execute some code if a condition is true.

```
if (condition)
          {
          code to be executed if condition is true
          }
```

Example:

There is no...Else... in this example. Just tell the code to execute some code if the condition is true

```
// JavaScript Document
function DoSubmit(reges)
{ if (reges.isbn.value=="") {alert("pleas Enter any
ISBN..");reges.isbn.focus();return false;}
```

3.6.1.2 With ELSE

If you want to execute some code if a condition is true and another code if a condition is false, use the if.... else statement.

```
if (condition)
{
code to be executed if condition is true
}
else
{
code to be executed if condition is false
}
```

Example:

```
// JavaScript Document
function DoSubmit(reges)
{if (reges.stname.value==""){alert("pleas Enter Member
Name");reges.stname.focus();return false;}
 else if (reges.surname.value==""){alert("pleas Enter
Surname");reges.surname.focus();return false;}
 else if (reges.stid.value==""){alert("pleas Enter Member
ID");reges.stid.focus();return false;}
else if (reges.department.value==""){alert("pleas Enter
Department ");reges.department.focus();return false;}
else if (reges.telno.value==""){alert("pleas Enter Tel
NO");reges.telno.focus();return false;}
else if (reges.email.value==""){alert("pleas Enter
Email");reges.email.focus();return false;}
else if (reges.fupload.value==""){alert("pleas Upload
Image:");reges.fupload.focus();return false;}
else {return true;}}
```

3.7 JavaScript Looping

Looping statements in JavaScript are used to execute the same block of code a specified number of times.

In JavaScript we have the following looping statements:

- While - loops through a block of code while a condition is true
- Do...while - loops through a block of code once, and then repeats the loop while a condition is true
- For - run statements a specified number of times

3.7.1 for Looping

The for statement will execute a block of code a specified number of times

```
for (initialization; condition; increment)
{
   code to be executed
}
```

Example:

```
function preload()

{for(i = 0; i<imgOn.length; i++)

{         imgObOn[i]=new Image();

          imgObOn[i].src=imgOn[i];

          imgObOff[i]=new Image();

          imgObOff[i].src=imgOff[i];

          }}
```

3.8 Summary

Java Script is used in HTML document to make some condition and application and the Java Script is lines of executable computer code.

Java Script in HTML will be executed into two sections:

- Body
- Head

CHAPTER 4

STRUCTURED QUERY LANGUAGE (MySQL)

4.1 Introduction

The MySQL (TM) software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server.

MySQL Server is intended for mission critical, heavy-load production systems as well as for embedding into mass-deployed software.

MySQL is a trademark of MySQL AB. The MySQL software has Dual Licensing, which means you can use the MySQL software free of charge under the GNU General Public License. You can also purchase commercial MySQL licenses from MySQL AB if you do not wish to be bound by the terms of the GPL.

The MySQL web site provides the latest information about the MySQL software.

4.1.1 What Is MySQL?

MySQL, the most popular Open Source SQL database, is developed and provided by MySQL AB. MySQL AB is a commercial company that builds its business providing services around the MySQL database.

The MySQL web site provides the latest information about MySQL software and MySQL AB. MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and exibility. The tables are linked by de_ned relations making it possible to combine data from several tables on request. The SQL part of \MySQL" stands for \Structured Query Language"the most common standardised language used to access databases. MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify. Anybody can download the MySQL software from the Internet and use it without paying anything. Anybody so inclined can study the source code and change it to _t their needs. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application you can buy a commercially licensed version from us.

4.1.2 Why Use The MySQL Database Server?

The MySQL Database Server is very fast, reliable, and easy to use. If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can a performance comparison of MySQL Server to some other database managers on our benchmark page.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Though under constant development, MySQL Server today o_ers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet. The technical features of MySQL Server For advanced technical information.

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports deferent backends, several deferent client programs and libraries, administrative tools, and a wide range of programming interfaces (APIs). We also provide MySQL Server as a multi-threaded library which you can link into your application to get a smaller, faster, easier-to-manage product. There is a large amount of contributed MySQL software available. It is very likely that you will _nd that your favorite application or language already supports the MySQL Database Server.

4.2 The MySQL Interface

One simple (in terms of its appearance and capabilities) way of accessing MySQL is through the standard interface. To enter the MySQL interface

```
unixprompt>mysql-u wbyeats -p
Enter password:
mysql>
```

The “-u” tag precedes the username, and the “-p” tag invokes the password prompt. To log in as any other user replace the username accordingly. MySQL commands are then typed on the mysql command line. The end of each command is defined by a semicolon ;. Once you have entered the mysql interface you can select a database to look at (with the use command) and use any MySQL queries to read, edit, or add data.

```
 mysql> use tissueinfo;
mysql> show tables;
+-----------------------------+
| Tables_in_tissueinfo |
+-----------------------------+
| gene_map |
| gene_info |
+-----------------------------+
mysql> quit
```

4.3 Basic MySQL Administration Tools

Usually only the root user has permission to create new databases and new users for the MySQL server. The MySQL user is independent of any other username. This means that an individual may use more than one MySQL username. To enter the MySQL interface as root:

```
unixprompt> mysql -u root -p
Enter password:
mysql>
```

4.4 MySQL Table Types

As of MySQL, you can choose between three basic table formats (ISAM, HEAP and MyISAM. Newer MySQL may support additional table type (InnoDB, or BDB), depending on how you compile it.

When you create a new table, you can tell MySQL which table type it should use for the table. MySQL will always create a `.frm' _le to hold the table and column dentitions.

Depending on the table type, the index and data will be stored in other _les.

Note that to use InnoDB tables you have to use at least the innodb _data _file _path startup option.

The default table type in MySQL is MyISAM. If you are trying to use a table type that is not compiled-in or activated, MySQL will instead create a table of type MyISAM. This is a very useful feature when you want to copy tables between deferent SQL servers that supports deferent table types (like copying tables to a slave that is optimized for speed by not having transactional tables). This automatic table changing can however also be very confusing for new MySQL users. We plan to _x this by introducing warnings in MySQL and giving a warning when a table type is automatically changed. You can convert tables between deferent types with the ALTER TABLE statement.

Note that MySQL supports two deferent kinds of tables: transaction-safe tables (InnoDB and BDB) and not transaction-safe tables (HEAP, ISAM, MERGE, and MyISAM). Advantages of transaction-safe tables (TST): Safer. Even if MySQL crashes or you get hardware problems, you can get your data back, either by automatic recovery or from a backup + the transaction log. You can combine many statements and accept these all in one go with the COMMIT command. You can execute ROLLBACK to ignore your changes (if you are not running in auto commit mode). If an update fails, all your changes will be restored. (With NTST tables all changes that have taken place are permanent) Advantages of not transaction-safe tables (NTST):

1. Much faster as there is no transaction overhead.
2. Will use less disk space as there is no overhead of transactions.
3. Will use less memory to do updates.
4. You can combine TST and NTST tables in the same statements to get the best of both worlds.

4.5 MySQL Database Structure

MySQL databases consist of a(ny) number of tables. Tables hold the data. Tables are made up of columns and rows. A user that has been given CREATE and DROP permissions on a database can create and remove tables of that database. The CREATE TABLE command simultaneously creates the table and defines its structure (although the structure of the table can later be changed using the ALTER TABLE command).

4.5.1 Create A MySQL Database

It is easy to create a new MySQL table within the MySQL prompt. mysql> CREATE DATABASE tissueinfo;

4.5.2 Users, Passwords And Privileges

We now have an empty database, but we don't yet have any users to access this database. To simultaneously create a user, assign a password, and grant access to this newly created database enter:

```
mysql> GRANT USAGE ON tissueinfo.* to wbyeats@localhost
IDENTIFIED BY 'ode2maud';
```

This creates the user "wbyeats" if it doesn't already exist, and sets the password to "ode2maud". Note that if the user "wbyeats" already exists the password will be set to "ode2maud" (even if the password was previously set to something different). This statement also grants wbyeats access to all of the tables within the tissueinfo database (specified by "tissueinfo.*").

One of the attractive features of MySQL is the strict security it gives your data. The tradeoff is some extra work for the database administrator, because access privileges must be individually set. Granting usage only allows the user to log in to the database, but not to actually look at the data or enter any data. To grant these privileges the root user must also specify:

```
mysql> GRANT SELECT, INSERT ON tissueinfo.* to wbyeats@localhost
IDENTIFIED BY 'ode2maud';
```

Which gives "wbyeats" permission to look at data (SELECT) and to add new data (INSERT). If you trust "wbyeats" you can grant all possible permissions (including permission to delete any data in the database) with the simple statement:

```
mysql> GRANT ALL ON tissueinfo.* to wbyeats@localhost
IDENTIFIED BY 'ode2maud';
```

4.5.3 Account Settings: .my.cnf

If you are frequently using mysql through the unix commands or the mysql interface then the requirement to specify username and password every time quickly becomes tedious. Within UNIX/Linux you can write these parameters into a file called .my.cnf in your home directory. This file should contain your username and password information in exactly the following format. [client]

```
user=wbyeats
host=localhost
password=ode2maud
```

MySQL will automatically read this information when you are using the MySQL interface or system commands (at the UNIX prompt), but not when connecting to the MySQL database from within a Perl script (see later). This means you do not need to specify -u wbyeats -p when executing commands. For the rest of this document the commands will be written as if this file is in place. If it is not you will need to add the -u wbyeats -p parameters to the command line.

4.6 MySQL Operations

4.6.1 Adding data to MySQL (INSERT)

Once you have created a table, you can start filling it with data. One simple way of adding a lot of data is by using the mysqlimport system command.

This will read in a text file where data for each table row are separated by newlines, and data for each column are separated by tabs and in the same order as the columns were defined. The file should be named according to the convention tablename.txt.table, replacing “tablename” appropriately. A sample file for the gene_map table might look like this:

```
agrc259 1 233.3
adp0 10 9.3
adp4 4 4.3
```

This is then imported into the tissueinfo database of MySQL with unixprompt> mysqlimport --local tissueinfo gene_map.txt.table Data rows may also be added one by one inside the MySQL interface:

```
mysql> INSERT INTO gene_map VALUES(’agrc259’,’1’,’233.3’,’NULL’);
mysql> INSERT INTO gene_map (gene, chromosome) VALUES(’adp0’, ’10’);
```

Note that if you specify a value for every column in order it is not necessary to declare the column names in the statement (e.g., first statement above).If you only want to add data to some columns, or to add them in a different order, then you must declare the column names after the table name in the statement (e.g., second statement above). In this case columns with no value specified will be “NULL” or will get the default value (“0” for numeric columns, “” for text columns, and the next integer for auto increment columns) if “NOT NULL” was specified during table definition.

4.6.2 Reading MySQL Databases: (SELECT)

The SELECT command is used to view all or some of the elements of a table or multiple tables. The basic format is SELECT column FROM table; “*” indicates all columns. “%” is the wildcard. Here are some examples:

```
mysql> SELECT * FROM gene_map;
mysql> SELECT * FROM gene_map
WHERE chromosome=’10’;
mysql> SELECT gene FROM gene_map
WHERE chromosome=’10’ AND cM_position
>= ’100’;
mysql>  SELECT  gene,  cM_position  FROM
gene_map
WHERE chromosome=’10’
ORDER BY cM_position;
mysql> SELECT * FROM gene_map
 WHERE gene LIKE ’asg%’;
```

It is also possible to select data from multiple tables where they have at least one column in common. The following example selects the chromosome column from the gene map table and the description column from the gene info table and links the data by the gene.

```
mysql>  SELECT  gene_map.chromosome,  gene_info.description
FROM gene_map, gene_info
WHERE gene_map.gene = gene_info.gene;
```

4.6.3 Deleting Data From MySQL (DELETE)

The Delete command is used to delete all or some of the elements from table or multiple tables.

Examples:

```
$mydelete=mysql_query("DELETE FROM member
WHERE  stidnumber=".$editnum."");
```

4.6.4 Changing Data In MySQL (UPDATE)

The Update command is used to Change all or some of the elements in the table or multiple tables.

Examples:

```
{$fineupdate1=mysql_query("UPDATE member SET stname='".$upvalue."'
WHERE stidnumber=".$id." ")or die("Update ERROR 1 :".mysql_error());}
```

CHAPTER 5

THE LIBRARY WEB PAGE DESIGN

5.1 Introduction

Firstly the project contains around 14 pages some pages connecting with other pages and the other one connecting with database.

Secondly the relationships between the files or pages as the bottom figure 5.1: Main page contain 2 pages its **BookSearch.php, admin.php.**

Every pages from this connecting with other as I said before the **BookSearch.php** connecting with **MySQL, advancesearch.php** and **searchavailability.php**.

The **advanesearch.php** and **searchavailability.php** with mySQL, the **admin.php** connects with **MySQL** to compare the user name and password with entry data to allow him go to the AdminSearch.php**, stinfo.php** and **return.php** pages.

The **adminsearch.php** connecting with **mySQL, bookregistry.php** and **bookedit.php**.

Bookregistry.php and **bookedit.php** connect with **MySQL**,

The page of member information it is **stinfo.php**, this page do connection with **mySQL**, **editmember.php**, **fin.php** and **stmember.php**.

Editmember.php,fin.php and **stmember.php** connect with **mySQL.**

The return books page it is **return,php** connect with **mySQL**

In the figure the student he can do search for books there is four ways for search by the name, author, field and ISBN.

For the members they must have the user name and password to enter the work pages.

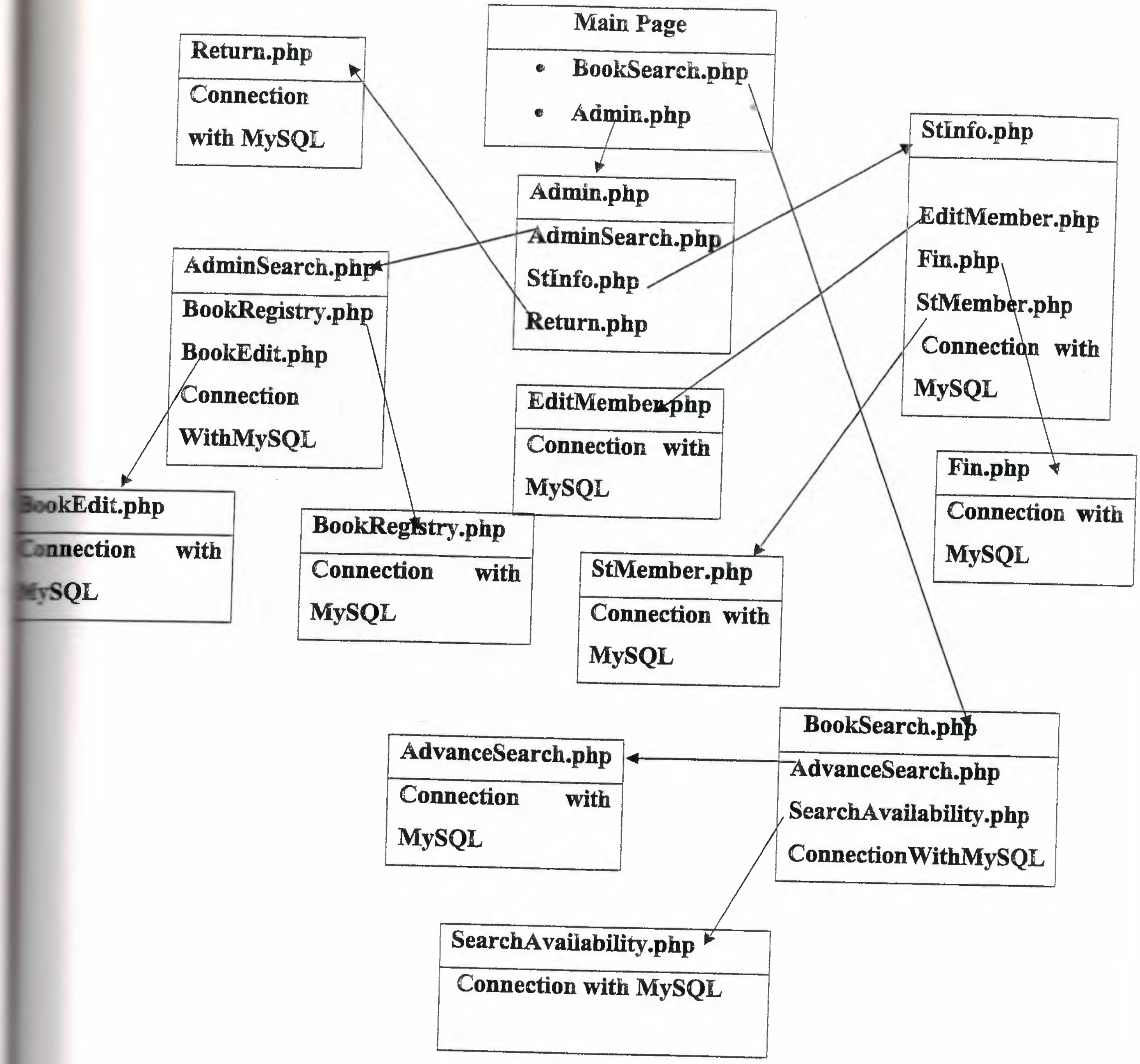


Figure 5.1 The Relationships Between Files

5.2 Home Page

5.2.1 BookSearch.php

The project in HTML file represent in this figure:

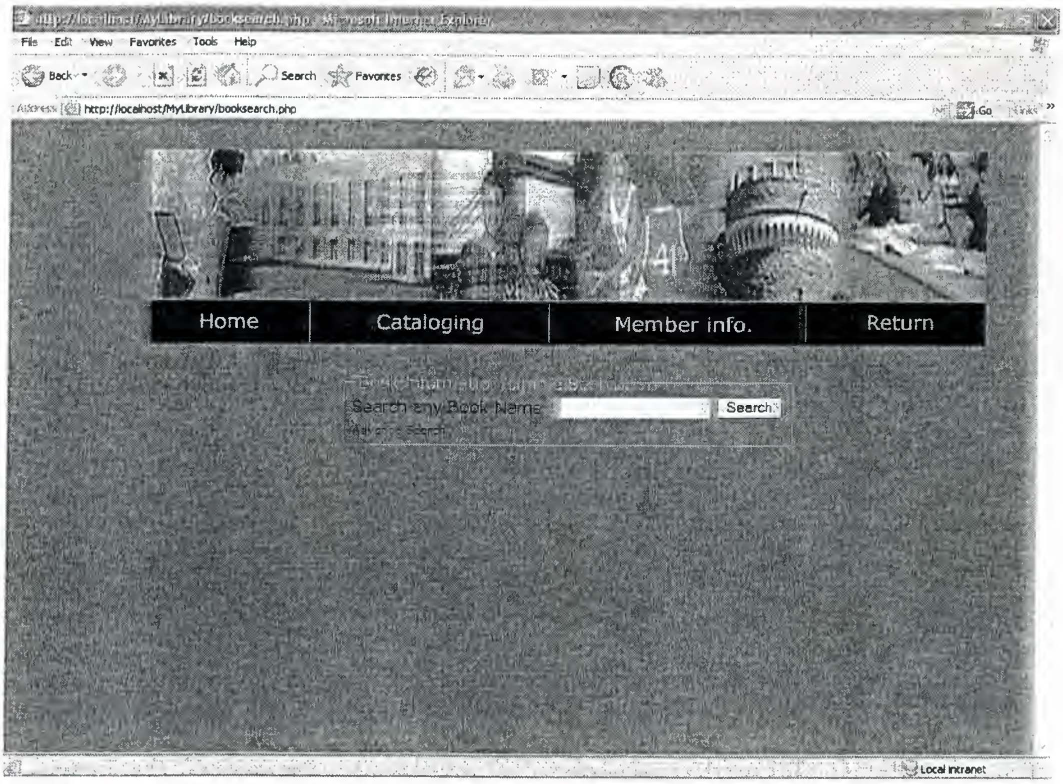


Figure 5.2 Home page in the project. BookSearch.php

When the user or student pen the web site he will see this page to do search for books by the name if he know the name or first letters from the name. Also this page connects with **AdvanceSearch.php** to search by other ways and **SearchAvailability.php** to check for books.

Now this page contains some technical tags as **Forms**, **Inputs** and then connect with **MySQL** to compare the entry name with **MySQL** data by using this code:

```
<table align="center"  border="0">
<tr><form action="booksearch.php" method="post" name="regestry"
onSubmit="return DoSubmit(this)">
<td> Search any Book Name :</td>
<td><input type="text" name="bname" size="20"></td>
<td><input type="submit" name="submit" value="Search"></td></form>
```

```
          </tr>
 <tr><td><a href="advancesearch.php">Advance Search</a></td></tr>
 </table></fieldset>
          <?php
          if($_SERVER["REQUEST_METHOD"]=="POST")
          {$bname=$_POST["bname"];
           if (isset($_POST["bname"]))
          {$sel3=mysql_query("SELECT * FROM mybook WHERE
                    bookname   LIKE   '".$bname."%'   or   bookname   LIKE
'%".$bname."'or   bookname   LIKE   '*".$bname."'   or   bookname   LIKE
'".$bname."*'   or   bookname   LIKE   '?".$bname."*'   or   bookname   LIKE
'".$bname."?' ")or die("selection error".mysql_error());
                    $row=mysql_num_rows($sel3);
                    echo "There are ".$row." Books result";
                    ?>
                    <table><tr><td   height="20"><img   src="images/blank.gif"
height="20"></td></tr></table>
<table  cellpadding="15" border="1">
<thead><th>Title</th>
<th>Author</th>
<th>Publish date</th>
<th>LC</th><th>CLASIFICATION</th>
<th>AVailablity</th> </thead><?php
          while($getinfon=mysql_fetch_array($sel3))
                    {$valid=$getinfon["duedate"];
                    ?>
<tr><td align="center"><?php echo  $getinfon["bookname"]; ?></td>
                    <td align="center"><?php echo $getinfon["author"]; ?></td>
                    <td  align="center"><?php  echo  $getinfon["publishigdate"];
?></td>
<td align="center"><?php echo $getinfon["shilveLC"]; ?></td>
<td align="center"><?php echo $getinfon["LCclasification"]; ?></td>
<td align="center"><?php if ($valid==0){ echo "Avilable";} if ($valid>0){
```

```
echo" Not Available"; }  ?></td></tr>
                <?php echo "<hr>";}
                ?></table>
```

The Connecting code in Home page

5.2.1.1 AdvanceSearch.php

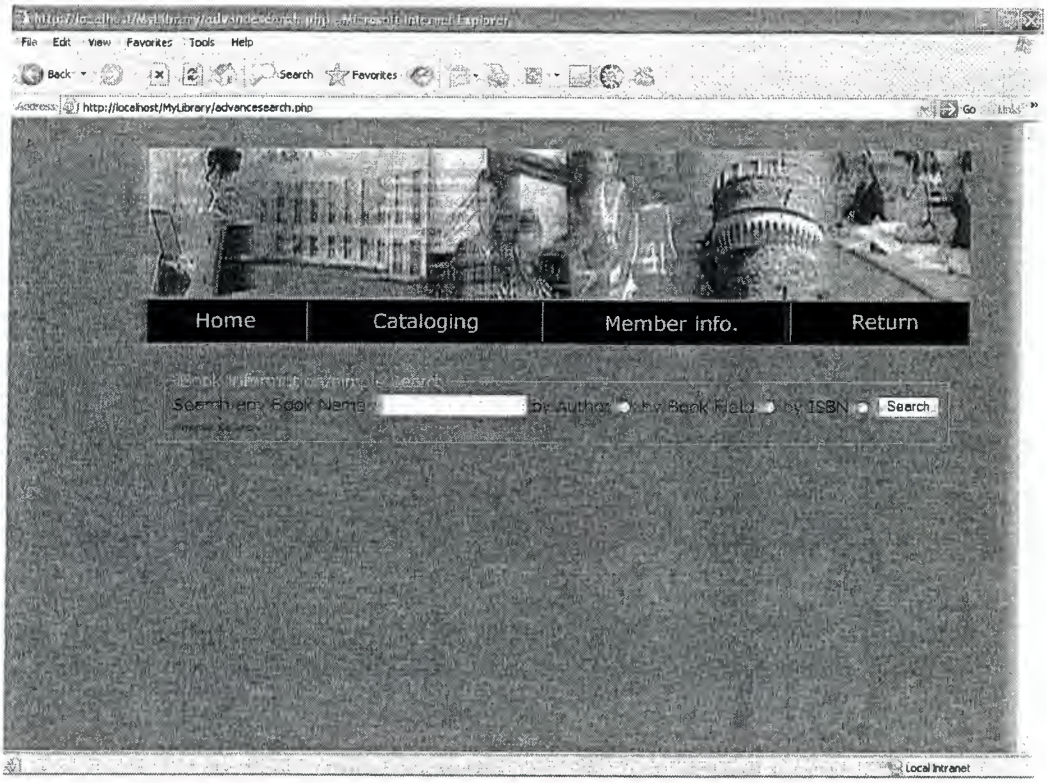


Figure 5.3. AdvanceSearch.php

This page has some technical tags as Radio Button, Forms and Input connecting with MySQL to do search for books.

Now the user or student he can do search for books by author or fields or serial number (ISBN) if he didn't know the name of book. This technical design is done by using this code:

```
<table align="center"  border="0"><tr>
<form    action="advancesearch.php"    method="post"    name="regestry"
onSubmit="return DoSubmit(this)">
```

```
<td> Search any Book Name :</td>
<td><input type="text" name="bname" size="20"></td>
<td>by Author</td><td><input  type="radio" name="a" value="1"><td>
<td>byBookField</td><td><input  type="radio" name="a" value="2"></td>
<td>by ISBN</td><td><input  type="radio" name="a" value="3"></td>
<td><input type="submit" name="submit" value="Search"></td>
</form></tr>
<tr><td><a href="booksearch.php">Simple Search</a></td></tr>
</table></fieldset>
                    <?php
                    if($_SERVER["REQUEST_METHOD"]=="POST")
                    {$bname=$_POST["bname"];
                    $value=$_POST["a"];
                     if (isset($_POST["bname"]))
                    {if($value==1){
                    $sel3=mysql_query("SELECT * FROM mybook WHERE
                    author LIKE '".$bname."%' or author LIKE '%".$bname."'or
author LIKE '*".$bname."' or author LIKE '".$bname."*' or author LIKE
'?".$bname."*' or author LIKE '".$bname."?' ")
                    or die("selection error 1".mysql_error());}
                    if($value==2){
                    $sel3=mysql_query("SELECT * FROM mybook WHERE
                    bookfield LIKE '".$bname."%' or bookfield LIKE
'%".$bname."'or bookfield LIKE '*".$bname."' or bookfield LIKE
'".$bname."*' or bookfield LIKE '?".$bname."*' or bookfield LIKE
'".$bname."?' ")
                    or die("selection error 2".mysql_error());}
                     if($value==3){
                    $sel3=mysql_query("SELECT * FROM mybook WHERE
                    isbn LIKE '".$bname."%' or isbn LIKE '%".$bname."'or isbn
LIKE '*".$bname."' or isbn LIKE '".$bname."*' or isbn LIKE '?".$bname."*'
or isbn LIKE '".$bname."?' ")
                    or die("selection error 3".mysql_error());}
```

```
                $row=mysql_num_rows($sel3);
                echo "There are ".$row." Books result";
                ?>
<table><tr><td          height="20"><img          src="images/blank.gif"
height="20"></td></tr></table>
<table  cellpadding="15" border="1">
                <thead>       <th>Title</th>
                              <th>Author</th>
                              <th>Publish date</th>
                              <th>LC</th>
                              <th>ISBN</th>
                              <th>CLASIFICATION</th>
                              <th>AVailablity</th> </thead>
        <?php
                while($getinfon=mysql_fetch_array($sel3))
                {$valid=$getinfon["duedate"];
                ?>
<tr><td align="center"><?php echo  $getinfon["bookname"]; ?></td>
     <td align="center"><?php echo $getinfon["author"]; ?></td>
      <td align="center"><?php echo $getinfon["publishigdate"]; ?></td>
      <td align="center"><?php echo $getinfon["shilveLC"]; ?></td>
       <td align="center"><?php echo $getinfon["isbn"]; ?></td>
       <td align="center"><?php echo $getinfon["LCclasification"]; ?></td>
       <td   align="center"><?php   if   ($valid==0){   echo   "Avilable";}   if
($valid>0){ echo" Not Available"; }  ?></td>
               </tr><?php echo "<hr>";}?></table>
```

The Connecting Code in AdvanceSearch.php

5.2.1.2 SearchAvailability.php

This page just using to do chick for the books if it is exist or not and give a message for user and it is done by using this code:

```
<table><tr><td><td>Adminstration</td>
<form action="searchavailabilty.php" method="post" name="regestry"
onSubmit="return DoSubmit(this)">
<td>Search any ISBN availability <td><input type="text"
name="myisbn"></td></td>
<td><td><input type="submit" name="submit" value="Search"></td></td>
</form> </tr></td></table>
<?php
$bname=$_POST["myisbn"];
if(($_SERVER["REQUEST_METHOD"]=="POST")and
isset($_POST["myisbn"])))
{$sel3=mysql_query("SELECT * FROM mybook WHERE isbn like
".$bname." ")or die("selection error".mysql_error());
$row=mysql_num_rows($sel3);
echo "There are ".$row." result";
if ($row > 0)
{while($getinfon=mysql_fetch_array($sel3))
{echo "<hr>";
?>
<table cellspacing="0" cellpadding="0" border="0">
<tr><td width="346">
<br><tr><td>
        <strong>ISBN is olredy available</strong></tr></td>
        <strong>Title</strong> :<?php echo $getinfon["bookname"];
?></tr></td><br>
        <strong>ISBN </strong> :<?php echo $getinfon["isbn"];
?></tr></td> <br> </table>
```

The Connecting Code in SearchAvailability.php

5.3 Administrator Site

5.3.1 Admin.php

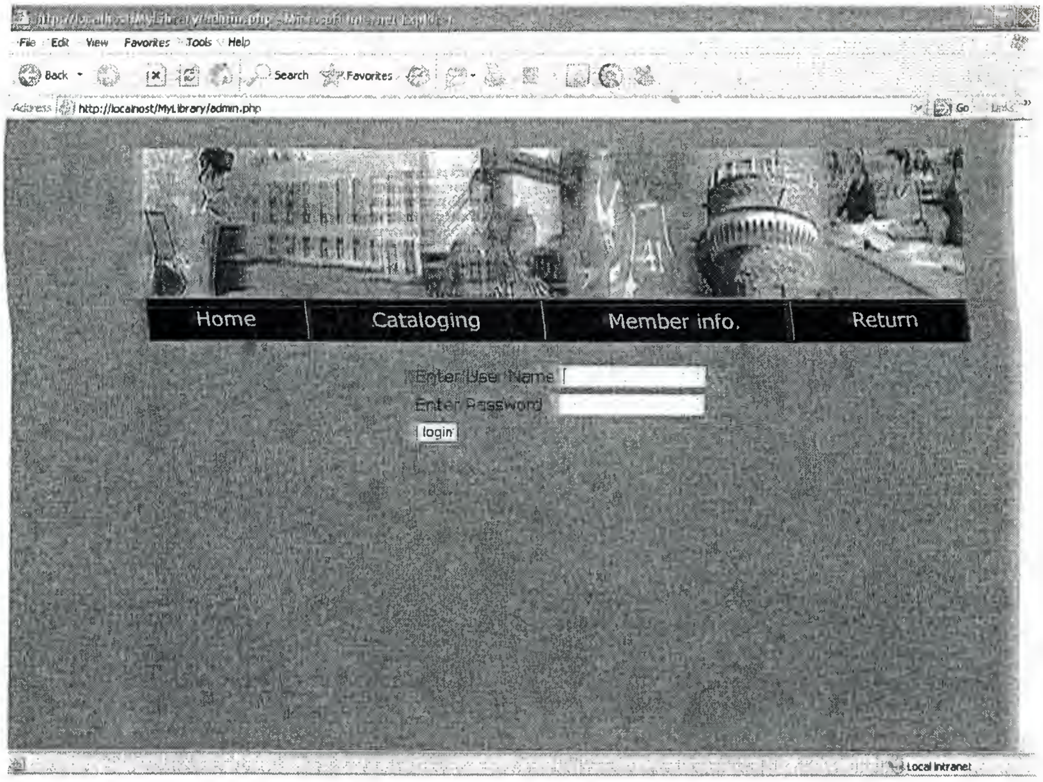


Figure 5.4 Administrator Page

This page connecting with MySQL to compare the user name and password if it correct or not, if it is correct he can go to next pages. The next is Cataloging (**AdminSearch.php)**,Member Info(**StInfo.php)** and Return(**Return.php)**.

Now the administrator must have user name and password to do some operation as checking for books, given the information about the users and return the books. This page is done by this code:

```
<?php
session_start();
$sqlink=mysql_connect('localhost','root');
mysql_select_db("mysql");
if (! $sqlink){
die('could not connect to server..!!'.mysql_error());
}
if ($_SERVER['REQUEST_METHOD']=="POST")
{       $username=$_POST["User"];
        $passname=$_POST["pass"];
        $result=mysql_query("SELECT * FROM admin WHERE
username='".$username."' and password='".$passname."' ") or
die(mysql_error());
        $numRows = mysql_num_rows($result);
        if($numRows>0)
                        {       $data=mysql_fetch_array($result);
                                $_SESSION["userid"]=$data["id"];
         if(isset($_SESSION["userid"]))
                        {       header ("Location: adminsearch.php");}}
         else
                        {print "your Name and Password are Uncorrect";}}
?>
<html><head><script type="text/javascript" language="javascript"
src="admin.js"></script></head>
<body bgcolor="#0066CC">
<?php include "mystyle.Css";?>
<table align="center" cellpadding="0" cellspacing="0"><tr><td>
<table align="center" border="0" cellpadding="0"
cellspacing="0"><tr><td><?php include "banner.php";?></td></tr></table>
<form action="<?php print $_SERVER['PHP_SELF']?>" method="post"
name="loginingin" onSubmit="return DoSubmit(this)">
<table align="center">
      <tr><td>Enter User Name</td><td><input type=text name="User"
size =20></td></tr>
<tr><td>Enter Password</td><td><input type="password" name="pass"
size=20> </td></tr>
<tr><td><input type="submit" value="login" ></td></tr></table></form>
</td></tr></table></body></html>
```

The Connecting Code in Admin.php

5.3.1.1 Cataloging (AdminSearch.php)

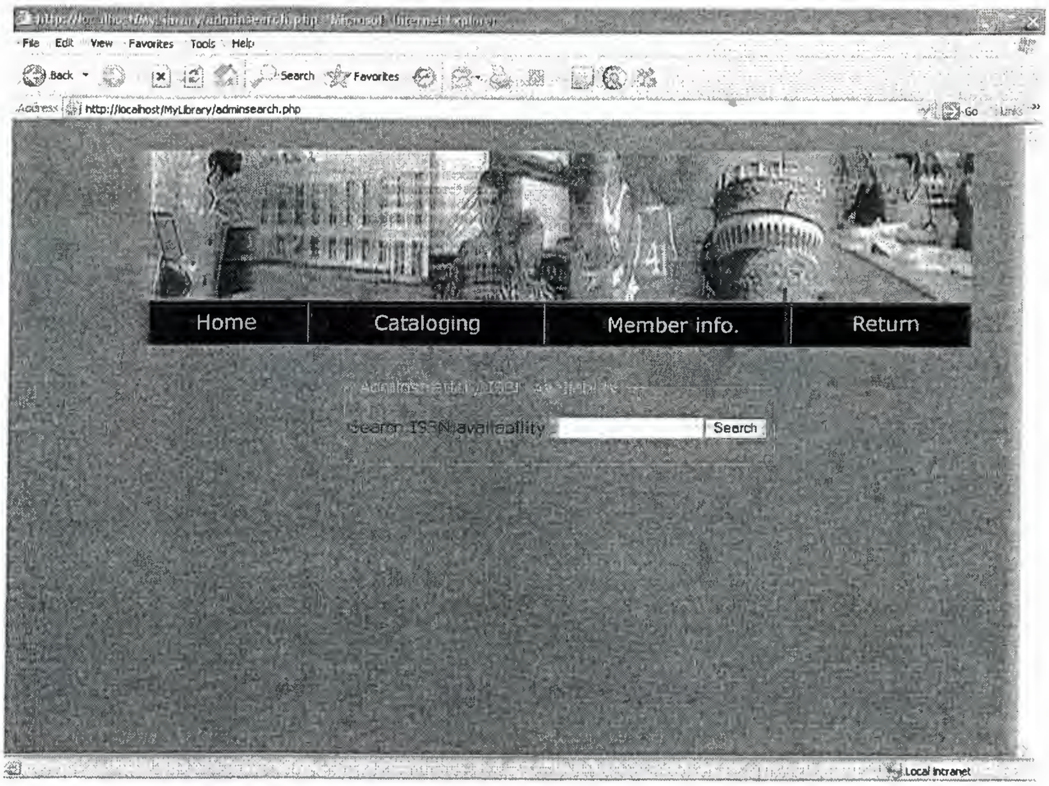


figure 5.5 Cataloging Page(AdminSearch.php)

When the administrator enters the user name and password he can open this page, in this page the administrator he can do search or check for the books by the serial number (ISBN) if it is available or not.

Now this page connect with tow pages first page is **BookRegistry.php** to add the new books, the second page is **BookEdit.php** to update the information of books.

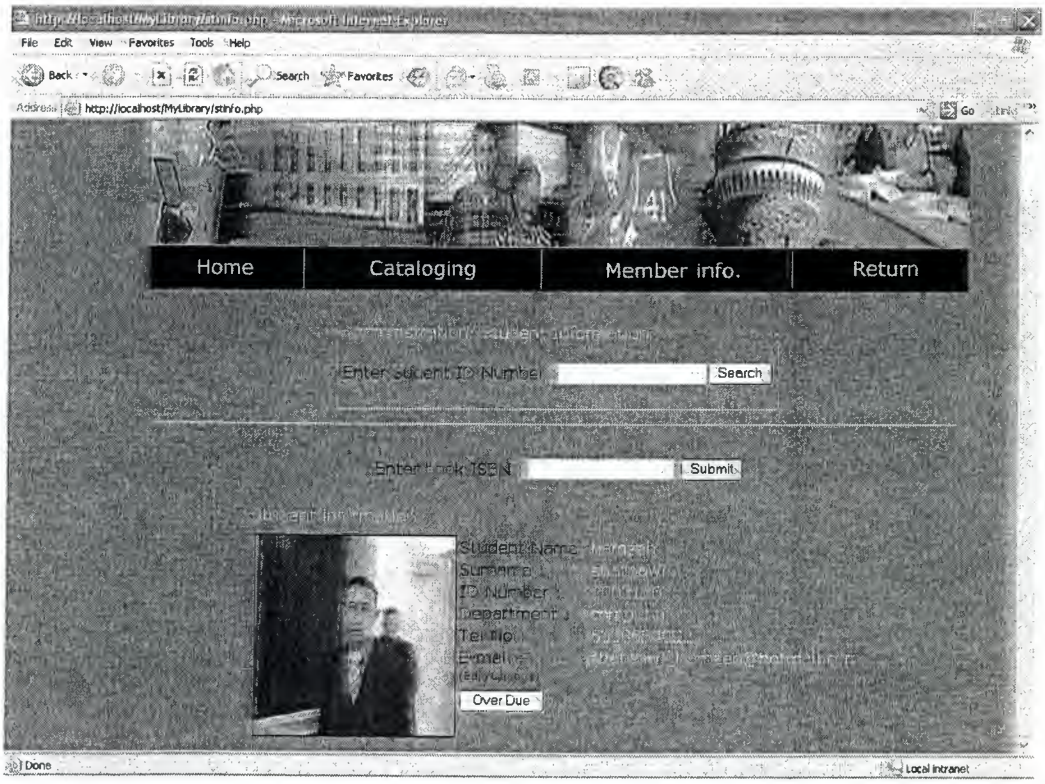
This page is done by using this code:

```
<?php
session_start();
if(! isset($_SESSION["userid"]))
{header("location: admin.php");}
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR :".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR :".mysql_error());
?><html><head>
<script type="text/javascript" language="javascript"  src="isbnavilaible.js"></script>
</head><body bgcolor="#0066CC">
```

```
<?php include "mystyle.Css";?>
<table  align="center" cellpadding="0" cellspacing="0" hspace="0"><tr><td>
<table align="center" border="0" cellpadding="0" cellspacing="0"><tr><td><?php include
"banner.php";?></td></tr></table>
<table        align="center"><tr><td        height="25"><img        src="images/blank.gif"
height="25"></td></tr>
                <tr><td><fieldset><legend  style="color:#FFFFFF">
                Adminstration/ ISBN Availabilty:     </legend>
<form action="<?php print $_SERVER['PHP_SELF']?>" method="post" name="regestry"
onSubmit="return DoSubmit(this)">
<table align="center" cellpadding="0" cellspacing="0">
<td>Search ISBN availability :</td><td><input type="text" name="isbn"></td>
<td><input type="submit" name="submit" value="Search"></td></table></form>
</fieldset>
<?php
if($_SERVER["REQUEST_METHOD"]=="POST")
{$bname=$_POST["isbn"];
if (isset($_POST["isbn"]))
{$sel3=mysql_query("SELECT * FROM mybook WHERE isbn='".$bname."' ")
 or die("selection error".mysql_error());
$row=mysql_num_rows($sel3);
if ($row==0)
{?>
<table><tr>    <td><?php echo "There are ".$row." result"; ?></td>
               <tr><td><?php echo "ISBN  is not available"; ?></td></tr>
               <td><a href="BookRegestry.php">Add New Records</a></td></tr></table>
<?php }
if ($row>0)
{while($getinfon=mysql_fetch_array($sel3)) {
echo "<hr>";
?><table cellspacing="0" cellpadding="0" border="0"><tr><td>          <tr>
                    <td><strong>ISBN is olredy available</strong></td></tr>
<tr><td><strong>Title:<?php echo  $getinfon["bookname"]; ?></strong></td>
<td><strong>ISBN:<?php echo  $getinfon["isbn"]; ?></strong></td>
<tr><td><a href="bookedit.php">(Edit Book Information)</a></td></tr></tr></td></tr>
</table></td></tr></table><?php}}}}
?>
</td></tr></table></body></html>
```

Cataloging Code

5.3.1.2 Member Info (StInfo.php)



Enter student ID Number

Enter Book ISBN

Student Name
Surname
ID Number
Department
Tel No
E-mail

Figure 5.6 Member Information (StInfo.php)

When the administrator enters the user name and password he can open this page, in this page the administrator he can do search for the student by the ID Number to appear the information about him and submit him book.

This page also connecting with **Fin.php** to check if there is dept on the the student or not, **StMember** to add new member or student and **EditMember.php** to update the information about the student or member, this page is done by this code:

```
<?php
session_start();
if(! isset($_SESSION["userid"]))
        {header("location: admin.php");     }
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR
:".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR
:".mysql_error());
?><head><script type="text/javascript" language="javascript"
src="member.js"></script></head><body bgcolor="#0066CC">
<?php include "mystyle.Css";?><table align="center" cellpadding="0"
cellspacing="0" width="800"><tr><td><table align="center" border="0"
cellpadding="0" cellspacing="0"><tr><td><?php include
"banner.php";?></td></tr></table>
<table align="center"><tr><td height="25"><img src="images/blank.gif"
height="25"></td></tr><tr> <td><fieldset><legend style=" color:#FFFFFF
">Adminstration/ Student Information</legend>
<form action="stinfo.php" method="post" name="findstudentinfo"
onSubmit="return DoSubmit(this)">
<table align="center" cellpadding="0" cellspacing="0" hspace="0"
vspace="0"><tr><td>Enter Sudent ID Number :</td><td><input type="text"
name="idno"></td><td><input type="submit" name="submit"
value="Search"><td></tr></table></form></fieldset></td></tr></table>
</form><?php
                if(($_SERVER["REQUEST_METHOD"]=="POST")and (!
isset($_POST["idno"])))
                {$isbnbk=$_POST["bisbn"];
                $mydate=$_POST["date"];
                $month=$_POST["mon"];
                $year=$_POST["year"];
                $idm=$_POST["pos"];
```

```
            if (isset($_POST["bisbn"]))
            {$sel2=mysql_query("SELECT * FROM mybook WHERE
isbn=".$isbnbk." ") or die("selection 2 Error".mysql_error());
                        $row2=mysql_num_rows($sel2);
                                if ($row2>0)
                                {?><table align="center"><tr><td><?php echo
"There are ".$row2." result";?></td></tr></table><?php
while($getinfon2=mysql_fetch_array($sel2))
{ $isbnbck2=$getinfon2["stid"];}}
$num=0;
if($isbnbck2==$num)
{$update=mysql_query("UPDATE mybook SET stid=".$idm.",duedate
=".$mydate.",mon=".$month.",myyear=".$year." WHERE isbn=".$isbnbk."
")or die("update ERROR :".mysql_error());}
else{?><table align="center"><tr><td><?php echo "This book has borwoed by
another Member"; ?></td></tr></table> <?php }    }}}
if    (($_SERVER["REQUEST_METHOD"]=="POST")    and    (!
isset($_POST["bisbn"])))
{$idno=$_POST["idno"];
if (isset($_POST["idno"]))
        {$sel=mysql_query("SELECT * FROM member WHERE stidnumber
='".$idno."' ")
 or die("selection1 error".mysql_error());
$row=mysql_num_rows($sel);
if ($row==0)
{?><table align="center"><tr><td><?php echo "There are ".$row." result";
?></td><td><?php    echo    "SOORY    ID    NOT
VALID";$newmember=1;?></td></tr>
<?php  if ($newmember==1){ ?><td><a href="stmember.php">add new
member</a></td></table><?php }}
if ($row>0){$worrning=0;
while($getinfon=mysql_fetch_array($sel))
{?><table  align="center"  width="800"><tr><td><?php  echo  "<hr>";
```

```
?></td></tr></table><?php
$id=$getinfon["stid"];
$date_array = getdate();
                                        $day=$date_array['mday'];
                                        $mon=$date_array['mon'];
                                        $finemone=$mon;
                                        $year=$date_array['year'];
                                        if ($day<24)
                                        {$day1=$day+7;}
                                        if ($day==24)
                                        {$day1=1;$mon=$mon+1;}
                                        if ($day==25)
                                        {$day1=2;$mon=$mon+1;}
                                        if ($day==26)
                                        {$day1=3;$mon=$mon+1;}
                                        if ($day==27)
                                        {$day1=4;$mon=$mon+1;}
                                        if ($day==28)
                                        {$day1=5;$mon=$mon+1;}
                                        if ($day==29)
                                        {$day1=6;$mon=$mon+1;}
                                        if ($day==30)
                                        {$day1=7;$mon=$mon+1;}
                                        if ($day==31)
                                        {$day1=8;$mon=$mon+1;}
        //$last=$date_array['mon']."/".$day1."/".$date_array['year'];
                                        //$holder = ( string ) $last;
                                        //$holder =( string )$last;
                                        //echo "".$holder."";
                                        //$holder=strval($last)?>
<form action="stinfo.php" name="BookISBN" method="post">
<table align="center"><tr><td>
<td>Enter book ISBN :<td><input type="text" name="bisbn"></td>
```

```
<inputtype="hidden"name="pos"value="<?phpecho$id=$getinfon["stid"];?>">
<input type="hidden" name="date" value="<?php echo "".$day1."";?>">
<input type="hidden" name="mon" value="<?php echo "".$mon."";?>">
<input type="hidden" name="year" value="<?php echo "".$year."";?>">
<td><input type="submit" name="su" value="Submit"></td></td>
        <tr><td></td></tr></td></tr></table></form>
<table align="center" cellspacing="0" cellpadding="0" border="0"><tr><td>
<tr><td style="color:#FFFFFF">Student information </td>
<tr><td height="10"><img src="images/blank.gif" height="10"></td></tr>
        </tr><tr><td rowspan="9"><img align="left" border=1 height="200"
width="200" src="upload/<?php echo $getinfon["stpicture"]; ?>"></td>
<tr><td > Student Name :<td style="color:#FFFFFF"><?php echo
$getinfon["stname"];?></td></td> </tr><tr>
<td> Surname :<td style="color:#FFFFFF"><?php echo
$getinfon["stsurname"]; ?></td></td> </tr><tr>
<td> ID Number :<td style="color:#FFFFFF"><?php echo
$getinfon["stidnumber"]; ?></td></td></tr> <tr><td> Department :<td
style="color:#FFFFFF"><?php echo $getinfon["stdepartment"]; ?></td></td>
</tr><tr><td> Tel No :<td style="color:#FFFFFF"><?php echo
$getinfon["telno"]; ?></td></td></tr><tr><td> E-mail :<td
style="color:#FFFFFF"><?php echo $getinfon["stemail"]; ?></td></td></tr>
<tr><td><a href="editmember.php">(Edit/Change)</a></td></tr>
<td><?php if($worning==0){?><form action="fin.php" name="id"
method="post"><input type="hidden" name="kamal" value="<?php echo
$getinfon["stid"];?>"><input type="submit" name="mid" value="Over
Due"></form><?php /* } */ ?></td> </td></tr></table><?php
$sel1=mysql_query("SELECT * FROM mybook WHERE stid=".$id." ")
 or die("selection error".mysql_error());
$row1=mysql_num_rows($sel1);
if ($row1>0)
{?>
<table align="center" width="800"><tr><td>
<?php echo "There are ".$row1." Borowed Books"; ?> </td></tr></table>
```

```
<?php
$sign=0;
$worrning=0;
while($getinfon1=mysql_fetch_array($sel1))
{$fine=0;
?><table align="center" width="800"><tr>
<?php echo "<hr>";?><td></td></tr><?php
$myday=$getinfon1["duedate"];
$mymon=$getinfon1["mon"];
$myyear=$getinfon1["myyear"];
$dbfine=$getinfon1["fine"];
$dbreturn=$getinfon1["return"];
if ($finemone==$mymon){
if ($day>$myday){$worrning="0"; $fine=($day-$myday);}
if ($day<$myday){$fine=($day-$myday);}
}//else{$fine=0;}
if ($finemone<$mymon){$worrning="0";$fine=0;}
if ($finemone<$mymon){$worrning="0"; $calcmon=($mymon-$finemone);
if($calcmon==1){$fine=-(((30-$day)+$myday));$sign=1;}}
if ($finemone>$mymon){$worrning=1; $calcmon=($finemone-$mymon);
if($calcmon==1){$fine=((30-$myday)+$day);/*$sign=1;*/}}
if($fine>32){$fine=60;}
if($myday==0){?>
<table border="0" align="center" cellpadding="0" cellspacing="0"
width="700" bgcolor="#FFFFFF"><tr>
            <td width="50">ISBN</td><td style=" color:#990000"><?php
echo $getinfon1["isbn"];?></td>
            <td>Title<td><td style="color:#990000"><?php echo
$getinfon1["bookname"];?>
            <td style=" color:#990000">Book Returnd</td>
            <td ></td><td style=" color:#990000"><?php echo "Fine
".$dbfine." :Days"; ?></td></tr></table><?php}if($myday>0){?>
<table align="center" border="0" cellpadding="0" cellspacing="0"
width="700">
<tr><td width="50" style="color:#FFFFFF">ISBN</td><td><img
src="images/blank.gif" width="2"></td><td><?php echo
$getinfon1["isbn"];?></td><td style="color:#FFFFFF">Title</td><td><?php
echo $getinfon1["bookname"];?></td>
<td style="color:#FFFFFF">Due Date</td><td><?php echo $myday;?>/<?php
echo $mymon;?>/<?php echo $myyear;?></td>
<td><td style="color:#FFFFFF"><?php echo "Due..".$fine." :Days";
?></td></td> </td></td></td></td> </td></tr></table>
<table align="center" width="800"><tr><?php
echo "<hr>";
$fineupdate=mysql_query("UPDATE mybook SET fine=".$fine." WHERE
isbn=".$getinfon1["isbn"]." ")or die("update ERROR :".mysql_error());
?><td></tr></table><?php }}}}}}}}?></td></tr></table></body>
```

The Member Information (StInfo.php)

5.3.1.3 Return (Return.php)

When the administrator enters the user name and password he can open this page, in this page the administrator he can return the books when he insert the serial number(ISBN) and it is done by this code:

```
<?php
session_start();
if(! isset($_SESSION["userid"]))
{header("location: admin.php");}
?><head><script type="text/javascript" language="javascript"
src="return.js"></script></head>
<body bgcolor="#0066CC">
<?php include "mystyle.Css";?>
<table align="center" cellpadding="0" cellspacing="0"><tr><td>
<table align="center" border="0" cellpadding="0"
cellspacing="0"><tr><td><?php include "banner.php";?></td></tr></table>
<table><tr><td height="25"><img src="images/blank.gif"
```

```
height="25"></td></tr></table><table align="center"><tr><td><fieldset>
<legend style=" color:#FFFFFF ">Adminstration/ Return Book</legend>
<table align="center"><form action="return.php" method="post"
name="returnbook" onSubmit="return DoSubmit(this)">
<td>Enter ISBN Number :</td><td><input type="text" name="isbn"></td>
<td><input type="submit" name="submit" value="Submit"></td></form>
</td></tr></table><?php
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR
:".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR
:".mysql_error());
if ($_SERVER["REQUEST_METHOD"]=="POST")
                {$isbnbk=$_POST["isbn"];
                if (isset($_POST["isbn"])){
                        $select=mysql_query("SELECT * FROM mybook
WHERE isbn='".$isbnbk."' ") or die(" My Selection  Error".mysql_error());
                        $row=mysql_num_rows($select);
                        ?><tr><table align="center"><tr><td><?php echo
"There are ".$row." result";?></td><?php
                        $null=0;
                        $dbreturn=1;
                $n1=0; $n2=0; $n3=0; $rn=0; $Error=0;
                if ($row>0)
                {while($fetching=mysql_fetch_array($select))
                {$dbfinevalue=$fetching["fine"];
        if($dbfinevalue>0){$returnupdate=mysql_query("UPDATE mybook
SET return=".$dbreturn.",duedate=".$n1.",mon=".$n2.",myyear=".$n3."
WHERE isbn=".$isbnbk." ")or die("update ERROR :".mysql_error());}
if(($dbfinevalue<0) or ($dbfinevalue==0)){
                                $MYupdate=mysql_query("UPDATE mybook
SET stid=".$null.",return=".$dbreturn." WHERE isbn=".$isbnbk." ")or
die("update ERROR :".mysql_error());}}}
```

```
                                   if  ($row==0){?><td><?php  echo  "RETURN
ERROR NO SUCH ISBN";?></td></table></tr>
                         <?php }}}
                  ?></td></table></td></tr></table></body>
```

The Return Code (Return.php)

5.3.2 Searching Procedure

5.3.2.1 Fin.php

In this page the administrator he can do search to check if there is any dept for the members or students when he return the books, because the system allow to members to take the books just one week and for every day delay it will calculate the dept, this page is done by using this code:

```
<?php
session_start();
if(! isset($_SESSION["userid"]))
{header("location: admin.php");}
?>
<body  bgcolor="#0066CC">
<?php include "mystyle.Css";?>
<table align="center"  width="800" cellpadding="0" cellspacing="0"><tr><td>
<table          align="center"                    width="800"border="0"          cellpadding="0"
cellspacing="0"><tr><td><?php include "banner.php";?></td></tr></table>
<?php
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR :".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR :".mysql_error());
if(($_SERVER["REQUEST_METHOD"]=="POST")and(! isset($_POST["payment"])))
{ $_POST["kamal"];
if (isset($_POST["kamal"]))
{$date_array = getdate();
                              $day=$date_array['mday'];
                              $mon=$date_array['mon'];
```

```
                                   $finemone=$mon;
                                   $year=$date_array['year'];
                                   if ($day<24)
                                   {$day1=$day+7;}
                                   if ($day==24)
                                   {$day1=1;$mon=$mon+1;}
                                   if ($day==25)
                                   {$day1=2;$mon=$mon+1;}
                                   if ($day==26)
                                   {$day1=3;$mon=$mon+1;}
                                   if ($day==27)
                                   {$day1=4;$mon=$mon+1;}
                                   if ($day==28)
                                   {$day1=5;$mon=$mon+1;}
                                   if ($day==29)
                                   {$day1=6;$mon=$mon+1;}
                                   if ($day==30)
                                   {$day1=7;$mon=$mon+1;}
                                   if ($day==31)
                                   {$day1=8;$mon=$mon+1;}
$sel1=mysql_query("SELECT * FROM mybook WHERE stid=".$_POST["kamal"]." order by
isbn asc ") or die("selection error".mysql_error());
$row1=mysql_num_rows($sel1);
if ($row1>0)
{echo "<hr>";
echo "There are ".$row1." Borowed Books";
$sign=0;
$worrning=0;
while($getinfon1=mysql_fetch_array($sel1))
{$fine=0;
/*echo "<hr>";*/
$myday=$getinfon1["duedate"];
$mymon=$getinfon1["mon"];
$myyear=$getinfon1["myyear"];
$dbfinevalue=$getinfon1["fine"];
if ($finemone==$mymon){
if ($day>$myday){$worrning="0"; $fine=($day-$myday);}
if ($day<$myday){$fine=($day-$myday);}}//else{$fine=0;}
if ($finemone<$mymon){$worrning="0";$fine=0;}
if ($finemone<$mymon){$worrning="0"; $calcmon=($mymon-$finemone);
if($calcmon==1){$fine=-(((30-$day)+$myday));$sign=1;}}
if ($finemone>$mymon){$worrning=1; $calcmon=($finemone-$mymon);
if($calcmon==1){$fine=((30-$myday)+$day);/*$sign=1;*/}}
if($fine>32){$fine=60;}
if($dbfinevalue>0){
?>
<table border="0" cellpadding="2" cellspacing="25" align="center"><tr>
<td height="20"><img src="images/blank.gif" height="20"></td>
<td style="color:#FFFFFF">ISBN</td><td><?php echo $getinfon1["isbn"];?></td>
<td style="color:#FFFFFF">Title</td><td><?php echo $getinfon1["bookname"];?></td>
<td style="color:#FFFFFF">Due Date</td><td><?php echo $myday;?>/<?php echo
$mymon;?>/<?php echo $myyear;?></td>
                <?php if($dbfinevalue>0)
                { $totalfine=$fine+$fine;?>
<form action="fin.php" name="id" method="post">
<input type="hidden" name="fyat" value="<?php echo "".$fine.""; ?> ">
<input type="hidden" name="upfineisbn" value="<?php echo $getinfon1["isbn"];?>">
<input type="hidden" name="totalfyat" value="<?php echo "".$totalfine.""; ?> ">
<td><input type="radio" name="payment" value="1" ></td>
<td><input type="submit" name="mid" value="pay >>"></td>
</form><td style="color:#FFFFFF"><?php } ?><?php echo "Due".$dbfinevalue." :YTL";
?></td></tr></table><?php
/* echo "<hr>";*/}}}}}
if (($_SERVER["REQUEST_METHOD"]=="POST") and (! isset($_POST["kamal"])))
{               $paid=$_POST["payment"];
                $upfine=$_POST["upfineisbn"];
                $pfine=$_POST["fyat"];
                $totalfine=$_POST["totalfyat"];
                if( isset($_POST["payment"])){
                $noval=0;
                $paidfine=0;
                $n1=0;
                $n2=0;
                $n3=0;
                $rn=0;
$debit=0;
if ($paid==1){ $finupdate=mysql_query("UPDATE mybook SET duedate=".$n1.",mon=".$n2.",myyear=".$n3.",fine=".$paidfine.",return=".$rn.",stid=".$noval." WHERE isbn=".$upfine." ")or die("update ERROR :".mysql_error());
?><table align="center" cellpadding="8" cellspacing="0"><tr><td style="color:#FFFFFF "><?php echo "".$pfine." YTL "; /*echo "".$totalfine."";*/ /*$debit=$totalfine-$pfine ; echo "Depit:=".$debit."";*/
?>
<td style="color:#990000" align="right"width="150">has been paid</td></td></tr></table>
<?php }}}?></td></tr></table></body>
```

The Searching Code (Fin.php)

5.3.3 Collecting Of Registers

5.3.3.1 Add New Member (StMember.php)

When the administrator need to add any new members or student he must use this page, in this page the admin he must fill the important information about the members to record in MySQL as you see below:

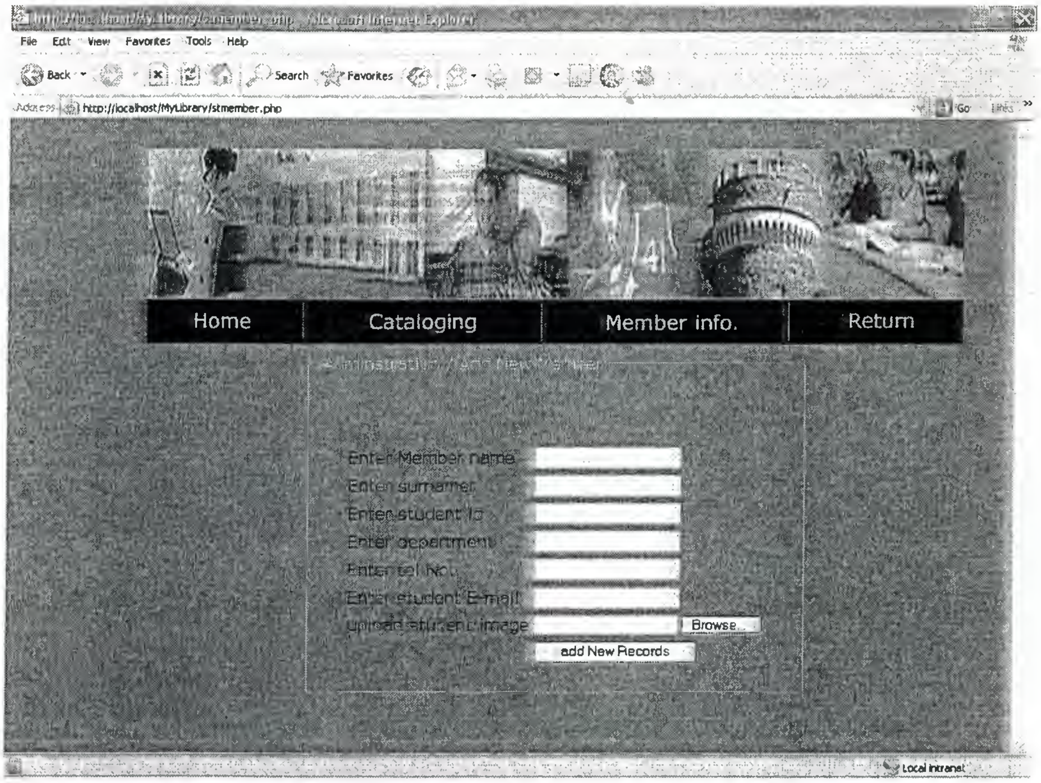


Figure 5.7 Add New Member (StMember.php)

And also it has done by using this code:

```
<?php
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR :".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR :".mysql_error());
if($_SERVER["REQUEST_METHOD"]=="POST")
{$stname=$_POST["stname"];
$surname=$_POST["surname"];
$stid=$_POST["stid"];
$dep=$_POST["department"];
$tel=$_POST["telno"];
$mail=$_POST["email"];
 if (isset($_POST["stname"]))
{$insertCat=mysql_query("INSERT INTO member(stname ,stsurname
,stidnumber,stdepartment,stpicture,telno ,stemail) VALUES ('".
$stname."','".$surname."','".$stid."','".$dep."','".$_FILES['fupload']['name']."','".$tel."','".$mail."')")
```

```
or die("inster ERROR :".mysql_error());
$source = $_FILES['fupload']['tmp_name'];
 $target = "upload/".$_FILES['fupload']['name'];
move_uploaded_file( $source, $target );// or die ("Couldn't copy");
$size = getImageSize( $target );
     $imgstr = "<p><img width=\"$size[0]\" height=\"$size[1]\" ";
      $imgstr .= "src=\"$target\" alt=\"uploaded image\" /></p>";?>
```

The Add New Member Code (StMember.php)

5.3.3.2 Add New Book (BookRegistry.php)

When the administrator need to add any new books he must use this page, in this page the admin he must fill the important information about the book to record in MySQL as you see below:

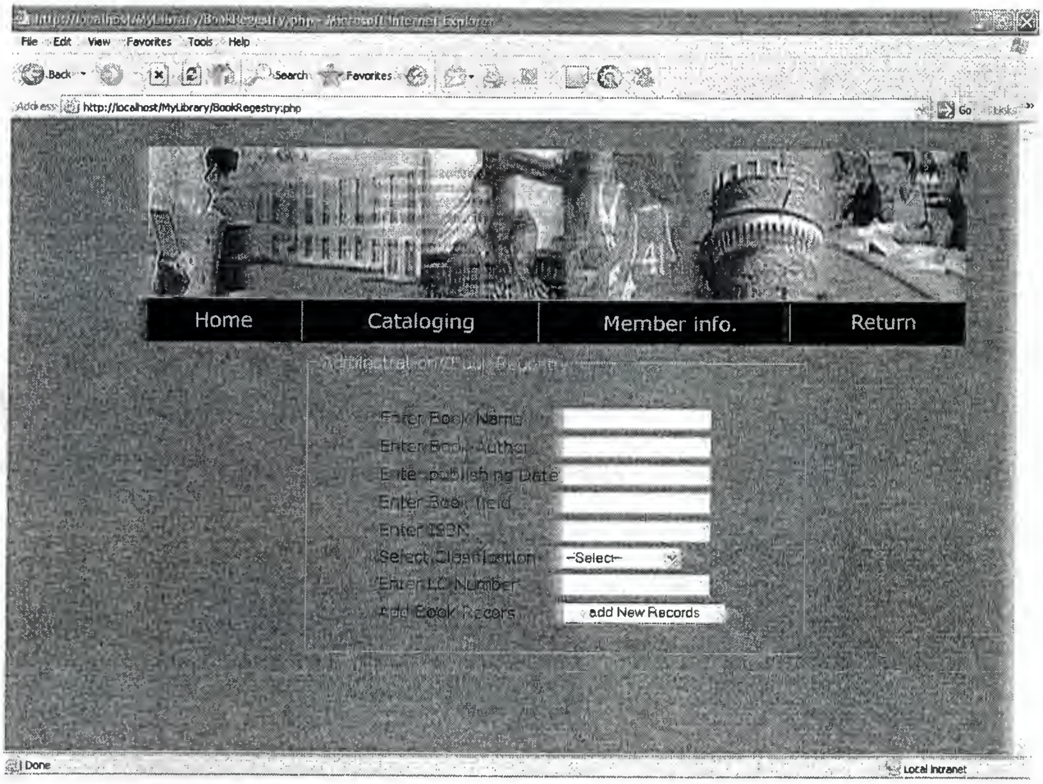


Figure 5.8 Add New Book(BookRegistry.php)

And also this page is done by this code:

```
<?php
session_start();
if(! isset($_SESSION["userid"]))
```

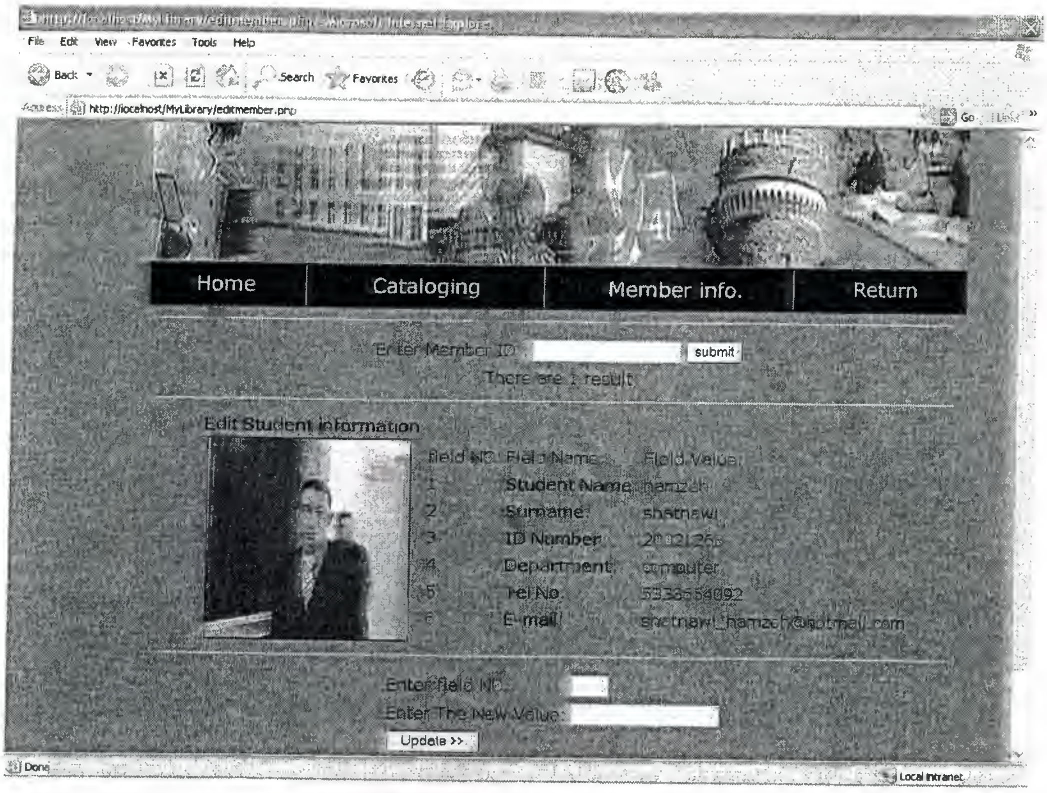
```
{header("location: admin.php");}
$db="mydb";
$connect=mysql_connect("localhost","root") or die("CONNECT ERROR :".mysql_error());
$select=mysql_select_db($db, $connect) or die("SELECT ERROR :".mysql_error());
if($_SERVER["REQUEST_METHOD"]=="POST")
{$bname=$_POST["bname"];
$bauthor=$_POST["bauthor"];
$pubdate=$_POST["Dpub"];
$field=$_POST["field"];
$ISBN=$_POST["isbn"];
$classif=$_POST["LCclasification"];
$shelving=$_POST["myshilveLC"];
//$stnumber=$_POST["stnum"];/*,".$stnumber."*/
if(isset($_POST["bname"]))
{$insertCat=mysql_query("INSERT                                                    INTO
mybook(bookname,author,publishigdate,shilveLC,bookfield,isbn,LCclasification) VALUES
('".$bname."','".$bauthor."','".$pubdate."','".$shelving."','".$field."','".$ISBN."','".$classif."')")
or die("inster ERROR :".mysql_error());}}?>
```

The Add New Book Code (BookRegistry.php)

5.3.4 Updating Procedure

5.3.4.1 Updating Member Information (EditMember.php)

When the administrator need to change some information about the member or student he must use this page, in this page the admin he must fill the new information about the member to updating the old data in MySQL as you see below:



Enter Member ID:

There are 1 result

Edit Student information

field NO	Field Name	Field Value
1	Student Name	hamzeh
2	Surname	shatnawi
3	ID Number	20021256
4	Department	computer
5	Tel No.	5338554092
6	E-mail	shatnawi_hamzeh@hotmail.com

Enter field No.:

Enter The New Value:

Figure 5.9 Updating Member Information (EditMember.php)

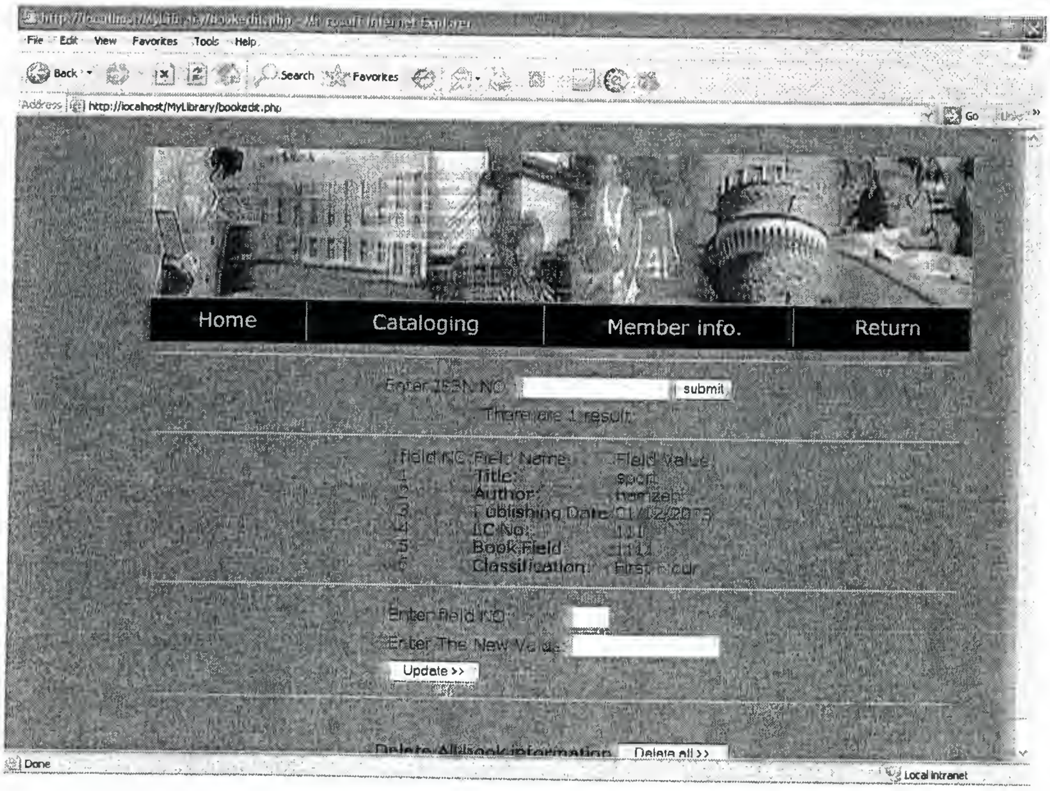
And it has done by using this code:

```
if (isset($_POST["num"]))
{if($editnum==1){$fineupdate1=mysql_query("UPDATE member SET
stname='".$upvalue."' WHERE stidnumber=".$id." ")or die("Update ERROR 1
:".mysql_error());}
if($editnum==2){$fineupdate2=mysql_query("UPDATE member SET
stsurname='".$upvalue."' WHERE stidnumber=".$id." ")or die("Update ERROR 2
:".mysql_error());}
if($editnum==3){$fineupdate3=mysql_query("UPDATE member SET
stidnumber=".$upvalue." WHERE stidnumber=".$id." ")or die("Update ERROR 3
:".mysql_error());}
if($editnum==4){$fineupdate4=mysql_query("UPDATE member SET
stdepartment='".$upvalue."' WHERE stidnumber=".$id." ")or die("Update ERROR 4
:".mysql_error());}
if($editnum==5){$fineupdate5=mysql_query("UPDATE member SET
telno='".$upvalue."' WHERE stidnumber=".$id." ")or die("Update ERROR 5
:".mysql_error());}
if($editnum==6){$fineupdate6=mysql_query("UPDATE member SET
stemail='".$upvalue."' WHERE stidnumber=".$id." ")or die("Update ERROR 6
:".mysql_error());}
if($editnum==7){$fineupdate7=mysql_query("UPDATE member SET
stpicture='".$updateimage."' WHERE stidnumber=".$id." ")or die("Update ERROR 7
:".mysql_error());}}}}}?>
```

The Updating Member Information Code (EditMember.php)

5.3.4.2 Updating Book Information (BookEdit.php)

When the administrator need to change some information about the book he must use this page, in this page the admin he must fill the correct information about the book to updating the old data in MySQL as you see below:



http://localhost/MyLibrary/bookedit.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://localhost/MyLibrary/bookedit.php
Go
Home
Cataloging
Member info.
Return
submit
Update >>
Delete all >>
Done
Local intranet

Enter ISBN NO:

There are 1 result:

Field NO	Field Name	Field Value
1	Title:	sport
2	Author:	hamzeh
3	Publishing Date	01/12/2003
4	LC No:	111
5	Book Field	1111
6	Classification:	First Floor

Enter field NO:

Enter The New Value:

Delete All book information

Figure 5.10 Updating Book Information (BookEdit.php)

And it has done by this code:

```
if (isset($_POST["num"]))
{if($editnum==1){$fineupdate1=mysql_query("UPDATE mybook SET
bookname='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 1
:".mysql_error());}
if($editnum==2){$fineupdate2=mysql_query("UPDATE mybook SET
author='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 2
:".mysql_error());}
if($editnum==3){$fineupdate3=mysql_query("UPDATE mybook SET
publishigdate=".$upvalue." WHERE isbn=".$id." ")or die("Update ERROR 3
:".mysql_error());}
if($editnum==4){$fineupdate4=mysql_query("UPDATE mybook SET
shilveLC='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 4
:".mysql_error());}
if($editnum==5){$fineupdate5=mysql_query("UPDATE mybook SET
bookfield='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 5
:".mysql_error());}
if($editnum==6){$fineupdate6=mysql_query("UPDATE mybook SET
isbn='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 6
:".mysql_error());}
if($editnum==7){$fineupdate7=mysql_query("UPDATE mybook SET
LCclasification='".$upvalue."' WHERE isbn=".$id." ")or die("Update ERROR 7
:".mysql_error());}}
```

The Updating Book Information Code (BookEdit.php)

5.3.5 Deleting Procedure

5.3.5.1 Deleting Member

When the administrator needs to delete the member from MySQL he must use the **MemberEdit.php** page to do deleting procedure, also it has done by using this code:

```
if(($_SERVER["REQUEST_METHOD"]=="POST") and (! isset($_POST["memid"])))
{if((! isset($_POST["num"]))){
$editnum=$_POST["delid"];
if (isset($_POST["delid"]))
{$mydelete=mysql_query("DELETE FROM member WHERE
stidnumber=".$editnum."");}
?><td>Member has deleted..</td><?php echo $editnum=$_POST["delid"];}}?>
```

The Deleting Member Code

5.3.5.2 Deleting Book

When the administrator needs to delete the books from MySQL he must use the **EditBook.php** page to do deleting procedure, also it has done by using this code:

```
if(($_SERVER["REQUEST_METHOD"]=="POST") and (! isset($_POST["isbn"])))
{if((! isset($_POST["num"]))){
$editnum=$_POST["delid"];
if (isset($_POST["delid"]))
{$mydelete=mysql_query("DELETE FROM mybook WHERE isbn=".$editnum."");}
?><td>Book ISBN  has deleted..</td><?php echo $editnum=$_POST["delid"];}}?>
```

The Deleting Books Code

CONCLUSION

During the point at the end of the project that using clever programming techniques and simple design, we can make a very well organized web page so that the user can be interact easily. There are many techniques for the web designing, that technique used is Personalized Home Page (PHP) and JavaScript.

The advantage of PHP file can contain text, HTML tags and scripts. Scripts in a PHP file are executed on the server. In addition, we can do to execute the connection with the server and send only report to the browser.

The advantage of JavaScript, we can call simple functions in head section and body section to make our web site like clock, date and animation, ect.

From the MySQL connection used to create an open connection to a data source, through this connection.

REFERENCES

[1] *Internet and World Wide web, How To program.* Third edition, Deitel, H. M

[2] *Database Design and prgramming, with Access, MySQL, and PHP, Second edition,*JOHN CARTER. published by JOHN WILEY & SON Ltd. in 1995.

[3] *Guide to Microsoft Access 2000 program*

[4] *Microsoft Encarta Encyclopedia 2002*

[5] http:// www.php.net

[6] *JavaScript for Beginners,* Copyright Martin Baier

[7] flicker and php by cal handerson

[8] Agent Based Internet (WWW) Services by László Gulyás, László Kovács, András Micsik, LászlRTersztenyák

[9] http://www.raleigh.ibm.com/iag/iagwp1.html

[10] http://www.trl.ibm.co.jp/aglets

[11] http://bf.cstar.ac.com/lifestyle/lf_dis.htm

[12] http://www.ics.uci.edu/ãckerman/docs/97.riao/dica.riao.html

[13] http://www.cs.washington.edu/homes/ctkwok/ingram.html

[14] http://www.php.net/

[15] http://www.w3schools.com/php/default.asp

[16] http://www.phpfreaks.com/