



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

**AN ONLINE WEB-BASED EDUCATIONAL  
SYSTEM**

**Graduation Project  
COM- 400**

**Student: İsmail Vehbi Özsoy (20033378)**

**Supervisor: Ümit Soyer**

**Nicosia – 2008**



## ACKNOWLEDGEMENTS

*" First of all, I would like to thank my supervisor Ümit Soyer for his support and precious advices through the Graduation Project process.*

*Second, I would like to express my gratitude to all my instructors at the Computer Engineering Department, for their support that made the work possible.*

*Finally, I would like to thank my family for their strong belief in my work and myself. "*

## **ABSTRACT**

The world of education and learning methods are rapidly changing after the Internet revolution. Nowadays, everyone can access huge information sources and educational documents with a computer and a modem while sitting at home. The old fashion ways of learning are quickly replacing with the new Internet based online educational systems and learning methods .

This project reports all the development processes and the steps of creating a dynamic online web-based educational system which covers specific educational topics.

The aim of this project is to supply useful information to the users. The web-based system contains specific information and lectures about popular and important programming languages and database related topics. Everyone can access those contents using the Internet. It is a kind of virtual school where the students follow the lectures.

The core system has a dynamic web interactive and design. It has a database to handle and response certain operations like user registration, authorization, uploading etc. Also the system and its users are divided into certain categories in order to increase the efficiency and the effectiveness of the system. The users have some privileges and limitations. The content management is based on these specifications.

The fundamental working principle of the system is to interact the users and provides them the requested contents considering their access rights.

The skeleton of the system is designed to be extendable and flexible. Additional features and functions can be adapted to the system very easily.

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>TABLE OF CONTENTS</b>	<b>iii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>vi</b>
<b>INTRODUCTION</b>	<b>1</b>
<b>CHAPTER ONE: INTRODUCTION TO PHP</b>	<b>3</b>
1.1. Overview of PHP	3
1.2. History of PHP	5
1.3. PHP Syntax	7
1.3.1. Statements	8
1.3.2. Data Types	9
1.3.3. Functions	10
1.3.4. Objects	10
<b>CHAPTER TWO: INTRODUCTION TO THE WEB, HTML, XHTML AND CSS</b>	<b>12</b>
2.1. How Does the Web Work	12
2.2. What is HTML and X	13
2.2.1. Elements	14
2.2.2. Attributes and Values	15
2.2.3. Structure of a XHTML Document	16
2.3. History and Versions of HTML and XHTML	19
2.3.1. HTML 1.0	19
2.3.2. HTML 2.0	19
2.3.3. HTML 3.2	20
2.3.4. HTML 4.0	20
2.3.5. XML 1.0	20
2.3.6. HTML 4.01	21
2.3.7. XHTML 1.0	21

2.3.8. XHTML 1.1	21
2.4. Cascading Style Sheets	22
2.4.1. Constructing a Style Rule	22
<b>CHAPTER THREE: MYSQL RELATIONAL DATABASE SYSTEM AND APACHE WEB SERVER</b>	<b>24</b>
3.1. Overview of MySQL Structure	24
3.1.1. Field Types	24
3.1.2. Indexes	26
3.2. Basic MySQL Syntax and Commands	27
3.3. Why MySQL	27
3.4. Apache Web Server	29
<b>CHAPTER FOUR: ONLINE WEB BASED EDUCATIONAL SYSTEM DESIGN DESCRIPTION</b>	<b>30</b>
4.1. Purpose	30
4.2. Software Requirements Analysis	30
4.2.1. PHP Server Side Scripting Language	30
4.2.2. MySQL Database System	30
4.2.3. Apache Web Server	31
4.3. System Design Overview and Basic Principles	31
4.4. System Structure and Web Contents	32
4.4.1. Database Section	32
4.4.2. Description and Working Principles of the Web Pages	34
4.4.2.1. Index Section	34
4.4.2.2. Lectures Section	36
4.4.2.3. Registration Section	38
4.4.2.4. User Login Section	39
4.4.2.5. User Information Section	39
4.4.2.6. Download Section	41
4.4.2.7. Information and Links Section	42
4.4.2.8. User Logout	44
4.4.2.9. Admin Login Section	44



4.4.2.10. Administration Section	44
4.4.2.11. Update User Information	46
4.4.2.12. Delete User Information	47
4.4.2.13. Add User/Admins and Manage Admins	47
4.4.2.14. Upload/Download Section	50
4.4.2.15. Admin Index Section	51
4.4.2.16. Admin Logout	52
<b>CONCLUSION</b>	<b>53</b>
<b>REFERENCES</b>	<b>54</b>
<b>APPENDIX A: MYSQL DATABASE AND ITS TABLES</b>	
<b>CREATION CODES AND QUERIES</b>	<b>55</b>
<b>APPENDIX B: USER'S PHP CODES AND MYSQL QUERIES</b>	<b>57</b>
<b>APPENDIX C: ADMINISTRATOR'S PHP CODES</b>	
<b>AND MYSQL QUERIES</b>	<b>98</b>

## **LIST OF ABBREVIATIONS**

PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language
RDBMS	Relational Database Management System
XML	Extensible Markup Language
SGML	Standard Generalized Markup Language
W3C	World Wide Web Consortium
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
WWW	World Wide Web
ADMIN	Administrator

## INTRODUCTION

The world of education, learning methods and the other educational systems are changing rapidly in nowadays. Technological developments and inventions directly effects this changing process.

The biggest revolution in technological and communicational areas is the invention of the Internet. It is obvious that the human-computer interaction is increased after this revolution. The old fashion methods in many areas are replaced with the new and complex ones. We can examine all the positive reflections caused by the Internet directly. For example we are now using the Internet to communicate, send mails, pay our bills or download an e-book to read.

Probably, the most important transformation takes place in the educational concepts and methods. The world becomes so small with the Internet and every barriers and limitations are meaningless in this virtual world. Now, the dynamic and interactive web pages provide all kinds of information. Endless knowledge can be accessible within a few seconds and anyone can access huge information sources with a few mouse clicks. Obviously, ordinary educational systems are deprecated and replaced with the modern ones in this situation.

It becomes an important concept to provide reliable and fast environments and mediums to the Internet users while accessing information through the Internet. The dynamic server-side web systems are designing to serve users interactively. From ordinary students to the proffesional users are using such kind of systems to access and get necessary information everyday. Educational resources are rapidly moving to the virtual learning portals to serve the public.

In this project the development of an online web-based system is covered from every aspects. The project consists of introduction, 4 chapters and conclusion.



Chapter One presents the introduction to PHP. The history of PHP, the structure of the language, syntax rules and server-side scripting concepts are covered in this chapter.

Chapter Two describes Web, HTML, XHTML and CSS topics. The working basics of the Web is briefly introduced. HTML/XHTML tags and elements, Cascading Style Sheet rules are also presented. Also every version of HTML and XHTML are discussed.

Chapter Three describes the fundamental structure of Relational Database Systems considering MySQL. The important features of MySQL is reported. At last Apache Web Server concept is briefly presented.

Chapter Four covers the most important concept of the project. The core system design and working principles of the online web-based educational system are deeply described and explained in this chapter. Database structure, system design description and contents of the system are reported in this chapter.

Finally, the conclusion section presents the important results obtained within the project.

## CHAPTER -1-

### INTRODUCTION TO PHP

#### 1.1 OVERVIEW OF PHP

PHP (PHP: Hypertext Preprocessor) is a widely-used Open Source general-purpose computer scripting language, originally designed for producing dynamic web pages. It is for server-side scripting, but can be used from a command line interface or in stand alone graphical applications. It is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

```
<div id="display_date">
<?php

$display_time = time();

$current_date = date("d.m.Y",$display_time);

$current_hour = date("H:i",$display_time);
?>
</div>
```

**Figure 1.1.** Embedded PHP code in an HTML document

PHP is a very strong language. It has a lot of high-level programming language properties. So we can create stand-alone applications or object-oriented programs with PHP besides creating dynamic web pages. PHP is a combination of the various popular programming languages. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

There are three main areas where PHP scripts are used:

- **Server-side scripting:** This is the most traditional and main target field for PHP. Servers-side process needs three major things to work correctly considering PHP; The PHP parser (CGI or server module), a web server and a web browser. We need to run the web server, with a connected PHP installation. We can access the PHP program output with a web browser, viewing the PHP page through the server.
- **Command line scripting:** We can make a PHP script to run it without any server or browser. The only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.
- **Writing desktop applications:** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

PHP is a server-side scripting language, which means that the scripts are executed on the server (the computer where the Web site is located). This is different than JavaScript, another popular language for dynamic Web sites. JavaScript is executed by the browser, on the user's computer. Thus, JavaScript is a clientside language. Because PHP scripts execute on the server, PHP can dynamically create the HTML code that generates the Web page, which allows individual users to see customized Web pages. Web page visitors see the output from scripts, but not the scripts themselves. The large majority of server-side scripts are related to either getting information from the user and saving it somewhere, or retrieving information from somewhere and presenting it. This "somewhere" is usually called a database, and all the serious and useful web applications needs a database. One of the strongest and most significant features in



PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple.

As I previously mentioned, the common and important usage or working principle of PHP is generally running on a web server, taking PHP code as its input and creating web pages as output. In this kind of usage, PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML.

PHP can be deployed on most web servers and on almost every operating system and platform free of charge. PHP is installed on more than 20 million websites and 1 million servers. It is also the most popular Apache module among computers using Apache as a web server. The most recent major release of PHP was version 5.2.6 on May 1, 2008.

## **1.2 HISTORY OF PHP**

PHP originally stood for Personal Home Page. It began in 1994 as a set of Common Gateway Interface binaries written in the C programming language by the Danish/Greenlandic programmer Rasmus Lerdorf. Lerdorf initially created these Personal Home Page Tools to replace a small set of Perl scripts he had been using to maintain his personal homepage. The tools were used to perform tasks such as displaying his resume and recording how much traffic his page was receiving. He combined these binaries with his Form Interpreter to create PHP/FI, which had more functionality. PHP/FI included a larger C implementation and could communicate with databases enabling the building of simple, dynamic web applications. He released PHP publicly on June 8, 1995 to speed up the finding of bugs and improving the code. This release was named PHP version 2 and already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax was similar to Perl but was more limited, simpler, and less consistent.

Zeev Suraski and Andi Gutmans, two Israeli developers at the Technion IIT, rewrote the parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive initialism PHP: Hypertext Preprocessor. The development team officially released PHP/FI 2 in November 1997 after months of beta testing. Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new rewrite of PHP's core, producing the Zend Engine in 1999. They also founded Zend Technologies in Ramat Gan, Israel, which manages the development of PHP.

On May 22, 2000, PHP 4, powered by the Zend Engine 1.0, was released. On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. The most recent update released by The PHP Group is for the older PHP version 4 code branch. As of January 2008, this branch is up to version 4.4.8. PHP 4 is no longer under active development but will be supported by security updates until August 8, 2008.

In 2008, PHP 5 became the only stable version under development. Late static binding has been missing from PHP and will be added in version 5.3. Alongside PHP 5, PHP 6 is also under active development. Major changes include the removal of `register_globals`, magic quotes, and safe mode.

PHP does not have complete native support for Unicode or multibyte strings; unicode support will be included in PHP 6. Many high profile open source projects ceased to support PHP 4 in new code as of February 5, 2008, due to the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

It runs in both 32-bit and 64-bit environments, but on Windows the only official distribution is 32-bit, requiring Windows 32-bit compatibility mode to be



enabled while using IIS in a 64-bit Windows environment. There is a third-party distribution available for 64-bit Windows.

### **1.3 PHP SYNTAX**

PHP's design philosophy and syntax structure is like other programming languages, especially like C. However, PHP started life as a handy utility for making quick-and-dirty Web pages, it emphasizes convenience for the programmer over correctness; rather than have a programmer do the extra work of redundantly specifying what is meant by a piece of code, PHP requires the minimum and then tries its best to figure out what was meant. Among other things, this means that certain syntactical features that show up in other languages, such as variable declarations and function prototypes, are simply not necessary.

The one of the most important thing to understand about PHP syntax is that it applies only within PHP. Because PHP is embedded in HTML documents, every part of such a document is interpreted as either PHP or HTML, depending on whether that section of the document is enclosed in PHP tags. PHP syntax is relevant only within PHP.

Previously mentioned, broadly speaking, PHP is like the C programming language. Because of that, a lot of C programming fundamentals are also valid for PHP.

PHP is whitespace insensitive. Whitespace is the thing you type that is typically invisible on the screen, including spaces, tabs, and carriage returns (end-of-line characters). But, of course it does not mean that we can leave spaces between variable names or in other declaration statements.

Another important thing in the syntax is the case sensitivity. PHP is sometimes case sensitive. In particular, all variables are case sensitive. The different capitalization schemes make for different variables. On the other hand, unlike in C, function names are not case sensitive, and neither are the basic language constructs (if, then, else, while, and the like).

### 1.3.1 Statements

A statement in PHP is any expression that is followed by a semicolon (;). If expressions correspond to phrases, statements correspond to entire sentences, and the semicolon is the full stop at the end. Any sequence of valid PHP statements that is enclosed by the PHP tags is a valid PHP program.

```
<?php  
  
print("Hello"); // side effect is printing to screen  
2 * 3 + 4; // useless - no side effect  
$value_num = 3 * 4 + 5; // side effect is assignment  
store_in_database(49.5); // side effect to Database  
  
?>
```

Figure 1.2. Simple PHP statements

In the above figure, there are some comments exist next to the statements. Comments are the portion of a program that exists only for the human reader. Comments are invaluable in helping the next person who reads your code figure out what you were thinking when you wrote it, even when that person is yourself a week from now. PHP drew its inspiration from several different programming languages, most notably C, Perl, and Unix shell scripts. As a result, PHP supports styles of comments from all those languages, and those styles can be intermixed freely in PHP code.

```
<?php  
  
/* This is  
a comment in  
PHP */  
  
//This is another comment  
  
# This is a comment, and  
# this is the second line of the comment  
  
?>
```

Figure 1.3. Some comment types

The main way to store information in the middle of a PHP program is by using a variable, a way to name and hang on to any value that you want to use later.

Here are the most important things to know about variables in PHP:

- All variables in PHP are denoted with a leading dollar sign (\$).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables have no intrinsic type other than the type of their current value.
- Variables used before they are assigned have default values.

There is another syntax difference in data types. PHP automatically determines which type we use in our codes. There is no need to write int, float or any other data type name. In the next section, I am going to examine the data types deeply.

```
<?php
$true_num = 3 + 0.14159;
$true_str = "Tried and true"
$true_array[49] = "An array element";
>false_array = array();
>false_null = NULL;
>false_num = 999 - 999;
>false_str = "";
?>
```

Figure 1.4. Data types

### 1.3.2 Data Types

PHP stores whole numbers in a platform-dependent range. This range is typically that of 32-bit signed integers. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Real numbers are also stored in a platform-



specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is NULL. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, or heredoc syntax.

### **1.3.3 Functions**

PHP has hundreds of base functions and thousands more from extensions. Functions are not first-class functions and can only be referenced by their name. User-defined functions can be created at any time without being prototyped. Functions can be defined inside code blocks, permitting a run-time decision as to whether or not a function should be defined. Function calls must use parentheses, with the exception of zero argument class constructor functions called with the PHP new operator, where parentheses are optional. PHP supports quasi-anonymous functions through the `create_function()` function, although they are not true anonymous functions because anonymous functions are nameless, but functions can only be referenced by name, or indirectly through a variable `$function_name();`, in PHP.

### **1.3.4 Objects**

Basic object-oriented programming functionality was added in PHP 3. Object handling was completely rewritten for PHP 5, expanding the feature set and

enhancing performance. In previous versions of PHP, objects were handled like primitive types. The drawback of this method was that the whole object was copied when a variable was assigned or passed as a parameter to a method. In the new approach, objects are referenced by handle, and not by value. PHP 5 introduced private and protected member variables and methods, along with abstract classes and final classes as well as abstract methods and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. Objects implementing `ArrayAccess` can be used with array syntax and objects implementing `Iterator` or `IteratorAggregate` can be used with the `foreach` language construct. There is no virtual table feature in the engine, so static variables are bound with a name instead of a reference at compile time.

If the developer creates a copy of an object using the reserved word `clone`, the Zend engine will check if a `_clone()` method has been defined or not. If not, it will call a default `_clone()` which will copy the object's properties. If a `_clone()` method is defined, then it will be responsible for setting the necessary properties in the created object. For convenience, the engine will supply a function that imports the properties of the source object, so that the programmer can start with a by-value replica of the source object and only override properties that need to be changed.

The basic syntax structure of PHP is similar to any other programming languages, especially like C, Perl and Java. The assignment statements, control statements, scopes, function declarations, and the other primitive concepts are similar to those languages. There are quite differences which I mentioned above sections. However the programming logic stays the same all the time.



## **CHAPTER -2-**

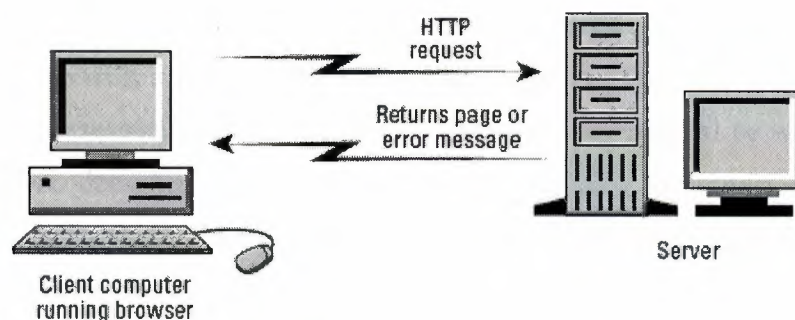
### **INTRODUCTION TO THE WEB, HTML, XHTML AND CSS**

The World Wide Web (the Web, for short) is a network of computers able to exchange text, graphics, and multimedia information via the Internet. By sitting at a computer that is attached to the Web, using either a dialup phone line or a much faster broadband (Ethernet, cable, or DSL connection), you can visit Web-connected computers next door, at a nearby university, or halfway around the world. And you can take full advantage of the resources these computers make available, including text, graphics, videos, sounds, and animation. We can think of the Web as the multimedia version of the Internet. The important thing is that the huge part of the information exchange is based on Web pages which contains HTML, XHTML and CSS content. They are the main building blocks of the Web pages.

#### **2.1 HOW DOES THE WEB WORK**

The computers that make all these Web pages available are called Web servers. On any computer that's connected to the Web, you can run an application called a Web browser. Technically, a Web browser is called a Web client, a program that's able to contact a Web server and request information. When the Web server receives the requested information, it looks for this information within its file system, and sends out the requested information via the Internet. They all speak a common language called HyperText Transfer Protocol (HTTP).

HTTP isn't really a language like the ones people speak. It's a set of rules or procedures, called protocols, that enables computers to exchange information over the Web. Regardless of where these computers reside they can communicate with each other through HTTP.



**Figure 2.1.** How HTTP works

The following steps are the basic explanation of how the HTTP works:

- Most Web pages contain hyperlinks, which are specially formatted words or phrases that enable you to access another page on the Web. Although the hyperlink usually doesn't make the address of this page visible, it contains all the information needed for your computer to request a Web page from another computer.
- When you click the hyperlink, your computer sends a message called an HTTP request. This message says, in effect, "Please send me the Web page that I want."
- The Web server receives the request, and looks within its stored files for the Web page you requested. When it finds the Web page, it sends it to your computer, and your Web browser displays it. If the page isn't found, you see an error message, which probably includes the HTTP code for this error: 404, "Not Found."

## **2.2 WHAT IS HTML AND XHTML**

Hypertext Markup Language (HTML) enables you to mark up text so that it can function as hypertext on the Web. The term markup comes from printing; editors mark up manuscript pages with funny-looking symbols that tell the printer how to print the page. HTML consists of its own set of symbols that tell Web browsers how to display the page. These symbols, called elements.

The Extensible Hypertext Markup Language (XHTML), is a markup language that has the same depth of expression as HTML, but also conforms to

XML syntax. Whereas HTML is an application of Standard Generalized Markup Language (SGML), a very flexible markup language, XHTML is an application of XML, a more restrictive subset of SGML. Because they need to be well-formed, true XHTML documents allow for automated processing to be performed using standard XML tools, unlike HTML, which requires a relatively complex, lenient, and generally custom parser. XHTML can be thought of as the intersection of HTML and XML in many respects, since it is a reformulation of HTML in XML. XHTML 1.0 became a World Wide Web Consortium (W3C) recommendation on January 26, 2000. XHTML 1.1 became a W3C recommendation on May 31, 2001.

XHTML is a proposed successor to HTML. As such, many consider XHTML as the current or latest version of HTML. However, the W3C continues to recommend the use of HTML 4.01 for web publishing, while the specification for HTML 5 is being actively developed.

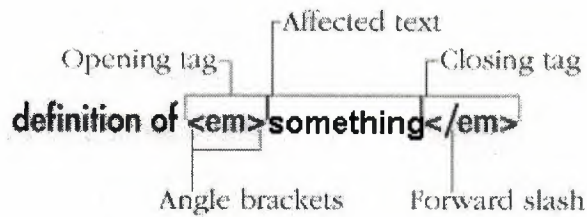
XHTML is an ingenious system of including information about the content right in a text document. This information called markup, accounting for them in XHTML can include formatting instructions as well as details about the relationships between parts of the document. However, because the markup itself is comprised chiefly of text, the document is practically universally accessible. XHTML has three principal types of markup: elements, attributes, and values.

### **2.2.1 Elements**

Elements are like little labels that identify and structure the different parts of a Web page: "This is a header, that thing over there is a paragraph, and that is important information." Some elements have one or more attributes.

Elements can contain text and/or other elements, or they can be empty. A non-empty element consists of an opening tag (the element's name and attributes, if any, enclosed in less than or greater than signs), the content, and a closing tag.



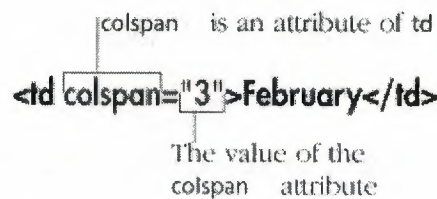


**Figure 2.2.** Typical XHTML element

An empty element looks like a combination opening and closing tag, with an initial less than sign, the element's name followed by any attributes it may have, a space, a forward slash, and the final greater than sign. In XHTML, the closing tag is always required but in HTML, it is sometimes optional.

### 2.2.2 Attributes and Values

Attributes contain information about the data in the document, as opposed to being that data itself. In XHTML, an attribute's value must always be enclosed in quotation marks. In HTML, quotes may sometimes be omitted.



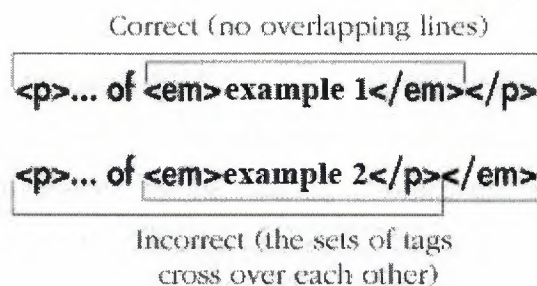
**Figure 2.3.** Attributes and values

Some attributes can accept any value at all, others are more limited. Perhaps the most common are those that accept enumerated or predefined values. In other words, we must select a value from a standard list of choices. In XHTML, enumerated values are always written in all lowercase letters. In HTML, the case doesn't matter.

Many attributes require a number or percentage for their value, particularly those describing size and length. A numeric value never includes units. Where

units are applicable, as in the height of text or the width of an image, they are understood to be pixels. The attributes controlling color can contain values that are either a color name or a hexadecimal representation of the red, green, and blue content of the color. Note that XHTML does not support numeric or percentage values for color. Some attributes reference other files and thus must contain values in the form of a URL, or Uniform Resource Locator, a file's unique address on the Web.

It is important to note that when elements contain other elements, each element must be properly nested, that is fully contained within its parent. Whenever you use a closing tag, it should correspond to the last unclosed opening tag. In other words, first open A then open B, then close B and then close A.



**Figure 2.4.** Elements must be properly nested

### 2.2.3 Structure of an XHTML Document

All the web pages should begin with a DOCTYPE which declares what type of HTML or XHTML we're using. The DOCTYPE lets browsers know what to expect and tells validators how to judge your code in order to check its syntax. Then, signal the beginning of the actual code with the opening html tag.

The XHTML standard defines three document type definitions. These are "XHTML 1.0 Strict", "XHTML 1.0 Transitional" and "XHTML 1.0 Frameset". The most common is the "XHTML Transitional".



We use “Strict” type when we want really clean markup, free of presentational clutter. It sticks to the strict rules.

The “Transitional” type is used when we need to take advantage of HTML's presentational features. It supports both HTML and XHTML markup rules and syntax.

Use “Frameset” type if we want to use HTML Frames to partition the browser window into two or more frames.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

**Figure 2.5** The DOCTYPE for a transitional XHTML document

Most Web pages are divided into two sections: the head and the body. The head section is where you define the title of your page, include information about your page for search engines, set the location of your page, add style sheets, and write scripts. Except for the title, the content of the head section is not readily visible to the visitor.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>

</head>
<body>

</body>
</html>
```

**Figure 2.6.** Example of a basic XHTML document

The head and body tags are required in XHTML. They're optional in HTML but even if you don't physically type them, the browser acts as if they are there and even lets you assign styles to them.

All text documents, XHTML files included, are saved with a character encoding. Since there are many encodings in use in the world, it's a good idea to declare which encoding your page was saved in right in the XHTML code. This makes it easier for browsers on systems with different default encodings to view the characters in your pages correctly. In the head section of your page, type `<meta http-equiv="content-type" content="text/html; charset=encoding" />`, where encoding is the character encoding with which you saved the file.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type"
content="text/html; charset=utf-8" />

</head>
<body>

</body>
</html>
```

**Figure 2.7.** Example of the character encoding declaration

Each XHTML page must have a title element. A title should be short and descriptive. In most browsers, the title appears in the title bar of the window. Perhaps even more importantly, the title is used by search indexes like Yahoo and Google as well as in your visitors' browsers' history lists and bookmarks.

The other important XHTML concepts are section headers, paragraphs, divisions. We can examine them all in the following figure:

```
<head>
<title>Welcome</title>
</head>
<body>
<div id="header">
<h1>This is the first Header</h1>
<h2>This is the other Header</h2>
<p>New Paragraph</p>
</div>
```

Figure 2.8. Example of various (X)HTML tags

## 2.3 HISTORY AND VERSIONS OF HTML AND XHTML

### 2.3.1 HTML 1.0

HTML 1.0 was never formally specified by the W3C because the W3C came along too late. HTML 1.0 was the original specification Mosaic 1.0 used, and it supported few elements. What you couldn't do on a page is more interesting than what you could do. You couldn't set the background color or background image of the page. There were no tables or frames. You couldn't dictate the font. All inline images had to be GIFs; JPEGs were used for out-of-line images. And there were no forms. Every page looked pretty much the same: gray background and Times Roman font. Links were indicated in blue until you'd visited them, and then they were red. Because scanners and image-manipulation software weren't as available then, the image limitation wasn't a huge problem. HTML 1.0 was only implemented in Mosaic and Lynx (a text-only browser that runs under UNIX).

### 2.3.2 HTML 2.0

Huge strides forward were made between HTML 1.0 and HTML 2.0. An HTML 1.1 actually did exist, created by Netscape to support what its first browser could do. Because only Netscape and Mosaic were available at the time (both written under the leadership of Marc Andreessen), browser makers were in the



habit of adding their own new features and creating names for HTML elements to use those features. Between HTML 1.0 and HTML 2.0, the W3C also came into being, under the leadership of Tim Berners-Lee, founder of the Web. HTML 2.0 was a huge improvement over HTML 1.0. Background colors and images could be set. Forms became available with a limited set of fields, but nevertheless, for the first time, visitors to a Web page could submit information. Tables also became possible.

### **2.3.3 HTML 3.2**

HTML 3.2 was vastly richer than HTML 2.0. It included support for style sheets (CSS level 1). Even though CSS was supported in the 3.2 specification, the browser manufacturers didn't support CSS well enough for a designer to make much use of it. HTML 3.2 expanded the number of attributes that enabled designers to customize the look of a page (exactly the opposite of HTML 4). HTML 3.2 didn't include support for frames, but the browser makers implemented them anyway.

### **2.3.4 HTML 4.0**

Then HTML 4.0 specification came, and it didn't contain so much new concepts and features. HTML 4.0 now clearly deprecates any uses of HTML that relate to forcing a browser to format an element a certain way. All formatting has been moved into the style sheets. With formatting information strewn throughout the pages, HTML 3.2 had reached a point where maintenance was expensive and difficult. This movement of presentation out of the document, once and for all, should facilitate the continued rapid growth of the Web.

### **2.3.5 XML 1.0**

Extensible Markup Language (XML) was originally designed to meet the needs of large-scale electronic publishing. As such, it was designed to help separate structure from presentation and provide enough power and flexibility to be applicable in a variety of publishing applications. In fact, many modern word processing programs contain XML components or even export their documents in XML-compliant formats.

### **2.3.6 HTML 4.01**

HTML 4.01 is a minor revision of the HTML 4.0 standard. In addition to fixing errors identified since the inception of 4.0, HTML 4.01 also provides the basis for meanings of XHTML elements and attributes, reducing the size of the XHTML 1.0 specification.

### **2.3.7 XHTML 1.0**

Extensible HyperText Markup Language (XHTML) is the first specification for the HTML and XML cross-breed. XHTML was created to be the next generation of markup languages, infusing the standard of HTML with the extensibility of XML. It was designed to be used in XML-compliant environments, yet compatible with standard HTML 4.01 user agents.

### **2.3.8 XHTML 1.1**

The most recent XHTML Recommendation by the W3C is XHTML 1.1: Module-based XHTML. This is a reformulation of XHTML 1.0 Strict, with minor modifications, using a set of modules selected from a larger set defined in Modularization of XHTML, which itself provides a modularized framework, a standard set of those modules, and various conformance definitions. All deprecated features of HTML, such as presentational elements and framesets, and lang and anchor name attributes, which were still allowed in XHTML 1.0 Strict, have been removed from this revision. Presentation is controlled purely by CSS. This version also allows for ruby markup support, needed for East-Asian languages (especially CJK).

Module-based XHTML adds the concept of a "strictly conforming" document: it must be a complete document containing only elements defined in the modules required by XHTML 1.1. For example, if a document is extended with the predefined XHTML Frames module, or some module formulated by a third-party, it does not qualify as strictly conforming XHTML 1.1 for use on the Web.



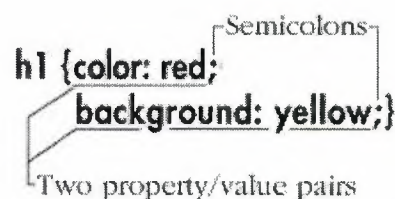
## 2.4 CASCADING STYLE SHEETS

While XHTML gives your Web pages their basic structure, CSS (Cascading Style Sheets) defines their appearance. A style sheet is simply a text file that contains one or more rules that determine through properties and values how certain elements in your Web page should be displayed. There are CSS properties for controlling such basic formatting as font size and color, layout properties such as positioning and float, and print controls for deciding where page breaks should appear. CSS also has a number of dynamic properties, which allow items to appear and disappear and are useful for creating drop-down menus and other interactive components.

The wonderful thing about CSS is that it can be created outside of a Web page and then applied to all the pages on your site at once. It is flexible, powerful, and efficient and can save you lots of time and bandwidth. To get the full benefit of CSS, your Web pages must be well structured. CSS appreciates the fastidiousness of XHTML because then it knows just what it has to do.

### 2.4.1 Constructing a Style Rule

Each style rule in a style sheet has two main parts: the selector, which determines which elements are affected and the declaration, made up of one or more property/value pairs, which specifies just what should be done.



The diagram shows a CSS style rule: `h1 {color: red; background: yellow;}`. A bracket on the right side of the rule, spanning from the opening curly brace to the closing curly brace, is labeled "Semicolons". Another bracket on the left side, spanning from the opening curly brace to the closing curly brace, is labeled "Two property/value pairs".

**Figure 2.9.** Simple style rule

We can arrange web page setup, layout, format and other visual concepts using CSS techniques so, it is very useful.

We declare CSS in the head section of the HTML file and codes are executed to create visual effects and formation to the related document.

```
<head>
<style type="text/css">
body{ background-color:#CCCCCC; font-family: Cambria; font-size: 16px;
color:#000000; text-align:justify}
h1{ margin-top:3%;text-align:center; color: #0066FF;
border:solid; border-color:#0066FF; margin-left:2%;
margin-right:2%; font-size:22px }
h2{margin-top:3%;text-align:center;
color: #0066FF; border:solid;
border-color:#0066FF; margin-left:2%; margin-right:2%; font-size:18px}
a{color:#0066FF; font:bold; }
hr{ color: #0066FF}
</style>
</head>
```

**Figure 2.10.** Style rule declaration in the head section

## **CHAPTER -3-**

# **MYSQL RELATIONAL DATABASE SYSTEM AND APACHE WEB SERVER**

### **3.1 Overview of MySQL Structure**

MySQL is a relational database system, which basically means that it can store bits of information in separate areas and link those areas together. You can store virtually anything in a database: the contents of an address book, a product catalog, or even a wish list of things you want for your birthday.

Because MySQL is a relational database management system, it allows you to separate information into tables or areas of pertinent information. In nonrelational database systems, all the information is stored in one big area, which makes it much more difficult and cumbersome to sort and extract only the data you want. In MySQL, each table consists of separate fields, which represent each bit of information. For example, one field could contain a customer's first name, and another field could contain his last name. Fields can hold different types of data, such as text, numbers, dates, and so on.

We create database tables based on what type of information we want to store in them. The separate tables of MySQL are then linked together with some common denominator, where the values of the common field are the same. For an example of this structure, imagine a table that includes a customer's name, address, and ID number, and another table that includes the customer's ID number and past orders he has placed. The common field is the customer's ID number, and the information stored in the two separate tables would be linked together via fields where the ID number is equal. This enables you to see all the information related to this customer at one time.

#### **3.1.1 Field Types**

When we create a table initially, we need to tell MySQL server what types of information will be stored in each field. The data is stored in the database



according that specified types. We can hold numbers, images, dates or any other kind of information by using field types.

The following table shows the most important types which MySQL database system contains:

**Table 3.1** The most important field types in MySQL

MySQL Field Type	Description	Example
<code>char (length)</code>	Any character can be in this field, but the field will have a fixed length.	Customer's State field always has two characters.
<code>varchar (length)</code>	Any character can be in this field, and the data can vary in length from 0 to 255 characters. Maximum length of field is denoted in parentheses.	Customer's Address field has letters and numbers and varies in length.
<code>int (length)</code>	Numeric field that stores integers that can range from -2147483648 to +2147483647, but can be limited with the <code>length</code> parameter. The <code>length</code> parameter limits the number of digits that can be shown, not the value. Mathematical functions can be performed on data in this field.	Quantity of a product on hand.

The basic types allow us to store fixed length, variable length characters and integer data. Sometimes it doesn't matter the size of the information or we can omit the size because the data is so small, or we want to store images, long texts, or other huge bytes of data. In this situations we need smaller or bigger sized fields.

The following table shows the other important types necessary to hold huge or small information:

**Table 3.2** The other field types in MySQL

MySQL Field Type	Description
<code>mediumint (length)</code>	Numeric field that stores integers from -8388608 to 8388607. (Adding the <code>unsigned</code> parameter allows storage of 0 to 16777215.)
<code>bigint (length)</code>	Numeric field that stores integers from -9223372036854775808 to 9223372036854775807. (Adding the <code>unsigned</code> parameter allows storage of 0 to 18446744073709551615.)
<code>tinytext</code>	Allows storage of up to 255 characters.
<code>mediumtext</code>	Allows storage of up to 1677215 characters.
<code>longtext</code>	Allows storage of up to 4294967295 characters.
<code>blob</code>	Equal to a text field, except it is case-sensitive when sorting and comparing. Stores up to 65535 characters.
<code>tinyblob</code>	Equal to the <code>tinytext</code> field, except it is case-sensitive when sorting and comparing.
<code>mediumblob</code>	Equal to the <code>mediumtext</code> field except it is case-sensitive when sorting and comparing.
<code>longblob</code>	Equal to the <code>longtext</code> field except it is case-sensitive when sorting and comparing.
<code>year (length)</code>	Stores a year in four-character format (by default). It is possible to specify a two-year format by signifying that with the <code>length</code> parameter.

### 3.1.2 Indexes

MySQL uses indexes to expedite the process of searching for a particular row of information. By using an internal filing system, MySQL can jump to the approximate location of your data much more quickly. It does this through the use of indexes, also known as keys.

MySQL requires at least one index on every table, so that it has something to go by. Normally, you would use a primary key, or unique identifier that helps keep the data separate. This field must be “not null” and “unique”; an example would be a customer ID number to keep your customers separate. (You could easily have two “John Smith” entries, so you need a way to tell the difference.)

MySQL also provides a feature that allows a value in a field to be automatically incremented by one. This `auto_increment` parameter is useful for making sure your primary key is being populated with unique numbers.



### 3.2 Basic MySQL Syntax and Commands

Although it is quite possible to access MySQL directly through a shell command prompt, we can access it through PHP codes. Regardless of the mode by which the MySQL

server gets its information and requests, the syntax is basically the same.

Typically, you keep the MySQL commands in all caps, although this is not necessary. The purpose of this is to help keep the MySQL syntax separate from the variables and table or database names. It is not case sensitive.

Common commands are the followings:

- **CREATE:** Creates new databases and tables
- **ALTER:** Modifies existing tables
- **SELECT:** Chooses the data you want
- **DELETE:** Erases the data from your table
- **DESCRIBE:** Lets you know the structure and specifics of the table
- **INSERT INTO "tablename" VALUES:** Puts values into the table
- **UPDATE:** Lets you modify data already in a table
- **DROP:** Deletes an entire table or database

### 3.3 WHY MySQL

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

MySQL is an open source, SQL Relational Database Management System (RDBMS) that is free for many uses. In commercial usage it has a certain price, but also it is still very cheap from the other dbms. Early in its history, MySQL occasionally faced opposition due to its lack of support for some core SQL constructs such as subselects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety



of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. MySQL has not failed to reward the loyalty of these users with the addition of both subselects and foreign keys as of the 4.1 series.

Databases in general are useful, arguably the most consistently useful family of software products-the “killer product” of modern computing. Like many competing products, both free and commercial, MySQL isn’t a database until you give it some structure and form. You might think of this as the difference between a database and an RDBMS (that is, RDBMS plus user requirements equals a database).

The main building block of the dynamic web systems is the databases. MySQL is the perfect choice for providing data via the Internet because of its ability to handle heavy loads and its advanced security measures.

Popular features of this program are as follows:

- Multiple CPUs usable through kernel threads
- Multi-platform operation
- Numerous column types cover virtually every type of data
- Group functions for mathematical calculations and sorting
- Commands that allow information about the databases to be easily and succinctly shown to the administrator
- Function names that do not affect table or column names
- A password and user verification system for added security
- Up to 32 indexes per table permitted; this feature has been successfully implemented at levels of 60,000 tables and 5,000,000,000 rows (version 4.1.2, currently in development, will allow 64 indexes)
- International error reporting usable in many different countries

The dynamic web contents can be created using both PHP and MySQL very easily and they will be very powerful and efficient. Because of all those features, the best choice in development is PHP and MySQL combination with a strong web server.

### 3.4 APACHE WEB SERVER

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

The features and server capabilities available in this version include the following:

- Password-protected pages for a multitude of users
- Customized error pages
- Display of code in numerous levels of HTML, and the capability to determine at what level the browser can accept the content
- Usage and error logs in multiple and customizable formats
- Virtual hosting for different IP addresses mapped to the same server
- DirectoryIndex directives to multiple files
- URL aliasing or rewriting with no fixed limit

According to the Netcraft Web site ([www.netcraft.com](http://www.netcraft.com)), at the recent time, Apache is running over 34 million Internet servers, more than Microsoft, Sun ONE, and Zeus combined.

Its flexibility, power, and, of course, price make it a popular choice. It can be used to host a Web site for the general public, or a company-wide intranet, or for simply testing your pages before they are uploaded to a secure server on another machine.

## **CHAPTER -4-**

### **ONLINE WEB BASED EDUCATIONAL SYSTEM DESIGN DESCRIPTION**

#### **4.1 PURPOSE**

The main purpose of this project is to create a dynamic server-side web page which contains educational documents and lessons related to specific topics. The system is provide useful documents and lectures about popular programming language concepts and oracle database system. The web page is presented to usage of everyone. Unregistered or registered users can access information through the system, but registred users have some other rights beside viewing only the content.

#### **4.2 SOFTWARE REQUIREMENTS ANALYSIS**

##### **4.2.1 PHP Server-side Scripting Language**

The fundamental structure of the system is based on PHP scripting language which allows us to create a server-side based, dynamic and interactive web pages. The most of the dynamic content is coded using PHP. Also the database creation and manipulation is done through the PHP statements. The security, authorization and privileges of the page are controlled by PHP statements.

Beside active contents, the system contains static HTML and XHTML. The dynamic contents are embedded to that static HTML structure.

##### **4.2.2 MySQL Database System**

One of the most important part of the project is database concept. Using a database management system is essential in dynamic server-side web based systems, because with the database, we can create more interactive web contents. The system user names, their personal information, password information, the content of the web page can easily stored and manipulated using the database systems. In this project, I used MySQL database system to store the necessary information and data related to the users and the system.



### **4.2.3 Apache Web Server**

In order to activate the system, it needs a web server. The web system runs on the server and through it, all the processes can be accessible. In this project, Apache web server is used because it is very powerful tool and fully supported by PHP and MySQL.

## **4.3 SYSTEM DESIGN OVERVIEW AND BASIC PRINCIPLES**

The core of the system contains PHP codes, HTML/XHTML elements and MySQL queries. To activate all the skeleton of the system, also we need a web server (in this project, the choice is Apache web server). Working principle of the system depends on all those concepts together.

The main design aim and principle of this system is very simple; to provide educational references, documents, lectures or other useful things to the users.

The users can enter the site and view the content without registration but some limitations occurs at that time. Registration is not necessary but anyone can easily register himself/herself and gain access to extra rights.

In the system we have users which are called "guests". These are unregistered users and they can access the web page without any privileges. The guests can only view the content. The web page contains download section, personal information area and so on. The guests are not permitted to download any documents. The unregistered

Another type of user is called "normal users" which are registered and logged in the system. They have extra privileges like accessing the download section, viewing personal area.

The last type of user in the system is called "admin". Administrators have superior rights and privileges. They can add or delete users, admins or upload lectures and documents, and so on.

We have two kind of admins; level 1 and level 2. The first level admin have full control of the system. They can add admins or delete admins. Upload rights is valid for both types of admins but, the content deletion can be done by only level 1 admins.

The system and its entities have much more attributes and functions. In the following section, I deeply cover all the concepts.

## 4.4 SYSTEM STRUCTURE AND WEB CONTENTS

























































### 4.4.1 Database Section

The database and all its tables are created using PHP embedded SQL queries. The database is named as “registration” and contains three separate tables; admin, user\_info and upload.

All queries and codes are described and shown in the Appendix A.

The following table shows the “user\_info” database table:

**Table 4.1** User information database table

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	email	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	username	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	password	varchar(255)	latin1_swedish_ci		No			      
<input type="checkbox"/>	first_name	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	last_name	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	city	varchar(50)	latin1_swedish_ci		Yes	NULL		      
<input type="checkbox"/>	country	varchar(50)	latin1_swedish_ci		Yes	NULL		      
<input type="checkbox"/>	id	int(10)			No		auto_increment	      

The system stores all the input information that users entered in the registration section in the user\_info table and when needed, the necessary manipulations like insertion, deletion or update is made in that table.


















































The id field specified as Primary Key constraint and it is set to auto increment.

The auto increment function is very useful in id assignments because, the database system automatically increment the incoming new id values started from zero. The database administrators don't need to give id numbers per users one by one or don't need to control the rest of the process.



That table shows the “admin” database table:




































**Table 4.2** Admin information table

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	username	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	password	varchar(255)	latin1_swedish_ci		No			      
<input type="checkbox"/>	first_name	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	last_name	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	email	varchar(50)	latin1_swedish_ci		No			      
<input type="checkbox"/>	admin_level	int(2)			No			      
<input type="checkbox"/>	<u>id</u>	int(10)			No		auto_increment	      

All the administrator’s account information and access levels are stored in the admin table.

The following table shows “upload” database table:

**Table 4.3** Upload database table

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>id</u>	int(11)			No		auto_increment	      
<input type="checkbox"/>	name	varchar(30)	latin1_swedish_ci		No			      
<input type="checkbox"/>	type	varchar(30)	latin1_swedish_ci		No			      
<input type="checkbox"/>	size	int(11)			No			      
<input type="checkbox"/>	content	mediumblob		BINARY	No			      

The registered users can download the uploaded content from the database. Only the administrators can upload content to the database.

The web page contains limited length of courses but the administrators upload new lectures to the download section and only the registered users can access them.

The database section is end here. All dynamic data is hold in those tables. The tables are separate from each other so it is a secure way to store important information. There is no references or foreign keys to connect the tables together.



After creation and setup the database, we need to initialize an admin account because the system needs a root administrator account to initialize and controls certain processes and users.

In order to create the root administrator account, I fill the “admin” table with the necessary administrator information. Once again, I used PHP embedded SQL queries to communicate and manipulate the database.

All queries and codes used in this process are described and shown in the Appendix A.

The following figure shows the initial admin account information:

←T→	username	password	first_name	last_name	email	admin_level	id
□ ✎ X	shadow	*23AE809CDACAF96AF0FD78ED04B6A265E05AA257	Ismail Vehbi	Ozsoy	ismailvehbiozsoy@gmail.com	1	1

**Figure 4.1.** The root admin information

The password field is encoded so, no one can see the real password. This feature is caused by a special PHP function. It is also described with all the other codes in Appendix A section.

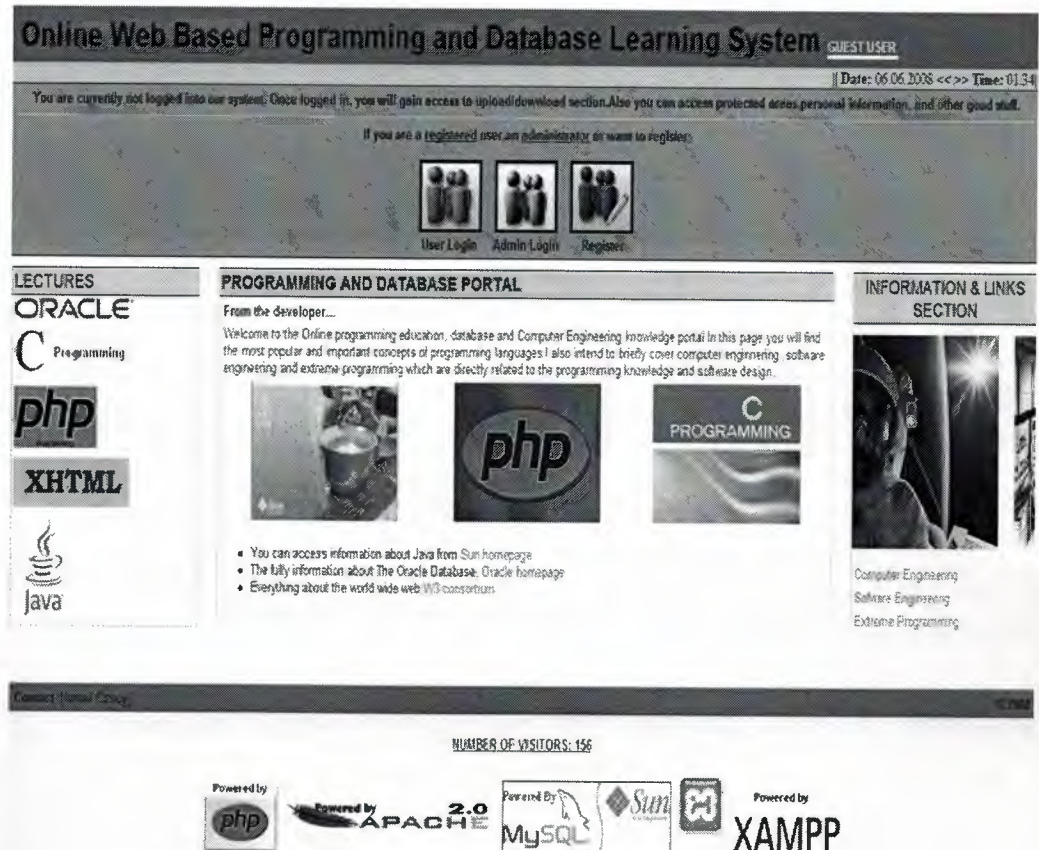
#### **4.4.2 Description and Working Principles of the Web Pages**

The system is designed to provide online lectures and documents to the users. The web interfaces of the system is created in order to easily accessible by the users.

All the PHP codes and SQL queries are added and described in Appendix B.

##### **4.4.2.1 Index Section**

The index section is designed to provide content and information to both type of users. It is coded such a way to become accessible by unregistered, registered and admin users.



**Figure 4.2.** Unlogged user index

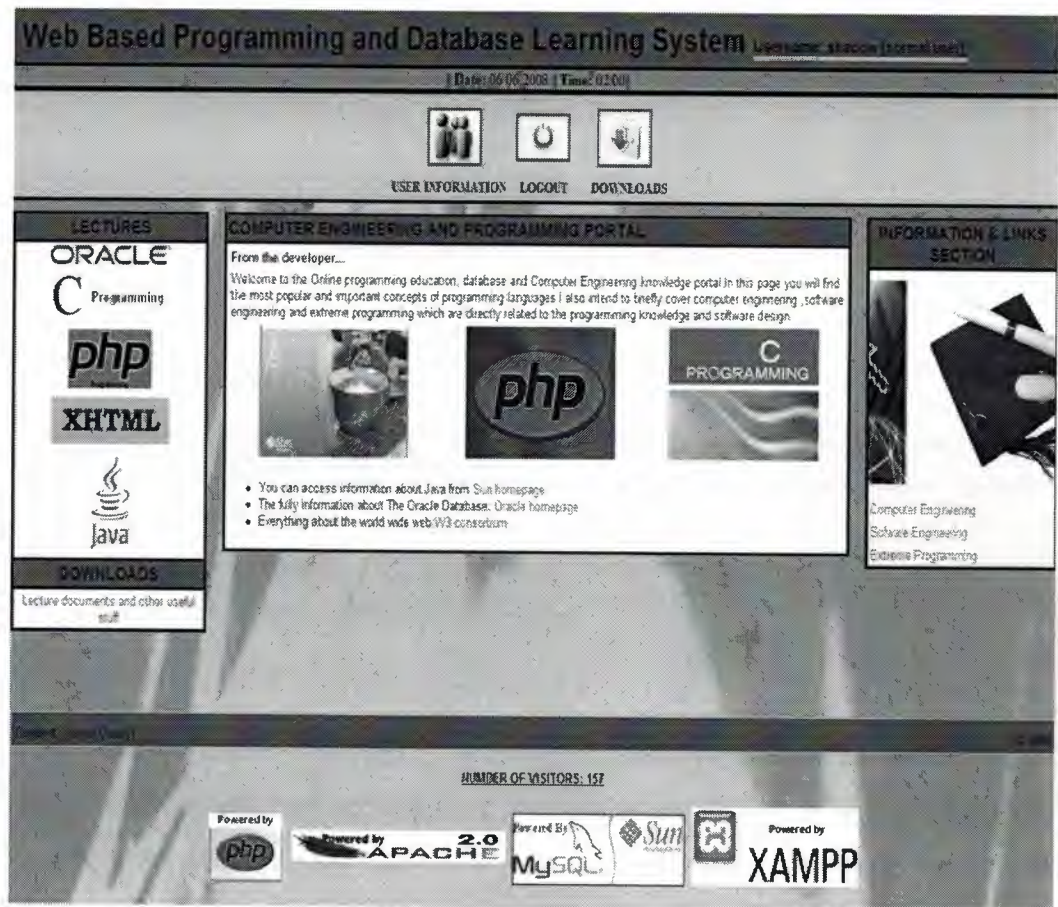
In the code, session variables are created to control whether a guest user, a registered user or an admin tries to enter the web page. The sessions must have null values or has input values, so if the user is logged in, the index includes logged user page, if not unlogged user page is showed to the user. The figure 4.2. shows the unregistered guest user index page.

The visitors can access lectures, information and links sections but cannot download any document from the database. In order to become a registered user, they have to register themselves from the register page.

If a registered user or an admin enters the index page, the other sections will reveal themselves. The download section, user information section or other sections will reveal in this situation.

The following figure shows the logged users index page:





**Figure 4.3.** Logged user index

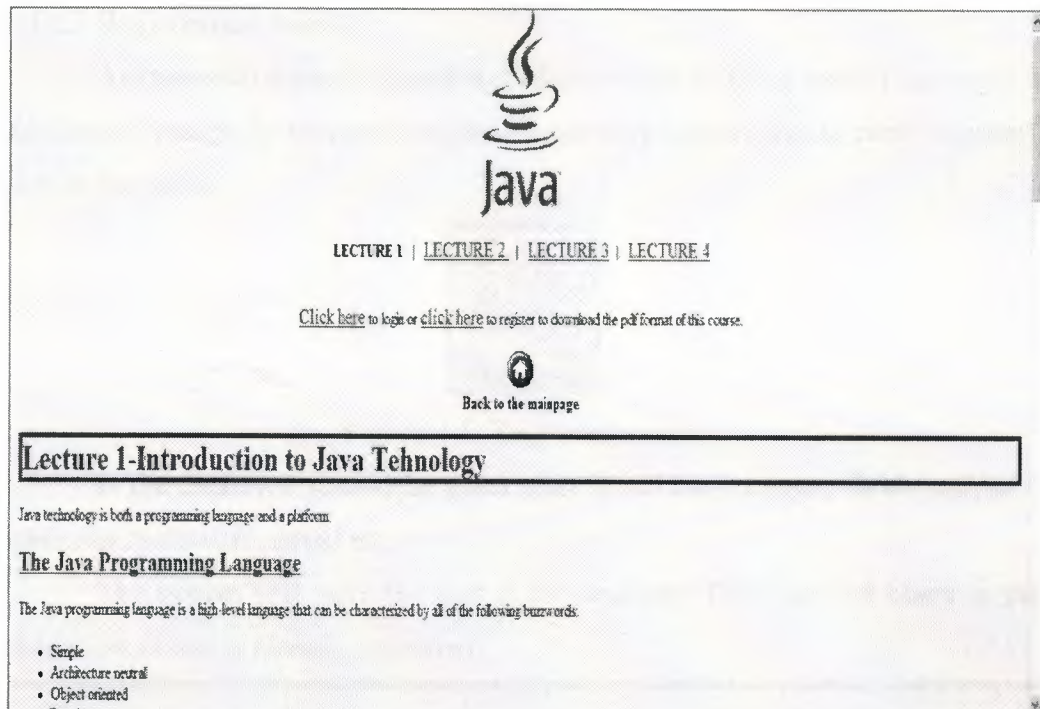
The user type recognition mechanism is very important because with it we can limit the users and their privileges to control the system. The username and its level which pass through the login section is automatically shown in the upper-right side of the page. If the user would be an admin, it would be indicated as its admin nickname and admin indicator by its side.

The index page also contains a visitor counter which shows the total number of entries to the page.

#### **4.4.2.2 Lectures Section**

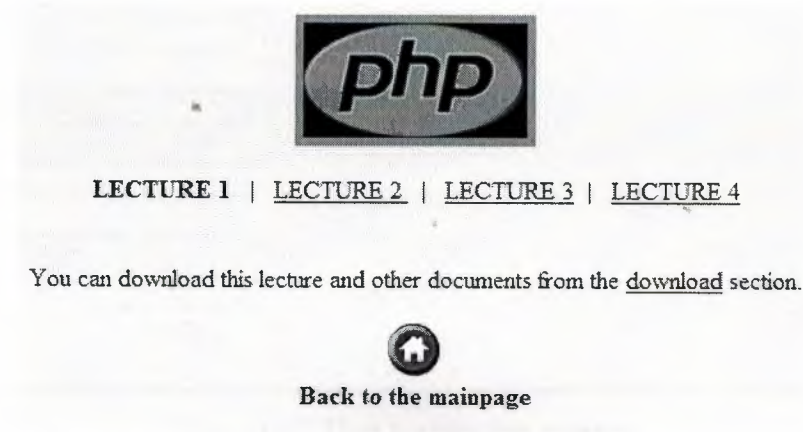
This section contains lectures about popular and important programming languages such as Java, PHP and C. Also mark up language XHTML and the Oracle database system lectures are provided to the users in the lecture section.





**Figure 4.4.** An example of an unlogged user's view of Java lectures

The unregistered and registered user's can access and follow the lectures. But the lectures are limited in the web page. The full version of all the lectures are uploaded to the database by the admins. In order to have them, the users must register to the system and then download the necessary lecture or document. If a registered user joins a lecture, the download section will reveal.



**Figure 4.5.** The download section reveals to the registered users

#### 4.4.2.3 Registration Section

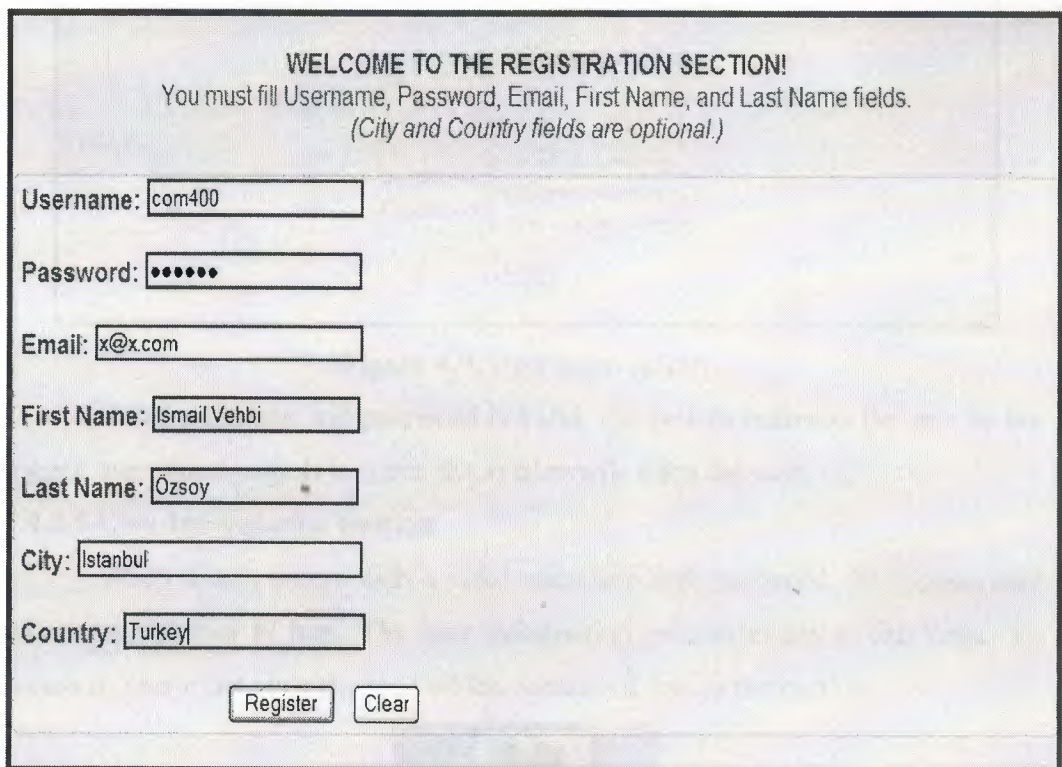
Anyone who wants to gain benefit from online learning system can register him/herself easily. To become a registered user, the visitor have to click “register” icon in the index.



**Figure 4.6.** Registration icon

In the displayed screen, the guest must fill all the necessary fields such as username, password, email etc.

The system will warn the user if the necessary fields are left blank or the chosen username is already registered.

A screenshot of a web registration form. At the top, it says "WELCOME TO THE REGISTRATION SECTION!" followed by "You must fill Username, Password, Email, First Name, and Last Name fields. (City and Country fields are optional.)". Below this are several input fields: "Username:" with the value "com400", "Password:" with masked characters "•••••", "Email:" with the value "x@x.com", "First Name:" with the value "Ismail Vehbi", "Last Name:" with the value "Özsoy", "City:" with the value "Istanbul", and "Country:" with the value "Turkey". At the bottom right of the form are two buttons: "Register" and "Clear".

**Figure 4.7** User registration example

After then, if the process is successful, the system redirects the user to the main index page.



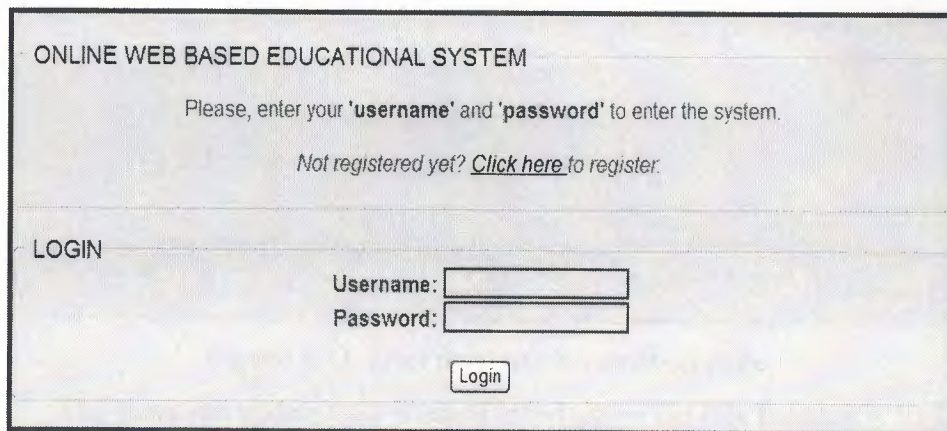
#### 4.4.2.4 User Login Section

All the registered users except administrators use this login section to enter the system with their valid usernames and passwords.



**Figure 4.8.** User login icon

In the index page, a user can click to the user login icon and enters the valid information necessary to enter the page. If they are not registered yet, the user can use the register link to register themselves to the system.



**Figure 4.9.** User login screen

If the username and password is valid, the system redirects the user to the logged user index page. If it is not, the system will warn the user.

#### 4.4.2.5 User Information Section

When a user enters with a valid username and password, the logged user index page shown to him. The user information section exists in this page. To access it, user must click the icon which contains a link to the section.



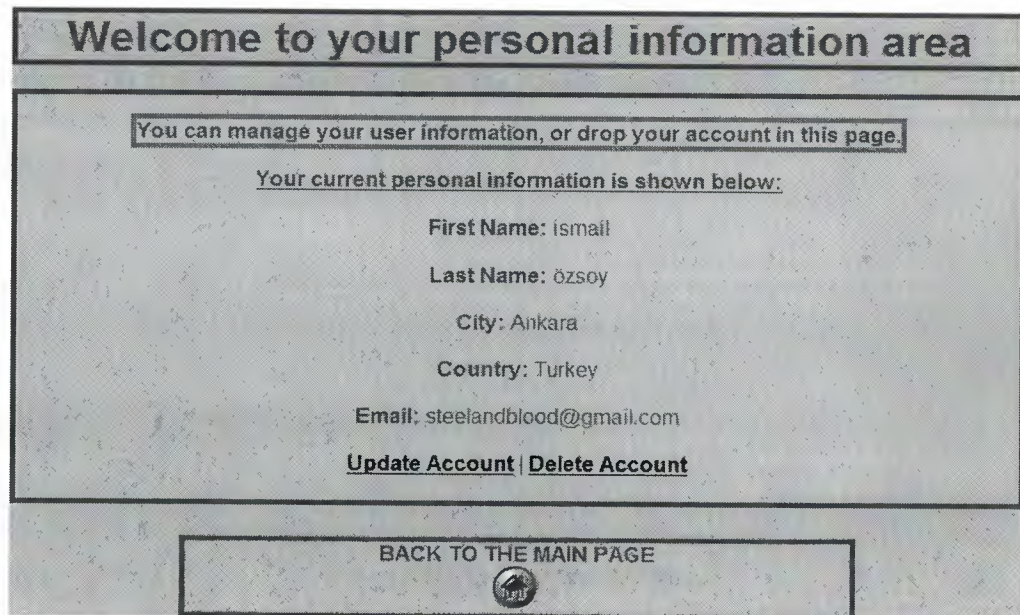
**Figure 4.10.** User information icon



All the information entered by the user in the registration area can be examined in this section.

The users can view their personal informations in this section. Also they can update or delete their account information.

The following figure shows a user personal information:



**Welcome to your personal information area**

You can manage your user information, or drop your account in this page.

Your current personal information is shown below:

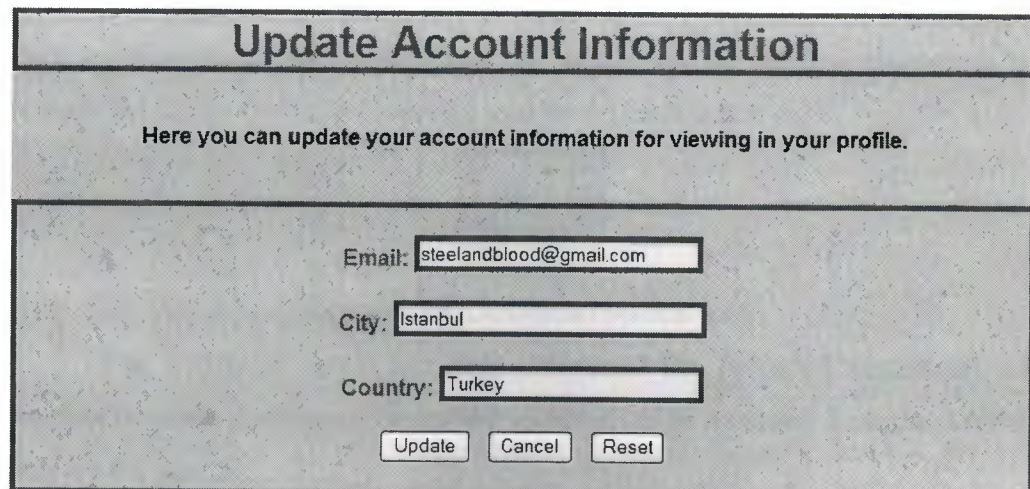
First Name: ismail  
Last Name: özsoy  
City: Ankara  
Country: Turkey  
Email: steelandblood@gmail.com

[Update Account](#) [Delete Account](#)

[BACK TO THE MAIN PAGE](#)

**Figure 4.11.** User personal information page

The users can update their account information but this function is limited. They cannot change every information which they previously entered to the system.



**Update Account Information**

Here you can update your account information for viewing in your profile.

Email:

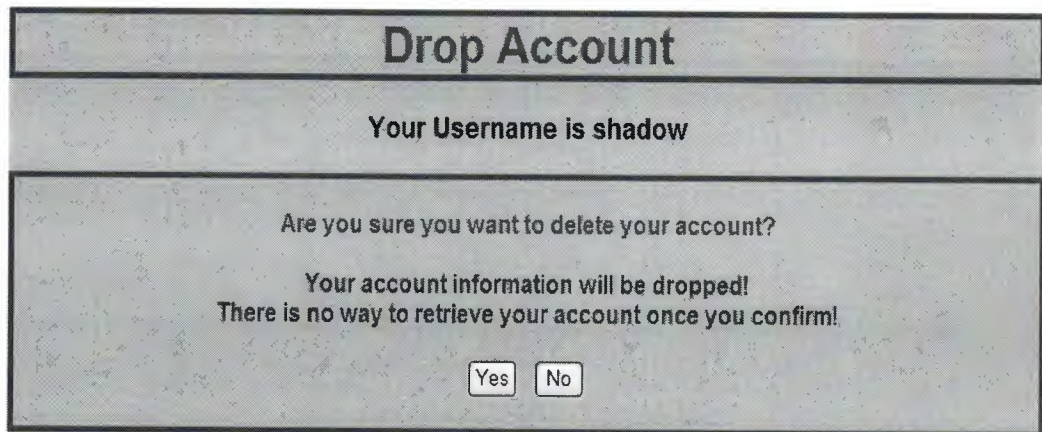
City:

Country:

**Figure 4.12.** Update user information

Only the email, city and country information can be changed by the users. The limitation exists because, the integrity and consistency of the system must be under control and they are very important. Every user should not change their usernames, names or other critical information very often. All the user personal information can only be changed by the administrators.

Another thing that the registered users can do is drop their accounts. In order to do that the user have to click the “delete account” link.



**Figure 4.13.** Account deletion

The system will warn the user whether to delete or cancel the process. If the user clicks to “yes”, the account information is deleted from the database. If the users selects “no”, the system takes the user to the previous page.

#### **4.4.2.6 Download Section**

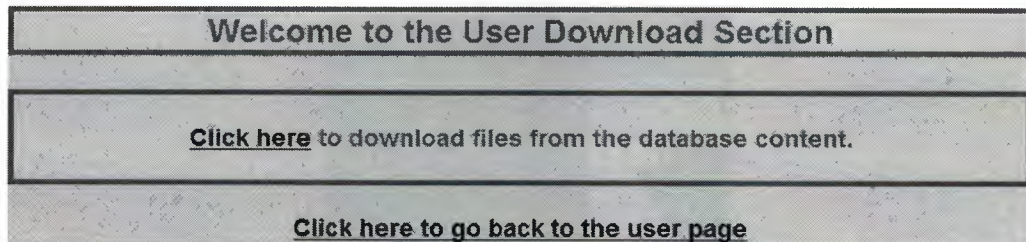
All the registered users have the right to access to the download section. This section can be accessible from the lecture pages, the link located in the bottom-left bar or from the download icon in the main index page.



**Figure 4.14.** Download section icon

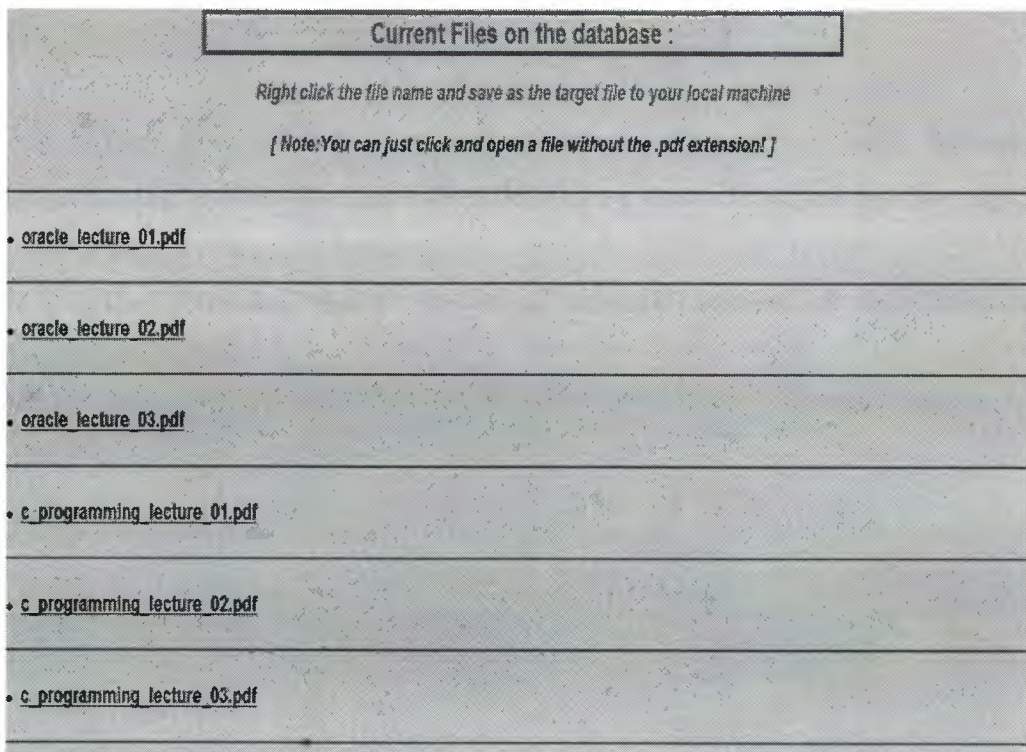
The registered users can only download the uploaded documents and lectures from the database. The content is added to or removed from the database only by the admins.





**Figure 4.15.** Download section

To access the content stored in the database, user have to click the link. The files currently exist in the databes will be listed then.



**Figure 4.16.** Downloadable lectures and documents by the users

The lectures are in pdf format, so to view them users have to download them and open with “Adobe Reader” program.

#### 4.4.2.7 Information and Links Section

This section provides brief but useful information and important links related to Computer Engineering, Software Engineering and Extreme Programming topics.

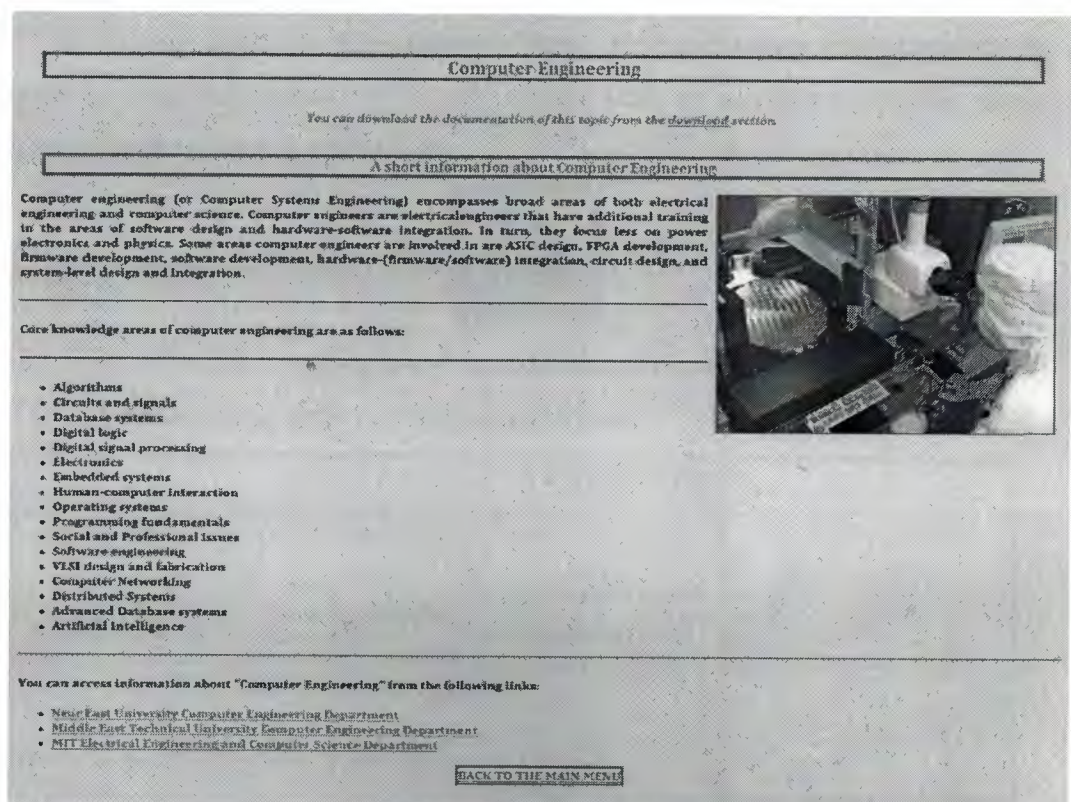




**Figure 4.17.** Information and links section

The Computer Engineering, Software Engineering and Extreme programming sections contain a short description about the topics and have some links to the university departments.

The following figure shows an example content of the Computer Engineering section:



**Figure 4.18.** An example of Computer Engineering information page

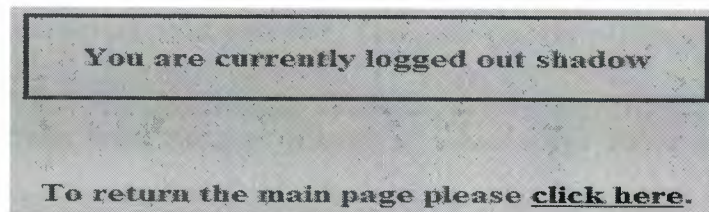
#### 4.4.2.8 User Logout

If the user wants to end its current active usage of the system under that username, by clicking the log out button, the session will over instantly.



**Figure 4.19.** Logout icon

After clicking the “logout”, the system displays a notification about the process.



**Figure 4.20.** Logout notification

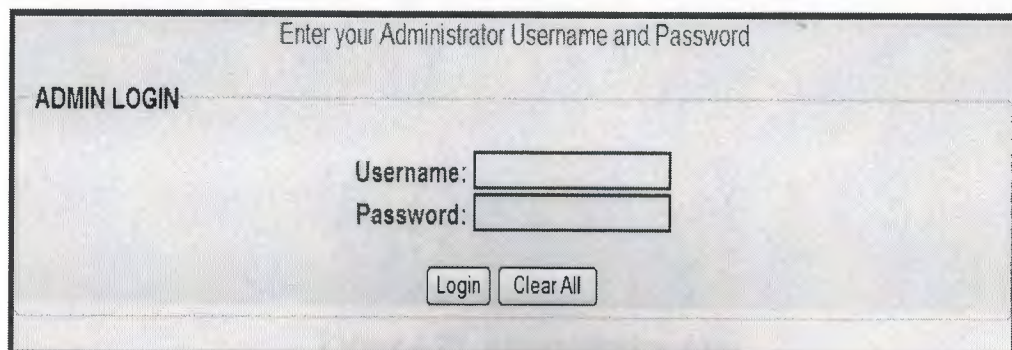
#### 4.4.2.9 Admin Login Section

In this page, the administrators enter their usernames and passwords in order to enter the system. To join the system with an admin account, the admin have to click to the link in the index.



▪ **Figure 4.21.** Admin login icon

The following page is displayed after then:



**Figure 4.22.** Admin login page



There is no separate admin registration section in the system. It means that a normal user can't register him/herself as an admin. The admin account creation can be done by only the initial admin. After that point the new created admin accounts can be used to manage the system.

There is a root administrator account which I initialized before. We can enter the root admin username and password to trigger the administration controls and areas.

The initial admin username is "shadow" and the password is "123". Entering these values will redirect us to the admin index page.

Our root admin level is set to 1. This means all the privileges and rights are assigned to that account.

After this point the root admin can create other admin accounts. New created admins can also create new admin accounts but their admin level is set to 1 in order to do that.



**Figure 4.23.** Administration Area



In the main admin page, username and level of the admin is indicated. There are three main links. One is to enter the system management area, another is entering the main index, and the last one is logging out.

#### 4.4.2.10 Administration Section

In this section, the administration functions exist. A level 1 admin can view, update or delete all the users currently registered. There are also add user/admin and upload sections. However, level 2 admins can view and update the current users, but they cannot delete any user from the database.

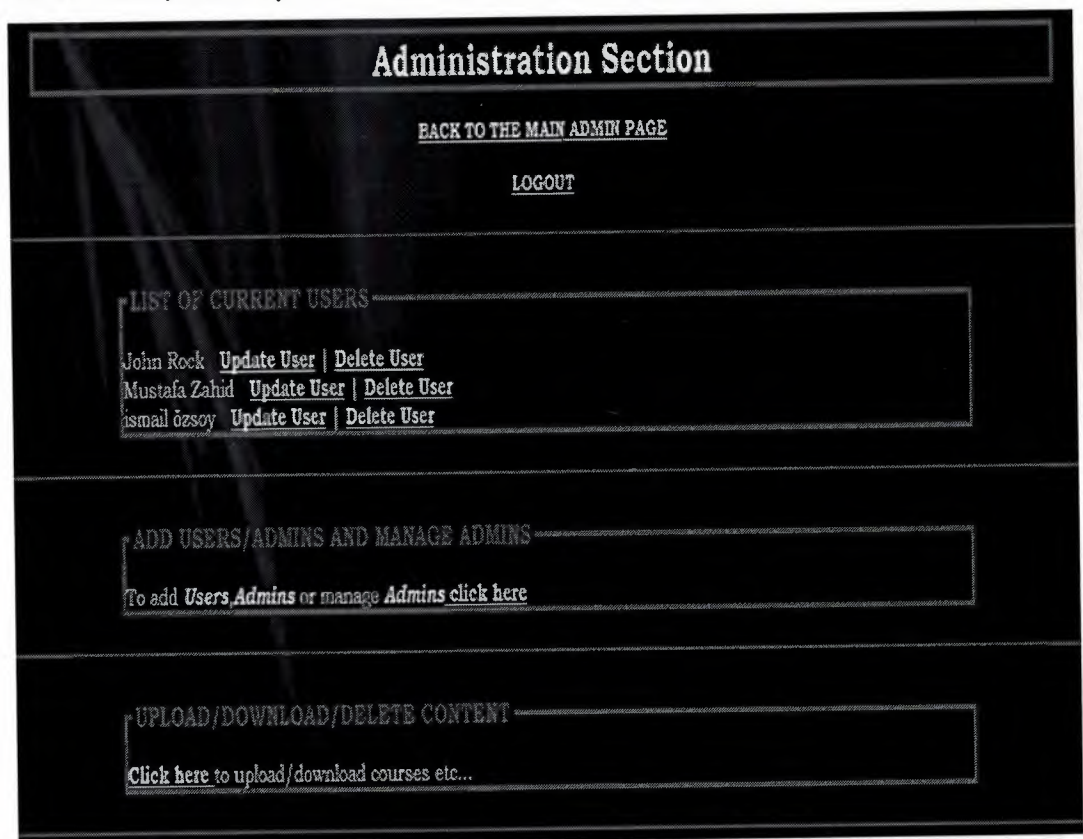


Figure 4.24. Administrator privileges and rights

#### 4.4.2.11 Update User Information

All the administrators can update a user information. To do that, they have to click the "update" link next to the user, then change the necessary sections. All the user information can be updated by the admins.

The screenshot shows a web form titled "Update User Information". At the top, a message states "User information has been updated." Below this, there are input fields for "Username" (containing "RedSky"), "Password" (with a "[Private]" label), "First Name" (containing "John"), "Last Name" (containing "Rock"), "Email" (containing "sky@red.com"), "City" (containing "New York"), and "Country" (containing "U.S.A"). An "Update" button is located below the "Country" field. At the bottom of the form, there is a link that says "Click here to return to the admin area".

**Figure 4.25.** User information update

#### 4.4.2.12 Delete User Information

All the level 1 admins have the account deletion rights. To access this right, the admin have to click “delete” next to the user.

The screenshot shows a confirmation dialog titled "Delete Account". The main text asks, "Are you sure you want to delete this user's account?" followed by a warning: "If you delete this account, there is no way to retrieve it back once you confirm!". At the bottom, there are two buttons: "Yes" and "No". Below the buttons, there is a link that says "Click here to return to the admin area".

**Figure 4.26.** Account deletion

#### 4.4.2.13 Add User/Admins and Manage Admins

This is the another important part of the system. Creating new users and admins and manage the administrators is essential in dynamic web systems.

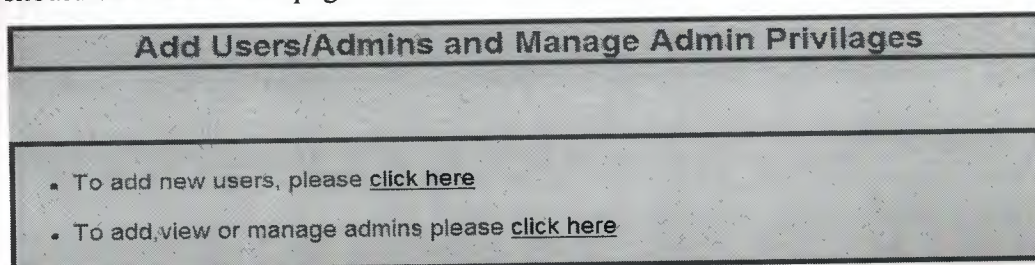
To access these functions, the admin have to click the “add users/admins or manage admins” link.





**Figure 4.27.** User and admin create/manage link

After then, the following screen is shown up. The necessary selection should be made in this page.



**Figure 4.28.** Selection screen

If the admin selects the "add new users", the new account creation section will appear. The admin then should fill the necessary sections of the form to register a user.

A screenshot of a web interface for user registration. At the top, there is a dark header bar with the text "ADD NEW USER TO THE SYSTEM" in a light, sans-serif font. Below the header, there is a line of text: "Username, Password, Email, First Name, and Last Name fields are required to register the user successfully!". Below this text, there is a form with several input fields: "Username:", "Password:", "Email:", "First Name:", "Last Name:", "City:", and "Country:". Each label is followed by a text input field. At the bottom of the form, there are two buttons: "Insert" and "Clear".

**Figure 4.29.** User registration by an admin



The system will display a warning message if the necessary fields are left blank or the username is already taken.

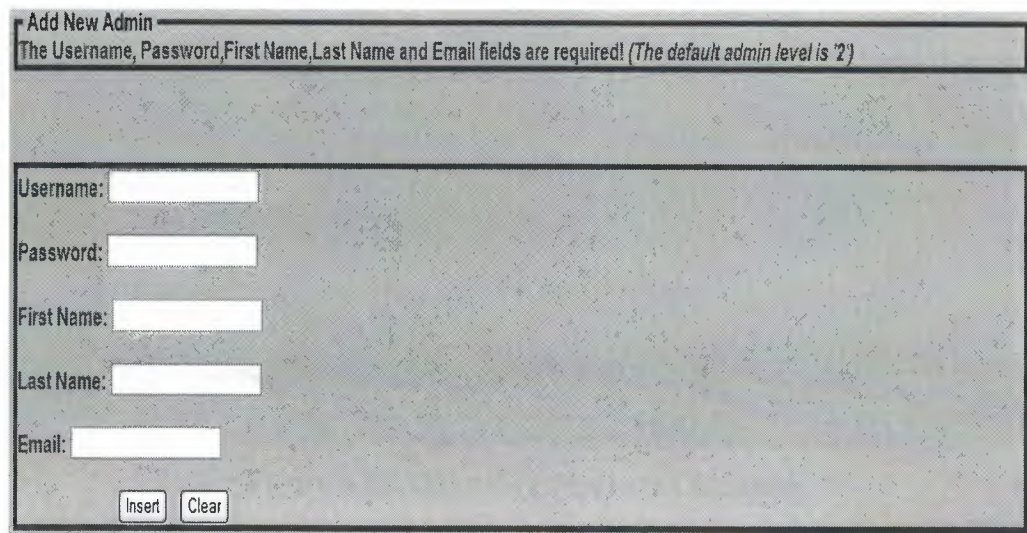
If the admin selects the “add, view or manage admins”, all the administrators are listed in the next page and “add new admin” section appears.

LIST OF CURRENT ADMINISTRATORS	
<ul style="list-style-type: none"><li>• Username: shadow Fullname: Ismail Vehbi Ozsoy Admin Level: 1</li></ul>	<a href="#">Update Admin</a> <a href="#">Delete Admin</a>
<ul style="list-style-type: none"><li>• Username: admin Fullname: Administrator Administrator Admin Level: 2</li></ul>	<a href="#">Update Admin</a> <a href="#">Delete Admin</a>

**Figure 4.30.** List of current admins

The “update admin” and “delete admin” functions are displayed for only level 1 administrators. Level 2 administrators can’t change or delete any admin account.

Creating new admin account can be made by both level of admins. The new created admin level is set to 2 by default. It can be changed by level 1 admins in the update admin section.



Add New Admin

The Username, Password, First Name, Last Name and Email fields are required! (The default admin level is '2')

Username:

Password:

First Name:

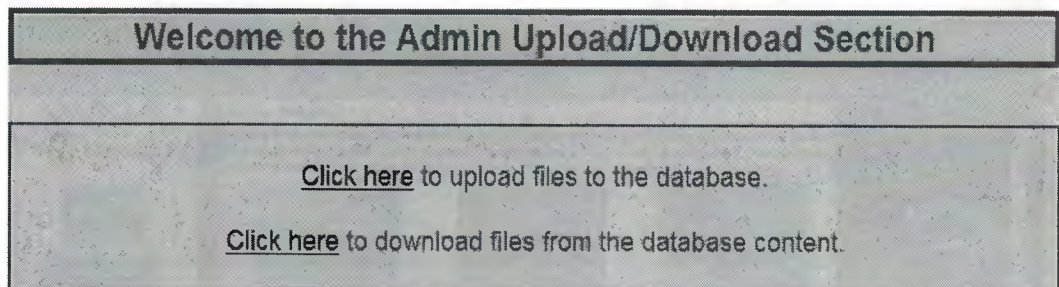
Last Name:

Email:

**Figure 4.31.** New administrator creation

#### 4.4.2.14 Upload / Download Section

Another most important section in the system is upload/download section. To provide lectures and documents the administrators are upload content to the database.



Welcome to the Admin Upload/Download Section

[Click here](#) to upload files to the database.

[Click here](#) to download files from the database content.

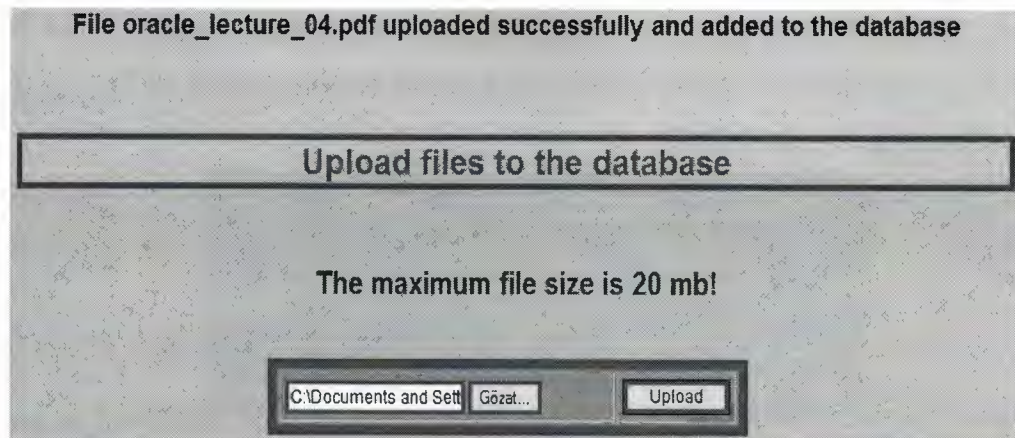
**Figure 4.32.** Upload/download section

The new lectures and documents are stored in the database. Only the registered users can download these content.

The file size is limited to 20 Mb, so the administrators have to careful about the file sizes of the documents. If the size limitation is exceeded, there won't be an error message, just nothing will be uploaded to the database.

If the file is successfully uploaded to the database, a notification about the process is displayed.





**Figure 4.33.** Uploading files to the database

#### 4.4.2.15 Admin Index Section

The index page is accessible by the admins also. All the content can be accessed by the administrators.

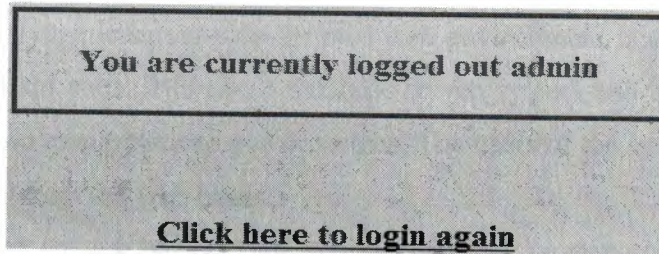


**Figure 4.34.** Admin index



#### 4.4.2.16 Admin Logout

If the admin wants to end its current active usage of the system under that username, by clicking the logout link, the session of that admin will over instantly.



**Figure 4.35.** Admin logout



## **CONCLUSION**

Developing an online web-based educational system and presenting the contents to the users efficiently are the main purposes of this project. The interactive and dynamic server-side-scripted web environment is created to fulfill and serve the main aims. It is also a database driven system and it is designed to handle the certain manipulations and processes. The users of the system can access all the functions from the web pages.

Providing enough background to understand the working principles of the system are covered in details. Software requirements and core system entities are presented in the project.

The system is designed to support the basic content management fundamentals. The flexible and extendable structure allows us to add or integrate new features or functions to the previous design.

## REFERENCES

- [1] Converse T. And Park J., *PHP5 and MySQL Bible*, Wiley Publishing Inc., Indianapolis, United States of America, 2004.
- [2] Vaswani V., *How to Do Everything with PHP & MySQL*, McGraw-Hill Publishing Co., New York, United States of America, 2005.
- [3] Castro E., *HTML, XHTML, & CSS, Sixth Edition: Visual QuickStart Guide*, Peachpit Press, CA, United States Of America, 2007.
- [4] Java Tutorial, August 1, 2007 from the World Wide Web  
“<http://java.sun.com/docs/books/tutorial>”.
- [5] XHTML Tutorial, June 1, 2008 from the World Wide Web  
“<http://www.w3schools.com/xhtml/default.asp>”.
- [6] Extereme Programming article, June 5, 2008 from the World Wide Web  
“[http://en.wikipedia.org/wiki/Extreme\\_Programming](http://en.wikipedia.org/wiki/Extreme_Programming)”.
- [7] Computer Engineering article, May 28, 2008 from the World Wide Web  
“[http://en.wikipedia.org/wiki/Computer\\_engineering](http://en.wikipedia.org/wiki/Computer_engineering)”.
- [8] Software Engineering article, May 26, 2008 from the World Wide Web  
“[http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering)”.



## APPENDIX A

### MySQL Database and its tables creation codes and queries

#### database create.php

```
<?php
//Connecting to MySQL
$connect_db = mysql_connect("localhost", "root", "com400") or
die ("Connection error");
//Creating the main database if it doesn't already exist
$create_db = mysql_query("CREATE DATABASE IF NOT EXISTS
registration")
or die(mysql_error());
//Selecting the new created database
mysql_select_db("registration");
//Creationg the user_info table which will hold the registered user information
$user_table_creation = "CREATE TABLE user_info (
email varchar(50) NOT NULL,
username varchar(50) NOT NULL,
password varchar(255) NOT NULL,
first_name varchar(50) NOT NULL,
last_name varchar(50) NOT NULL,
city varchar(50) default NULL,
country varchar(50) default NULL,
id int(10) NOT NULL auto_increment,
PRIMARY KEY (id)
)";
$results = mysql_query($user_table_creation)
or die (mysql_error());
echo "User information table successfully created!";
//Creating the admin table which will hold the admin's information
$admin_table_creation = "CREATE TABLE admin (
```

```

username varchar(50) NOT NULL,
password varchar(255) NOT NULL,
first_name varchar(50) NOT NULL,
last_name varchar(50) NOT NULL,
email varchar(50) NOT NULL,
admin_level int(2) NOT NULL,
id int(10) NOT NULL auto_increment,
PRIMARY KEY (id)
);";

$results = mysql_query($admin_table_creation)
or die(mysql_error());

echo "Admin information table successfully created!";

//Creating the upload table
$upload_table_creation="CREATE TABLE upload(
id INT NOT NULL AUTO_INCREMENT,
name VARCHAR(30) NOT NULL,
type VARCHAR(30) NOT NULL,
size INT NOT NULL,
content MEDIUMBLOB NOT NULL,
PRIMARY KEY(id));";

$result=mysql_query($upload_table_creation)
or die (mysql_error());

echo " Upload table successfully created! ";
echo "Registration database successfully created!";

```

## APPENDIX B

### User's PHP codes and MySQL Queries

#### auth\_user.inc.php

```
<?php
session_start(); // Starting a session
if ((isset($_SESSION['user_logged']) && // Controlling whether the user logged
or not
$_SESSION['user_logged'] != "") ||
(isset($_SESSION['user_password']) &&
$_SESSION['user_password'] != "")) {
// Do Nothing
} else {
$redirect = $_SERVER['PHP_SELF'];
header("Refresh: 5; URL=user_login.php?redirect=$redirect");
echo "You are currently not logged in, we are redirecting you, " .
"be patient!<br>";
echo "(If your browser doesn't support this, " .
"<a href=\"user_login.php?redirect=$redirect\">click here</a>);";
die();
}
?>
```

#### conn.inc.php

```
<?php
$conn = mysql_connect("localhost", "root", "com400") // Connecting to the
or die(mysql_error());                               //MySQL Database
$db = mysql_select_db("registration")
or die(mysql_error());
?>
```



### **Index.php**

```
<?php
// If the user registered and logged in include the logged user page and show,
//otherwise show the unlogged user page
session_start();
if ((isset($_SESSION['user_logged'])&&
    $_SESSION['user_logged'] != "" ) ||
    (isset($_SESSION['user_password']) &&
    $_SESSION['user_password'] != "")) {
    include "logged_user.php";
} else {
    include "unlogged_user.php";
}
?>
```

### **logged\_user.php**

```
<?php // Included page to the index for logged users
session_start();
if ((isset($_SESSION['user_logged']) && // Display for only users
    $_SESSION['user_logged'] != "" ) ||
    (isset($_SESSION['user_password']) &&
    $_SESSION['user_password'] != "")) { ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Web Based Programming and Database Learning System-Logged
User</title>
<style type="text/css">
```

```

body { background-image: url(backgrounds/background_main_02.jpg);
background-repeat:no-repeat; background-position:center; background-
attachment:fixed;margin: 0; padding: 0; font: normal 82% sans-serif; }
h1 {margin: 0; margin-bottom: 0px; padding: 8px; color: black; text-align:left;
background-color:#0066FF;}
.tagline { font-size: 50%; color: #000033; border-bottom: solid ;border-bottom-
color: #00FFFF; }
#display_date { font-size:16px; font-family:"Times New Roman", Times,
serif;background-color:#46A3FF; border-top:solid; border-top-color:#000000;
border-bottom:solid; border-bottom-color:#000000; color:#000000; text-
align:center; }
#b {margin-top: 0px; margin-bottom: 10px; padding: 5px; border-top: solid 1px
#000000; border-bottom: solid 3px #000000; background-color: #00FFFF;text-
align:center;}
#b a {margin-left: 2px; margin-right: 2px; color: #000000; text-decoration: none;}
#b a:hover {text-decoration: underline;}
#image_down { font-size:14px;font-style:normal; font-family:"Times New
Roman", Times, serif; text-align:center;}
#left {float: left; width: 18%; border: solid 3px #000000;margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}
#left ul {list-style: none; margin:0; padding: 0; text-align:center;}
#left li {margin: 0px; padding: 0px;}
#left a {display: block ; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#left a:hover {border: solid 1px #0066FF; background-color:#0066FF; color:
#ffffff;}
#left h3 {margin: 0; padding: 2px; font-weight: normal; background-color:
#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000; text-
align:center;}
#right{float: right; width: 18%; border: solid 3px #000000; margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}

```

```

#right ul {list-style: none; margin: 0; padding: 0;}
#right li { list-style:margin: 0px; padding: 0px;}
#right a {display: block; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#right a:hover {border: solid 1px #0066cc; background-color: #0066cc; color:
#ffffff;}
#right h3 {margin: 0; padding: 2px; font-weight: normal; background-
color:#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000;
text-align:center;}
#content {margin-left: 20%; margin-right:20%; margin-bottom:5%; padding: 0;
border-left: solid 3px #000000; border-right:solid 3px #000000; border-top: 0;
border-bottom: solid 3px #000000; background-color: #ffffff;}
#content h3 {margin: 0; padding: 2px; font-weight: normal; background-color:
#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000;}
#content p {margin: 5px; color: #000000;}
#content a {color: #0066cc; text-decoration: none;}
#content a:hover {color: #0000ff; text-decoration: underline;}
#content .point {margin: 3%; padding: 0px; border: solid 1px #000000; border-
top: 0; background-color: #f9f9f9;}
#content .point h3 {}
#content .code {margin: 3%; padding: 7px; border: solid 1px #cccccc;
background-color: #eeeeee; font: normal 100% courier; color: #000000;}
#footer {clear: both; margin-top: 0px; padding: 5px; border: solid 1px #cccccc;
background-color: #0066FF;}
#footer p {float: left; margin: 0;}
#footer h5 {margin: 0; text-align: right; color: #000000;}
#footer a {margin-left: 2px; margin-right: 2px; color: #000000; text-decoration:
none;}
#footer a:hover {text-decoration: underline;}
</style>

```





```

<li><a href="http://localhost/graduationproject/lectures/oracle.php"></a></li>
<li><a href="http://localhost/graduationproject/lectures/c.php"></a></li>
<li><a href="http://localhost/graduationproject/lectures/php_1.php"></a></li>
<li><a href="http://localhost/graduationproject/lectures/xh.php"></a></li>
<li><a href="http://localhost/graduationproject/lectures/java.php"></a></li>
</ul>
<h3><strong>DOWNLOADS</strong></h3>
<ul>
<li><a href="http://localhost/graduationproject/upload/updown.php">Lecture
documents and other useful stuff</a></li>
</ul>
</div>
<div id="right">
<h3><strong>INFORMATION & LINKS SECTION</strong></h3>
<marquee direction="left" onmouseover="stop()" onmouseout="start()"
title="WORLD OF ENGINEERING" behavior="alternate" >
</marquee>
```

```
<ul>
```

```
<li><a href="lectures/com_eng.php">Computer Engineering</a></li>
```

```
<li><a href="lectures/soft_eng.php">Software Engineering</a></li>
```

```
<li><a href="lectures/xp.php">Extreme Programming</a></li>
```

```
</ul>
```

```
</div>
```

```
<div id="content">
```

```
<h3><strong>PROGRAMMING AND DATABASE PORTAL</strong></h3>
```

```
<p><b>From the developer...</b>
```

```
<p>Welcome to the Online programming education, database and Computer  
Engineering knowledge portal.In this page you
```

```
will find the most popular and important concepts of programming languages.I  
also intend to briefly cover computer engineering
```

```
,software engineering and extreme programming which are
```

```
directly related to the programming knowledge and software design.
```

```
</p></p>
```

```

```

```
<ul>
```

```
<li>You can access information about Java from <a
```

```
href="http://www.sun.com">Sun homepage</a></li>
```

```
<li>The fully information about The Oracle Database: <a
```

```
href="http://www.oracle.com/index.html">Oracle homepage</a></li>
```

```
<li>Everything about the world wide web:<a href="http://www.w3.org/">W3  
consortium</a></li>
```

```
</ul>
```

```
</div>
```

```
<div id="footer">
```



```

<p><b>Contact :</b>&#091;<a href=mailto:steelandblood@gmail.com>Ismail
Özsoy</a>&#093;</p>
<h5>&#169 2008 </h5>
</div>
<center><p><b><u><?php include ("counter.php");?></u></b></p></center>
<div id="powered" align="center"><a href="http://www.php.org" ></a>&nbsp;
<a href="http://www.apache.com"></a>&nbsp;
<a href="http://www.mysql.com"></a>&nbsp;
<a href="http://www.xampp.com"></a></div>
</p>
</div>
</body>
</html>
<?php } else {?>
<?php
    if ((isset($_SESSION['admin_logged'])&&
        $_SESSION['admin_logged']) != "" ||
        (isset($_SESSION['admin_password']) &&
            $_SESSION['admin_password'] != "")) { // Display for only Admins
        ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

```

```

<title>Online Web Based Programming and Database Learning System-Logged
Admin</title>
<style type="text/css">
body { background-image: url(backgrounds/background_main_01.jpg);
background-repeat:no-repeat; background-position:center; background-
attachment:fixed;margin: 0; padding: 0; font: normal 82% sans-serif; }
h1 {margin: 0; margin-bottom: 0px; padding: 8px; color: black; text-align:left;
background-color:#0066FF;}
.tagline { font-size: 50%; color: #000033; border-bottom: solid ;border-bottom-
color: #00FFFF; }
#display_date { font-size:16px; font-family:"Times New Roman", Times,
serif;background-color:#46A3FF; border-top:solid; border-top-color:#000000;
border-bottom:solid; border-bottom-color:#000000; color:#000000; text-
align:center; }
#b {margin-top: 0px; margin-bottom: 10px; padding: 5px; border-top: solid 1px
#000000; border-bottom: solid 3px #000000; background-color:#0D86FF;text-
align:center;}
#b a {margin-left: 2px; margin-right: 2px; color: #000000; text-decoration: none;}
#b a:hover {text-decoration: underline;}
#image_down { font-size:14px;font-style:normal; font-family:"Times New
Roman", Times, serif; text-align:center;}
#left {float: left; width: 18%; border: solid 3px #000000;margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}
#left ul {list-style: none; margin:0; padding: 0; text-align:center;}
#left li {margin: 0px; padding: 0px;}
#left a {display: block ; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#left a:hover {border: solid 1px #0066FF; background-color:#0066FF; color:
#ffffff;}

```

```

#left h3 {margin: 0; padding: 2px; font-weight: normal; background-color:
#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000; text-
align:center;}
#right{float: right; width: 18%; border: solid 3px #000000; margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}
#right ul {list-style: none; margin: 0; padding: 0;}
#right li { list-style:margin: 0px; padding: 0px;}
#right a {display: block; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#right a:hover {border: solid 1px #0066cc; background-color: #0066cc; color:
#ffffff;}
#right h3 {margin: 0; padding: 2px; font-weight: normal; background-
color:#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000;
text-align:center;}
#content {margin-left: 20%; margin-right:20%; margin-bottom:5%; padding: 0;
border-left: solid 3px #000000; border-right:solid 3px #000000; border-top: 0;
border-bottom: solid 3px #000000; background-color: #ffffff;}
#content h3 {margin: 0; padding: 2px; font-weight: normal; background-color:
#0066FF; border-top: solid 3px #000000; border-bottom: solid 3px #000000;}
#content p {margin: 5px; color: #000000;}
#content a {color: #0066cc; text-decoration: none;}
#content a:hover {color: #0000ff; text-decoration: underline;}
#content .point {margin: 3%; padding: 0px; border: solid 1px #000000; border-
top: 0; background-color: #f9f9f9;}
#content .point h3 {}
#content .code {margin: 3%; padding: 7px; border: solid 1px #cccccc;
background-color: #eeeeee; font: normal 100% courier; color: #000000;}
#footer {clear: both; margin-top: 0px; padding: 5px; border: solid 1px #cccccc;
background-color: #0066FF;}
#footer p {float: left; margin: 0;}
#footer h5 {margin: 0; text-align: right; color: #000000;}

```





</div>

<div id="left">

<h3><strong>LECTURES</strong></h3>

<ul>

<li><a href="http://localhost/graduationproject/lectures/oracle.php"></a></li>

<li><a href="http://localhost/graduationproject/lectures/c.php"></a></li>

<li><a href="http://localhost/graduationproject/lectures/php\_1.php"></a></li>

<li><a href="http://localhost/graduationproject/lectures/xh.php"></a></li>

<li><a href="http://localhost/graduationproject/lectures/java.php"></a></li>

</ul>

<h3><strong>DOWNLOADS</strong></h3>

<ul>

<li><a href="http://localhost/graduationproject/upload/updown.php">Lecture  
documents and other useful stuff</a></li>

</ul>

</div>

<div id="right">

<h3><strong>INFORMATION & LINKS SECTION</strong></h3>

<marquee direction="left" onmouseover="stop()" onmouseout="start()"  
title="WORLD OF ENGINEERING" behavior="alternate" >

</marquee>
<ul>
<li><a href="lectures/com_eng.php">Computer Engineering</a></li>
<li><a href="lectures/soft_eng.php">Software Engineering</a></li>
<li><a href="lectures/xp.php">Extreme Programming</a></li>
</ul>
</div>
<div id="content">
<h3><strong> PROGRAMMING PORTAL AND DATABASE
PORTAL</strong></h3>
<p><b>From the developer...</b>
<p>Welcome to the Online programming education, database and Computer
Engineering knowledge portal.In this page you
will find the most popular and important concepts of programming languages.I
also intend to briefly cover computer engineering
,software engineering and extreme programming which are
directly related to the programming knowledge and software design
.</p></p>

<ul>
<li>You can access information about Java from <a
href="http://www.sun.com">Sun homepage</a></li>
<li>The fully information about The Oracle Database: <a
href="http://www.oracle.com/index.html">Oracle homepage</a></li>
<li>Everything about the world wide web:<a href="http://www.w3.org/">W3
consortium</a></li>

```



```

</ul>
</div>
<div id="footer">
<p><b>Contact :</b>&#091;<a href=mailto:steelandblood@gmail.com>Ismail
Özsoy</a>&#093;</p>
<h5>&#169 2008 </h5>
</div>
<center><p><b><u><?php include ("counter.php");?></u></b></p></center>
<div id="powered" align="center"><a href="http://www.php.org" ></a>&nbsp;
<a href="http://www.apache.com"></a>&nbsp;
<a href="http://www.mysql.com"></a>&nbsp;
<a href="http://www.xampp.com"></a> </div>
</p>
</div>
</body>
</html>
<?php } else { ?>
<strong>You are not logged in.Please <a href="unlogged_user.php">Login</a>
first!<br />
Not registered? <a href="register.php">Click here</a></strong>
<?php } }?>
unlogged_user.php
<?php // Included file to the index for only unlogged users ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Online Web Based Programming and Database Learning System-Welcome
Guest!</title>
<style type="text/css">
body {margin: 0; padding: 0; font: normal 82% sans-serif;}
h1 {margin: 0; margin-bottom: 0px; padding: 8px; color: black; background-
color:#666666;}
.tagline {font-size: 40%; color:#FFFFFF; border-bottom:solid; border-bottom-
color:#FFFFFF; }
#display_date { font-size:16px; font-family:"Times New Roman", Times,
serif;background-color:#CCCCCC; color:#000000; text-align:right; }
#b {margin-top: 0px; margin-bottom: 10px; padding: 5px; border-top: solid 1px
#000000; border-bottom: solid 1px #000000; background-color:#999999; text-
align:center;}
#b a {margin-left: 2px; margin-right: 2px; color: #000000; text-decoration: none;}
#b a:hover {text-decoration: underline;}
#left {float: left; width: 18%; border: solid 1px #cccccc; margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}
#left ul {list-style: none; margin:0; padding: 0;}
#left li {margin: 0px; padding: 0px; }
#left a {display: block; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#left a:hover {border: solid 1px #0066cc; background-color: #0066cc; color:
#ffffff;}
#left h3 {margin: 0; padding: 2px; font-weight: normal; background-
color:#CCCCCC; border-top: solid 1px #000000; border-bottom: solid 1px
#000000;}
#right {float: right; width: 18%; border: solid 1px #cccccc; margin-bottom: 15px;
background-color: #ffffff; border-top: 0;}
#right ul {list-style: none; margin: 0; padding: 0;}

```

```

#right li { list-style:margin: 0px; padding: 0px;}
#right a {display: block; width: 95.5%; margin: 0px; padding: 2px; border: solid
1px #ffffff; color: #0066cc; text-decoration: none;}
#right a:hover {border: solid 1px #0066cc; background-color: #0066cc; color:
#ffffff;}
#right h3 {margin: 0; padding: 2px; font-weight: normal; background-
color:#CCCCCC; border-top: solid 1px #000000; border-bottom: solid 1px
#000000; text-align:center;}
#content {margin-left: 20%; margin-right:20%; margin-bottom:5%; padding: 0;
border-left: solid 1px #cccccc; border-right:solid 1px #cccccc; border-top:0;
border-bottom: solid 1px #cccccc; background-color: #ffffff;}
#content h3 {margin: 0; padding: 2px; font-weight: normal; background-color:
#CCCCCC; border-top: solid 1px #000000; border-bottom: solid 1px #000000;}
#content p {margin: 5px; color: #000000;}
#content a {color: #0066cc; text-decoration: none;}
#content a:hover {color: #0000ff; text-decoration: underline;}
#content .point {margin: 3%; padding: 0px; border: solid 1px #000000; border-
top: 0; background-color: #f9f9f9;}
#content .point h3 {}
#content .code {margin: 3%; padding: 7px; border: solid 1px #cccccc;
background-color: #eeeeee; font: normal 100% courier; color: #000000;}
#footer {clear: both; margin-top: 0px; padding: 5px; border: solid 1px #cccccc;
background-color:#666666;}
#footer p {float: left; margin: 0;}
#footer h5 {margin: 0; text-align: right; color: #000000;}
#footer a {margin-left: 2px; margin-right: 2px; color: #000000; text-decoration:
none;}
#footer a:hover {text-decoration: underline;}
</style>
</head>
<body>

```



```

<h1>Online Web Based Programming and Database Learning System
<span class="tagline">
GUEST USER
</span></h1>
<div id="display_date">
<?php
$display_time = time();
$current_date = date("d.m.Y",$display_time);
$current_hour = date("H:i",$display_time);
?><strong>|| <?php echo "Date: ";?></strong> <?php echo "$current_date";?><b>
<< >></b> <strong> <?php echo "Time: ";?></strong> <?php
echo "$current_hour";?><strong>||</strong>
</div>
<div id="b">
<strong>You are currently not logged into our system. Once logged in, you will
gain access to upload/download section. Also you can access protected
areas, personal information, and other good stuff.<br/>
<hr />
If you are a <u>registered</u> user, an <u>administrator</u> or want to
register;</strong>
<br />
<br />
<a href="user_login.php"></a> &nbsp; <a
href="http://localhost/graduationproject/admin/admin_login.php"></a> &nbsp; <a
href="register.php"></a> <br />
<strong>&nbsp; &nbsp; User Login &nbsp; &nbsp; Admin Login &nbsp; &nbsp;
&nbsp; Register &nbsp; &nbsp; &nbsp;
</strong>
</div>
<div id="left">
<h3>LECTURES</h3>
<ul>
<li><a
href="http://localhost/graduationproject/lectures/oracle_unlogged.html"></a></li>
<li><a href="http://localhost/graduationproject/lectures/c_unlogged.html"></a></li>
<li><a
href="http://localhost/graduationproject/lectures/php_unlogged.html"></a></li>
<li><a href="http://localhost/graduationproject/lectures/xh_unlogged.html"></a></li>
<li><a
href="http://localhost/graduationproject/lectures/java_unlogged.html"></a></li>
</ul>

```

```

</div>
<div id="right">
<h3>INFORMATION & LINKS SECTION</h3>
<marquee direction="left" onmouseover="stop()" onmouseout="start()"
title="WORLD OF ENGINEERING" behavior="alternate" >
</marquee>
<ul>
<li><a href="lectures/com_eng.html">Computer Engineering</a></li>
<li><a href="lectures/soft_eng.html">Software Engineering</a></li>
<li><a href="lectures/xp.html">Extreme Programming</a></li>
</ul>
</div>
<div id="content">
<h3><strong>PROGRAMMING AND DATABASE PORTAL</strong></h3>
<p><b>From the developer...</b>
<p>Welcome to the Online programming education, database and Computer
Engineering knowledge portal.In this page you
will find the most popular and important concepts of programming languages.I
also intend to briefly cover computer enginnering
,software engineering and extreme programming which are
directly related to the programming knowledge and software design.
</p></p>

```



```


<ul>
<li>You can access information about Java from <a
href="http://www.sun.com">Sun homepage</a></li>
<li>The fully information about The Oracle Database: <a
href="http://www.oracle.com/index.html">Oracle homepage</a></li>
<li>Everything about the world wide web:<a href="http://www.w3.org/">W3
consortium</a></li>
</ul>
</div>
<div id="footer">
<p><b>Contact :</b>&#091;<a href="mailto:steelandblood@gmail.com">Ismail
Özsoy</a>&#093;</p>
<h5>&#169 2008 </h5>
</div>
<center><p><strong><u><?php include
("counter.php");?></u></strong></p></center>
<div id="powered" align="center"><a href="http://www.php.org"></a>&nbsp;
<a href="http://www.apache.com"></a>&nbsp;
<a href="http://www.mysql.com"></a>&nbsp;
<a href="http://www.xampp.com"></a>
</div>
</p>
</div>
</body>
</html>

```

### user\_personal.php

```
<?php // Contains personal information of both users and admins
session_start();
if ((isset($_SESSION['user_logged']) &&
    $_SESSION['user_logged'] != "") ||
    (isset($_SESSION['user_password']) &&
    $_SESSION['user_password'] != "")) { ?>
<?php
include "auth_user.inc.php";
include "conn.inc.php";
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>User Account Information</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
h1{ margin-top:4%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
a{ color:#000000; font:bold}
#field_set{border:solid; margin-left:20%; margin-right:20%; border-
color:#0066FF; text-align:center; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:2%}
#inline{border: solid; border-color: #0099FF;}
</style>
</head>
<body>
```

```

<h1>Welcome to your personal information area</h1>
<div id="field_set"><p>
    <span id="inline"><strong>You can manage your user information,
    or drop your account in this page.</strong></span><br><br>
    <strong><u>Your current personal information is shown
below:</u></strong><br>
    <br>
<?php
$query = "SELECT * FROM user_info " .
    "WHERE username = '" . $_SESSION['user_logged'] . "' " .
    "AND password = (PASSWORD('" .
    $_SESSION['user_password'] . "'))";
$result = mysql_query($query)
    or die(mysql_error());
$row = mysql_fetch_array($result);
?>
    <strong>First Name:</strong> <?php echo $row['first_name']; ?><br><br>
    <strong>Last Name:</strong> <?php echo $row['last_name']; ?><br><br>
    <strong>City:</strong> <?php echo $row['city']; ?><br><br>
    <strong>Country:</strong> <?php echo $row['country']; ?><br><br>
    <strong>Email:</strong> <?php echo $row['email']; ?><br><br>
    <a href="update_account.php">Update Account</a> |
    <a href="delete_account.php">Delete Account</a>
</p></div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="index.php"></a>
</div>
</body>
</html>
<?php } else { ?>

```



```

<?php      if ((isset($_SESSION['admin_logged'])&&
$_SESSION['admin_logged']) != "" ||
(isset($_SESSION['admin_password']) &&
$_SESSION['admin_password'] != "")) { ?>
    <?php include "conn.inc.php";?>
    <html>
    <head>
    <title>Admin Account Information</title>
    <style type="text/css">
        body{ background-color: #DDEEFF; font-
family:Arial, Helvetica, sans-serif; font-size: 16px}
        h1{ margin-top:8%;text-align:center; color:#0066FF;
border:solid; border-color:#0066FF; margin-left:20%; margin-right:20%; }
        #field_set{border:solid; margin-left:20%; margin-
right:20%; border-color:#0066FF; text-align:center; color:#0066FF;}
        #footer{text-align:center; color:#0066FF;
border:solid; border-color:#0066FF; margin-left:30%; margin-right:30%; margin-
top:2%}

        #inline{border: solid; border-color: #0099FF;}
    </style>
    </head>
    <body>
    <h1>Welcome to your personal information area</h1>
        <div id="field_set"><p>
            <span id="inline"><strong>You can view your
<u>Administrator</u> information from below
            .</span></strong><br><br>
    <?php

$query = "SELECT * FROM admin " .
"WHERE username = " . $_SESSION['admin_logged'] . "

```

```

" . "AND password = (PASSWORD("'" . $_SESSION['admin_password'] . "'))";

$result = mysql_query($query)
or die(mysql_error());
$row =
mysql_fetch_array($result);

?>

<strong>First
Name:</strong> <?php echo $row['first_name']; ?><br><br>
<strong>Last Name:</strong>
<?php echo $row['last_name']; ?><br><br>
<strong>Email:</strong>
<?php echo $row['email']; ?><br><br>
<strong>Admin
Level:</strong><?php echo $row['admin_level']; ?><br><br>

</p>
</div>
<div id="footer"><a
href="http://localhost/graduationproject/admin/index.php"></a>&nbsp; &nbsp;
<a href="logged_user.php"></a>
</div>

</body>
</html>
<?php } }?>

```

### register.php

```
<?php // Registration section
session_start();
ob_start();
include "conn.inc.php";
?>

<html>
<head>
<title>Registration Page</title>
<style type="text/css">
body{ background-color: #DDEEFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
#field_set{border:solid; margin-left:20%; margin-right:20%; margin-
top:3%;border-color:#0066FF; text-align:left; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:2%}
</style>
</head>
<body>
<?php
if (isset($_POST['submit']) && $_POST['submit'] == "Register") {
    if ($_POST['username'] != "" &&
        $_POST['password'] != "" &&
        $_POST['first_name'] != "" &&
        $_POST['last_name'] != "" &&
        $_POST['email'] != "") {
        $query = "SELECT username FROM user_info " .
            "WHERE username = " . $_POST['username'] . " ";
        $result = mysql_query($query)
            or die(mysql_error());
```



```

    if (mysql_num_rows($result) != 0) {
?>
<div id="field_set">
    <strong><font color="#FF0000">The Username,
    <?php echo $_POST['username']; ?>, is already in use, please choose
    another Username and try again!</font></strong>
    <form action="register.php" method="post">
        <strong>Username:</strong> <input type="text" name="username"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><br><br>
        <strong>Password:</strong> <input type="password" name="password"
style="border:solid #0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['password']; ?>"><br><br>
        <strong>Email:</strong> <input type="text" name="email" style="border:solid
#0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['email']; ?>"><br><br>
        <strong>First Name:</strong> <input type="text" name="first_name"
style="border:solid #0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['first_name']; ?>"><br><br>
        <strong>Last Name:</strong> <input type="text" name="last_name"
style="border:solid #0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['last_name']; ?>"><br><br>
        <strong>City:</strong> <input type="text" name="city" style="border:solid
#0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['city']; ?>"><br><br>
        <strong>Country:</strong> <input type="text" name="country"
style="border:solid #0066FF 2px; background-color:#A6FFFF"
        value="<?php echo $_POST['country']; ?>"><br><br>

        <input type="submit" name="submit" value="Register"> &nbsp;
        <input type="reset" value="Clear">
    </form>

```

```

</div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="unlogged_user.php"></a>
</div>
<?php
    } else {
        $query = "INSERT INTO user_info (username, password, email, " .
            "first_name, last_name, city, country) " .
            "VALUES (" . $_POST['username'] . ", " .
            "(PASSWORD(" . $_POST['password'] . ")), " .
            $_POST['email'] . ", " . $_POST['first_name'] .
            ", " . $_POST['last_name'] . ", " . $_POST['city'] .
            ", " . $_POST['country'] . ");";
        $result = mysql_query($query)
        or die(mysql_error());
        $_SESSION['user_logged'] = $_POST['username'];
        $_SESSION['user_password'] = $_POST['password'];
    ?>
<p>
    Thank you, <?php echo $_POST['first_name'] . " " .
    $_POST['last_name']; ?> for registering!<br>
<?php
    header("Refresh: 5; URL=index.php");
    echo "Your registration is complete! " .
        "You are being sent to the page you requested!<br>";
    echo "(If your browser doesn't support this, " .
        "<a href='\"index.php\"'>click here</a>);";
    die();
}
} else {

```

?>

<div id="field\_set">

<strong><font color="#FF0000">The Username, Password, Email, First Name,  
and Last Name fields are required!</font></strong>

<form action="register.php" method="post">

<fieldset>

<strong>Username:</strong> <input type="text" name="username"  
size="21" style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['username']; ?>"><br><br>

<strong>Password:</strong> <input type="password" name="password"  
size="21" style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['password']; ?>"><br><br>

<strong>Email:</strong> <input type="text" name="email" size="27"  
style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['email']; ?>"><br><br>

<strong>First Name:</strong> <input type="text" name="first\_name"  
style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['first\_name']; ?>"><br><br>

<strong>Last Name:</strong> <input type="text" name="last\_name"  
style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['last\_name']; ?>"><br><br>

<strong>City:</strong> <input type="text" name="city" size="28"  
style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['city']; ?>"><br><br>

<strong>Country:</strong> <input type="text" name="country" size="23"  
style="border:solid #0066FF 2px; background-color:#A6FFFF"  
value="<?php echo \$\_POST['country']; ?>"><br><br>

<input type="submit" name="submit" value="Register"> &nbsp;

<input type="reset" value="Clear">

</fieldset>

</form>



```

</div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="unlogged_user.php"></a>
</div>
<?php
    }
} else {
?>
<div id="field_set">
<p><center>
    <strong>WELCOME TO THE REGISTRATION SECTION!</strong><br>
    You must fill Username, Password, Email, First Name, and Last Name
fields.<br>
    <em>(City and Country fields are optional.)</em></center></p>
    <form action="register.php" method="post" >
        <fieldset >
            <p><strong>Username:</strong> <input type="text" name="username"
size="21" style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>
            <p><strong>Password:</strong> <input type="password" name="password"
size="21" style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>
            <p><strong>Email:</strong> <input type="text" name="email" size="27"
style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>
            <p><strong>First Name:</strong> <input type="text" name="first_name"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><p>
            <p><strong>Last Name:</strong> <input type="text" name="last_name"
style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>
            <p><strong>City:</strong> <input type="text" name="city" size="29"
style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>
            <p><strong>Country:</strong> <input type="text" name="country" size="24"
style="border:solid #0066FF 2px; background-color:#A6FFFF"></p>

```

```

        <input type="submit" name="submit" value="Register"> &nbsp;
        <input type="reset" value="Clear">
        </fieldset>
    </form>
</div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="unlogged_user.php"></a>
</div>
<?php
}
?>
</body>
</html>

```

### **user\_login.php**

```

<?php // Registered users use this page for login to the system with their username
// and password combination
session_start();
include "conn.inc.php";
if (isset($_POST['submit'])) {
    $query = "SELECT username, password FROM user_info " .
        "WHERE username = '" . $_POST['username'] . "' " .
        "AND password = (PASSWORD('" . $_POST['password'] . "'))";
    $result = mysql_query($query)
        or die(mysql_error());
    if (mysql_num_rows($result) == 1) {
        $_SESSION['user_logged'] = $_POST['username'];
        $_SESSION['user_password'] = $_POST['password'];
        header ("Refresh: 8; URL=" . $_POST['redirect'] . "");
    }
    <br><br>

```

```

<br><br>
<center> <strong> <?php
    echo "YOU ARE BEING REDIRECTED TO THE MAIN PAGE,PLEASE
WAIT !<br>";
    echo "(If your browser doesn't support redirection, " .
        "<a href=\"\" . $_POST['redirect']. "\"" . ">click here</a>";?> <br><br></strong></center><?php
    } else {
?>
<html>
<head>
<title>System Login Page</title>
<style type="text/css">
body{ background-color: #DDEEFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
#field_set{border:solid; margin-left:20%; margin-right:20%; margin-
top:10%;border-color:#0066FF; text-align:center; color:#0066FF;}
.legend{font-size:18px; }
.prgh{ text-align:center; font-size:20px; color:#FF0000;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:5%}
a{color:#000000}
</style>
</head>
<body >
<div id="field_set">
<span class="prgh"><p>
    Invalid Username and/or Password<br></span>
<em> Not registered?
<a href="register.php">Click here</a> to register.</em><br>
<form action="user_login.php" method="post">

```



```

<fieldset>
<span class="legend">
<legend>LOGIN</legend></span>
  <input type="hidden" name="redirect"
    value="<?php echo $_POST['redirect']; ?>">
  Username: <input type="text" name="username" style="border:solid #0066FF
2px; background-color:#A6FFFF"><br>
  Password: <input type="password" name="password" style="border:solid
#0066FF 2px; background-color:#A6FFFF"><br><br>
  <input type="submit" name="submit" value="Login">
</fieldset>
</form>
</p></div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="unlogged_user.php"></a>
</div>
</body>
</html>
<?php
}
} else {
  if (isset($_GET['redirect'])) {
    $redirect = $_GET['redirect'];
  } else {
    $redirect = "index.php";
  }
?>
<html>
<head>
<title>System Login Page</title>

```

```

<style type="text/css">
body{ background-color: #DDEEFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
#field_set{border:solid; margin-left:20%; margin-right:20%; margin-
top:10%;border-color:#0066FF; text-align:center; color:#0066FF;}
.legend{ font-size:18px; }
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:5%}
a{ color:#000000}
</style>
</head>
<body>
<div id="field_set">
<p>
    <fieldset >
        <span class="legend"> <legend >ONLINE WEB BASED EDUCATIONAL
SYSTEM</legend></span><br>
        Please, enter your <strong>'username'</strong> and <strong>'password'</strong>
to enter the system.<br><br>
        <em>Not registered yet? <a href="register.php"> Click here </a> to
register.</em><br><br>
    </fieldset>
    <form action="user_login.php" method="post">
        <fieldset>
            <span class="legend"><legend>LOGIN</legend></span>
            <input type="hidden" name="redirect"
            value="<?php echo $redirect; ?>">
            <strong>Username:</strong> <input type="text" name="username"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><br>

```

```

        <strong>Password:</strong> <input type="password" name="password"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><br><br>
        <input type="submit" name="submit" value="Login">
    </fieldset>
</form>
</p></div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="unlogged_user.php"></a>
</div>
</body>
</html>
<?php
}
?>

```

### **update\_account.php**

```

<?php // Users use this page to update their -limited- personal information
// Admins cannot update their info's through this page
// They have to access from the Admin Area
include "auth_user.inc.php";
include "conn.inc.php";
?>
<html>
<head>
<title>Update Your Account</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
h1{ margin-top:4%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }

```



```

fieldset{border:solid; margin-left:20%; margin-right:20%; border-color:#0066FF;
text-align:center; font:bold; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:2%}
</style>
</head>
<body>
<h1>Update Account Information</h1>
<br>
<p>
    <center><strong>Here you can update your account information for viewing in
your
    profile.</strong></center><br><br>
<?php
if (isset($_POST['submit']) && $_POST['submit'] == "Update") {
    $query_update = "UPDATE user_info SET
        email = " . $_POST['email'] . ",
        city = " . $_POST['city'] . ",
        country = " . $_POST['country'] . "
        WHERE username = " . $_SESSION['user_logged'] . "
        AND password = (PASSWORD(" .
        $_SESSION['user_password'] . "));";
    $result_update = mysql_query($query_update)
    or die(mysql_error());
    $query = "SELECT * FROM user_info " .
        "WHERE username = " . $_SESSION['user_logged'] . " " .
        "AND password = (PASSWORD(" .
        $_SESSION['user_password'] . "));";
    $result = mysql_query($query)
    or die(mysql_error());
    $row = mysql_fetch_array($result);

```

?>

<center> <b><font color="#FF0000">Your personal account information has  
been updated!</font></b></center><br>

<p><fieldset>

<form action="update\_account.php" method="post"><br>

Email: <input type="text" name="email" size="30" style="border:solid #0066FF;  
background-color:#A6FFFF"

value="<?php echo \$row['email']; ?>"><br><br>

City: <input type="text" name="city" size="33" style="border:solid #0066FF;  
background-color:#A6FFFF"

value="<?php echo \$row['city']; ?>"><br><br>

Country: <input type="text" name="country" size="27" style="border:solid  
#0066FF; background-color:#A6FFFF"

value="<?php echo \$row['country']; ?>"><br><br>

<input type="submit" name="submit" value="Update"> &nbsp;

<input type="button" value="Cancel" onClick="history.go(-1);">

</form>

</fieldset>

</p>

<div id="footer"><strong>BACK TO THE PREVIOUS PAGE</strong><br />

<a href="user\_personal.php"></a>

</div>

</body>

</html>

<?php

} else {

\$query = "SELECT \* FROM user\_info " .

"WHERE username = '" . \$\_SESSION['user\_logged']. "' " .

"AND password = (PASSWORD('" . \$\_SESSION['user\_password'] . "'))";

\$result = mysql\_query(\$query)

```

    or die(mysql_error());
    $row = mysql_fetch_array($result);
    ?>
<p><fieldset>
    <form action="update_account.php" method="post">
        <br>
        Email: <input type="text" name="email" size="30" style="border:solid
#0066FF; background-color:#A6FFFF"
        value="<?php echo $row['email']; ?>"><br><br>
        City: <input type="text" name="city" size="33" style="border:solid #0066FF;
background-color:#A6FFFF"
        value="<?php echo $row['city']; ?>"><br><br>
        Country: <input type="text" name="country" size="27" style="border:solid
#0066FF; background-color:#A6FFFF"
        value="<?php echo $row['country']; ?>"><br><br>
        <input type="submit" name="submit" value="Update"> &nbsp;
        <input type="button" value="Cancel" onClick="history.go(-1);"> &nbsp;
        <input type="reset" value="Reset" name="reset">
    </form>
</fieldset>
</p>
<div id="footer"><strong>BACK TO THE PREVIOUS PAGE</strong><br />
<a href="user_personal.php"></a>
</div>
</body>
</html><?php?>

```

### **delete\_account.php**

```

<?php // Users can Drop their account from this page
include "auth_user.inc.php";
include "conn.inc.php";

```



```

if (isset($_POST['submit']) && $_POST['submit'] == "Yes") {
    $query_delete = "DELETE FROM user_info " .
        "WHERE username = '" . $_SESSION['user_logged'] .
        "' AND password = (PASSWORD('" .
        $_SESSION['user_password'] . "'))";
    $result_delete = mysql_query($query_delete)
        or die(mysql_error());
    $_SESSION['user_logged'] = "";
    $_SESSION['user_password'] = "";
    header("Refresh: 5; URL=index.php");
    ?>
    <br>
    <br>
    <br>
    <center><strong>
    <?php
    echo "YOUR ACCOUNT HAS BEEN DELETED!<br><br> " .
        "YOU ARE BEING REDIRECTED TO THE MAIN PAGE,PLEASE WAIT
    !<br><br>";
    echo "(If you're browser doesn't support this, " .
        "<a href='\"index.php\">click here</a>)" . "></strong></center><?php
    die();
    } else {
    ?>
    <html>
    <head>
    <title>Delete Your Account</title>
    <style type="text/css">
    body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
    size: 16px}

```

```

h1{ margin-top:4%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:20%; margin-right:20%; border-color:#0066FF;
text-align:center; font:bold; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:2%}
#inline{ text-align:center; font:bold; font-size:18px}
</style>
</head>
<body>
<h1>Drop Account</h1>
<p>
<div id="inline">Your Username is <?php echo $_SESSION['user_logged']
?></div><br>
<fieldset>
<br>
    Are you sure you want to delete your account?<br><br>
    <font color="#FF0000">Your account information will be dropped!<br> There is
no way to retrieve your account once you confirm!</font><br>
    <form action="delete_account.php" method="post">
        <input type="submit" name="submit" value="Yes"> &nbsp;
        <input type="button" value=" No " onClick="history.go(-1);">
    </form>
</fieldset>
</p>
<div id="footer"><strong>BACK TO THE PREVIOUS PAGE</strong><br />
<a href="user_personal.php"></a>
</div>
</body>
</html>

```

```
<?php}?>
```

### **counter.php**

```
<?php // Basic hit counter, counts the page entry and write into a text document
$count_my_page = ("hitcounter.txt");
$hits = file($count_my_page);
$hits[0] ++;
$fp = fopen($count_my_page , "w");
fputs($fp , "$hits[0]");
fclose($fp);
echo "NUMBER OF VISITORS: ".$hits[0];
?>
```

### **log\_out.php**

```
<?php
// re-create the previous session
session_start();
// print the value of the session variable
?>

<style type="text/css">
body{ background-color:#9BCDFF; color:#0066FF}
#p1{border:solid 3px #0066FF; text-align:center; font:bold 20px; margin-
top:10%; margin-left:35%; margin-right:35%}
#p2{text-align:center; font:bold 20px; margin-top:5%}
a { color:#000000 }
</style>

<div id="p1"><p>You are currently logged out <b><?php echo
$_SESSION['user_logged'] ;?></b></p></div>

<?php
// reset session array
$_SESSION = array();
// destroy session
session_destroy();
```



?>

<div id="p2"><p>To return the main page please <a href="index.php">click  
here</a>.</p></div>

## APPENDIX C

### Administrator's PHP codes and MySQL Queries

#### admin\_initialization.php

```
<?php // Inserting the root admin information into Admin database table
$conn = mysql_connect("localhost", "root", "com400")
or die(mysql_error());
$db = mysql_select_db("registration")
or die(mysql_error());
$query = "INSERT INTO admin (username, password ,first_name,
last_name,email,admin_level,id)
VALUES ('shadow', PASSWORD('123'),'Ismail Vehbi',
'Ozsoy','ismailvehbiozsoy@gmail.com','1','");";
mysql_query($query) or die('Error, query failed');
echo "Admin initialization successfully completed !";
?>
```

#### conn.inc.php

```
<?php // Opening a connection to the MySQL Database
$conn = mysql_connect("localhost", "root", "com400")
or die(mysql_error());
$db = mysql_select_db("registration")
or die(mysql_error());
?>
```

#### admin\_login.php

```
<?php // Admins use their usernames and passwords to enter system
session_start();
include "conn.inc.php";
if (isset($_POST['submit'])) {
    $query = "SELECT username, password, admin_level FROM admin " .
        "WHERE username = '" . $_POST['username'] . "' " .
        "AND password = (password('" . $_POST['password'] . "'))";
```

```

$result = mysql_query($query)
    or die(mysql_error());
$row = mysql_fetch_array($result);
if (mysql_num_rows($result) == 1) {
    $_SESSION['admin_logged'] = $_POST['username'];
    $_SESSION['admin_password'] = $_POST['password'];
    $_SESSION['admin_level'] = $row['admin_level'];
    header ("Refresh: 5; URL=" . $_POST['redirect'] . "");?>
<br><br>
<br><br>
<center> <strong> <?php
    echo "YOU ARE BEING REDIRECTED TO THE ADMINISTRATOR MAIN
PAGE,PLEASE WAIT!<br>";
    echo "(If your browser doesn't support this,or wait too much " .
        "<a href=\"\" . $_POST['redirect']. \"\">click here</a>)"?> <br><br></strong></center><?php
    } else {
?>
<html>
<head>
<title>Admin Login Page</title>
<style type="text/css">
body{ background-color: #DDEEFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
#field_set{border:solid; margin-left:20%; margin-right:20%; margin-
top:10%;border-color:#0066FF; text-align:center; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:5%}
</style>
</head>
<body>

```



```

<div id="field_set">
    <font color="#FF0000" style="font-size:24px">Invalid Username and/or
    Password<br/>
    Access Denied!</font>
    <form action="admin_login.php" method="post">
    <fieldset>
        <legend><strong>ADMIN LOGIN</strong></legend>
        <input type="hidden" name="redirect"
        value="<?php echo $_POST['redirect']; ?>">
        <br>
        <strong>Username:</strong> <input type="text" name="username"
        style="border:solid #0066FF 2px; background-color:#A6FFFF"><br>
        <strong>Password:</strong> <input type="password" name="password"
        style="border:solid #0066FF 2px; background-color:#A6FFFF"><br><br>
        <input type="submit" name="submit" value="Login">
        <input type="reset" name="reset" value="Clear All">
    </fieldset>
    </form>
</div>

<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="../unlogged_user.php"></a>
</div>
</body>
</html>
<?php
    }
} else {
    if (isset($_GET['redirect'])) {
        $redirect = $_GET['redirect'];
    } else {

```

```

    $redirect = "index.php";
}
?>
<html>
<head>
<title>Admin Login Page</title>
<style type="text/css">
body{ background-color: #DDEEFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px}
#field_set{border:solid; margin-left:20%; margin-right:20%; margin-
top:10%;border-color:#0066FF; text-align:center; color:#0066FF;}
#footer{text-align:center; color:#0066FF; border:solid; border-color:#0066FF;
margin-left:30%; margin-right:30%; margin-top:5%}
</style>
</head>
<body>
<div id="field_set">
Enter your Administrator Username and Password<br>
<form action="admin_login.php" method="post">
<fieldset>
    <legend><strong>ADMIN LOGIN</strong></legend>
    <input type="hidden" name="redirect"
value="<?php echo $redirect; ?>">
<br>
    <strong>Username:</strong> <input type="text" name="username"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><br>
    <strong>Password:</strong> <input type="password" name="password"
style="border:solid #0066FF 2px; background-color:#A6FFFF"><br><br>
    <input type="submit" name="submit" value="Login">
    <input type="reset" name="reset" value="Clear All">
</fieldset>

```

```

    </form>
</div>
<div id="footer"><strong>BACK TO THE MAIN PAGE</strong><br />
<a href="../unlogged_user.php"></a>
</div>
</body>
</html>
<?php}?>

```

### **Index.php(admin)**

```

<?php // Controlling whether an admin logged or not and show the related pages
session_start();
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
    include "logged_admin.php";
} else {
    include "unlogged_admin.php";
}
?>

```

### **logged\_admin.php**

```

<?php // Included file to the logged admins
include "conn.inc.php";
session_start();
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
?>
<html>

```





```

<p>
  <a href="admin_area.php">Click here</a>
  to access the administration section.
</p>
<p><a href="http://localhost/graduationproject/logged_user.php">Click here</a>
to go to main page</p>
<p><a href="admin_logout.php">LOGOUT</a></p></div>
</body>
</html>
<?php } ?>

```

#### unlogged\_admin.php

```

<?php // Unlogged admin contents added to the index ?>
<html>
<head>
<title>Unlogged Admin</title>
<style type="text/css">
body{ background-color:#000000; font-family: "Bookman Old Style"; font-size:
16px; color:#FFFFFF; text-align:center; background-
image:url(../backgrounds/darkaurora_01.jpg);
background-position:center; background-repeat:no-repeat; background-
attachment:fixed;}
h1{ margin-top:7%;text-align:center; color: #FFFFFF; border:solid; border-
color:#6AA6A7; margin-left:20%; margin-right:20%; }
a{color:#FFFFFF;}
</style>
</head>
<body >
<h1>Welcome to the Unlogged Admin Area!</h1>
<p>
  <font size="5" color="#FF0000">You are not currently logged in.</font><br>
  You will have gain access to your administrator rights once login.<br>

```

```
<a href="admin_login.php">Click here</a> to login.
```

```
</p>
```

```
</body>
```

```
</html>
```

### admin\_area.php

```
<?php // Main administration area
```

```
session_start();
```

```
include "auth_admin.inc.php";
```

```
include "conn.inc.php";
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title>Administration Page</title>
```

```
<style type="text/css">
```

```
body{ background-color:#000000; font-family: "Bookman Old Style"; font-size:
16px; background-image:url(../backgrounds/darkaurora_02.jpg);
background-position:center; background-repeat:no-repeat; background-
attachment:fixed;}
```

```
h1{ margin-top:3%;text-align:center; color: #FFFFFF; border:solid; border-
color:#0066FF; margin-left:2%; margin-right:2%; }
```

```
a{color:#FFFFFF; font:bold; }
```

```
legend{color:#0066FF; font-size:20px;}
```

```
fieldset{font-size:18px; margin-left:10%;margin-right:10%; margin-
top:2%;border-bottom:solid; border-bottom-color:#0066FF; border-top:solid;
border-top-color:#0066FF; border-left:solid; border-left-color:#0066FF; border-
right:solid; border-right-color:#0066FF; text-align: justify;color: #FFFFFF; }
```

```
hr{ color: #0066FF}
```

```
#logout {color:#FFFFFF; font-size:18px; text-align:left}
```

```
</style>
```

```
</head>
```

```
<body >
```



```

<h1>Administration Section</h1>
<center><a href="index.php">BACK TO THE MAIN ADMIN
PAGE</a></center><br>
<p><center><a href="admin_logout.php">LOGOUT</a></center></p>
<hr>
<fieldset>
  <legend><strong>LIST OF CURRENT USERS</strong></legend>
  <br>
<?php
if (isset($_SESSION['admin_level']) &&
    $_SESSION['admin_level'] != "1") {
    $query = "SELECT first_name, last_name, id FROM user_info " .
        "ORDER BY last_name";
    $result = mysql_query($query)
        or die(mysql_error());
    while ($row = mysql_fetch_array($result)) {
        echo $row['first_name'] . " " . $row['last_name'];
    }
    &nbsp;&nbsp;<a href="update_user.php?id=?php
    echo $row['id']; ?>">Update User</a><br>
<?php
}
} else {
    $query = "SELECT first_name, last_name, id FROM user_info " .
        "ORDER BY last_name";
    $result = mysql_query($query)
        or die(mysql_error());
    while ($row = mysql_fetch_array($result)) {
        echo $row['first_name'] . " " . $row['last_name'];
    }
    &nbsp;&nbsp;<a href="update_user.php?id=?php

```

```

        echo $row['id']; ?>">Update User</a> |
        <a href="delete_user.php?id=?php echo $row['id'];?>">Delete User</a><br>
    <?php
    }
    ?>
    <?php
    }
    ?>
</fieldset>
<br>
<hr>
<br>
<fieldset>
<legend><strong>ADD USERS/ADMINS AND MANAGE
ADMINS</strong></legend>
<p>To add <b><em>Users</em></b>,<em><b>Admins</b></em> or manage
<b><em>Admins</em></b><a href="adduser.php"> click here </a></p>
</fieldset>
<br>
<hr>
<br>
<fieldset>
<legend><strong>UPLOAD/DOWNLOAD/DELETE
CONTENT</strong></legend>
<p><a href="http://localhost/graduationproject/upload/updown.php">Click here
</a>to upload/download courses etc...</p>
</fieldset>
<br>
<hr>
<br>
<hr/>

```

```
</body></html>
```

### **adduser.php**

```
<?php // The main add user page
```

```
session_start();
```

```
if ((isset($_SESSION['admin_logged']) &&  
    $_SESSION['admin_logged'] != "" ||  
    (isset($_SESSION['admin_password']) &&  
    $_SESSION['admin_password'] != "")) {
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title>ADD USERS</title>
```

```
<style type="text/css">
```

```
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-  
size: 16px; font:bold;}
```

```
h2{ margin-top:8%;text-align:center; color:#0066FF; border:solid; border-  
color:#0066FF; margin-left:20%; margin-right:20%; }
```

```
#field_set{border:solid; margin-left:20%; margin-right:20%; border-  
color:#0066FF; text-align:justify; color:#0066FF; }
```

```
a {color:#000000;}
```

```
ul { font:larger;}
```

```
#footer{text-align:center; color:#0066FF;}
```

```
hr {color:#0066FF};
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Add Users/Admins and Manage Admin Privilages</h2>
```

```
<br>
```

```
<br>
```

```
<div id="field_set">
```

```
<ul>
```



```

</li>To add new users, please <a href="adduser1.php">click here</a></p></li>
<li>To add,view or manage admins please <a href="adduser2.php">click
here</a></p></li>
</ul></p></div>
<br>
<br>
<hr>
<br>
<br>
<div id="footer"><a href="admin_area.php">Click here</a> to go back.
</body>
</html>
<?php }
else{
echo "You are NOT logged in,plese login first!";
?><p><a href="admin_login.php">Click here </a>to <b>LOGIN</b> with an
<b>ADMIN</b> account.</p>
<?php
}
?>

```

### **adduser1.php**

```

<?php // Admins can add users using this page
session_start();
include "conn.inc.php";
?>
<html>
<head>
<title>Add New User</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}

```

```

h1 { margin-top:10%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:20%; margin-right:20%; border-color:#0066FF;
text-align:justify; color:#0066FF; text-align:center;margin-top:3%}
a {color:#000000;}
#add_success {border:solid; border-color:#0066FF; margin-left:20%; margin-
top:10%; margin-right:20%;}
</style>
</head>
<body bgcolor="#D9FFFF">
<?php
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged']) != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
if (isset($_POST['submit']) && $_POST['submit'] == "Insert") {
    if ($_POST['username'] != "" &&
        $_POST['password'] != "" &&
        $_POST['first_name'] != "" &&
        $_POST['last_name'] != "" &&
        $_POST['email'] != "") {
        $query = "SELECT username FROM user_info " .
            "WHERE username = " . $_POST['username'] . " ";
        $result = mysql_query($query)
            or die(mysql_error());
        if (mysql_num_rows($result) != 0) {
?>
<p>
<br>
<br>
<br><br>

```

```

<br>
<center> <font color="#FF0000"><b>The username,
<?php echo $_POST['username']; ?>, is already in use, please choose
another!</b></font></center>
<fieldset>
<br>
<form action="adduser1.php" method="post">
    Username: <input type="text" name="username"><br><br>
    Password: <input type="password" name="password"
        value="<?php echo $_POST['password']; ?>"><br><br>
    Email: <input type="text" name="email"
        value="<?php echo $_POST['email']; ?>"><br><br>
    First Name: <input type="text" name="first_name"
        value="<?php echo $_POST['first_name']; ?>"><br><br>
    Last Name: <input type="text" name="last_name"
        value="<?php echo $_POST['last_name']; ?>"><br><br>
    City: <input type="text" name="city"
        value="<?php echo $_POST['city']; ?>"><br><br>
    Country: <input type="text" name="country"
        value="<?php echo $_POST['country']; ?>"><br><br>

    <input type="submit" name="submit" value="Insert"> &nbsp;
    <input type="reset" value="Clear">
</form><br></fieldset>
</p>
<?php
} else {
    $query = "INSERT INTO user_info (username, password, email, " .
        "first_name, last_name, city, country) " .
        "VALUES (" . $_POST['username'] . ", " .
        "(PASSWORD(" . $_POST['password'] . ")), " .

```



```

$_POST['email'] . " , " . $_POST['first_name'] .
" , " . $_POST['last_name'] . " , " . $_POST['city'] .
" , " . $_POST['country'] . " );";
$result = mysql_query($query)
or die(mysql_error());
?>
<div id="add_success">
<p>
The new user<font color="#FF0000"> <?php echo $_POST['first_name'] . " " .
$_POST['last_name']; ?></font> has been added successfully to the
system.<br></p></div>
<?php

}
} else {
?>
<p>
<br>
<br><br>
<br><br>
<center><font color="#FF0000"><b>The Username, Password, Email, First
Name,
and Last Name fields are required!</b></font></center>
<form action="adduser1.php" method="post">
<fieldset>
<br>
Username: <input type="text" name="username"
value="<?php echo $_POST['username']; ?>"><br><br>
Password: <input type="password" name="password"
value="<?php echo $_POST['password']; ?>"><br><br>
Email: <input type="text" name="email"

```

```

        value="<?php echo $_POST['email']; ?>"><br><br>
First Name: <input type="text" name="first_name"
        value="<?php echo $_POST['first_name']; ?>"><br><br>
Last Name: <input type="text" name="last_name"
        value="<?php echo $_POST['last_name']; ?>"><br><br>
City: <input type="text" name="city"
        value="<?php echo $_POST['city']; ?>"><br><br>
Country: <input type="text" name="country"
        value="<?php echo $_POST['country']; ?>"><br><br>

<input type="submit" name="submit" value="Insert"> &nbsp;
<input type="reset" value="Clear">
    <br>
    <br>
</fieldset>
</form>
</p>
<?php
    }
} else {
?>
<p>
    <fieldset>
    <legend>ADD NEW USER TO THE SYSTEM</legend>
    <br>
    Username, Password, Email, First Name, and Last Name fields
    are required to register the user successfully!<br><br>
    </fieldset>
    <form action="adduser1.php" method="post" >
    <fieldset >
        <br>

```

```

        <p>Username: <input type="text" name="username"></p>
        <p>Password: <input type="password" name="password"></p>
        <p>Email: <input type="text" name="email"></p>
        <p>First Name: <input type="text" name="first_name"><p>
        <p>Last Name: <input type="text" name="last_name"></p>
        <p>City: <input type="text" name="city"></p>
        <p>Country: <input type="text" name="country"></p>
        <input type="submit" name="submit" value="Insert"> &nbsp;
        <input type="reset" value="Clear">
        <br>
        <br>
    </fieldset>
</form>
</p>
<?php
}
}
?>
<center><p><a href="adduser.php">Click here </a> to go back </p></center>
</body>
</html>

```

### **adduser2.php**

```

<?php // Admins can add new Admins to the system using this page
include "conn.inc.php";
session_start();
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
?>
<html>

```



```

<head>
<title>Add/Display Admins</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}
h1{ margin-top:10%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:10%; margin-right:10%; border-color:#0066FF;
text-align:justify; color:#0066FF; margin-top:5%}
a {color: #000000;}
hr{color: #0099FF}
</style>
</head>
<body>
<fieldset><legend>LIST OF CURRENT ADMINISTRATORS</legend>
<?php
    $query="SELECT username,first_name,last_name,id,admin_level,email
FROM admin ORDER by admin_level";
    $result=mysql_query($query) or die (mysql_error());
    while ($row=mysql_fetch_array($result))
    {
        ?>
        <ul><li><b><em>Username:</em></b> <?php echo
$row['username'];?><br>
        <b>Fullname:</b> <?php echo $row['first_name'] ."
".$row['last_name'];?><br>
        <b>Admin Level:</b> <?php echo $row['admin_level'];?><br><br>
        <?php
            if ($_SESSION['admin_level'] == "1") {?>
                <a href="update_admin.php?id=<?php
echo $row['id']; ?>">Update Admin</a><br>

```

```

        <a href="delete_admin.php?id=?php
echo $row['id']; ?>">Delete Admin</a>

```

```

        <?php } ?>
    </li></ul>

```

```

<hr>

```

```

<?php } ?>

```

```

</fieldset>

```

```

<?php

```

```

if (isset($_POST['submit']) && $_POST['submit'] == "Insert") {

```

```

    if ($_POST['username'] != "" &&

```

```

        $_POST['password'] != "" &&

```

```

        $_POST['first_name'] != "" &&

```

```

        $_POST['last_name'] != "" &&

```

```

        $_POST['email'] != "")

```

```

    {

```

```

        $query = "SELECT username FROM admin " .

```

```

        "WHERE username = " . $_POST['username'] . " ";

```

```

        $result = mysql_query($query)

```

```

        or die(mysql_error());

```

```

        if (mysql_num_rows($result) != 0) {

```

```

    ?>

```

```

    <fieldset><font color="#FF0000"><b>The Username,

```

```

    <?php echo $_POST['username']; ?>, is already in use, please choose
    another!</b></font>

```

```

    <form action="adduser2.php" method="post">

```

```

        Username: <input type="text" name="username"><br>

```

```

        Password: <input type="password" name="password"

```

```

            value="<?php echo $_POST['password']; ?>"><br>

```

```

        First Name: <input type="text" name="first_name"

```

```

            value="<?php echo $_POST['first_name']; ?>"><br>

```

```

Last Name: <input type="text" name="last_name"
        value="<?php echo $_POST['last_name']; ?>"><br>
Email: <input type="text" name="email"
        value="<?php echo $_POST['email']; ?>"><br>
<input type="submit" name="submit" value="Insert"> &nbsp;
<input type="reset" value="Clear">
</form></fieldset>
<?php
    } else {
        $query = "INSERT INTO admin (username, password, " .
            "first_name, last_name,email,admin_level ) " .
            "VALUES (" . $_POST['username'] . ", " .
            "(PASSWORD(" . $_POST['password'] . ")), " .
            $_POST['first_name'] . ", " . $_POST['last_name'] .
            ", " . $_POST['email'] . ", '2');"
        $result = mysql_query($query)
        or die(mysql_error());
    ?>
<p>
    <fieldset style="text-align:center"> The new admin<?php echo
    $_POST['first_name'] . " " .
    $_POST['last_name']; ?></font> has been added successfully to the
    system.<br></fieldset>
</p>
<?php
    }
    } else {
    ?>
<p>
    <font color="#FF0000"><b>The Username, Password, Email, First Name,
    and Last Name fields are required!</b></font>

```



```

<form action="adduser1.php" method="post">
  <fieldset>
    Username: <input type="text" name="username"
      value="<?php echo $_POST['username']; ?>"><br>
    Password: <input type="password" name="password"
      value="<?php echo $_POST['password']; ?>"><br>
    First Name: <input type="text" name="first_name"
      value="<?php echo $_POST['first_name']; ?>"><br>
    Last Name: <input type="text" name="last_name"
      value="<?php echo $_POST['last_name']; ?>"><br>
    Email: <input type="text" name="email"
      value="<?php echo $_POST['email']; ?>"><br>
    <input type="submit" name="submit" value="Insert"> &nbsp;
    <input type="reset" value="Clear">
  </fieldset>
</form>
</p>
<?php
}
} else {
?>
<p>
  <fieldset>
    <legend>Add New Admin</legend>
    The Username, Password,First Name,Last Name and Email fields
    are required! <em>(The default admin level is '2')</em>
  </fieldset>
  <form action="adduser2.php" method="post" >
    <fieldset >
      <p>Username: <input type="text" name="username"></p>
      <p>Password: <input type="password" name="password"></p>

```

```

    <p>First Name: <input type="text" name="first_name"><p>
    <p>Last Name: <input type="text" name="last_name"></p>
    <p>Email: <input type="text" name="email"></p>
    <input type="submit" name="submit" value="Insert"> &nbsp;
    <input type="reset" value="Clear">
    </fieldset>
</form>
</p>
<?php }}
else { echo "You are not authorized,please <a href='admin_login.php'>logged</a>
in first .";}>
<center><p><a href="admin_area.php">Click here </a>to go back</p></center>
</body>
</html>

```

### **update\_user.php**

```

<?php // Admins can update users through this page
session_start();
include "auth_admin.inc.php";
include "conn.inc.php";
?>
<html>
<head>
<title>Admin Update User</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}
h1{ margin-top:3%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
#field_set{border:solid; margin-left:20%; margin-right:20%; border-
color:#0066FF; text-align:justify; color:#0066FF; padding-left:18%}
a {color:#000000}

```

```

</style>
</head>
<body>
<h1>Update User Information</h1>
<p>
<?php
if (isset($_POST['submit']) && $_POST['submit'] == "Update") {
    $query_update = "UPDATE user_info SET
                        username = " . $_POST['username'] . ",
                        password = (PASSWORD(" . $_POST['password']
. ")),
                        first_name = " . $_POST['first_name'] . ",
                        last_name = " . $_POST['last_name'] . ",
                        email = " . $_POST['email'] . ",
                        city = " . $_POST['city'] . ",
                        country = " . $_POST['country'] . "
                        WHERE id = " . $_POST['id'] . """;

    $result_update = mysql_query($query_update)
    or die(mysql_error());
    $query = "SELECT * FROM user_info WHERE id = " . $_POST['id'] . """;
    $result = mysql_query($query)
    or die(mysql_error());
    $row = mysql_fetch_array($result);
    ?> <br><br><div id="field_set">
    <font color="#FF0000">User information has been updated.</font><br><br>
    <form action="update_user.php" method="post">
    <input type="hidden" name="id" value="<?php echo $_POST['id']; ?>">
    Username: <input type="text" name="username" style="border:solid #0066FF
2px; background-color:#A6FFFF"
    value="<?php echo $row['username']; ?>"><br><br>

```



Password: <input type="password" name="password" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="">[Private]<br><br>

First Name: <input type="text" name="first\_name" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="<?php echo \$row['first\_name']; ?>"><br><br>

Last Name: <input type="text" name="last\_name" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="<?php echo \$row['last\_name']; ?>"><br><br>

Email: <input type="text" name="email" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="27"

value="<?php echo \$row['email']; ?>"><br><br>

City: <input type="text" name="city" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="29"

value="<?php echo \$row['city']; ?>"><br><br>

Country: <input type="text" name="country" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="24"

value="<?php echo \$row['country']; ?>"><br><br>

<input type="submit" name="submit" value="Update">

</form></div><center><a href="admin\_area.php">Click here</a> to return to the admin area</center>

<?php

} else {

\$query = "SELECT \* FROM user\_info WHERE id = " . \$\_GET['id'] . "";

\$result = mysql\_query(\$query)

or die(mysql\_error());

\$row = mysql\_fetch\_array(\$result);

?>

<div id="field\_set">

<form action="update\_user.php" method="post">

<input type="hidden" name="id" value="<?php echo \$\_GET['id']; ?>">

Username: <input type="text" name="username" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="<?php echo \$row['username']; ?>"><br><br>

Password: <input type="password" name="password" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="" > [Private]<br><br>

First Name: <input type="text" name="first\_name" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="<?php echo \$row['first\_name']; ?>"><br><br>

Last Name: <input type="text" name="last\_name" style="border:solid #0066FF 2px; background-color:#A6FFFF"

value="<?php echo \$row['last\_name']; ?>"><br><br>

Email: <input type="text" name="email" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="27"

value="<?php echo \$row['email']; ?>"><br><br>

City: <input type="text" name="city" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="29"

value="<?php echo \$row['city']; ?>"><br><br>

Country: <input type="text" name="country" style="border:solid #0066FF 2px; background-color:#A6FFFF" size="24"

value="<?php echo \$row['country']; ?>"><br><br>

<input type="submit" name="submit" value="Update"> &nbsp;

<input type="button" value="Cancel" onClick="history.go(-1);">

</form></div><center><a href="admin\_area.php">Click here</a> to return to the admin area</center>

<?php

}

?>

</p>

</body>

</html>

### delete\_user.php

```
<?php // Admins can delete users from the system
session_start();
include "auth_admin.inc.php";
include "conn.inc.php";
if ($_SESSION['admin_level'] == "1") {
    if (isset($_POST['submit']) && $_POST['submit'] == "Yes") {
        $query_delete = "DELETE FROM user_info " .
            "WHERE id = " . $_POST['id'] . "" ;
        $result_delete = mysql_query($query_delete)
            or die(mysql_error());
        $_SESSION['user_logged'] = "";
        $_SESSION['user_password'] = "";
        header("Refresh: 5; URL=admin_area.php");
        echo "Account has been deleted! " .
            "You are being sent to the admin area!<br>";
        echo "(If you're browser doesn't support this, " .
            "<a href='\"admin_area.php\">click here</a>";
        die();
    } else {
?>
<html>
<head>
<title>Admin Delete User</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}
h1 { margin-top:10%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
```



```

#field_set{border:solid; margin-left:20%; margin-right:20%; border-
color:#0066FF; text-align:justify; color:#0066FF; text-align:center;margin-
top:5%}
a {color:#000000;}
</style>
</head>
<body>
<h1>Delete Account</h1>
<div id="field_set"><p>
    Are you sure you want to<font color="#FF0000"><u> delete</u></font> this
    user's account?<br><br>
    If you delete this account,there is no way to retrieve it back once you
    confirm!<br>
    <form action="delete_user.php" method="post">
        <input type="hidden" name="id" value="<?php echo $_GET['id']; ?>">
        <input type="submit" name="submit" value="Yes"> &nbsp;
        <input type="button" value=" No " onClick="history.go(-1);">
    </form>
</p></div><center><a href="admin_area.php">Click here</a> to return to the
admin area</center>
</body>
</html>
<?php
    }
} else {
?>
    <center><font color="#FF0000">You don't have a high enough privilege to
    delete a user.<br>
    <a href="admin_area.php">Click here</a> to go back.</font></center>
<?php
} ?>

```

### update\_admin.php

```
<?php // High level admins can update current admins
session_start();
include "auth_admin.inc.php";
include "conn.inc.php";
?>

<html>
<head>
<title>Update Admin Accounts</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}
h1{ margin-top:3%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
#field_set{border:solid; margin-left:20%; margin-right:20%; border-
color:#0066FF; text-align:justify; color:#0066FF; padding-left:18%}
a {color:#000000}
</style>
</head>
<body>
<h1>Update Admin Information</h1>
<p>
<?php
if (isset($_POST['submit']) && $_POST['submit'] == "Update") {
    $query_update = "UPDATE admin SET
                                username = " . $_POST['username'] . " ,
                                password = (PASSWORD(" . $_POST['password']
. ")),
                                first_name = " . $_POST['first_name'] . " ,
                                last_name = " . $_POST['last_name'] . " ,
                                email = " . $_POST['email'] . " ,
```

```

        admin_level= "" . $_POST['admin_level'] . ""

        WHERE id = "" . $_POST['id'] . """;

$result_update = mysql_query($query_update)
    or die(mysql_error());
$query = "SELECT * FROM admin WHERE id = "" . $_POST['id'] . """;
$result = mysql_query($query)
    or die(mysql_error());
$row = mysql_fetch_array($result);
?> <br><br><div id="field_set">
<font color="#FF0000">Admin information has been updated.</font><br><br>
<form action="update_admin.php" method="post">
<input type="hidden" name="id" value="<?php echo $_POST['id']; ?>">
Username: <input type="text" name="username"
        value="<?php echo $row['username']; ?>"><br><br>
Password: <input type="password" name="password"
        value="">[Private]<br><br>
First Name: <input type="text" name="first_name"
        value="<?php echo $row['first_name']; ?>"><br><br>
Last Name: <input type="text" name="last_name"
        value="<?php echo $row['last_name']; ?>"><br><br>
Email: <input type="text" name="email"
        value="<?php echo $row['email']; ?>"><br><br>
Admin Level: <input type="text" name="admin_level"
        value="<?php echo $row['admin_level']; ?>"><br><br>
<input type="submit" name="submit" value="Update">
</form></div><center><a href="admin_area.php">Click here</a> to return to
the admin area</center>
<?php
} else {
    $query = "SELECT * FROM admin WHERE id = "" . $_GET['id'] . """;

```



```

$result = mysql_query($query)
    or die(mysql_error());
$row = mysql_fetch_array($result);
?>
<div id="field_set">
<form action="update_admin.php" method="post">
    <input type="hidden" name="id" value="<?php echo $_GET['id']; ?>">
    Username: <input type="text" name="username"
        value="<?php echo $row['username']; ?>"><br><br>
    Password: <input type="password" name="password"
        value="" > [Private]<br><br>
    First Name: <input type="text" name="first_name"
        value="<?php echo $row['first_name']; ?>"><br><br>
    Last Name: <input type="text" name="last_name"
        value="<?php echo $row['last_name']; ?>"><br><br>
    Email: <input type="text" name="email"
        value="<?php echo $row['email']; ?>"><br><br>
    Admin Level: <input type="text" name="admin_level"
        value="<?php echo $row['admin_level']; ?>"><br><br>
    <input type="submit" name="submit" value="Update"> &nbsp;
    <input type="button" value="Cancel" onClick="history.go(-1);">
</form></div><center><a href="admin_area.php">Click here</a> to return to
the admin area</center>
<?php
}
?>
</p>
</body>
</html>

```

### delete\_admin.php

```
<?php // High level admins can delete admins from the system
session_start();
include "auth_admin.inc.php";
include "conn.inc.php";
if ($_SESSION['admin_level'] == "1") {
    if (isset($_POST['submit']) && $_POST['submit'] == "Yes") {
        $query_delete = "DELETE FROM admin " .
            "WHERE id = " . $_POST['id'] . " ";
        $result_delete = mysql_query($query_delete)
            or die(mysql_error());
        header("Refresh: 5; URL=adduser2.php");
        echo "Account has been deleted! " .
            "You are being sent to the admin area!<br>";
        echo "(If your browser doesn't support this, " .
            "<a href='\"adduser2.php\">click here</a>)" .
            die();
    } else {
?>
<html>
<head>
<title>Admin Deletion</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 16px; font:bold;}
h1{ margin-top:10%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
#field_set{border:solid; margin-left:20%; margin-right:20%; border-
color:#0066FF; text-align:justify; color:#0066FF; text-align:center;margin-
top:5%}
a {color:#000000;}
```

```

</style>
</head>
<body>
<h1>Delete Admin Account</h1>
<div id="field_set"><p>
  Are you sure you want to<font color="#FF0000"><u> delete</u></font> this
  admin's account?<br><br>
  If you delete this account,there is no way to retrieve it back once you
  confirm!<br>
  <form action="delete_admin.php" method="post">
    <input type="hidden" name="id" value="<?php echo $_GET['id']; ?>">
    <input type="submit" name="submit" value="Yes"> &nbsp;
    <input type="button" value=" No " onClick="history.go(-1);">
  </form>
</p></div><center><a href="adduser2.php">Click here</a> to return to the admin
  area</center>
</body>
</html>
<?php
  }
} else {
?>
  <center><font color="#FF0000">You don't have a high enough privilege to
  delete an admin!<br>
  <a href="admin_area.php">Click here</a> to go back.</font></center>
<?php
  }
?>

```



### **admin\_logout.php**

```
<?php //Logout function
session_start();
?>

<style type="text/css">
body{ background-color:#9BCDFF; color:#0066FF}
#p1{border:solid 3px #0066FF; text-align:center; font:bold 20px; margin-
top:10%; margin-left:35%; margin-right:35%}
#p2{text-align:center; font:bold 20px; margin-top:5%}
a { color:#000000 }
</style>

<div id="p1"><p>You are currently logged out <b><?php echo
$_SESSION['admin_logged'] ;?></b></p></div>

<?php
session_unset();
session_destroy();
?>

<div id="p2"><p><a href="admin_login.php">Click here to login
again</a></p></div>
```

### **auth\_admin.inc.php**

```
<?php // Controlling the access privileges
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
    //Do Nothing!
} else {
    $redirect = $_SERVER['PHP_SELF'];
    header("Refresh: 5; URL=admin_login.php?redirect=$redirect");
    echo "You are currently not logged in, we are redirecting you, " .
        "be patient!<br>";
```

```
echo "(If your browser doesn't support this, " .  
    "<a href=\"admin_login.php?redirect=$redirect\">click here</a>);"  
die();  
}  
?>
```

## APPENDIX C

### Upload and Download PHP Codes and MySQL Queries

#### auth.php

```
<?php
session_start();
if ((isset($_SESSION['user_logged']) &&
    $_SESSION['user_logged'] != "") || (isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['user_password']) &&
    $_SESSION['user_password'] != "") || (isset($_SESSION['admin_password'])
&&
    $_SESSION['admin_password'] != "")) {
    //Do Nothing!
} else {
    $redirect = $_SERVER['PHP_SELF'];
    header("Refresh: 5; URL=admin_login.php?redirect=$redirect");
    echo "You are currently not logged in, we are redirecting you, " .
        "be patient!<br>";
    echo "(If your browser doesn't support this, " .
        "<a href=\"admin_login.php?redirect=$redirect\">click here</a>)" .
        die();
}
?>
```

#### delete\_content.php

```
<?php
$connect=mysql_connect("localhost","root","com400")
or die (mysql_error());
$db_select=mysql_select_db("registration")
or die (mysql_error());
```



```

session_start();
if ((isset($_SESSION['admin_logged']) && $_SESSION['admin_level'] == "1"
&&
    $_SESSION['admin_logged']) != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
if (isset($_POST['submit']) && $_POST['submit'] == "Yes") {
    $query_delete = "DELETE FROM upload " .
        "WHERE id = " . $_POST['id'] . """;
    $result_delete = mysql_query($query_delete)
    or die(mysql_error());
    header("Refresh: 5; URL=download.php");
    echo "The file has been successfully removed from the database! " .
        "You are being sent to the download area!<br>";
    echo "(If your browser doesn't support this, " .
        "<a href='\"download.php\"'>click here</a>";
    die();
} else {
?>
<html>
<head>
<title>Delete Contents From the Database</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 18px; font:bold;}
h2{ margin-top:7%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
a {color:#000000;}
hr{ color:#0066FF}
p { text-align:center}
</style>

```

```

</head>
<body>
<h2>Delete file</h2>
<p>
Are you sure you want to <font color="#FF0000"> delete</font> this file ?<br>
You cannot retrieve the file once you click <em>'Yes'</em>&nbsp;&nbsp;&nbsp;!<br>
<form action="delete_content.php" method="post">
  <input type="hidden" name="id" value="<?php echo $_GET['id']; ?>">
  <input type="submit" name="submit" value="Yes"> &nbsp;&nbsp;&nbsp;
  <input type="button" value=" No " onClick="history.go(-1);">
</form>
</p>
</body>
</html>
<?php
}
} else {
?>
  You don't have a high enough privilege to delete this file.<br>
  <a href="download.php">Click here</a> to go back.
<?php
}
?>

```

### download.php

```

<?php session_start();?>
<?php include "auth.php";?>
<?php $connect=mysql_connect("localhost","root","com400")
or die (mysql_error());
$db_select=mysql_select_db("registration")
or die (mysql_error());

```

```

if(isset($_GET['id']))
{
    $id = $_GET['id'];
    $query = "SELECT name, type, size, content,id FROM upload WHERE
id = '$id'";
    $result = mysql_query($query) or die('Error, query failed');
    list($name, $type, $size, $content) = mysql_fetch_array($result);
    echo $content;
}
?>
<html>
<head>
<title>Download Section</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 18px; font:bold;}
h2{ margin-top:5%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
a {color:#000000;}
hr{ color:#0066FF}
</style>
</head>
<body>
<h2>Current Files on the database :</h2>
<center><em><font color="#0066FF">Right click the file name and save as the
target file to your local machine</font></em></center><br>
<center><em><font color="#FF0000">[&nbsp;Note:</font>You can just click
and open a file without the .pdf extension!<font
color="#FF0000">&nbsp;]</font></em></center><br>
<hr/>

```



```

<?
$query = "SELECT id, name FROM upload";
$result = mysql_query($query) or die('Error, query failed');
if(mysql_num_rows($result) == 0)
{
    echo "Database is empty <br>";
}
else
{
    while(list($id, $name) = mysql_fetch_array($result))
    {

?>

        <?php if($_SESSION['admin_level'] == "1" || $_SESSION['admin_level']
== "2" ) {?>
            <ul><li>
                <a href="download.php?id=?=$id;?>">?=$name;?></a> | <a
href="delete_content.php?id=?=$id;?>">Delete File</a></li></ul><hr/>
                <?php }else{?>
                    <ul><li>
                        <a href="download.php?id=?=$id;?>">?=$name;?></a> </li>
                    </ul><hr/>
                <?php }?>
            </ul><hr/>
        }
    }
}
?>

<center><a href="updown.php">Click here to go back</a></center>
</body>
</html>

```

### updown.php

```
<?php
session_start();
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Upload/download section</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 18px; font:bold;}
h2{ margin-top:5%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:20%; margin-right:20%; border-color:#0066FF;
color:#0066FF; text-align:center; margin-top:3%}
a {color:#000000;}
</style>
</head>
<body>
<h2>Welcome to the Admin Upload/Download Section</h2>
<fieldset>
<br />
<a href="upload.php">Click here</a> to upload files to the database.<br /><br />
<a href="download.php">Click here</a> to download files from the database
content.
```

```

<br /><br /></fieldset><br />
<center><a
href="http://localhost/graduationproject/admin/admin_area.php">Click here to go
back</a></center>
</body>
</html>
<?php } else { ?>
<?php session_start();
if ((isset($_SESSION['user_logged']) &&
$_SESSION['user_logged'] != "") ||
(isset($_SESSION['user_password']) &&
$_SESSION['user_password'] != "")) { ?>
<head>
<title>Upload/download section</title>
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 18px; font:bold;}
h2{ margin-top:8%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:20%; margin-right:20%; border-color:#0066FF;
color:#0066FF; text-align:center; margin-top:2%}
a {color:#000000;}
</style>
</head>
<body>
<h2>Welcome to the User Download Section</h2>
<fieldset><br />
<a href="download.php">Click here</a> to download files from the database
content.<br /><br /></fieldset><br />
<center><a href="http://localhost/graduationproject/logged_user.php">Click here
to go back to the user page</a></center>

```



```

</body>
</html>
<?php }} ?>
upload.php
<?php
$conn = mysql_connect("localhost", "root", "com400")
    or die(mysql_error());
$db = mysql_select_db("registration")
    or die(mysql_error());
?>
<?php
session_start();
if ((isset($_SESSION['admin_logged']) &&
    $_SESSION['admin_logged'] != "" ||
    (isset($_SESSION['admin_password']) &&
    $_SESSION['admin_password'] != "")) {
?>
<html>
<head>
<title>Upload Files</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
body{ background-color: #99CCFF; font-family:Arial, Helvetica, sans-serif; font-
size: 18px; font:bold;}
h2{ margin-top:5%;text-align:center; color:#0066FF; border:solid; border-
color:#0066FF; margin-left:20%; margin-right:20%; }
fieldset{border:solid; margin-left:35%; margin-right:60%; border-color:#0066FF;
color:#0066FF; text-align:center; margin-top:3%;}
a {color:#000000;}
#maxsize { margin-top:4%;text-align:center; font:bold; color:#333333; font-
size:20px;}

```

```

.box {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px ;
    border: 3px solid teal;
}
</style>
</head>
<body >
<?
if(isset($_POST['upload']))
{
    $fileName = $_FILES['userfile']['name'];
    $tmpName = $_FILES['userfile']['tmp_name'];
    $fileSize = $_FILES['userfile']['size'];
    $fileType = $_FILES['userfile']['type'];

    $fp = fopen($tmpName, 'r');
    $content = fread($fp, $fileSize);
    $content = addslashes($content);
    fclose($fp);

    if(!get_magic_quotes_gpc())
    {
        $fileName = addslashes($fileName);
    }

    $query = "INSERT INTO upload (name, size, type, content ) ".
        "VALUES ('$fileName', '$fileSize', '$fileType', '$content')";
}

```

```

        mysql_query($query) or die('Error, query failed');
        echo "<br><br><br><center> File $fileName uploaded
successfully and added to the database</center><br>";
    }
    ?>

<h2>Upload files to the database</h2>
<div id="maxsize">The maximum file size is 20 mb!</div>
<fieldset>
<form action="" method="post" enctype="multipart/form-data"
name="uploadform">
    <table align="left" bgcolor="#00CCCC" width="350" border="" cellpadding="3"
cellspacing="2" class="box">
        <tr>
            <td width="250"><input type="hidden" name="MAX_FILE_SIZE"
value="2000000"><input name="userfile" type="file" class="box" id="userfile">
                </td>
            <td width="80"><input name="upload" type="submit" class="box"
id="upload" value=" Upload "></td>
        </tr>
    </table>
</form>
</fieldset>
<center><p><a
href="http://localhost/graduationproject/upload/updown.php">Click here </a>to
go back</p></center>
</body>
</html>

<?php }else { ?>
<p>You are <b>unauthorized </b>, please login with and <b>admin</b>
account!<a href="http://localhost/graduationproject/admin/admin_login.php">
Click here </a> to login!</p><?php } ?>

```