NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

ONLINE WEB APPLICATION

RENT A CAR WEB SITE DESIGN

Graduation Project
COM400

Student: Fırat Silinir(20011433)

Supervisor: Mr. Ümit Soyer

Nicosia-2006

ACKNOWLEDGMENT

My gratitude for the following people cannot be exppress in this short acknowledgment

I wish to thank who made this Project posibble to me, my supervisor Mr. Umit Soyer Esspecially thank to my friends who share their knowledge with me, Murat Silinir, Mahsum Akbaş, and Murat Ölçen

I dedicate this Project to for my family and my home friends because during two months, they cooked washed the dishes , cleaned the house form e and helped corrections of projects, M.Can Dikici, Kamuran Kurtar and M. Veysel Cihangir.

ABSTRACT

Human being is confronting very curicial multy dimensional transformations. İn this complicated procces, terms such as time communication, speed, internet information and transportation have gained very strategic importance. Our economic vision is being shaped with in this genaral perspective.In this framework, we have chosen transformation sector to serve people. This is rent car service. As mentioned, in this global procces, we aimed to serve people in the web. We have thought that whith this choosing, we can arrive all potantial clients easily.

With the this genaral perspective, Project has been formed to serve people on the web line. When this Project was constructin, two elements were not ignored. These are technical infrastructure innovations and marketing methods.We can say that this web site was constructed on the two important pillars. In addition to this, web developing tools and programs such as PHP, Apache, and MySQL were used to constitute web site. Tools and components which used in the this Project have taken up within the analytical framework.

In sum, Project aims to constitute interactiveness, understandable, easiness, easy connection, good governace, good public relations and efficiency by the this web site.In the this framework, home page of web site was tried to be very attractive for both commercial and information operations.

INTRODUCTION

In the Internet world the need for fast and reliable web sites becomes more important and gets more popular and meaningful day after day. And web sites are basic components of the Internet, and they are getting into every field. Therefore here in this project we aim to bring about a web site that meets the very last requirements and consists of basic web-site features.

A web site is location on a computer network that makes information in the form of page or documents available to visitors who connect to the site by using a Web browser

A web browser can be publishes in the form of HTML pages, or in other document formats. To view the information available on a Web site, visitors use Web browser software programs ,like Internet Explorer or Netscape, which translate HTML pages on Web site to text and graphics on their monitors.

A home page is an entry page for a set f web pages. It is a document written in HTML format that might be describe the content available on the site

The primary function of a home page is to introduce visitors to our Web site and help them navigate through its page

Chapter's givess techniques about HTML documents, Javascript, Apache Web Server, PHP and MySQL

Chapter 1 : shows the basic html document and usage of basic html tags and its attributes.

Chapter 2 : describe the Javascript fundamentals and usage with html forms.

Chapter 3 : describe apache Web Server and how configurating

Chapter 4 : tells the PHP basics and objects.

Chapter 5 : gives information about the MySQL and shows the connection with database

Chapter 6 : gives information about designing and used funtions in pages.

Chapter 7 : gives applications codes..

TABLE OF CONTENS

CHAPTER 1 – HYPERTEXT MARKUP LANGUAGE (HTML)

1.1 What is HTML

HTML (Hypertext Markup Language) the authoring language used to create documents on the World Wide Web. HTML defines the structure and layout of a Web document by using a variety of tags and attributes. All the information you'd like to include in your Web page fits in between the tags, and is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element. Some elements come in pairs that indicate when some display effect is to begin and when it is to end. HTML is a formal Recommendation by the World Wide Web Consortium (W3C) and is generally adhered to by the major browsers.

1.2 What is an HTML File?

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

```
<html>
<head>
<title>Title of page</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

The first tag in your HTML document is <html>. This tag tells your browser that this is the start of an HTML document. The last tag in your document is </html>. This tag tells your browser that this is the end of the HTML document.
The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.
The text between the <title> tags is the title of your document. The title is displayed in your browser's caption.
The text between the <body> tags is the text that will be displayed in your browser.
The text between the <b> and </b> tags will be displayed in a bold font.

1.2.1 HTM or HTML Extension?

When you save an HTML file, you can use either the .htm or the .html extension. We have used .htm in our examples. It might be a bad habit inherited from the past when some of the commonly used software only allowed three letter extensions.
With newer software we think it will be perfectly safe to use .html.

1.2.2 Headings

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

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

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

1.2.3 Paragraphs

Paragraphs are defined with the <p> tag.

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

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

1.2.4 Line Breaks

The <br> tag is used when you want to end a line, but don't want to start a new paragraph. The <br> tag forces a line break wherever you place it.

```
<p>This <br> is a para<br>graph with line breaks</p>
```

The <br> tag is an empty tag. It has no closing tag.

1.2.5 Comments in HTML

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

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

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

1.3 HTML Elements

This is an HTML element:

```
<b>This text is bold</b>
```

The HTML element starts with a start tag: <b>
The content of the HTML element is: This text is bold
The HTML element ends with an end tag: </b>
The purpose of the <b> tag is to define an HTML element that should be displayed as bold.
This is also an HTML element:

```
<body>
This is my first homepage. <b>This text is bold</b>
</body>
```

This HTML element starts with the start tag <body>, and ends with the end tag </body>.
The purpose of the <body> tag is to define the HTML element that contains the body of the HTML document.

1.3.1 Why do We Use Lowercase Tags?

We have just said that HTML tags are not case sensitive: <B> means the same as <b>.
When you surf the Web, you will notice that most tutorials use uppercase HTML tags in their examples. We always use lowercase tags. Why?
If you want to prepare yourself for the next generations of HTML, you should start using lowercase tags. The World Wide Web Consortium (W3C) recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) demands lowercase tags.

1.4 Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page.
This tag defines the body element of your HTML page: <body>. With an added bgcolor attribute, you can tell the browser that the background color of your page should be red, like this: <body bgcolor="red">.
This tag defines an HTML table: <table>. With an added border attribute, you can tell the browser that the table should have no borders: <table border="0">
Attributes always come in name/value pairs like this: name="value".
Attributes are always added to the start tag of an HTML element.

1.4.1 Quote Styles, "red" or 'red' ?

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed.
In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:
name='John "ShotGun" Nelson'

CHAPTER TWO : JAVASCRIPT

2.1 Introduction

JavaScript has changed from being a language which improves web sites to a language which destroys them. This is because there are huge JavaScript sites which have thousands of scripts for download. These usually involve things which do not benefit a website at all, like status bar effects and scrolling text which do not add much to a website.

JavaScript must not be confused with Java. Java is a completely different programming language. It is usually used for text effects and games, although there are some JavaScript games around.

So why should you use JavaScript? Well, JavaScript can allow you to create new things on your website that are both dynamic and interactive, allowing you to do things like find out some information about a user (like monitor resolution and browser), check that forms have been filled in correctly, rotate images, make random text, do calculations and many other things.

2.2 Declaring JavaScript

Adding JavaScript to a web page is actually surprisingly easy! To add a JavaScript all you need to add is the following (either between the <head></head> tags or between the <body></body> tags - I will explain more about this later):

```
<script language="JavaScript">

JavaScript

</script>
```

As you can see the JavaScript is just contained in a normal HTML tag set. The next thing you must do is make sure that the older browsers ignore it. If you don't do this the code for the JavaScript will be shown which will look awful.

There are two things you must do to hide the code from older browsers and show something instead:

```
<script language="JavaScript">
<!--Begin Hide

JavaScript
```

```
// End Hide-->
</script>
<noscript>

HTML Code

</noscript>
```

As you can see this makes the code look a lot more complex, but it is really quite simple. If you look closely you can see that all that has been done is that the JavaScript is now contained in an HTML comment tag. This is so that any old browsers which do not understand <script> will just think it is an HTML comment and ignore it.

Because of this the <noscript> tag was created. This will be ignored by browsers which understand <script> but will be read by the older browsers. All you need to do is put between <noscript> and </noscript> what you want to appear if the browser does not support JavaScript. This could be an alternative feature or just a message saying it is not available. You do not have to include the tag if you don't want anything shown instead.

2.2.1 Alerts

alert()

This command will make a popup box appear . This can be useful for warning users about things or (in some cases) giving them instructions. It is used in the following way:

```
alert('Text for alert box');
```

In the example above I have used single quotations ' but you could use double quotations if you want to ". They work exactly the same way. The reason I use single quotes is because, later on, when you are using HTML code and JavaScript together you will need to use them and it is good to be consistent.

Here is the full JavaScript for the earlier example:

```
<script>
<!-- Start Hide

// Display the alert box
alert('This text is in an alert box');

// End Hide-->
```

```
</script>
```

This is placed between the <head> and </head> tags of the page. As you can see, I have used a comment tag as well as the alert box code. This makes your code more readable but is not essential.

2.2.2 Variables

Variables in JavaScript, as in other computer languages, can be used to store information. For example we could tell the computer that the variable called:

my_number

should have the value:

3456

Variables can also contain text for example the variable:

my_name

could have the value:

David Gowans

Variables can be very useful for text or numbers that you repeat several times in a program, for doing calculations or for getting input from a user. Variables are declared as follows:

Number:

```
var my_number = 3456;
```

Strings (text):

```
var my_name = 'David Gowans';
```

As you can see a string is included in quotes (either single or double) but a number does not. If you include a number in quotes it will not be treated as a number. You may also notice that each line ends with a semicolon. This is standard JavaScript code to show the end of a line. Always remember to put it in.

When naming your strings you can use any word or combination of words as long as it is not already in use by JavaScript (so don't use alert as a variable name etc.). Do not include spaces, replace them with _.

2.2.3 Getting Information From The User

Once you have started using variables you will realize that it will be quite useful to get some information from the user. You can do this by using the:

2.2.4 prompt()

First of all, the new prompt command is used. I set the variable your_n
ame using it:

```
var your_name = prompt('Please enter your name', 'Your Name');
```

The text between the first set of quotes is what is written on the prompt box. The text between the second set of quotes is what is the default text for the input section of the box.

```
var output_text = welcome_text + your_name + closing_text;
```

As you can see this is much the same as adding 3 numbers together but, as these are strings they will be put one after the other (you could have also used quotes in here to add text and strings together). This added the text I had set as the welcome_text to the input I had received and then put the closing_text on the end.

Finally I displayed the output_text variable in an alert box with the following code:

```
alert(output_text);
```

which, instead of having text defined as the content for the alert box, places the string in the box.

2.2.5 document.writeln

This command is very useful as it will output information to a web page. I will start with a basic example of how to use it:

```
document.writeln('Hello there!');
```

This basically tells the JavaScript to write to the document (web page) on a new line the text Hello there!. The really useful bit of this is that you can tell the JavaScript to output text, HTML and variables. First of all I will show you how to output HTML:

```
document.writeln('<font face="Arial" size="5" color="red">Hello there!</font>');
```

This will display the following on the page:

Hello there!

As you can see, this allows you to just put standard HTML code into a web page using JavaScript. If you can't see a good reason for this it is not surprising at the moment but next I will introduce variables to the script.

2.3 Remote JavaScript

Now you have learnt how to use the document.writeln command you can now start using one of the most useful applications of JavaScript, remote scripting. This allows you to write a JavaScript in a file which can then be 'called' from any other page on the web.

This can be used for things on your own site which you may like to update site-wide (like a footer on the bottom of every page) or something used on remote sites (for e xample newsfeed or some counters).

To insert a remote JavaScript you do the following:

```
<script language="JavaScript" src="http://www.yourdomain.com/javascript.js">
</script>
```

Which will then run the JavaScript stored at http://www.yourdomain.com/javascript.js. The .js file is also very simple to make, all you have to do is write your JavaScript (omitting the <script>, </script>, <!--Start Hide and // End Hide--> tags into a simple text file and save it as something.js.

If you want to include information for browsers which do not support JavaScript you will need to put the <noscript></noscript> tags in the HTML, not the JavaScript file.

There is one problem with using remote JavaScript which is that only the recent browsers support it. Some browsers which support JavaScript do not support Remote JavaScript. This makes it advisable not to use this for navigation bars and important parts of your site.

2.4 Link Events

A link event is a different way of including JavaScript on your page. Instead of having <script> tags in your HTML you can set JavaScript that will only be executed when certain things happen.

There are three ways of executing some JavaScript code in a link. They are:

onClick
onMouseOver
onMouseOut

They can have many different uses but the most common is for image swaps (mouseover images).

2.4.1 onClick

onClick works in exactly the same way as a standard HTML link. It is executed when someone clicks on a link in a page. It is inserted as follows:

```
<a href="#" onClick="JavaScript Code">Click Here</a>
```

As you can see, one main difference is that the href of the link points to a #. This is nothing to do with the JavaScript, it just neabs that, instead of opening a new page, the link will not do anything. You could, of course, include a link in here and you would be able to open a new page AND execute some code at the same time. This can be useful if you want to change the content of more than one browser window or frame at the same time.

2.4.2 onMouseOver and onMouseOut

onMouseOver and onMouseOut work in exactly the same way as onClick except for one major difference with each.

onMouseOver, as the name suggests, will execut the code when the mouse passes over the link. The onMouseOver will execute a piece of code when the mouse moves away from the link. They are used in exactly the same way as onClick.

2.5 Rollover Images

This is one of the main ways of using link events. If you have not seen rollover images before, they are images (usually used on navigation bars like the one at the top of this page) and when the mouse moves over the link it changes the image which is displayed.

This is done using a combination of the onMouseOver and onMouseOut events. To explain - when the mouse passes over the link you want the image to change to the new image, when it moves away from the link, the old picture should be displayed again.

The first thing you must do is edit the <img> tag you use to insert the image you want to change. Instead of just having something like this:

```
<img src="home.gif" alt="Home">
```

you would have the following:

```
<img src="home.gif" alt="Home" name="home_button">
```

The name for the image could be anything and, like the window names from the last part, is used to refer to the image later on.

Now you have given a name to the image you are using you will want to create the rollover. The first thing you must do is create the image for the rollover and save it. Then create a link round the image. If you don't want to have a link on the image you can still do a rollover by specifying the href as # which will make the link do nothing eg:

```
<a href="#"></a>
```

The following code will make a mouseover on your image (assuming you inserted the image as shown above and called your mouseover image homeon.gif):

```
<a href="index.htm" onMouseOver="home_button.src='homeon.gif';"
onMouseOut="home_button.src='home.gif';"><img src="home.gif" alt="Home"
name="home_button" border="0"></a>
```

Firstly you are creating a standard link to index.htm. Then you are telling the browser that when the mouse moves over the link the image with the name home_button will change to homeon.gif. Then you are telling it that when the mouse moves away from the link to change the image called home_button to home.gif. Finally you are inserting an image called home_button with an alt tag of 'Home' and no border round it.

onClick, onMouseOver and onMouseOut can be used with text links as well as images in exactly the same way as you did above. This, of course, means that you can create interesting effects by, when the mouse moves over an image, another image changes. This could be very useful for placing a description of a link on a page.

2.6 Status Bar

The status bar is the grey bar along the bottom of a web browser where information like, how much the page has loaded and the URL which a link is pointing to appears. You can make your own text apppear in the status bar using the JavaScript you already know using the following code:

```
window.status='Your Text In Here';
```

You can include this in standard code, onClick, onMouseOver or onMouseOut. This allows you to do things like display a description of a link when the mouse moves over it.

CHAPTER THREE : APACHE WEB SERVER

3.1 The WWW

The Internet has been around for a long time. More than 30 years now. But for most of that time, it was entirely the domain of geeks and hobbyists. The main reason for this was that it was hard to use.
In 1991, Tim Berners-Lee developed something that he called the World Wide Web, while working at CERN. His purpose was to give quick and easy access to documents for geographically distributed people collaborating on projects. Along with a lot of help from the standards community (and, notably, Roy Fielding), they defined HTTP, HTML, URLs, and the other necessary components of making the Web a reality. He then went off, and with the help of colleagues around the world, communicating via email, developed the CERN web server, and a simple Web client, which he dubbed a ``browser." The name came about because there was very little of real value on the Web at that time, and all you ever really did was browse. Ironic that the name stuck!

3.1.1 NCSA

As more and more people got involved in the project, it was several Universities that contributed to the project the most. From very early on, one of the front-runners was the National Center for Supercomupting Activities (NCSA) at the University of Illinois at Urbana Champaign (UIUC). NCSA started working on the NCSA HTTPd (HyperText Transfer Protocol Daemon).
Rob McCool wrote the original code for the NCSA HTTPd, and this code was distributed without charge to the community, for them to use, with the understanding that if they fixed bugs, or added features, that they would then contribute them back to Rob to put into future versions.

3.2 Introduction - What is Apache

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

3.2.1 The Apache Server

When Rob left the project, it left a problem. There were still a lot of people using his code, and actively making patches to the code, but there was no longer anyone collecting those patches.
In 1995, Brian Behlendorf and a small group of other developers started collecting these patches in a central repository. Brian got some space donated on a server, and set up a CVS tree so that developers could check in patches. And in April of 1995, they released the first official release (Version 0.6.2), which was given the name Apache, because it was ``a patchy server''.
The Apache Group, as they were known at that time, had no formal organizational structure, never met, communicated only over email, and worked entirely in their free time, on a volunteer basis. Early the next year, Apache passed NCSA as the most widely used server on the Internet, and is now used on more than 60% of all web servers on the Internet.

3.2.2 Apache's Architecture

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

3.2.3 More Recent History

In the last few years, Open Source has been getting a lot of press, because of Linux, Perl, and Apache. In 1998, IBM decided to abandon development of a web server engine to go into WebSphere - an application server for the web - and use Apache instead. This decision, along with Netscape's decision to release the source code for the Netscape browser, earlier that same year, showed the business world that Open Source was more than just a lot of long-haired anti-establishment types out to bring down the software industry, but that it was actually a good business model. It produces code more quickly, and that code is more reliable, because, in the words of Eric Raymond, with enough eyes, all bugs are shallow.
In June of 1999, The Apache Software Foundation was officially incorporated in the state of Delaware. The ASF has a much broader mission than just the Apache HTTP server, and has several other projects that exist under the larger umbrella of the ASF
The stated goals of the ASF are:
provide a foundation for open, collaborative software development projects by supplying hardware, communication, and business infrastructure;
create an independent legal entity to which companies and individuals can donate resources and be assured that those resources will be used for the public benefit;
provide a means for individual volunteers to be sheltered from legal suits directed at the Foundation's projects; and,

protect the 'Apache' brand, as applied to its software products, from being abused by other organizations.
Some of the better-known projects under the ASF are the Apache web server, mod_perl, mod_php, and Jakarta.

3.2.4 The Future of Apache

At ApacheCon in Orlando, back in March, Apache 2.0 was released. This is largely a rewrite from earlier versions, and uses a threading model that will increase performance substantially on most platforms. As of this writing, version Alpha 6 of the 2.0 server has been released.
The Apache Group, as mentioned above, has become the Apache Software Foundation, and continues to take on new projects that seem to fit the larger vision that the ASF has for the future. Open Source, and open standards, produce better software. In the end, this makes life better for all of us, and we should support the ASF in all its endeavors, if only for purely selfish reasons.

3.2.5 Support for Apache

As an Open Source software product, Apache falls prey to the myth of no support. There are two main ways to obtain support for Apache. First, there's the traditional email and Usenet methods. There are a variety of mailing lists on which you can obtain support for Apache. And there are two main Usenet groups - comp.infosystems.www.servers.unix and, for those running Apache on Windows, comp.infosystems.www.servers.mswindows
You can find information about the ``official'' Apache mailing lists on the Apache.org web site.
Secondly, there's also commercial support for Apache, available through Covalent Technologies. Covalent offers support contracts for Apache, and they also have add-on products for Apache, such as Raven SSL. And the author of Comanche (which we'll discuss later) works at Covalent.

3.3 Obtaining and Installing Apache

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

3.3.1 Hardware/Software Requirements

Apache runs on anything. Almost. It will almost certainly run on whatever you have. I've run Apache on a 386 with 4 MB of RAM. And I've run it on a 4-processor machine with 1GB of RAM. It was happy both places. The Apache.org web site does not list any hardware requirements. It will run on any hardware that runs the supported operating systems.

Apache will run on any flavor of *nix, and also on Microsoft Windows (95, 98, NT, 2000), Mac, and OS/2.

Compiling and installing

Most of the settings for your server, governing how it will operate, are done at the configuration stage, when you modify the configuration files that the server loads when it starts up. However, due to the modular architecture of Apache, a lot also depends on what modules you enable when you compile the server. The available configuration directives depend on the modules that are loaded.

You can either compile your server the quick, easy way, and get a default installation with the most common functionality, or you can get in there and pick and choose what you actually want.

The simple way

The installation process for Apache is really simple for most folks. If you are just wanting to set up a simple web site to do the normal things like serve web pages, and maybe do some CGI, the installation process looks like this:

```
tar -zxf apache_1.3.12.tar.gz
cd apache_1.3.12
./configure --prefix=/usr/local/apache
make
make install
/usr/local/apache/bin/apachectl start
```

Assuming you have a reasonably fast machine, this entire process does not take much more than 10 or 15 minutes, and you have a functioning web site. The configure process figures out reasonable settings for your system, and so the configuration files will have reasonable things in them so that you can immediately start serving web pages from your server. The --prefix setting tells the configure process where you want to install the server. /usr/local/apache is the normal place to do this, but if you want to put it somewhere else, just specify that on the command line:

```
./configure --prefix=/home/rbowen/devserver
```

apachectl is a handy tool that Apache installs to make it simple to start, stop, and restart the server, as well as some other handy functionality. More about this a little later.

3.3.2 Advanced Installation

If you're like me, you are not satisfied with the default installation. It does not have all the modules that I want, it has some stuff that I'd just as soon leave out, and there are some things I just do differently because I do them differently. I'm strange that way. There are two ways to handle this that I'm going to talk about. First, you can actually edit the configuration file, and specifically choose what you want to compile into the server. Or, you can just throw everything in, but do it in such a way that you can go back and add and remove stuff at your leisure. I tend to go with the latter approach, but the former approach gets more coverage in the docs, and so is used more frequently.

You are advised to use the simple method above the first few times you install Apache. Also, in version 2.0, there will only be one installation method, and it will look more like the quick easy method above, than like these methods here.

3.3.3 Editing The Configuration Scripts

In our previous example, we ran a script called configure (``small-c configure") in the main Apache directory. In this method, we're going to go down into the src/ directory and actually look at the configuration files. After all, the motto of Linux is ``do it yourself."

The process starts out the same:

```
tar -zxf apache_1.3.12.tar.gz
```

But rather than just going into the apache_1.3.12 directory, you need to go down into the src directory:

```
cd apache_1.3.12/src
```

Then, using your favorite editor, edit the file Configuration (``big-C Configuration") and comment, or uncomment, the lines that refer to options that you are interested in.

If you don't see a file called Configuration, copy the file Configuration.tmpl to Configuration, and use that as your template.

Once you have gone through and made sure that you have everything that you want in there, save the file, and run the following:

```
./Configure
make
make install
```

The simple method, which we talked about first, is doing all of this for you behind the scenes. If you run ``small-c" configuration, you will have a file called Configuration.apaci created for you, which will then get used in this configuration step.

3.3.4 Dynamic Shared Objects (DSO's)

I get tired of rebuilding and reinstalling my web server every time I want to add in a new module, or when I decide to take one out, because I never really use it. This is where shared objects are handy. A shared object is something that gets loaded dynamically by a process when it needs it. This saves you from having to compile that code into the program executable, which, in turn, makes the executable smaller, and load up faster. By making your Apache modules into shared objects, you can build everything into your server, but only actually use the parts that you want at any one time, and leave out everything else that you're not using.

On Windows, these are things called ``dynamic link libraries", or DLLs. On Linux, they are called shared objects, or .so files.

In order to enable shared objects, you have to compile a module called mod_so, which, in turn, loads all the modules that you have compiled as shared objects. mod_so itself cannot be shared object, of course, because there would be no way to load it. Chicken, egg.

So, to build your Apache server to use shared objects, run the following commands:

```
./configure --prefix=/path/to/apache \
            --enable-module=most \
            --enable-shared=max
```

(You can either literally type those \ characters, or just put this command all on one line and omit the \'s)

What does this command do? Well, it compiles all of the modules that ship with Apache, except those that are considered experimental or unstable, and enables them all as DSO's. This means that Apache will be loading up a bunch of modules that you don't really want, so you need to edit the configuration file and comment out those modules that you're not really going to use.
But it also means that if you want to add in a particular module, you can do so by putting it in the configuration file, rather than having to recompile your server from source. This is particularly handy if you change your server configuration a lot, or when you are testing out different configurations to see what it is that you want.
A server that is loading all the modules dynamically, rather than having those modules compiled in, takes a little longer to start up, but this penalty is paid only at server start, and after that the servers run at the same speed. That is, a server running modules as DSO's does not run any slower.

3.4 Configuring

Once you've compiled and installed your server, you need to configure it for your particular environment. Many of the configuration directives got set when you ran configure (or Configure), and so the server should work correctly immediately. However, you will probably want to change some things, since the default installation is very generic, and not precisely suited to your needs.
Apache, unlike most of its competitors in the web server market, lets you configure everything, down to the smallest detail. And if there's really something that you want to configure that you can't, you have the source code, so you can change it if you are so inclined.

3.4.1 Configuration Files

The configuration for your Apache server is located in a file called httpd.conf, which is usually located at /usr/local/apache/conf/httpd.conf.
Note that if you installed Apache with a RPM (don't do that!) then the files will be in bizarre places that have no relation to logic. Uninstall the RPM, and install from source. It's a simple process, and reduces your pain in the long run.
Note: I made a comment like the above in one of my articles on ApacheToday.com, and got thoroughly chastized for it by some Red Hat fanatics that read the article. While I found their comments, and their reasons for their comments, to be rather amusing, I should emphasize that this is just my opinion, and should not be taken as some sort of transcendent truth. If you really want to spread your files all over your file system, go right ahead. You might notice, however, that a default installation of Apache puts everthing in /usr/local/apache, so it's a safe bet that the Apache developers agree with me on this one.
The format of httpd.conf is very simple.
There are comments, which consist of a hash sign (#) at the beginning of a line:

```
# Based upon the NCSA server configuration files originally by Rob McCool.
```

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

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

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

```
    <Directory /usr/local/apache/cgi-bin>
        AllowOverride None
    </Directory>
```

Sections can contain directives, and those directives apply to the the resources defined by the container definition. In the above example, the AllowOverride directive will apply to files located in the specified directory.
You can edit these configuration files with your favorite text editor. You need to restart the server when you are done editing the configuration files in order for the new configuration to take effect.
You can use the apachectl script to test your configuration file to make sure that you did not make any errors.

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

More about this below.

3.4.2 Comanche

One of the battles that *nix continually has to fight is the notion that it's hard to use. Much of this notion comes from the fact that everything you want to use on *nix has a configuration file, and every configuration file has a different format. Learning all these different formats is a pain, and it's so easy to get it wrong. Sendmail is one of the worst offenders in this arena, but even something as simple as Apache gets difficult to configure. Its modular architecture means that it can be extended forever, and every extension has its own configuration directives. This can be a little overwhelming.
Daniel Lopez took on this problem as his Master's thesis, and developed Comanche - the Configuration Manager for Apache. Comanche is a graphical configuration tool, written in Tcl, which lets you configure Apache in a more intuitive interface. It tells you what each directive means, and asks you questions that make sense. Your answers are put back into the configuration files in a format that Apache can understand.
Comanche can also be used to configure other applications, such as Samba, which have text configuration flies. There is not yet a plug-in for configuring Sendmail, but this is something that Daniel is frequently asked for, so perhaps there will be some day. And you can write your own extensions to Comanche to configure anything that has a text configuration file.

3.4.3 Starting, Stopping, Restarting

There are a variety of ways to control your Apache server. We'll focus on a script that ships with Apache, called apachectl, which does a few other things in addition to just starting, stopping, and restarting.
apachectl
apachectl, which presumably stands for ``Apache control", is located in the bin directory of your Apache installation. It is a shell script which does many of the things that you'll want to do in controlling your Apache server. It can be run with any of the following arguments:
start

Starts the server.
stop
Stops the server.
restart
Restarts the server, if running, by sending a SIGHUP. If the server is not running, starts it.
fullstatus
Displays the full status of the server. Requires that mod_status is enabled, and that lynx is installed.
status
Displays a brief status report for the server. Requires that mod_status is enables, and that lynx is installed.
graceful
Does a graceful restart by sending a SIGUSR1, if the server is running. If the server is not running, it will start it. A graceful restart has the advantage over a simple restart in that child processes that are currently serving content will be permitted to complete their current connection before they are killed.
configtest
Reads the configuration file and parses it for syntax errors.
help
Displays usage information about the apachectl script.

Starting your Apache server on system restart
Linux has a process for starting processes on system startup. This consists of a directory /etc/rc.d containing scripts for each of the processes that you want to start.
If you place a file in /etc/rc.d, called rd.httpd, it will be run on server startup. rc.httpd should contain the following command:

```
    /usr/local/apache/bin/apachectl start
```

If you're running Red Hat, or Mandrake, or one of the other Linuxes that looke like them, you'll find that there are a number of subdirectories of /etc/rc.d that look like rc2.d, rc3.d, and so on, which contain the startup scripts for all your various services. Actually, symlinks to them. On these systems you should create a file at /etc/rc.d/init.d/httpd, containing the command above. You should then create links to it from the directories rc3.d and rc5.d. Each of those directories corresponds to a runlevel. You'll usually be in either runlevel 3 or 5, so that's when you want to start Apache.

3.5 Integrating Apache With The Rest of Your business

The common wisdom is that every company needs a web site, because every company needs a web site. And so lots of companies have web sites which are nothing more than a electronic sales brochure.
With more and more people online every day, many users will expect to get just as good service from your web site as they would in person, or over the phone. In fact, the expectation is often higher. After all, this is a computer. They should be able to get direct access to the answers that they need, and it should be instantaneous.

3.5.1 CGI

The most common way to tie your web site to you database or other processing is with CGI programs. The Common Gateway Interface is a protocol to let your web server serve dynamically generated content from some process running on your server.
Of course, I've oversimplified it in that statement, but that's probably sufficient for our purposes.
A CGI program is a program written in any language you like, which formats its output in a certain way so that your browser can understand it. This allows you to write programs to put any of your existing databases onto your web site, and interact with your online customers in realtime, directly on your web site. You can let the customer customize their experience of your web site.
There are a plethora of good CGI tutorials on the web. (and even more bad ones!) But the basic concepts are pretty simple

3.5.2 MIME Headers

Any output that your CGI program produces must be preceeded by a MIME header that tells the client (the browser) what sort of output they are receiving. This will look something like:

```
content-type: text/html
```

HTTP headers are followed by a blank line, which is how the client knows that the headers are done, and the next things that it sees are the body of the document. If you were to write a CGI program in Perl, for example, this would look like:

```
print "content-type: text/html\r\n\r\n";
```

\r\n is called a crlf, which is short for ``carriage return line feed." Frequently, you will see CGI programs that have just \n, rather than both, but it is more correct to use both. This used to be more of a big deal that it is now. Most web browsers are quite happy to accept one or the other.

3.5.3 Reading Client Input

Input from the client comes in to your program on STDIN. This means that you can read client input as though it was coming from the command line or from the keyboard.
Of course, most languages that you are likely to use for CGI programming have libraries readily available that will handle most of the mundane details of CGI programmig for you, and leave you to do your work.
For example, in Perl, there is the CGI.pm module, and a few others, such as the CGI_Lite.pm module, that handle such things as reading form input and managing cookies, and in the case of CGI.pm generating your headers and output
In C, there are libraries available from Tom Boutell at Boutell.com which provide similar functionality from C.
Perl is the language of choice for CGI programming, because it is very conducive to the sort of rapid prototyping and development that is often demanded by the web.
Output in HTML (usually)

Since your output is going to a browser, you will almost always want to have your output in HTML. Occasionally, you'll want your output to be a gif image, or plain text.
Example CGI program
The following is an example CGI program written in Perl. It does not actually do anything useful, but it gives you an idea of what is the minimum necessary requirement for a CGI program.

```
    #!/usr/bin/perl
    print "content-type: text/html\r\n\r\n";
    print "<b>Hello, World!</b>";
```

Not very exciting, is it?
Rather than provide a lot of example CGI programs, I'd encourage you to look at all the resources at the end of this paper for examples.

3.5.4 mod_perl

Something that you will eventually discover when using CGI is that it is slow. This has nothing to do with the quality of the code that you write, but is intrinsic to CGI. The problem is that every time a client requests a resource that involves running a CGI program, Apache has to launch that program. That takes a lot of time. This is the case whether the program is a Perl script or a compiled binary executable. Almost all of your time will be spent in the startup of that program, not in the actual run time of the program. There are programs with intensive database access, where this will not be the case, but they are in the minority.
There are a number of different technologies that address this problem. Most of them involve having your CGI programs somehow cached, so that when they are invoked, you don't have to pay that startup time, because they are already there in memory, ready to go. Perhaps the most popular of these solutions is mod_perl. mod_perl is an Apache module that significantly enhances the performance of Perl CGI programs. It has other benefits, such as the ability to write Apache modules in Perl, but it is primarily used as a CGI enhancer.
Your Perl CGI programs are compiled, and kept in memory, so that every time the resource is requested, there is no time spent launching the Perl interpreter, or loading your program from disk.
mod_perl is extremely memory-intensive, since all this code is stored in memory. But the performance enhancements are several orders of magnitude, producing a very noticeable speed increase.

3.5.5 SSL and E-Commerce

Even better than telling your customers about your business, is actually doing business online. You can put your catalog online, and let customers order, and pay for, your merchandise directly on your web site.
The one problem is that people are rather picky about who they give their credit card information to. And they are extrememly reluctant to type it in and submit it across the Internet without some assurance that what they are doing is secure.

By default, data that is passed to web servers is ``in the clear'', meaning that it is not encrypted, and anyone that is watching the wire would be able to see anything that went past. Like, for example, your credit card number. Even when you are in a password protected area on a web site, the username and password, and any data exchanged, is all passed in the clear.
One way around this is SSL. SSL, which stands for Secure Socket Layers, is a technology that encrypts traffic between the server and the client using a private key/public key technique. That means that only the people on the two ends can understand it. And even if someone were to intercept the entire message, they would not be able to decrypt it.
There are a number of SSL implementations that run on top of Apache. Two of the better known ones are Raven, from Covalent (http://www.covalent.com/raven/ssl/), and Stronghold (http://www.c2.net/products/sh2/). These are both commercial products.
The Open Source alternative is mod_ssl (http://www.modssl.org/), which runs in conjunction with OpenSSL (http://www.openssl.org).
With the recent changes in the crypto laws, there's a good chance that mod_ssl will ship as one of the standard Apache modules in future releases.
For more information on SSL, you should attend one of the SSL talks here at ApacheCon.

3.5.6 Authentication

Authentication is the process of verifying that you are who you say you are. This is usually accomplished by requesting some variety of username and password. There are a number of different implementations of this for use with Apache.

3.5.7 mod_auth

The ``standard'' Authorization technique is to use HTTP authentication provided by the Apache module called mod_auth. mod_auth is part of a standard installation of Apache, and is turned on by default.
To enable authentication for a particular directory, you need to do several things.
Create a password file
Using the htpasswd utility that comes with Apache, you need to create a password file, which tells apache what password is required for what username, in order to get to the resources in question.
The help for htpasswd says the following:

```
 Usage:
  htpasswd [-cmdps] passwordfile username
  htpasswd -b[cmdps] passwordfile username password
  -c  Create a new file.
  -m  Force MD5 encryption of the password.
  -d  Force CRYPT encryption of the password (default).
  -p  Do not encrypt the password (plaintext).
  -s  Force SHA encryption of the password.
  -b  Use the password from the command line rather than prompting for it.
  On Windows and TPF systems the '-m' flag is used by default.
```

On all other systems, the '-p' flag will probably not work.
The htpasswd utility is (usually) located in /usr/local/apache/bin.
So, for example, to create a new password file, you would type:

```
htpasswd -c htpasswd rbowen
```

You will then be asked for the password that you want that user to have, and then you'll be asked to type it again to confirm it.
To add a password to an existing file, type the same command, but without the -c.
Create a group file
If you want to allow more than one user to have access to a particular resource, you can create a group of users. This is done by creating a group file which lists group names and the members in those groups. A line in the group file might look like this:

```
TCG: rbowen sungo chad tom
```

Put your files somewhere safe
You should store these files (the password file and the group file) somewhere outside of the document directory, so that they cannot be downloaded for leisurely off-line hacking.
Create a .htaccess file pointing at these files
In the directory that you want to protect, create a file called .htaccess, containing something like the following:

```
AuthName "Members Only"
AuthType Basic
AuthUserFile /path/to/htpasswd
AuthGroupFile /path/to/htgroup
Require group TCG
```

The AuthName is the string that appears in the authentication dialog that pops up when you visit a protected area.
AuthType is the method of authentication. It is one of Basic or Digest, but Basic is the only one of these methods that is widely implemented in browsers, so you should probably stick to that.
AuthUserFile and AuthGroupFile refer to the locations of the user file and group files that we created in the steps above.
Require is the directive that tells Apache what user(s) or group(s) can get the content specified. You can Require a particular user, or several users, rather than a particular group:

```
Require user Tom,Dick,Larry
```

The configuration detailed above will protect all files in a particular directory and all subdirectories thereof. You can also protect individual files with a <Files> section. See the documentation for more details.
mod_auth_db, mod_auth_mysql, etc
There are a variety of other modules that allow you to authenticate against usernames and passwords stored in a variety of other places, from DBM files, to MySQL databases, to Oracle databases, to Netware directory services. And a variety of other things. There are modules for authentication against a NT domain, or against Lotus Notes. Check out modules.apache.org for an impressive listing of Apache modules for these and other uses.

3.5.8 Log Files

Apache writes two log files as it runs - the access_log, which keeps a record of every request that your server receives, and the error_log, which keeps track of everything that goes wrong, or other less urgent information, such as server startup, stop, and restarts.
access_log
access_log, by default, is stored in the common log format, which contains the following information:
Address of the client
The address of the remote machine requesting content from your server. This is usually just the IP address, but if you turn HostNameLookups on, this will be, whenever possible, the fully qualified domain name of the client.
ident
The information returned by ident, or other similar lookup. This used to frequently actually contain the email address of the remote user, but this practice stopped as soon as it was realized that people were collecting this information for spam lists.
Username
If the resource requested was password protected, this field will contain the username that was used to gain access.
date/time
The date and time of the request.
Request
The first line of the request that was made to the server
Status
The return code the server returned to the client. 2xx messages mean everything went well. 3xx messages mean that the server redirected the request. 4xx messages mean the user did something wrong. 5xx messages mean that the server did something wrong.
Bytes sent
How many bytes were actually sent to the client.
error_log
The error_log contains errors and various other messages that the server generates during operation. This is particularly useful for troublshooting CGI programs that are not behaving as expected.
Custom log files
With the LogFormat and CustomLog directives, you can create your own log files that contain whatever information you'd like to collect. See the Apache documentation for more details on generating these log formats.
Contributing to the project
Apache is a volunteer-driven project. That means that it relies largely on the users to contribute patches, suggestions, bug reports, and comments.
And big piles of money, of course.
If you think that you have any of the above, and you want to contribute them, here's how to go about it.
Within each project in the Apache Software Foundation, development is completely autonomous from the Foundation as a whole. Each project is left to manage affairs as best

suits that project. Each project has its own web site off of the main www.apache.org web site, and you can usually find information on those sites about contributing code or documentation patches.

3.6 A Conclusion for Apache

Apache is the web server that you need to be using. There's really no question about it. 60% of all web site administrators can't be wrong.
When your web site is becoming such an important, integral part of your business, you really can't afford to be running anything but the best.

CHAPTER FOUR : PERSONAL HOME PAGE (PHP)

4.1 What is PHP?

PHP is a server-side scripting language designed specifically for th web.Within an HTML page,we can embed PHP code that will be executed each time the page is visited.Our PHP code is interpreted at the Web server generates HTML or other output that the visitor will see.

PHP was conceived in 1994 and was originally the work of one man,Rasmus Lerford.It was adopted by other talented people and has gone through there major rewrites to brings us the broad, mature prduct we see today. As of January 2001, it was in use on nearl five million domains wrldwide, and this number is growing rapidly.

PHP is an Open Soure product.You have access to the source code. You can use it,alter it, and redistribute it all without charge.
PHP originally stood for Personal Home Page, but was changed in line with the GNU recursive naming convention (GNU = Gnu's Not Unix) and now stands for PHP Hypertext Processor.

4.2 Writing PHP

Writing PHP on your computer is actually very simple. You don't need any specail software, except for a text editor (like Notepad in Windows). Run this and you are ready to write your first PHP script.

4.2.1 Basic PHP Syntax

A PHP scripting block always starts with <?php and ends with ?>. A PHP scripting block can be placed anywhere in the document.
On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.
However, for maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

```
<?php
?>
```

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.
Below, we have an example of a simple PHP script which sends the text "Hello World" to the browser:

```
<html>
<body>
<?php
echo "Hello World";
?>
</body>
</html>
```

Each code line in PHP must end with a semicolon. The semicolon is a separator and is used to distinguish one set of instructions from another.
There are two basic statements to output text with PHP: echo and print. In the example above we have used the echo statement to output the text "Hello World".

4.2.2 Comments in PHP

In PHP, we use // to make a single-line comment or /* and */ to make a large comment block.

```
<html>
<body>
<?php
//This is a comment
/*
This is
a comment
block
*/
?>
</body>
</html>
```

4.2.3 Declaring PHP

PHP scripts are always enclosed in between two PHP tags. This tells your server to parse the information between them as PHP. The three different forms are as follows:

```
<?
PHP Code In Here
?>

<?php
PHP Code In Here
php?>

<script language="php">
PHP Code In Here
</script>
```

All of these work in exactly the same way but in this tutorial I will be using the first option (<? and ?>). There is no particular reason for this, though, and you can use either of the options. You must remember, though, to start and end your code with the same tag (you can't start with <? and end with </script> for example).

The first PHP script you will be writing is very basic. All it will do is print out all the information about PHP on your server. Type the following code into your text editor:

```
<?
phpinfo();
?>
```

As you can see this actually just one line of code. It is a standard PHP function called phpinfo which will tell the server to print out a standard table of information giving you information on the setup of the server.

This is very important. As with many other scripting and programming languages nearly all lines are ended with a semicolon and if you miss it out you will get an error.

Almost all of this is instantly recognizable to you as plain old HTML code -- all except that one curious line in the middle: <?php phpinfo(); ?> That's the PHP code.
A PHP code snippet is surrounded by an opening and a closing delimiter. The opening delimiter is <?php and the closing is ?> the code in between is PHP code. In this example we are calling the PHP function "phpinfo". The function name is followed by opening and closing parentheses which surround any parameters being passed to the function -- in this case there are none. The function call is terminated with a semicolon. Go ahead and create a file containing this code -- the file name should end with a .php suffix -- send it up to your server and pull it up in your web browser just like you would any other web page. you should see lots of interesting info about the server system and PHP itself. Take a look at the source of the resulting page (View Source). You'll notice that the HTML from your original page is still there, but the PHP code has been replaced by all the information you see. That's basically how the engine works. The PHP code you write is replace in the resulting web page by the results of running the PHP code. Nobody gets to see your actual PHP code -- only the result of it running. That's significant, as I imagine you already realize!
For a little further note about PHP syntax; the code snippet above, contained between the beginning and ending delimiters is similar to HTML tags in appearance. It is in fact called a PHP tag. There are four forms of PHP tags. The one we've used here is called an XML style tag and is the one we will continue to use throughout this tutorial. It is the most common form. another style is called script style and will also look somewhat familiar to you if you have written any other script code such as JavaScript. This example is exactly equivalent to our snippet above:

```
<script language=php>
phpinfo():
</script>
```

The other two forms are called the short style of tag and the ASP style. Both of these require special settings in the PHP configuration files. They are less commonly used and I will only provide this one example of each in this series. Here is the same code in the Short style and ASP style, respectively:

```
<? phpinfo(); ?>
<% phpinfo(); %>
```

As I previously mentioned, each statement in your PHP code is ended with a semicolon. Leaving this semicolon off is a common syntax error and when something isn't working the way you expect it to should be one of the first things you check.
Whitespace (spaces, tabs and newlines) are ignored in the syntax of PHP. These three examples are completely equivalent:

```
<?php phpinfo(); ?>
<?php    phpinfo();     ?>
<?php
phpinfo();
?>
```

4.3 Variables

As with other programming languages, PHP allows you to define variables. In PHP there are several variable types, but the most common is called a String. It can hold text and numbers. All strings begin with a $ sign. To assign some text to a string you would use the following code:

```
$welcome_text = "Hello and welcome to my website.";
```

This is quite a simple line to understand, everything inside the quotation marks will be assigned to the string. You must remember a few rules about strings though:

Strings are case sensetive so $Welcome_Text is not the same as $welcome_text
String names can contain letters, numbers and underscores but cannot begin with a number or underscore
When assigning numbers to strings you do not need to include the quotes so:

```
$user_id = 987
```

would be allowed.

4.3.1 Outputting Variables

To display a variable on the screen uses exactly the same code as to display text but in a slightly different form. The following code would display your welcome text:

```
<?
$welcome_text = "Hello and welcome to my website.";
print($welcome_text);
?>
```

As you can see, the only major difference is that you do not need the quotation marks if you are printing a variable.

4.3.2 Formatting Text

Everything is just output in the browser's default font. It is very easy, though, to format your text using HTML. This is because, as PHP is a server side language, the code is executed before the page is sent to the browser. This means that only the resulting information from the script is sent, so in the example above the browser would just be sent the text:

Hello and welcome to my website.

This means, though, that you can include standard HTML markup in your scripts and strings. The only problem with this is that many HTML tags require the " sign. You may notice that this will clash with the quotation marks used to print your text. This means that you must tell the script which quotes should be used (the ones at the beginning and end of the output) and which ones should be ignored (the ones in the HTML code).

Change the text to the Arial font in red. The normal code for this would be:

```
<font face="Arial" color="#FF0000">
</font>
```

As you can see this code contains 4 quotation marks so would confuse the script. Because of this you must add a backslash before each quotation mark to make the PHP script ignore it. The code would chang
e to:

```
<font face=\"Arial\" color=\"#FF0000\">
</font>
```

You can now include this in your print statement:

```
print("<font face=\"Arial\" color\"#FF0000\">Hello and welcome to my
website.</font>");
```

which will make the browser display:

Hello and welcome to my website.

because it has only been sent the code:

```
<font face="Arial" color="#FF0000">Hello and welcome to my website.</font>
```

4.4 PHP Operators

This section lists the different operators used in PHP.

4.4.1 Arithmetic Operators

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

4.4.2 Assignment Operators

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

4.4.3 Comparison Operators

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false

<=	is less than or equal to	5<=8 returns true

4.4.4 Logical Operators

Operator	Description	Example

&&	and	x=6
		y=3
		(x < 10 && y > 1) returns true

||	or	x=6
		y=3
		(x==5 || y==5) returns false

!	not	x=6
		y=3
		!(x==y) returns true

4.5 Conditional Statements

Very often when you write code, you want to perform different actions for different decisions.
You can use conditional statements in your code to do this.
if...else statement - use this statement if you want to execute a set of code when a condition is true and another if the condition is not true
elseif statement - is used with the if...else statement to execute a set of code if one of several condition are true

4.5.1 If...Else Statement

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

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

Example

The following example will output "Have a nice weekend!" if the current day is Friday, otherwise it will output "Have a nice day!":

```
<html>
<body>
<?php
$d=date("D");
```

```
if ($d=="Fri")
echo "Have a nice weekend!";
else
echo "Have a nice day!";
?>
</body>
</html>
```

If more than one line should be executed when a condition is true, the lines should be enclosed within curly braces:

```
<html>
<body>
<?php
$x=10;
if ($x==10)
{
echo "Hello<br />";
echo "Good morning<br />";
}
?>
</body>
</html>
```

4.5.2 The Else If Statement

If you want to execute some code if one of several conditions are true use the If_Else statement with Else If

Syntax

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

Example

The following example will output "Have a nice weekend!" if the current day is Friday, and "Have a nice Sunday!" if the current day is Sunday. Otherwise it will output "Have a nice day!":

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
echo "Have a nice weekend!";
elseif ($d=="Sun")
echo "Have a nice Sunday!";
```

```
else
echo "Have a nice day!";
?>
</body>
</html>
```

4.6 PHP String Processing

The string functions allow you to manipulate strings.

4.6.1 PHP String Functions

PHP: indicates the earliest version of PHP that supports the function.

Function	Description	PHP
addcslashes()	Returns a string with backslashes in front of the specified characters	4
addslashes()	Returns a string with backslashes in front of predefined characters	3
bin2hex()	Converts a string of ASCII characters to hexadecimal values	3
chop()	Alias of rtrim()	3
chr()	Returns a character from a specified ASCII value	3
chunk_split()	Splits a string into a series of smaller parts	3
convert_cyr_string()	Converts a string from one Cyrillic character-set to another	3
convert_uudecode()	Decodes a uuencoded string	5
convert_uuencode()	Encodes a string using the uuencode algorithm	5
count_chars()	Returns how many times an ASCII character occurs within a string and returns the information	4
crc32()	Calculates a 32-bit CRC for a string	4
crypt()	One-way string encryption (hashing)	3
echo()	Outputs strings	3
explode()	Breaks a string into an array	3
fprintf()	Writes a formatted string to a specified output stream	5
get_html_translation_table()	Returns the translation table used by htmlspecialchars() and htmlentities()	4
hebrev()	Converts Hebrew text to visual text	3
hebrevc()	Converts Hebrew text to visual text and new lines (\n) into <br />	3
html_entity_decode()	Converts HTML entities to characters	4
htmlentities()	Converts characters to HTML entities	3
htmlspecialchars_decode()	Converts some predefined HTML entities to	5

strtr()	Translates certain characters in a string	3
substr()	Returns a part of a string	3
substr_compare()	Compares two strings from a specified start position (binary safe and optionally case-sensitive)	5
substr_count()	Counts the number of times a substring occurs in a string	4
substr_replace()	Replaces a part of a string with another string	4
trim()	Strips whitespace from both sides of a string	3
ucfirst()	Converts the first character of a string to uppercase	3
ucwords()	Converts the first character of each word in a string to uppercase	3
vfprintf()	Writes a formatted string to a specified output stream	5
vprintf()	Outputs a formatted string	4
vsprintf()	Writes a formatted string to a variable	4
wordwrap()	Wraps a string to a given number of characters	4

4.7 The mail() Function

The mail() function is used to send emails.

Syntax

```
mail(to,subject,message,headers,parameters)
```

Parameter	Description
to	Required. Specifies the receiver / receivers of the email
subject	Required. Specifies the subject of the email. Note: This parameter cannot contain any newline characters
message	Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
headers	Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
parameters	Optional. Specifies an additional parameter to the sendmail program (the one defined in the sendmail_path configuration setting). (i.e. this can be used to set the envelope sender address when using sendmail with the -f sendmail option)

4.7.1 PHP Simple Text E-Mail

The simplest way to send an email with PHP is to send a simple text email. This is a simple text email where we define the variables and send a mail:

```
<?php
```

```
$to = "someone@someplace.com";
$subject = "Test mail";
$message = "Hello! This is a simple text email message.";
$from = "someonelse@anotherplace.com";
$headers = "From: $from";

mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

4.7.2 PHP Mail Form

Using PHP you can create a feedback form for your website. In this example it sends a text message to a specified e-mail.
When using HTML forms with PHP, any form element in the HTML form will automatically be available to the PHP script.
This is how this example works:
Check if the email input is set
If it is not set (like when the page is first visited) it will output the HTML mail form
If the email input is set (like after the form is filled out) it will send the mail from the form
When submit is pressed after the form is filled out, the page reloads, sees that the email input is set, and sends the mail.

```
<html>
<body>
<?php
if (isset($_REQUEST['email']))
  {
  $email = $_REQUEST['email'] ;
  $subject = $_REQUEST['subject'] ;
  $message = $_REQUEST['message'] ;
  mail( "someone@someplace.com", "Subject: $subject",
  $message, "From: $email" );

  echo "Thank you for using our mail form";
  }
else
  {
  echo "<form method='post' action='mailform.php'>
  Email: <input name='email' type='text' /><br />
  Subject: <input name='subject' type='text' /><br />
  Message:<br />
  <textarea name='message' rows='15' cols='40'>
  </textarea><br />
```

```
 <input type='submit' />
 </form>";
 }
?>
</body>
</html>
```

4.7.3 Requirements

For the mail functions to be available, PHP requires an installed and working email system. The program to be used is defined by the configuration settings in the php.ini file.

The mail functions are part of the PHP core. There is no installation needed to use these functions.

4.7.4 Runtime Configuration

The behavior of the mail functions is affected by settings in the php.ini file. Mail configuration options:

Name	Default	Description	Changeable
SMTP	"localhost"	Windows only: The DNS name or IP address of the SMTP server	PHP_INI_ALL
smtp_port	"25"	Windows only: The SMTP port number. Available since PHP 4.3	PHP_INI_ALL
sendmail_from	NULL	Windows only: Specifies the "from" address to be used in email sent from PHP	PHP_INI_ALL
sendmail_path	NULL	Unix systems only: Specifies where the sendmail program can be found (usually /usr/sbin/sendmail or /usr/lib/sendmail)	PHP_INI_SYSTEM

4.7.5 PHP Mail Functions

PHP: indicates the earliest version of PHP that supports the function.

Function	Description	PHP
ezmlm_hash()	Calculates the hash value needed by the EZMLM mailing list system	3
mail()	Allows you to send emails directly from a script	3

4.8 Cookies

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests for a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

4.8.1 How to Create a Cookie

The setcookie() function is used to create cookies.
Note: The setcookie() function must appear BEFORE the <html> tag.
Syntax

```
setcookie(name, value, expire, path, domain);
```

Example
The following example sets a cookie named "uname" - that expires after ten hours.

```
<?php
setcookie("uname", $name, time()+36000);
?>
<html>
<body>
<p>
A cookie was set on this page! The cookie will be active when
the client has sent the cookie back to the server.
</p>
</body>
</html>
```

4.8.2 How to Retrieve a Cookie Value

When a cookie is set, PHP uses the cookie name as a variable.
To access a cookie you just refer to the cookie name as a variable.
Tip: Use the isset() function to find out if a cookie has been set.
Example
The following example tests if the uname cookie has been set, and prints an appropriate message.

```
<html>
<body>
<?php
if (isset($_COOKIE["uname"]))
echo "Welcome " . $_COOKIE["uname"] . "!<br />";
else
echo "You are not logged in!<br />";
?>
</body>
</html>
```

CHAPTER FIVE : MySQL

5.1 What is MySQL?

MYSQL is a type of SQL database management featured in the Linux hosting plans. A database is an organized collection of information that a computer uses to select and display data. Databases can help organize and enhance our site content. Sites with dynamic pages and/or shopping cart software often need an underlying database structure.
MYSQL is also a popular database server which is available for Linux, FreeBSD and other flavors of UNIX, and also Win32 platforms. MYSQL is often used as a database back-end to PHP web applications or CGIs invoked by Perl database modules.
MYSQL has highly configurable user/permissions model as well as network access permissions configuration. MYSQL is typically accessed via the client software at the command prompt on a Linux or FreeBSD server. MYSQL is much faster than Oracle or Microsoft Access, it's also considered to be the fastest data base server available.
We can also define MYSQL as a relational database management system, which means it stores data in separate tables rather than putting all the data in one big area. This adds flexibility, as well as speed. The SQL part of MYSQL stands for "Structured Query Language," which is the most common language used to access databases.

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

5.2 Database Construction

MySQL databases have a standard setup. They are made up of a database, in which is contained tables. Each of these tables is quite separate and can have different fields etc. even though it is part of one database. Each table contains records which are made up of fields.

5.2.1 Databases And Logins

The process of setting up a MySQL database varies from host to host, you will however end up with a database name, a user name and a password. This information will be required to log in to the database.

If you have PHPMyAdmin (or a similar program) installed you can just go to it to log in with your user name and password. If not you must do all your database administration using PHP scripts.

5.3 Creating A Table

Before you can do anything with your database, you must create a table. A table is a section of the database for storing related information. In a table you will set up the different fields which will be used in that table. Because of this construction, nearly all of a site's database needs can be satisfied using just one database.

Creating a table in PHPMyAdmin is simple, just type the name, select the number of fields and click the button. You will then be taken to a setup screen where you must create the fields for the database. If you are using a PHP script to create your database, the whole creation and setup will be done in one command.

5.3.1 Fields

There are a wide variety of fields and attributes available in MySQL and I will cover a few of these here:

Field Type	Description
TINYINT	Small Integer Number
SMALLINT	Small Integer Number
MEDIUMINT	Integer Number
INT	Integer Number
VARCHAR	Text (maximum 256 characters)
TEXT	Text

These are just a few of the fields which are available. A search on the internet will provide lists of all the field types allowed.

5.3.2 Creating A Table With PHP

To create a table in PHP is slightly more difficult than with MySQL. It takes the following format:

```
CREATE TABLE tablename {

Fields

}
```

The fields are defined as follows:

```
fieldname type(length) extra info,
```

The final field entered should not have a comma after it.

I will give full an example of using these later in the section.

5.3.3 The Contacts Database

The contacts database will contain all the conact information for the people you enter and the information will be able to be edited and viewed on the internet. The following fields will be used in the database:

Name	Type	Length	Description
id	INT	6	A unique identifier for each record
first	VARCHAR	15	The person's first name
last	VARCHAR	15	The person's last name
phone	VARCHAR	20	The person's phone number
mobile	VARCHAR	20	The person's mobile number
fax	VARCHAR	20	The person's fax number
email	VARCHAR	30	The person's e-mail address
web	VARCHAR	30	The person's web address

You may be wondering why I have used VARCHAR fields for the phone/fax numbers even though they are made up of digits. You could use INT fields but I prefer to use VARCHAR as it will allow dashes and spaces in the number, as well as textual numbers (like 1800-COMPANY) and as we will not be initiating phone calls from the web it is not a problem.

There is one other thing you should be aware of in this database. The id field will also be set as PRIMARY, INDEX, UNIQUE and will be set to auto_increment (found under Extra in PH
PMyAdmin). The reason for this is that this will be the field identifier (primary and index) and so must be unique. The auto increment setting means that whenever you add a record, as long as you don't specify an id, it will be given the next number.

If you are using PHPMyAdmin or a management program you can now create this in a table called contacts.

5.4 Connect to MySQL Server

The first thing you'll want to do is to establish a connection to your MySQL database server. This is accomplished with the mysql_connect function. Here's an example:
$connid = mysql_connect ('servername' , 'username' , 'password');
$connid is an identifier (a positive integer, returned by the mysql_connect function) that identifies this connection and is used in subsequent function calls to tie those calls to this

connection. The mysql_connect function is provided three parameters, the name of the server (commonly 'localhost' when the server is on the same computer as PHP), the username and password authorized to access the server.
Next, we need to check that the connection was successfully established. For the sake of this example we are going to print a message on the screen to let us know of whether or not the call was good. In practice, you will probably take a different course of action (see also the "error control operator, below.)

```
if ($connid = = false)
 { print "Server connection failed";  }
else
 { print "Server connected!"; }
```

When we are all done, it is a sound programming practice to close things that we have opened. To close our connection we will use the mysql_close function. Notice that we use the $connid to reference the connection that we opened.

```
mysql_close ($connid);
```

Nice and tidy!

5.4.1 Using The Error Control Operator

The error control operator can be used to prevent PHP from displaying an error message in the user's browser when the attemp to connect to the MySQL server fails. In the real world, this is the more common requirement. The error control operator is an @ sign placed next to the function call like this:

```
$connid = @mysql_connect ('servername' , 'username' , 'password');
```

5.4.2 Using the Die Function

Another common real world practice is to use the die function to terminate the program and display a message. Without the connection to the database, it's most likely that there's no point in continuing!

```
$connid = @mysql_connect ('servername' , 'username' , 'password') or die("Connection to the DBMS server failed");
```

5.5 Creating a Database

Create a database in MySQL with PHP.
MySQL Syntax

```
CREATE DATABASE database_name
```

Now we use this together with the mysql_query() function.
All we have to do is to add the MySQL syntax to the mysql_query() function.
Example
Here we create a database called "my_db":

```
<?php
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
```

```
 {
 die('Could not connect: ' . mysql_error());
 }
$sql = "CREATE DATABASE my_db";
if (mysql_query($sql,$con))
 {
 echo "Database my_db created";
 }
else
 {
 echo "Error creating database: " . mysql_error();
 }
?>
```

MySQL Syntax To create a table in a database:

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
.......
)
```

Now we use this together with the mysql_query() function.

Example

This example demonstrates how you can create a table named "Person", with three columns. The column names will be "FirstName", "LastName" and "Age":

```
mysql_select_db("my_db", $connection);
$sql = "CREATE TABLE Person
(
FirstName varchar(15),
LastName varchar(15),
Age int
)";
mysql_query($sql,$con);
```

Note: A database must be selected before a table can be created. This is done in the first line of the example above.

Note: While using PHP to create the varchar data type in a table, you must add the max length parameter, like shown above.

Here is the different MySQL data types that can be used:

Numbers	Description

int(size)
smallint(size)
tinyint(size)
mediumint(size)
bigint(size)
	Hold integers only. The maximum number of digits are specified in parenthesis

decimal(size,d)
double(size,d)
float(size,d)
	Hold numbers with fractions. The maximum number of digits are specified in "size". The maximum number of digits to the right of the decimal is specified in "d"

Text	Description
char(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis
varchar(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis
tinytext	Holds a variable string with a maximum length of 255 characters
text blob	Holds a variable string with a maximum length of 65535 characters
mediumtext mediumblob	Holds a variable string with a maximum length of 16777215 characters
longtext longblob	Holds a variable string with a maximum length of 4294967295 characters

Date	Description

date(yyyy-mm-dd)
datetime(yyyy-mm-dd hh:mm:ss)
timestamp(yyyymmddhhmmss)
time(hh:mm:ss)
	Holds date and/or time

Misc	Description
enum(value1,value2,ect)	ENUM is short for ENUMERATED list. Can store one of up to 65535 values listed within the ( ) brackets. If a value is inserted that is not in the list, a blank value will be inserted
set	SET is similar to ENUM. However, SET can have up to 64 list items and can store more than one choice

5.6 Primary Key and Auto Increment

Each table should have an unique identifier field. This field is called a primary key.
The primary key field is often an ID number, and is often used with the AUTO_INCREMENT setting.
When used, AUTO_INCREMENT adds 1 to the value of the field each time a new entry is added.
To make sure that no primary key fields can be NULL, we add the NOT NULL setting to ensure that the ID value can not be NULL.

Example
This is the same example from above, but with an primary key ID column using AUTO_INCREMENT and NOT NULL:

```
$sql = "CREATE TABLE Person
(
id int NOT NULL AUTO_INCREMENT,
PRIMARY KEY(id),
FirstName varchar(15),
LastName varchar(15),
Age int
)";
mysql_query($sql,$con);
```

5.7 PHP MySQL Functions

Function	Description	PHP
mysql_affected_rows	Returns the number of affected rows in the previous MySQL operation	3
mysql_change_user	Deprecated. Changes the user of the current MySQL connection	3
mysql_client_encoding	Returns the name of the character set for the current connection	4
mysql_close	Closes a non-persistent MySQL connection	3
mysql_connect	Opens a non-persistent MySQL connection	3
mysql_create_db	Deprecated. Creates a new MySQL database. Use mysql_query() instead	3
mysql_data_seek	Moves the record pointer	3
mysql_db_name	Returns a database name from a call to mysql_list_dbs()	3
mysql_db_query	Deprecated. Sends a MySQL query. Use mysql_select_db() and mysql_query() instead	3
mysql_drop_db	Deprecated. Deletes a MySQL database. Use mysql_query() instead	3
mysql_errno	Returns the error number of the last MySQL operation	3
mysql_error	Returns the error description of the last MySQL operation	3
mysql_escape_string	Deprecated. Escapes a string for use in a mysql_query. Use mysql_real_escape_string() instead	4
mysql_fetch_array	Returns a row from a recordset as an associative array and/or a numeric array	3
mysql_fetch_assoc	Returns a row from a recordset as an associative array	4
mysql_fetch_field	Returns column info from a recordset as an object	3
mysql_fetch_lengths	Returns the length of the contents of each field in a result row	3
mysql_fetch_object	Returns a row from a recordset as an object	3

mysql_fetch_row	Returns a row from a recordset as a numeric array	3
mysql_field_flags	Returns the flags associated with a field in a recordset	3
mysql_field_len	Returns the maximum length of a field in a recordset	3
mysql_field_name	Returns the name of a field in a recordset	3
mysql_field_seek	Moves the result pointer to a specified field	3
mysql_field_table	Returns the name of the table the specified field is in	3
mysql_field_type	Returns the type of a field in a recordset	3
mysql_free_result	Free result memory	3
mysql_get_client_info	Returns MySQL client info	4
mysql_get_host_info	Returns MySQL host info	4
mysql_get_proto_info	Returns MySQL protocol info	4
mysql_get_server_info	Returns MySQL server info	4
mysql_info	Returns information about the last query	4
mysql_insert_id	Returns the AUTO_INCREMENT ID generated from the previous INSERT operation	3
mysql_list_dbs	Lists available databases on a MySQL server	3
mysql_list_fields	Deprecated. Lists MySQL table fields. Use mysql_query() instead	3
mysql_list_processes	Lists MySQL processes	4
mysql_list_tables	Deprecated. Lists tables in a MySQL database. Use mysql_query() instead	3
mysql_num_fields	Returns the number of fields in a recordset	3
mysql_num_rows	Returns the number of rows in a recordset	3
mysql_pconnect	Opens a persistent MySQL connection	3
mysql_ping	Pings a server connection or reconnects if there is no connection	4
mysql_query	Executes a query on a MySQL database	3
mysql_real_escape_string	Escapes a string for use in SQL statements	4
mysql_result	Returns the value of a field in a recordset	3
mysql_select_db	Sets the active MySQL database	3
mysql_stat	Returns the current system status of the MySQL server	4
mysql_tablename	Deprecated. Returns the table name of field. Use mysql_query() instead	3
mysql_thread_id	Returns the current thread ID	4
mysql_unbuffered_query	Executes a query on a MySQL database (without fetching / buffering the result)	4

5.7.1 Selecting a Database

Here's something that's not too complicated: to select a MySQL database in PHP use the mysql_select_db ( ) function, like this:

mysql_select_db ("mydatabase");
Couldn't be simpler, as long as everything went well -- more on that in a minute.

5.7.2 Handling Errors

In the last part of this tutorial series, I discussed the use of the Error Control Operator and the die ( ) function. Now we'll add one more function to our arsenal. The MySQL support in PHP includes a function called mysql_error ( ) whose very useful purpose is to provide a text version of an error returned by the MySQL server. We can certainly take advantage of that in our error handling!
I also want to introduce another element here. This is purely a matter of coding technique. There are many ways to write this piece of code, but I feel this form to be quite elegant. You will remember that an "if" statement has this basic format:

```
if (condition)
  {do this if condition is true}
 else
  {do this if condition is false}
```

We can take advantage of that. When a function performs properly it returns with a condition of "true", and when something goes wrong it returns with a "false" condition. Of course, it can also return values, error codes and the like, but for the moment lets ponder just the condition. Since it returns a condition, we can use the function call as the "condition" in an "if" statement. This will allow us to put code in the "true" side of the statement to do what we want when everything is ok, and to put code in the "false" side to handle our errors.
Let's build an example. First, we're going to connect to the server, and if everything is good, we'll create a database. I there were errors, we'll terminate the program, displaying an error message. In the process of creating the database, if everything goes well, we'll select the database so that we can use it, and if there are errors we'll terminate the program, displaying an appropriate message. Similarly, when selecting the database, we'll use messages to let us know how things went.
Programmatically, this means putting an "if" statement inside the "do this if condition is true" code. When an "if" statement is put into either side of another "if" statement in this fashion, it is know as "nesting" the "if" statements. When you write nested "if"s it is highly recommended that you use indentation and spacing to help you keep track of where you are. It is altogether too easy to have and extra or a missing parenthesis and completely change the logic flow of your program -- often without causing any PHP syntax error. Be warned! Be careful!
Now, here's our code (you might want to spend a little time following through this to make sure you fully understand it):

```
if ( $connid = mysql_connect ( 'localhost' 'username' 'password' ) )
  {  print '<p> Connected to MySQL Server. </p>';
     if ( @mysql_query ( 'CREATE DATABASE murphy' ) )
       {  print '<p> Database murphy created. <p>';
          if ( @mysql_select_db ( 'murphy' ) )
            { print '<p> Databse murphy selected </p>';
            }
          else
           {  die ( ' <p> Database selection failed because ' , mysql_error ( ) , ' </p>' )
            }
        }
      else
        {  die ( ' <p> Database creation failed because ' , mysql_error ( ) , ' </p>' )
          }
   }
  else
  {  die ( ' <p> Database Server connection failed because ' , mysql_error ( ) , ' </p>' )
    }
```

.

5.8 Inserting Data From a Form to a Database

In PHP we can allow the users to use a form to insert or edit data in a database. First we create a form:

```
<form action="insert_db.php" method="POST">
Enter your Firstname: <input type="text" name="firstname" />
Enter your Lastname: <input type="text" name="lastname" />
Enter your Age: <input type="text" name="age" />
<input type="submit" />
</form>
```

Then the "insert_db.php" page:

```
$con = mysql_connect("localhost","peter","abc123");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("my_db", $con);
$sql="INSERT INTO person
(firstname,lastname,age)
VALUES
('$_POST[firstname]','$_POST[lastname]','$_POST[age]')";
if (!mysql_query($sql,$con))
  {
  die('Error: ' . mysql_error());
  }
echo "Success!";
```

5.9 ODBC With PHP

ODBC is an Application Programming Interface (API) that allows you to connect to a data source (e.g. an MS Access database).
Create an ODBC Connection
With an ODBC connection, you can connect to any database, on any computer in your network, as long as an ODBC connection is available.
Here is how to create an ODBC connection to a MS Access Database:
Open the Administrative Tools icon in your Control Panel.
Double-click on the Data Sources (ODBC) icon inside.
Choose the System DSN tab.
Click on Add in the System DSN tab.
Select the Microsoft Access Driver. Click Finish.
In the next screen, click Select to locate the database.
Give the database a Data Source Name (DSN).
Click OK.
Note that this configuration has to be done on the computer where your web site is located. If you are running Internet Information Server (IIS) on your own computer, the instructions above will work, but if your web site is located on a remote server, you have to have physical access to that server, or ask your web host to to set up a DSN for you to use.

5.9.1 Connecting to an ODBC

The odbc_connect() function is used to connect to an ODBC data source. The function takes four parameters: the data source name, username, password, and an optional cursor type.
The odbc_exec() function is used to execute an SQL statement.
Example
The following example creates a connection to a DSN called northwind, with no username and no password. It then creates an SQL and executes it:

```
$conn=odbc_connect('northwind','','');
$sql="SELECT * FROM customers";
$rs=odbc_exec($conn,$sql);
```

5.9.2 Retrieving Records

The odbc_fetch_rows() function is used to return records from the result-set. This function returns true if it is able to return rows, otherwise false.
The function takes two parameters: the ODBC result identifier and an optional row number:

```
odbc_fetch_row($rs)
```

Retrieving Fields from a Record
The odbc_result() function is used to read fields from a record. This function takes two parameters: the ODBC result identifier and a field number or name.
The code line below returns the value of the first field from the record:

```
$compname=odbc_result($rs,1);
```

The code line below returns the value of a field called "CompanyName":

```
$compname=odbc_result($rs,"CompanyName");
```

5.9.3 Closing an ODBC Connection

The odbc_close() function is used to close an ODBC connection.

```
odbc_close($conn);
```

An ODBC Example
The following example shows how to first create a database connection, then a result-set, and then display the data in an HTML table.

```
<html>
<body>
<?php
$conn=odbc_connect('northwind','','');
if (!$conn)
  {exit("Connection Failed: " . $conn);}
$sql="SELECT * FROM customers";
$rs=odbc_exec($conn,$sql);
if (!$rs)
  {exit("Error in SQL");}
echo "<table><tr>";
echo "<th>Companyname</th>";
echo "<th>Contactname</th></tr>";
while (odbc_fetch_row($rs))
{
  $compname=odbc_result($rs,"CompanyName");
  $conname=odbc_result($rs,"ContactName");
  echo "<tr><td>$compname</td>";
  echo "<td>$conname</td></tr>";
}
odbc_close($conn);
echo "</table>";
?>
</body>
</html>
```

CHAPTER SIX : ABOUT PROJECT

6.1 User Interface

This chapter reflects main content and aims of this Project. This includes usage of symbols and meaning of terms.

As we mentioned above, human beings facing very important multy dimensional changing and transformations. In this complicated procces, terms such as time communication, internet information and transportation have gained very strategic importance. Our economic vision is being shaped with in this genaral perspective.In this framework, we have chosen transformation sector to serve people. This is rent car service. As mentioned, in this global procces, we aimed to serve people in the web. With this choise, we can arrive all potantial clients easily.

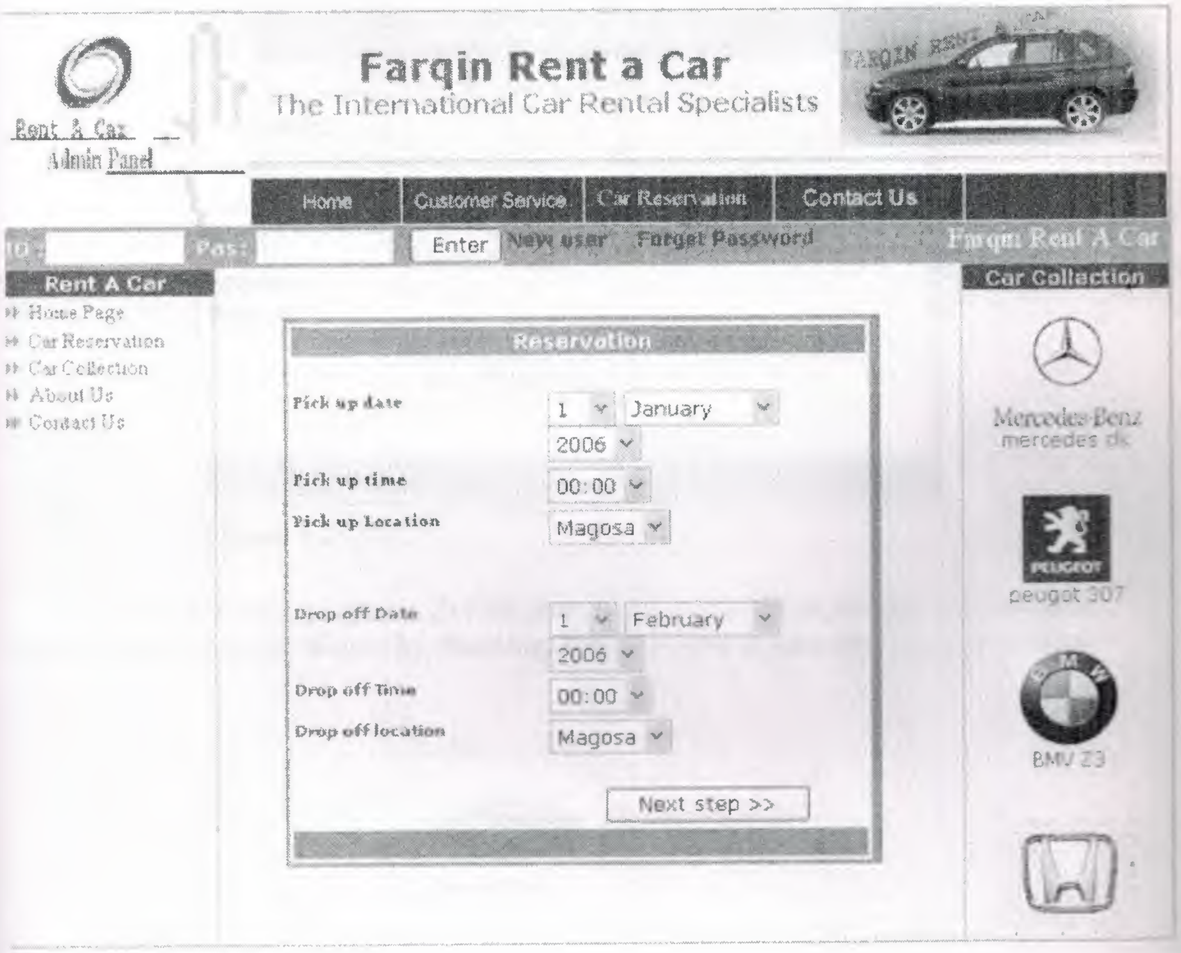
This proje formed in two parts. The first is user part, the other is administration panel.

In the home page there are several links that are information about web site, feuture of cars and memberships form. Home page was tried to be very attractive for renting.

All modifications that are about web site can be made, in the administration panel. For instance, visiter or customer of web site can regulate or delete cars that are rented and can appoint manager for renting. There are several missions of managers. First is evaluation of reservations that have come on the internet. The second is to ratify informations which has come by web by confirming user inputs. The last is to realize sale procces succesfully. Superviser has not got authority as much as manager.

This project shortly, creates opportunity to rent car, no matter where you are in the world.

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

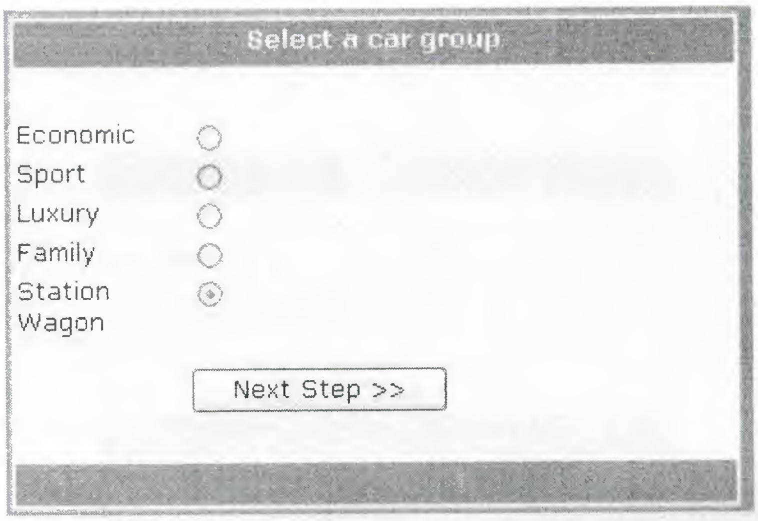


Reservation

Pick up date	1 January 2006
Pick up time	00:00
Pick up Location	Magosa
Drop off Date	1 February 2006
Drop off Time	00:00
Drop off location	Magosa

Figure 1.1

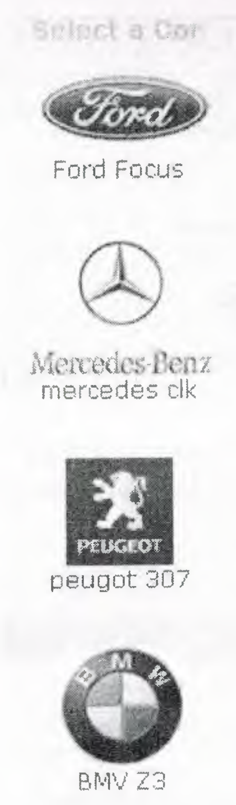
In the left part of page, there are links that related connections of site. In the right part of page, also there are car brands.There are two forms in the home page. One is for user entry, the other has reservation form to rent car. In the reservation form part, there are dates of pick up and drop of time for determination. After reservation, user passes to second step by clicking next buton.


Select a car group

Economic ○
Sport ○
Luxury ○
Family ○
Station Wagon [✓ ○]

Figure 1.2

In the this stage,(Figure 1.2) Cars groups are chosen by customers. For instance, one can choise Station Wagon by choosing, then it is time to pass onother step or stage.


Select a Car

Ford Focus

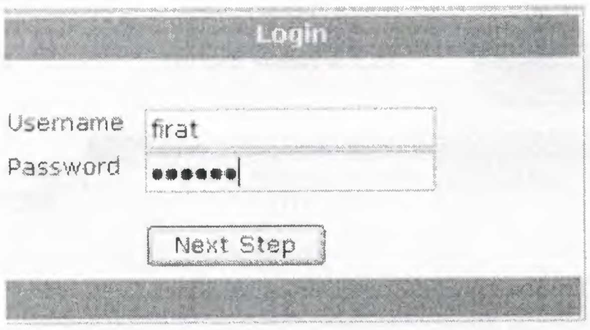
mercedes clk

peugot 307

BMV Z3

Figure 1.3

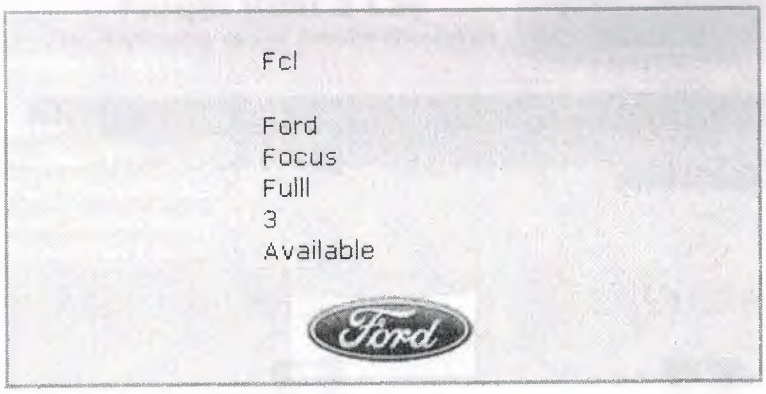
After choosing of car groups, viseters or customers select any brands that they demand by clicking button (Figure1.3).


Login

Username	firat
Password	●●●●●●

Figure1.4

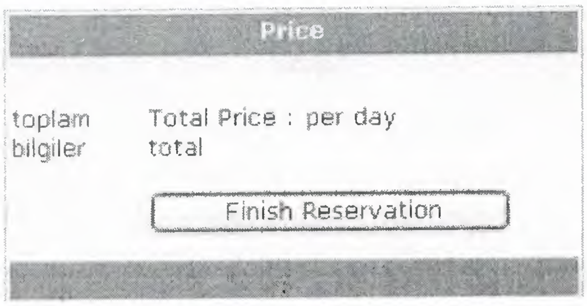
After selection of car brand, user enter user check. İf user is not log in, user will be directed to figure1.4 form part to enter user entry.


Fcl

Ford
Focus
Fulll
3
Available

Figure1.5

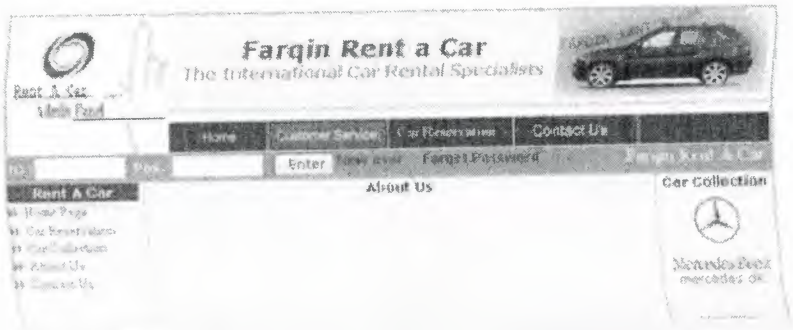
In the figure 1. 5, there are informations about selected cars such as numberplate,model,future and resarvation informations about past.


Price

toplam	Total Price : per day
bilgiler	total

Figure 1.6

There are informations that are about total price of cars which is choosed in the this part.



Figre 1.7

In this part, there are genaral informations about firm

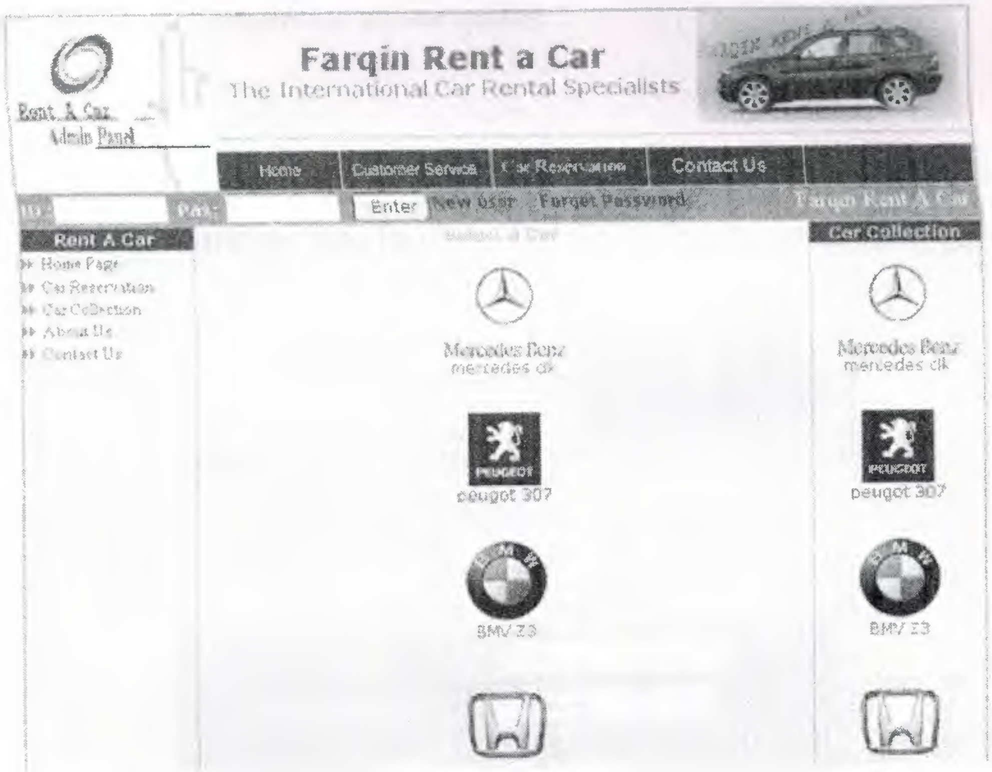


Figure1.8

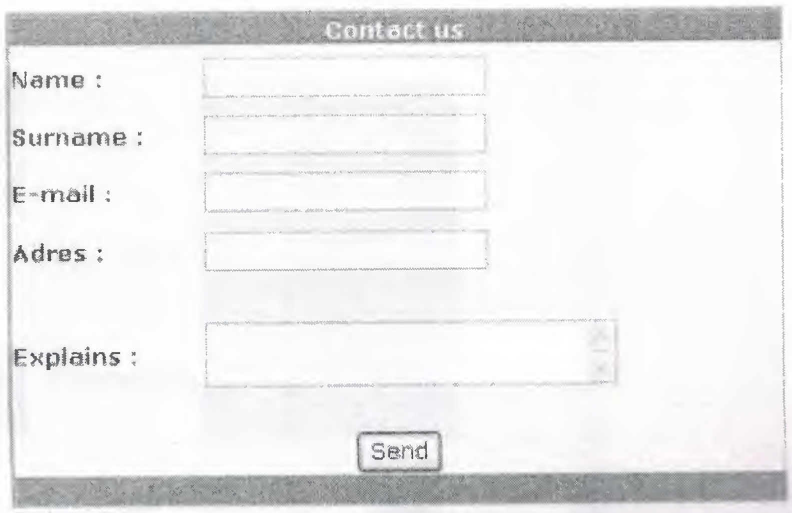


Figure 1.9

In the figure 1.9 part, There is a form for every sort of communications. In the "explain" part, customers can leave their views .This views provide connection with the managers about complaintments or service quality.

6.2 Administration Interface

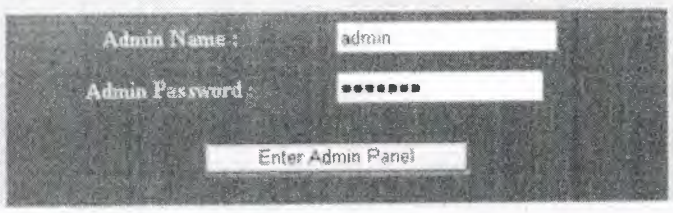
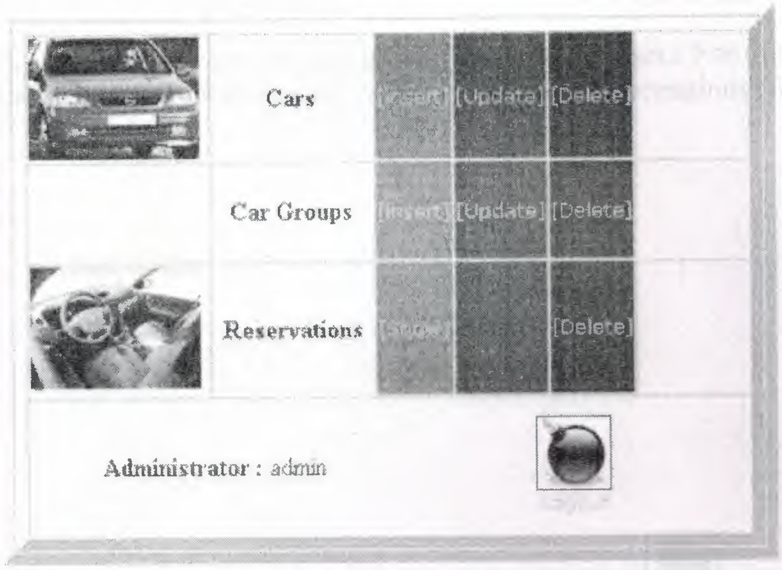


Figure 1.10

Figure 1.10 is entry page of site that goes to administration panel. Only managers can reach to this panel. Entry of panel requires user name and password.

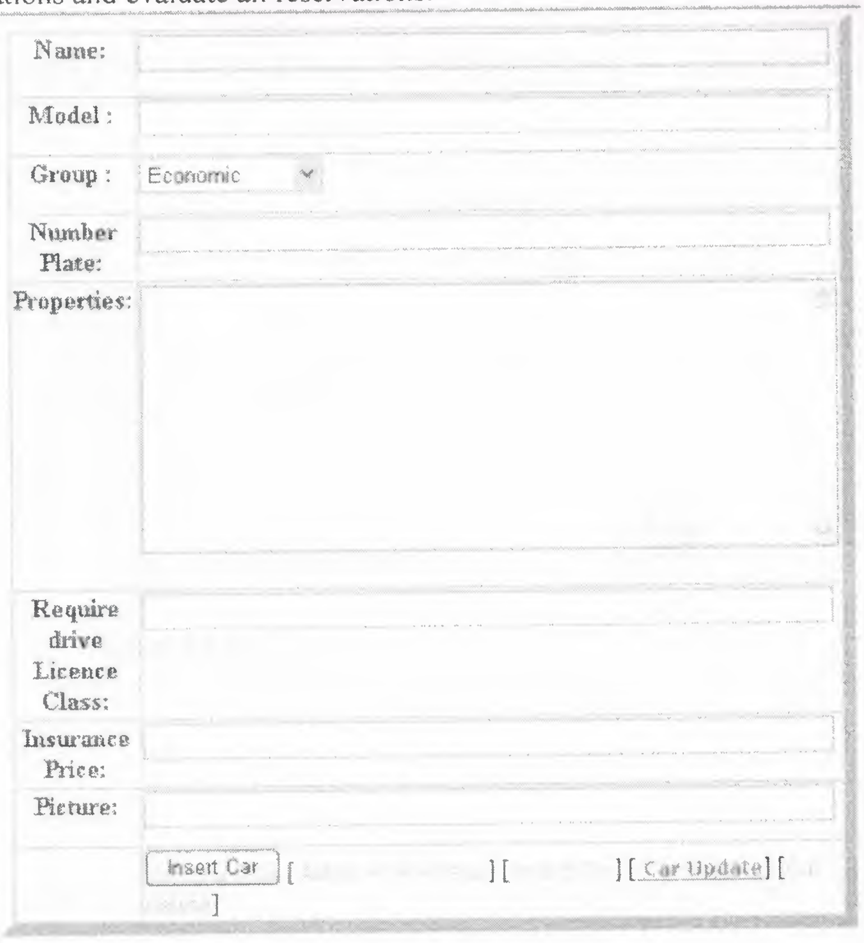


	Cars	[Insert]	[Update]	[Delete]
	Car Groups	[Insert]	[Update]	[Delete]
	Reservations	[Show]		[Delete]

Administrator : admin

Figure 1.11

In the figure of 2.2, there are three alternatives in the control panel. These are entry of car, car groups and reservations. Manager in the this panel can change all modifications and evaluate all reservations.



Name:	
Model :	
Group :	Economic
Number Plate:	
Properties:	
Require drive Licence Class:	
Insurance Price:	
Picture:	
	Insert Car [ ] [ ] [Car Update] [ ]

Figure 1.12

Manager can add car groups that who he wants. For this, manager regulates the car alternatives for users by entering necessary informations (figure 1.11)

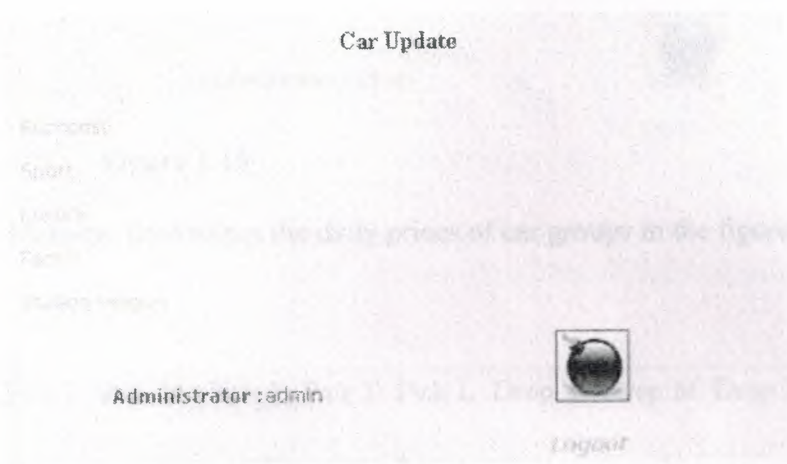


Figure 1.13

In the this figure there are car groups. Manager can change all informations.

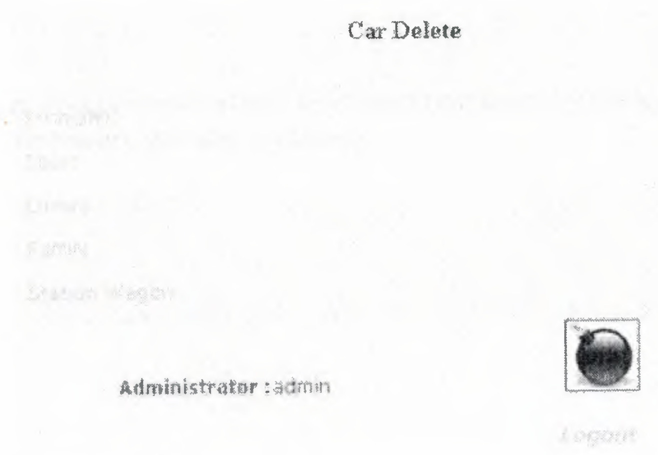
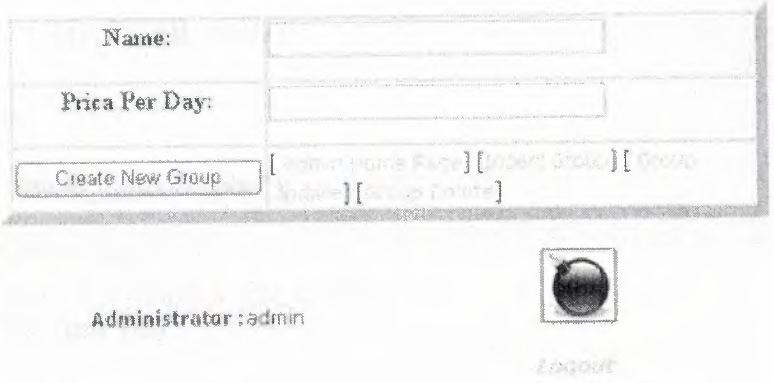


Figure 1.14



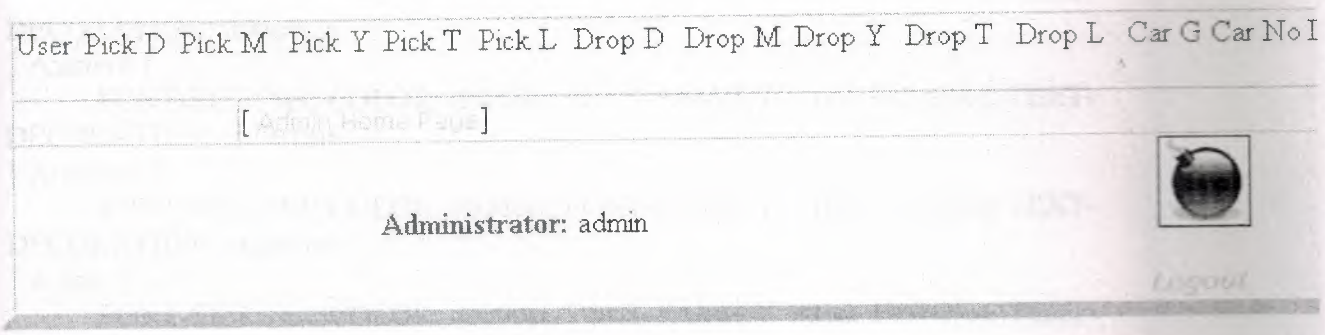
Name:

Prica Per Day:

Create New Group
[Admin Home Page] [Insert Group] [Group Update] [Group Delete]
Administrator : admin
Logout

Figure 1.15

Manager determines the daily prices of car groups in the figure 1.15



User	Pick D	Pick M	Pick Y	Pick T	Pick L	Drop D	Drop M	Drop Y	Drop T	Drop L	Car G	Car No I

Figure 1.16

Manager accepts resarvations by connecting users by clicking "accept" link. After this operation, necessary procces continues.

CHAPTER SEVEN : APPLICATION CODES

7.1.1 index.php

```
<?
require_once("rentcar_function.php");
session_start();
$today = date("d"); //today
$month_day = date("t"); //days in this month
$year= date("Y"); //this year
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <head>
<title>Welcome to  Farqin Global Car Rental</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<STYLE>A {
        FONT-SIZE: 9pt; COLOR: #0000ff; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: underline
} A:active {
        FONT-SIZE: 9pt; COLOR: #ff0000; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: underline
} A:visited {
        FONT-SIZE: 9pt; COLOR: #0000ff; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: underline
} A:link {
        FONT-SIZE: 9pt; COLOR: #0000ff; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: underline
}  A.nu {
        FONT-SIZE: 9pt; COLOR: #000000; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: none
} A.nu:active {
        FONT-SIZE: 9pt; COLOR: #ff0000; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: none
} A.nu:visited {
        FONT-SIZE: 9pt; COLOR: #000000; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: none
} A.nu:link {
        FONT-SIZE: 9pt; COLOR: #000000; FONT-FAMILY: Arial, Helvetica; TEXT-
DECORATION: none
}</STYLE>
<link href="stil/master.css" rel="stylesheet" type="text/css">
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
```

```
}function MM_swapImgRestore() { //v3.0  var i,x,a=document.MM_sr;
for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;          }
function MM_findObj(n, d) { //v4.01 var p,i,x;  if(!d) d=document;
if((p=n.indexOf("?"))>0&&parent.frames.length) {
d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
if(!x && d.getElementById) x=d.getElementById(n); return x;
}  function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array;
for(i=0;i<(a.length-2);i+=3)
if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src;
x.src=a[i+2];}
}//--></script>  </head> <body ><table bgColor=#f5f5f5>  <tr align="left"
valign="top"> <td height="130"  colspan="3" > <table >  <tr valign="top">  <td
colspan="2" align="left"> <table ><tr> <td ><img src="images/logo.jpg" width="153"
height="133"></td><td > <table width="100%" height="123" border="0"
cellpadding="0" cellspacing="0"><tr> <td  align="left" valign="top">
<strong></strong> <table ><tr> <td width="75%"><div align="center"><strong><font
size="5">Farqin Rent a Car </font><font ><b><br><font size="3">The International
Car Rental Specialists</font></b></font> </strong></div></td><td width="25%"><img
src="images/car.jpg" width="200" height="85"></td>  </tr>  </table></td> </tr> <tr>
<td align="left" valign="bottom"> <hr size="3"> <table >  <tr> <td ><a
href="index.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('menu1','','images/btnHome_on.jpg',1)"><img
src="images/btnHome.jpg" alt="Home" name="menu1" > </a> </td> <td ><a
href="sasa.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('menu2','','images/btnCustomerService_on.jpg',1)"><i
mg src="images/btnCustomerService.jpg" alt="Customer Service" name="menu2"
width="110" height="25" border="0"></a></td>
<td ><a href="reservation.php?step=1" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('menu3','','images/reservon1.jpg',1)"><img
src="images/reserv.jpg" alt="Reservation" name="menu3" width="115" height="25"
border="0"></a></td> <td align="center" width="21%" bgcolor="#333366"><a
href="contact.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('menu4','','images/contact_on.jpg',1)"><img
src="images/contact.jpg" alt="Reservation" name="menu4" width="115" height="25"
border="0"></a></td>
<td align="center" width="24%" bgcolor="#333366">&nbsp;</td>
</tr></table></td></tr></table></td></tr></table></td></tr><tr valign="top">
<td  > <table ><tr> <td > </td></tr>   </table>
<? if(!is_user())
{   ?>
<table > <tr> <form name="form1" method="post" action="user.php">
<td width="25"  align="center" valign="middle"><font color="#FFFFFF"
size="2"><strong>ID: </strong></font></td><td width="88"  align="left" valign="top">
```

```
<input name="username" type="text" size="10" ></td><td width="41" align="center"
valign="middle"><font " size="2"><strong>Pas: </strong></font></td> <td width="88"
align="left" valign="top"> <input name="user_pass" type="password" size="10"> </td>
<td width="67" align="right" valign="top"><input type="submit" name="Submit2"
value=" Enter"> </td> <td width="266" align="left" valign="top" nowrap><a
href="register.php"><strong> New user</strong> </a> &nbsp;&nbsp;<strong> <a
href="forget_password.php">Forget Password</a></strong> </td>     </form>      </tr>
</table>
<?  } else
{ echo "<a href='signout.php'>Sign out </a>";
}  ?>          </td>
<td > Farqin Rent A Car </td>     </tr>     </table></td>  </tr>  <tr
bgColor=#f5f5f5>  <table>  <tr><td ><table >  <tr> <td>   <strong>Rent A
Car</strong></td>   </tr> <tr>  <td ><img src="images/ok.GIF" width="5"
height="9"><img src="images/ok.GIF" > <a href="index.php"> Home Page</a></td>
</tr>   <tr>  <td><img src="images/ok.GIF" ><img src="images/ok.GIF" <a
href="reservation.php?step=1"><font size="2"> Car Reservation</font></a></td>  </tr>
<tr> <td><img src="images/ok.GIF" ><img src="images/ok.GIF> <a href="car.php>Car
Collection </a></td>  </tr>     <tr>   <td><img src="images/ok.GIF" ><img
src="images/ok.GIF" width="5" height="9">           <a href="about_us.php"><font
size="2" face="Times New Roman, Times, serif"> About Us </font></a></td>  </tr>
<tr>  <td><img src="images/ok.GIF" ><img src="images/ok.GIF" > <a
href="contact.php"> Contact Us</a></td>   </tr>  <tr>   <td >&nbsp;</td>            </tr>
<tr>  <td >&nbsp;</td>   </tr> </table></td>  </tr>      </table>  </td><td><table
width="100%" border="0" cellspacing="0" cellpadding="0">  <tr align="left"
valign="top">  <td  > </td>  <td ><br><br>  <table> <tr>  <td>
<table><form name="form2" method="post" action="reservation.php?step=2">   <tr>
 <td colspan="2"> <div align="center"> Reservation </div></td> </tr><tr >
<td width="44%">&nbsp;</td><td >&nbsp;</td>  </tr> <tr >  <td><strong>Pick up
date</strong></td> <td><select name="pick_day" id="pick_day">
   <? for($g=1; $g<=31; $g++)
{    echo "<option value=\"$g\">$g</option>";           }        ?>
</select> <select name="pick_month">   <option value="1">January</option>
<option value="2">February</option>     <option value="3">March</option>
<option value="4">April</option>         <option value="5">May</option>
<option value="6">June</option>          <option value="7">July</option>
<option value="8">August</option>      <option value="9">September</option>
<option value="10">October</option>   <option value="11">Nowember</option>
<option value="12">December</option> </select> <select name="pick_year"
id="select18">
  <? $last=$year+10;
for($y=$year; $y<=$last; $y++)
{   echo "<option value=\"$year\">$year</option>";
$year=$year+1;          } ?>
</select></td>    </tr> <tr ><strong>Pick up time </strong> </td> <td><select
name="pick_time">
```

```
<option value="0">00:00</option><option value="1">01:00</option>
<option value="2">02:00</option><option value="3">03:00</option>
<option value="4">04:00</option><option value="5">05:00</option>
<option value="8">08:00</option><option value="9">09:00</option>
<option value="10">10:00</option><option value="11">11:00</option>
<option value="12">12:00</option><option value="13">13:00</option>
<option value="14">14:00</option><option value="15">15:00</option>
<option value="16">16:00</option><option value="17">17:00</option>
<option value="18">18:00</option><option value="19">19:00</option>
<option value="20">20:00</option><option value="21">21:00</option>
<option value="22">22:00</option><option value="23">23:00</option>
</select></td>  </tr>   <tr align="left" valign="top">
<td>Pick  up Location</td>
<td><select name="pick_loc"> <option value="Magosa">Magosa</option>
<option value="Girne">Girne</option><option value="Lefkosa">Lefkoþa</option>
<option value="Lefke">Lefke</option><option value="Iskele">Iskele</option>
</select> <br><br><br></td></tr><tr align="left" valign="top">
<td>Drop off Date</td><td><select name="drop_day" id="select11">
<? for($dg=1; $dg<=31; $dg++){     echo "<option value=\"$dg\">$dg</option>";
}   ?></select> <select name="drop_month">
<option value="1">January</option>
<option value="2" selected>February</option>
<option value="3">March</option><option value="4">April</option>
<option value="5">May</option><option value="6">June</option>
<option value="7">July</option><option value="8">August</option>
<option value="9">September</option><option value="10">October</option>
<option value="11">Nowember</option><option value="12">December</option>
</select> <select name="drop_year" id="select13">
<? $son=2016;
for($dy=2006; $dy<=$son; $dy++)
{  echo "<option value=\"$dy\">$dy</option>";    }                      ?>
</select></td>  </tr> <tr align="left" valign="top">
<td><font color="#333366" size="1"><strong>Drop off Time</strong></font></td>
<td><select name="drop_time">
<option value="0">00:00</option><option value="1">01:00</option>
<option value="2">02:00</option><option value="3">03:00</option>
<option value="4">04:00</option><option value="5">05:00</option>
<option value="6">06:00</option><option value="7">07:00</option>
<option value="8">08:00</option><option value="9">09:00</option>
<option value="10">10:00</option><option value="11">11:00</option>
<option value="12">12:00</option><option value="13">13:00</option>
<option value="14">14:00</option><option value="15">15:00</option>
<option value="16">16:00</option><option value="17">17:00</option>
<option value="18">18:00</option><option value="19">19:00</option>
<option value="20">20:00</option><option value="21">21:00</option>
<option value="22">22:00</option><option value="23">23:00</option>
```

```
</select></td></tr>  <tr > <td><div align="left"><font color="#333366"
size="1"><strong>Drop off location </strong></font></div></td>  <td ><select
name="drop_loc"><option value="Magosa">Magosa</option><option
value="Girne">Girne</option><option value="Lefkosa">Lefkoþa</option><option
value="Lefke">Lefke</option><option value="Iskele">Iskele</option></select> </td>
</tr><tr > <td>&nbsp;</td><td><br> <input type="submit" name="Submit"
value="Next step &gt;&gt;"></td></tr>  <tr > <td ></td> </tr></form></table></td>
</tr></table></td> </tr><tr > <td >&nbsp;</td><td></td>  </tr> </table></td><td>
<table>  <tr>  <td> Car Collection</td> </tr>  <tr><td >&nbsp; </td> </tr>      <?
$query_car = mysql_query("select * from car order by car_no desc");
$num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{  $car = mysql_fetch_array($query_car);   $model=$car['car_model'];
  $name=$car['car_name'];  $no=$car[car_no];            ?>       <tr>
<td ><? echo "<img src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name
$model <br><br><br></a>"; ?>          </td>       </tr>       <?                    }
          ?>
<tr>     <td >&nbsp;</td></tr>  <tr>
<td >&nbsp; </td>    </tr> </table></td>
  </tr>  <tr >
   </body></html>
```

7.1.2 Reservation.php

```
<? }  else
{  echo "<a href='signout.php'>Sign out </a>"; }       ?>
</td><td  ><font size="4" >Farqin
Rent A Car </font></td></tr></table></td></tr><tr>  <td >
<tr><table ><tr>
<td >Rent A Car</td></tr><tr> <td><img src="images/ok.GIF" ><img
src="images/ok.GIF" > <a href="index.php"><font size="2" >Home
Page</font></a></td></tr><tr><td> <img src="images/ok.GIF><img
src="images/ok.GIF> <a href="reservation.php?step=1"><font size="2" face="Times
New Roman, Times, serif">Car Reservation</font></a></td>
</tr><tr> <td><img src="images/ok.GIF" ><img src="images/ok.GIF" ><a
href="car.php"><font size="2" face="Times New Roman, Times, serif">Car Collection
</font></a></td></tr><tr> <td><img src="images/ok.GIF" ><img src="images/ok.GIF"
width="5" height="9">
<a href="about_us.php"><font size="2" face="Times New Roman, Times, serif">About
Us </font></a></td></tr><tr> <td><img src="images/ok.GIF"><img
src="images/ok.GIF" >
<a href="contact.php"><font size="2" face="Times New Roman, Times, serif">Contact
Us</font></a></td> </tr><tr><td >&nbsp;</td></tr><tr> <td >&nbsp; </td>
</tr></table></td></tr></table> </td><td ><table > <tr><td>
<?
if($step==1)  {  ?> <br><br><br>
<table>
```

```
<form name="form2" method="post" action="reservation.php?step=2">
<tr><td bgcolor="#7476A0"> <table> <td><strong> Reservation </font></strong>
</div></td></tr><tr > <td >&nbsp;</td><td >&nbsp;</td></tr>
<tr > <td> Pick up date</td> <td><select name="pick_day" id="pick_day">
<?
for($g=1; $g<=31; $g++)
{ echo "<option value=\"$g\">$g</option>"; }
?>
</select> <select name="pick_month"><option value="1">January</option>
<option value="2">February</option><option value="3">March</option>
<option value="4">April</option><option value="5">May</option>
<option value="6">June</option><option value="7">July</option>
<option value="8">August</option><option value="9">September</option>
<option value="10">October</option><option value="11">Nowember</option>
<option value="12">December</option></select> <select name="pick_year"
id="select18">
<? $last=$year+10;
for($y=$year; $y<=$last; $y++)
{ echo "<option value=\"$year\">$year</option>"; - $year=$year+1;
} ?>     </select></td></tr><tr align="left" valign="top">
<td><strong>Pick up time</strong></td><td> <select name="pick_time">
<option value="0">00:00</option><option value="1">01:00</option>
<option value="2">02:00</option><option value="3">03:00</option>
<option value="4">04:00</option><option value="5">05:00</option>
<option value="6">06:00</option><option value="7">07:00</option>
<option value="8">08:00</option><option value="9">09:00</option>
<option value="10">10:00</option><option value="11">11:00</option>
<option value="12">12:00</option><option value="13">13:00</option>
<option value="14">14:00</option><option value="15">15:00</option>
<option value="16">16:00</option><option value="17">17:00</option>
<option value="18">18:00</option><option value="19">19:00</option>
<option value="20">20:00</option><option value="21">21:00</option>
<option value="22">22:00</option><option value="23">23:00</option>
</select></td>
</tr><tr align="left" valign="top"> <td><strong>Pick up Location
</strong></td><td><select name="pick_loc"><option value="Magosa">
Magosa</option><option value="Girne">Girne</option><option value="Lefkosa">
Lefkoþa</option><option value="Lefke"> Lefke</option>
<option value="Iskele">Iskele</option></select> <br> <br> <br> </td>
</tr><tr align="left" valign="top">td><font color="#7476A0" size="1"><strong>Drop
off Date</strong></font></td>
<td><select name="drop_day" id="select11">
<?  for($dg=1; $dg<=31; $dg++)
{  echo "<option value=\"$dg\">$dg</option>"; } ?>
</select> <select name="drop_month">
<option value="1">January</option><option value="2">February</option>
```

```
<option value="3">March</option><option value="4">April</option>
<option value="5">May</option><option value="6">June</option>
<option value="7">July</option><option value="8">August</option>
<option value="9">September</option><option value="10">October</option>
<option value="11">Nowember</option><option
value="12">December</option></select> <select name="drop_year" id="select13">
<? $son=2016;
for($dy=2006; $dy<=$son; $dy++)
{ echo "<option value=\"$dy\">$dy</option>";       }
?>
</select></td> </tr> <tr align="left" valign="top">
<td><font color="#7476A0" size="1"><strong>Drop off Time</strong></font></td>
<td><select name="drop_time">
<option value="0">00:00</option><option value="1">01:00</option>
<option value="2">02:00</option><option value="3">03:00</option>
<option value="4">04:00</option><option value="5">05:00</option>
<option value="6">06:00</option><option value="7">07:00</option>
<option value="8">08:00</option><option value="9">09:00</option>
<option value="10">10:00</option><option value="11">11:00</option>
<option value="12">12:00</option><option value="13">13:00</option>
<option value="14">14:00</option><option value="15">15:00</option>
<option value="16">16:00</option><option value="17">17:00</option>
<option value="18">18:00</option><option value="19">19:00</option>
<option value="20">20:00</option><option value="21">21:00</option>
<option value="22">22:00</option><option value="23">23:00</option>
</select></td></tr> <tr align="center" valign="top">
<td><div align="left"><font color="#7476A0" size="1"><strong>Drop
off location</strong></font></div></td><td align="left"> <select name="drop_loc">
<option value="Magosa">Magosa</option>
<option value="Girne">Girne</option><option value="Lefkosa">Lefkoþa
</option><option value="Lefke">Lefke</option><option value="Iskele">
Iskele</option></select> </td></tr><tr align="center" valign="top">
<td>&nbsp;</td><td><br> <input type="submit" name="Submit" value="Next step
&gt;&gt;"></td></tr> <tr align="center" valign="top">
<td colspan="2" bgcolor="#7476A0">&nbsp;</td> </tr>
</table></td></tr>
</form></table> <br> <br>
<? }   else if($step==2)
{
$new_reg = mysql_query("insert into reservation values
(NULL,NULL,$pick_day,$pick_month,$pick_year,'$pick_time','$pick_loc',$drop_day,$d
rop_month,$drop_year,'$drop_time','$drop_loc',NULL,NULL,NULL,0,'$ss_id',now())");
?><div align="center"><br><br><table width="88%" border="1" cellpadding="0"
cellspacing="0" bordercolor="#7476A0"><tr> <td> <tr> <td > <div
align="center"><strong><font color="#FFFFFF">Select a car group</font> </strong>
</div></td></tr><tr><td >&nbsp;</td> </tr><tr> <td > <form name="form2"
```

```
method="post" action="reservation.php?step=3"><table ><? $query_group =
mysql_query("select * from car_group");
$num_group=mysql_num_rows($query_group);
for($i=1; $i<=$num_group; $i++)
{  $grp = mysql_fetch_array($query_group); $group=$grp['group_name'];
   $g_no=$grp[group_no];   ?>
<tr align="left" valign="top">
<td width="24%">
<? echo "$group"; ?></td>
<td width="76%"><input name="car_group" type="radio" value="<? echo "$g_no"; ?>"
checked></td> </tr>
<?
}
?>
<tr> <td><br> </td><td><br> <input type="submit" name="Submit2" value="Next Step
&gt;&gt;"></td></tr> </table></form></td></tr><tr > <td></td></tr><tr>
<td>&nbsp;</td></tr></table></td></tr></table></div>
<?
}
else if($step==3)
{
$upd_reg = mysql_query("update reservation set car_group = $car_group where
ss_id='$ss_id' ");
?> <div align="center"><table><tr> <td> <div align="center"><font
color="#FFFFFF"><strong>Select a Car </strong></font></div></td>
</tr><tr> </tr><tr align="left" valign="top"> <td height="47"> <table width="100%"
border="0" cellspacing="0" cellpadding="0">
<form name="form2" method="post" action="reservation.php?step=4">
<?
$query_model = mysql_query("select * from car where group_no = $car_group AND
reserv != 'y' ");$num_car=mysql_num_rows($query_model);
for($c=1; $c<=$num_car; $c++)
{
  $car = mysql_fetch_array($query_model);  $name=$car['car_name'];
    $model=$car['car_model']; $cm=$name."-".$model;
            $c_no=$car[car_no];                         ?>
<tr align="left" valign="top"> td width="24%"><? echo "$name $model"; ?></td><td
width="76%">resim <input name="car_no" type="radio" value="<? echo "$c_no";
?>"></td></tr>
<?
}
?>
<tr> <td><input name="group_no" type="hidden" value="<? echo "$car_group";
?>"></td>
<td><br> <input type="submit" name="Submit22" value="Next Step
&gt;&gt;"></td></tr></form></table></td> </tr><tr align="left">
```

```
<td valign="top"></td></tr>
</table>
<p><br>
<?
} else if($step==4)
{ $new_reg = mysql_query("update reservation set car_no = $car_no where
ss_id='$ss_id' ");
$query_price= mysql_query("select * from car_group where group_no = $group_no ");
$pricee = mysql_fetch_array($query_price);
  $price=$pricee[price];
  $query_rent_days= mysql_query("select * from reservation  where ss_id='$ss_id' ");
$days = mysql_fetch_array($query_rent_days);
  $p_day=$days[pick_day];    $d_day=$days[drop_day];
  $total_day = $d_day - $p_day;
  $total_rent_price=$total_day * $price;                ?>
</p>
<table"><tr> <td> Login</td> </tr><tr><td>&nbsp;</td></tr><table ><form
name="form2" method="post" action= "reservation.php?step=5"> <tr > <td
width="24%"> Username</td> <td ><input type="text" name="username"></td></tr><tr
align="left" valign="top"> <td width="24%">Password</td> <td > <input
type="password" name="user_pass"></td> </tr><tr align="left" valign="top">
<td>&nbsp;</td><td><br> <input name="Submit2222" type="submit" value="Next
Step"></td></tr></form></table></td></tr><tr align="left"><td > </td> </tr><tr> <td
>&nbsp;</td> </tr></table>
<?
} else if($step==5)
{  if(!is_user())
{
?>
<table ><tr> <td ><div align="center">Price </div></td></tr><tr><td >&nbsp;</td>
</tr><tr align="left" valign="top"> <td height="47"> <table><form name="form2"
method="post" action="reservation.php?step=6"><tr > <td>toplam bilgiler</td><td
width="76%">Total Price : <? echo "per day $price <br> total $total_rent_price";
?></td> </tr><tr align="left" valign="top">
<td><? echo "$group_no"; ?></td><td><br> <input type="submit" name="Submit222"
value="  Finish Reservation "></td></tr></form> </table></td> </tr> <tr align="left">
<td></td> </tr> <tr> <td>&nbsp;</td> </tr> </table> <br>
<? }else //user or not function end
{  global $valid_user;
echo "you rae user and reservation is accepted<br><br>";
$comp= mysql_query("update reservation set complete = 'y' ,user = '$valid_user' where
ss_id='$ss_id' ");
if($comp)  echo "successfull";
} } else if($step==6)
{ if (user_login($username, $user_pass))
{$valid_user=$username; session_register("valid_user");
echo "<meta http-equiv='refresh' content='3; URL=reservation.php?step=5'>";  }  else {
echo "<font color='red'> mERROR!!!..</font><br>"; echo "You are entered wrong
username or password..please try again!!! ";  exit;  } }
else
{ echo "error taskkk";}          ?> </td> </tr> </table>
```

7.1.3 About_us.php

```
   <?
$query_car = mysql_query("select * from car order by car_no desc");
        $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{
 $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
?>
```

7.1.4 Rentcar_funtion.php

```
<?
 include("database.php");
$connect = rentcar_database();
//----user login
function user_login($username, $user_pass)
{  $result = mysql_query("select * from user  where username='$username'  and
user_pass ='$user_pass'");
 if (!$result)
    return 0;
   if (mysql_num_rows($result)>0)
    return 1;
 else
    return 0;
}
//----check_user
function check_user()
{ global $user;
 if (!session_is_registered("user"))
 {$err_login = "ERROR..You are not user.you must login to see this page .to login enter
your username and password";
return $err_login;
  } }
//----user_register
```

```
function
new_user_register($user_id,$username,$user_surname,$user_pass,$user_mail,$user_tel,$
user_address,$driver_licence)
{  $new_reg = mysql_query("insert into user values
(NULL,'$user_id','$username','$user_surname','$user_pass','$user_mail','$user_tel','$user
_address','$driver_licence',now())");
 if(!$new_reg)
return false;
return true;
}
//----is_user
function is_user()
{  global $user;
  if (session_is_registered("user"))
{return true;
}else
{return false;
} }
?>
```

7.1.5 Car.php

```
<?  if (!$car_no)
{        ?>
<strong><font color="#CCCC00">Select a Car</font></strong><br>
<?  $query_car2 = mysql_query("select * from car order by car_no desc");
         $num_car2=mysql_num_rows($query_car2);
for($i=1; $i<=$num_car2; $i++)
{  $car2 = mysql_fetch_array($query_car2);
  $model2=$car2['car_model'];
  $name2=$car2['car_name'];
  $no2=$car2[car_no];
echo "<img src='images/$no2.gif'><br><a href='car.php?car_no=$no2'>$name2 $model2
<br><br><br></a>";
} }
else if($car_no > 0)
{      $selected_car = mysql_query("select * from car where car_no=$car_no");
  $s_car = mysql_fetch_array($selected_car);
  $f_name=$s_car['car_name'];
  $f_model=$s_car['car_model'];
  $f_number_plate=$s_car['car_numberplate'];
  $f_car_property=$s_car['car_property'];
  $f_car_insurance=$s_car['car_insurance_price'];
  $f_car_picture=$s_car['picture'];
  $f_reserve=$s_car['reserv'];
  if ($f_reserve == 'y')
  $state="This Car is Reserved - Not available";
```

```
 else
 $state = "Available";
?> <table> <tr> <td><table ><tr><td > Number Plate :</td><td><?  echo
"$f_number_plate";?></td> <td >&nbsp; </td>      </tr><tr> <td> >Name:</td><td><?
echo "$f_name";?>  </td><td>&nbsp; </td><td>&nbsp; </td></tr><tr>
<td><strong><font size="2" >Model: </font></strong></td><td> <? echo "$f_model";?>
</td><td>&nbsp; </td><td>&nbsp;</td></tr><tr><td>Car Properties :</td>
<td><?  echo "$f_car_property";?></td><td>&nbsp;</td><td>&nbsp;</td>
</tr><tr><td>Car Insurance :</td><td> <?  echo "$f_car_insurance";?>
</td><td>&nbsp;</td><td>&nbsp;</td></tr><tr> <td>Reservation:</td>
<td><?  echo "$state"; ?> </td><td>&nbsp;</td><td>&nbsp;</td></tr>
<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td>  <td>&nbsp;</td></tr><tr > <td
colspan="4"><?  echo "<img src='images/$car_no.gif'>";?></td></tr></table></td></tr>
</table> <?  }?>
<?
$query_car = mysql_query("select * from car order by car_no desc");
        $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{
 $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
?>
<tr>
<td height="22" align="center" valign="top"><? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?> </td></tr>                    <?      }           ?>
```

7.1.6 Contact.php

```
<?
}else
{echo "<a href='signout.php'>Sign out </a>";
}
?>
</td><td > Farqin
Rent A Car </td></table></td></tr> <tr > <td ><table ><tr><td><table <tr><td>
<strong>  Rent A Car</strong><</td></tr><tr><td><img src="images/ok.GIF" ><img
src="images/ok.GIF" ><a href="index.php">Home Page</a></td></tr><tr><td> <img
src="images/ok.GIF" ><img src="images/ok.GIF">a href="reservation.php?step=1">Car
Reservation</a></td></tr><tr> <td><img src="images/ok.GIF" ><img
src="images/ok.GIF" > <a href="car.php"> Car Collection </a></td></tr><tr><td><img
src="images/ok.GIF" ><img src="images/ok.GIF" ><a href="about_us.php>About Us
</a></td></tr> <tr><td><img src="images/ok.GIF" ><img src="images/ok.GIF" ><a
href="contact.php"> ContactUs</a></td></tr>
```

```
<tr><td ><strong>Name :</strong></td><td ><input type="text"  name="textfield">
</td><td>&nbsp;</td></tr><tr> <td ><strong>Surname :</strong></td> <td> <input
type="text" name="textfield2"></td><td>&nbsp;</td></tr><tr> <td ><strong>E-mail
:</strong></td><td><input type="text"  name="textfield3">
</td><td>&nbsp;</td></tr><tr> <td ><strong>Adres :</strong></td><td><input
type="text" name="textfield4"></td><td>&nbsp;</td>       </tr><tr>
<td><strong>Explains :</strong></td>
<td><textarea name="textarea" cols="30"></textarea></td>
<td>&nbsp;</td></tr><tr align="center" > <td colspan="3"><input type="submit"
name="Submit" value="Send"></td></tr></form></table> /td></tr></table></div></td>
<td > <tr> <td height="22" align="center" valign="top"><strong>Car
Collection</strong></td></tr>
<?
$query_car = mysql_query("select * from car order by car_no desc");
        $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{
 $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
?>
<tr> <td height="22" align="center" valign="top"> <? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?>
</td></tr>      <?      }  ?>
```

7.1.7 Database.php

```
<?
function rentcar_database()
{  $rentcar_connect = mysql_pconnect("localhost", "firat", "silinir");
  if (!$rentcar_connect)
    return false;
  mysql_select_db("rentcar");
  return $rentcar_connect;
}
?>
<?
require_once("rentcar_function.php");
?>
```

7.1.8 Forget _password.php

```
<?
$query_car = mysql_query("select * from car order by car_no desc");
```

```
        $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{
 $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
?>
<tr>
      <td height="22" align="center" valign="top"> <? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?>   </td>  </tr>
<?
}
?>
```

7.1.9 New_user.php

```
<?
if(new_user_register($user_id,$username,$user_surname,$user_pass,$user_mail,$user_te
l,$user_address,$driver_licence))
{echo "<br><br>"; echo "<center>Your registration successfull</center>";
}else
{
echo "<br><br>";
echo "<center>ERROR<br>no registration</center>";
} ?>
</td><td > <tr> <td ><strong>Car Collection</strong></td></tr>
<?$query_car = mysql_query("select * from car order by car_no desc");
$num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{  $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
?>
<tr> <td height="22" align="center" valign="top"> <? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?> </td> </tr>
<?
}
?>
```

7.1.10 Register.php

```
        <?
$query_car = mysql_query("select * from car order by car_no desc");
            $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{
  $car = mysql_fetch_array($query_car);
  $model=$car['car_model'];
  $name=$car['car_name'];
  $no=$car[car_no];
?>
<tr> <td height="22" align="center" valign="top"> <? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?></td></tr>
<?
}
?>
```

7.1.11 Signout.php

```
<?
require_once("rentcar_function.php");
session_start();
$user_delete = $user;
$delete_session = session_unregister("user_delete");
$user_dest = session_destroy();
if (empty($user_delete))
{ echo "You Are Not User";
 }else
{if ($user_delete && $user_dest)
  {    echo "You are logout successfulll<br>";
echo "<meta http-equiv='refresh' content='3;URL=index.php'>";
}  else
  {    echo "ERROR!!! You did not logout!!!.<br>";
  }
}
?>
```

7.1.12 User.php

```
        <?
$query_car = mysql_query("select * from car order by car_no desc");
            $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{  $car = mysql_fetch_array($query_car);
  $model=$car['car_model'];
```

```
 $name=$car['car_name'];
 $no=$car[car_no];
?>
      <tr>
<td height="22" align="center" valign="top"><? echo "<img
src='images/$no.gif'><br><a href='car.php?car_no=$no'>$name $model
<br><br><br></a>"; ?> </td>    </tr>
      <?
               }
               ?>
```

7.2. Administration Interface

7.2.1 İndex.php

```
<?
 session_start();
?>
<html><head><title>Farqin Rent A Car Admin Service</title></head> <body > <table>
<tr><td ><tr><td ><div align="center"><p><strong><font  size="5">Farqin Rent a Car
</font><font color="#666699"><b><br><font size="3">The International Car Rental
Specialists</font></b></font></strong>ADMINISTRATION PANEL <strong>Admin
Password :</strong></font></td><td ><input name="admin_pass" type="password"
size="20"></td></tr><tr > <td colspan="2"><br> <input type="submit" name="Submit"
value="  Enter Admin Panel  "></td></tr></form></html>
```

7.2.2 Admin.php

```
<?
require_once("all_admin_function.php");
session_start();
$conn = db_connect();
if ($admin_name && $admin_pass)
{   if (admin_login($admin_name, $admin_pass))
{$valid_admin=$admin_name;
session_register("valid_admin");
} else
{echo "<font color='red'>ERROR!!!..</font><br>";
echo "You are enter wrong admin name or password..please try again";
exit;   }  }
check_admin(); global $admin_name; $task_query=mysql_query("select * from
administrator where admin_name = '$valid_admin' ");
$taskk=mysql_fetch_array($task_query);
$task_no = $taskk['task'];
$isim = $taskk['admin_pass'];
?>

<html><head><title>Farqin Rent A Car Administrator Service</title>
```

```
<link href="stil/style.css" rel="stylesheet" type="text/css">
</head><body><table><tr><td ><table ><tr><td ><div align="center">
<p><strong>Farqin Rent a Car <br>The International Car Rental
Specialists</font></b></font></strong><br><br>
<font color="#CC9900"><strong>ADMINISTRATION PANEL</strong></font></p>
</div></td><td ><img src="images/car.jpg" ></td></tr></table></td></tr></table>
<table >  <tr><td><table >
      <?
           if($task_no == '1')
           {
           ?>
<tr><td > <strong><img ="images/foto1.jpg"></strong></td> <td > <strong> Cars
</strong></td><td><a href="car.php">[insert]</a></td><td ><a
href="car.php?task=2">[Update]</a></td><td ><a href="car.php?task=4"> [Delete]
</a></td><td >&nbsp;</td></tr><tr><td >&nbsp;</td><td > <strong>Car Groups
</strong></td><td > <a href="group.php">[insert]</a></td><td ><a
href="group.php?task=2">[Update]</a></td><td ><a href="group.php?task=4">
[Delete]</a></td><td>&nbsp;</td>
</tr>
<?
         }
?>
      <tr> <img src="images/rezerve01.jpg" ></td><td> <strong> Reservations
</strong></td><td><a href="reservation.php?task=show">[Show]</a></td><td
>&nbsp;</td><td><a href="reservation.php?op=4"> [Delete]</a></td><td
>&nbsp;</td></tr><tr ><td ><table ><tr>
<td >Administrator : <? echo "$valid_admin";  ?></td><td ><a
href="admin_logout.php"> <img src="images/logout.gif" ></a><br><a
href="admin_logout.php"> Logout</a></td></tr></table></td></tr></table>
</body></html>
```

7.2.3 Admin_function.php

```
<?
require_once("database.php");
$conn = db_connect();
//----admin_login
function admin_login($admin_name, $admin_pass)
{ $result = mysql_query("select * from administrator  where
admin_name='$admin_name'  and admin_pass ='$admin_pass'");
  if (!$result)
    return 0;
   if (mysql_num_rows($result)>0)
    return 1;
  else
    return 0;
}
```

```
//----check_admin
function check_admin()
{  global $valid_admin;
  if (session_is_registered("valid_admin"))
  {
  }
  else
  {  echo"<font size='5' color='red'>ERROR</font><br>";
     echo "Please enter both admin name and password.!.<br>";
         exit;
  }
}
//---- insert_car
function
insert_car($car_name,$car_model,$group_no,$car_numberplate,$car_property,$drive_lic
ence,$car_insurance_price,$picture)
{$car_insert_query=mysql_query("insert into car values
 (
NULL,'$car_name','$car_model',$group_no,'$car_numberplate','$car_property','$drive_li
cence',$car_insurance_price,'$picture','',now() )");
if (!$car_insert_query)
return false;
return true;
}//----new_group
function new_roup($group_name,$price)
{$group_query=mysql_query("insert into car_group values (
NULL,'$group_name',$price )");
if (!$group_query)
return false;
return true;
}  ?>
```

7.2.4 Admin_logout.php

```
<?
require_once("all_admin_function.php");
session_start();
$old_admin = $valid_admin;
$admin_del = session_unregister("old_admin");
$admin_dest = session_destroy();
if (!empty($old_admin))
{ if ($admin_del && $admin_dest)
  {
   // if they were logged in and are now logged out
echo "You are logout successfulll<br>";
echo "<meta http-equiv=\"refresh\" content=\"3;URL=http://localhost/rentcar\">";

} else
```

```
 {  echo "ERROR!!! YOu did not logout!!!.<br>";
include("yonetici.php");
 }
}
else
{ echo "<font size='6' color='red'>You are not enter so you did not
logout!!!.<br></font>";
}
?>
```

7.2.5 All_admin_funcion.php

```
<?
  require_once("database.php");
  require_once("admin_function.php");

?>
```

7.2.6 Car.php

```
<?
require_once("all_admin_function.php");
session_start();
check_admin();
$conn = db_connect();
?>
<?
if(!$op)
  {
            ?>
<table ><form name="form" method="post" action="car.php?op=1" onsubmit="return
validator(this)"><tr> <td ><strong>Name:</strong></td><td> <input
name="car_name2" type="text" id="car_name2" value="" size="40"><br></td> </tr><tr
><td><strong>Model :</strong></td><td > <input name="car_model2" type="text"
id="car_model2" value="" size="40"><br> </td></tr><tr><td ><strong>Group
:</strong></td><td > <select name="select">
<?$query_group = mysql_query("select * from car_group");
           $num_group=mysql_num_rows($query_group);
for($i=1; $i<=$num_group; $i++)
{  $grp = mysql_fetch_array($query_group);
  $group=$grp['group_name'];
  $no=$grp[group_no];
 echo " <option value=\"$no\"> $group</option>";
                    }
?>
```

```
</select> <br> </td></tr><tr ><td ><strong>Number Plate:</strong></td><td > <input
name="car_numberplate2" type="text" id="car_numberplate2" size="40">
<br> </td></tr><tr><td ><strong>Properties:</strong></td><td> <textarea
name="textarea" cols="40" rows="10" id="textarea"></textarea><br> </td> </tr> <tr
><td > <strong>Require drive Licence Class:</strong></td><td ><input
name="drive_licence2" type="text" id="drive_licence2" value="" size="40"><br> </td>
</tr><tr><td ><strong>Insurance Price:</strong></td><td> <input
name="car_insurance_price2" type="text" id="car_insurance_price2" value=""
size="40"><br> </td></tr><tr><td ><strong>Picture: </strong></td><td> <input
name="picture" type="text"  value="" size="40"><br> </td></tr><tr><td><input
type="submit" name="Submit2" value="Insert Car"> </td><td> [<a href="admin.php">
Admin Home Page</a>] [<a href="car.php">Insert Car</a>]<br>[<a
href="car.php?op=2"> Car Update</a>] [<a href="car.php?op=4">Car Delete</a>]
</td></tr></form></table>
<?      }
          else if($op==1)
 {
if(insert_car($car_name,$car_model,$group_no,$car_numberplate,$car_property,$drive_
licence,$car_insurance_price,$picture))
{echo "<br><br><br><center><b>New Car inserted </b><br> <br><br><br>
<br><br></center>";echo "<meta http-equiv='refresh' content='3; URL=car.php'>";
}else
{echo "ERROR !!! car did nor insert";
echo "<meta http-equiv='refresh' content='3; URL=car.php'>";
} }       else if($op==2)
          {
 //edit car
if(!$car_no)
{echo "<center><b>Car Update</center></b><br><br>";
$query_car = mysql_query("select * from car order by car_no desc");
        $num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{  $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $no=$car[car_no];
echo "<a href='car.php?op=2&car_no=$no'>$model $name<br><br></a>";
}}
else
{$query_car = mysql_query("select * from car where car_no=$car_no");
        $num_car=mysql_num_rows($query_car);
 $car = mysql_fetch_array($query_car);
 $model=$car['car_model'];
 $name=$car['car_name'];
 $car_numberplate=$car['car_numberplate'];
 $car_property=$car['car_property'];
 $drive_licence=$car['drive_licence'];
 $car_price=$car[car_price];
 $car_insurance_price=$car[car_insurance_price];
?>
<form name="form" method="post" action="car.php?op=3&car_no=<? echo "$car_no";
?>" ><table ><tr> <td><strong>Name:</strong></td><td > <input name="car_name"
type="text" value="<? echo "$name"; ?>" size="40"><br> </td></tr><tr > <td
><strong>Model :</strong></td> <input name="car_model" type="text" id="car_model"
value="<? echo "$model"; ?>" size="40"><br> </td></tr><tr > <td ><strong> Group
:</strong></td><td valign="top" class="admin"> <select name="group_no"> <?
        $query_group = mysql_query("select * from
car_group");$num_group=mysql_num_rows($query_group);
for($i=1; $i<=$num_group; $i++){  $grp = mysql_fetch_array($query_group);
 $group=$grp['group_name'];  $no=$grp[group_no];   echo " <option value=\"$no\">
$group</option>";     }         ?>
</select> <br> </td> </tr><td ><strong> Number Plate:</strong></td> <td > <input
name="car_numberplate" type="text" id="car_numberplate" value="<? echo
"$car_numberplate"; ?>" size="40"><br> </td></tr><tr > <td><strong>
Properties:</strong></td><td > <textarea name="car_property" cols="40" rows="10"
id="textarea4"><? echo "$car_property"; ?></textarea><br> </td></tr> <tr ><td
><strong>Require drive Licence Class:</strong></td><td class="admin"> <input
name="drive_licence" type="text" id="drive_licence" value="<? echo "$drive_licence";
?>" size="40"><br> </td></tr><tr ><td ><strong>Insurance Price:</strong></td><td >
<input name="car_insurance_price" type="text" id="car_insurance_price" value="<?
echo "$car_insurance_price"; ?>" size="40"><br> </td></tr><tr >
<td ><input type="submit" name="Submit" value="Update Car"></td> <td
class="admin"> [<a href="index.php"> Admin Home Page</a>] [<a
href="car.php">Insert Car</a>] <br>
[<a href="car.php?op=2"> Car Update</a>] [<a href="car.php?op=4">Car
Delete</a>] </td> </tr></table>  </form>
   <?
}//op 2 finish..
}else if($op==3)
{ //edit fonksiyonlarý
$update_group=mysql_query("update car set car_name='$car_name' ,
car_model='$car_model' ,group_no=$group_no ,car_numberplate='$car_numberplate' ,
car_property='$car_property' , drive_licence='$drive_licence'
,car_insurance_price=$car_insurance_price where car_no=$car_no ");
if ($update_group)
{echo "<br><br><br><center><b>Car Group
Edited</b><br><br><br><br><br><br></center>";
echo "<meta http-equiv='refresh' content='3; URL=car.php?op=2'>";
}else
{echo "ERROR!!!!";
}// op 3 finish
}else if($op==4)
```

```
{ // functions of deleting.........
if(!$car_no)
{echo "<center><b>Car Delete</center></b><br><br>";
$query_car = mysql_query("select * from car order by car_no desc");
$num_car=mysql_num_rows($query_car);
for($i=1; $i<=$num_car; $i++)
{  $car = mysql_fetch_array($query_car);
  $model=$car['car_model'];
  $name=$car['car_name'];
  $no=$car[car_no];
echo "<a href='car.php?op=4&car_no=$no'>$model $name<br><br></a>";
}}
else
{$delete_group=mysql_query("delete from car where car_no=$car_no");
if($delete_group)
{echo "<br><br><br><center><b>Deleted
Car</b><br><br><br><br><br><br></center>";
echo "<meta http-equiv='refresh' content='3; URL=car.php?op=4'>";
}else
{echo "ERROR...no delete";
}}
// op 4 panel delete finish
}else
{echo "Wrong Command... Please Wait";
echo "<meta http-equiv='refresh' content='3; URL=panel.php?op=2'>";
}        ?>
   </td>  </tr></table></body></html>
```

7.2.7 Database.php

```
<?
function db_connect()
{   $result = mysql_pconnect("localhost", "firat", "silinir");
  if (!$result)
     return false;
  if (!mysql_select_db("rentcar"))
     return false;
  return $result;
}
?>
```

7.2.8 Group.php

```
<?
if(!$task)
```

```
 {
 ?>
<form name="form" method="post" action="group.php?task=1"><table><tr> <td>
<strong>  Name:</strong></td><td > <input ame="group_name" type="text" value=""
size="30"> <br> </td></tr><tr> <td><strong>Prica Per Day: </strong></td><td > <input
name="price" type="text" id="price" value="" size="30"> <br> </td></tr><tr > <td
><input type="submit" name="Submit2" value="Create New Group"></td><td
class="admin"> [<a href="admin.php"> Admin Home Page</a>] [<a href="group.php">
Insert Group</a>]  [<a href="group.php?task=2"> Group Update</a>] [<a
href="group.php?task=4"> Group Delete</a>] </td></tr> </table>   </form>
    <?
            }
            else if($task==1)
            {
//
if(new_roup($group_name,$price))
{echo "<br><br><br><center><b>New group Created
</b><br><br><br><br><br><br></center>";
echo "<meta http-equiv='refresh' content='3; URL=group.php'>";
}else
{echo "ERROR !!! car did nor insert";
echo "<meta http-equiv='refresh' content='3; URL=group.php'>";
} }        else if($task==2)
            {       if(!$group_no)
{echo "<center><b>Car Update</center></b><br><br>";
$query_group = mysql_query("select * from car_group");
       $num_group=mysql_num_rows($query_group);
for($i=1; $i<=$num_group; $i++)
{  $grp = mysql_fetch_array($query_group);
 $group=$grp['group_name'];
 $no=$grp[group_no];
echo "<a href='group.php?task=2&group_no=$no'>$group<br><br></a>";
}}      else
{$query_gr = mysql_query("select * from car_group where group_no=$group_no");
 $grup = mysql_fetch_array($query_gr);
 $group_name=$grup['group_name'];
  $price=$grup[price];
?>
  <form name="form" method="post" action="group.php?task=3&group_no=<? echo
"$group_no"; ?>" ><table > <tr > <td ><strong>  Name:</strong></td><td > <input
name="group_name" type="text" value="<? echo "$group_name"; ?>" size="40"> <br>
</td> </tr> <tr > <td ><strong>  Prica Per Day:</strong></td>
<td valign="top" class="admin"> <input name="price" type="text" value="<? echo
"$price"; ?>" size="40"> <br> </td> </tr>
<tr > <td class="admin"><input type="submit" name="Submit" value="Update
Car"></td><td > [<a href="index.php"> Admin Home Page</a>] [<a
href="group.php">Insert Group </a>] [<a href="group.php?task=2"> Group Update</a>]
[<a href="group.php?task=4">Group   Delete</a>] </td></tr>
</table>  </form>
     <?
}//op2 finished
}else if($task==3)
{ //function of edit...........
$update_grp=mysql_query("update car_group set group_name='$group_name' , price =
$price where group_no=$group_no ");
if ($update_grp)
{echo "<br><br><br><center><b>Group Name
Edited</b><br><br><br><br><br><br></center>";
echo "<meta http-equiv='refresh' content='3; URL=group.php?task=2'>";
}else
{echo "ERROR!!!!";
}             // op 3 finish
}else if($task==4)
{ //
if(!$group_no)
{echo "<center><b>Car Delete</center></b><br><br>";
$query_group = mysql_query("select * from car_group");
          $num_group=mysql_num_rows($query_group);
for($i=1; $i<=$num_group; $i++)
{  $grp = mysql_fetch_array($query_group);
  $group=$grp['group_name'];
  $no=$grp[group_no];
echo "<a href='group.php?task=4&group_no=$no'>$group<br><br></a>";
}}  else
{$group_d=mysql_query("delete from car_group where group_no=$group_no");
if($group_d)
{echo "<br><br><br><center><b>Deleted Group
Name</b><br><br><br><br><br><br></center>";
echo "<meta http-equiv='refresh' content='3; URL=group.php?tsak=4'>";
}else
{echo "ERROR!!!! not deleted";
}}            // op 4 panel delete finish
}else
{echo "Error.. Please Wait.....";
echo "<meta http-equiv='refresh' content='3; URL=group.php?task=2'>";
}          ?>  </td>  </tr>  <tr align="left" valign="top" bgcolor="ffffcc">
<td colspan="2"><table width="99%" border="0" cellpadding="0" cellspacing="0">
<tr> <td width="52%" align="center" valign="middle"><a
href="admin_logout.php>Administrator: <? echo $valid_admin;
?></font></a></strong></td>  <td > <p><a href="admin_logout.php"><img
src="images/logout.gif" width="48" height="47" border="0"></a></p>
```

```
<p><a href="admin_logout.php"><fontsize="2"> <strong><em>Logout </em>
</strong> </font></a></p></td></tr></table></td></tr></table>
</body></html>
```

7.2.9 Reservation.php

```
   <?
$show_res=mysql_query("select * from reservation where complete = 'y' AND
admin_check = 0 ");
$num_res=mysql_num_rows($show_res);
for($r=1; $r<=$num_res; $r++)
{$reservation=mysql_fetch_array($show_res);
$r_no = $reservation[reserv_no]; $user = $reservation['user'];
$p_d = $reservation[pick_day]; $p_m = $reservation[pick_month];
$p_y = $reservation[pick_year]; $p_t = $reservation['pick_time'];
$p_l = $reservation['pick_loc']; $d_d = $reservation[drop_day];
$d_m = $reservation[drop_month]; $d_y = $reservation[drop_year];
$d_t = $reservation['drop_time']; $d_l = $reservation['drop_loc'];
$group = $reservation['car_group']; $c_no = $reservation[car_no];
$res_time = $reservation[date];
?><tr align="left" valign="top"><td><font size="1"><a
href="show_user_info.php?user_name= <? echo "$user"; ?>" target="_blank"><? echo
"$user"; ?> </a></font></td><td align="center"><font size="2"><? echo "$p_d";
?></font></td><td><font size="2"><? echo "$p_m"; ?></font></td><td><font
size="2"><? echo "$p_y"; ?></font></td><td><font size="2"><? echo "$p_t";
?></font></td><td><font size="2"><? echo "$p_l"; ?></font></td>
<td><font size="2"><? echo "$d_d"; ?></font></td><td><font size="2"><? echo
"$d_m"; ?></font></td><td><font size="2"><? echo "$d_y"; ?></font></td><td><font
size="2"><? echo "$d_t"; ?></font></td><td><font size="2"><? echo "$d_l";
?></font></td><td><font size="2"><? echo "$group"; ?></font></td><td><font
size="2"><? echo "$c_no"; ?></font></td><td><font size="2"><? echo "$res_time";
?></font></td><td><a href="reservation.php?task=1&reserv_no=<? echo
"$r_no&car_no=$c_no"; ?>">Accept</a></td>    </tr>
        <?        }  ?> </table>
 <?         } else if($task==1) {echo "checkin iþlemi yapýlacak.. no =  $car_no";
$upd_res = mysql_query("update car set reserv = 'y' where car_no = $car_no ");$upd_res
= mysql_query("update reservation set admin_check = 1 where car_no = $car_no AND
reserv_no = $reserv_no ");
         if( $upd_res)
                  {  echo "car reserv state changed";
                    echo "<br><br>";
                    }              }
                          ?>
```

CONCLUSION

The web site Project which is called Farqin Rent Car was made to serve people on the web line.There were used several methods that includes technical and marketing dimensions.Tecnical methods were formed from some web developing tools and programs such as PHP, Apache, and MySQL. All of these tools and components have been discussed and explored throughout the report and after necessary details been presented the steps of the application are figured with screenshots and each screenshot is explained briefly.

Marketing approches have not ignored, when Project formed. Therefore, Project aims to constitute interactiveness, understandable, easiness, easy connection, good governace, good public relations and efficiency by the this web site.In the this framework, home page of web site was tried to be very attractive for both commercial and information operations.

REFERENCES

1. http://www.php.com
2. http://www.mysql.com
3. http://www.apache.com
4. http://www.metatorial.com/
5. http://www.erptoday.com/
6. http://www.contentmanager.eu.com
7. http://www.kosezade.com
8. http://www.netshinesoftware.com