

NEAR EAST UNIVERSITY



Faculty of Engineering

Department of Computer Engineering

Internet Base Publishing

**Graduation Project
COM- 400**

Student : Erdinç Kolukısa

Supervisor : Mr.Ümit Soyer

Nicosia- 2007

ACKNOWLEDGEMENT

"Firstly, I would like to thank to my supervisor Mr Ümit Soyer for his great advise and recommendation for finishing my project properly also, teaching and guiding me in others lectures

I am greatly indepted to my family for their endless support from my starting day in my educational life until today. I will never forget the things that my parents.

Althoug, I encountered many problems in writing program. In that times, My best friends, who helped me during preparing of my project are Müge Günay, Seniha Direl, Mücahit Cihat Bozkurt . My sincere thanks to them and other my friends.

I thank all the staff of the faculty of engineering for giving facilities to practise, teaching and solving problem in my comlete undergraduation program

Finally, I promise to do my best in my life as a bachelor of engineer.

ABSTRACT

Human being is confronting very curicial multy dimensional transformations. In this complicated process, terms such as time, communication, speed, internet information have gained very strategic importance. Our economic vision is being shaped with in this general perspective. In this framework we have chosen media sector to serve people. This is a web page of a weekly newspaper. As mentioned in this global process, we aimed to serve people in the web. We have thought that with this choosing, we can arrive all readers easily.

With this general perspective, project has been formed to serve people on the web line. when this project was constructing, two elements were not ignored. These are technical infrastructure innovations and satisfaction of paper readers. 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.

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

INTRODUCTION

In the internet world the need for fast and reliable web sites become 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 nowadays' 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, 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 of web pages. It is a document written in HTML format that might be describe the content available on the site.

Chapters gives techniques about Apache Web Server, PHP, and MySQL

Chapter 1 : describes Apache Web Server and how configuring

Chapter 2 : describes PHP basics and components

Chapter 3 : gives information about the MySQL and shows the connection with database

Chapter 4 : gives information about design of the pages.

Chapter 5 : gives application codes.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
INTRODUCTION	iii
TABLE OF CONTENTS	iv
CHAPTER ONE : APACHE WEB SERVER	1
1.1 The WWW	1
1.1.1 NCSA	1
1.2 Introduction : What is Apache	1
1.2.1 The Apache Server	2
1.2.2 Apache's Architecture	2
1.2.3 More Recent History	2
1.2.4 The Future of Apache	3
1.2.5 Support For Apache	3
1.3 Obtaining and Installing Apache	4
1.3.1 Hardware / Software Requirements	4
1.3.2 Advanced Installation	5
1.3.3 Editing the Configuration Script	5
1.3.4 Dynamic Shared Objects (DSO's)	6
1.4 Configuring	7
1.4.1 Configuring Files	7
1.4.2 Comanche	7
1.4.3 Starting, Stopping, Restarting	8
1.5 Integrating Apache With The Rest Of Your Business	9
1.5.1 CGI	9
1.5.2 MIME Headers	10
1.5.3 Reading Client Input	10
1.5.4 mod_perl	11

1.5.5	SSI and E-commerce	12
1.5.6	Authentication	12
1.5.7	mod_auth	12
1.5.8	Log Files	14
1.6	A Conclusion For Apache	16
CHAPTER TWO : PERSONAL HOME PAGE—PHP		17
2.1	What is PHP	17
2.2	Writing PHP	17
2.2.1	Basic PHP Syntax	17
2.2.2	Comments in PHP	18
2.2.3	Declaring PHP	18
2.3	Variables	20
2.3.1	Outputting Variables	20
2.3.2	Formatting Text	21
2.4	PHP Operators	21
2.4.1	Arithmetic Operators	22
2.4.2	Assignment Operators	22
2.4.3	Comparison Operators	23
2.4.4	Logical Operators	23
2.5	Conditional Statements	23
2.5.1	If.....Else Statement	24
2.5.2	The Else If Statement	24
2.6	PHP String Processing	25
2.6.1	PHP String Functions	25
2.7	The Mail() Function	28
2.7.1	PHP Simple Text E- mail	28
2.7.2	PHP Mail Form	29
2.7.3	Requirements	30
2.7.4	Runtime Configuration	30
2.7.5	PHP Mail Functions	30
2.8	Cookies	30
2.8.1	How To Create Cookie	30

2.8.2 How To Retrieve a Cookie Value	31
CHAPTER THREE : MySQL	32
3.1 What is MySQL	32
3.2 Database Construction	32
3.2.1 Database and Login	32
3.3 Creating Table	33
3.3.1 Fields	33
3.3.2 Creating Table With PHP	33
3.3.3 The Contacts Database	34
3.4 Connect To MySQL Server	34
3.4.1 Using The Error Control Operator	35
3.4.2 Using The Die Function	35
3.5 Creating Database	35
3.6 Primary Key and Auto Increment	37
3.7 PHP MySQL Function	38
3.7.1 Selecting Database	39
3.7.2 Handling Errors	39
3.8 Inserting Data From A Form To a Database	41
3.9 ODBC With PHP	41
3.9.1 Connecting To an ODBC	42
3.9.2 Retrieving Records	42
3.9.3 Closing An ODBC Connection	43
CHAPTER FOUR : APPLICATION CODES	
4.1 User PHP	
4.1.1 anketal.php	44
4.1.2 ara.php	49
4.1.3 index.php	54
4.1.4 kategori.php	59
4.1.5 uyeol.php	66

4.2 Administrator PHP	
4.2.1 anketekle.php	72
4.2.2 haberekle.php	76
4.2.3 kategoriler.php	85
4.2.4 yaziekle.php	89
CONCLUSION	97
REFERENCES	98

CHAPTER ONE : APACHE WEB SERVER

1.1 The WWW

The Internet has been around a long time. More than 30 years now. But for most of the 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". That 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. Ironical that the name stuck!

1.1.1 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 Supercomputing Activities (NCSA) at the University of Illinois at Urbana Champaign (UIUC). NCSA started working on the NCSA HTTPd (Hyper Text 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.

1.2 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 for 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.

1.2.1 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 anyone no longer 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 spaces 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 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.

1.2.2 Apache's architecture

Since the 1.0 release of Apache has a modular design. The core of the server is very light- weight and all other functions are implemented as modules that plug in 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.

1.2.3 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 was 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 a just 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 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 be 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.
- 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

1.2.4 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 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 us, and we should support the ASF in all its endeavors, if only for purely selfish reason.

1.2.5 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 is the traditional email for Apache. 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 is 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 works at Covalent.

1.3 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.

1.3.1 Hardware / Software Requirements

Apache runs on anything. Almost. It will almost certainly run on whatever you have. 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-zkf 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 figure 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.

1.3.2 Advanced Installation

If you did not satisfied with the default installation which does not have all modules you can install Apache that you want all modules. There two ways to handle that. 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 later approach, but the former approach gets more coverage in the docs, and so is used more frequently.

You are advised to use 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.

1.3.3 Editing the Configuration Scripts

In our previous example, we run a script called `configure` (“small-c configure”) in the main Apache directory. In this method, we are 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 start out the same:

```
tar-zxfapache_1.3.12.tar.gz
```

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

```
cdapache_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 do not 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 “samall-c” configuration, you will have a file called Configuration.apaci created for you, which will then get used in this configuration step.

1.3.4 Dynamic Shared Objects (DSO's)

I get tired of rebuilding and reinstalling my web server everytime i want 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 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 sever., but only actually use the parts that you want at any time, and leave out everything else that you are 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 can not 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
```

What does this command do?

Well it compiles all of the modules that ship with Apache, except those are considered experimental or unstable, and enables them all as DSO's. This means that Apache will be loadnig up a bunch of modules that you dont really want, so you need to

edit the configuration file and comment out those modules that you are 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 server runs at the same speed. That is a server running modules as DSOs does not run any slower.

1.4 Configuring

Once you have compiled and installed your server, you need to configure it for your particular environment. Many of the configuration directives got set when you run 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 markets, lets you configure everything down to smallest detail. And if there is really something that you want to configure that you can not, you have the source code so you can change it if you are so inclined.

1.4.1 Configuration Files

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

Note that if you installed Apache with a RPM, then the files will be in bizarre places that have no relation to logic. Uninstall the RPM, and install from source. It is a simple process, and reduces your pain in the long run.

1.4.2 Comanche

One of the battles that *nix continually has to fight is the notion that it is 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 is so easy to get it wrong. Send mail 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 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 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 files. 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.

1.4.3 Starting, Stopping, Restarting

There are a variety of ways to control your Apache server. We will 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 which does many of the things that you will 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 enabled, and that `lynx` is installed.

`graceful`

does a graceful restart by sending a SIGUSR 1, 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 process 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 `rc.httpd`, it will be run on the server startup. `rc.httpd` should contain in the following command:

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

If you are running Red Hat, or Mandrake, or one of the other Linuxes that look like them, you will 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 will usually be in either runlevel 3 or 5, so that is when you want to start Apache.

1.5 Integrating Apache With The Rest of Your Business

The common wisdom is that every company needs a web site, because every company needs web site. And so lots of have web sites which are nothing more than a electronic sales brochure.

With more and more people online everyday, 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 answer that they need, and it should be instantaneous.

1.5.1 CGI

The most common way to tie your web site to your 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.

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 pf ypur web site.

There are plethora of good CGI tutorials on the web. But the basic concepts are pretty simple.

1.5.2 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 ses CGI programs that have just `\n`, rather than both, but it is more correct to use both. This used be more of a big deal that it is now. Most web browsers are quite happy to accept one or the other.

1.5.3 Reading Client Input

Input from the client comes into your program on STDIN. This means that you can read client as though it was coming from the command line or from the keyboard.

Of course, most languages taht you are likely to use for CGI programming have libraries readily available that will handle most of the mundane details of CGI programming 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 cokies, 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 languae of choice for CGI programming, because it is very conductive to the sort of the rapidprototyping 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 will 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 exiting ,is it?
```

Rather than provide a lot of example CGI programs,I'd encourage you to look at all the resources at the ned of this paper for examples.

1.5.4 mod_perl

Something that oyu wil ,l 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 it 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 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 solutionx is mod_perl.

Mod_perl is an Apache module that significantly enhances the performance of Perl Cgi programs. It has other benefits, suxh 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 disc.

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.

1.5.5 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 extremely 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 number of SSL implementations that run on top of Apache. Two of better known ones are Raven, from Covalent (<http://www.covalent.com/rqven/ssl>), and Stronghold (<http://www.c2.net/products/sh2/>). These are both commercial products. The Open Source alternative is mod_ssl (<http://modssl.org>), which runs in conjunction with OpenSSL(<http://www.openssl.org>).

With the recent changes in the crypto laws, there is a good chance that mod_ssl will ship as one of the standard Apache modules in future releases.

1.5.6 Authentication

Authentication is the process of verifying that you are who 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.

1.5.7 mod_auth T

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 us 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] password file username
```

```
htpasswd -b[cmdps] password file username password
```

-c Create new file

, m force MD5 encryption of the password

,d force CRYPT encryption of the password (default)

,p do not encrypt the password

-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 the 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 will 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 member 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 can not 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"
AuthTypeBasic
Auth User File /path/to/htpasswd
Auth Group File /path/to/htgroup
Require group TCG
```

The Auth Name is the string that appears in the authentication dialog that pops up when you visit a protected area.

Auth type is the method of authentication. It is one of the 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.

Auth User File and Auth Group File 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 users or groups 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.

1.5.8 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 user name 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 troubleshooting CGI programs that are not behaving as expected.

Custom log files

With the Log Format and Custom Log directives, you can create your own log files that contain whatever information you would 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 is 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 of the main web page www.apache.org web site, and you can usually find information on those sites about contributing code or documentation patches.

1.6 A Conclusion for Apache

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

CHAPTER TWO : PERSONAL HOME PAGE (PHP)

2.1 What is PHP ?

PHP is a server side scripting language designed specifically for the web. Within a HTML page, we can embed PHP codes 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 the man, Rasmus Lerdorf. It was adopted by other talented people and has gone through there major rewrites to brings us the broad, mature product we see today. As of January 2001, it was in use on nearly five million domains world wide, and this number is growing rapidly.

PHP is an Open Source 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.

2.2 Writing in PHP

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

2.2.1 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 the text "Hello World".

2.2.2 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>
```

2.2.3 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 is Here
?php>
```

```
<script language="php">
PHP Code is here
</script>
```

All of this works 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.

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:

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


As you can see 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 set up 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 is the PHP code. A PHP code snippet is surrounded by an opening and 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 parenthesis 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 a lot of interesting info about the server system and PHP itself. Take a look at the source of the resulting page (view source). You will 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 is basically how the engine works. The PHP code you write is replaced in the resulting web page by results of running the PHP code. Nobody gets to see your actual PHP codes—only the result of it running. That is 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 have used here is called 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 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 on the series. Here is the same code in the short style and ASP style respectively:

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

As 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 is not 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:

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

2.3 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 \$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 through :

Strings are case sensitive so \$Welcome_text is not the same as \$welcome_text

String names can contain letters, numbers and underscores but can not 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.

2.3.1 Outputting Variables

To display a variable on the screen uses exactly the same code as to display 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.

2.3.2 Formatting Text

Everything is just output in the browser's default form. 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:

Helleo and welcome to my website .

This means though, that you can include standard HTML tags require the "sign. You may notice that this will clash with the question marks used to print your text. This means that you must tell the script which quotes should be used and which ones should be ignored.

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

```
<fontface="arial" color ="#FF000">
</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 change to:

```
<font face=\"arial\" color\"#FF000\">
</font>
```

You can now include this in your print statement:

```
print(<font face=\"arial\" color\"#FF000\">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 to code:

```
<font face="arial" color="FF000">:Hello and welcome to my website.</font>
```

2.4 PHP Operators

This section lists the different operators used in PHP.

2.4.1 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	1 2
++	Increment	X=5 X++	X=6
--	decrement	X=5 x--	X=4

2.4.2 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

2.4.3 Comparison Operators

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

2.4.4 Logical Operations

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

2.5 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 staement if you want to execute a set of code when a condition is true and another if the condition is not true.

else if statement is used with the if.... else statement to execute a set of code if one of several condition are true.

2.5.1 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>
```

2.5.2 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>
```

2.6 PHP String Processing

The strig function allow you to manipulate strings.

2.6.1 PHP String Functions

PHP indicates the earliest version of PHP supports the function.

Function	Description	php
addslashes()	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 string	4
crypt()	One way string encryption	3
echo()	Outputs string	3
explode()	Breaks a string into array	3
fprintf()	Writes a formatted string to a specified output stream.	5
hebrew()	Converts Hebrew text to visual text	3
html_entity_decode()	Converts HTML entities to characters	4
htmlentities()	Converts characters to HTML entities	3
htmlspecialchars()	Converts some predefined character to html entities.	3
implode()	Returns a string from the elements of an array	3
join()	Alias of implode()	3
levenshtein()	Returns the Levenshtein distance between two strings	3
localeconv()	Returns local numeric and monetary formatting information	4
ltrim()	Strips whitespace from the left side of a string	3
md5()	Calculates the MD5 hash of a string	3
metaphone()	Calculates the metaphone key of a string	4
money_format()	Returns a string formatted as a currency string.	4
nl_langinfo()	Returns a specific local information	4
number_format	Formats a number with grouped thousands	3
parse_str()	Parses a query string into variables	3
print()	Outputs a string	3
printf()	Outputs a formatted string	3

quoted_printable_decode()	Decodes a quoted printable string	3
quotemeta()	Quotes meta characters	3
rtrim()	Strips whitespace from the right side of the string	3
setlocale()	Sets locale information	3
sha_1()	Calculates the SHA-1 hash of a string	4
sha_1_files()	Calculates the SHA-1 hash of a file	4
similar_text()	Calculates the similarity between two strings	3
soundex()	Calculate the soundex key of a string	3
sprintf()	Writes a formatted string to a variable	3
sscanf()	Parses input from a string according to a format	3
str_pad()	Pads a string to a new length	4
str_repeat()	Repeats a string a specified number of times	4
str_replace()	Replaces some characters in a string	3
str_rot13()	Performs the ROT 13 encoding on a string	4
str_shuffle()	Randomly shuffles all characters in a string	4
str_split()	Splits a string into an array	5
str_word_count()	Count the number of words in a string	4
strcasecmp()	Compares two strings	3
strchr()	Finds the first occurrence of a string inside another	3
strcmp()	Compares two strings	3
strcoll()	Locale based string comparison	4
strip_tags()	Strips HTML and PHP tags from a string	3
stripclashes()	Unquotes a string quoted with addslashes()	4
stripslashes()	Unquotes a string quoted with addslashes()	3
strstr()	Finds the first occurrence of a string inside another string	3
strlen()	Returns the length of a string	3
strnatcmp()	Compares two strings using a natural order algorithm	4
strncasecmp()	String comparison of the first n characters	4
strpbrk()	Searches a string for any of a set of characters	5
strpos()	Returns the position of the first occurrence of a string inside another string	3
strchr()	Finds the last occurrence of a string inside another string	3
strrev()	Reverses a string	3
stripos()	Finds the position of the last occurrence of a string inside another string	5
strrpos()	Finds the position of the last occurrence of a string inside another string	3
strspn()	Returns the number of characters found in a string that contains only characters from a specified charlist	3
strstr()	Finds the first occurrence of a string inside another string	3
strtok()	Splits a string into smaller strings	3

strtolower()	Converts a string to lowercase letters	3
strtoupper()	Converts a string to uppercase letters	3
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	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 specific output stream	5
vprintf()	Outputs a formatted string	4
vsprintf()	Writes a formatted strings to a variable	4
wordwrap()	Wraps a string to a given number of characters	4

2.7. The mail() Function

The mail() function is used to send e mail.

Syntax:

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

Parameter	Description
to	Required, specifies the receiver/ receivers of the mail.
subject	Required, specifies the subject of the email. This parameter can not contain any newlinw character.
message	Required. Defines the message to be sent. Each line should seperated with a LF (\n). Lines should be seperated with a CRLF(\r\n)
Headers	Optional. Specifies additional headers, like from,Cc,and Bcc.The additional headers should be seperated with a CRLF (\r\n)
Parameters	Optional. Specifies an additional parameter to send mail program.

2.7.1 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 email.


```

<?php
$to="someone@someplace.com";
$subject="test mail";
$message="hello! This is a simple text mail message";
$from=someonelse@anotherplace.com;
$headers="from :$from";
mail($to, $subject, $message, $headers);
echo"Mail Sent";
?>

```

2.7.2 PHP Mail Form

Using PHP you can create a feedback form for your web site. 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 be 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 it will output the HTML mail form.

If the email input is set 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 email.

```

<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>
```

2.7.3 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 the part of the PHP core. There is no installation needed to use these functions.

2.7.4 Runtime Functions

The behaviour 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_form	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	PHP_INI_ALL

2.7.5 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

2.8 Cookies

A cookie is often used to identify the 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.

2.8.1 How to Create a Cookie

The setcookie() function is used to create cookies.

Note: the setcookie() function must be 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 is set on this page! The cookie will be active
when the client has sent the cookie back to the server.
</p>
</body>
</html>
```

2.8.2 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/>";
?php>
</body>
</html>
```

CHAPTER THREE : MySQL

3.1 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 application or CGI invoked by Perl database modules. MySQL has a 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 the Oracle or Microsoft Access, it is also considered to be the fastest database 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 the "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.

3.2 Database Construction

MySQL databases have a standard setup. They are made up of a database, in which are 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.

3.2.1 Databases and Logins

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

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

3.3 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 databse 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, thewhole creation and setup will be done in one command.

3.3.1 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 numbers
MEDIUMINT	Integer number
INT	Integer number
VARCHAR	Text (maximum 256 character)
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.

3.3.2 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.

3.3.3 The Contacts Database

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

Name	Type	Length
id	SMALLINT	6
sayfa	CHAR	3
sayi	VARCHAR	5
resim1	VARCHAR	100
resim2	VARCHAR	100

3.4 Connect to MySQL Server

The first thing you will want to do is to establish a connection to your MySQL database server. This is accomplished with the `mysql_connect` function. Here is 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 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

```
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);
```

3.4.1 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 attempt to connect to the MySQL server fails. In the real world, this is the more common requirements. The error control operator is an `@` sign placed next to the function call like this:

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

3.5 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('Couldnt 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 "First Name", "Last Name", and "Age":

```
mysql_select_db("my_db",$connection);  
$sql="CREATE TABLE Person  
(  
First Name varchar(15),  
Last Name varchar(15),  
Age int,  
)";  
mysql_query($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) smalint(size) tinyint(size) mediumint(size) bigint(size)	Hold integers only. The maximum number of digits are specified in paranthesis
decimal (size,d) double(size,d) Ffloat(size,d)	Hold numbers with fractions. The maimum 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 paranthesis
varchar(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in paranthesis.

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 character.

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.

3.6 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 tht 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),
First Name varchar(15,)
LastName varchar(15),
Age int
```

```
);  
mysql_Query($sql,$con);
```



3.7 PHP MySQL Functions

Function	Description
mysql_affected_rows	Returns the number of affected rows in the previous MySQL operation.
mysql_change_user	Deprecated. Changes the user of the current MySQL connection
mysql_client_encoding	Returns the name of the character set for the current connection.
mysql_close	Closes a non-persistent MySQL connection
mysql_connect	Opens a non-persistent MySQL connection
mysql_create_db	Deprecated. Creates a new MySQL database.
mysql_data_seek	Moves the recor pointer
mysql_db_name	Returns a database name from a call to mysql_list_dbs()
mysql_db_query	Deprecated. Sends a MySQL query.
mysql_drop_db	Deprecated. Deletes a MySQL databse
mysql_errno	Returns the error number of the last MySQL operation
mysql_error	Returns the error description of the last MySQL operation
mysql_escape_string	Deprecated. Escapes a string for use in a mysql_query.
mysql_fetch_array	Returns a row from a recordset as an associative array and / or a numeric array
mysql_fetch_assoc	Returns a row from recordset as an associative array
mysql_fetch_field	Returns a column info from a recordset as an object
mysql_fetch_lengths	Returns the length of the contents of each field in a result row
mysql_fetch_object	Returns a row from recordset as an obkect
mysql_fetch_row	Returns a row from a recordset as a numeric array
mysql_field_flags	Returns the flags associated with a field in a recordset
mysql_field_len	Returns the maximum length of a field in a recordset
mysql_field_name	Returns the name of a field in a recordset
mysql_field_seek	Moves the result pointer to a specified field
mysql_field_table	Returns the name of the table the specified field is in
mysql_field_type	Returns the type of a field in a record set
mysql_free_result	Free result memory
mysql_get_client_info	Returns MySQL client info
mysql_get_host_info	Returns MySQL host info
mysql_get_proto_info	Returns MySQL protocol info
mysql_get_server_info	Returns MySQL server info
mysql_info	Returns information about yhe last query
mysql_insert_id	Returns the AUTO INCREMENT ID generated from the previous INSERT operation.
mysql_list_dbs	Lists available databases on a MySQL server
mysql_list_fields	Deprecated. Lists MySQL table fields.
mysql_list_process	List MySQL processes

mysql_list_tables	Deprecated. Lists tables in MySQL database
mysql_num_fields	Returns the number of fields in a recordset
mysql_num_rows	Returns the number of rows in arecordset
mysql_pconnect	Opens a persistent MySQL connection
mysql_ping	Pings a server connection or reconnects if there is no connection
mysql_query	Executes a query on a MySQL database
mysql_real_escape_string	Escapes a string for use in SQL statement
mysql_result	Returns the value of a field in a recordset
mysql_select_db	Sets the active MySQL database
mysql_stat	Returns the current system status of the MySQL server
mysql_tablename	Deprecated. Returns the table name of field.
mysql_thread_id	Returns the current thread IF
mysql_unbuffered_query	Executes a query on a MySQL database

3.7.1 Selecting A Database

Here is something that is not too complicated. To select MySQL database in PHP use the mysql_select_db() function like this:

```
mysql_select_db ("mydatabase");
```

couldnt be simpler, as long as everythig went well ,-- more on that in a minute.

3.7.2 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 will add one more 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 alos 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 remeber that an "if" statement has this basis 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 when everything is ok, and to put code in the false side to handle our errors.

Let us build an example. First, we are going to connect to the server, and if everything is good, we will create a database. If there were errors, we will terminate the program, displaying an error message. In the process of creating the database, if everything goes well, we will select the database `si` that we can use it, and if there are errors we will terminate the program, displaying an appropriate message. Similarly, when selecting the database, we will 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 known as “nesting” the “if” statements. When you write nested “if” statements it is highly recommended that you use indentation and spacing to help you keep track of where you are. It is all together too easy to have an extra or a missing parenthesis and completely change the logic flow of your program—often without causing any PHP syntax error. Now here is our code:

```
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>Database 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>')
}
```


3.8 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 database.

First we create a form:

```
<form action="insert_db.php" method=POST">
Enter Your First Nmae: <input type="text" name="firstname"/>
Enter Your Last Nmae: <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('Couldnt 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!";
```

3.9 ODBC With PHP

ODBC is an Application Programming Interface(API) that allows you to connectto a data source .

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 panle.

Double click on the Data Sources (ODBC) icon inside

Choose the System DSN tab.

Click on Add in the System DSN.

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 Service (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 set up a DSN for you to use.

3.9.1 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);
```

3.9.2 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 "Company Name":

```
$compname=odbc_result($rs,"company Name");
```

3.9.3 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>company name</th>";
echo"<th>contactname </th></tr>";
while(odbc_fetch_row($rs))
{
    $compname=odbc_result($rs,"Company Name");
    $conname=odbc_result($rs,"Contact Name");
    echo"<tr><td>$compname</td>";
    echo"<td>$conname</td></tr>";
}
odbc_close($conn);
echo"<table>";
?>
</body>
</html>
```

CHAPTER FOUR : APPLICATION CODES

4.1 User PHP

4.1.1 Anketal.php

```
<? ob_start(); $sonuc=@$HTTP_GET_VARS["sonuc"];?>
<html>
<head>
<title>Anket Sonuçları - Telgraf Gazetesi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link rel="stylesheet" href="include/stil.css" type="text/css">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    color: #000000;
}
.style1 {
    color: #CC0000;
    font-weight: bold;
}
-->
</style></head>
<body>
<br>
<table width="505" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="9">&nbsp;</td>
<td><table width="495" border="0" cellpadding="0" cellspacing="0">
```



```

<tr>
  <td colspan="3"></td>
</tr>
<tr>
  <td width="26"></td>
  <td width="440" valign="top" bgcolor="#E6E6E6">
    <table width="430" border="0" align="center" cellpadding="0" cellspacing="0">
      <tr>
        <td><div align="center" class="style1">
          <?
include ("include/baglan.php");
if ($sonuc!="1"){
$anketk=@$HTTP_COOKIE_VARS["telgrafanket"];
$anket=$HTTP_POST_VARS["id"];
$oy=$HTTP_POST_VARS["oy"];

if ($anketk!=$anket) { //
setcookie("telgrafanket",$anket,time()+86400);
mysql_query("update anket set $oy=$oy+1 where id=$anket ");

echo"<center>Anketimize katıldığınız için teşekkür ederiz<br>Anket
Sonuçları</center>";
} //kuki kontrol end
else
{
echo"<center>Daha önce oy kullandınız!</center>";
} }
?>

      </div></td>
    </tr>
  <tr>
    <td><div align="center"><b>
    <?
$si=0;

```



```

        <td width="10">&nbsp;</td>
        <td width="230" valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c1;?>"
<i>% <?php echo $foran1; ?></i> </b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <tr>
        <td valign="top"><?php echo "<b>$s2</b>&nbsp;&nbsp;<i>($c2 oy)</i>";
?></td>
        <td width="10">&nbsp;</td>
        <td valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c2;?>" <i>%
<?php echo $foran2; ?></i></b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <?php
        if ($s3!="")
            echo "<tr>
                <td valign=\"top\"><b>$s3</b>&nbsp;&nbsp;<i>($c3 oy)</i></td>
                <td width=\"10\">&nbsp;</td>
                <td valign=\"top\"><b><img src=\"images/gstrg.gif\"
width=\"<$foran3_gosterge\" height=\"13\" alt=\"Toplam oy sayısı=$c3\"> <i>%
$foran3</i></b></td>
            </tr>
            <tr>
                <td height=\"10\" colspan=\"3\"></td>
            </tr>";
        if ($s4!="")
            echo "<tr>
                <td valign=\"top\"><b>$s4</b>&nbsp;&nbsp;<i>($c4 oy)</i></td>

```

```

        <td width="10">&nbsp;</td>
        <td valign="top"><b> <i>%
$foran4</i></b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>";
?>

        <tr>

        <td>&nbsp;</td>
        <td width="10">&nbsp;</td>
        <td><div align="right">Toplam oy sayısı: <?php echo $toplamlar_oy;
?></div></td>
    </tr>
</table></td>
    <td width="29"></td>
</tr>
<tr valign="top">
    <td height="20" colspan="3" background="images/anket2.gif"><table
width="97%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td height="3"></td>
    </tr>
    <tr>
        <td><div align="center"><a href="javascript:window.close()" class="kats
kalin" >Pencereyi Kapat</a></div></td>
    </tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</body> </html>

```


4.1.2 ara.php

```
<? ob_start(); $sonuc=@$HTTP_GET_VARS["sonuc"];?>
<html>
<head>
<title>Anket Sonuçları - Telgraf Gazetesi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link rel="stylesheet" href="include/stil.css" type="text/css">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    color: #000000;
}
.style1 {
    color: #CC0000;
    font-weight: bold;
}
-->
</style></head>
<body>
<br>
<table width="505" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="9">&nbsp;</td>
<td><table width="495" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="3"></td>
</tr>
```

```

<tr>
  <td width="26"></td>
  <td width="440" valign="top" bgcolor="#E6E6E6">
    <table width="430" border="0" align="center" cellpadding="0" cellspacing="0">
      <tr>
        <td><div align="center" class="style1">
          <?
include ("include/baglan.php");
if ($sonuc!="1"){
$anketk=@$HTTP_COOKIE_VARS["telgrafanket"];
$anket=$HTTP_POST_VARS["id"];
$oy=$HTTP_POST_VARS["oy"];

if ($anketk!=$anket) { //
setcookie("telgrafanket",$anket,time()+86400);
mysql_query("update anket set Soy=$oy+1 where id=$anket ");

echo"<center>Anketimize katıldığınız için teşekkür ederiz<br>Anket
Sonuçları</center>";
} //kuki kontrol end
else
{
echo"<center>Daha önce oy kullandınız!</center>";
} }
?>

        </div></td>
      </tr>
      <tr>
        <td><div align="center"><b>
          <?
$si=0;
$sorgu=mysql_query("SELECT * from anket order by id desc limit 0,1");
//$adet=mysql_numrows($sorgu);
$si = mysql_result ($sorgu,$si,"id");

```

```

$soru = mysql_result ($sorgu,$i,"soru");
$s1 = mysql_result ($sorgu,$i,"s1");
$c1 = mysql_result ($sorgu,$i,"c1");
$s2 = mysql_result ($sorgu,$i,"s2");
$c2 = mysql_result ($sorgu,$i,"c2");
$s3 = mysql_result ($sorgu,$i,"s3");
$c3 = mysql_result ($sorgu,$i,"c3");
$s4 = mysql_result ($sorgu,$i,"s4");
$c4 = mysql_result ($sorgu,$i,"c4");

//-----> alt kysym alynan de?erleri toplayarak
her soruya dü?en orany bulur ve resim büyüklü?ünü ayaralar
$toplaml_oy=$c1+$c2+$c3+$c4;
@$foran1=floor($c1*100/$toplaml_oy);
$foran1_gosterge=$foran1*2;
@$foran2=floor($c2*100/$toplaml_oy);
$foran2_gosterge=$foran2*2;
@$foran3=floor($c3*100/$toplaml_oy);
$foran3_gosterge=$foran3*2;
@$foran4=floor($c4*100/$toplaml_oy);
$foran4_gosterge=$foran4*2;
//echo"$sisim1 $foran1 <br> $sisim2 $foran2 <br> $sisim3 $foran3<br> $sisim4
$foran4<br> $sisim5 $foran5<br> $sisim6 $foran6<br> $sisim7 $foran7";
?>

<?php echo $soru; ?></b></div></td>
</tr>
</table>
<br>
<table width="430" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="190" valign="top"><?php echo
"<b>$s1</b>&nbsp;&nbsp;&nbsp;<i>($c1 oy)</i>"; ?></td>
<td width="10">&nbsp;&nbsp;&nbsp;</td>

```

```

        <td width="230" valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c1;?>"
<i>% <?php echo $foran1; ?></i> </b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <tr>
        <td valign="top"><?php echo "<b>$s2</b>&nbsp;&nbsp;&nbsp;<i>($c2 oy)</i>";
?></td>
        <td width="10">&nbsp;&nbsp;&nbsp;</td>
        <td valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c2;?>" <i>%
<?php echo $foran2; ?></i></b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <?php
        if ($s3!="")
            echo "<tr>
                <td valign=\"top\"><b>$s3</b>&nbsp;&nbsp;&nbsp;<i>($c3 oy)</i></td>
                <td width=\"10\">&nbsp;&nbsp;&nbsp;</td>
                <td valign=\"top\"><b><img src=\"images/gstrg.gif\"
width=\"$foran3_gosterge\" height=\"13\" alt=\"Toplam oy sayısı=$c3\"> <i>%
$foran3</i></b></td>
            </tr>
            <tr>
                <td height=\"10\" colspan=\"3\"></td>
            </tr>";
        if ($s4!="")
            echo "<tr>
                <td valign=\"top\"><b>$s4</b>&nbsp;&nbsp;&nbsp;<i>($c4 oy)</i></td>
                <td width=\"10\">&nbsp;&nbsp;&nbsp;</td>

```



```

        <td valign="top"><b> <i>%
$foran4</i></b></td>

    </tr>

    <tr>

        <td height="10" colspan="3"></td>

    </tr>";

?>

        <tr>

            <td>&nbsp;</td>

            <td width="10">&nbsp;</td>

            <td><div align="right">Toplam oy sayısı: <?php echo $toplam_oy;
?></div></td>

        </tr>

    </table></td>

    <td width="29"></td>

</tr>

<tr valign="top">

    <td height="20" colspan="3" background="images/anket2.gif"><table
width="97%" border="0" align="center" cellpadding="0" cellspacing="0">

        <tr>

            <td height="3"></td>

        </tr>

        <tr>

            <td><div align="center"><a href="javascript:window.close()" class="kats
kalin">Pencereyi Kapat</a></div></td>

        </tr>

    </table></td>

</tr>

</table></td>

</tr>

</table>

</body>

</html>

```

4.1.3 index.php

```
<? ob_start(); $sonuc=@$HTTP_GET_VARS["sonuc"];?>
<html>
<head>
<title>Anket Sonuçları - Telgraf Gazetesi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<link rel="stylesheet" href="include/stil.css" type="text/css">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    color: #000000;
}
.style1 {
    color: #CC0000;
    font-weight: bold;
}
-->
</style></head>
<body>
<br>
<table width="505" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="9">&nbsp;</td>
<td><table width="495" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="3"></td>
</tr>
</tr>
```

```

<tr>
  <td width="26"></td>
  <td width="440" valign="top" bgcolor="#E6E6E6">
    <table width="430" border="0" align="center" cellpadding="0" cellspacing="0">
      <tr>
        <td><div align="center" class="style1">
          <?
include ("include/baglan.php");
if ($sonuc!="1"){
$anketk=@$HTTP_COOKIE_VARS["telgrafanket"];
$anket=$HTTP_POST_VARS["id"];
$oy=$HTTP_POST_VARS["oy"];

if ($anketk!=$anket) { //
setcookie("telgrafanket",$anket,time()+86400);
mysql_query("update anket set $oy=$oy+1 where id=$anket ");

echo"<center>Anketimize katıldığınız için teşekkür ederiz<br>Anket
Sonuçları</center>";
} //kuki kontrol end
else
{
echo"<center>Daha önce oy kullandınız!</center>";
} }
?>

        </div></td>
      </tr>
      <tr>
        <td><div align="center"><b>
          <?
$si=0;
$sorgu=mysql_query("SELECT * from anket order by id desc limit 0,1");
//$adet=mysql_numrows($sorgu);
$id = mysql_result ($sorgu,$i,"id");

```

```

$soru = mysql_result ($sorgu,$i,"soru");
$s1 = mysql_result ($sorgu,$i,"s1");
$c1 = mysql_result ($sorgu,$i,"c1");
$s2 = mysql_result ($sorgu,$i,"s2");
$c2 = mysql_result ($sorgu,$i,"c2");
$s3 = mysql_result ($sorgu,$i,"s3");
$c3 = mysql_result ($sorgu,$i,"c3");
$s4 = mysql_result ($sorgu,$i,"s4");
$c4 = mysql_result ($sorgu,$i,"c4");

//-----> alt kysym alynan de?erleri toplayarak
her soruya dü?en orany bulur ve resim büyüklü?ünü ayaralar
$toplaml_oy=$c1+$c2+$c3+$c4;
@$foran1=floor($c1*100/$toplaml_oy);
$foran1_gosterge=$foran1*2;
@$foran2=floor($c2*100/$toplaml_oy);
$foran2_gosterge=$foran2*2;
@$foran3=floor($c3*100/$toplaml_oy);
$foran3_gosterge=$foran3*2;
@$foran4=floor($c4*100/$toplaml_oy);
$foran4_gosterge=$foran4*2;
//echo"$sisim1 $foran1 <br> $sisim2 $foran2 <br> $sisim3 $foran3<br> $sisim4
$foran4<br> $sisim5 $foran5<br> $sisim6 $foran6<br> $sisim7 $foran7";
?>

```

```

<?php echo $soru; ?></b></div></td>

</tr>

</table>

<br>

<table width="430" border="0" align="center" cellpadding="0" cellspacing="0">

<tr>

<td width="190" valign="top"><?php echo
"<b>$s1</b>&nbsp;&nbsp;&nbsp;<i>($c1 oy)</i>"; ?></td>

<td width="10">&nbsp;&nbsp;&nbsp;</td>

```



```

        <td width="230" valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c1;?>">
<i>% <?php echo $foran1; ?></i> </b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <tr>
        <td valign="top"><?php echo "<b>$s2</b>&nbsp;&nbsp;&nbsp;<i>($c2 oy)</i>";
?></td>
        <td width="10">&nbsp;</td>
        <td valign="top"><b>" height="13" alt="Toplam oy sayısı=<?php echo $c2;?>"> <i>%
<?php echo $foran2; ?></i></b></td>
    </tr>
    <tr>
        <td height="10" colspan="3"></td>
    </tr>
    <?php
        if ($s3!="")
            echo "<tr>
                <td valign=\"top\"><b>$s3</b>&nbsp;&nbsp;&nbsp;<i>($c3 oy)</i></td>
                <td width=\"10\">&nbsp;</td>
                <td valign=\"top\"><b><img src=\"images/gstrg.gif\"
width=\"\$foran3_gosterge\" height=\"13\" alt=\"Toplam oy sayısı=$c3\"> <i>%
$foran3</i></b></td>
            </tr>
            <tr>
                <td height=\"10\" colspan=\"3\"></td>
            </tr>";
        if ($s4!="")
            echo "<tr>
                <td valign=\"top\"><b>$s4</b>&nbsp;&nbsp;&nbsp;<i>($c4 oy)</i></td>
                <td width=\"10\">&nbsp;</td>

```

```

        <td valign="top"><b> <i>%
\$foran4</i></b></td>
    </tr>
    <tr>
        <td height="\10\" colspan="\3\"></td>
    </tr>";
?>

        <tr>
            <td>&nbsp;</td>
            <td width="10">&nbsp;</td>
            <td><div align="right">Toplam oy sayısı: <?php echo \$toplamlam_oy;
?></div></td>
        </tr>
    </table></td>
    <td width="29"></td>
</tr>
<tr valign="top">
    <td height="20" colspan="3" background="images/anket2.gif"><table
width="97%" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td height="3"></td>
        </tr>
        <tr>
            <td><div align="center"><a href="javascript:window.close()" class="kats
kalin">Pencereyi Kapat</a></div></td>
        </tr>
    </table></td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</body>
</html>

```

4.1.4 kategori.php

```
<?php
session_start();
ob_start();
?>

<html>
<head>
<?php
include("include/baglan.php");
?>

<title><?php echo "$title"; ?></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<link rel="stylesheet" href="include/stil.css" type="text/css">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
    background-color: #c2c2c2;
}
body,td,th {
    font-family: Arial;
    font-size: 11px;
    color: #333333;
}
-->
</style>
<meta name="keywords"
content="telgraf,telgraph,gazete,ekonomi,siyaset,politika,gaste,newspaper,daily,press,journal,gaziantep,yerel,turkey,türkiye,turqia">
<meta name="description" content="Telgraf.Net">
<link REL="SHORTCUT ICON" HREF="/images/telgraf.ico">
```

```

</head>
<script language="JavaScript" type="text/JavaScript">
<!--
function git(targ,selObj,restore){ //v3.0
    eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
    if (restore) selObj.selectedIndex=0;
}
//-->
</script>
<body>
<table width="820" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
    <td width="10" valign="top" background="images/bg_k1.gif"></td>
    <td width="800" valign="top" bgcolor="#FFFFFF"><?php
include("include/ust.php"); ?>
    <table width="800" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="336"></td>
            <td valign="top"><?php include("include/reklam.php"); ?></td>
        </tr>
    </table>
    <table width="800" height="100%" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td width="139" valign="top" background="images/bg_sol.gif"
bgcolor="#f9faf7"><?php include("include/kategoriler.php"); ?><table width="139"
border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td><a
href="#"></a></td>
                </tr>
            </table>
        </table>

```



```

        <table width="133" border="0" align="center" cellpadding="0"
cellspacing="0">
            <tr>
                <td width="139" valign="top"><?php include("include/anket.php"); ?></td>
            </tr>
        </table></td>

        <td width="661" valign="top"><table width="661" border="0" cellspacing="0"
cellpadding="0">
            <tr>
                <td width="98" rowspan="2" valign="top"></td>
                <td width="563" height="21" background="images/bg_doviz.gif"><table
width="560" border="0" cellspacing="0" cellpadding="0">
                    <tr>
                        <td width="100"></td>
                        <td valign="bottom"><?php include("doviz.php"); ?></td>
                    </tr>
                </table>
            </td>
            <tr>
                <td background="images/bg_arsiv.gif"><?php include("include/arama.php");
?></td>
            </tr>
        </table>

        <table width="661" height="100%" border="0" cellpadding="0"
cellspacing="0">
            <tr>
                <td width="4">&nbsp;</td>
                <td valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
                    <tr>
                        <td height="26" bgcolor="#797768"><?php
                            $kat_gelen=$_GET["id"];

```

```

                                $SQL = "SELECT * FROM kategoriler WHERE
id='$kat_gelen'";

$sorgu = mysql_query($SQL,$connect);
while ($record = mysql_fetch_array($sorgu))
{
    $kat_ismi=$record["isim"];
    $kat_link=$record["link"];
}
if ($kat_link!="")
header("Location:$kat_link?sayi=$sayi");
$hlist=0;
                                $SQLh = "SELECT * FROM haberler WHERE
kategori='$kat_gelen' and sayi='$sayi' ORDER by manset DESC,oncelik DESC,id
DESC";

$sorguh = mysql_query($SQLh,$connect);
$habersayisi=mysql_num_rows($sorguh);
while ($recordh = mysql_fetch_array($sorguh))
{
    $haber_id[$hlist]=$recordh["id"];
    $haber_baslik[$hlist]=$recordh["baslik"];
    $haber_tarih[$hlist]=$recordh["tarih"];
    $haber_manset[$hlist]=$recordh["manset"];

    if ($hlist<=1)
        $haber_detay[$hlist]=$recordh["detay"];
    if ($haber_manset[$hlist]=="1") {
        $mmm = explode("@detay@", $haber_detay[$hlist]);
        $mmm[0]=substr($mmm[0],0,180)."...";
        //echo "$mmm[0]";
        $haber_detay[$hlist]=$mmm[0]; }
    else
        $haber_detay[$hlist]
=substr($haber_detay[$hlist],0,120)."...";

```

```

        $hlist++;
        //echo "$haber_detay[4]";
    }

    ?>&nbsp;<span class="katbaslik"><?php echo
$kat_ismi;?></span></td>
    </tr>
    <tr>
    <td>

        <?php if ($habersayisi>=1) { ?>
        <table width="654" border="0" align="center"
cellpadding="0" cellspacing="0">
        <tr>
        <td colspan="3">&nbsp;</td>
        </tr>
        <tr>
        <td width="322" valign="top">

            <? if ($haber_id[0]!="")
            { ?>
            <table width="320" border="0"
cellspacing="0" cellpadding="0">
            <tr>
            <td><span class="13pxbold"><?php echo
$haber_baslik[0];?></span></td>
            </tr>
            <tr>
            <td><a href="detay.php<?php echo
"?id=$haber_id[0]&sayi=$sayi";?>" class="kat-liste"><span class="kat-liste2
kalin"><?php echo $haber_detay[0];?></span></a></td>
            </tr>
            <tr>
            <td class="10px"><?php echo $haber_tarih[0];?></td>
            </tr>
        </table>

```

```

        <? } ?>
    </td>

    <td width="6" background="images/bg_kara.gif"></td>
    <td width="326" valign="top"><? if ($haber_id[1]!="")
        { ?>
        <table width="320" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td><span class="13pxbold"><?php echo
$haber_baslik[1];?></span></td>
            </tr>
            <tr>
                <td><a href="detay.php<?php echo
"?id=$haber_id[1]&sayi=$sayi";?>" class="kat-liste"><span class="kat-liste2
kalin"><?php echo $haber_detay[1];?></span></a></td>
            </tr>
            <tr>
                <td class="10px"><?php echo $haber_tarih[1];?></td>
            </tr>
        </table>
        <? } ?></td>
    </tr>
    <tr>
        <td colspan="3" valign="top">&nbsp;</td>
    </tr>
    <?php if ($haber_id[2]!="") { ?>
        <tr>
            <td height="34" colspan="3" valign="top"></td>
        </tr>
        <tr>
            <td colspan="3" valign="top">
                <?php

```



```

echo "<table width=\"654\" border=\"0\"
cellpadding=\"0\" cellspacing=\"0\" bgcolor=\"#e1e0d7\">";
    foreach ($haber_id as $ix2 => $deger) {
        if (($haber_id[$ix2]!=$haber_id[0])
and ($haber_id[$ix2]!=$haber_id[1]))
            echo "<tr>
                <td width=\"22\"><img src=\"images/diger_h_baslik.gif\"
width=\"18\" height=\"19\"></td>
                <td width=\"642\"><a
href=\"detay.php?id=$haber_id[$ix2]&sayi=$sayi\" class=\"kat-
liste2\">$haber_baslik[$ix2]</a></td>
                </tr><tr><td colspan=2 height=1 bgcolor=white></td></tr>";
            }
        echo "</table>"; }

?>

</td>

</tr>
</table>

<?php }
else
echo "<center><br>Haber
Bulunamadı...</center>";

?>

</td>

</tr>
</table></td>
<td width="3">&nbsp;</td>
</tr>
<tr>
    <td height="4" colspan="3"></td>
</tr>
</table></td>
</tr>

```

```

        </table></td><td width="10" valign="top"
background="images/bg_k2.gif">&nbsp;</td>
    </tr>
</table>
<table width="820" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td><?php include("include/alt.php"); ?></td>
    </tr>
</table>
</body>
</html>

```

4.1.5 uyeol.php

```

<?php
session_start();
ob_start();
?>
<html>
<head>
<?php
include("include/baglan.php");
?>
<title><?php echo "$title"; ?></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<link rel="stylesheet" href="include/stil.css" type="text/css">
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
    background-color: #c2c2c2;
}
body,td,th {

```

```

        font-family: Arial;
        font-size: 11px;
        color: #333333;
    }
-->
</style>
<meta name="keywords"
content="telgraf,telgraph,gazete,ekonomi,siyaset,politika,gaste,newspaper,daily,press,journal,gaziantep,yerel,turkey,türkiye,turqia">
<meta name="description" content="Telgraf.Net">
<link REL="SHORTCUT ICON" HREF="/images/telgraf.ico">
</head>
<SCRIPT language=JavaScript src="include/script.js"></SCRIPT>
<script language="JavaScript" type="text/JavaScript">
<!--
function git(targ,selObj,restore){ //v3.0
    eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
    if (restore) selObj.selectedIndex=0;
}
//-->
</script>
<body>
<table width="820" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="10" valign="top" background="images/bg_k1.gif"></td>
        <td width="800" valign="top" bgcolor="#FFFFFF"><?php
include("include/ust.php"); ?>
            <table width="800" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td width="336"></td>
                    <td valign="top"><?php include("include/reklam.php"); ?></td>
                </tr>
            </table>
        </td>
    </tr>
</table>

```

```

<table width="800" height="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="139" valign="top" background="images/bg_sol.gif"
bgcolor="#f9faf7"><?php include("include/kategoriler.php"); ?><table width="139"
border="0" cellspacing="0" cellpadding="0">
      <tr>
        <td><a
href="#"></a></td>
      </tr>
    </table>
    <table width="133" border="0" align="center" cellpadding="0"
cellspacing="0">
      <tr>
        <td width="139" valign="top"><?php include("include/anket.php"); ?></td>
      </tr>
    </table></td>
    <td width="661" valign="top"><table width="661" border="0" cellspacing="0"
cellpadding="0">
      <tr>
        <td width="98" rowspan="2" valign="top"></td>
        <td width="563" height="21" background="images/bg_doviz.gif"><table
width="560" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td width="100"></td>
            <td valign="bottom"><?php include("doviz.php"); ?></td>
          </tr>
        </table>
      </td>
    </tr>
    <tr>
      <td background="images/bg_arsiv.gif"><?php include("include/arama.php");
?></td>

```



```

        </tr>
    </table>
    <table width="661" height="100%" border="0" cellpadding="0"
cellspacing="0">
        <tr>
            <td width="4">&nbsp;</td>
            <td valign="top"><table width="100%" border="0" cellpadding="0"
cellspacing="0">
                <tr>
                    <td height="26" bgcolor="#797768">&nbsp;<span
class="katbaslik">Üyelik Formu </span></td>
                </tr>
            </table>
            <table width="524" border="0" align="center" cellpadding="0"
cellspacing="0">
                <tr>
                    <td><div align="justify">
                        <?php
                            if (!session_is_registered("oturum"))
                                {
                                    ?>
                                    <br>
                                    <br>
                                    <form action="uyeol2.php" method="post" name="uyeform"
onSubmit="popupform(this, 'join')">
                                        <table width="500" border="0" align="center" cellpadding="4"
cellspacing="2">
                                            <tr>
                                                <td width="127" class="style9"><div align="left">Kullanıcı
Adı</div></td>
                                                <td width="4" class="style9">:</td>
                                                <td width="402" valign="middle">
                                                    <div align="left">

```



```

<td><div align="left">
    <input name="firma" type="text" class="input1" id="firma"
size="40" maxlength="150">
</div></td>
</tr>
<tr>
<td class="style9"><div align="left">Mail (*)</div></td>
<td class="style9">:</td>
<td>
<div align="left">
    <input name="mail" type="text" class="input1" id="mail"
size="40" maxlength="255">
</div></td>
</tr>
<tr>
<td valign="top" class="style9">&nbsp;</td>
<td valign="top" class="style9">&nbsp;</td>
<td><input name="Submit" type="submit" class="button_03"
value=" Gönder ">
<input name="Submit2" type="reset" class="button_03"
value="Temizle">
</td>
</tr>
</table>
</form>
<?php }

else
echo "<center>Üyelik formu için çıkış
yapmalısınız</center>";
?>

</div></td>
</tr>
</table></td>
<td width="3">&nbsp;</td>

```

```

        </tr>
        <tr>
            <td height="4" colspan="3"></td>
        </tr>
    </table></td>
</tr>
</table></td><td width="10" valign="top"
background="images/bg_k2.gif">&nbsp;</td>
</tr>
</table>
<table width="820" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td><?php include("include/alt.php"); ?></td>
    </tr>
</table>
</body>
</html>

```

4.2. Administrator PHP

4.2.1. anketekle.php

```

<?php ob_start(); ?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<link rel="stylesheet" href="/stil.css" type="text/css">
<SCRIPT language=JavaScript src="scrp.js"></SCRIPT>
<script language="Javascript1.2">
function disableIt(obj)
{
    obj.disabled = !(obj.disabled);
    var z = (obj.disabled) ? 'disabled' : 'enabled';
    //alert(obj.type + ' now ' + z);

```



```

        <td colspan="2">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="2"><form action="anketekle2.php" method="post"
enctype="multipart/form-data" name="form1"
onsubmit="javascript:disableIt(document.forms[0].ekle)">
            <table width="860" border="0" cellspacing="1" cellpadding="3">
                <tr bgcolor="#CCCCCC">
                    <td width="143"><strong>Anket Sorusu <span
class="bordo12pxbold">(*)</span></strong></td>
                    <td width="9"></td>
                    <td width="686"><input name="soru" type="text" id="soru" style="border-
style:solid; border-width:1;" size="70" maxlength="100">
                        <span class="fonton">(mümkün olduğunca kısa ve anlaşılır)</span></td>
                </tr>
                <tr>
                    <td bgcolor="#E9E9E9"><strong>Cevap 1 <span
class="bordo12pxbold">(*)</span></strong></td>
                    <td bgcolor="#E9E9E9"></td>
                    <td bgcolor="#E9E9E9"><input name="s1" type="text" id="s1" style="border-
style:solid; border-width:1;" size="30" maxlength="25">
                        <span class="fonton">(Tüm cevaplar; kısa, anlaşılır kelimeler ile, max 25
karakter)</span></td>
                </tr>
                <tr bgcolor="#CCCCCC">
                    <td><strong>Cevap 2 <span class="bordo12pxbold">(*)</span></strong></td>
                    <td></td>
                    <td><input name="s2" type="text" id="s2" style="border-style:solid; border-
width:1;" size="30" maxlength="25">
                        </td>
                </tr>
                <tr>
                    <td bgcolor="#E9E9E9"><strong>Cevap 3 </strong></td>
                    <td bgcolor="#E9E9E9"></td>

```

```
<td bgcolor="#E9E9E9"><input name="s3" type="text" id="s3" style="border-  
style:solid; border-width:1;" size="30" maxlength="25"> </td>
```

```
</tr>
```

```
<tr bgcolor="#CCCCCC">
```

```
<td><strong>Cevap 4 <span class="bordo12pxbold"></span></strong></td>
```

```
<td>:</td>
```

```
<td><input name="s4" type="text" id="s4" style="border-style:solid; border-  
width:1;" size="30" maxlength="25"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
<td><input type="submit" name="ekle" value=" Ekle ">
```

```
<input type="reset" name="sil" value="Temizle"> </td>
```

```
</tr>
```

```
</table>
```

```
<table width="860" border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td width="128">&nbsp;</td>
```

```
<td width="12">&nbsp;</td>
```

```
<td width="610">&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
<td class="bordo12pxbold"><span class="12redx"><strong>Açıklamalar  
!</strong></span></td>
```

```
</tr>
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
<td>- (*) işaretli alanları boş bırakmayın. <br>
```

```
- Soru ve cevaplar kutuların yanında yazan uyarılar dikkate alınarak yazılmalı.
```

```
<br>
```

- Bir ankete en az 2, en fazla ise 4 cevap yazabilirsiniz. Boş bırakılan cevap kutuları sitede görünmeyecektir ve oylama hesaplamalarında dikkate alınmayacaktır.</td>

</tr>

</table>

</form></td>

</tr>

</table>

</body>

</html>

4.2.2 haberekle.php

```
<?php ob_start(); ?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Untitled Document</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
<link rel="stylesheet" href="/stil.css" type="text/css">
```

```
<SCRIPT language=JavaScript src="scrp.js"></SCRIPT>
```

```
<script language="Javascript1.2"><!-- // load htmlarea
```

```
_editor_url = ""; // URL to htmlarea files
```

```
_editor_field = "";
```

```
var win_ie_ver = parseFloat(navigator.appVersion.split("MSIE")[1]);
```

```
if (navigator.userAgent.indexOf('Mac') >= 0) { win_ie_ver = 0; }
```

```
if (navigator.userAgent.indexOf('Windows CE') >= 0) { win_ie_ver = 0; }
```

```
if (navigator.userAgent.indexOf('Opera') >= 0) { win_ie_ver = 0; }
```

```
if (win_ie_ver >= 5.5) {
```

```
    document.write('<scr' + 'ipt src="' + _editor_url + 'editor.js"');
```

```
    document.write(' language="Javascript1.2"></scr' + 'ipt>');
```

```
} else { document.write('<scr'+ipt>function editor_generate() { return false;
```

```
}</scr'+ipt>'); }
```

```
var MyBgColor = '#FFFFFF';
```



```

var MyBgImg = "";
// --></script>
<script>
function updateAttributes(){
document.myform.bgcolor.value=MyBgColor;
document.myform.background.value=MyBgImg;
}
</script>
<script language="Javascript1.2">
function disableIt(obj)
{
    obj.disabled = !(obj.disabled);
    var z = (obj.disabled) ? 'disabled' : 'enabled';
    //alert(obj.type + ' now ' + z);
}
</script>
<style type="text/css">
<!--
body {
    margin-left: 15px;
    margin-top: 0px;
    background-image: url(images/obg.jpg);
    background-position-x: center;
    background-repeat: repeat-y;
    background-color: #FFFFFF;
}
.style5 {color: #FFFFFF}
.style6 {font-size: 10px}
-->
</style></head>

<body>
<table width="870" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>

```



```

</tr>
<tr bgcolor="#E9E9E9">
    <td width="128"><strong>Haber Tarihi <span
class="bordo12pxbold">(*)</span></strong></td>
    <td width="12"></td>
    <td width="610"><input name="tarih" type="text" id="tarih" style="border-
style:solid; border-width:1;" value="<?php echo date("d-m-Y");?>" size="6"
maxlength="10">
        <span class="fonton">(gün-ay-yıl şeklinde, 05-10-2006 gibi)</span></td>
</tr>
<tr bgcolor="#CCCCCC">
    <td><strong>Haber Başlığı (*)</strong></td>
    <td></td>
    <td><input name="baslik" type="text" id="baslik" style="border-style:solid;
border-width:1;" size="60" maxlength="140">
        <span class="fonton">(max. 140 karakter)</span></td>
</tr>
<tr bgcolor="#E9E9E9">
    <td><strong>Haber Kategori (*)</strong></td>
    <td></td>
    <td><?php
        echo "<select name=\"kategori\">";
        $SQL = "SELECT * FROM kategoriler WHERE link=" order by oncelik
ASC";

        $sorgu = mysql_query($SQL,$connect);
        while ($record = mysql_fetch_array($sorgu))
        {
            $k_isim =$record["isim"];
            $kid =$record["id"];
            echo "<option value=\"\$kid\">$k_isim</option>";
        }

        echo "</select>";
        ?></td>

```

```

</tr>
<tr bgcolor="#CCCCCC">
  <td valign="top"><strong>Haber Detayı (*)</strong></td>
  <td valign="top">:</td>
  <td><textarea name="detay" cols="70" rows="12" class="FieldStyle1"
id="detay"></textarea>
  <br>
  <span class="fonton">Manşet için haber özetini yazdıktan sonra @detay@
ibaresini ekleyin</span></td>
</tr>
<tr bgcolor="#999999">
  <td><strong>Manşet ?</strong></td>
  <td>:</td>
  <td><select name="manset" id="manset">
    <option value="1">Evet</option>
    <option value="0" selected>Hayır</option>
  </select></td>
</tr>
<tr bgcolor="#999999">
  <td><strong>Manşet Görünümü ? </strong></td>
  <td>:</td>
  <td><select name="mansetsekli" id="mansetsekli">
    <option value="1" selected>Yatay ---</option>
    <option value="2">Dikey |</option>
  </select></td>
</tr>
<tr bgcolor="#999999">
  <td><strong>Manşet Resim</strong></td>
  <td>:</td>
  <td>
    <input name="mansetresim" type="file" id="mansetresim" style="border-
style:solid; border-width:1;" size="45">
    <span class="fonton"><br>

```


(yatay için 260x120 pixel, dikey için 130x220 pixel ebatında jpg,png veya gif.
resim kenarı #ebebcbf renkte 1 pixel border)</td>

</tr>

<tr bgcolor="#CCCCCC">

<td>Resim 1 </td>

<td>:</td>

<td>

<input name="resim1" type="file" id="resim1" style="border-style:solid;
border-width:1;" size="45">

(genişlik max. 600 pixel, jpg,png veya gif)</td>

</tr>

<tr bgcolor="#E9E9E9">

<td>Resim 2 </td>

<td>:</td>

<td>

<input name="resim2" type="file" id="resim2" style="border-style:solid;
border-width:1;" size="45">

(genişlik max. 600 pixel, jpg,png veya gif)</td>

</tr>

<tr bgcolor="#CCCCCC">

<td>Öncelik ? </td>

<td>:</td>

<td><select name="oncelik" id="oncelik">

<option value="1">Evet</option>

<option value="0" selected>Hayır</option>

</select></td>

</tr>

<tr>

<td> </td>

<td> </td>

<td><input type="submit" name="ekle" value=" Ekle ">

<input type="reset" name="sil" value="Temizle"> </td>

</tr>

</table>

```
<script language='javascript1.2'>
editor_generate('detay');
WriteMenuItems('detay');
</script>
<table width="860" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="610">&nbsp;</td>
  </tr>
  <tr>
    <td class="12redx"><strong>Açıklamalar !</strong></td>
  </tr>
  <tr>
    <td>- (*) işaretli alanları boş bırakmayın.<br>
      - Tarih formatına uygun girilmeli (gün-ay-yıl) şeklinde. örn. 05-10-2006
      gibi<br>
      - Haber başlığı mümkün olduğunca kısa olmalı. Uzun başlıklar anasayfadaki
      görünüm bozacaktır.<br>
      - Yazının ekleneceği kategoriye listeden seçin. <br>
      - Detay kısmında kopyala-yapıştır şeklinde haber eklenecek ise, metni önce
      dreamweaver adlı programda düzenleyip html kodlarını yapıştırın. Metinde resim
      olmamasına dikkat edin. Çok uzun metinleri kısaltın. Manşetler için özet-detay
      kullanmalısınız. Bunun için yaklaşık 250 karakterlik bir özet oluşturup detay kısmının
      başına yapıştırın. Hemen sonrasında @detay@ ibaresini koyup hemen ardına haberin
      detayını yapıştırın.<br>
      - Haberiniz anasayfada manşet olarak eklemek için Manşet? alanından Evet i
      seçin.<br>
      - Manşet alanında Evet i seçtiyseniz manşet haberin nasıl görüneceğini seçin.
      Haberiniz manşet değilse bu alanı boş geçin.<br>
      - Haberiniz manşet olacaksa seçtiğiniz görünüme göre oluşturulan resminizi
      seçin. Kutucuğun altındaki ölçülere muylaka dikkat edin. Farklı ölçüler resmin kırık
      görünmesine sebep olur.<br>
      - Resim 1 ve Resim 2 opsiyoneldir. Genişliği en fazla 600 pixel olmalı. Daha
      büyük resimler sayfanın yapısını bozacaktır. Format olarak jpg, png ve gif tercih edin.
```

Kullandığınız grafik programında resmin kenarına ekleyeceğiniz 1 pixellik çerçeve haberin daha güzel görünmesini sağlayacaktır.

- Kategori sayfasında haberin öncelikli olarak görünmesi için Öncelik? alanını evet seçin. Haber manşet ise burada Eveti seçmenize gerek yoktur.</td>

</tr>

</table>

<script language="javascript1.2">

var config = new Object(); // create new config object

config.width = "90%";

config.height = "200px";

config.bodyStyle = 'background-color: white; font-family: "Verdana"; font-size: x-small;';

config.debug = 0;

// NOTE: You can remove any of these blocks and use the default config!

config.toolbar = [

['fontname'],

['fontsize'],

['fontstyle'],

['linebreak'],

['bold','italic','underline','separator'],

// ['strikethrough','subscript','superscript','separator'],

['justifyleft','justifycenter','justifyright','separator'],

['OrderedList','UnOrderedList','Outdent','Indent','separator'],

['forecolor','backcolor','separator'],

['line','Insertlink','InsertImage','htmlmode','separator'],

['about','help','popupeditor'],

];

config.fontnames = {

"Arial": "arial, helvetica, sans-serif",

"Courier New": "courier new, courier, mono",

```

"Georgia":      "Georgia, Times New Roman, Times, Serif",
"Tahoma":      "Tahoma, Arial, Helvetica, sans-serif",
"Times New Roman": "times new roman, times, serif",
"Verdana":      "Verdana, Arial, Helvetica, sans-serif",
"impact":      "impact",
"WingDings":    "WingDings"
};

config.fontsizes = {
  "1 (8 pt)": "1",
  "2 (10 pt)": "2",
  "3 (12 pt)": "3",
  "4 (14 pt)": "4",
  "5 (18 pt)": "5",
  "6 (24 pt)": "6",
  "7 (36 pt)": "7"
};

//config.stylesheet = "http://www.domain.com/sample.css";

config.fontstyles = [ // make sure classNames are defined in the page the content is
being display as well in or they won't work!
  { name: "headline",   className: "headline", classStyle: "font-family: arial black,
arial; font-size: 28px; letter-spacing: -2px;" },
  { name: "arial red",  className: "headline2", classStyle: "font-family: arial black,
arial; font-size: 12px; letter-spacing: -2px; color:red" },
  { name: "verdana blue", className: "headline4", classStyle: "font-family: verdana;
font-size: 18px; letter-spacing: -2px; color:blue" }

// leave classStyle blank if it's defined in config.stylesheet (above), like this:
// { name: "verdana blue", className: "headline4", classStyle: "" }
];

```

```

</script>
</form></td>
</tr>
</table>
</body>
</html>

```

4.2.3 kategoriler.php

```

<?php ob_start(); ?>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<link rel="stylesheet" href="./stil.css" type="text/css">
<SCRIPT language=JavaScript src="scrp.js"></SCRIPT>

<style type="text/css">
<!--
body {
    margin-left: 15px;
    margin-top: 0px;
    background-image: url(images/obg.jpg);
    background-position-x: center;
    background-repeat: repeat-y;
    background-color: #FFFFFF;
}
.style5 {color: #FFFFFF}
-->
</style></head>

<body>
<table width="870" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="225" valign="top" class="style5"><table width="224" height="18"
border="0" cellpadding="0" cellspacing="0" background="images/yerz.gif">

```



```

        <td colspan="2"><br><span class="bordo12pxbold"><strong>$ozet</strong>
kategori eklendi.</span></td>
    </tr>";

    if ($islem=="duzenle")
    echo "<tr>
        <td colspan="2"><br><span class="bordo12pxbold"><strong>$ids</strong>
isimli kategori deđiřtirildi.</span></td>
    </tr>";
    ?>
    <tr>
        <td colspan="2">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="2"><table width="870" border="0" cellspacing="1" cellpadding="1">
            <tr bgcolor="#666666">
                <td width="30" bgcolor="#666666" class="style5"><div
align="center"><strong>S.No</strong></div></td>
                <td width="430" bgcolor="#666666"
class="style5"><strong>&nbsp;&nbsp;&nbsp;İSİM</strong></td>
                <td width="70" class="style5"><div
align="center"><strong>Öncelik</strong></div></td>
                <td width="70" class="style5"><div
align="center"><strong>Sıralama</strong></div></td>
                <td width="70" class="style5"><div
align="center"><strong>Link</strong></div></td>
                <td width="100" bgcolor="#666666" class="style5"><div
align="center"><strong>Sil</strong></div></td>
                <td width="100" bgcolor="#666666" class="style5"><div
align="center"><strong>Düzenle</strong></div></td>
            </tr>
        </table>
    </tr>
    <?php
        $SQL = "SELECT * FROM kategoriler ORDER BY oncelik ASC,siralama
ASC";

```

```

$sorgu=mysql_query($SQL,$connect);
if (! $sorgu)
{
    echo "Veritabani hatası ..!<BR>Hatalı SQL sorgulaması...";
    mysql_close($connect);
    exit;
}
$i=1;
$x=0;
$count=mysql_num_rows($sorgu);
echo" <tr>
<td colspan=7 height=2></td></tr>";
while ($record=mysql_fetch_array($sorgu))
{
    $id=$record["id"];
    $isim=$record["isim"];
    $oncelik=$record["oncelik"];
    $siralama=$record["siralama"];
    $link=$record["link"];
    if ($link!="")
    $link="<img src=\"images/link.jpg\">";

    if ($oncelik!="1")
    $oncelik="<img src=\"images/alt.gif\">";
    else
    $oncelik="<img src=\"images/ust.gif\">";

    if ($x==0) {$renk="#CCCCCC"; $x=1;} else
    {$renk="#E9E9E9"; $x=0;}

    echo" <tr bgcolor=\"{$renk}\">
<td><div align=\"center\">$i</div></td>
<td>&nbsp;<font size=\"4\"><strong>$isim</strong></font></td>
    <td align=\"center\">$oncelik</td>
    <td align=\"center\">$siralama</td>

```

```
<td align=\"center\">$link</td>
```

```
<td><div align=\"center\"><a href=\"kategorisil.php?id=$id\"><img  
src=\"images/sil1.gif\" alt=\"Kaydı ve resim/dosya sil\" border=\"0\"></a></div></td>
```

```
<td><div align=\"center\"><a href=\"kategoriduzenle.php?id=$id\"><img  
src=\"images/duzenle1.gif\" alt=\"Kaydı düzenle\" border=\"0\"></a></div></td>";
```

```
$i++;
```

```
}
```

```
$k=$i-1;
```

```
echo" <tr>
```

```
<td colspan=7 height=2></td></tr>";
```

```
echo" <tr bgcolor=\"#666666\">
```

```
<td colspan=7><span class=\"style5\">Toplam kayıt sayısı :
```

```
$k</span></td></tr>";
```

```
?>
```

```
</table></td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

4.2.4 yaziekle.php

```
<?php ob_start(); ?><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
```

```
Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Untitled Document</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
<link rel="stylesheet" href="/stil.css" type="text/css">
```

```
<SCRIPT language=JavaScript src="scrp.js"></SCRIPT>
```

```
<script language="Javascript1.2"><!-- // load htmlarea
```

```
_editor_url = ""; // URL to htmlarea files
```

```
_editor_field = "";
```

```

var win_ie_ver = parseFloat(navigator.appVersion.split("MSIE")[1]);
if (navigator.userAgent.indexOf('Mac')    >= 0) { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Windows CE') >= 0) { win_ie_ver = 0; }
if (navigator.userAgent.indexOf('Opera')    >= 0) { win_ie_ver = 0; }
if (win_ie_ver >= 5.5) {
    document.write('<scr' + 'ipt src="' + _editor_url + 'editor.js"');
    document.write(' language="Javascript1.2"></scr' + 'ipt>');
} else { document.write('<scr'+ 'ipt>function editor_generate() { return false;
} </scr'+ 'ipt>'); }
var MyBgColor = '#FFFFFF';
var MyBgImg = "";
// --></script>

<script>
function updateAttributes(){
document.myform.bgcolor.value=MyBgColor;
document.myform.background.value=MyBgImg;
}
</script>
<script language="Javascript1.2">
function disableIt(obj)
{
    obj.disabled = !(obj.disabled);
    var z = (obj.disabled) ? 'disabled' : 'enabled';
    //alert(obj.type + ' now ' + z);
}
</script>
<style type="text/css">
<!--
body {
    margin-left: 15px;
    margin-top: 0px;
    background-image: url(images/obg.jpg);
    background-position-x: center;
    background-repeat: repeat-y;

```



```

<td>:</td>
<td><?php

$gsayi=$_GET["sayi"];

echo "<strong>$gsayi</strong>";

?>

<input name="esayi" type="hidden" id="esayi" value="<?php echo
$gsayi;?>"></td>
</tr>
<tr bgcolor="#E9E9E9">
<td width="128"><strong>Yazı Tarihi <span
class="bordo12pxbold">(*)</span></strong></td>
<td width="12">:</td>
<td width="610"><input name="tarih" type="text" id="tarih" style="border-
style:solid; border-width:1;" value="<?php echo date("d-m-Y");?>" size="6"
maxlength="10">
<span class="fonton">(gün-ay-yıl şeklinde, 05-10-2006 gibi)</span></td>
</tr>
<tr bgcolor="#CCCCCC">
<td><strong>Yazı Başlığı (*)</strong></td>
<td>:</td>
<td><input name="baslik" type="text" id="baslik" style="border-style:solid;
border-width:1;" size="60" maxlength="120">
<span class="fonton">(max. 120 karakter)</span></td>
</tr>
<tr bgcolor="#E9E9E9">
<td><strong>Yazarı (*)</strong></td>
<td>:</td>
<td><?php
echo "<select name=\"yazar\">";
$SQL = "SELECT * FROM yazarlar order by isim ASC";

$sorgu = mysql_query($SQL,$connect);

```

```

        while ($record = mysql_fetch_array($sorgu))
        {
            $y_isim=$record["isim"];
            $yid=$record["id"];
            echo "<option value=\"\$yid\">$y_isim</option>";
        }

        echo "</select>";
        ?></td>

</tr>
<tr bgcolor="#CCCCCC">
    <td valign="top"><strong>Yazı Detayı (*)</strong></td>
    <td valign="top">:</td>
    <td><textarea name="detay" cols="70" rows="12" class="FieldStyle1"
id="detay"></textarea>
        </td>
</tr>
<tr bgcolor="#E9E9E9">
    <td><strong>Öncelik ? </strong></td>
    <td>:</td>
    <td><select name="oncelik" id="oncelik">
        <option value="1">Evet</option>
        <option value="0" selected>Hayır</option>
        </select></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td><input type="submit" name="ekle" value=" Ekle ">
        <input type="reset" name="sil" value="Temizle"> </td>
</tr>
</table>

<script language='javascript1.2'>
editor_generate('detay');

```

```
WriteMenuItems('detay');
```

```
</script>
```

```
<table width="860" border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td width="610">&nbsp;</td>
```

```
</tr>
```

```
<tr>
```

```
<td class="12redx"><strong>Açıklamalar !</strong></td>
```

```
</tr>
```

```
<tr>
```

```
<td>- (*) işaretli alanları boş bırakmayın.<br>
```

- Tarih formatına uygun girilmeli (gün-ay-yıl) şeklinde. örn. 05-10-2006

gibi

- Yazı başlığı mümkün olduğunca kısa olmalı. Uzun başlıklar sayfadaki

görünüm bozabilir.

- Yazıyı gönderen yazarı listeden seçin.

- Detay kısmında kopyala-yapıştır şeklinde haber eklenecek ise, metni önce dreamweaver adlı programda düzenleyip html kodlarını yapıştırın. Metinde resim olmamasına dikkat edin.

- Bu yazının öncelikli olmasını istiyorsanız öncelik? listesinden Evet seçeneğini seçebilirsiniz. </td>

```
</tr>
```

```
</table>
```

```
<script language="javascript1.2">
```

```
var config = new Object(); // create new config object
```

```
config.width = "90%";
```

```
config.height = "200px";
```

```
config.bodyStyle = "background-color: white; font-family: "Verdana"; font-size: x-small;";
```

```
config.debug = 0;
```

```
// NOTE: You can remove any of these blocks and use the default config!
```

```

config.toolbar = [
    ['fontname'],
    ['fontsize'],
    ['fontstyle'],
    ['linebreak'],
    ['bold','italic','underline','separator'],
    // ['strikethrough','subscript','superscript','separator'],
    ['justifyleft','justifycenter','justifyright','separator'],
    ['OrderedList','UnOrderedList','Outdent','Indent','separator'],
    ['forecolor','backcolor','separator'],
    ['line','Insertlink','InsertImage','htmlmode','separator'],
    ['about','help','popUpEditior'],
];

config.fontnames = {
    "Arial":      "arial, helvetica, sans-serif",
    "Courier New": "courier new, courier, mono",
    "Georgia":    "Georgia, Times New Roman, Times, Serif",
    "Tahoma":    "Tahoma, Arial, Helvetica, sans-serif",
    "Times New Roman": "times new roman, times, serif",
    "Verdana":    "Verdana, Arial, Helvetica, sans-serif",
    "impact":     "impact",
    "WingDings":  "WingDings"
};

config.fontsizes = {
    "1 (8 pt)": "1",
    "2 (10 pt)": "2",
    "3 (12 pt)": "3",
    "4 (14 pt)": "4",
    "5 (18 pt)": "5",
    "6 (24 pt)": "6",
    "7 (36 pt)": "7"
};

```



```
//config.stylesheet = "http://www.domain.com/sample.css";
```

```
config.fontstyles = [ // make sure classNames are defined in the page the content is  
being display as well in or they won't work!
```

```
  { name: "headline",   className: "headline", classStyle: "font-family: arial black,  
arial; font-size: 28px; letter-spacing: -2px;" },
```

```
  { name: "arial red",   className: "headline2", classStyle: "font-family: arial black,  
arial; font-size: 12px; letter-spacing: -2px; color:red" },
```

```
  { name: "verdana blue", className: "headline4", classStyle: "font-family: verdana;  
font-size: 18px; letter-spacing: -2px; color:blue" }
```

```
// leave classStyle blank if it's defined in config.stylesheet (above), like this:
```

```
// { name: "verdana blue", className: "headline4", classStyle: "" }
```

```
];
```

```
</script>
```

```
</form></td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

CONCLUSION

The web site project which is a web site of a weekly newspaper was made to serve the people on the web line. There are several methods that includes technical and the satisfaction of the peoples. Technical methods were formed from some web developing tools and programs such as PHP, Apache, MySQL. All of these tools and the components been discussed and explored briefly.

The satisfaction of the visitors is not ignored. To make it we design the site constitute, interactiveness, understandable. Easiness, easy connection, good governance, good public relations and efficiency.

REFERENCES

1. <http://www.php.com>
2. <http://www.mysql.com>
3. <http://www.apache.com>
4. <http://www.netshinesoftware.com>