



Jaypee University of Information Technology
Solan (H.P.)
LEARNING RESOURCE CENTER

Acc. Num. SP06066 Call Num:

General Guidelines:

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

Learning Resource Centre-JUIT



SP06066

CONTENT MANAGEMENT SYSTEM

Submitted in partial fulfillment of the Degree of Bachelor of
Technology

By

NISHA SHWETA-061266

VIPUL SHARMA-061309

ISHAN JANARTHA-061329

SUMEET SHARMA-061454



This is to certify that the work entitled "Content Management System" submitted by Nisha Shweta(061266), Vipul Sharma(061309), Ishan Janartha(061329), Sumeet Sharma(061454) in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology in the month of May 2010 at Jaypee University of Information Technology, Wazirpur, Solan (H.P.) has not been submitted partially or wholly to any other university or institution for the award of any other degree or diploma.

MR. SUNIL K. PATEL

MAY-2010

**Submitted in partial fulfillment of the Degree of Bachelor of
Technology**

**DEPARTMENT OF COMPUTER SCIENCE
JAYPEE UNIVERSITY OF INFORMATION
TECHNOLOGY-WAKHNAGHAT**

MAY-2010

CERTIFICATE

This is to certify that the work entitled, "Content Management System" submitted by Nisha Shewta(061266), Vipul Sharma(061309), Ishan Janartha(061329), Sumeet Sharma(061454)" in partial fulfillment for the award of degree of Bachelor of Technology in CSE&IT of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.


Mr. SANJEEV PATEL

ACKNOWLEDGEMENT

The project team owes a deep sense of reverence to esteemed Mr. Sanjeev Patel for his painstaking and close guidance, constant encouragement and constructive suggestions. Without their able guidance and constructive criticism, it would not have been possible for us to bring out this report in its present form.

We also take privilege to record our heartiest thanks to the other faculty members of the department for their suggestions and help.

We would fail in our duty if we do not acknowledge the assistance extended to us by the laboratory staff in letting us work in the laboratory even during odd hours and for making available all the resources required.

We would also like to thank our dearest colleagues whose constant suggestions regarding our project report was unbounded source of inspiration for us.

Nisha Shweta(061266) *Nisha*

Vipul Sharma(061309) *Vipul*

IshanJanartha(061329) *Ishan*

SumeetSharma(061454) *Sumeet
Sharma*

TABLE OF CONTENTS

CERTIFICATE.....	3
ACKNOWLEDGEMENT.....	4
CHAPTER 1 INTRODUCTION.....	8
1.1 <u>Content Management System</u>	8
1.2 <u>Objective</u>	8
CHAPTER 2 INITIAL INVESTIGATION.....	9
2.1 <u>Capabilities</u>	9
2.2 <u>Types</u>	10
2.3 <u>Feasibility Study</u>	12
CHAPTER 3 ANALYSIS.....	13
3.1 <u>Planning the CMS</u>	13
3.2 <u>Object-Oriented Programming</u>	14
3.3 <u>Technologies Used</u>	17
CHAPTER 4 DESIGN.....	20
4.1 <u>CMS Structure</u>	20
4.2 <u>Includes Folder</u>	21
4.3 <u>Validation</u>	23
4.4 <u>Cmsadmin Folder</u>	24
4.5 <u>Common Folder</u>	24
4.6 <u>Templates Folder</u>	25

4.7	<u>Tables</u>	27
4.7.1	<u>Cmsarticles</u>	27
4.7.2	<u>Sections</u>	28
4.7.3	<u>Cmsgroups</u>	28
4.7.4	<u>Cmsusers</u>	29
4.7.5	<u>Slideshow</u>	29
4.7.6	<u>Settings</u>	30
4.7.7	<u>Cmspages</u>	31
4.7.8	<u>Cmsarticlespage</u>	31

CHAPTER 5 FEATURES.....32

5.1	<u>Add article</u>	33
5.2	<u>Remove article</u>	34
5.3	<u>View articles</u>	35
5.4	<u>Edit sections</u>	36
5.5	<u>View sections</u>	37
5.6	<u>Select Theme</u>	38

CHAPTER 6 THE WEB SITES.....40

6.1	<u>The Doors</u>	40
6.2	<u>Shimla</u>	41

CHAPTER 7 INSTALLATION.....42

7.1	<u>Requirement</u>	42
-----	--------------------	----

FUTURE WORK.....	42
CONCLUSION.....	43
BIBLIOGRAPHY.....	43
APPENDIX.....	44

CODE FOR VARIOUS PAGES 44

<u>SystemComponent Class&DbConnector Class</u>	45
<u>Common-head.php</u>	46
<u>Index&Pageindex.php</u>	47
<u>Addarticle.php</u>	48
<u>Removearticle.php</u>	51
<u>Viewarticles.php</u>	52
<u>Editsections.php</u>	53
<u>Viewsections.php</u>	54
<u>Addgallery&Addpage.php</u>	55
<u>Addtopage.php</u>	56
<u>Removepage.php</u>	62
<u>Removepagearticle.php</u>	65
<u>SelectTheme.php</u>	68
<u>Widgets.php</u>	71
Theme: <u>Home.php</u>	73
<u>Page.php</u>	78

CHAPTER 1

INTRODUCTION

Content Management System

A Web content management system (WCMS or Web CMS) is content management system (CMS) software, usually implemented as a Web application, for creating and managing HTML content. It is used to manage and control a large, dynamic collection of Web material (HTML documents and their associated images). A WCMS facilitates content creation, content control, editing, and many essential Web maintenance functions.

Usually the software provides authoring (and other) tools designed to allow users with little or no knowledge of programming languages or markup languages to create and manage content with relative ease of use.

Most systems use a database to store content, metadata, and/or artifacts that might be needed by the system. Content is frequently, but not universally, stored as XML, to facilitate reuse and enable flexible presentation options.

A presentation layer displays the content to regular Web-site visitors based on a set of templates. The templates are sometimes XSLT files.

Administration is typically done through browser-based interfaces, but some systems require the use of a fat client.

Unlike Web-site builders like Microsoft FrontPage or Adobe Dreamweaver, a WCMS allows non-technical users to make changes to an existing website with little or no training. A WCMS typically requires an experienced coder to set up and add features, but is primarily a Web-site maintenance tool for non-technical administrators.

Objective

“To implement a content management system that captures, manages and distributes content in a collaborative fashion that minimizes the need for web authors to have significant technical skills in web design, implementation and publishing”

CHAPTER 2

Initial Investigation

Capabilities

A WCMS is a software system used to manage and control a large, dynamic collection of Web material (HTML documents and their associated images). A CMS facilitates document control, auditing, editing, and timeline management. A WCMS provides the following key features:

Automated templates

Create standard output templates (usually HTML and XML) that can be automatically applied to new and existing content, allowing the appearance of all content to be changed from one central place.

Easily editable content

Once content is separated from the visual presentation of a site, it usually becomes much easier and quicker to edit and manipulate. Most WCMS software includes WYSIWYG editing tools allowing non-technical individuals to create and edit content.

Scalable feature sets

Most WCMS software includes plug-ins or modules that can be easily installed to extend an existing site's functionality.

Web standards upgrades

Active WCMS software usually receives regular updates that include new feature sets and keep the system up to current web standards.

Workflow management

Workflow is the process of creating cycles of sequential and parallel tasks that must be accomplished in the CMS. For example, a content creator can submit a story, but it is not published until the copy editor cleans it up and the editor-in-chief approves it.

Delegation

Some CMS software allows for various user groups to have limited privileges over specific content on the website, spreading out the responsibility of content management.

Document management

CMS software may provide a means of managing the life cycle of a document from initial creation time, through revisions, publication, archive, and document destruction.

Content virtualization

CMS software may provide a means of allowing each user to work within a virtual copy of the entire Web site, document set, and/or code base. This enables changes to multiple interdependent resources to be viewed and/or executed in-context prior to submission.

TYPES

There are three major types of WCMS: offline processing, online processing, and hybrid systems. These terms describe the deployment pattern for the WCMS in terms of when presentation templates are applied to render Web pages from structured content. Seth Gottlieb has used the terms 'baking', 'frying', and 'parbaking' to describe the three alternatives.

Offline processing

These systems pre-process all content, applying templates before publication to generate Web pages. Vignette CMS and Bricolage are examples of this type of system. Since pre-processing

systems do not require a server to apply the templates at request time, they may also exist purely as design-time tools; Adobe Contribute is an example of this approach.

Online processing

These systems apply templates on-demand. HTML may be generated when a user visits the page, or pulled from a cache. Hosted CMSs are provided by such SaaS developers as AspireCMS, Bravenet, UcoZ, Freewebs.

Some of the better known open source systems that produce pages on demand are Mambo, Joomla!, Drupal, TYPO3, Zikula and Plone.

DotNetNuke is a partially open source CMS that runs on asp.net and is free to download and install. DNN produces pages on demand but levels and types of caching can be set. There are also many additional "modules" that can be purchased or installed for free to extend the functionality of DNN as needed, many of which create data and content dynamically.

Most Web application frameworks perform template processing in this way, but they do not necessarily incorporate content management features. Wikis, e.g. MediaWiki and TWiki generally follow an online model (with varying degrees of caching), but generally do not provide document workflow.

Hybrid Systems

Some systems combine the offline and online approaches. Some systems write out executable code (e.g. JSP, ASP, PHP, ColdFusion, Perl) rather than just static HTML, so that the CMS itself does not need to be deployed on every Web server. Other hybrids, such as Blossom, are capable of operating in either an online or offline mode.

FEASIBILITY STUDY

Creating a PHP-Based Content Management System

If you're going to run an intranet site, then you'll probably want a content management system (CMS) — a tool used to organize documents and keep track of what's where.

It's not like the off-the-shelf systems lack features or stability. On the contrary, many have been crafted by hundreds of man-hours of work, and are successfully implemented by thousands of Web sites and intranets. But when it comes down to it, it's hard to have much clue as to how they work. If you want to customize the way these systems operate, you'll often have to wade through vast amounts of (often badly documented) code to find what needs changing.

Writing your own CMS, on the other hand, can lead to a solution that is better suited to your requirements, better addresses the needs of your users, and is better understood by your development team. If you have the time and expertise to write your own in-house system, it may well prove the better option.

The system we create will be written using the **PHP programming language**, which excels in the development of Web-based systems. We'll be using **MySQL** as the database server, but the system will be written to allow the use of alternative databases, such as PostgreSQL or SQL Server.

So what will this system actually do? First and foremost, it will allow the bulk of the intranet or Internet site's content to be easily stored and managed in a database. We'll also include a number of other features required for running a successful site, such as authenticating users and managing files.

CHAPTER 3

ANALYSIS

Planning the CMS

To begin with, we'll plan how our PHP-based content management system will work.

The first step is a basic specification of what our CMS must do:

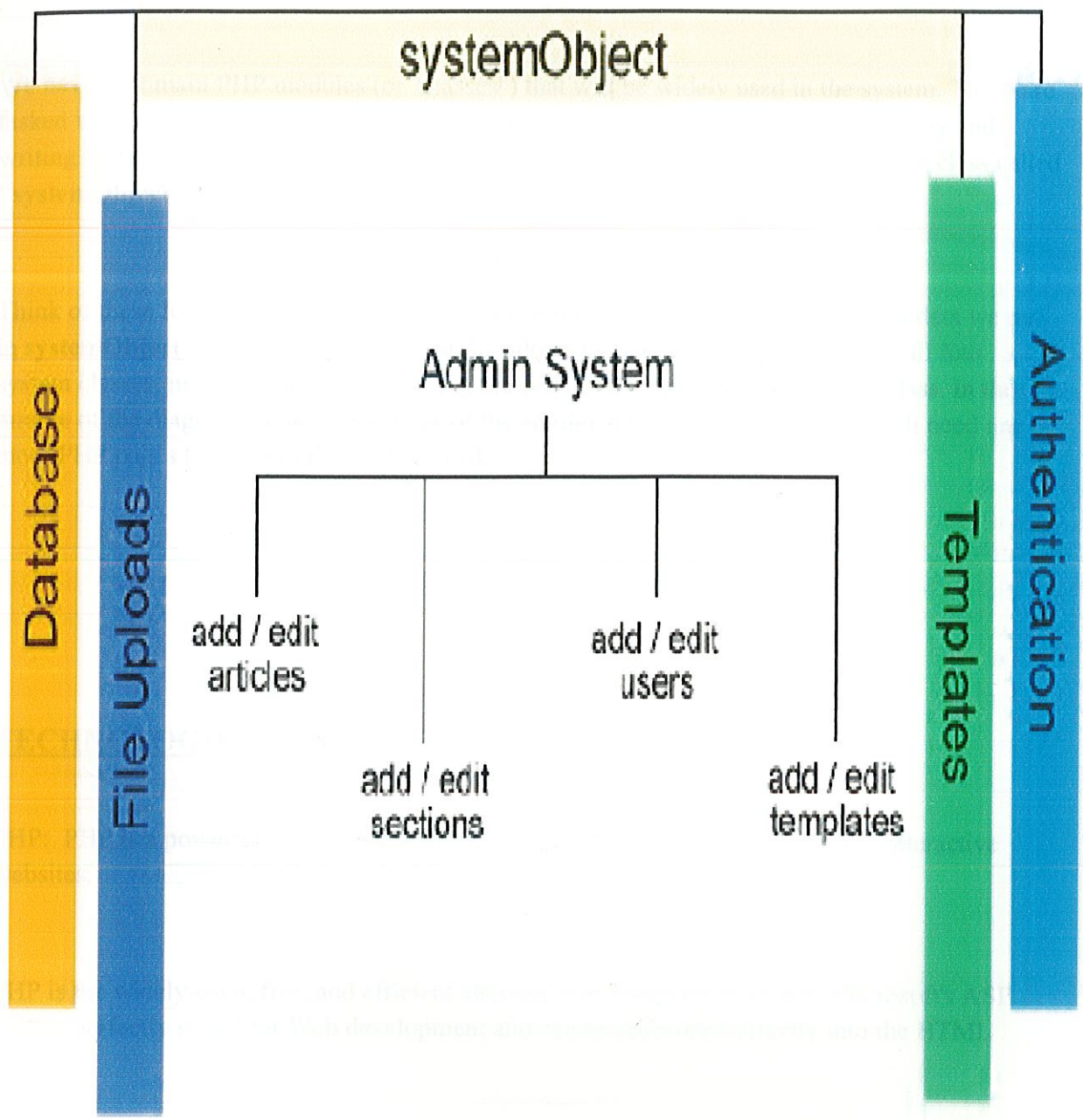
- **Content Management:** Probably the most vital function of the system, it must store content such as documents and news in a database, and display to the user whatever he or she requests. An easy-to-use interface is required to allow editors to add, remove, or modify content.
- **User authentication:** There may be certain areas of the intranet or Internet site to which we wish to limit access. At the very least this will be the "admin" area, where the editor of the site will be able to add, edit or modify content. You may also wish to have areas only available to certain departments or staff.
- **Page uniformity/templates:** The system should have a uniform look and feel, and this design element needs to be separated from the logic element, e.g., the programming required to display an article should be separated from how that article looks (stylistically) on the screen.

Object-Oriented Programming

PHP helps the design process by supporting object-oriented programming (OOP). When putting together our system, there are certain chunks of programming that are needed again and again, such as database access, user authentication, etc. To keep this code neat and tidy, we bundle it together in PHP files called "**classes.**" We can then create **instances (or "objects")** of these classes whenever they are needed. Thus, the class can be thought of as a blueprint for one or more instances.

We create a class with code for connecting to a database, and then create an instance of that class whenever we need to query the database. This method of programming allows a complex system to be broken down into smaller and simpler blocks, which makes life easier when it comes to management, modification, and error finding.

Let's now consider how the system will fit together. But below is a basic outline:



We have four main PHP modules (or "classes") that will be widely used in the system. These are tasked with accessing the database, allowing the user to upload files to the site, reading and writing templates, and logging users in and out. These classes all "extend" one parent class called "systemObject."

Think of these four as being independent of one another, yet all inheriting whatever data we put in **systemObject**. This technique of hierarchy allows us to make changes effecting all four system classes, just by adding or modifying the code in the **systemObject** parent class. In the middle of the diagram are the basic areas of the administration system, and each will need one or more PHP pages to perform the required tasks.

TECHNOLOGIES USED

PHP: PHP is a powerful server-side scripting language for creating dynamic and interactive websites.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code.

The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.

MySQL: MySQL is a relational database management system (RDBMS) which has more than 11 million installations. The program runs as a server providing multi-user access to a number of databases.

MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now a subsidiary of Sun Microsystems, which holds the copyright to most of the codebase. The project's source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements.

Apache: The Apache HTTP Server, commonly referred to simply as Apache is a web server notable for playing a key role in the initial growth of the World Wide Web. Apache was the first viable alternative to the Netscape Communications Corporation web server (currently known as Sun Java System Web Server), and has since evolved to rival other Unix-based web servers in terms of functionality and performance. The majority of all web servers using Apache are Linux web servers.

Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The application is available for a wide variety of operating systems, including Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF, and eComStation. Released under the Apache License, Apache is characterized as free software and open source software.

Since April 1996 Apache has been the most popular HTTP server on the World Wide Web. As of December 2008 Apache served over 51% of all websites.

XAMPP: XAMPP is a free and open source cross-platform web server package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

The program is released under the terms of the GNU General Public License and acts as a free web server capable of serving dynamic pages. XAMPP is available for Microsoft Windows, Linux, Solaris, and Mac OS X, and is mainly used for web development projects.

LAMP: The acronym LAMP refers to a solution stack of software, usually free and open source software, used to run dynamic Web sites or servers. The original expansion is as follows:

Linux, referring to the operating system;

Apache, the Web server;

MySQL or mSQL, the database management system (or database server);

PHP or others, i.e., Perl, Python, the programming languages.

The combination of these technologies is used primarily to define a web server infrastructure, define a programming paradigm of developing software, and establish a software distribution package.

Though the originators of these open source programs did not design them all to work specifically with each other, the combination has become popular because of its low acquisition cost and because of the ubiquity of its components (which come bundled with most current Linux distributions). When used in combination they represent a solution stack of technologies that support application servers.

YouTube API: The YouTube APIs and Tools enable you to integrate YouTube's video content and functionality into your website, software application, or device.

The Video Search Control is a simple to use application of the Google AJAX Search API that is designed to let you easily add a video search form and a block of videos to your pages, sites, and blogs.

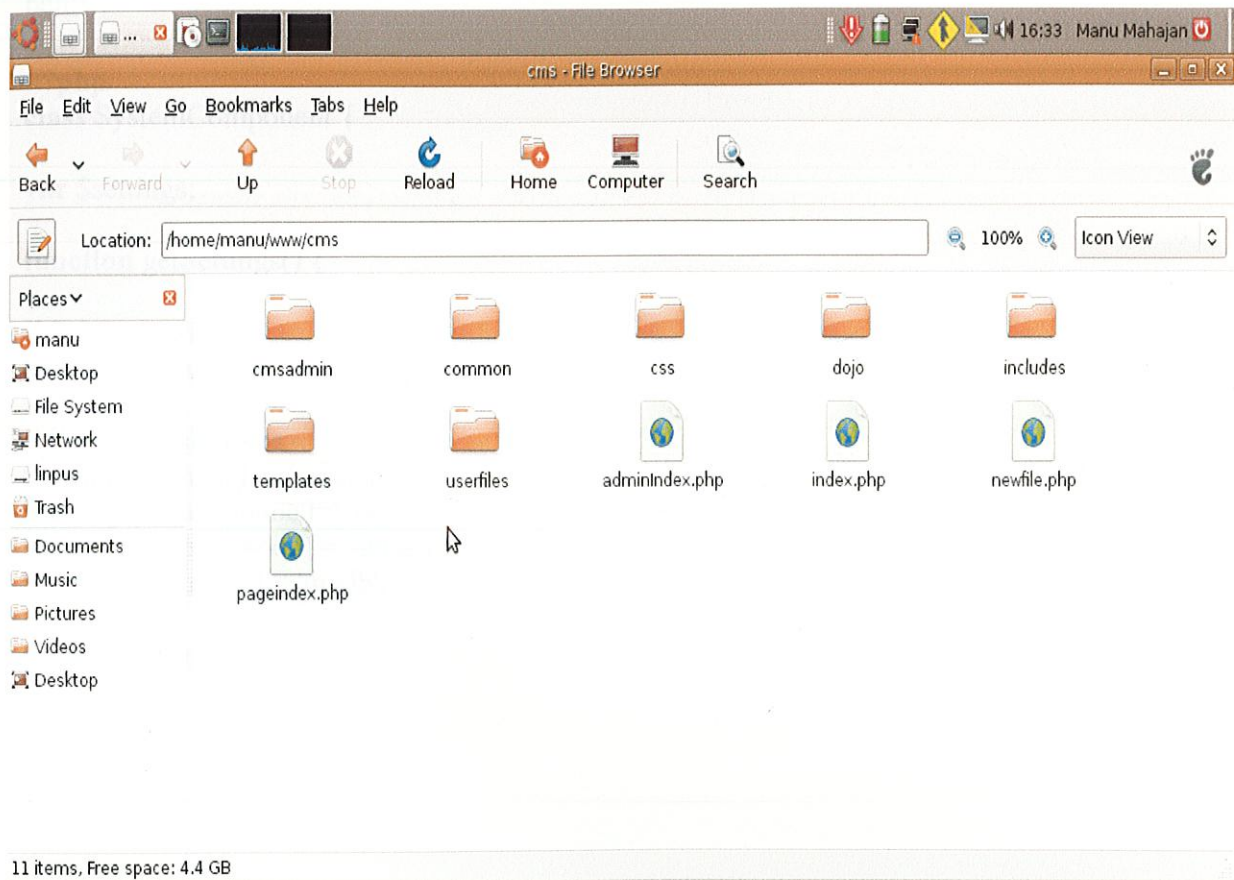
DOJO: The Dojo Toolkit is an open source modular JavaScript library (or more specifically JavaScript toolkit) designed to ease the rapid development of cross platform, JavaScript/Ajax based applications and web sites. The Dojo Foundation is a non-profit organization designed to promote the adoption of the toolkit.

CHAPTER 4

DESIGN

CMS Structure

Our CMS will be stored in a number of folders, structured as follows:



Includes Folder

We're going to start by creating the PHP class which all others will "extend." This will be the root of the administration system, and anything we put in it (such as variables and functions) will trickle down to the other classes.

This root class will be called '**SystemComponent**'. The code follows, and a full explanation is below:

```
<?php
class SystemComponent {

var $settings;

function getSettings() {

// System variables
$settings['siteDir'] = '/path/to/your/intranet/';

// Database variables
$settings['dbhost'] = 'hostname';
$settings['dbusername'] = 'dbuser';
$settings['dbpassword'] = 'dbpass';
$settings['dbname'] = 'mydb';

return $settings;

}

}
?>
```

The code starts off by telling PHP that our class will be called '**SystemComponent**'. Between the braces we declare the variable **\$settings**, and a function called '**getSettings**'. The purpose of this is to store a number of values in **\$settings**, containing the path on the server to the intranet ('siteDir'), and the details of the database. This system uses **MySQL**. Finally, the 'return' command sends **\$settings** to whichever class or function has requested it.



All of the information to be displayed in our Content Management System will be stored in a database. It is sensible, therefore, to create a reusable PHP class that we can call upon whenever we need to access our data. The code listed here is for connecting to a MySQL database.

After we've named the class '**DbConnector**', we state '**extends SystemComponent**'. This tells PHP to grab all of the data and functions from SystemComponent, and provide us with access to them.

```
<?php
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Class: DbConnector
// Purpose: Connect to a database, MySQL version
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
require_once 'SystemComponent.php';

class DbConnector extends SystemComponent {

var $theQuery;
var $link;

/** Function: DbConnector, Purpose: Connect to the database **
function DbConnector(){

// Load settings from parent class
$settings = SystemComponent::getSettings();

// Get the main settings from the array we just loaded
$host = $settings['dbhost'];
$db = $settings['dbname'];
$user = $settings['dbusername'];
$pass = $settings['dbpassword'];

// Connect to the database
$this->link = mysql_connect($host, $user, $pass);
mysql_select_db($db);
register_shutdown_function(array(&$this, 'close'));

}/** Function: query, Purpose: Execute a database query **
function query($query) {

$this->theQuery = $query;
return mysql_query($query, $this->link);
```

```

}/** Function: fetchArray, Purpose: Get array of query results **
function fetchArray($result) {

return mysql_fetch_array($result);

}/** Function: close, Purpose: Close the connection **
function close() {

mysql_close($this->link);

}

?>

```

The first function, 'DbConnector', has the same name as the class that contains it, meaning it's run automatically when DbConnector loads. It firstly calls the 'getSettings' function we wrote earlier, and extracts from it the various database settings. It then uses these settings to connect to the database.

The other functions are explained below:

Function	Purpose
query	Execute a database query
fetchArray	Create an array containing each record found using the 'query' function (above)
close	Closes the database connection. The register_shutdown_function command in the DbConnector function ensures this happens automatically when the object is no longer in use.

Validation

Few basic types of data we need to validate:

- General: Just check something was typed in
- Text Only (i.e., no punctuation or other symbols allowed)
- Text Only and no white spaces allowed
- E-Mail addresses
- Numbers
- Dates

Cmsadmin Folder

Contains the PHP pages of the administrator area:

- adminlogin.php
- addarticle.php
- removearticle.php
- viewarticles.php
- editsections.php
- viewsections.php
- addgallery.php
- viewgallery.php
- removegallery.php
- selectTheme.php
- widgets.php
- addpage.php
- addtopage.php
- removepage.php
- removepagearticle.php

Common Folder

Common-head.php

Contains links to all css files and javascript files that are common to all pages.

Slideshow.php

Code for the slideshow.

Youtube.php

Code for using the youtube API.

Tree.php

Code for generating the DOJO tree in the administrator area.

Globals.php

Contains all the global variables.

Templates Folder

Contains folders for all the themes.

Theme Folder

- Home.php : For the Source Code refer to the Appendix Page No. 68
- Page.php : For the Source Code refer to the Appendix Page No. 73
- Images Folder
- Common Folder

Creating the Database

The **first table** we're going to add to our database will store articles, for display on the web site.

Field	Purpose	Type
ID	A unique number given to each article, and the primary key of the table.	Integer
Title	The title of the article	Varchar(300)
Tagline	A very short summary of the article	Varchar(600)
Section	The category to which the article belongs	Integer
TheArticle	The article itself	Text

```
CREATE TABLE `databasename`.`cmsarticles` (  
  `ID` int(6) unsigned NOT NULL auto_increment COMMENT 'The unique ID of the article',  
  `title` varchar(200) NULL COMMENT 'The article title',  
  `tagline` varchar(255) NULL COMMENT 'Short summary of the article',  
  `section` int(4) NULL DEFAULT 0 COMMENT 'The section of the article',  
  `thearticle` text NULL COMMENT 'The article itself',  
  PRIMARY KEY (`ID`)  
);
```

Table 1: cmsarticles

Applications Places System Thu Feb 26, 5:28 PM Ishan Mahajan

MySQL Enterprise Help

SELECT * FROM cmsarticles c LIMIT 0,1000

Back Next Execute Step

ID	Title	Tagline	Section	TheArticle
72	DVD release Of first album	Eponymous Debut Album To Be Released As Classic Albums DVD	6	<p style="text-align: justify;">The Doors' Eponym
6	sdfdsq	dgdhfd	35	Your article heredsqgdgdgdfgfg h hftgh hghhhhhh
8	article7	test data	2	Objectives:«¶¶
10	test	aaa	3	ddffffggggggggggg
12	2 February	Test data	3	Lorem ipsum dolor sit amet, consectetur adipis cing
56	t	f	2	<p>This is the initial value.
58	Jim Morrison	James Douglas Morrison	4	<p style="text-align: center;">
39	Feb 13	test data	1	<u> «¶</u>«¶
60	Ray Manzarek	Raymond Daniel Manzarek	4	<p style="text-align: center;">
62	Robby Krieger	Robert Alan Krieger	4	<p style="text-align: center;">
63	Back Door Man	Wha yeah C mon yeah Yeah cmon yeah	5	<p style="text-align: justify;"><span style="font-fa
64	Crystal Ship	Before you slip into unconsciousness	5	<p style="text-align: justify;"><span style="font-fa
65	Hello I love you	Wont you tell me your name	5	<p style="text-align: justify;"><span style="font-fa
66	Love Me Two Times	Love me two time baby	5	<p><
67	Love Street	She lives on Love Street	5	<p><
68	Break On Through	You know the day destroys the night	5	<p><
69	Light my fire	You know that it would be untrue You know that I would be a liar	5	<p style="text-align: justify;"><span style="font-fa
70	The End	This is the end Beautiful friend	5	<p style="text-align: justify;"><span style="font-fa
71	Father of Jim Morrison dies	Father of Late Doors star Jim Morrison dies	6	<p style="text-align: justify;">Late Doors star Jim M
77	First Poem by Jim	First Poem Added To Hall Of Fame	6	<p style="text-align: justify;">Jim Morrison's First P
76	Jim Morrison Alive	Rock legend Jim Morrison may be alive and living	6	<p style="text-align: justify;">Jim Morrison Alive An

25 rows fetched in 0:00.0504 Start Editing Apply Changes First Last Search

Query finished

[Manage Widgets - Mo...] [cms - File Browser] MySQL Query Browser

Table2: sections

Applications Places System Thu Feb 26, 5:29 PM Ishan Mahajan

MySQL Enterprise Help

SELECT * FROM cmssections c LIMIT 0,1000

ID	name	parentid
4	Members	0
6	Blogs	0

3 rows fetched in 0:00.9150

Query finished.

MySQL Query Browser

Table3: cmsgroups

Applications Places System Thu Feb 26, 5:29 PM Ishan Mahajan

MySQL Query Browser - root@localhost via socket

MySQL Enterprise Help

SELECT * FROM cmsgroups c LIMIT 0,1000

ID	groupname
2	Editors
4	NULL
5	NULL
6	NULL
7	NULL
8	NULL
9	NULL
10	Anonymous

9 rows fetched in 0:00.1580

Query finished.

MySQL Query Browser

Table4: cmsusers

Applications Places System Thu Feb 26, 5:29 PM Ishan Mahajan

MySQL Query Browser: root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

SELECT * FROM cmsusers c LIMIT 0,1000

Back Next Execute Stop

ID	user	pass	thegroup	firstname	surname	enabled
2	admin2	admin2	1	Mr	Admin2	1

Schemata Bookmarks

- information_schem
- cmsdb
 - cmsarticles
 - cmsgroups
 - cmssections
 - settings
 - slideshow
- joomla
- mysql
- op

Syntax Functions Params

Data Definition Stat

- ALTER DATABASE
- ALTER TABLE Syn
- CREATE DATABAS
- CREATE INDEX Sy
- DROP DATABASE
- DROP INDEX Syn

2 rows fetched in 0:00.0860

Start Editing Apply Changes First Last Search

Query finished.

[Manage Widgets - Mo... [cms - File Browser] MySQL Query Browser

Table5: slideshow

Applications Places System Thu Feb 26, 5:30 PM Ishan Mahajan

MySQL Query Browser: root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

SELECT * FROM slideshow s LIMIT 0,1000

Back Next Execute Stop

ID	Name	Thumbnail
2	jim2.jpg	tjim2.jpg
3	jim3.jpg	tjim3.jpg
5	jim4.jpg	tjim4.jpg

Schemata Bookmarks

- information_schem
- cmsdb
 - cmsarticles
 - cmsgroups
 - cmssections
 - cmsusers
 - settings
- joomla
- mysql
- op

Syntax Functions Params

Data Definition Stat

- ALTER DATABASE
- ALTER TABLE Syn
- CREATE DATABAS
- CREATE INDEX Sy
- DROP DATABASE
- DROP INDEX Syn

4 rows fetched in 0:00.0139

Start Editing Apply Changes First Last Search

Query finished.

[Manage Widgets - Mo... [cms - File Browser] MySQL Query Browser

Table6: settings

Applications Places System Thu Feb 26, 5:30 PM Ishan Mahajan

MySQL Enterprise Help

SELECT FROM settings s LIMIT 0,1000

Back Next Execute Stop

Name	Value
youtube	1
slideshow	1

Schemata | Bookmarks

- information_schem
- cmsdb
 - cmsarticles
 - cmsgroups
 - cmssections
 - cmsusers
- slideshow
- joomla
- mysql
- op

Syntax | Functions | Params | ID

- Data Definition Stat
 - ALTER DATABASE
 - ALTER TABLE Syn
 - CREATE DATABAS
 - CREATE INDEX Sy
 - DROP DATABASE
 - DROP INDEX Syn

3 rows fetched in 0:00.0910 Start Editing Apply Changes First Last Search

Query finished

(Manage Widgets - Mo... [cms - File Browser] MySQL Query Browser

Table7: cmspages

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

SELECT * FROM cmspages c LIMIT 0,1000

Back Next Execute Step

ID	name
1	aboutus
2	abcd
3	asd

3 rows fetched in 0:00.0220

Start Editing Apply Changes First Last Search

Query finished.

Table8: cmsarticlespage

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools MySQL Enterprise Help

SELECT * FROM cmsarticlespage c LIMIT 0,1000

Back Next Execute Step

ID	Title	Tagline	Page	TheArticle
1	WE	VIII sem	1	<p>We are six guys from VIII semester.</p>
2	Test	test data	2	lorem ipsum.....aaaaaaaaaaaaadddddddggggggggg

2 rows fetched in 0:00.0229

Start Editing Apply Changes First Last Search

Query finished.

CHAPTER 5

FEATURES

Features:

- [-] Home
 - [-] Manage Artices
 - [+] Add Article
 - [+] Remove Article
 - [+] View Articles
 - [-] Manage Pages
 - [+] Add Article to Page
 - [+] Remove Page Article
 - [+] Add Pages
 - [+] Delete Pages
 - [-] Manage Sections
 - [+] Add/Edit Sections
 - [+] View Sections
 - [-] Manage Photo Gallery
 - [+] Add Gallery
 - [+] Remove Gallery
 - [+] View Galleries
 - [-] Manage Site
 - [+] Select a Theme
 - [+] Manage Widgets

[Logout](#)

The Administrative control panel in the form of a tree made using **DOJO**.

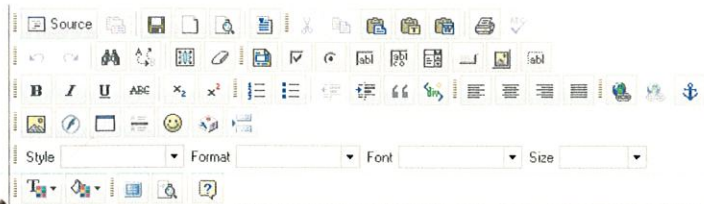
Add article

Add New Article

Title:

Tagline:

Section: **Blogs** ▼



This is the initial value.

Features:

- [-] Home
 - [-] Manage Articles
 - [+] Add Article
 - [+] Remove Article
 - [+] View Articles
 - [-] Manage Pages
 - [+] Add Article to Page
 - [+] Remove Page Article
 - [+] Add Pages
 - [+] Delete Pages
 - [-] Manage Sections
 - [+] Add/Edit Sections
 - [+] View Sections
 - [-] Manage Photo Gallery
 - [+] Add Gallery
 - [+] Remove Gallery
 - [+] View Galleries
 - [-] Manage Site
 - [+] Select a Theme
 - [+] Manage Widgets

[Logout](#)

Removearticle.php

Lyrics

The End - [Delete](#)
Light my fire - [Delete](#)
Break On Through - [Delete](#)
Love Street - [Delete](#)
Love Me Two Times - [Delete](#)
Hello I love you - [Delete](#)
Crystal Ship - [Delete](#)
Back Door Man - [Delete](#)

Members

Robby Krieger - [Delete](#)
John Densmore - [Delete](#)
Ray Manzarek - [Delete](#)
Jim Morrison - [Delete](#)

Blogs

First Poem by Jim - [Delete](#)
Jim Morrison Alive - [Delete](#)
CAFE MORRISON - [Delete](#)
DVD release Of first album - [Delete](#)
Father of Jim Morrison dies - [Delete](#)

Features:

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edit Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galleries
- Manage Site
 - Select a Theme
 - Manage Widgets

[Logout](#)

For the Source Code refer to the Appendix Page No. 48

Viewarticles.php

Lyrics

The End

This is the end Beautiful friend
[View](#)

Light my fire

You know that it would be untrue You know that I would be a liar
[View](#)

Break On Through

You know the day destroys the night
[View](#)

Love Street

She lives on Love Street
[View](#)

Love Me Two Times

Love me two time baby
[View](#)

Hello I love you

Wont you tell me your name
[View](#)

Crystal Ship

Before you slip into unconsciousness
[View](#)

Features

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edit Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galeries
- Manage Site
 - Select a Theme
 - Manage Widgets

[Logout](#)

For the Source Code refer to the Appendix Page No. 49

Editsections.php

Edit Sections

Links - [Home](#)
Members - [Logout](#)
Blogs - [Create](#)

Add a Section

Name:

Parent:

Features:

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edit Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galleries
- Manage Site
 - Select a Theme
 - Manage Widgets

[Logout](#)

For the Source Code refer to the Appendix Page No. 49

Viewsections.php

Lyrics

Members

Blogs

Features:

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edit Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galleries
- Manage Site
 - Select a Theme
 - Manage Widgets

Logout

For the Source Code refer to the Appendix Page No. 51

selectTheme.php

Select Theme

Select Theme ▾
Select Theme
Theme 1
Theme 2
Theme 3

Features

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edt Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galleries
- Manage Site
 - Select a Theme
 - Manage Widgets

Logout

For the Source Code refer to the Appendix Page No. 64

Widgets.php

Manage Widgets

Select Widget
Select Widget
Youtube
Slideshow

Features:

- Home
- Manage Articles
 - Add Article
 - Remove Article
 - View Articles
- Manage Pages
 - Add Article to Page
 - Remove Page Article
 - Add Pages
 - Delete Pages
- Manage Sections
 - Add/Edit Sections
 - View Sections
- Manage Photo Gallery
 - Add Gallery
 - Remove Gallery
 - View Galleries
- Manage Site
 - Select a Theme
 - Manage Widgets

Logout

For the Source Code refer to the Appendix Page No. 67

CHAPTER 6

THE WEB SITES

Web Site 1 : The Doors

HOME DISCOGRAPHY DOWNLOADS ABOUT US REFERENCES

Home

There are things known, and there are things unknown, and in between are...

Introduction

Members

- Robby Krieger
- John Densmore
- Ray Manzarek
- Jim Morrison

Lyrics

- The End
- Light my fire
- Break On Through
- Love Street
- Blogs
- First Poem by Jim

The Doors were an American rock band formed in 1965 in Los Angeles, California by vocalist Jim Morrison, keyboardist Ray Manzarek, drummer John Densmore, and guitarist Robby Krieger. The Doors' music was a fusion of garage rock, blues, and acid rock. They were considered a controversial band, due mostly to Morrison's cryptic lyrics and unpredictable stage persona. After Morrison's death on July 3, 1971, the remaining members continued as a threesome until disbanding in 1973. Despite a career that barely totaled eight years, The Doors still enjoy a huge cult following as well as status in the mainstream music industry as being hugely influential and original. According to the RIAA, they have sold over 32 million albums in the US alone.

Search powered by Google™

the doors jim Morrison

upload your own video

The Doors

Web Site 2 : Shimla

The screenshot shows a website for Shimla with a blue header. The main content area is divided into three columns. The left column contains a search bar and a list of categories. The middle column features an article titled 'the mall' with a photograph of a train. The right column has a 'recent posts' section with links to 'Kalka Simla Rail' and 'HP University', and a 'slideshow' section with a black placeholder image.

shimla queen of hills

search

Search

categories

History

- [Kalka Simla Rail](#)

Destinations

- [Chail](#)
- [Kufri](#)

Institutions

- [St Bedes College](#)
- [HP University](#)
- [Indian Institute of Advanced Study](#)

the mall

Mall Road is the main street in Shimla, the capital city of Himachal Pradesh, India. Constructed during the British colonial rule, the Mall road is located a level below the ridge. The offices of municipal corporation, fire service, and police headquarters are located here. Automobiles, except emergency vehicles are not allowed on this road. For this reason, the locals – and tourists – are often found strolling on this road in the evenings. Mall road has a number of showrooms, department stores, shops, restaurants and cafes. A Himachal emporium that offers handicraft products of Himachal Pradesh like locally designed woollen cloths, branded cloths, pottery items, and jewellery is also located here. There are at least three books shops including one that sells old books.

recent posts

- Kalka Simla Rail**
constructed in 1906
[St Bedes College](#)
College founded by the Religious of Jesus and Mary in 1904
- HP University**
Himachal Pradesh University

slideshow

CHAPTER 7

INSTALLATION

Requirement – Apache , MySQL, PHP, Web Browser

For Windows – Put the CMS folder in *htdocs* folder and the use of an integrated environment like XAMPP is recommended.

For Linux – Put the CMS folder in */var/www* folder and the use of an integrated environment like LAMP is recommended.

To run the program :

Put the database username and password in *cms/includes/SystemComponent.php*

Put the address <http://localhost/cms> in the web browser , preferably Mozilla Firefox

FUTURE SCOPE

The foremost work in the near future would be to roll out the project towards the intended audience. The application can serve a justifiable tool for the communication purposes. The reach of the application would also be expanded. We ,therefore would aim at presenting the application out to various organizations whereby they could also benefit from its features.

The roll-out process would be carefully analyzed.i.e all the working details-member updates, message updates, search box will be regularly monitored for errors. In addition to the roll out process ,the addition of new widgets in the application and other entertainment links would also be taken under consideration

The other future plan includes imbibing various social networking features like embedding of the chat option to its users and various other dynamic options which is the demand of today. The security of the application would also be a feature which will be worked on. Also , we would be more than happy to present this project to the respected Open Source Community wherein by their help the application would only expand .

So remaining humble as ever, we would like this project to only grow-small it may be, but we will not hesitate in seeking the precious opinion of any person related directly or indirectly to the project.

CONCLUSION

With the completion of the project we can boastfully acknowledge the systematic approach followed by the software industry. The project after the completion have not only made us familiar with project culture followed by the industry but has also taught us the importance of practicality. Our knowledge about all the software technologies would only be fallacious without us implementing them on a practical scenario-This is what "Content Management System" development has taught us over the entire year or so.

Our team also now acknowledges the beauty of Team Work. All the group members have reverently extracted the amazing ecstasy of working as a whole. More importantly this project has not only increased our intellectual knowledge but has taught us more important aspects of life i.e humbleness, calmness and introspection. The project, though a small step in this infinite learning curve has helped us in acquainting the general aspects of Project Development as a whole which will definitely help us in our future arena.

In the end we would again like to take the honourable privilege to reverently thank our project guide Mr Sanjeev Patel without whom the project would have remained merely a name. Our thanks also to the other teaching staff, the technical and non technical staff of our institution whose help had only moved the project ahead.

With the development of this we have only moved further in our quest for knowledge.

BIBLIOGRAPHY:

Websites:-

www.intranetjournal.com

www.w3schools.com

www.dojotoolkit.org

www.youtube.com/api

www.wikipedia.org

Books:-

- PHP and MySQL Bible by Tim Converse, Joyce Parks, Clark Morgon
- Learning PHP and MySQL O'REILLY
- PHP and MySQL by Janed Valade

APPENDIX

CODE FOR VARIOUS PAGES

SystemComponent Class

```
<?php
class SystemComponent {

var $settings;

function getSettings() {

// System variables
$settings['siteDir'] = '/path/to/your/intranet/';

// Database variables
$settings['dbhost'] = 'hostname';
$settings['dbusername'] = 'dbuser';
$settings['dbpassword'] = 'dbpass';
$settings['dbname'] = 'mydb';

return $settings;

}

}
?>
```


DbConnector Class

```
<?php
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Class: DbConnector
// Purpose: Connect to a database, MySQL version
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
require_once 'SystemComponent.php';

class DbConnector extends SystemComponent {

var $theQuery;
var $link;

/** Function: DbConnector, Purpose: Connect to the database **
function DbConnector(){

// Load settings from parent class
$settings = SystemComponent::getSettings();

// Get the main settings from the array we just loaded
$host = $settings['dbhost'];
$db = $settings['dbname'];
$user = $settings['dbusername'];
$pass = $settings['dbpassword'];

// Connect to the database
$this->link = mysql_connect($host, $user, $pass);
mysql_select_db($db);
register_shutdown_function(array(&$this, 'close'));

}/** Function: query, Purpose: Execute a database query **
function query($query) {

$this->theQuery = $query;
return mysql_query($query, $this->link);
```

```
}/** Function: fetchArray, Purpose: Get array of query results ***  
function fetchArray($result) {
```

```
return mysql_fetch_array($result);
```

```
}/** Function: close, Purpose: Close the connection ***  
function close() {
```

```
mysql_close($this->link);
```

```
}
```

```
}
```

```
?>
```

Common-head.php

```
<?php
```

```
// Create an object (instance) of the DbConnector
```

```
$connector = new DbConnector();
```

```
        $result = $connector->query('SELECT Name, Value FROM settings WHERE  
Name="theme" ');
```

```
while ($row = $connector->fetchArray($result)){
```

```
    if ($row['Value'] == 1 )
```

```
        echo '<link href="'. $applicationPath .'/css/common.css" rel="stylesheet" type="text/css"
```

```
>';
```

```
    else if ($row['Value'] == 2 )
```

```
        echo '<link href="'. $applicationPath .'/css/theme2.css" rel="stylesheet" type="text/css" />';
```

```
    else if ($row['Value'] == 3 )
```

```
        echo '<link href="'. $applicationPath .'/css/jimred.css" rel="stylesheet" type="text/css" />';
```

```
    else if ($row['Value'] == 4 )
```

```
        echo '<link href="'. $applicationPath .'/css/jimgreen.css" rel="stylesheet" type="text/css"
```

```
>';
```

```
}
```

```
?>
```

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

```
@import "<?php echo $applicationPath ?>/dojo/dojo/resources/dojo.css";
```

```
@import "<?php echo $applicationPath ?>/dojo/dijit/themes/tundra/tundra.css";
```

```
@import "<?php echo $applicationPath ?>/common/style.css";
```

```
</style>
```

```
<script type="text/javascript" src="<?php echo $applicationPath ?>/dojo/dojo/dojo.js"
    djConfig="parseOnLoad: true"></script>
```

```
<script>
    dojo.require("dojo.data.ItemFileReadStore");
    dojo.require("dijit.Tree");
    dojo.require("dojo.parser");
</script>
```

Index.php

```
<?php
    require_once('includes/DbConnector.php')

$connector = new DbConnector();

    $result = $connector->query("SELECT Name,Value FROM settings WHERE
Name="theme" ");

while ($row = $connector->fetchArray($result)){

    if ($row['Value'] == 1 )

        include 'templates/Themeblue/home.php';

    else if ($row['Value'] == 2 )

        include 'templates/compressed/home.php';

    else if ($row['Value'] == 3 )

        include 'templates/unlink/home.php';

}

?>
```

Pageindex.php

```
<?php
    require_once('includes/DbConnector.php');

$connector = new DbConnector();

    $result = $connector->query('SELECT Name,Value FROM settings WHERE
Name="theme" ');

while ($row = $connector->fetchArray($result)){
    if ($row['Value'] == 1 )
        include 'templates/Themeblue/page.php';
    else if ($row['Value'] == 2 )
        include 'templates/compressed/page.php';
    else if ($row['Value'] == 3 )
        include 'templates/unlink/page.php';
}
?>
```

Addarticle.php

```
<?php
require_once '../common/globals.php';
$page_title = 'Add Article';
require_once('../includes/DbConnector.php');
?>

<html>
<head>
<title>Add an Article</title>
```

```

<link href="<?php echo $applicationPath ?>/css/common.css"
      rel="stylesheet" type="text/css" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script type="text/javascript" src="fckeditor/fckeditor.js"></script>
<script type="text/javascript">
window.onload = function()
{
var oFCKeditor = new FCKeditor( 'thearticle' ) ;
oFCKeditor.BasePath = "fckeditor/" ;
oFCKeditor.ReplaceTextarea() ;
}
</script>

<style>
table.articleTitle {
    margin-bottom: 1em;
}

table.articleTitle td {
    padding: .5em 3em .5em 0;
}
</style>

<?php require_once('../common/common-head.php'); ?>

</head>
<?php
// Get the PHP file containing the DbConnector class
require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');

// Create an instance of DbConnector
$connector = new DbConnector();

// Check whether a form has been submitted. If so, carry on
if ($HTTP_POST_VARS){

    // Validate the entries
    $validator = new Validator();

    $validator->validateTextOnly($HTTP_POST_VARS['title'],'Article Title');
    $validator->validateTextOnly($HTTP_POST_VARS['tagline'],'Tagline');
    $validator->validateNumber($HTTP_POST_VARS['section'],'Section');
    $validator->validateGeneral($HTTP_POST_VARS['thearticle'],'Article');
}
}

```

```

// Check whether the validator found any problems
if ( $validator->foundErrors() ){
    echo 'There was a problem with: <br>'. $validator->listErrors('<br>'); // Show the
errors, with a line between each
}else{

    // Create an SQL query (MySQL version)
    // The 'addslashes' command is used 5 lines below for added security
    // Remember to use 'stripslashes' later to remove them (they are inserted in front
of any
    // special characters

    $insertQuery = "INSERT INTO cmsarticles (title,tagline,section,thearticle)
VALUES (".
"".$HTTP_POST_VARS['title'].", ".
"".$HTTP_POST_VARS['tagline'].", ".
    $HTTP_POST_VARS['section'].", ".
"".$HTTP_POST_VARS['thearticle'].")";

    // Save the form data into the database
    if ($result = $connector->query($insertQuery)){

        // It worked, give confirmation
        echo '<center><b>Article added to the database</b></center><br>';

    }else{

        // It hasn't worked so stop. Better error handling code would be good here!
        exit('<center>Sorry, there was an error saving to the database</center>');

    }
}
?>

<body>

<?php
include '../common/header.php';
?>

<div id="main" class="container">

<div id="left-center">

```

```

<form name="form1" method="post" action="addarticle.php">

<h2>Add New Article</h2>

<table class="articleTitle">
  <tr>
    <td>Title:</td>
    <td><input name="title" type="text" id="title" /></td>
  </tr>
  <tr>
    <td>Tagline:</td>
    <td><input name="tagline" type="text" id="tagline" /></td>
  </tr>
  <tr>
    <td>Section:</td>
    <td><select name="section" id="section">
      <?PHP // Generate a drop-down list of sections.
      // NOTE : Requires database modifications in article 4

      $result = $connector->query('SELECT ID,name FROM cmssections ORDER BY
name');

      // Get an array containing the results.
      // Loop for each item in that array
      while ($row = $connector->fetchArray($result)){
        echo '<option value="'. $row['ID']. "'>'. $row['name']. ' </option>;
      }
      ?>
    </select></td>
  </tr>
</table>

<textarea id="thearticle" name="thearticle">This is <b>the</b> initial value.</textarea>

<p align="center"><input type="submit" name="Submit" value="Submit"></p>
</form>
</div>

  <?php
  include './common/tree.php';
  ?></div>

</body>

```



```
</html>
```

```
Removearticle.php
```

```
require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');
// Create an object (instance) of the DbConnector
$connector = new DbConnector();
$validator = new Validator();

if ($HTTP_GET_VARS['action'] == 'delete'){
    // Store the article ID to be deleted in a variable
    $articleID = $HTTP_GET_VARS['id'];
    // Validate the sectionID, and if it's ok delete the section
    if ( $validator->validateNumber($articleID,'Section ID') ){
        // The validator returned true, so go ahead and delete the section
        $connector->query('DELETE FROM cmsarticles WHERE ID = '.$articleID);
        echo 'Article Deleted.<br>';
    }else{
        // The validator returned false, meaning there was a problem
        echo "Couldn't delete. There was a problem with: ".$validator->listErrors();
    }
}
$result = $connector->query('SELECT ID,name FROM cmssections');
while ($row = $connector->fetchArray($result)){
    echo '<h2>'.$row['name'].'</h2>';
    echo '<br>';
    // Execute the query to retrieve articles
    $result2 = $connector->query('SELECT ID,Title,Section FROM cmsarticles WHERE
Section='.$row['ID'].' ORDER BY ID DESC');
    // Get an array containing the results.
    // Loop for each item in that array
    while ($row = $connector->fetchArray($result2)){
        echo $row['Title'].' - &nbsp;&nbsp;&nbsp;'; // Show the title of article
        echo '<a href="removearticle.php?action=delete&id='.$row['ID'].'"> Delete </a>';
    }
}
// Show the delete link
echo '<br>'; // Show a carriage return
}
```

```
Viewarticles.php
```

```

<?php
require_once('../includes/DbConnector.php');
$connector = new DbConnector();
$result = $connector->query('SELECT ID,name FROM cmssections');
while ($row = $connector->fetchArray($result)){
    echo '<h2>'.$row['name'].'</h2>';
    echo '<br>';
    // Execute the query to retrieve articles
    $result2 = $connector->query('SELECT ID,Title,Tagline,Section FROM cmsarticles WHERE
    Section='.$row['ID'].' ORDER BY ID DESC');
    // Get an array containing the results.
    // Loop for each item in that array
    while ($row = $connector->fetchArray($result2)){

        echo '<h3>'.$row['Title'].'</h3> &nbsp;&nbsp;&nbsp;<br>'; // Show the title of article
        echo $row['Tagline'].'<br>'; // Show the title of article

        echo '<a href="..index.php?id='.$row['ID'].'">View</a>'; // Show the delete link
        echo '<br><br>'; // Show a carriage return
    }
}??>

```

Editsections.php

```

<?php
require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');
$connector = new DbConnector();
$validator = new Validator();
// DELETE SECTIONS //////////////////////////////////////
if ($HTTP_GET_VARS['action'] == 'delete'){
    // Store the section ID to be deleted in a variable
    $sectionID = $HTTP_GET_VARS['id'];
    // Validate the sectionID, and if it's ok delete the section
    if ( $validator->validateNumber($sectionID,'Section ID') ){
        // The validator returned true, so go ahead and delete the section
        $connector->query('DELETE FROM cmssections WHERE ID = '.$sectionID);
        echo 'Section Deleted.<br>';
    }else{
        // The validator returned false, meaning there was a problem
        echo "Couldn't delete. There was a problem with: ".$validator->listErrors();}
}
// ADD SECTION //////////////////////////////////////

```


Viewsections.php

```
require_once('../includes/DbConnector.php');
// Create an object (instance) of the DbConnector
$connector = new DbConnector();
$result = $connector->query('SELECT ID,name FROM cmssections');

while ($row = $connector->fetchArray($result)){

    echo '<h2>'.$row['name'].'</h2>';
    echo '<br>';
}
```

Addgallery.php

```
<p>Add Gallery</p>
<form enctype="multipart/form-data" action="addgallery.php" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="10000000000" />
Choose a file to upload: <input name="uploadedfile" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
```

```
<?php
```

```
$target_path = "../common/photos/";
```

```
$target_path = $target_path . basename( $_FILES['uploadedfile']['name']);
if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
    echo "The file ". basename( $_FILES['uploadedfile']['name']).
    " has been uploaded";
} else{
    echo "There was an error uploading the file, please try again!";
}??
```

addpage.php

```
<?php
require_once '../common/globals.php';
$page_title = 'Sections';
require_once('../includes/DbConnector.php');
?>

<html>

<head>

<title>Add Page</title>

<link href="<?php echo $applicationPath ?>/css/common.css"
      rel="stylesheet" type="text/css" />

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<?php require_once('../common/common-head.php'); ?>

<style>

table.articleTitle {
    margin-bottom: 1em;
}

table.articleTitle td {
    padding: .5em 3em .5em 0;
}

</style>

</head>

<body>

<div id="main" class="container">

<div id="left-center">
```

```

<form name="form1" method="post" action="addpage.php?action=add"><?php
// Require the classes

require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');

// Create an object (instance) of the DbConnector and Validator

$connector = new DbConnector();
$validator = new Validator();

// ADD SECTION ////////////////////////////////////////

if ($HTTP_GET_VARS['action'] == 'add'){

    $validator->validateTextOnlyNoSpaces($HTTP_POST_VARS['name'],'section name');

    if (!$validator->foundErrors()){

        $connector->query('INSERT INTO cmspages (name) VALUES
        ("'.$HTTP_POST_VARS['name'].'");

    }else{

        echo '<b>Please correct '.$validator->listErrors().</b><br><br>';

    }

}

// LIST SECTIONS ////////////////////////////////////////

?><table class="articleTitle">

    <h2>Add a Page</h2>

    <tr>

        <td>Name:</td>

```

```
<td><input name="name" type="text" id="name"></td>
</tr>
<tr>
<td><input type="submit" name="Submit" value="Create"></td>
</tr>
</table>
</form>
</div>
<?php
include '../common/tree.php';
?></div>
</body>
</html>
```

addtopage.php

```
<?php
require_once '../common/globals.php';
$page_title = 'Add Article';
require_once('../includes/DbConnector.php');
?>
<html>
<head>
<title>Add an Article</title>
<link href="<?php echo $applicationPath ?>/css/common.css"
```



```

        rel="stylesheet" type="text/css" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script type="text/javascript" src="fckeditor/fckeditor.js"></script>
<script type="text/javascript">
window.onload = function()
{
var oFCKEditor = new FCKEditor( 'thearticle' );
oFCKEditor.BasePath = "fckeditor/" ;
oFCKEditor.ReplaceTextarea() ;
}
</script>
<style>
table.articleTitle {
    margin-bottom: 1em;
}
table.articleTitle td {
    padding: .5em 3em .5em 0;
}
</style>
<?php require_once('../common/common-head.php'); ?>
</head>
<?php
// Get the PHP file containing the DbConnector class
require_once('../includes/DbConnector.php');

```

```

require_once('./includes/Validator.php');

// Create an instance of DbConnector

$connector = new DbConnector();

// Check whether a form has been submitted. If so, carry on

if ($HTTP_POST_VARS){

    // Validate the entries

    $validator = new Validator();

    $validator->validateTextOnly($HTTP_POST_VARS['title'],'Article Title');

    $validator->validateTextOnly($HTTP_POST_VARS['tagline'],'Tagline');

    $validator->validateNumber($HTTP_POST_VARS['section'],'Section');

    $validator->validateGeneral($HTTP_POST_VARS['thearticle'],'Article');

// Check whether the validator found any problems

    if ( $validator->foundErrors() ){

        echo 'There was a problem with: <br>'. $validator->listErrors('<br>'); // Show the
errors, with a line between each

    }else{

        // Create an SQL query (MySQL version)

        // The 'addslashes' command is used 5 lines below for added security

        // Remember to use 'stripslashes' later to remove them (they are inserted in front
of any

        // special characters

        $insertQuery = "INSERT INTO cmsarticlespage (Title,Tagline,Page,TheArticle)
VALUES ("

        "".$HTTP_POST_VARS['title'].", ".

        "".$HTTP_POST_VARS['tagline'].", ".

        $HTTP_POST_VARS['section'].", ".

```

```

""".addslashes($_HTTP_POST_VARS['thearticle']).""");
    // Save the form data into the database
    if ($result = $connector->query($insertQuery)){
// It worked, give confirmation
        echo '<center><b>Article added to the database</b></center><br>';
    }else{

        // It hasn't worked so stop. Better error handling code would be good here!
        exit('<center>Sorry, there was an error saving to the database</center>');
    }
}
}
?>
<body>
<div id="main" class="container">
<div id="left-center">
<form name="form1" method="post" action="addtopage.php">
<h2>Add New Article</h2>
<table class="articleTitle">
    <tr>
        <td>Title:</td>
        <td><input name="title" type="text" id="title" /></td>
    </tr>
    <tr>

```

```

        <td>Tagline:</td>

        <td><input name="tagline" type="text" id="tagline" /></td>

    </tr>

    <tr>

        <td>Page:</td>

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

            <?PHP          // Generate a drop-down list of sections.

            // NOTE : Requires database modifications in article 4

            $result = $connector->query('SELECT ID,name FROM cmspages ORDER BY name');// Get an
            array containing the results.

            // Loop for each item in that array

            while ($row = $connector->fetchArray($result)){

                echo '<option value="'. $row['ID']. "'>' . $row['name']. '</option>';

            }

            ?>

        </select></td>

    </tr>

</table>

<textarea id="thearticle" name="thearticle">This is <b>the</b> initial value.</textarea>

<p align="center"><input type="submit" name="Submit" value="Submit"></p>

</form>

</div>

<?php

    include '../common/tree.php';

?></div>

```



```
</body>
```

```
</html>
```

```
removepage.php
```

```
<?php
```

```
    require_once '../common/globals.php';
```

```
    $page_title = 'Sections';
```

```
    require_once('../includes/DbConnector.php');
```

```
    ?>
```

```
<html>
```

```
<head>
```

```
<title>Edit Sections</title>
```

```
<link href="<?php echo $applicationPath ?>/css/common.css"
```

```
    rel="stylesheet" type="text/css" />
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<?php require_once('../common/common-head.php'); ?>
```

```
<style>
```

```
table.articleTitle{
```

```
    margin-bottom: 1em;
```

```
}
```

```
table.articleTitle td{
```

```
    padding:.5em 3em .5em 0;
```

```
}
```

```
</style>
```

```

</head>
<body>
<div id="main" class="container">
<div id="left-center">
<form name="form1" method="post" action="removepage.php?action=delete">
<h2>Remove Page</h2>
<?php
// Require the classes
require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');
// Create an object (instance) of the DbConnector and Validator
$connector = new DbConnector();
$validator = new Validator();

// DELETE SECTIONS //////////////////////////////////////
if ($HTTP_GET_VARS['action'] == 'delete'){
    // Store the section ID to be deleted in a variable
    $pageID = $HTTP_GET_VARS['id'];
    // Validate the sectionID, and if it's ok delete the section
    // The validator returned true, so go ahead and delete the section
    $connector->query('DELETE FROM cmspages WHERE ID = '.$pageID);
    echo 'Page Deleted.<br>';
}

// LIST SECTIONS //////////////////////////////////////

```



```

?>
<html>
<head>
<title>View Articles</title>
<link href="<?php echo $applicationPath ?>/css/common.css"
        rel="stylesheet" type="text/css" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<?php require_once('../common/common-head.php'); ?>
</head>
<body>
<div id="main" class="container">
<div id="left-center"><?php
// Require the database class
require_once('../includes/DbConnector.php');
require_once('../includes/Validator.php');
// Create an object (instance) of the DbConnector
$connector = new DbConnector();
$validator = new Validator();

if ($HTTP_GET_VARS['action'] == 'delete'){
    // Store the article ID to be deleted in a variable
    $articleID = $HTTP_GET_VARS['id'];
    // Validate the sectionID, and if it's ok delete the section
    if ( $validator->validateNumber($articleID,'Section ID') ){

```



```

        // The validator returned true, so go ahead and delete the section
        $connector->query('DELETE FROM cmsarticlespage WHERE ID = '.$articleID);
        echo 'Article Deleted.<br>';
    }else{
        // The validator returned false, meaning there was a problem
        echo "Couldn't delete. There was a problem with: ".$validator->listErrors();
    }
}
$result = $connector->query('SELECT ID,name FROM cmspages');
while ($row = $connector->fetchArray($result)){
    echo '<h2>'.$row['name'].'</h2>';
    echo '<br>';
    // Execute the query to retrieve articles
    $result2 = $connector->query('SELECT ID,Title,Page FROM cmsarticlespage WHERE
Page='.$row['ID'].' ORDER BY ID DESC');
    // Get an array containing the results.
    // Loop for each item in that array
    while ($row = $connector->fetchArray($result2)){
        echo $row['Title'].' - &nbsp;&nbsp;&nbsp;'; // Show the title of article
        echo '<a href="removepagearticle.php?action=delete&id='.$row['ID'].'"> Delete
</a>'; // Show the delete link
        echo '<br>'; // Show a carriage return
    }
}
?></div>

```

```
<?php
include '../common/tree.php';
?></div>
</body></html>
```

selectTheme.php

```
<?php •
    require_once '../common/globals.php';
    $page_title = 'Select Theme';
    require_once('../includes/DbConnector.php');
    ?>
<html>
<head>
<title>Select Theme</title>
<link href="<?php echo $applicationPath ?>/css/common.css"
    rel="stylesheet" type="text/css" />
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<?php require_once('../common/common-head.php'); ?>
</head>
<body>
<div id="main" class="container">
<div id="left-center">
<p>Select Theme</p>
```

```

<form name="form1" method="post" action="selectTheme.php">
<select name="theme" id="theme">
    // Generate a drop-down list of sections.
    // NOTE : Requires database modifications in
article 4
    // Get an array containing the results.
    // Loop for each item in that array
        <option > Select Theme </option>;
        <option value="1" id="theme1" > Theme 1
</option>;
        <option value="2" id="theme2"> Theme 2
</option>;
        <option value="3" id="theme3"> Theme 3
</option>;

    </select>

    <br><br>
<input type="submit" name="Submit" value="Apply">
</form>
<?php
// Get the PHP file containing the DbConnector class
require_once('../includes/Validator.php');
// Create an instance of DbConnector
$connector = new DbConnector();
// Check whether a form has been submitted. If so, carry on

```



```

if($HTTP_POST_VARS){
// Validate the entries

$validator = new Validator();
$validator->validateNumber($HTTP_POST_VARS['theme'],'Theme');

// Check whether the validator found any problems

if ( $validator->foundErrors() ){
    echo 'There was a problem with: <br>'.$validator->listErrors('<br>'); // Show the errors,
with a line between each
}else{
// Create an SQL query (MySQL version)
// The 'addslashes' command is used 5 lines below for added security
// Remember to use 'stripslashes' later to remove them (they are inserted in front of any
// special characters

// Save the form data into the database

if ($result = $connector->query("UPDATE settings SET
Value='.$HTTP_POST_VARS['theme'].' WHERE Name="theme"")){

// It worked, give confirmation

echo '<center><b>Theme changed</b></center><br>';

}else{

// It hasn't worked so stop. Better error handling code would be good here!

exit('<center>Sorry, there was an error saving to the database</center>');

}

}

}??>

</div>

```

```

<?php
include '../common/tree.php';
?>
</div>
</body>
</html>

```

Widgets.php

```

<div id="left-center">
<p>Manage Widgets</p>
<form name="form1" method="post" action="widgets.php"><select
name="widgets" id="widgets">
<option>Select Widget</option>
';
<option value="Youtube" id="theme1">Youtube</option>
';
<option value="Slideshow" id="theme2">Slideshow</option>
';
</select> <br>
<br>
<br>
<input type="submit" name="Submit" value="Enable/Disable"></form>

<?php
// Get the PHP file containing the DbConnector class
require_once('../includes/Validator.php');
// Create an instance of DbConnector
$connector = new DbConnector();
// Check whether a form has been submitted. If so, carry on
if ($_HTTP_POST_VARS){
    $connector = new DbConnector();
    // Execute the query to retrieve articles
    $result = $connector->query("SELECT Value FROM settings WHERE
Name='".$_HTTP_POST_VARS["widgets"]."'");

```

```
while ($row = $connector->fetchArray($result)){
    if ($row['Value'] == 1 )
    {
        $connector->query("UPDATE settings SET Value=0 WHERE
Name='".$SHTTP_POST_VARS["widgets"]." ' ");
        echo '<center><b>Widget Disabled</b></center><br>';
    }
    else{
        $connector->query("UPDATE settings SET Value=1 WHERE
Name='".$SHTTP_POST_VARS["widgets"]." ' ");
        echo '<center><b>Widget Enabled</b></center><br>';
    }
}
}
?>
```


Theme:

Home.php

```
<?php
    require_once('common/globals.php');

    $page_title = 'Home';

    require_once('includes/DbConnector.php');

    ?>

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

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>The Doors</title>

<link rel="stylesheet" type="text/css" href="templates/Themeblue/thcommon/common.css" />

    <?php require_once('common/common-head.php'); ?>

</head>

<body>

<?php
include 'templates/Themeblue/thcommon/header.php';

?>

<!-- Start Main Content -->

<div id="main" class="container"><!-- left column (products and features) -->

<div id="leftcolumn" ><?php

// Require the database class
```

```

//require_once('includes/DbConnector.php');

// Create an object (instance) of the DbConnector

$connector = new DbConnector();

// Execute the query to retrieve articles

$result = $connector->query('SELECT ID,name FROM cmssections ORDER BY ID');

// Get an array containing the results.

// Loop for each item in that array

while ($row = $connector->fetchArray($result)){

    echo '<h3 class="leftbox"> ';

    echo $row['name'];

    echo '</h3>';

    $result2 = $connector->query(' SELECT ID,title FROM cmsarticles WHERE
Section='.$row['ID'].' ORDER BY ID DESC LIMIT 0,4');

    while ($row2 = $connector->fetchArray($result2)){

        echo '<ul class="leftbox borderedlist"><li> <a href="?id='.$row2['ID'].'">';

        echo $row2['title'];

        echo '</a></li> </ul>';

    }

}

?>

<hr />

</div>

<!-- main content area -->

<div id="center">

<div class="article_wrapper">

```



```

<?php
// IMPORTANT!!! Validate the ID number. See below
// Create an object (instance) of the DbConnector
$connector = new DbConnector();
$id = $_GET['id'];
if (isset($id)){
    // Execute the query to retrieve the selected article
    $result = $connector->query('SELECT title,thearticle FROM cmsarticles WHERE ID =
'. $id);
}
else {
    $result = $connector->query(' SELECT title,thearticle FROM cmsarticles WHERE
Section = 50');
}
// Get an array containing the resulting record
$row = $connector->fetchArray($result);
?>
<h2><?php echo $row['title'];?></h2>
<?php echo stripslashes($row['thearticle']);?>
</div>
<hr />
</div>
<!-- product sales boxes -->
<div id="rightcolumn">
    <div class="rightbox_wrapper">

```

```
<div class="rightbox">
```

```
<?php
```

```
$connector = new DbConnector();
```

```
// Execute the query to retrieve articles
```

```
$result = $connector->query("SELECT Name,Value FROM settings WHERE Name='youtube'");
```

```
while ($row = $connector->fetchArray($result)){
```

```
if ($row['Value'] == 1 )
```

```
{
```

```
include 'common/youtube.php';
```

```
}
```

```
}
```

```
?>
```

```
class="product_image" />
```

```
<!-- 
```

```
<h4>Product Sales</h4>
```

```
<p>Lorem ipsum dolor sit amet, conseqing elit, sed  
diam nonummy nibh dunt ut laoreet dolore magna aliquat. Ut wisi enim ad minim veniam,  
exerci tation ullamcorper s... <a href="#" title="Read More">more</a></p>
```

```
</div>-->
```

```
</div>
```

```
</div>
```

```
<div class="rightbox_wrapper">
```

```
<div class="rightbox">
```

```

        <?php
$connector = new DbConnector();
// Execute the query to retrieve articles
$result = $connector->query("SELECT Name,Value FROM settings WHERE Name='slideshow'
");while ($row = $connector->fetchArray($result)){
if ($row['Value'] == 1 )
{
include 'common/slideshow2.php';
}
}
?>

```

```

        </div>
</div>
<div class="rightbox_wrapper lastbox">
    <div class="rightbox">
        
        <div class="product_wrapper">
            <h4>Product Sales</h4>
            <p>Lorem ipsum dolor sit amet, conseqing elit, sed
diam nonummy nibh dunt ut laoreet dolore magna aliquat. Ut wisi enim ad minim veniam,
exerci tation ullamcorper s... <a href="#" title="Read More">more</a></p>
        </div>
    </div>
</div>
</div>

```

```
        <hr />
    </div>
</div>
</div>
<?php
include 'templates/Themeblue/thcommon/footer.php';
?>
</body>
</html>
```

Page.php

```
<?php
require_once('common/globals.php');
$page_title = 'Home';
require_once('includes/DbConnector.php');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>The Doors</title>
```

```

<link rel="stylesheet" type="text/css"
    href="templates/Themeblue/thcommon/common.css" />
<?php require_once('common/common-head.php'); ?>
</head>

<body>

<?php
include 'templates/Themeblue/thcommon/header.php';
?>

<!-- Start Main Content -->

<div id="main" class="container"><!-- left column (products and features) -->
<div id="leftcolumn">
<hr />
</div>

<!-- main content area -->
<div id="center">
<div class="article_wrapper"><?php
// IMPORTANT!!! Validate the ID number. See below
// Create an object (instance) of the DbConnector
$connector = new DbConnector();
$id = $_GET['id'];
if (isset($id)){
    // Execute the query to retrieve the selected article
    $result = $connector->query('SELECT title,thearticle FROM cmsarticlespage WHERE
Page = '.$id);

```

```

}
else {
    $result = $connector->query(' SELECT title,thearticle FROM cmsarticles WHERE
Section = 50');
}

// Get an array containing the resulting record
while ($row = $connector->fetchArray($result)){
    ?>
    <h2><?php echo $row['title'];?></h2>
        <?php echo stripslashes($row['thearticle']);?></div>
    <hr />
</div>
<!-- product sales boxes -->
<div id="rightcolumn">
    <div class="rightbox_wrapper">
        <div class="rightbox"><?php
$connector = new DbConnector();
// Execute the query to retrieve articles
$result = $connector->query("SELECT Name,Value FROM settings WHERE Name='youtube'
");
while ($row = $connector->fetchArray($result)){
    if ($row['Value'] == 1 )
    {
        include 'common/youtube.php';

```

```

    }
}
?></div>

</div>

<div class="rightbox_wrapper">

<div class="rightbox"><?php

$connector = new DbConnector();

// Execute the query to retrieve articles

$result = $connector->query("SELECT Name,Value FROM settings WHERE Name='slideshow'
");

while ($row = $connector->fetchArray($result)){

    if ($row['Value'] == 1 )

    {

        include 'common/slideshow2.php';

    }

}

?></div>

</div>

<div class="rightbox_wrapper lastbox">

<div class="rightbox">

<div class="product_wrapper">

<h4>Product Sales</h4>

<p>Lorem ipsum dolor sit amet, conseqing elit, sed diam nonummy nibh
dunt ut laoreet dolore magna aliqupat. Ut wisi enim ad minim veniam,

```

exerci tation ullamcorper s... more</p>

</div>

</div>

</div>

<hr />

</div>

</div>

</div>

<?php

include 'templates/Themeblue/thcommon/footer.php';

?>

</body>

</html>