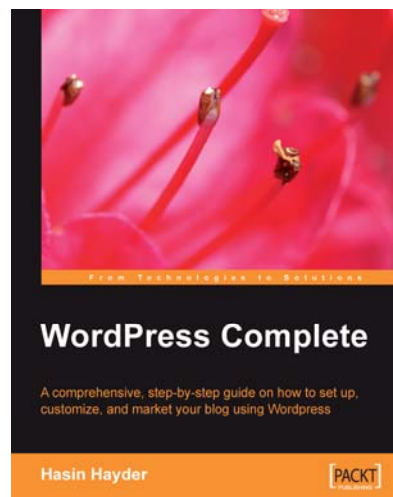




WordPress Complete

A comprehensive, step-by-step guide on how to set up, customize, and market your blog using WordPress

Hasin Hayder



Chapter 5 "Non-Blog Content"

In this package, you will find:

- A Biography of the authors of the book
- A preview chapter from the book, Chapter 5 "Non-Blog Content"
- A synopsis of the book's content
- Information on where to buy this book

About the Author

Hasin Hayder

Hasin Hayder graduated in Civil Engineering from the Rajshahi University of Engineering and Technology (RUET) in Bangladesh. He is a Zend-certified Engineer and an expert in localization. Hasin is currently working in one of the leading Ajax startpage company Pageflakes Ltd (www.pageflakes.com) as a Development Engineer. He is also working in Somewhere In (www.somewherein.net) as a Web Application Developer. Hasin is an expert WordPress user and has developed several themes and plug-ins for the community. He also maintains the WordPress4SQLite project, which is an unofficial port of WordPress to be used with SQLite. You can reach Hasin at hasin@somewherein.net as well as at hasin@pageflakes.com. You can also visit Hasin's personal blog at <http://hasin.wordpress.com> when you are free.

First of all, I would like to thank David Barnes, Patricia Weir, Abhijeet Deobhakta, Nanda Padmanabhan, Nikhil Bangera, and Priyanka Baruah, without whom the book would have never seen the daylight. I would also like to thank Mohan Rapheal and Jimmy Karumalil for their efforts. After writing for so many sleepless nights, the book is finally over; but I want to thank all those who supported me at that time. My wife Ayesha for storytelling, Little Afif for missing his Papa, the staff of Somewhere In for receiving a sleepy developer in the morning, the staff of Pageflakes for their inspiration, and all the members of my PHP group, phpexperts. I would also like to thank all my family members for their great support during this period. Finally, I dedicate this book to the person who would have been the happiest person to see it; my father, Ali Akbar Mohammad Mohiuddin Hayder (1934-2006).

For More Information: <http://www.packtpub.com/wordpress/book>

About the Reviewers

Nikhil Bangera

Nikhil Bangera graduated with a degree in Computer Science from Mumbai University and currently works as an Editor. His academic interests lie in Databases and Computer Security and he feels passionately about anything open source. This made him choose WordPress while looking for a platform to profess his obeisance to The Beatles.

I would like to thank my Mom and Dad for all their love and for keeping me with them for so long. To all my friends who love me, though I still show no promise of ever doing anything useful.

For More Information: <http://www.packtpub.com/wordpress/book>

WordPress Complete

WordPress is an open-source blog engine released under the GNU general public license. It allows users to easily create dynamic blogs with great content and many outstanding features. It is an ideal tool for developing blogs and though it is chiefly used for blogging, it can also be used as a complete CMS with very little effort. Its versatility and ease of use have attracted a large, enthusiastic, and helpful community of users.

If you want to create powerful, fully featured blogs in no time, this book is for you. This book will help you explore WordPress showing you what it offers and how to go about building your blog with the system. You will be introduced to the main aspects of a blog — users, communities, posts, comments, news feeds — and learn how to manage them using WordPress.

You will develop the skills and confidence to manage all types of content, be it text or images, on your blog, and also understand how users interact with the blog. In working through the book you'll be inspired as well as informed, and have the capability and the ideas to make your blog cutting edge and exciting to maximize its impact.

What This Book Covers

Chapter 1 will take you to the world of blogging by introducing different blog engines available on the Internet. This chapter will also introduce you to the types of blog and the core parts of a blog. You will find this chapter very helpful to find out what the major blog engines are and why we choose WordPress among them.

In *Chapter 2*, you will learn how to start using WordPress as a blog engine. You can either register your blog at wordpress.com or set up in your own web host. You will learn how to install WordPress using FTP, how to manage permissions in the MySQL database, how to install it from cPanel, and how you can upgrade your old WordPress blog to a newer one. This chapter also shows the basic operations to kick start your blog.

One of the main attractive features of WordPress is the availability of thousands of themes.

Chapter 3 will guide you to where you can get best themes for your blog and how to choose from them. You will also learn basic modification of themes for a quick change in your blog's appearance.

Chapter 4 will guide you through the details of administering articles and comments in your blog. It will introduce you to all exciting features that may come handy while writing articles. This chapter also shows you how you can post remotely via using the MetaWeblog API and XML RPC interfaces.

Chapter 5 guides you through converting a WordPress blog into full fledged website by using its content management features. With a minimal level of coding you can use WordPress as an awesome content engine for any general purpose website. In this chapter we discuss this process step by step that you can understand it easily.

For More Information: <http://www.packtpub.com/wordpress/book>

Delivering feeds and podcasts are what you will learn from *Chapter 6*. Using all these tricks you can use WordPress as a great tool for audio blogging. You can increase the traffic and quality of your blog by delivering RSS feeds. This chapter will guide you through all these features in a nice and readable way so that you can grasp it quickly.

Chapter 7 is one of the most exciting chapter in this book, which will teach you to develop awesome themes for WordPress. You don't need to be a PHP pro to write themes at all. This chapter will show you how you can add cool features to your WordPress theme with a minimal level of coding. This chapter will also introduce you to online theme generators, which will ease your life.

Blogging is for community and that is what you will learn from *Chapter 8*. How to turn your blog into a community blogging site and how to manager privileges among users is covered in this chapter.

Chapter 9 will introduce you to the world of plug-ins and widgets. You can extend the WordPress engine by using these plug-ins and widgets, add awesome features to your blog, and release them to community. This chapter will guide you through the detailed lifecycle of WordPress plug-ins, developing them and working with then cleverly with automated installations. This is one of the coolest chapter from this book.

Chapter 10 will describe all the administrative tasks of your WordPress blog. You will learn how to upgrade WordPress, how to trouble-shoot some installation and post installation errors, how to install WordPress MU, and how to back up and restore your blog. This is one of the most important chapters in this book, which you will want to read time and again.

For More Information: <http://www.packtpub.com/wordpress/book>

5

Non-Blog Content

WordPress is a powerful content management system, which you can use for building general websites for your organization. A typical website consists of several pages and sometimes some special services like payment processing, RSS, forums, shoutbox, etc. In WordPress, you can do almost everything with the help of several plug-ins. You can also develop a general-purpose website by developing pages, which are basically considered as non-blog content. Before developing a site for your company or product, you may need to plan the site structure; otherwise, the site may look quite messy.

In this chapter, we are going to develop a general website that serves as a showcase for a production company. As a sample, we will develop a website for Packt Publishing, where it will display its books for sale and accept feedback from readers. Let's decide the goal of this site, and then we will proceed towards the site structure.

Goals and Site Structure

We are going to design a website for a publishing company. These sites have some specific goals to fulfill. For example, in our site we will be delivering the following content. However, in real life, a publishing site must also have payment-processing options besides these listed options:

- About page
- Homepage, where visitors get the latest news
- Books page
- Search box
- A page for authors
- Navigation panel
- Top-ranked book list
- An image gallery

For More Information: <http://www.packtpub.com/wordpress/book>

Our site has a front page where we will display our latest events. From this page, users will be able to navigate to the book gallery. They can also visit our top-ranked books, where we rank our books based on the hit counts. There must be a page that describes our company, and a page for authors to help them kick-start writing their book with our company.

In our book gallery page, visitors will be able to search books on different criteria. For example, they can sort them by the date of publishing, topics, alphabetically, or by price. From this page, they can click over any book for a detailed description.

A search option must be present so that users can search for books on any topic. They must be able to search by author name, book title, or any specific keyword.

A list of top-ranked books must be present in our front page. You may wonder how we calculate the top-ranked books. For the sake of simplicity, we will calculate the top-ranked books by counting the number of times they have been visited. As each book has its own page (to count hits), we will be able to achieve this functionality. However, in real life there may be several other measures on which the book rank is calculated; for example, the quantity of sales, quantity of positive feedbacks, and quantity of reviews. Among these, hit count is also a reliable process. If people are not interested in your book, definitely your book will receive a relatively lower count of hits compared to other books.

In the front page, we will also provide links to some sites that we think should be displayed there. A calendar must also be present for easier navigation by displaying daily or weekly posts as links in it, so that you can easily find them.

Getting Started

The only way we can create a new book entry is making a post. Then we can sort books according to different criteria.

Making the Site

Now it's time to find a suitable theme for our site. Green Marine always delivers a nice look and feel to your site; so let us try with Green Marine for the time being. We need to modify the functionality of this theme to meet our requirements. There will be small changes that you will have to make and you don't need to be a PHP coder for that.


Our first task is to add some pages where we need less modification. We will add the front page, the about us page, and the author page. Keep in mind that our front page will be a regular WordPress page that excludes the post, which will display only some specific content that we want to display on the front page.

In general, all posts come to the front page instantly. However, in our site we are planning to display only some specific posts. There is no built-in option for this purpose, so we need to modify our theme. These are very small modifications and you don't need to be a regular PHP developer for that.

Step 1: Create a Special Category

From the previous chapter, you have learned how to create categories and how to make some post under a category. To solve our problem, we need to create a category that we will use only for some selected posts to be displayed on the front page. Let us create a category **frontpage** in our blog. You can follow these steps for creating a category:

- Log into the WordPress administration panel.
- Click on the **Manage** menu, and select **Categories**.
- Scroll down to the **Add New Category** section.
- Give the category name as **frontpage** and click on **Add Category**. Please note that it is a top category with no parent. You may add an optional description for each and every category.



The screenshot shows the 'Add New Category' form in WordPress. The form is titled 'Add New Category' and contains the following fields:

- Name:** A text input field containing the text 'frontpage'.
- Category parent:** A dropdown menu with 'None' selected.
- Description: (optional):** A text area containing the text 'posts for frontpage only'.

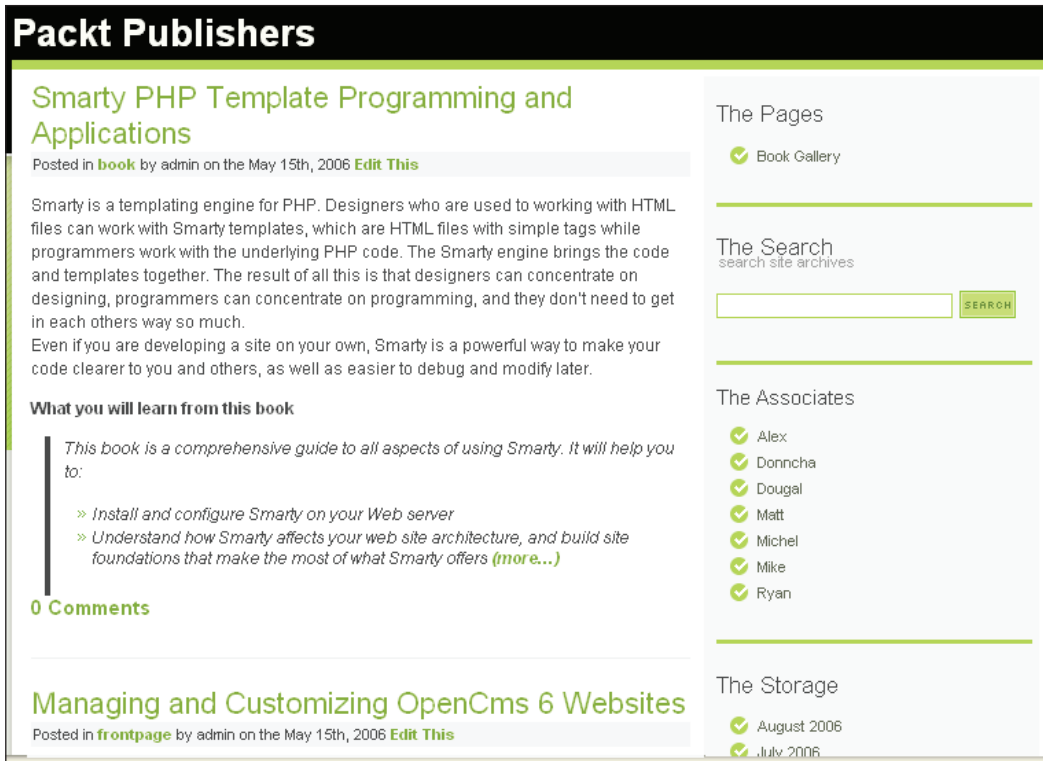
At the bottom right of the form is a button labeled 'Add Category »'.

Step 2: Make Some Posts under this Category

After creating the category **frontpage**, add some posts and select **frontpage** as their category from the right-hand side of the post section. Please be sure that your post belongs to this category only.

Now it's time to add some general posts. For this, create another category **book** and make some posts under that **book** category.

If you look at your blog, you will find all the posts from all categories displayed on the front page. However, according to our requirement we will display only specific posts under the **frontpage** category. So we need to modify our theme a bit and tell WordPress to collect those posts only.



If you look at the preceding screenshot, you will find that posts from all the categories (here **book** and **frontpage**) are displayed on the index page.

Step 3: Modify Main Index Template of the Theme

Using any FTP client or the built-in **Theme Editor**, open the `index.php` page of your theme for editing. As we are using Green Marine, you will find this file in the path `wp-content/themes/greenmarine/index.php`.



If you are not sure about the **Theme Editor**, it's accessible as a sub menu under the **Presentation** menu. From the **Theme Editor**, you will find a list of files in the current selected theme that are editable from here, but they must have write permission.

Add the following line just before the `<?php if (have_posts()) : while (have_posts()) : the_post(); ?>` line:

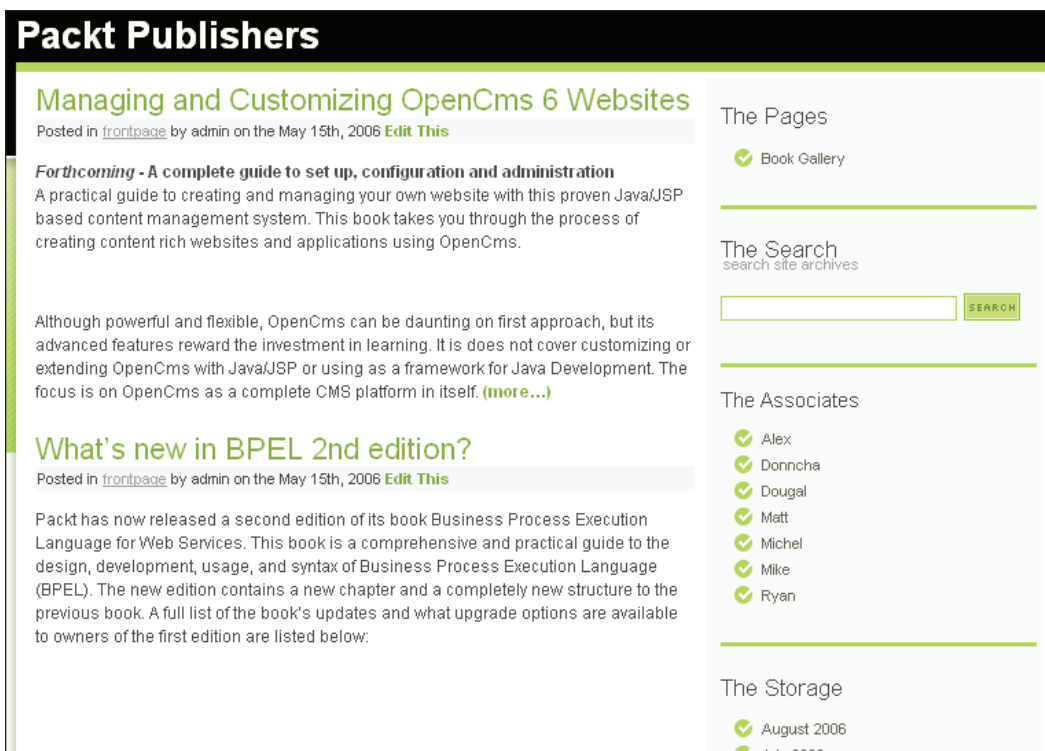
```
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
}
?>
```

The code means that if this is the front page, we will display posts from the category **frontpage** only. So the first ten lines of this file will look like this:

```
<?php get_header(); ?>
<?php get_sidebar(); ?>
```

```
<hr />
<div id="content">
<? if (is_home())
{
    query_posts("category_name='frontpage'");
}
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
```

Save this file and refresh the front page of your blog. You will find that all the posts are gone except those categorized as **frontpage**.

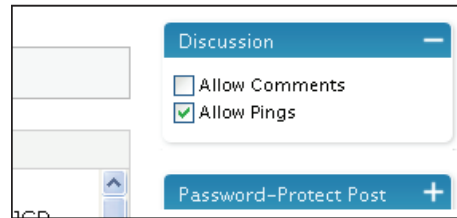


We are now seeing posts only from the **frontpage** category, which was a pretty easy task.

Step 4: Turn off Commenting on Front Page Posts

As it is a site for a publishing firm, we may not want to display public comments on the front page posts or allow commenting on the posts that we are displaying on the front page. Therefore, to turn off commenting for these posts, just open the posts

for editing and turn off commenting by unchecking the checkbox **Allow Comments** from the right side, under the **Discussion** panel.



If you refresh the front page and click on any of the posts, you will find them displaying **Comments are closed**. However, the link **0 Comments** is still present on the front page. Now it's time to remove that link from the front page.

Open the `index.php` file using any editor and search for the following code block:

```
<div class="comments">
  <?php wp_link_pages(); ?>
  <?php comments_popup_link(__('<strong>0</strong> Comments'), __
    ('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
```

We need to tell WordPress that if it is the home page, then don't display the link. So add a small logic test before this code block and modify it like this:

```
<? if (single_cat_title("",false)!="frontpage"):?>
  <div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
      ('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
  </div>
<? endif; ?>
```

The line `single_cat_title("",false)` returns the category name of the current post. So we simply checked that the current category is not **frontpage**; if not we still display the number of comments, using the function `comments_popup_link`; otherwise the link is skipped.

Packt Publishers

Managing and Customizing OpenCms 6 Websites

Posted in [frontpage](#) by admin on the May 15th, 2006

Forthcoming - A complete guide to set up, configuration and administration
A practical guide to creating and managing your own website with this proven Java/JSP based content management system. This book takes you through the process of creating content rich websites and applications using OpenCms. Although powerful and flexible, OpenCms can be daunting on first approach, but its advanced features reward the investment in learning. It is does not cover customizing or extending OpenCms with Java/JSP or using as a framework for Java Development. The focus is on OpenCms as a complete CMS platform in itself. [\(more...\)](#)

What's new in BPEL 2nd edition?

Posted in [frontpage](#) by admin on the May 15th, 2006

Packt has now released a second edition of its book Business Process Execution Language for Web Services. This book is a comprehensive and practical guide to the design, development, usage, and syntax of Business Process Execution Language (BPEL). The new edition contains a new chapter and a completely new structure to the previous book. A full list of the book's updates and what upgrade options are available to owners of the first edition are listed below:

The Pages

The Search
search site archives

The Associates

- ✓ Alex
- ✓ Donncha
- ✓ Dougal
- ✓ Matt
- ✓ Michel
- ✓ Mike
- ✓ Ryan

Step 5: Populate Some Data

Now it's time to add some data to our database. As our website is a general-purpose website for publishers, we must have some books. We have to categorize each book in different book categories. In our sample website, we have four categories. Before populating data, let's create these four categories from the administration panel. We will make every book entry as a regular post, but they must belong to appropriate categories.

- Open-source
- Java
- PHP
- Content Management System

Don't worry about the display of these posts. They will not appear in the front page, since we are displaying only some specific posts on our front page.

Step 6: Create Category Pages

Now we have to create some category pages. If our visitor clicks on any category, they will see just the titles of the books. When they click on the title of one of these books, they can see the details about that particular book.

So we have to modify our theme again. Let us open the `index.php` file inside our theme, i.e. in the **Green Marine** folder, and modify it as shown in the following code segment:

```
<?if (is_category()) $cat=true;?>
<?php get_header(); ?>
<?php get_sidebar(); ?>

<hr />
<div id="content">
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
    $homepage= true;
}
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <?php // Post dates off by default the_date('','<h2>','</h2>'); ?>
    <div id="post-<?php the_ID(); ?>">
    <h2><a href="<?php the_permalink() ?>" rel="bookmark">
    <?php the_title(); ?></a></h2>
<? if (!$cat):?>
    <div class="meta">
    <?php _e("Posted in"); ?> <?php the_category(',') ?>
    by <?php the_author() ?> on the <?php the_time('F jS, Y') ?>

    <?php edit_post_link(__('Edit This')); ?></div>
    <div class="main">
        <?php the_content(__('(more...)')); ?>
    </div>
<?endif;?>
</div>

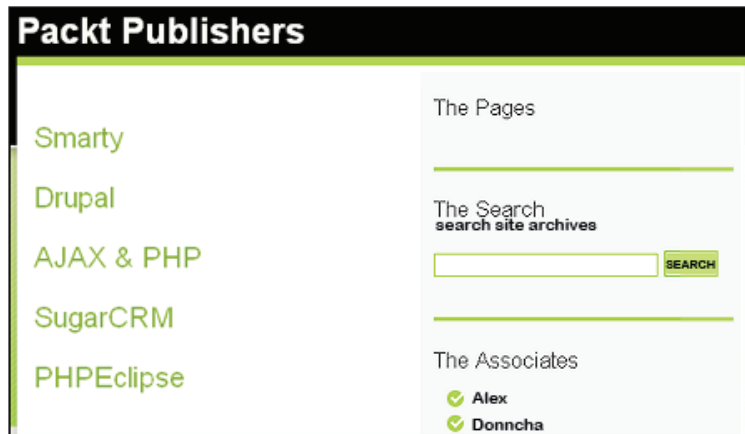
<? if (single_cat_title("",false)!="frontpage"):?>
<? if (!$cat):?>
<div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
<? endif;?>
<? endif; ?>
<?php comments_template(); ?>
```

```
<?php endwhile; else: ?>
<div class="warning">
  <p><?php _e('Sorry, no posts matched your criteria, please try and
search again. '); ?></p>
</div>
<?php endif; ?>

<?php posts_nav_link(' &#8212; ', __('&laquo; Previous Page'), __
('Next Page &raquo;')); ?>

</div>
<!-- End float clearing -->
</div>
<!-- End content -->
<?php get_footer(); ?>
```

If you look at the preceding code, you will find the modified code highlighted. At the top-most line, we just check whether the page is a category page. If so, we initialize a variable as `true`. We did this at the top of the page because the `is_category()` function always returns `true` inside a WordPress 'post' loop; so we must check it outside the loop. After that we skipped displaying the details of the post, if it is a category page. Finally, we hid the pop-up comment link. Now if you visit a particular category, you will see the following screen:

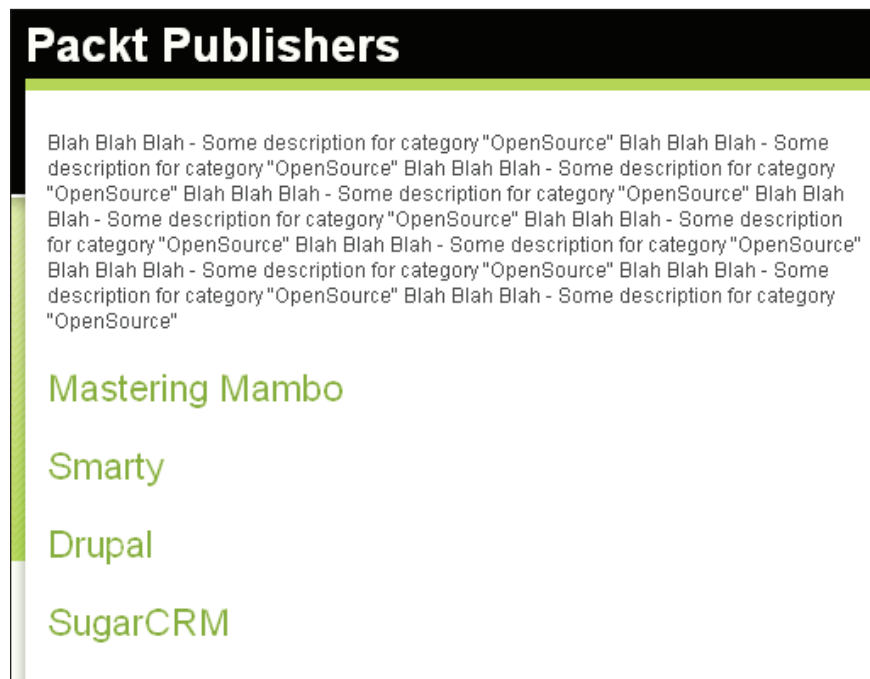


This is the typical look of a category where we display only the titles of the books. Each of these links is clickable and will navigate to the book's details when clicked.

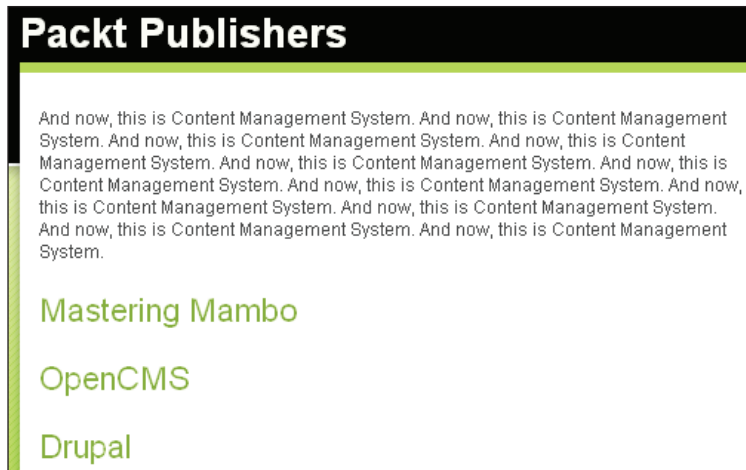
You may not find the layout of the screen satisfactory. However, you can display it any way you want; you just have to know some CSS coding and that's all. You can do the rest. For the sake of simplicity, I skipped that part in this book.

Step 7: Add Some Description to Each Category Page

We will now add some description at the top of each category page. For example, when people click on the **Open-Source** category, they will see a detailed description of this category and what type of books are covered here. After that description, we will display our regular listing of books. Each category page must show its unique description. Let us take a look at the following screenshot:



The preceding page is displayed when someone clicks on the **Open-Source** category. Let us take a look at another category page:



This is the **CMS** category. Please note that these two category pages each display their own description at the top of the page.

To achieve this functionality, we must add some description to each category from the administration panel. Let us log into the admin panel and select the **Manage** menu. Now click on the **Categories** sub-menu. You will find every category in your WordPress blog listed in this category management panel. Beside each category there is an **Edit** link. Click on that link and you will find the page where you can add a description for them.

Edit Category

Category name: CMS

Category slug: cms

Category parent: None

Description: And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And

Edit category »

After adding some description, save it by clicking on the **Edit Category** button.

Now we have to modify our theme so that we can display these descriptions at the top of each category page. We first have to extract the description of the current category. Unfortunately, there is no built-in function in WordPress to do that. However, we can make use of some other functions. Firstly, we have to extract the current category name. We can do that using the `single_cat_title()` function. After that we must retrieve the current category object from which we can get the category description. Let's take a look at the code of the `index.php` file in our theme:

```
<?if (is_category()) $cat=true;?>
<?php get_header(); ?>
<?php get_sidebar(); ?>

<hr />
<div id="content">
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
    $homepage= true;
}
?>
<?
$catobj=get_category(get_cat_ID(single_cat_title('',false)));
echo $catobj->category_description;
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <?php // Post dates off by default the_date('','<h2>','</h2>'); ?>
    <div id="post-<?php the_ID(); ?>">
    <h2><a href="<?php the_permalink() ?>" rel="bookmark">
    <?php the_title(); ?></a></h2>
<? if(!$cat):?>
    <div class="meta">
    <?php _e("Posted in"); ?> <?php the_category(',') ?> by
    <?php the_author() ?> on the <?php the_time('F jS, Y') ?>

    <?php edit_post_link(__('Edit This')); ?></div>
    <div class="main">
        <?php the_content(__('(more...)')); ?>
    </div>
<?endif;?>
</div>

<? if (single_cat_title("",false)!="frontpage"):?>
```

```
<? if(!$cat):?>
<div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
<? endif;?>
<? endif; ?>
<?php comments_template(); ?>

<?php endwhile; else: ?>
<div class="warning">
    <p><?php _e('Sorry, no posts matched your criteria, please try and
search again. '); ?>
    </p>
</div>
<?php endif; ?>

<?php posts_nav_link(' &#8212; ', __('&laquo; Previous Page'), __
('Next Page &raquo;')); ?>

</div>
<!-- End float clearing -->
</div>
<!-- End content -->
<?php get_footer(); ?>
```

If you take a look at the preceding highlighted code, you will find that we used three functions, namely `single_cat_title()`, `get_cat_ID()`, and `get_category()`. The `get_cat_id()` function returns the ID of a category when the name of that category is supplied to this function as a parameter, and the `get_category()` function returns a category object when the category ID is passed as a parameter to it. When the following line of code executes, we get a category object in the `$catObj` variable:

```
$catobj=get_category(get_cat_ID(single_cat_title('',false)));
```

Now we will just access its description property and display it:

```
echo $catobj->category_description;
```

That's it! Notice how we can achieve some great functionality with just a few lines of code.

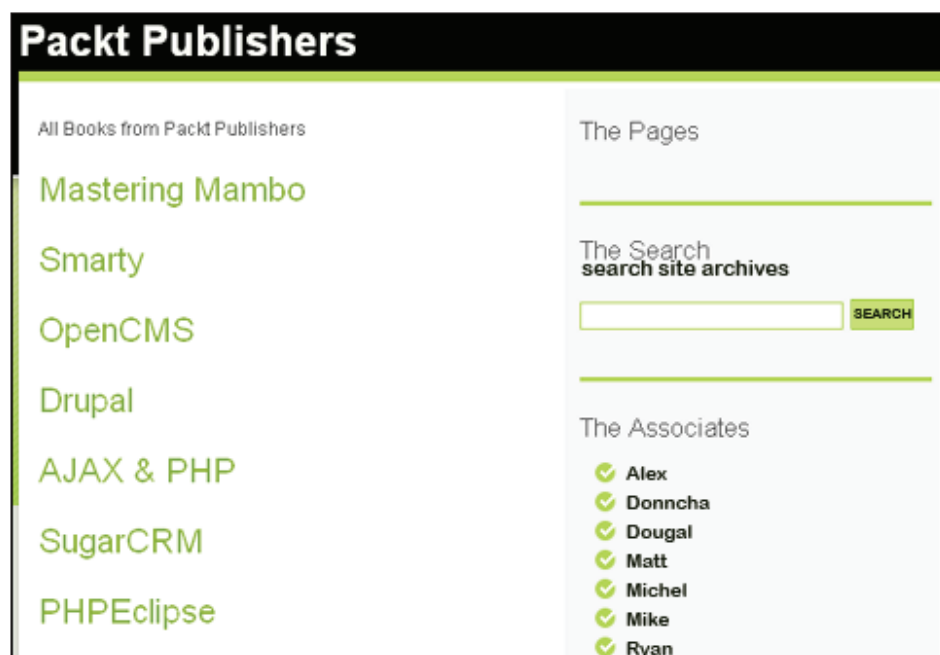
Step 8: An All Books Section

Well, now we need to display all the books in one single page despite their parent category. That is, whether a book belongs to the **Java** category or the **open-source** category, they must all show up in the **All Books** page.

We have to first create a category named **All Books** from the **Category** section. Then we can proceed further with the other tasks.

Next, we just have to modify all our book posts and add the category that we created earlier as **All Books**. That is, a book that belongs to the **PHP** category or whatever other category, must also belong to this **All Books** category.

For this, just log into the admin panel and then select the **Manage** menu. Create a category called **All Books** from the **Category** sub-menu. Now click on the **Posts** sub-menu and edit each post to add the **All Books** category to it from the right-hand side category pane. From now on, for every book that you will add, add a parent category under which that book should display and of course this **All Books** category.

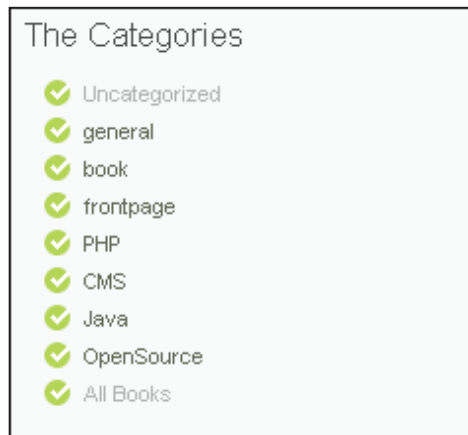


Now we have all our books displayed on one single page. Simple tricks save a lot of time. When you are not very familiar with the internals of WordPress, you may be scared to do a lot of coding for simple tasks. However, in real life WordPress is

so flexible and extensible that you can do almost everything with this amazing blog engine.

Step 9: Customize the Categories

Now it's time to decorate our **Category** menu so that it shows only the necessary categories and no extra ones. If we look at our **Category** menu now, we will find all our categories being displayed and creating a total mess. Let us take a look:



That's definitely not the look we want. We want to display only the following menus here:

- PHP
- CMS
- Open-Source
- Java
- All Books

We need to exclude all other unnecessary menus from the list. There are again some built-in features that we can make use of. Now we need to modify the `sidebar.php` file inside our theme. If you open the file, you will find that the following segment is responsible for displaying the categories in a list.

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats(); ?>
</ul>
```

This `wp_list_cats()` function generates a list from all the available categories and returns them in a list. Now we want to exclude some of these categories from being displayed. The `wp_list_cats()` function supports category exclusion. All we have to do is to supply the IDs of the categories that we want to hide as a parameter to this function. However, we need to know the category IDs for this.

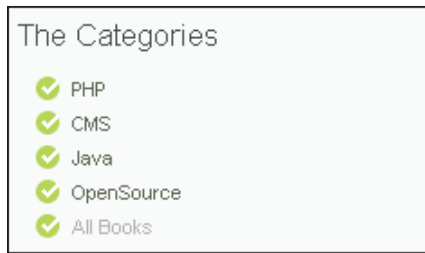
Now log into the administration panel and choose the **Categories** sub-menu from **Manage**. You will find a page that looks like the following screenshot:

ID	Name	Description	# Posts	Action
12	All Books	All Books from Packt Publishers	7	Edit Delete
5	book		1	Edit Delete
9	CMS	And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System.	3	Edit Delete
6	frontpage		7	Edit Delete
2	general		1	Edit Delete
10	Java		1	Edit Delete
		Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category		

Here, you find all the categories in our blog. In the left-most column, you will find their ID. In the preceding screenshot, we have noted the IDs we want to exclude. Those are **5**, **6**, **2**, and **1**. Now let us edit the `wp_list_cats()` function in `sidebar.php` as follows:

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6"); ?>
</ul>
```

Now refresh your WordPress blog and take a look at the category panel.



Now we have excluded all the unnecessary categories from our category panel. However, there is still something more to do. If you look at your category panel, you will find that the categories are displayed in no specific order. We can show them in ascending or descending order. For this we need to modify the preceding code block as shown below:

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
```



Now it is sorted and looks perfect. However, we don't want a title like **The Categories**. Let us make it **Book Categories**. Just modify the preceding code block and replace the The Categories text with Book Categories.

```
<h3><?php _e('Book Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
```



Now it perfectly fits with what we want.

Step 10: Adding the About Us Page and Author Page

Now we will add some other static pages as we planned earlier while planning the site structure. There will be a page about the company and a page solely for authors. Both of these pages are static and they will be displayed as a page, not regular posts. Besides these two static pages, we must have an image gallery and a top-ranked book list.

Let us add these static pages first. That will not be a big job at all. Just add them as pages, and you are done!

Step 11: Top Ranked Book List

This is a slightly tough part of this chapter. To display some books as top-listed books we have several options. We may create a manual list by editing the sidebar; however, it will not be dynamic at all and is not an acceptable solution. A better solution is to create a real top books list by counting the number of views of each book and dynamically displaying them. For this purpose, we need to create a plug-in on our own. Don't worry, we will keep it simple.

To make this plug-in work properly, we have to store the number of hits in persistent storage, that is where we can re-use them. You can use the database for storing them and that will be the best solution. However, as we are going for a simple plug-in, we will be using a simple text file for this purpose; but remember to use the database (either MySQL or SQLite) wherever possible. If you are already familiar with PHP coding, you can modify the code to suit any database table instead of a flat file.

Now we will design a plug-in, which adds a one pixel by one pixel image to the content of every post dynamically. The image will refer to a PHP file that will act as a hit counter after each call. The following code explains this:

```
<?
/*
Plugin Name: Hit Counter
Plugin URI: Uh.. No URL
Description: A plugin which counts the number of times a post been
viewed
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
```



```
?>
<?
add_filter("the_content", "HitCount");
function HitCount($content)
{
    global $id;

    $categories = get_the_category();
    foreach($categories as $category)
    {
        if ($category->cat_name=="All Books")
            return $content."<img style='display:none '
                src='http://wordpress_url/hitcount.php?id={ $id} ' >;
        }
    }
    return $content;
}
?>
```

Please replace `wordpress_url` with your WordPress installation URL. Save this file as `hitcounter.plugin.php` and place this file in the `/wp-content/plugins` folder using any FTP client.

Now log into the admin panel and click on the **Plugin** menu. You will see a list of all the available plug-ins. The plug-in (**HitCounter**) also appears here. From this admin panel, you can activate and deactivate any plug-in. However, don't activate the **HitCounter** plug-in. We have something else to do.

Plugin Management			
Plugins extend and expand the functionality of WordPress. Once a plugin is installed, you may activate it or deactivate it here.			
Plugin	Version	Description	Action
Akismet	1.14	Akismet checks your comments against the Akismet web service to see if they look like spam or not. You need a WordPress.com API key to use this service. You can review the spam it catches under "Manage" and it automatically deletes old spam after 15 days. Hat tip: Michael Hampton and Chris J. Davis for help with the plugin. <i>By Matt Mullenweg.</i>	Activate
Hit Counter	1.0	A plugin which counts the number of times a post has been viewed <i>By Hasin Hayder.</i>	Activate
Hello Dolly	1.5	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. <i>By Matt Mullenweg.</i>	Activate

Now we need to create another file called `hitcount.php` in the root directory of our WordPress installation.

```
<?
$id = $_GET['id'];
$unserialized_data = file_get_contents("hits.txt");
if (empty($unserialized_data))
{
    $hits = array();
}
else
{
    $hits = unserialize($unserialized_data);
}
//echo "<pre>".print_r($hits, true)."</pre>";
$hits[$id]=$hits[$id]+1 ;
$fp=fopen("hits.txt", "w");
flock($fp, LOCK_NB);
fwrite($fp, serialize($hits));
flock($fp, LOCK_UN);
fclose($fp);
?>
```

We need to do one more thing. Let us create a file called `hits.txt` in your WordPress root directory using an FTP client and change its permission to `665` or `rw-rw-r---`.

Now we need to create a function that generates our top books list. Let us modify our `hitcounter.plugin.php` file as follows:

```
<?
/*
Plugin Name: Hit Counter
Plugin URI: Uh.. No URL
Description: A plugin which counts the number of times a post been
viewed
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
?>
<?
add_filter("the_content", "HitCount");
function HitCount($content)
{
```

```
global $id;
$categories = get_the_category();
foreach($categories as $category)
{
    if ($category->cat_name=="All Books")
        return $content."<img style='display:none '
            src='http://localhost/wp/hitcount.php?id={$id}'>";
}
return $content;

function GetTopBooks($limit=10)
{
    $top_books = array();
    $unserialized_data = file_get_contents("hits.txt");
    $hits = @unserialize($unserialized_data);
    $limit=$limit>count($hits)?count($hits):$limit;
    $i=0;
    if (is_array($hits))
    {
        arsort($hits, SORT_DESC);

        foreach($hits as $post_id=>$post_hit)
        {
            $i++;
            if ($i>$limit) break;
            //$post_id = $hits[$i];
            $current_post = get_post($post_id);
            $top_books[] =
                "<li><a href='http://localhost/wp/?p={$post_id}'>".
                $current_post->post_title."<a/></li>";
        }
    }
    return implode($top_books);
}
?>
```

Now we have a `GetTopBooks()` function that returns the list of top books in an array. We are going to use it in `sidebar.php`.

So modify `sidebar.php` as follows. The added portion of code has been highlighted.

```
<!-- begin sidebar -->
<div id="right">
    <!--
        <div id="author">
```

Here is a section you can use to briefly talk about yourself or your site. Uncomment and delete this line to use.

```

    <h3><?php _e('The Author'); ?></h3>
    <p>Your description here.</p>
</div>
<div class="line"></div>
-->
<div id="links">

<div id="pages">
    <h3><?php _e('The Pages'); ?></h3>
    <ul>
        <?php wp_list_pages('title_li='); ?>
    </ul>
</div>

<div class="line"></div>

<h3>The Search</h3>
<p class="searchinfo">search site archives</p>
<div id="search">
    <div id="search_area">
        <form id="searchform" method="get" action=
            "<?php echo $_SERVER['PHP_SELF']; ?>">
            <input class="searchfield" type="text" name="s"
                id="s" value="" title="Enter keyword to search" />
            <input class="submit" type="submit" name=
                "submit" value="" title="Click to search archives" />
        </form>
    </div>
</div>
<div class="line"></div>
<h3>Top Books</h3>
<ul>
<?=GetTopBooks() ?>
</ul>

<div class="line"></div>

<h3><?php _e('The Associates'); ?></h3>
    <ul>
        <?php get_links('-1', '<li>', '</li>', '', 0, 'name', 0, 0,
-1, 0); ?>
    </ul>
<div class="line"></div>

```

```
<h3><?php _e('The Storage'); ?></h3>
<ul>
  <?php wp_get_archives('type=monthly'); ?>
</ul>

<div class="line"></div>

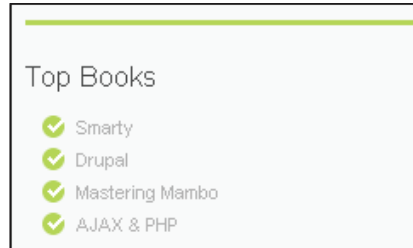
<h3><?php _e('Book Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
<div class="line"></div>

<h3><?php _e('The Meta'); ?></h3>
<ul>
  <!-- <li><?php // wp_register(); ?></li> -->
  <li>Hello</li>
  <li><?php wp_loginout(); ?></li>
  <li><a href="http://wordpress.org/" title=
    "<?php _e('Powered by WordPress, state-of-the-art
      semantic personal publishing platform.');" ?>"
    <abbr title="WordPress">WP</abbr>
  </a></li>
  <li><a href="http://gmpg.org/xfn/">
    <abbr title="XHTML Friends Network">XFN</abbr>
  </a></li>
  <li><a href="feed:<?php bloginfo('rss2_url'); ?>"
    title="<?php _e('Syndicate this site using RSS'); ?>"
    <?php _e('<abbr title="Really Simple Syndication">
      RSS</abbr>'); ?>
  </a></li>
  <li><a href="feed:<?php bloginfo('comments_rss2_url'); ?>"
    title="<?php _e('The latest comments to all posts
      in RSS'); ?>"
    <?php _e('Comments <abbr title="Really Simple
      Syndication">RSS</abbr>'); ?>
  </a></li>
  <li><a href="#content" title="back to top">Back to top
  </a></li>
  <?php wp_meta(); ?>
</ul>

</div>

</div>
<!-- end sidebar -->
```

This will add the top books list in the sidebar as follows:



As soon as any visitor visits any book, the data will be updated and the list will be sorted accordingly.

Step 12: Adding Links of Associates

Now it's time to add some links of our associates. The associates will be displayed under a special **Associates** section. To add a link in our WordPress installation, log into the admin panel and select **Links**. You will see the available list of links as shown in the following screenshot:

Name	URI	Category	rel	Image	Visible	Action	
Alex Description:	alexking.org	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Donncha Description:	blogs.linux.ie/xeer	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Dougal Description:	dougal.gunters.org	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Matt Description:	photomatt.net	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Michel Description:	zengun.org/weblog	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Mike Description:	zed1.com/journalized	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>
Ryan Description:	boren.nu	Blogroll		No	Yes	Edit	Delete <input type="checkbox"/>

You can manage all existing links from here. However, to add a new link just select the **Add Link** sub-menu under this menu. You will be redirected to a page where you can add new links.

Manage Links **Add Link** Link Categories Import Links

Add a link:

Basics

URI:

Link Name:

Short description:

Category: 1 : Blogroll

As soon as you add a link, it will be visible in the sidebar. You can also add links under different categories from the **Link Categories** menu.

Step 13: Adding an Image Gallery

Now here's another tough part of our project. We need to add an image gallery where we will display all cover pages of our books as images. To do this, let us see our possible choices:

- We can use a plug-in that extracts all images from a specific directory and displays them.
- We can store our images in any photo organizer (like Flickr) and displaying them from that organizer using any compatible plug-in; but we are not going to do that when we are deploy our blog in our self-managed hosting.
- We can add images as a separate post that will link to each of the individual original posts.

In this section, we will see how to implement all these possible solutions. The best solution for our problem is the third option. However, in different scenarios the other two options would be considered as the best solution. So let's start one by one.

Solution 1: Extract Images from a Specific Directory

For this, we need to design a plug-in that extracts all images from any specific directory and displays them as a gallery. Moreover, when someone clicks on any of those images, it will redirect to its corresponding book post. To accomplish this, we need to follow one rule, and that is, we must name each image according to the format `PostId-imageTitle.ImageExtension`.

Let us now design the plug-in. It will extract all images from the **coverpage** folder inside our WordPress root directory.

```
<?
/*
Plugin Name: Image Gallery
Plugin URI: Uh.. No URL
Description: A plugin which displays all images under any dir as image
gallery
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
?>
<?
add_filter("the_content", "ImageGallery");

function ImageGallery($content)
{
    if (strpos($content, "{gallery}")!==false)
    {
        $content = str_replace("{gallery}","", $content );
        $fp = opendir("coverpage");
        $table = "<table cellpadding='5' cellspacing='5'>";
        $i=false;
        while ($dir = readdir($fp)) {
            if (is_file("coverpage/{ $dir}"))
            {
                if(false==$i)
                {
                    $table .="<tr>";
                }
                $i = !$i;
                $file = split("-", $dir);
                $post_id = $file[0];

                $table .="<td>
```



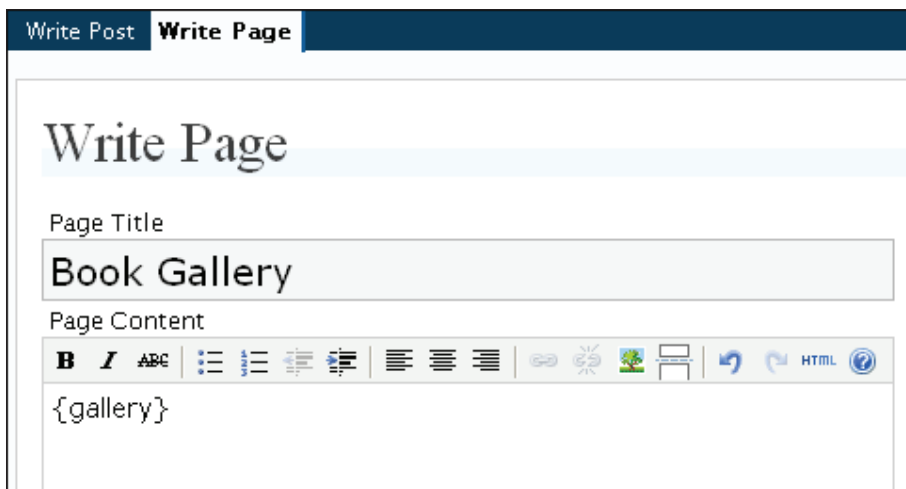
```

        <a href='http://localhost/wp/?p={ $post_id }' >
        <img src=' 'http://localhost/wp/coverpage/{ $dir } ' ></a></td>";
        if (false==$i)
        {
            $table .="</tr>";
        }
    }
}
$table .="</table>";
}
return $content.$table;
}
?>
```

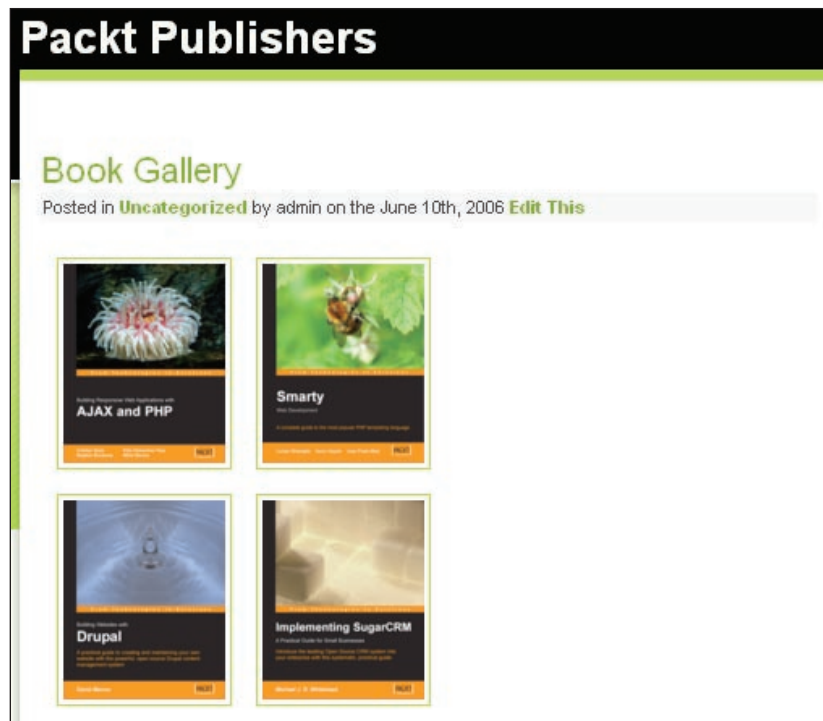
Now save this file as `imagegallery.plugin.php` in the `wp-content/plugin` folder. Next enable the plug-in from the **Plugin** administration panel.

To see the plug-in in action, create a folder named **coverpage** in the WordPress root directory. Inside the directory, place some image following the exact naming convention `postid-filename.extension`.

Now let us create a new page where you just add the string **{gallery}**. Please note that it is a page and not a regular post, since we are displaying this gallery as a separate page in our blog. You can also add some additional text; but **{gallery}** must be present in the content of the text.



That is all! Visit this page in your browser, and you will see the image gallery.



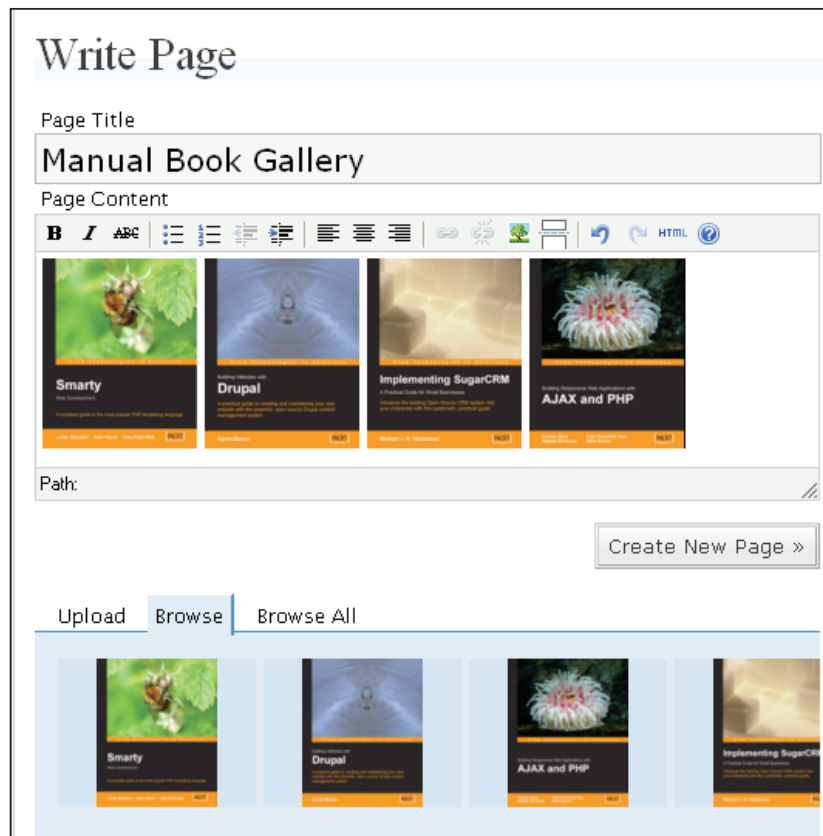
Please note that you must add the string `{gallery}` wherever you want to display this image gallery. The string `{gallery}` can be added to any post or any page. Whenever you add the text `{gallery}` in the content, it will display the gallery there. It is not specific to this page only.

Solution 2: Image from Photo Organizer

There are some interesting plug-ins already available for this purpose. FAlbum does its job pretty well, if you are using photos from your existing Flickr account. If you are using any other popular image organizer/gallery script like Gallery or Coppermine, please take a look at the Codex plug-in directory for appropriate plug-ins.

Solution 3: Creating a Gallery Manually

This is the easiest step among these three solutions. Here all we need to do is to create a page where we add every image using the `` tag or simply using the built-in image browser button at the bottom of the **Write Page**. Before that, we need to upload each and every book image with the built-in uploader of WordPress. Next, we will add those pictures one by one manually to our page. We can also add hyperlinks to these images using the `<a>` tag or by clicking the link button from the toolbar. That's it. Take a look at the following screenshot:



Summary

In this chapter, we have learned how to create a complete general-purpose website with WordPress. These posts are considered as non-blog content. WordPress provides an extremely flexible set of APIs with which you can extend your theme and achieve almost every kind of functionality with minimum effort and code modification.

If you study the WordPress API from Codex, you will learn how to extend WordPress in more depth than discussed here. Moreover, we have two complete chapters on how to develop themes and plug-ins from scratch. In those chapters, we will discuss WordPress APIs in more details. For now, please review the modifications that you have just done in this chapter and try to play with them by modifying them according to your requirements.

Where to buy this book

You can buy WordPress Complete from the Packt Publishing website:
<http://www.packtpub.com/wordpress/book>

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



www.PacktPub.com

For More Information: <http://www.packtpub.com/wordpress/book>