WEBD 236 Lab 2

If you use an external source (i.e. a web-page, the required textbook, or an additional book) to help you answer the questions, then be sure to cite that source. You should probably always be citing a source.

Problem

Blog engines represent a good problem to solve in an introductory web programming course. A blog (a shortening of “web log”) is a place for a person to post their thoughts about any subject. A blog post is an entry in the blog. Blog posts have a title, content, tags, and a date of posting. Your assignment is to write a simple blog engine that lets you post as many items as you wish. *We will be revisiting this assignment in subsequent labs (i.e. to hyperlink tags, add commenting, add authentication and sessions, etc.)* So, it behooves you to do a good job here.

You should have an index page located at ‘somedir/index’ where somedir is the name of the directory in which your project is stored. It should look something like this:

```
My Blog

Recent Posts

First Post
Posted on: 2012-01-23
Filed under: rocks posts
My first post really rocks....

Second post
Posted on: 2012-01-23
Another attempt at an interesting blog post...

A longer post with several paragraphs
Posted on: 2012-01-23
Filed under: tid
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut...

Add a post
Copyright © 2012 Todd Whittaker
```

Notice that this screen displays the most recent five blog posts, and has a link at the bottom for adding more blog posts. If there are fewer than 5 recent posts, only the most recent 5 should be displayed. Also, notice that you can click the title of the blog post and view it in detail. Here is the long post displayed:
From this page, you can click the ‘Edit’ link to edit the post, or the ‘Delete’ link to delete it entirely. Also notice that each individual paragraph is displayed with appropriate line breaks. If you store raw text in your database, you can use the nl2br() function (newline to break) to display nice paragraphs. Clicking the edit button will yield the following page:
You should do error checking on any edits. For example, there must be a title and there must be content. Tags are optional. If you don’t have required fields, then you should *not* update the database, but instead should show error messages and ask for the issue to be fixed. Please *keep* good form data so that users don’t need to enter it in a second time. Notice what happens if I delete the title:
The same kind of validation should take place when adding a post:
Finally, deleting a post should remove the post from the database and redirect back to the index page.

**Helpful Hints**

- You’ll only need one database table called “posts.” This table should have five columns: the ID number, the date of posting, the title, the content, and the tags as a single text field (not as a related entity).
- You should use SQLite (not MySQL) for this project.
- Make sure that your project works on any server on any directory. In other words, you should never hard-code a URL with the name or IP address of your machine. We won’t be using your machine when we test it. Also, you should not hard-code a directory name in your application. It should run as http://localhost/blog/index or as http://localhost/myblog/index or any other directory URL.
- Use the MVC framework developed in class. This will help.
- To keep data in forms, you will need to echo out the content of variables within tags. For example, you may have something that looks similar to this:

```php
<input type='text' name='title' id='title' value='<?php echo($title); ?>'/>
```
Of course, you need to make sure that the variable $title has some content in it from the last post.

- Remember that every operation that writes to the database should be followed by a redirect whereas every ‘get’ (generally) should be followed by a forward (i.e. a template render).

**Submission instructions**

Create a ZIP file of your entire project contents as it is found under your C:\XAMPP\htdocs\blog\ directory. This ZIP file should contain your database, your PHP scripts, style files, etc. Follow proper coding conventions (indentation, commenting, etc.) that you would have learned in previous courses.

Submit the ZIP file to the dropbox for this assignment in the course.