**Assignment 3 - Blogging Application**

**Goal**

Use [these starting files](https://drive.google.com/file/d/1--s9udSniyOX0kcaVahyVvWZ_eCrYM5l/view?usp=share_link). Failure to do so will result in a mark of 0 for this assignment.

The goal of this assignment is to create a simple blogging application. This application will include username and password authentication along with the full suite of CRUD tasks for blog posts.

This assignment will also be a test of your overall PHP coding abilities as you will need to make use of a wide range of PHP skills to complete the blogging application. You will be marked using the rubric at the end of this assignment. *Be sure to read over this rubric before you begin coding.*

A working version of this assignment can be seen here: <http://stungeye.com/school/blog/>

*(user:* ghostface *pass:* killa*)*

**Blog Posts**

For this assignment a blog post will consist of the following elements:

* Title of the Post
* Content of the Post
* Date/Time Stamp when the Post was saved -- use TIMESTAMP datatype in MySQL (which sets the column value to the current date/time by default)

**Authentication**

Authentication will be handled using simple HTTP authentication (See Appendix a). In the sections below I will define the different user stories available to authenticated and unauthenticated users.

**Unauthenticated User Stories**

As an non-authenticated user I should be able to:

* View a home page that lists the title, date/time stamp and excerpt of the 5 most recently posted blog entries (in reverse chronological order).
* Click the title and "Read Full Post" links on the home page to view a full blog post.

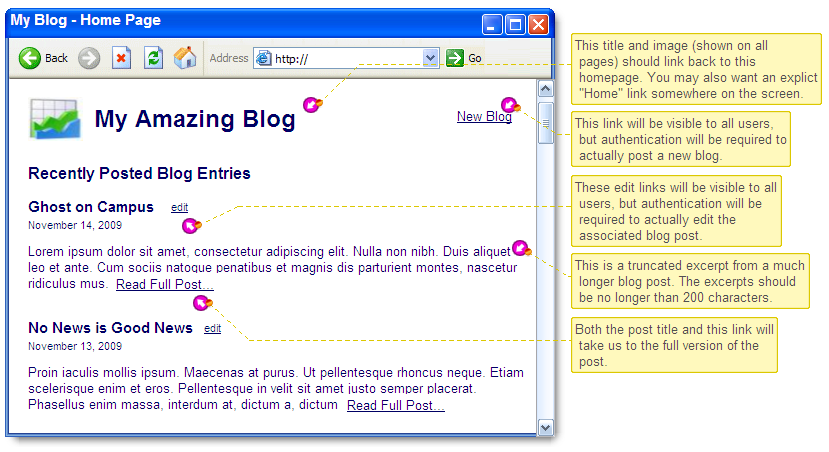
**Authenticated User Stories**

As an authenticated user I should be able to:

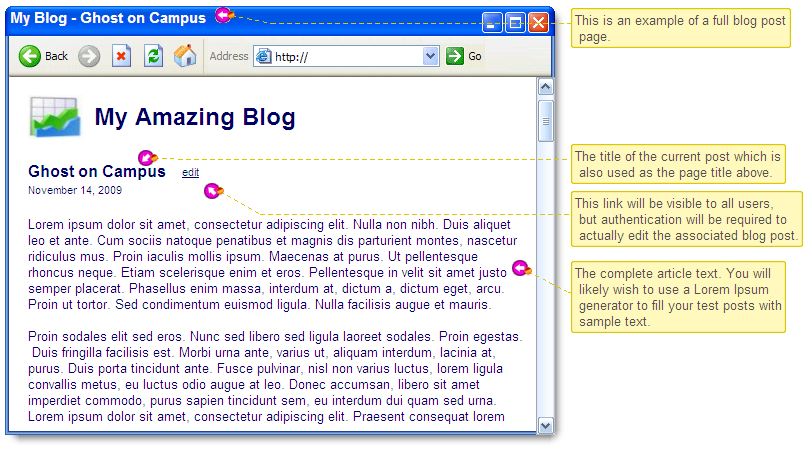
* Post a new blog entry using an HTML form.
* Edit any of the existing post using an HTML form.
* Delete any of the existing posts.

**Screen Mockups**

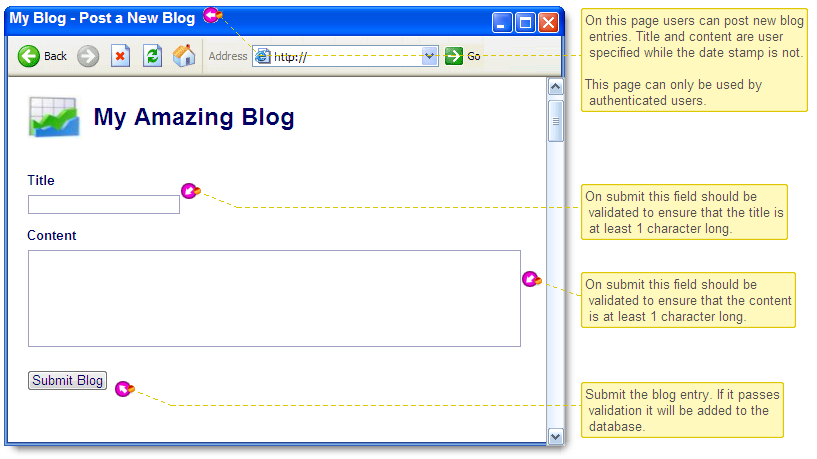
**Home Page:**

****

**Full Blog Post Example:**

****

**New Post Page (Authenticated Users Only):**

****

**Edit Post Page (Authenticated Users Only):**

****

## Submission and Rubric

**You will submit your blogging application within a zip file to the Learn Dropbox folder.**

***Please be sure to include an export of your database in SQL format. (Use phpMyAdmin to export your database.)***

**Deduct from a total of 10 marks the points associated with incomplete or incorrectly implemented items listed below.**

**Home Page: (1 point each)**

* Five most recent blog posts displayed in reverse chronological order.
* For each of these posts you should display: Title, Timestamp, Content
* Blog post titles link to full page for each post. (This link includes a GET parameter to specify the post id.)
* If blog content is larger than 200 characters the displayed content is truncated to 200 characters.
* If the content is truncated a "Read Full Post" link should be displayed after the content. (This link includes a GET parameter to specify the post id.)
* An edit link is displayed for each post. (The link includes a GET parameter to specify the post id.)
* A "New Post" link is present somewhere on this page.

**Full Post Page: (1 point each)**

* Displayed on this page: Post title, timestamp, full post content, edit link
* The blog displayed is determined by a GET parameter in the URL. This parameter should be the post's primary key id from the database table.

**New Post Creation: (1 point each)**

* Provides a form where the user can enter a new post title and contents.
* The form includes a button for submitting the post to the database.
* This page is protected by HTTP authentication.

**Post Update and Delete: (1 point each)**

* Provides a form where the user can edit a specific post title and contents.
* The post being edited is determined by a GET parameter in the URL. This parameter should be the post's primary key id from the database table.
* The title and content of the post being edited should appear in the form.
* The form includes a button for updating the post in the database.
* The form includes a button to delete the current post from the database.
* This page is protected by HTTP authentication.

**Security: (2 points each)**

* Ensure that any id values passed by the user are validated as integers before you use them in a SQL query. This is especially important when updating or deleting a post. If you receive a non-numeric id, redirect the user back to the index page.
* All user submitted strings (POSTed titles and blog content) must be sanitized using input\_filter and inserted/updated using PDO statements with placeholders bound to values.

**Validation and Formatting: (1 point each)**

* All dates should be formatted: "MonthName dd, yyyy, hh:mm am/pm"
  + Hint: You will need a PHP function to format the date. Also check out the strtotime function.
* New / Updated Posts are validated to ensure the title and content are both at least 1 character in length.
* A validation error message is displayed if validation fails.

**General: (1 point each)**

* All scripts include a comment block at the top describing the purpose of the script.
* No PHP/SQL errors are triggered and/or displayed while interacting with your blog.
* *All* PHP code is properly indented when nested.
* All markup must validates as HTML5.

**Appendix A - HTTP Authentication**

Adding authentication to a specific script should be as simple as adding the following to the top of the script:

require 'authenticate.php';

Where the authenticate.php file contains:

<?php

define('ADMIN\_LOGIN','wally');

define('ADMIN\_PASSWORD','mypass');

if (!isset($\_SERVER['PHP\_AUTH\_USER']) || !isset($\_SERVER['PHP\_AUTH\_PW'])

|| ($\_SERVER['PHP\_AUTH\_USER'] != ADMIN\_LOGIN)

|| ($\_SERVER['PHP\_AUTH\_PW'] != ADMIN\_PASSWORD)) {

header('HTTP/1.1 401 Unauthorized');

header('WWW-Authenticate: Basic realm="Our Blog"');

exit("Access Denied: Username and password required.");

}

?>

This script will prompt for a username and password. If the incorrect user/pass is provided it will exit and display an error message. Feel free to change the username and password by modifying the ADMIN\_LOGIN and ADMIN\_PASSWORD constants.

SHUPDITO:

**Home Page: (1 point each)**

* Five most recent blog posts displayed in reverse chronological order.
* For each of these posts you should display: Title, Timestamp, Content
* Blog post titles link to full page for each post. (This link includes a GET parameter to specify the post id.)
* If blog content is larger than 200 characters the displayed content is truncated to 200 characters.
* If the content is truncated a "Read Full Post" link should be displayed after the content. (This link includes a GET parameter to specify the post id.)
* An edit link is displayed for each post. (The link includes a GET parameter to specify the post id.)
* A "New Post" link is present somewhere on this page.

**Full Post Page: (1 point each)**

* Displayed on this page: Post title, timestamp, full post content, edit link
* The blog displayed is determined by a GET parameter in the URL. This parameter should be the post's primary key id from the database table.

**New Post Creation: (1 point each)**

* Provides a form where the user can enter a new post title and contents.
* The form includes a button for submitting the post to the database.
* This page is protected by HTTP authentication.

**Post Update and Delete: (1 point each)**

* Provides a form where the user can edit a specific post title and contents.
* The post being edited is determined by a GET parameter in the URL. This parameter should be the post's primary key id from the database table.
* The title and content of the post being edited should appear in the form.
* The form includes a button for updating the post in the database.
* The form includes a button to delete the current post from the database.
* This page is protected by HTTP authentication.

**Security: (2 points each)**

* Ensure that any id values passed by the user are validated as integers before you use them in a SQL query. This is especially important when updating or deleting a post. If you receive a non-numeric id, redirect the user back to the index page.
* All user submitted strings (POSTed titles and blog content) must be sanitized using input\_filter and inserted/updated using PDO statements with placeholders bound to values.

**Validation and Formatting: (1 point each)**

* All dates should be formatted: "MonthName dd, yyyy, hh:mm am/pm"
  + Hint: You will need a PHP function to format the date. Also check out the strtotime function.
* New / Updated Posts are validated to ensure the title and content are both at least 1 character in length.
* A validation error message is displayed if validation fails.

– 1 NO MESSAGE

**General: (1 point each)**

* All scripts include a comment block at the top describing the purpose of the script.
* No PHP/SQL errors are triggered and/or displayed while interacting with your blog.
* *All* PHP code is properly indented when nested.
* All markup must validates as HTML5.
* -1 HTML ERROR