

CSIS 3280 – Assignment #1 – File-based CMS

This assignment may only be done alone or in pairs (`count($people) == 2`). Assignments giving errors, warnings or informational messages will receive a ZERO mark. Please refer to the assignment submission guidelines for further details.

Solution

The flow of the program is as follows:

- This app will present the user with two workflows. Content Browsing and Content Management
- Your app will work off of a flat-file PSV. It will allow the user to render the content contained in your PSV file against the templating system that you implement.
- You will implement a template based on a downloadable template from oswd.org
- The administrative portion of this assignment will allow the user to do CRUD operations on a flat file in the form of creating, reading, updating and deleting the articles.

Requirements:

1. The app can only read off of one PSV file
2. You may not alter the PSV file structure (any of the headers)
3. The app must immediately update the pages in the application.
4. The app must implement an extensible templating system consisting of at least a header and a footer.
5. You must implement a template from OSWD.org, you may pick any template but the ones with pre-existing navigation are the most suitable. If you pick one without navigation you must implement it yourself and it must be implemented well.
6. The same template must be used when the workflow is Content Management
7. You must pick a template from oswd.org to implement in your application. You may not use the one provided in the example.
8. Your template must support the listing of articles for your CMS, this is a requirement, if the template does not support it you must implement it manually.
9. If there are any input errors or problems with the file the program should notify the user.
10. You must use a Article Class consisting of the same fields in the file (shortname, title, summary, content)
11. CRUD operations to any Article must be implemented using the ArticleServices static class.
12. You are not permitted to put any HTML in a directory other than the 'templates/' directory.
13. All the users actions must access one file only via the web browser.
14. You may not use PHP sessions. Or any super global variables for storage other than `$_SERVER`, `$_POST` and `$_GET`.
15. When submitting your assignment ensure that you place both you and your partner's name and student number in the footer of your template. **Students who's names do not show up in the page footer will not receive any marks for the assignment**
16. **Your program must not display any errors or warnings or information messages.**

Structure

The following structure must be adhered to with regards to the files and particularly the templates

Filename	Description
Assign01_GroupX.php	This is the main CMS file, users use this file to access your CMS. This file should be able to load a default "homepage" as well as any other page requested by the user.
admin.php	This is the main CMS admin page.
inc/ArticleService.class.php	Handles CRUD operations for the Article
inc/IArticleService.class.php	Interface for implementing CRUD operations for the ArticleService
inc/FileService.class.php	Handles file operations and supports ArticleService where required
inc/IFileService.class.php	Interface for implementing FileService Operations
inc/config.inc.php	Stores constants for the application including but not limited to: setting the proper time zone, setting the TEMPLATE_PATH, ARTICLE_FILE
templates/header.php	The header file for your theme
templates/footer.php	The footer file for your theme
templates/*	Any other files that your template requires, you may only have HTML in this folder, you may not have any html in any folder other than this one
templates/admin/	Any template files that are special to the admin interface such as editArticle.php, listArticles.php, and newArticles.php, adminnavigation.php

How the templating works (Page class - NEW)

In our in class examples we will use a static page class, in doing so we can call any part of the template that we require by passing it the data to the methods a static Page class.

```
$article = ArticleService::getArticle("homepage");
$p->new Page();
$p->viewArticle($article);
```

Page.class.php->viewArticle(\$article) might contain: self::header(\$article); self::navigation(\$articles); self::footer();

How the templating works (include files - OLD)

Another approach to the problem of passing data to the page template is if the 'include' keyword is used and the template automatically expects the data is in in scope we can use include files to draw the page.

For example:

```
$article = getArticle("homepage");  
include [TEMPLATE_DIR] ./viewArticle.php
```

viewArticle.php might contain:

```
include 'header.php'  
include 'navigation.php'  
  
<!-- Some HTML... -->  
<?php ... some PHP ?>  
  
include 'footer.php'
```

About the templates/ folder

The 'templates' folder, you may have additional files that are required for your theme such as some JavaScript, images and CSS, this is ok but these files are ONLY allowed in the templates/ folder and nowhere else.

You may extend you theme as needed to add navigation and administrative functions but you may only put those files in the templates/

About the interfaces

You may not alter the interfaces provided in the template.

Submission Guidelines:

As this is an assignment please use the Assignment Submission Naming convention instead of the Lab Submission Guidelines.

Submissions will be graded on a sliding scale according the following grid:

Category	Points	Feedback
Project Submitted and files named properly	1	
Program fully commented including classes, functions, and any other logical operations.	2	
File is parsed into People objects, error handling is implemented with the file parsing.	3	
Program structure is adhered to. Classes are used appropriately, class files include files are used with proper separation of concerns. The one exception is of course the template you implement.	5	

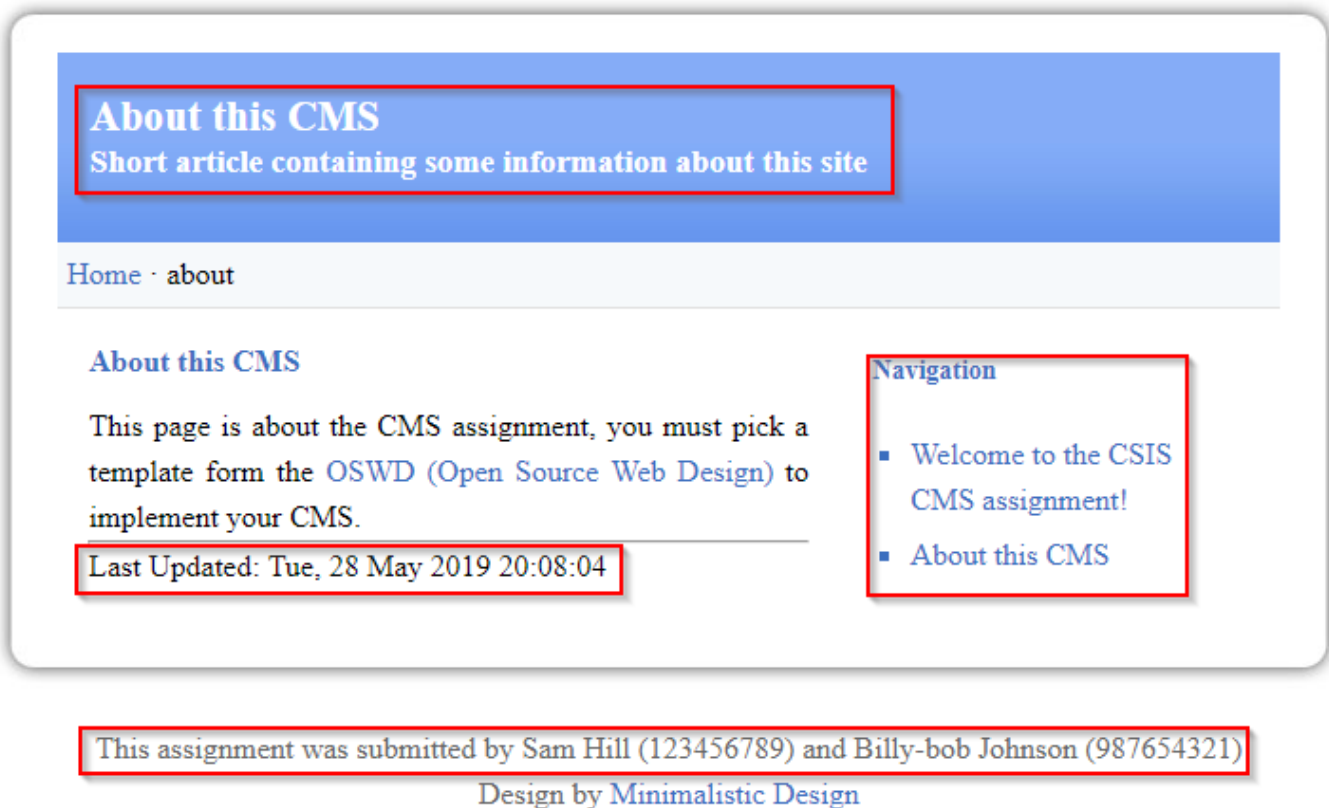
Category	Points	Feedback
Program Interface - Interface is improved upon from the screenshot below, you are permitted to use any HTML framework you would like but you must put a reference in your page class.	1	

User Flow

Article View

Example screenshot of an Article, the items highlighted are required:

- Title and Summary
- Navigation
- Last Updated (must be updated every time the article is saved.)
- Footer with the students details



Admin View

Example screenshot of the admin page, the items highlighted are required:

- Navigation changes to "New Article" or "ListArticles"
- List Articles page allows to edit or delete any article except the home page.
- All Articles must be listed.
- Articles list must automatically update once an operation is complete

Admin - Articles

Page for Editing Articles

[Home](#)

Admin Page

Article Shortname	Title	Edit	Delete
homepage	Welcome to the CSIS CMS assignment!	Edit	Delete
about	About this CMS	Edit	Delete

Admin Menu

Please pick from one of the admin options below:

[List Articles](#)[New Article](#)

This assignment was submitted by Sam Hill (123456789) and Billy-bob Johnson (987654321)

Design by [Minimalistic Design](#)

New/Edit View

- New and Edit View must go back to the main admin view once the operation has completed.

About this CMS

Short article containing some information about this site

[Home](#) · [about](#)

Editing Article: About this CMS

ShortName

Title

Summary

Content

```
<p>This page is about the CMS
assignment, you must pick a template
form the <A HREF="http://oswd.org"
target="_new">OSWD (Open Source Web
Design)</A> to implement your CMS.
```

Admin Menu

Please pick from one of the admin options below:

- [List Articles](#)
- [New Article](#)

This assignment was submitted by Sam Hill (123456789) and Billy-bob Johnson (987654321)

Design by [Minimalistic Design](#)

Submission Guidelines:

As this is an assignment please use the Assignment Submission Naming convention instead of the Lab Submission Guidelines.

Submissions will be graded on a sliding scale according the following grid:

Category	Points	Feedback
Project Submitted and files named properly	1	
Program fully commented including classes, functions, and any other logical operations.	2	
File is parsed into Article objects, error handling is implemented with the file parsing.	3	
Program structure is adhered to. Interfaces are implemented, include files are used with proper separation of concerns	3	
Template is used and citation/credit is kept for original author, HTML only exists in template template layout is well structured	3	

Copyright (c)2018 Rahim Virani and others. NOT FOR REDISTRUBUTION.
STUDENTS FOUND REDISTRUBUTING COURSE MATERIAL WILL BE IN VIOLATION OF ACADEMIC
INTEGRITY POLICIES AND WILL FACE DISCIPLINARY ACTION BY COLLEGE ADMINISTRATION.
