ITMD 462/562

Assignment 2

Introduction

This assignment will introduce the concepts of CRUD operations (create, read, update, delete). We will be making a note taking app and persisting the data to a file instead of a database for this one. The assignment is due Sunday, October 18, 2015 at 11:59pm Chicago time. You will need to make sure you have access to a computer hosting a web server capable of running PHP. A local development environment running XAMPP, WAMP, or MAMP as described and demonstrated in class is needed. Read this document completely and any graduate students must complete the additional graduate student requirements listed within.

Objective

The objective of this assignment is to make a PHP application that allows the user to create, edit, view, or delete a note. The assignment is like a note taking app you may have used on your phone or other devices. There will be a page that lists all the notes, from there the user can view, add, edit, or delete any note. We will not be using a database to store the persistent data. You will need to persist the data using a file on the file system.

Requirements

- Build a note taking application in PHP
- No authentication on this application this time
- Entry point to the application should be the index.php page
- index.php page should be the one to show the list of notes and a button to add a new note
- A note should consist of the following data
    - Note ID (Unique Identifier to use as a key, not user entered/changeable, generated in code)
    - Subject line (Required to save the note)
    - Note Body (Allowed to be empty)
    - Author Name (Required to save the note)
    - Character count of the note body (not user entered, generated in code)
    - Date/time created or last edited (not user entered, generated in code)
- Note data can be modeled as an array or object (graduates must use object)
- List of notes should be stored as an array
- List of notes needs to be saved to a data file by serializing the list or using csv. If you use an array data structure, csv might work fine, if you model the note with an object, serialization would be a better option. Every page load needs to read that data back into a workable data structure. Every time the data is modified you need to save it back to the file. (see graduate extension)

- On the index.php page the list of notes should display the Subject Line, the Date/Time created or last edited, and the number of characters in the note. The subject line should be a hyperlink that will display the full note.
- A page that will display the full note data. This is the page that the note's subject from index.php should link to. Use a GET parameter to determine which note to show.
- The note view page should have a button to delete that note. Do not use GET for this. Delete actions should always be POST. (Hint: you will need a separate form element for only this on the page) It should also have an edit button.
- You need a pages to add/create a note and edit/update an existing note. I am leaving this up to you to decide how you want to handle this.
- The add & edit pages should POST to themselves and then show you the results if valid and have a link back to the song list. If it is not valid show the form again with error messages. This part is very similar to AS1.
- Make sure you validate required fields in PHP and do not allow the user to inject HTML/CSS/JavaScript into any form field. Remove all leading or trailing whitespace in any form field. Sanitize inputs.

Graduate Additional Requirements

If you are involved in **any section of 562** you need to complete the additional requirements listed here.

- You must model the note data as a custom Class/Object.
- You need to save the data to the file by serializing the list of note objects to a file and unserializing the data when reading.
- Delete note action needs to have a confirmation page before deletion.
- Namespace your classes using your userName as vendor and as2 as package (\userName\as2)
- Use some HTML/CSS to present the application nicely (bootstrap or other CSS framework is ok)

README File

Produce a **PDF README** file in your project zip that describes some components of this assignment. The readme doesn't have to be long but should include a few things.

- Your Name
- Your Email
- The Class and Assignment Numbers
- The following sections
    - **Project Description** – Brief introduction describing the project in your own words.
    - **Development Environment** – Description of your development environment. Include things like computer OS, what you are using for your server stack, PHP

version number, editors used, and any other details you think you should include about your environment
- **Installation Instructions** – Include any instructions to us if we need them to install or configure your project.
- **Insights and Results** – Include here things you learned, challenges or problems you faced, and any details about the results you received. If you couldn't get something working this would be the place to describe what you tried and what wouldn't work. Feel free to embed screenshots for working and non-working features.
- **References** – Any resources you need to cite of that you might have used for assistance. Not an excuse to copy and paste but if you need to cite a source, do it here.

Due Date / Late Policy

This assignment is due **Sunday October 18, 2015 11:59 PM Chicago Time**. Late assignments will receive a **10% per day deduction starting at 1 minute late**. We will discuss this in the following class so **no assignments will be accepted as of class (6:00pm) that night**. See syllabus for full late policy. **No Extensions**.

Submission Guidelines

You must upload your submission, including source and all supporting files, as a zip archive to the blackboard assignment by the due date. The submission must be in the following format and structure. If you do not submit your assignment exactly as specified, you will receive a immediate 5% deduction.

Submission Format Specification:

YourUsername_YourIdA#_AS2.zip
|----------src *(folder)*
| **|----------index.php**
| **|----------***All other documents and folders as required*
|
|----------README.pdf *(describes your submission per above)*

Hints

You will need to use the Note ID to determine what note to show, edit, or delete. Pass it around in all form controls or links. In links it would be a GET parameter. On forms it can be a hidden form control.

PHP hash() or uniqid() functions could be used to generate a unique id based on time created.