Assignment 1

Description: This assignment will test your ability to intermix php and html as well as produce html from php. You are tasked with producing a 10 x 10 html table of randomly selected colors using php.

Requirements Your grade will reflect how well I think your script meets the following requirements:

This assignment is due by the beginning of the first class session of week 2. Late work will be penalized 5 pts each week or portion of a week that it is overdue.

The script must run without a fatal error.

The table must have ten columns and ten rows produced using nested for loops.

Colors should be randomly produced and expressed as hexadecimal color codes and not integer or text values.

The html table must be constructed using php loops and not static html.

The background color of each cell will be the same as the color selected and set using proper inline css.

The cell shall display the hexadecimal color code being represented by the cell background as both white and black text so that the color code may be read no matter the lightness or darkness of the background color.

All variables must be appropriately named with descriptive identifiers. For example, your row loop counter should be named something like $row and not $r.

The produced html should be ‘standard’ html. (Paragraphs must be properly closed, there should be a head and a body, “empty” tags like bb must be properly expressed   
etc.)

The script should reside in a subfolder of xampp/htdocs to be named a1.

The script itself shall be named index.php

Assignment 2

Before the beginning of the next class you need to complete the following work in phpMyAdmin:

1. Create a database called **music**
2. Create a dbo account for **music** to be called **music** and with a password of **password**
3. Create a crud account for **music** to be called **musicphp** with a password of **password**
4. Create a **songs** table in the music database with the following fields. Choose appropriate datatypes unless specified

* **id** (this is the primary key field. It should be an INT, Index should be PRIMARY and A\_I should have a check mark.
* **name**  (this is the name of a song) Use a varchar of appropriate length
* **artist** (this is the name of the artist. It isn't first and last names because that doesn't for bands like Pink Floyd or The Cure. in reality, it would be a foreign key reference to a different table.) Use a varchar of appropriate length
* **running\_time** (this is the length of the song expressed as seconds, so use medium int)
* **lyrics** (this is the song lyrics and could be of some length, so a text field.)
* **rating** (this is your rating of the song, so a fractional of some type so that you can store something like 2.5)
* Create some sample data. At least two records worth.
* Export the database, including all tables and data and upload the resulting sql file here.

So, using the same techniques discussed in class, create a MVC web application that will present a list of the songs by title as hypertext links. When the user clicks a link, all of the details about that song should be added to the page. You should wind up with a **songs** subfolder in htdocs. songs should have subfolders of **models** and **views** with the appropriate files allocated to them. Make sure you export your sql from phpmyadmin and include that .sql file in the songs folder. Then, zip the songs folder (songs, not htdocs) and upload the zip to me as a new submission for assignment 2.

Points will be deducted if you do not use appropriate file and/or folder names, appropriate variable or function names, if you forget to include the sql export in the zip, if it's late and if running results in errors or your resulting pages are not properly structured html.

Assignment 3

Modify the songs application to allow for adding new songs, deleting songs, and editing songs. Due before the start of class on 4 Feb.

Do not forget to include the exported db as a sql file.

Assignment 4

Modify the songs app so that a user must authenticate before being allowed to edit, delete or add new songs

Assignment 5

Create a script (in a folder cc) that solicits a credit card number, name of cardholder, exp. date, ccv verification number and type from a user and stores that information in a table called **ccData**(just go ahead and put it in the music database) as the fields **ccNum, ccName, ccExpDate, ccCCV, ccType** if and only if the card number passes validity checks using pcre regular expressions (preg\_match and preg\_replace). Note, you should use preg\_replace to get rid of any non-digit characters in the cc number first!

Look up how to use checkdate( ) and use it to reject invalid expiration dates. The tricky part here is you've got to remember that credit cards don't always include the day of the month in their expiration (for example, 10/2016) but they always expire on the last day of the month.

The criteria for valid cards is as follows:

**American Express**

First digit must be a 3 and second digit must be a 4 or 7. Valid length: 15 digits.

**Diners Club and Carte Blanche**

First digit must be 3 and second digit must be 0, 6, or 8. Valid length: 14 digits.

**Discover  (5% EXTRA CREDIT)**

Valid length: 16 digits. First 8 digits must be in one of the following ranges:

60110000 through 60119999

65000000 through 65999999

62212600 through 62292599

**enRoute**

First four digits must be 2014 or 2149. Valid length: 15 digits.

**JCB (5% EXTRA CREDIT)**

First four digits must be 3088, 3096, 3112, 3158, 3337, or the first eight digits must be in the range 35280000 through 35899999. Valid length: 16 digits.

**MasterCard**

First digit must be a 5 and second digit must be in the range 1 through 5 inclusive. Valid length: 16 digits.

**Visa**

First digit must be a 4. Valid length: 13 or 16 digits.

Final Project

You will submit (and demonstrate to your instructor) a final project on week 10. This final project will have the same weight as any other assignment in your grades. It is intended to be a culmination of everything related to PHP that you have learned so far.

Requirements:

-There must be at least two tables that are related.

-An administrators table with the user names and passwords of site admins can exist or that can be folded into a Users table with an admin boolean field if you need a Users table.

-There must be a public interface as well as a private password protected (using sessions) interface which is to be used for editing table data.

-There must be an admin userid/password combination of clark (userid) and LLaP7of9 (pwd). -You are to do you own work.

-The beautification of the site via images and css is not required.

Possible Extra Credits (25% max): More tables than the minimum. Allowing for file uploads. Hashing the passwords. Search capabilities. Basically, anything that I feel goes above and beyond the requirements, at my discretion.