**Assignment 2 – Intro to Web Programming**

**Option B – Max Grade 100%**

**Content Management System**

This assignment requires you to insert and select data from a database, validate user inputs and send email using PHP scripting. Your mark counts for 30% of your final grade.

**Submission Dates & Details**

**Due: Tue Apr 13 @ 5:00 pm**

* Submit a link to your new private GitHub repository and invite me – ifotn
* Make a minimum of 4 descriptively named commits spread over 2 dates minimum. You need to use the git commands rather than direct upload.
* Submit a link to your live web site on AWS (<https://lamp.computerstudi.es/~xxx/folder/file.php>)
* Create a README file that:
  + Explains the purpose of your application
  + Includes a link to the live web site running on AWS

**All** work must be your own. Failure to submit an independent assignment will result in a minimum penalty of grade of zero. You may use my GitHub code as a **reference only** but cannot download the code, modify it and submit as your own assignment. All submissions are subject to an in-person code review - you may be asked to walk through and explain your code to help ensure it is entirely your independent work.

YOU MAY NOT MODIFY YOUR REPOSITORY OR SITE AFTER THE FINAL DUE DATE OF TUE APR 13 @ 5:00 pm.

**Project Overiew:**

# Content Management Site - you are making a "web site builder" (similar to WordPress) that allows users to manage their own web site content and navigation.

Video Walkthrough of Sample Application:

<https://www.youtube.com/watch?v=5Hmp-9OnA9Y>

In this assignment you will use PHP, HTML, and SQL to:

* Build a password protected admin site
* Allow admin users to add, edit, select, and delete web page records in a database
* Create a public web site where the page links and content are displayed
* Document your code with comments within your scripts
* Incorporate a basic CSS-based site design

**Application Requirements:**

1. Document each section of your PHP scripts with comments. You do not need to document every single line.
2. Build a site control panel with pages to:
   1. Build a register page to allow people to sign up as admin users of your site.
      1. Passwords must be hashed.
      2. All inputs must be validated.
      3. Usernames should be validated as properly-formatted email addresses.
      4. 2 users cannot sign up with the same email address
   2. Log in / log out of your admin site
3. List registered users of the admin site
4. Edit a new admin user
5. Delete an admin user, preceded by a confirmation message
6. Ability to list / add / edit / delete web pages from your database. Each page needs a title and content. \*\*\* NOTE: each "page" is really just a record in your database \*\*\*
7. All these pages must be password-protected
8. Area to upload a site logo which should appear in the header of all admin and public pages
9. Create a public web site that includes:
   1. A common header. Header must include a navigation bar that is populated by the Page Names listed in your database (this is managed through your site admin panel. So if I add a new page or change a page name in the admin panel it will update the public navigation bar automatically.)
   2. A content-managed site so that the only page that loads is called “index.php”. This page takes a querystring parameter that indicates which page ID is being requested. The page title, and content will then be selected from the database based on the page ID. If a page is edited in the admin panel the changes should automatically be reflected on the public site. So the links in your navigation bar may look like this:
      1. Home (links to index.php?id=1)
      2. About (links to index.php?id=2)
      3. Services (links to index.php?id=3)
      4. Contact (links to index.php?id=4)
   3. Your site needs to include at least 5 public pages which are created and managed through your site control panel.
   4. Page content which you will populate in your site control panel can be fairly be basic but should reflect content of a typical business. The business type is up to you and can be fictitious.
10. Implement CSS, either your own, Bootstrap, or a CSS or Bootstrap theme that you find online. If you use a 3rd-party CSS, you must document where you found it in your README.md file on GitHub as well as in your header.php file where you will link to it. If you use Bootstrap you must change the styles from the ones we have used for our in-class application. If you use all the same css classes so both sites look exactly the same you will not receive marks for this part of the project.
11. Implement proper error handling:
    1. 404 error page with the correct .htaccess file
    2. General error page
    3. Try Catch blocks on all of your code blocks that access the database. If errors occur the user should be redirected to your General error page

**Evaluation Method**

Your work will be evaluated based on how your PHP scripts perform when I load and try your web pages on the AWS server.

# Evaluation Criteria (max 32 marks + 2 possible bonus marks)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **0-2** | **3-4** | **5-6** | **7-8** | **Marks** |
| **Authentication** | - No Register, Login, or Security | - Partly complete security model | - Mostly complete security model | - Full Registration & Login, all admin pages secured | **8** |
| **User Administration** | - No functioning CRUD operations for users | - Some functioning RUD operations for users | - Most functioning RUD operations for users | - All functioning CRUD operations for users | **4** |
| **Site Control Panel** | - No functioning CRUD operations for pages | - Some functioning CRUD operations for pages | - Most functioning CRUD operations for pages | - Fully secured functioning CRUD operations for pages | **4** |
| Public Content Managed Pages | - No common elements and content managed pages functionality working | - Some common elements and content managed pages functionality working | - Most common elements and content managed pages functionality working | - All common elements and content managed pages working | **4** |
| **Site Logo Upload / Display** | * Not implemented | * - Attempted but incomplete | * Mostly complete image upload + header display | * Complete image upload in Control Panel + Display in Header | **4** |
| **Error Handling** | * No error handling | * - Some error handling | * Most error handling | * Complete error handling with full use of try catch and bound parameters statements + 404 | **4** |
| Visual Appeal | -Colour scheme is evident;  -Font is inconsistent, not properly sized or does not stand out against background | -Colour scheme is somewhat; Font is mostly consistent, properly sized  and colour stands out fairly well against background suitable | -Colour scheme is appropriate; Font is consistent, properly sized  and colour stands out against background | -Colour scheme is appealing; Font is consistent, properly sized  and colour stands out against background. Site looks different from in-class app. Any 3rd party CSS is documented. | **2** |
| **Deployment** | No deployment | Partly working deployment | Deployed to GitHsub but not all requirements met | Min 4 commits to private repo, README file & server deployment | **2** |
| **\*\*\* WILD-CARD BONUS \*\*\*** |  |  |  | Additional working functionality of your choosing | **2** |