

# Project 3: Image Album

## Introduction

Project 3 is designed to give you practice building and implementing a database backed website from scratch. As you may have encountered in Project 2, text files can be very difficult to open, read, and write. Databases provide a clean, easy alternative for storing and organizing information. Your task is to develop a personal online image album, using PHP to interact with a MySQL database you will create and populate yourself.

## GRADING SCHEME

The project has been divided into 3 milestones, with an additional Code Review. The weighted average of the 4 parts will be your final grade for this project. **WOW points will only be given for Milestone 3.**

[Milestone 1](#): Due Mar.10th, 15% of final grade

[Milestone 2](#): Due Mar.17th, 15% of final grade

[Milestone 3](#): Due Mar.24th, 65% of final grade

[Code Review](#): Starting Mar. 25th, 5% of final grade

## IMPORTANT NOTES

You will find code online using the [mysql\\_connect functions](#) but they are deprecated and you should not use them. Use the mysqli functions as taught in class. **If there is any appearance of these functions in your code, 10 points will be taken off your grade for that milestone.**

If you use file functions, make sure your files are open to the public for read, write, and execute. **If we receive a file permission error during grading and any part fails to function properly because you will receive a zero for that part--we will no longer email you to fix your permissions.**

Please start early and make sure you understand every part of your code, as you will need to explain everything you did during the Code Review.

## **Milestone 1 - Initial Design (Due Mar.10th)**

### **REQUIREMENTS**

#### **Create the relational schema for your database.**

- Create a list of all the table information you need to make your site functional.
- Write your schema in a manner which, for each table, shows field names, types (string, integer, date, ...), size (if string), allow nulls, primary key) .For example:

table1

name	type	size	allow null	primary key
field_1	int		false	true
field_2	string	32	false	false

- Upload your text document to CMS as **P3\_1.pdf**. Also upload a backup copy of P3\_1.pdf to the course server.

*Note: Everything else will come much easier if you have a good understanding about what tables you need, and what the attributes / fields in each table should be. More fields are better than too few at this point – you can revise the schema in Milestone 2.*

#### **Use MySQL to create and fill tables in the database matching your relational schema.**

- Use your info230\_SP15\_username database on the 2300 server for your tables.
- Fill your database with at least 1 album and 3 images linked to the album.

#### **Create basic skeleton for your site.**

- Set up the pages that you think you will need in your website.
- On each page, include a title and brief description of what will go on that page.
  - e.g. home page, display all albums, display images from specific album, display login form
- Link your pages with a clear, consistent navigation. There should not be any dead-end pages on your site that require the user to click the “back” button in order to continue navigation.

#### **Implement a View Full Album feature to display all images from an album on one page.**

- Users should be able to select an album and view all of its corresponding images at once (either as thumbnails or at full size).
- Implement this by accessing the database dynamically using PHP (don't hardcode it).
- Note that no other functionality is required for this milestone.

## Create basic styles for your site with CSS

- Make sure that the pages are user friendly and aesthetically pleasing.
- Consider your layout, colors, and fonts.
- Does not need to be a fully polished product and can be changed later, but note that the more design you accomplish now, the less you will have to do in later functionality-heavy milestones.

## WHAT TO SUBMIT

- Relational Schema to CMS as P3\_1.pdf.
- Relational Schema to course server as P3\_1.pdf.
- Implemented tables to course database.
- Basic website to course server inside P3 folder.

## GRADING RUBRIC

- Initial schema (\_\_/40)
  - Workable schema uploaded to CMS in P3\_1.pdf (\_\_/10)
  - Implementation of the described schema in MySQL on the INFO 2300 server (\_\_/20)
  - Album and Image tables exist and have the correct type of database relationship? (\_\_/6)
  - Tables are populated with at least one album and three images (\_\_/4)
- Initial functionality (\_\_/40):
  - Does the code work and display images from the album in the database? (\_\_/20)
    - 100% works - 20pts; few errors - 15pts; somewhat works but many errors - 10pts; does not work but code present - 3pts.
  - Does it do so via PHP calls to the database?(\_\_/20)
- Initial design and navigability of pages (\_\_/20)
  - Does the site have functional navigation? (\_\_/10)
  - Is the site navigation consistent and easy to use? (\_\_/3)
  - Is the site styled appropriately (all content is legible and easy to read, uses appropriate color, typography, layout, and positioning)? (\_\_/7)

-10pts if [mysql\\_connect functions](#) is used.

## **Milestone 2 - More Functionality (Due Mar.17th)**

### **REQUIREMENTS**

**Add the code for the following additional functionalities on your site:**

1. Display all albums (and not the images in them) on the albums page (at least 2 albums).
2. Add a new album.
3. Display all the images in a single album (at least 3 images in album). Users should be able to select which album they want to view and only see photos from the chosen album.
4. Upload an image and add it to zero or more albums.

**Upload an updated relational schema to CMS as P3\_2.pdf.**

- Include schema from P3\_1.pdf in this file if changes were made.

### **WHAT TO SUBMIT**

- Relational Schema to CMS as P3\_2.pdf.
- Relational Schema to course server as P3\_2.pdf.
- Updated website to course server under P3.

### **GRADING RUBRIC**

P3\_2.pdf in CMS (\_/5)

Display all albums on the albums page (\_/10)

- Page has at least 2 albums listed (\_/1)
- Albums are displayed by correctly accessing the database dynamically using PHP (not hardcoded) (\_/5)
- Page works and albums display without errors (\_/4)

Add a new album (\_/30)

- The form allows user to submit successfully and functions without error (\_/5)
- The new album is added by correctly accessing the database dynamically using PHP (not hardcoded) (\_/15)
- The new album is displayed on the albums page without errors and persists when page is closed and reopened (\_/10)

Display a single album and the images in it (\_/25)

- Each album contains at least 3 images (\_/2)
- Images are displayed by correctly accessing the database dynamically using PHP (not hardcoded) (\_/13)

- Page works and images display without errors (\_/10)

Upload an image and add it to zero or more albums (\_/30)

- An image can be uploaded successfully without errors (\_/10)
- The image can be added to an album and displays on the page of that album after being added (\_/10)
- The image persists when website is closed and reopened (\_/10)

-10pts if [mysql\\_connect functions](#) is used.

## **Milestone 3 - The Finished Product (Due Mar.24th)**

### **REQUIREMENTS**

In Milestone 1, you started the initial design and created the initial database and code to access the database. In Milestone 2, you continued building the functionality to display and add albums, and to add images to albums. In this Milestone you will polish your design and complete all remaining required functionality of the project (e.g. logins, editing, and search features).

**Make sure your completed site includes all of the following elements:**

#### Albums

- Functionality to create, update, and delete albums without any limitations on the number of albums on the site.
- Each album must have a title, date created, and date last modified (last time that album entry was modified).
- At least 3 albums and 8 images in each album in the database

#### Images

- Functionality to upload, add to album, update, and delete images without any limitations on the number of images on the site.
  - Multiple image upload and resizing are not required.
- When added, images should be uploaded to the server with the URL of the image stored in the database.
- Each image must have a unique URL, date taken, and caption.
- A single image can be used within zero or more albums and must persist even when not in any albums.

#### Other Features

- The site should support Admin Mode (login for add, edit, and delete privileges) and Viewer Mode (no editing privileges, just allowed to view albums/images).
  - Note: Do not need to include functionality to add a new user.
- Search functionality on image captions
  - individual words, not just entire captions
  - Results should show matching images and the albums to which they belong.
- Form validation. Rigorous SQL injection prevention is not required, but include appropriate string checking and other validation methods.
- Creative and rational design.

## **Submit a Design Rationale to CMS as rationale.pdf.**

In the Design Rationale:

- Provide your **login username and password**.
- Explain your aesthetic design choices (e.g. color, layout, navigation, theme).
- Explain the functionality design choices (e.g. how pages are implemented clearly, where functionality is placed, additional functionality added).
- Explain improvements from milestone feedback (either how feedback was implemented or why it was not).
- Explain any WOW features you implemented.

## **WHAT TO SUBMIT**

- Design Rationale to CMS as rationale.pdf.
- Design Rationale to course server as rationale.pdf.
- Completed website to course server under P3.

## **GRADING RUBRIC**

### **Design - (\_\_/18)**

- Content is legible, easy to read and understand (\_\_/3)
- Navigation is easy to use and no dangling pages (\_\_/3)
- Cross-browser Compatibility: displays reasonably well across Safari, Firefox, and Chrome (\_\_/3)
- Appropriate color, typography, layout, and positioning for the chosen theme (\_\_/5)
- Pleasant to look at; Creative and interesting (\_\_/4)

### **Design Rationale - (\_\_/23)**

- Provided login username and password for the website and the database (\_\_/4)
- Explanation of Design Choices (e.g., the choice of color, layout, nav, and chosen theme) (\_\_/5)
- Explanations on functionality design (How album pages are implemented cleanly, additional functionality added etc.) (\_\_/5)
- Explanation on improvements from milestone feedback (\_\_/5)
- Rationale should be in pdf format ( \_\_/4)

### **PHP and MySQL Functionality (\_\_/45) - specify the search**

- Uses MySQL for a table structure and accesses this dynamically using PHP (\_\_/5)
- Capability to display all albums in the database and all images within an album (\_\_/5)
- Clicking on an individual image should display the original image and its caption (\_\_/5)
- Your site should include forms to do the following:
  - Add a new album (\_\_/2)
  - Add a new image to an album (\_\_/2)
  - Images should be able to be uploaded to the server from the user's file browser (\_\_/3)
  - Update album/image information (\_\_/5)

- Add and Update Forms only visible to a single administrative user who logs into the site via a username and password (\_\_/5)
- Site should include a form to search on the text in captions, title, album etc. this form is visible to all users (search should include partial search; should be able to search across multiple fields) (\_\_/5)
- Error checking for malicious and incorrect input should be done on all forms with appropriate error messages (\_\_/5)
- Login system must utilize sessions as discussed in class (the password should be hashed) (\_\_/3)

### **Content (\_\_/9)**

- All album entry in the database includes a title, date created and date last modified (\_\_/5)  
(date last modified worth 2; others 1)
- A image entry in the database should include a caption, image url, and date taken (\_\_/2)
- Database contains at least (3) different albums with at least (8) images in each (\_\_/2)

### **Organization (\_\_/5)**

- readable and clear PHP, HTML, and CSS with comments in appropriate places(\_\_/2)
- HTML validates (for complete pages, not for include() or require() pages) (\_\_/1)
- pages are well organized in server directories (images are in an img folder, etc) (\_\_/2)

-10pts if [mysql\\_connect functions](#) is used.

### **Wow Points (up to 5)**

This is reserved for websites that go above and beyond the requirements. These 5 points will be applied to other parts of this assignment where you may have lost points, but will not take your overall score for this project over 100. Elements of wow factor include, but are not limited to:

- Awesome design
- Excellent use of JavaScript, jQuery, or Ajax to simplify user interactions
- Thorough and thoughtful implementation of search and user input checking
- Interesting features beyond those required
- Display images not in any album



## **Code Review (Starting March 24th)**

Following the submission of Project 3, each student will participate in a code review. The purpose of the review is to show off, explain, and answer questions about your project.

Students will be asked to sign up for a 15 minute slot via CMS with one of the course staff. Sign ups will begin March 17th, and will continue until the project due date, and will be scheduled on a first come, first serve basis, so make sure you sign up early! Code reviews will begin March 25th, and will continue the week of April 6th (Projects are due March 24th, and the following week is Spring break).

**What to Expect:** Students will be asked to demo their sites to the TA, making sure to cover all of the core functionalities. Students should be prepared to answer general questions about course material relevant to the third project, as well as more specific questions regarding their individual implementations. Note: there is no advantage in trying to hide flaws or bugs. Instead, it would be best to explain why issues occur, as well as what steps were taken to attempt to address them.