Lab 7: PHP Image Gallery

Overview: Based on the lessons in PHP images and Uploads, build an **Image Gallery** application. This will allow an authenticated administrator to upload an **image file**, enter a **Title** and **Description** (and possibly other info) for this image, and then the image will have both thumbnail view and display-size view available for all users.

Note:

- *Any images provided by the instructor during the lessons in PHP Image Methods or Uploads cannot be used as content in this lab. Each student must resource their own images as content.*
- *Student are advised to make sure that Maximum file upload size has been increased on their web accounts.*

1. Admin section functionality :

   3 marks
   - Insert page shows form fields for Title, Description, and File Upload.
   - Insert: Must upload an image and create a thumbnail (recommended to create a few thumbnails of varying sizes) and display size image.
   - Secured by a sessions-based security login.
   - Edit/delete are NOT necessary in this app, but can be done for Challenge marks.

2. Public output: What the public sees. A very well styled image gallery with both a thumbnail view and some sort of display-photo view (with ALL info such as description).

   2 marks
   - Thumbnails View: You may show a title with thumbnails, but description not necessary for thumbnail view.
   - Display View: If user clicks a thumbnail, then the full-size display image is shown in a new page with all information (title, description, etc.)

3. Content/Usability: *2 marks.*
   - Content: At least 15 images (photos) uploaded with good titles and descriptions. Highly recommended that you use your own photos assuming you own a camera.
     - Feel free to have a theme for your Gallery (example: Halloween Party photos).
     - Any use of the testing images provided by the instructor will result in losing a mark. It is up to the student to provide images to use in their gallery.
   - Usability: Marks deducted for poor usability practices such as dead links, poor validation, poor readability, etc.

4. Special features: *3 marks total* - Research and figure out any of the following yourself. *1 mark for each feature (unless specified otherwise).*

   - Watermark all fullsize display images.
     - Note: This must be done on the display image as it is uploaded and resized from your insert page, NOT when it is output to the user.
     - Please create your own tasteful watermark. Do NOT use the one provided by the instructor in a previous lesson.

- Enable PNG uploads as well as JPG. You would have to detect the filetype (easy) and then perhaps create another function for resizeImage().
    - If you're using the watermark as a feature, then you will need to redo that to accept PNG's as well.
- Incorporate a tags feature WITH linked tags that filter the results.
- Incorporate an Edit/Delete page in admin. This should allow the admin to edit title/description and possibly upload a new image.
    - Be careful here. Make sure that if this row is updated with uploading a new image, the existing image filename is not overwritten with a blank value!
    - It is acceptable to allow only editing of title and description without being able to edit the actual image file.
- EXIF data: Have this extracted from photos, and then entered into your DB.
- Create *Next | Previous* links that scroll through all photos in the display page.
    - This MUST check the DB to make sure the ID exists first. You will need two additional separate queries for these.
    - No marks if you create a simple PHP script that adds or subtracts from the previous ID. This will NOT work if images are deleted from the DB.
- Create a unique name for each uploaded image instead of using the default desktop name. This will eliminate the possibility of filename conflicts. Try `uniqid()`.
- Anything else you think might be cool if you ask the instructor about it first.

Step-by-Step walk-though of the basics:

- Create an /admin/ subfolder for all secured admin files, and an index.php file (in the root) for all users to see.
- Create a MySQL table that allows for an image filename, a title, and a description to be stored for each image.
- Create an insert.php file with a form that allows for a browse/file for image upload, a title, and a description. Please remember the enctype="multipart/form-data" attribute.
- After form is submitted, image file type & size are validated, title and description validated (up to you), an image is uploaded, thumbnail versions created, and all data is put into the DB.
- To accommodate larger images easily, try creating another folder called /display/ and then calling the resizeImage() again, but this time outputting a second copy image at a greater size to that folder. The /display/ image will then be the ones shown after user clicks a thumbnail, NOT the /originals/.
  resizeImage ($thisFileName, "thumbs/",100);
  resizeImage ($thisFileName, "display/", 800);
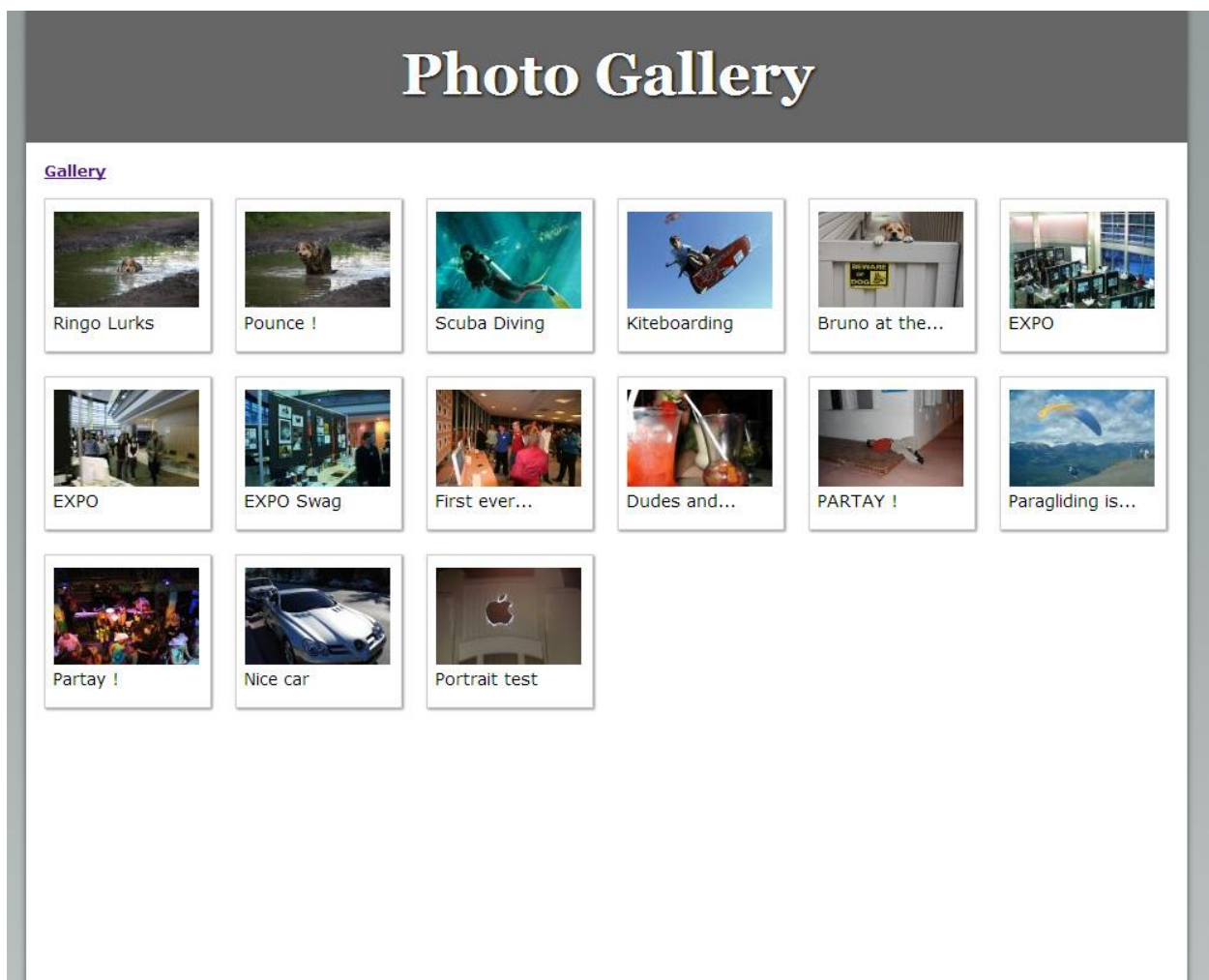- In the gallery file (index.php), output all data from the database, and create tags for all info.

```
<div class="thumb">
<div class="title">This image title</div>
<div class="image"><a href="detail.php?id=17"><img
src="thumbs/pelican.jpg " /></a></div>
```

- Style the output of your thumbnails. For best results, try *float:left* on all your parent divs that contain both title and the image. Here is some suggested CSS formatting:

```
.thumb {
        float:left;
        width: 120px;
        height: 120px;
        border: 1px solid #ccc;
        margin: 10px;
        padding:3px;
```

- Link to a *display page* that shows the display size image plus other info similar to the companyProfile view we used in the Contacts Database.
- Please make sure you have the basic input/output of this application working before you move on. The instructor will assist during work classes.
- Figure the rest out yourself. Enjoy!

index.php: Your public output of thumbnails.

DMIT2025 – PHP/MySQL

New page for display image: Clicking a thumbnail redirects to a new page with the display image and all info.