DMIT2025 – Catalogue Project

The Catalogue Project is the final project for this course. In it, you will create a simple PHP/MySQL application that catalogues a collection of similar items.

It is designed to give you a degree of freedom in choosing your topic, the layout of the application, and the coding challenges. This project is meant to look different from what your classmates create so that you can have a portfolio piece that stands out.

Milestone Breakdown

This project consists of three parts:

Submission	Overall Weight
Planning Document Students will write a brief outline that describes their back-end infrastructure, including their main table(s) and the columns it requires. They will state their topic, what pages they will need, their overall directory structure, and the features on each page.	5%
Milestone 1 (Image Application) Students will complete a proof-of-concept for their catalogue, including a simple CRUD application that handles images.	10%
Final Catalogue Project The final submission will build upon the previous lab, including an administrative login, a search feature, and several challenges.	10%

Each milestone will have its own submission requirements and due date.

Topics

You may choose any number of topics including, but not limited to, the following:

Hobbies

- Art supplies
- Birdwatching guide
- Gardening catalogue
- Modelling and miniatures
- Woodworking projects

Science and Nature

- Animal species (including habitats, scientific classification, etc.)
- Astronomical objects

Literature and Art

- Famous authors
- Art movements
- Filmography (i.e. a collection of films related to a shared thing)
- Fictional worlds or locations

History & Culture

- Historical events (in a specific region or era)
- Cultural artefacts
- Famous paintings (belonging to various schools of art)
- Historical structures and architecture

Food & Cooking

- Culinary herbs, spices, and common ingredients
- Recipes (categorised by cuisine, difficulty level, etc.)

Travel

- Famous landmarks
- Hiking trails in a specific region
- Things to see and do (in a city other than Edmonton)
- Places that you've visited (or would like to)

You **may not** select a topic that has already been used for in-class demonstrations and your previous Labs. If you are unsure about a topic, speak with your instructor before beginning your planning document.

Data Requirements

Each item or entry into your catalogue must include the following information:

1. An **image**. This image must be uploaded to the server via a web form and have various associated metadata stored in the database, such as the filename.
2. A **title**. This is the name of the specific thing being catalogued.
3. A **description**. The description must be at least one complete sentence, but may be longer.
4. **Categories**. The topic of choice will dictate what these categories are.

In addition to these columns, you must have **6-8 additional** columns, for a minimum **total of 10-12 columns** of data.

When designing this table, keep in mind the features of your website that will use them, such as searches and filters.

A Note on Images

Your images must be **copyright free**, meaning that they are images you took yourself or images from a licence-free stock photo repository.

Some free stock photo repositories include:

- Pexels: https://www.pexels.com/
- Unsplash: https://unsplash.com/
- Pixabay: https://pixabay.com/

You may not use images with a watermark or that you do not have permission to use.

Application Requirements & Functionality

Admin Section Specifications

Access Control:

- The administrative (admin) section must be secured. This means that only an authenticated (logged in) administrator can access it.
- Authentication will be managed through a table named `catalogue_admin` with two specific entries – one for the developer (you) and one for the instructor (me).

The plain text username and password for the instructor will be:

> user: instructor
> pass: Password2!

Note that you must **never store a plain text password**. To generate a hash to insert into your new table, either echo out a generated hash and copy the output, or use an online encryption tool, such as this one: https://www.browserling.com/tools/bcrypt

Admin Area Functionality:

1. **Insert a New Item:**
   - Admins must be able to insert new items into the catalogue.
   - Each new item must meet specific validation rules to ensure all required information is correct.
2. **Upload an Image:**
   - Admins must be able to upload a single image to represent each item. This image must then be used to create a thumbnail and a full-sized (720px) version.
3. **Edit an Existing Item:**
   - Admins must be able to edit any existing item in the catalogue.
   - When the admin edits an entry, they will be given a form that is similar to the form they are given when adding an item. This form will be prepopulated with all of the existing information currently in the database.
4. **Delete an Item:**
   - Admins must be able to delete any item from the catalogue.
   - Before deleting an item, they must be prompted for confirmation. This can be done with JavaScript, a modal window, a separate page, or by altering the delete button after the first click (effectively making them click it twice).

Public Display Specifications

Introductory Homepage:

- The homepage must explain what the project is about.
- It must include a couple of paragraphs of original content describing the project.
- Feature boxes can be used to link to specific categories or items, providing a visual and interactive experience.

Main Display Page (Browse):

- Must show all items within the catalogue; however, not all information will be shown at once. Instead, the user will be provided with the option to view more, which will take them to the Single Item View Page.
- Must include **filtering options**, allowing users to narrow results by any 3 factors relevant to the project. Examples include filtering by genre, year, decade, etc.
- Must include a **search feature** enabling users to find specific items.
- Thumbnails must be used for list views, and a larger image must be displayed for a single item view.

Single Item View Page:

- If a user selects an item from a listing, they must be able to see that one item with all related information.
- Full-size image, description, and all other relevant details must be displayed.

Graphical Output and Query Usage:

- All queries (except for simple SELECT queries) must use prepared statements.
- The implementation should allow for efficient filtering or presentation of content.

Additional Challenges

In addition to the requirements outlined above, you must complete at least three (3) of the following challenges.

Profile Photo

Objective: Allow registered users to upload and display a profile photo.

Requirements:

- Provide an option for users to upload a profile photo within their account settings.
- Validate the uploaded file to ensure it's an image format (e.g., JPG, PNG) and meets size constraints.
- Choose where to display the photo. This might include creating a profile page for each user, or in each item that they create in the catalogue (ex. including a 'posted by' area at the top of each item).

Hints:

Consider assigning a default profile photo to each user until they update it.

Adding which user posted an item and displaying their profile photo will require an INNER JOIN, else grabbing the data from the $_SESSION superglobal, and storing which user last uploaded or edited an item in a separate column.

Added By / Last Edited

Objective: Track who added an item and when it was last edited.

Requirements:

- Record the username or ID of the admin who added the item.
- Store the date and time when the item was added.
- Track and display the username or ID and the date/time of the last person who edited the item.
- Show this information in the admin interface for transparency.

Hints:

Just like the previous challenge, you will need to capture the logged-in admin's username or ID during the item creation and edit processes. You will also need to use built-in functions to fetch the current data and time, then insert it into relevant fields in the database whenever an item is added or edited.

Fun Facts

Objective: Highlight fun facts or featured items to engage users.

Requirements:

- Create a section for fun facts on the homepage or relevant pages.
- Admins must have the ability to add/edit/delete fun facts.
- Integrate this widget as part of the overall design.
- Use a random number function to change the featured fact every time the user accesses or refreshes the page.

Hints:

Design a dedicated database table for storing your facts.

Edit Item Photo

Objective: Allow admins to change the photo of an existing item.

Requirements:

- Provide an option to edit the photo in the item editing interface.
- Validate the new image file as per specified formats and sizes.
- Replace the existing photo with the new one upon successful upload.

Hints:

Use similar logic to the "Profile Photo" challenge, but apply it to items instead of user profiles. Consider how to handle the existing image file when a new one is uploaded (e.g., replace or archive it).

Pagination

Objective: Implement pagination for every time catalogue items are listed. This include the ‘browse’ page, as well as any search results or after filters are applied.

Requirements:

- Break down the list of items into multiple pages, with a specified number of items per page.
- Provide navigation controls to move between pages.
- Display the current page number and total pages to guide the user.

Hints:

- Determine the total number of items and divide by the number of items per page to calculate total pages.
- Use SQL `LIMIT` and `OFFSET` clauses to fetch the correct subset of items for each page.
- Implement navigation controls with links to previous, next, and specific page numbers.

Advanced Search

Objective: Create an advanced search functionality that enables users to search based on multiple criteria.

Requirements:

- Provide search filters that allow users to narrow down results by various attributes, such as category, date, rating, etc.
- Allow users to combine multiple search criteria.
- Display the search results in a user-friendly manner, with relevant information shown.

Hints:

- Create a form with multiple input fields or dropdowns to capture search criteria.
- Build a dynamic SQL query that includes conditions based on the user's input.
- Handle cases where some or all search fields are left blank.

Tagging System

Objective: Implement a tagging system to categorise items.

Requirements:

- Allow admins to create and apply tags when inserting a new item to the catalogue.
- Provide a filtering option to view items based on selected tags.
- Display tags associated with an item in a visually appealing way.
- On either a separate page or in a widget, display the most popular tags.

Hints:

There are many ways to create a tagging system. One might be to simply store the tags as comma separated values in each individual entry or record. This means that when you are figuring out what the most popular tags are, you will need to separate each string into individual tags, store them in another variable or data type, then count the number of occurrences of each.

You may also consider creating a table to store the tags and a junction table. This will allow you to take advantage of many-to-many relationships. This junction table would map the relationship between tags and the items they're tagging.

Custom Challenge

Objective: Create a challenge that aligns with the project's theme or something you are personally interested in learning.

Requirements:

- **Speak with your instructor for approval**. During your discussion, you must detail the specific objective and constraints of the custom challenge.
- If the challenge is too simple, your instructor may suggest ways to make it more challenging; if it is too hard, or will take too long to implement, your instructor may also suggest alternatives.