

DMIT2503 – Advanced Web Concepts

Lab 2 - Community Gallery

Overview: Based on the lessons in user registration, the multi-user blog, and your previous experience creating an Image Gallery in DMIT2025, please create a Community or Multi-User Gallery.

Use the login script the same way we did with the Multi-User Blog. You may use your own Gallery from DMIT2025, or you can use the basic one provided.

This is worth 10 marks and 10% of your final mark. Several hours of class time will be allocated for this project.

Functionality:

Demonstrate a working Gallery with the following basic features:

- Anyone can view public output.
- Anyone wanting to upload images is forced to register and login.
- All basic upload functionality, validation, etc. is in place (in provided example or your own build from DMIT2025).
- A nested query structure (query within a query) or MySQL JOIN must be used to output thumbnails according to different Author Galleries.
- Single image view is in a display page (NOT in a modal Lightbox-type feature): Full size display photo shown on a new page with all necessary information about the picture including all information about author (the person that uploaded that picture) such as first and last name.

Initial Steps:

- Copy the /login/ folder from MUB, and change the configuration files to match the new project. Since you already have a *users* table with users registered, you can test with the same user's data.
- Create an *author_id* field in your *mugallery* table.
- Secure the insert.php page with the login code.
- Tag each photo uploaded with the ID of the author (the logged-in user who is uploading a photo).

Index: Thumbnail View by Author Galleries

Here's one way of grouping thumbnails into Author Galleries as we display them (see screen shot below).

We will use one outside query to grab the ID's of any authors that have published photos. We could get the data from the *users* table, but in case we have registered users that have not published, we will get the data from the *mugallery* table instead. So we can get the ID of each user only once, we can use MySQL DISTINCT or GROUP BY (you will want to research this).

This will retrieve each *author_id* of contributing authors, but no duplicates.

Your gallery should show a limited number of images from each author but also allow to see ALL images from one author.

DMIT2503 – Advanced Web Concepts

Inside this outside query, we will have 2 nested queries:


1. User data: One that uses the *author_id* from the *mugallery* table, and matches it to the *user_id* field of the *users* table. We can then retrieve first and last name of the author.
 - While we have the author name, create a <div> to contain them and echo out the name as a heading of sorts.
2. Gallery data: Another that retrieves all thumbnails from the gallery, but only WHERE the *author_id* matches the particular author in the current iteration of the loop from the outside query.


Feel free to deviate from this and do it your own way if you like.


Main page (index.php) showing thumbnails output by author galleries.


Community Gallery


many people; many pictures


**Amber Waves** [More by Amber](#)


Miami Peeps


More club fun


Dancing Girl


Street scene


**Dirk Diggler** [More by Dirk](#)


Chasing something


Standoff


Romping in the park


Stalking his prey

**Reed Rothchild** [More by Reed](#)

Launching

Fly away...

Above launch

Into the sunset...

DMIT2503 – Advanced Web Concepts

Display page: A single image with all information is shown.

Community Gallery

many people; many pictures

Romping in the park

Bruno the wonder-mutt enjoying the summer.



Dirk Diggler



[<< Back to Gallery](#)

Advanced Feature: Ratings System

A Ratings System might look like the following:

When displaying a single picture, and authenticated user (logged in) will have the ability to rate a picture from 1 to 5. They can only rate this picture once; any subsequent view of this same picture will inform them they have already rated the picture.

DMIT2503 – Advanced Web Concepts

Ratings

1

2

3

4

5

Submit Rating

Average Rating for this picture:
2.5 out of 5
from 2 members.

Ratings

You have already rated this picture

Average Rating for this picture:
2.5 out of 5
from 2 members.

The Ratings System will also display the average rating from all members rating plus the number of users that have rated it.

The DB table may look like this:

```
SELECT * FROM `mugallery_ratings`
```

Options		picture_id	rater_id	rating	ratings_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		17	3	4	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		18	3	4	2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		36	3	5	3
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		21	3	2	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		14	3	1	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		17	4	2	6
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		36	4	4	7

Getting Started: This will involve a few queries. Perhaps the first one would be to see if the user has rated this picture before or not, and display a ratings form if not.

DMIT2503 – Advanced Web Concepts

```
$rating = mysqli_query($con,"SELECT * FROM mugallery_ratings WHERE picture_id = '$imageid' AND  
rater_id = $user_id");
```

```
$rowcount=mysqli_num_rows($rating);  
if($rowcount == 0){  
    if (isset($_SESSION['user_id'])){  
        include("ratingsform.php");  
    }  
}else{  
    echo "<p>You have already rated this picture</p>";  
}
```

This form will submit to a new file called admin/ratings.php; user is redirected back to \$_SERVER['HTTP_REFERER'] after adding data about rating.

Marks:

- Basic Functionality: Multiple users, etc. [4 marks].
 - No marks for the basic gallery: We did that last term and I am giving you one to work with. It MUST be multi-user.
 - Access to admin functionality (insert, edit, delete) must be secured using the login script we have learned.
- Output into Author Galleries and filter by author ("More by User...") : [2 marks].
- Overall design, usability, and content: [2 marks].
 - Registration/login forms better designed. All login pages must be integrated into your app (header, footer).
 - All links added.
 - Please have at least 3+ authors and 5+ photos each.
 - Feel free to ask a classmate to login to your project and contribute their photos.
- Ratings features: [2 marks total]
 - User can rate picture from the single picture display page.
 - Each user can rate each picture only once.
 - Average rating and number of ratings for this picture displayed.