

Project type: Web design and development with database design and development

Project title:

Physical Media Inventory Manager (Vinyl, Cassette, CD)

Analysis of the problem (10 marks).....	3
Description of the problem.....	3
UML use case diagram.....	4
Requirements specification	4
Project plan.....	7
Design of the solution (20 marks)	9
Design of Advanced Higher concepts	9
Design of integration	10
Design of additional functional requirements.....	12
User interface design.....	14
Implementation (30 marks)	16
Log of ongoing testing.....	17
Advanced Higher concepts	19
Implementation of integration	21
Implementation of additional requirements	23
User interface implementation.....	27
Research and development of new skills and/or knowledge	30
Testing the solution (15 marks).....	32
Persona and test cases(3 marks)	32
Comprehensive test plan	34
Evidence of requirements testing	36
Evaluation (5 marks)	44
Fitness for purpose	44
Future maintainability	45
Robustness	46
Appendix	46

Analysis of the problem (10 marks)

Before you begin designing and developing a solution, you must analyse the problem that you are going to solve, to ensure that you fully understand every aspect of it. This stage of your project should take around 5 hours.

Description of the problem

(2 marks)

Outline of the problem

Your outline should identify the web design and development Advanced Higher concepts and the integration with a database that will be used in your project.

Refer to page 19 of the Coursework Assessment Task to help with this.

(150-300 words)

People like my parents (including myself) who own lots of different types and forms of physical media (Vinyl, Cassette, VHS, DVD and CD) struggle to find and remember what they do and don't have, as well as where each item is actually kept. Having a system that keeps track of everything in one place will make it much easier to search, filter and manage.

This will require a website that:

- Uses a database table to store login details of different users relating to their media details
- Uses a database table to store details about the different owned physical media
- Uses HTML form to receive details about the media such as: title, type, year and location
- Uses a session variable to temporarily store the search filters
- Uses a media query to align elements on my website for mobile use
- Connects with the database and queries it to get the physical media data and display the details and location of the item
- Will validate the item details input to make sure they all are formatted the same (sanitising them)

In addition, the project could be expanded to include a wish list feature that lets users store items they want to buy in the future, which would use a separate database table and form.

Constraints

Describe any constraints for your project.

(100 words)

One of the main constraints of this project will be time, as the website needs to be: analysed, designed, implemented, tested and evaluated within the school year deadlines. Hardware will not be a major issue since I will be developing on my personal laptop, which was designed for this sort of task. Although the project will still be limited to a local server environment such as XAMPP for building and testing. Another constraint is my own coding ability, so I will need to keep the features achievable within my current knowledge of HTML, PHP and SQL as to not overwhelm myself.

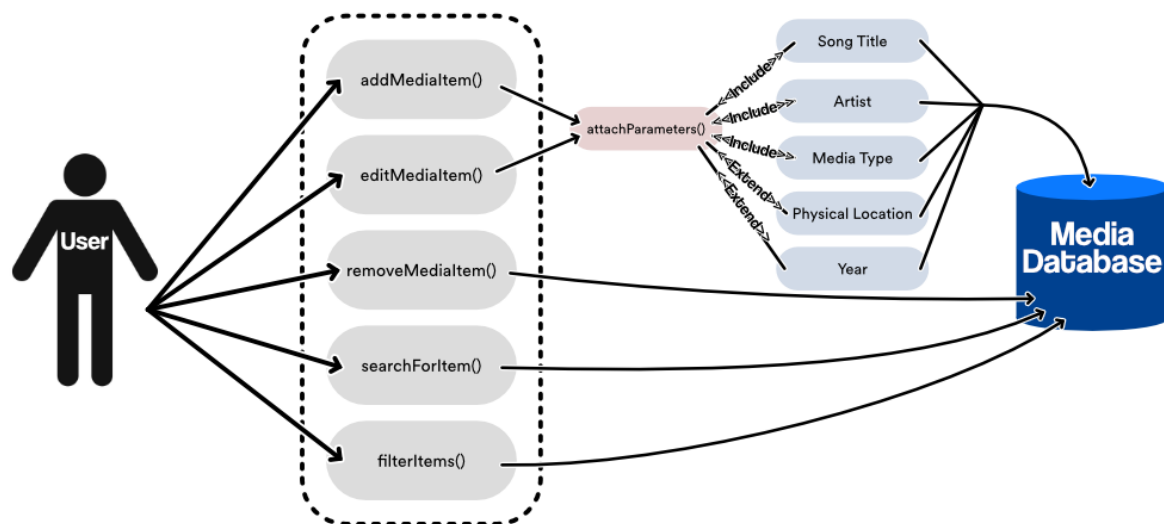
UML use case diagram

(2 marks)

Draw a UML use case diagram for your problem. Your diagram should define the following:

- ◆ actors
- ◆ use cases
- ◆ relationships

Your diagram should include the integration with the database that you intend to implement.



Requirements specification

(4 marks)

End user requirements

Identify a maximum of six end-user requirements for your project.

	The end-users of the solution should be able to:
EU1	The user should be able create and/or login to their account to view their personal collection
EU2	The user must be able to add new physical media to the database using a form to enter things such as: title, type, year and storage location
EU3	The user must be able to view their full collection in a clear and organised format
EU4	The user must be able to search their collection by title or artist
EU5	The user must be able to sort/filter their collection by type (e.g. Vinyl, Cassette, CD)
EU6	The user should be able to edit the details of the physical media or remove physical media

Functional requirements

Identify the functional requirements for your project.

Advanced Higher concepts

Complete the pre-populated functional requirements for the Advanced Higher concepts to describe how they will be applied in your project.

	The solution is required to:
FR1	A website with multiple pages, one of which has form elements (action, method, and name) to input details about the physical media into the database that can be accessed from anywhere.
FR2	a media query to adjust the website layout for smaller screens that is triggered when the browser window width is below a certain size
FR3	use PHP to assign form variables and process form data to check the entered values are valid
FR4	use PHP to assign and use session variables to temporarily store a user's userID, username and login time to keep users signed in on all pages

Integration

Complete the pre-populated functional requirements for integration to describe how they will be applied in your project.

	The solution is required to:
FR5	Have a database table to store details of all physical media items, including title, artist, type, year, and storage location.
FR6	Connect to the database to execute a query to get, insert, update or remove media records depending on the user's choice.
FR7	Generate an interface to receive query input values and display formatted query output so that users can enter search terms or filters and view a organised list of matching media items.

Additional functional requirements

Identify between four and eight additional functional requirements, a minimum of two of these must relate to validation of keyboard input.

This could include SQL queries in addition to the one required for integration.

A single functional requirement must relate to a single function the program will carry out. This should not include refinements of a functional requirement with multiple bullet points.

	The solution is required to...
FR8	validate the 'year' field so it only accepts numerical inputs so the user cannot enter invalid characters
FR9	validate the 'media format' field so it only accepts: vinyl, cassette or CD so it always matches valid formats
FR10	provide a search function so users can easily find media items by category
FR11	allow editing and deletion of media items so that users can keep their collection up to date
...	

Project plan

(2 marks)

Create a project plan for the four remaining stages of your project.

Your project plan should include:

- ◆ the tasks you complete in each stage
- ◆ any resources you need to implement your solution
- ◆ an estimate of how long each stage and tasks will take

Guidance for producing a project plan

Your list of tasks for each stage could include:

- ◆ user-interface design
- ◆ implementation of input validation
- ◆ ongoing testing

Resources can include access to specialist software and development tools. Some of these could be available at any time, while others may only be available at certain times. You need to plan to ensure that your project is not held up waiting for resources.

Your timings should allow for holidays, or other events that affect how much time you can spend on your project.

You should review and update your project plan as you work through each stage.

Insert the **final version** of your project plan including timings and list any resources needed to implement your solution.

Things to do:

- Design each functional requirement
- Design user interface
- Setup XAMPP workspace
- Make basic non function website just with structure of all
- Get website to talk to a database in a meaningful way
- Get the website to update dynamicly based on database
- Style website in a user friendly and modern way
- Get people around me to test my website/database
- Stress test the database to make sure it all works nicely
- Write up evaluation

Resources:

- Computer and WiFi connection
- Pen drive or SD card to store database

Tools To Be Used:

- VS Code (code editor)
- XAMPP (PHP and SQL workspace manager + server)
- Chrome (web browser)

	Sep	Oct	Nov	Dec	Jan	Febr	Mar	Apr
Analysis								
Design								
Implementation								
Testing								
Evaluation								

Design of the solution (20 marks)

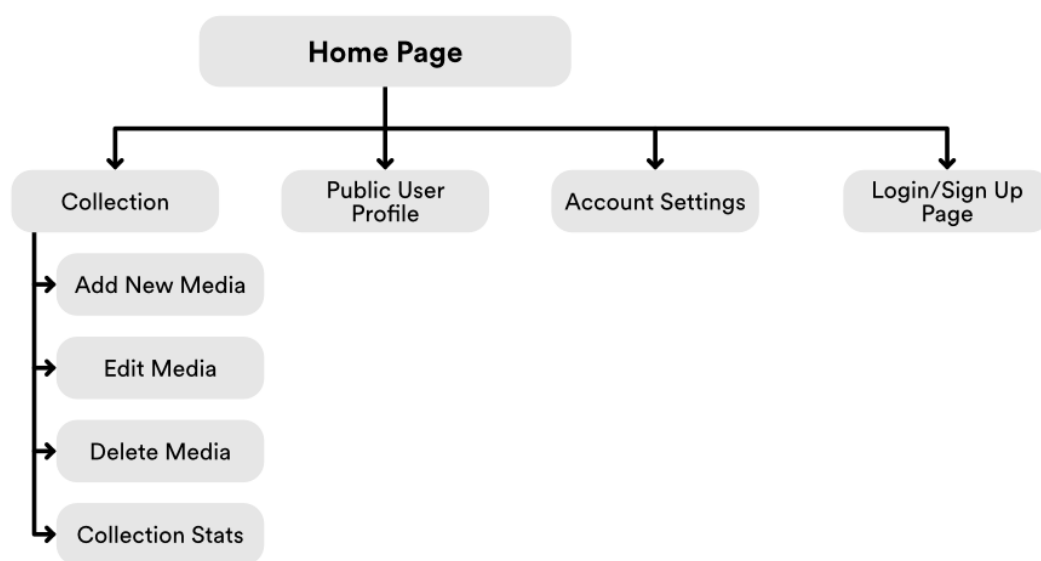
The design stage of your project should take around 10 hours, including time to revisit and adapt as the project progresses.

Design of Advanced Higher concepts

(6 marks)

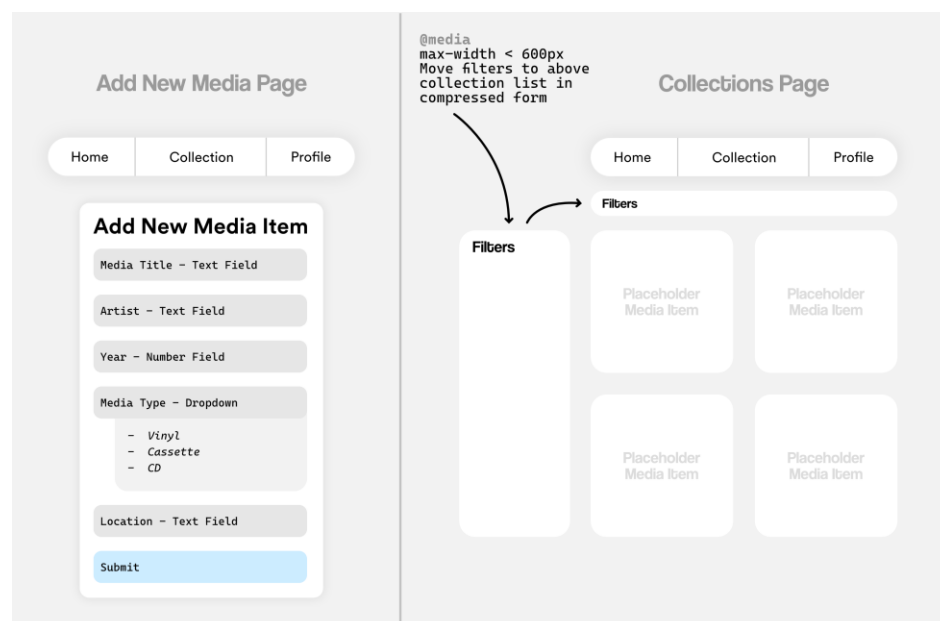
Design of the Advanced Higher concepts identified during your analysis.

Design for FR1 - A website navigation structure indicating values being passed between pages



Design for FR2 - Design of media query should be shown in user-interface design

Design of form elements should be shown in user-interface design



Design for FR3 - Pseudocode design showing:

- ◆ assignment of form variables
- ◆ processing of form data

```
#Assignment
SET title TO form["title"]
SET artist TO form["artist"]
SET year TO form["year"]
SET format TO form["format"]
SET location TO form["location"]

#Processing
IF title IS EMPTY
    DISPLAY "Error: Title is required"
ELSE IF artist IS EMPTY
    DISPLAY "Error: Artist is required"
ELSE IF year IS NOT NUMERIC OR year < 1900 OR year > CURRENT_YEAR
    DISPLAY "Error: Enter a valid year"
ELSE IF format IS EMPTY
    DISPLAY "Error: Please select a format"
```

Design for FR4 - Pseudocode design showing:

- ◆ assignment of session variables
- ◆ use of session variables (where appropriate, this could be shown in the user interface design)

```
#Session Assignment
SET session["userID"] TO form["userID"]
SET session["username"] TO form["username"]
SET session["login_time"] TO current_time

#Use of vars - keep user signed in
IF session["userID"] IS NOT EMPTY AND session["username"] IS NOT EMPTY THEN
    KEEP user signed in across all pages
ELSE
    REDIRECT user TO login page
```

Design of integration

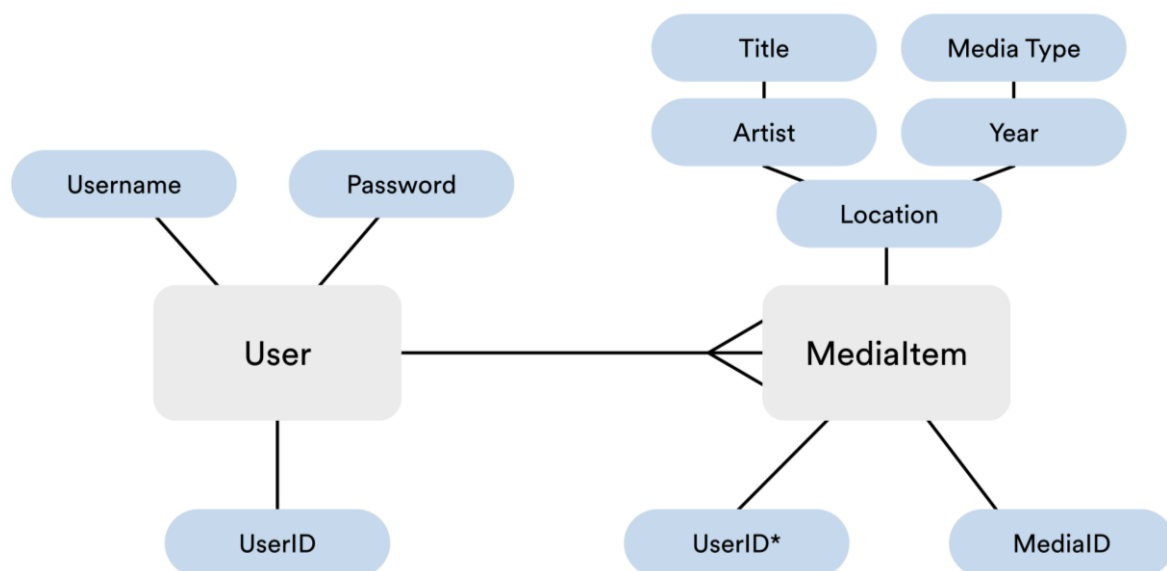
(5 marks)

Design the integration identified during your analysis.

Design of FR5 - data dictionary and entity relationship diagram (if more than one table)

Attribute name	Key	Type	Size	Required	Validation
userID	PK	Int		Y	
username		VarChar	50	Y	
password		VarChar	20	Y	

Attribute name	Key	Type	Size	Required	Validation
userID	FK	Int		Y	
mediaID	PK	Int		Y	
title		VarChar	50	Y	
artist		VarChar	50	Y	
mediaType		VarChar	20	Y	It is either: CD, Vinyl, Cassette
year		Int	4	N	
location		VarChar	50	N	



Design of FR6

Design of connection to the database and design of execution of query (using pseudocode)

Design of query:

```

#Use of vars
IF session["search_type_filter"] IS "title" AND session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE title LIKE session['search_filter']" TO sql_query

IF session["search_type_filter"] IS "artist" AND session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE artist LIKE session['search_filter']" TO sql_query

IF session["search_type_filter"] IS "location" AND session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE location LIKE session['search_filter']" TO sql_query

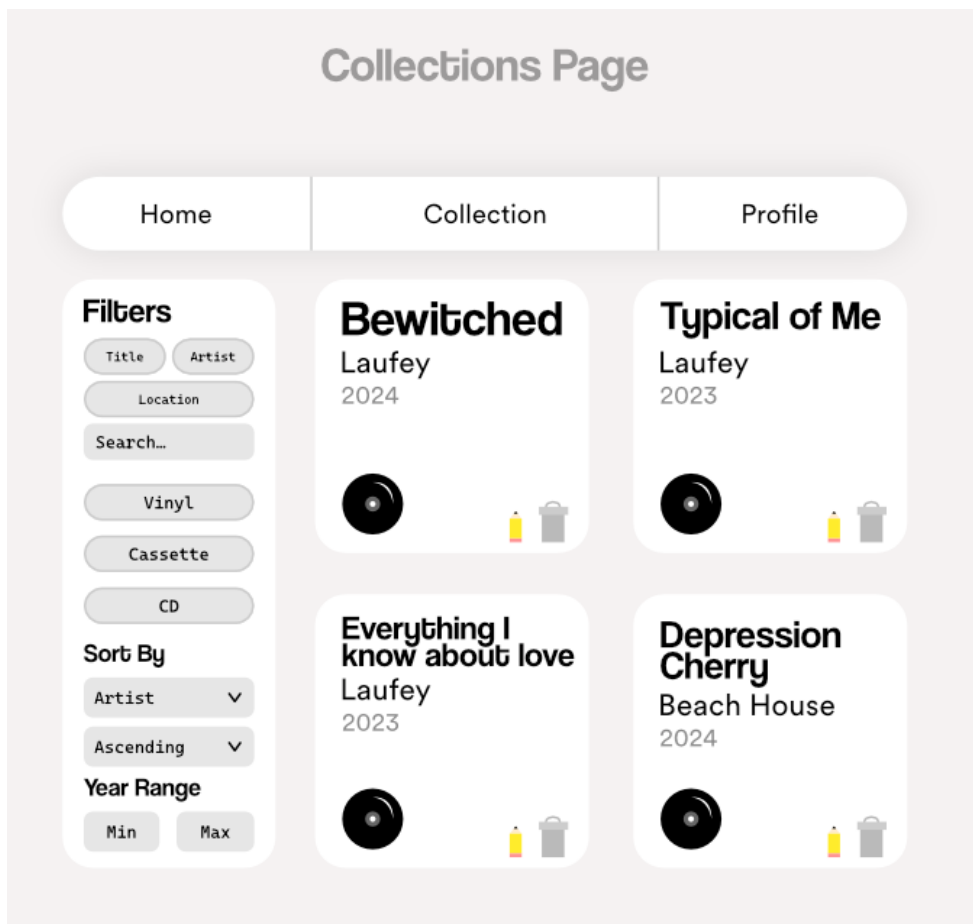
IF session["format_filter"] IS "Vinyl" OR session["format_filter"] IS "Cassette" OR session["format_
filter"] IS "CD" THEN
    APPEND " AND format = session['format_filter']" TO sql_query

IF session["year_filter"] IS NOT EMPTY AND IS INT THEN
    APPEND " AND year = session['year_filter']" TO sql_query

#Connection and Execution
CONNECT TO database
SET sql_query TO "INSERT INTO media (title, artist, year, format, location)"
EXECUTE sql_query
CLOSE database CONNECTION

```

Design of FR7 - This evidence should be shown in the user-interface design



Design of additional functional requirements

(4 marks)

Design the additional functional requirements (maximum of 8), including validation of keyboard input, identified during your analysis.

Design of FR8

```
IF session["year_filter"] IS NOT EMPTY AND IS INT THEN
```

Design of FR9

```
IF session["format_filter"] IS "Vinyl" OR session["format_filter"] IS "Cassette" OR session["format_filter"] IS "CD" THEN
```

Design of FR10

```
#Session Assignment
SET session["search_type_filter"] TO form["search_type_filter"]
SET session["search_filter"] TO form["search_filter"]
SET session["format_filter"] TO form["format_filter"]
SET session["year_filter"] TO form["year_filter"]
```

```
#Use of vars
IF session["search_type_filter"] IS "title" AND
session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE title LIKE session['search_filter']" TO
sql_query
```

```
IF session["search_type_filter"] IS "artist" AND
session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE artist LIKE session['search_filter']" TO
sql_query
```

```
IF session["search_type_filter"] IS "location" AND
session["search_filter"] IS NOT EMPTY THEN
    APPEND " WHERE location LIKE session['search_filter']"
TO sql_query
```

Design of FR11

```
#Edit media item
CONNECT TO database
SET sql_query TO "UPDATE media
    SET title = new_title,
        artist = new_artist,
        year = new_year,
        format = new_format,
        location = new_location
    WHERE id = media_id"
EXECUTE sql_query
CLOSE database CONNECTION
```

```
#Delete media item
CONNECT TO database
SET sql_query TO "DELETE FROM media WHERE id = media_id"
EXECUTE sql_query
CLOSE database CONNECTION
```

User interface design

(5 marks)

Design of the user interface for your solution.

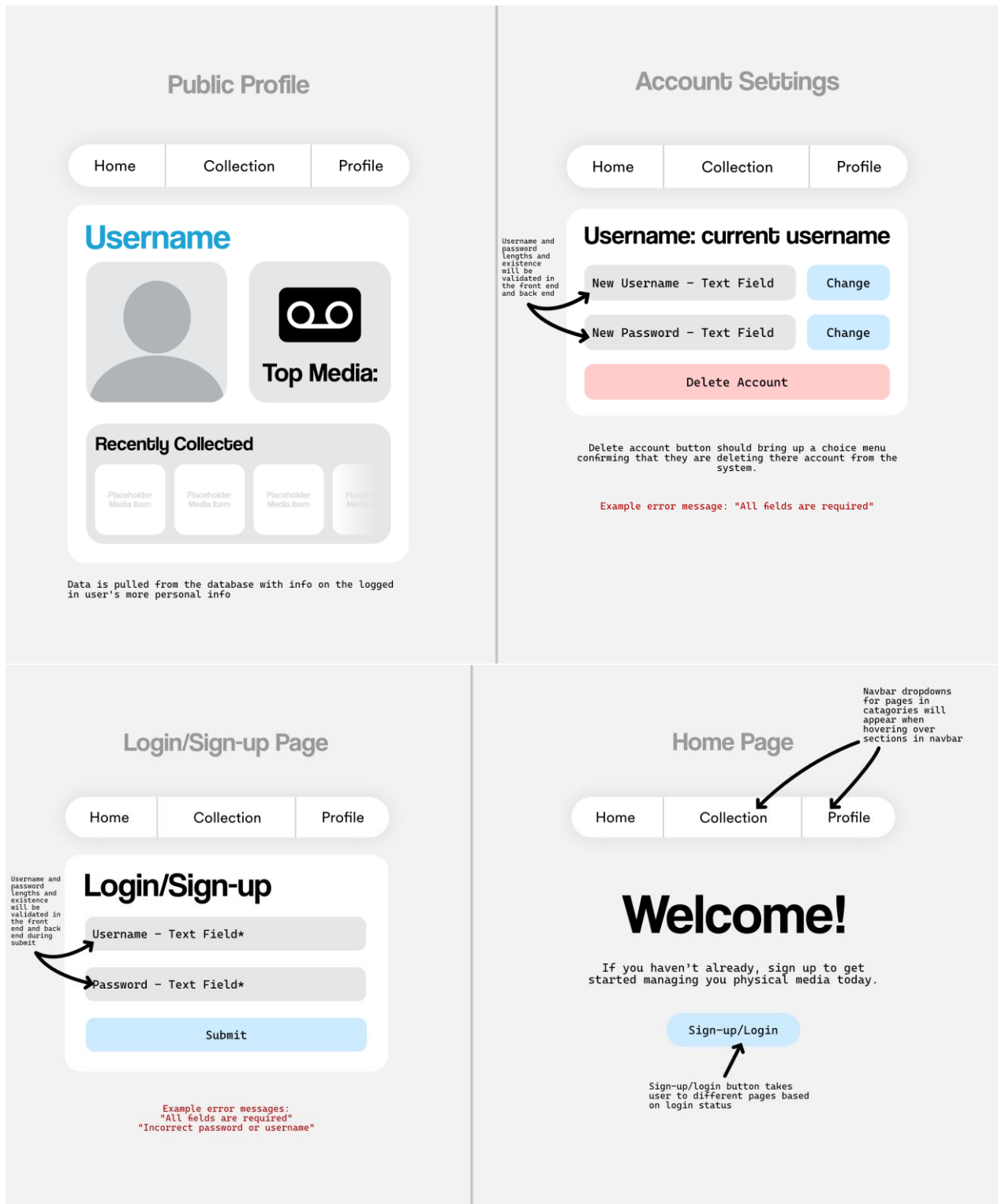
Your user interface design should be annotated wireframes showing:

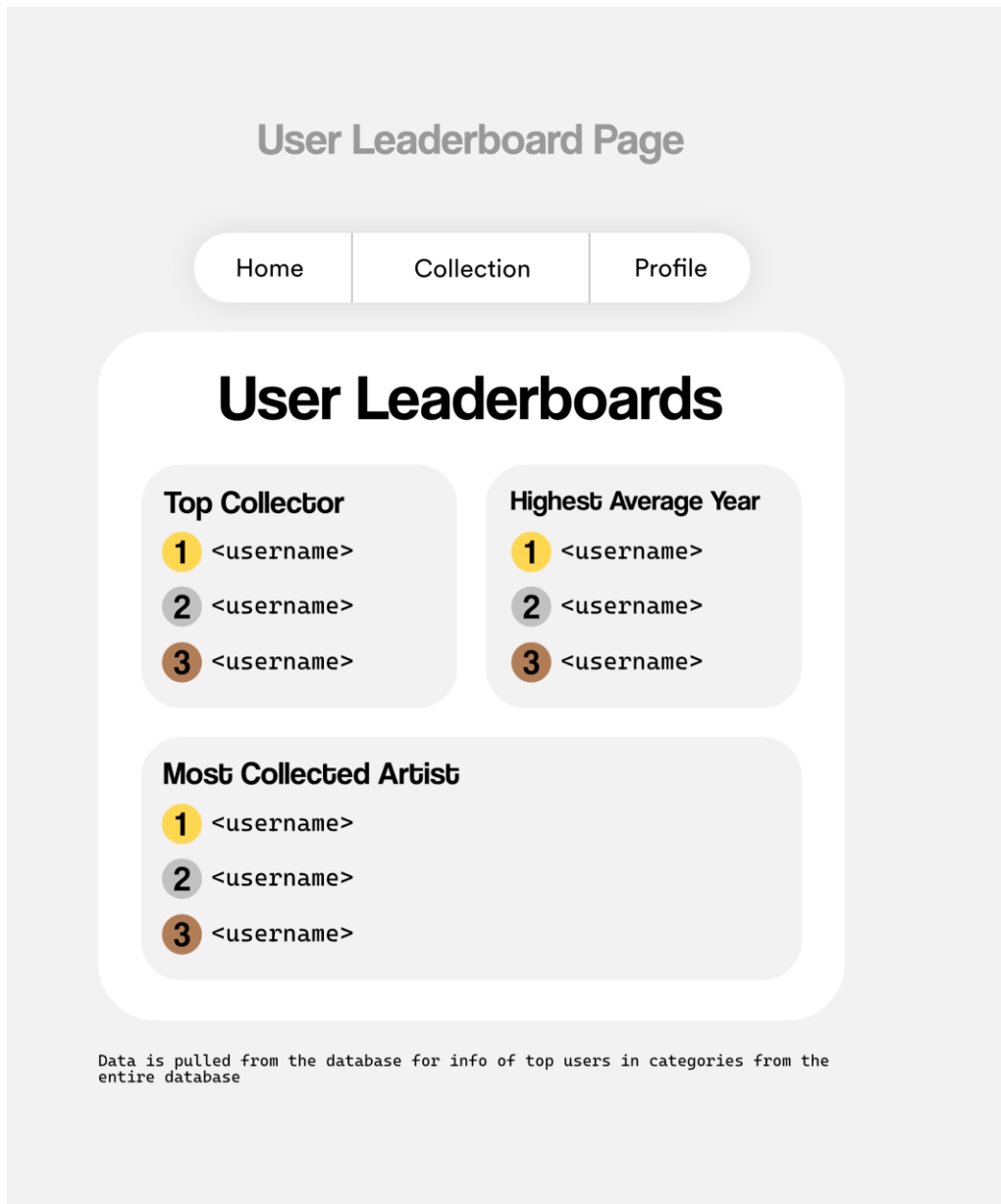
- Input forms showing validation of keyboard input
- Buttons and menu options showing underlying processes
- Output screens including error messages resulting from input validation

Your design should refer to relevant end-user and/or functional requirements.

The wireframes illustrate a music collection management application with the following components and annotations:

- Add New Media Page:** Features a navigation bar (Home, Collection, Profile) and a form titled "Add New Media Item". The form includes fields for "Media Title", "Artist", "Year", "Media Type" (a dropdown menu with options: Vinyl, Cassette, CD), and "Location". A "Submit" button is at the bottom. Annotations include: "Submit button sends form request with POST to allow the database backend to be executed inserting the data into the correct table" and "Example error message: 'Title, Artist and Media Type are required'".
- Collections Page:** Features a navigation bar and a "Filters" section with radio buttons for "Title", "Artist", and "Location", a "Search..." field, and buttons for "Vinyl", "Cassette", and "CD". It also has "Sort By" (Artist, Ascending) and "Year Range" (Min, Max) options. The main area displays four media items: "Bewitched Laufey 2024", "Typical of Me Laufey 2023", "Everything I know about love Laufey 2023", and "Depression Cherry Beach House 2024". Annotations include: "Radio buttons switch between category to filter", "Numerical input only" (pointing to the Year Range fields), "Widgets are dynamically created from retrieving data from the database based on the query generated from the filters.", and "Edit and delete buttons redirect to corresponding page for specific element (example edit.php?id=4)".
- Edit Media Page:** Features a navigation bar and a form titled "Edit Media Item". The form includes fields for "New Media Title", "New Artist", "New Year", "New Media Type" (a dropdown menu with options: Vinyl, Cassette, CD), and "New Location". A "Submit Change" button is at the bottom. Annotations include: "Data is loaded into the form fields from the preexisting data in the database", "GET method will be used to determine the ID", "Field validation will occur on the front and back end, for things such as type (numbers, 2 media types) and existence in required.", and "Submit button sends form request with POST to allow the database backend to be executed updating the correct data in the table".
- Collections Stats Page:** Features a navigation bar and displays statistics: "Collection Total: 34", "Average Year: 1998", "Top Media: [album icon]", and "Top Artist: Laufey". An annotation states: "Data is pulled from the database with a query that gets the top stats for the user logged in".





Implementation (30 marks)

The implementation stage of your project should take around 15 hours, including time to revisit and adapt as the project progresses.

Your implemented code should meet the functional requirements identified at the analysis stage and match you completed design.

The full code for your project must be added to your submission as an appendix at the end.

Log of ongoing testing

(5 marks)

As you implement your solution, you will encounter errors or problems that you need to solve before you can continue. Take notes of errors, solutions and any reference materials you use, for example websites, forums, textbooks or learning resources. You will need to refer to these notes to produce evidence of ongoing testing.

Complete the log of ongoing testing below.

	Describe how functional requirement was tested	Describe any issues encountered during testing	Describe how you resolved the issue noting any references
FR1	I clicked every link in the navbar and hovered over all the different elements to make sure they take you to the correct page.	No issues encountered	
FR2	I used the dev tools responsive dimensions feature to make sure my page correctly displays its contents on a variety of screen sizes. Pages such as the collection view with the grid structure that is meant to adapt.	I found a lot of pages had far too big margins when on smaller screens leaving a lot of the screen blank that could be used for info	I reduced the margins for smaller screens using media query
FR3	I also typed in very long names and left some entries blank in all the different forms to make sure all the corrected sized and correct data is passed in.	I found on the account settings page you could enter an infinity long username	I added a HTML maxLength limit to the input field but to make sure no one edits it I also now trim the string to be 25 chars long
FR4	I went to every page that uses session variables to test that they are being correctly accessed, used and set	On pages that didn't require login it wouldn't correctly start a session	I moved the start session code to be in a login required section of code into a login check section so even when not logged in it would start a session.

FR5	I used the XAMPP admin console to make sure all the correct data is in the correct columns and tables.	No issues encountered	
FR6	I manually wrote SQL queries for: Insert, Update, Delete and Select to test my database can do all the basic things.	No issues encountered	
FR7	I used the collection view filter buttons and inputs by changing them in a wide range of combinations to make sure they all work together to get a detailed search.	No issues encountered	
FR8	I removed the number tag to the year input to make sure my system fails as year is not a numerical number.	It did not stop the string as the year input	I added is_numeric to the if statement before submitting the data to the database. (found docs on php.net)
FR9	I edited the dropdown using the client-side dev tools to accept other values other than: Vinyl, Cassette and CD	It just let the new value through not stopping it	I made it validate the mediaType in the if statement before submitting data to server.
FR10	I used the category search input and radio buttons in as many ways as I could with various exceptional use cases	No issues encountered	
FR11	I used the edit and deletion functions repeatedly in lots of different ways to make sure it all works correct	No issues encountered	
...			

Advanced Higher concepts

(12 marks)

Provide evidence of your implemented Advanced Higher concepts.

This evidence will include code and, where appropriate, 'before' and 'after' screenshots to show that the code works correctly.

Implementation of FR1- Implementation of HTML form code

```
media_add.php
<h2 style="margin-bottom:16px; font-size:40px">Add New Media Item</h2>
<form method="POST">
  <div class="input-group">
    <input type="text" id="title" placeholder="Title*" name="title" required>
    <input type="text" list="artist" id="artist" placeholder="Artist*" name="artist" required>
    <datalist id="artist">
      <?php foreach ($allArtists as $artist): ?>
        <option value="<?= htmlspecialchars($artist['artist']) ?>"></option>
      <?php endforeach; ?>
    </datalist>
    <select id="mediaType" name="mediaType" required>
      <option value="">Media Type*</option>
      <option value="Vinyl">Vinyl</option>
      <option value="Cassette">Cassette</option>
      <option value="CD">CD</option>
    </select>
    <input type="number" id="year" name="year" placeholder="Year" min="1900" max="3000">
    <input type="text" id="location" placeholder="Location" name="location">
    <input type="submit" value="Add Media Item">
  </div>
</form>
```

Implementation of FR2- Implementation of media query

```
media.css
@media (max-width: 1100px) {
  #collection-filters {
    flex-direction: column;
    align-items: center;
    margin: 0;
    padding: 0 16px;
  }
  #collection-container {
    margin-right: 0;
    margin-bottom: 16px;
    min-width: 100%;
    width: 100%;
    order: 2;
  }
  #filters-container {
    max-width: 100%;
    margin-right: 0;
    margin-bottom: 16px;
    order: 1;
    align-self: center;
  }
  .collection-widget {
    width: auto;
  }
  #filters-container label {
    padding: 10px 20px;
    font-size: 14px;
    margin: 6px 2px;
  }
  .inline-inputs {
    display: flex;
    flex-wrap: wrap;
    gap: 8px;
  }
  .inline-inputs input[type="text"],
  .inline-inputs input[type="number"] {
    flex: 1;
    min-width: 120px;
  }
}

@media (max-width: 600px) {
  .collection-grid {
    grid-template-columns: repeat(1, 1fr);
  }
}

media.css
@media (max-width: 1100px) {
  #collection-filters {
    flex-direction: column;
    align-items: center;
    margin: 0;
    padding: 0 16px;
  }
  #collection-container {
    margin-right: 0;
    margin-bottom: 16px;
    min-width: 100%;
    width: 100%;
    order: 2;
  }
  #filters-container {
    max-width: 100%;
    margin-right: 0;
    margin-bottom: 16px;
    order: 1;
    align-self: center;
  }
  .collection-widget {
    width: auto;
  }
  #filters-container label {
    padding: 10px 20px;
    font-size: 14px;
    margin: 6px 2px;
  }
  .inline-inputs {
    display: flex;
    flex-wrap: wrap;
    gap: 8px;
  }
  .inline-inputs input[type="text"],
  .inline-inputs input[type="number"] {
    flex: 1;
    min-width: 120px;
  }
}

@media (max-width: 600px) {
  .collection-grid {
    grid-template-columns: repeat(1, 1fr);
  }
}
```

Implementation of FR3 and FR4 - Implementation of PHP code



account_settings.php

```
$newUsername = trim($_POST['change_username'] ?? '');  
$newUsername = substr($newUsername, 0, 25);  
if ($newUsername) {  
    $stmt = $pdo->prepare("UPDATE login_details SET username = ? WHERE userID = ?");  
    $stmt->execute([$newUsername, $_SESSION['user_id']]);  
}
```



login_details.php

```
$_SESSION['user_id'] = $existingUser['userID'];  
$_SESSION['username'] = $existingUser['username'];  
$_SESSION['login_time'] = time();
```

Implementation of integration

(6 marks)

Provide evidence of your implemented integration.

This evidence will include code and, where appropriate, 'before' and 'after' screenshots to show that the code works correctly.

Implementation of FR5 - Implementation of database structure and initial values in table(s).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	userID	int(11)			No	None			Change Drop More
☐ 2	mediaID	int(11)			No	None			Change Drop More
☐ 3	title	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐ 4	artist	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐ 5	mediaType	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
☐ 6	year	int(11)			No	None			Change Drop More
☐ 7	location	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	userID	int(11)			No	None			Change Drop More
☐ 2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐ 3	password	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Implementation of FR6 - Implementation of connection to the database and execution of query.

```
media_delete.php

$stmt = $pdo->prepare("DELETE FROM media_items WHERE mediaID = ? AND userID = ?");
$stmt->execute([$deleteMediaID, $_SESSION['user_id']]);
```

```
media_add.php

$stmt = $pdo->prepare("INSERT INTO media_items (userID, mediaID, title, artist, mediaType, year, location) VALUES (?, ?, ?, ?, ?, ?, ?)");
$stmt->execute([$userID, $mediaID, $title, $artist, $mediaType, $year, $location]);
```



media_edit.php

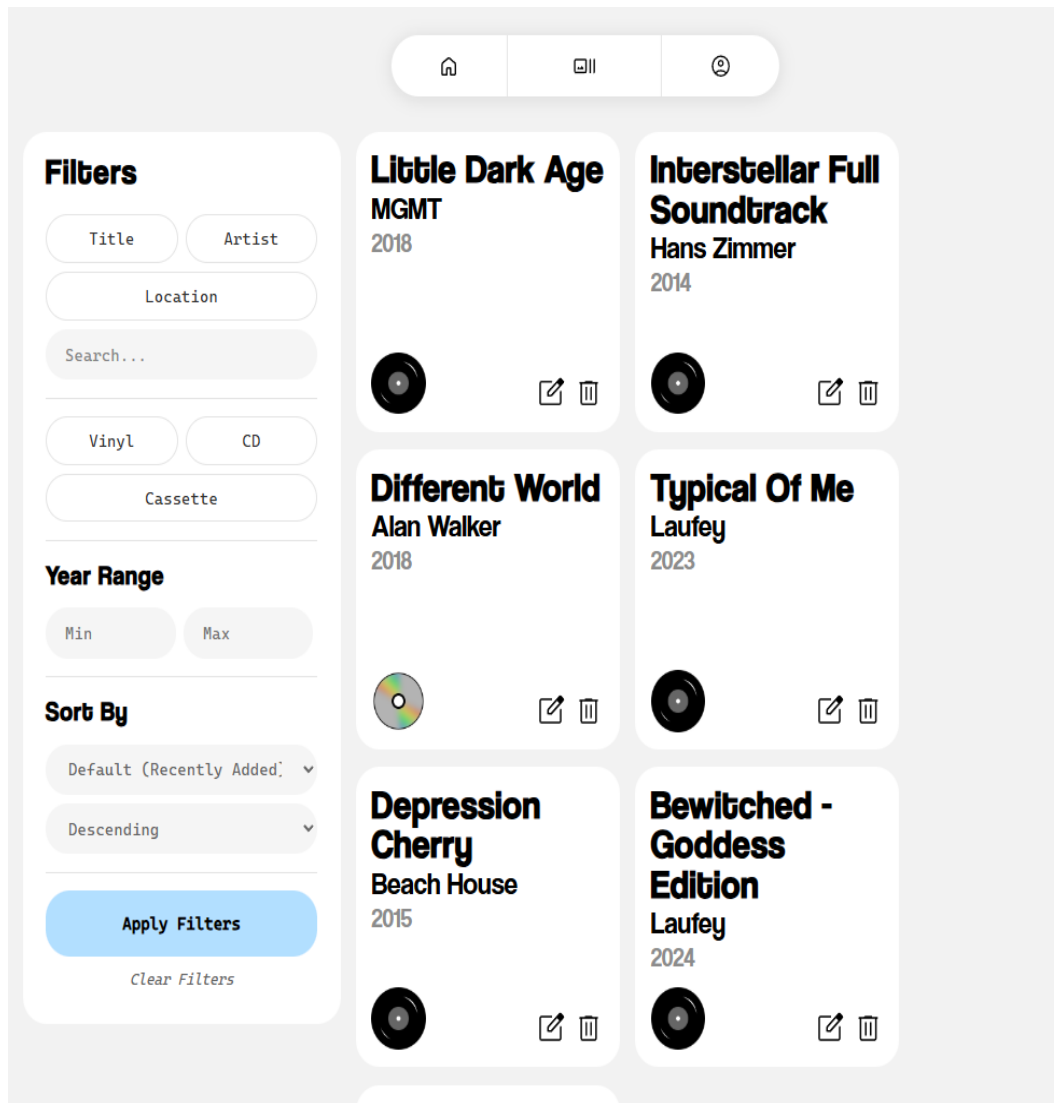
```
$stmt = $pdo->prepare("SELECT * FROM media_items WHERE mediaID = ? AND userID = ?");  
$stmt->execute([$mediaID, $_SESSION['user_id']]);  
$mediaItem = $stmt->fetch(PDO::FETCH_ASSOC);
```



media_edit.php

```
$stmt = $pdo->prepare("UPDATE media_items SET title = ?, artist = ?, mediaType = ?, year = ?, location = ? WHERE mediaID = ?  
AND userID = ?");  
$stmt->execute([$title, $artist, $mediaType, $year, $location, $mediaItem['mediaID'], $userID]);
```

Implementation of FR7 - This evidence should be shown in the implementation of user-interface



Implementation of additional requirements

Provide evidence of your implemented additional functional requirements (maximum 8).

This evidence will include code and, where appropriate, 'before' and 'after' screenshots to show that the code works correctly.

Implementation of FR8 and FR9

Before

```
media_edit.php
if ($userID && $mediaID && $title && $artist && $mediaType) {
```

After

```
media_edit.php
if ($userID && $mediaID && $title && $artist && $mediaType && is_numeric($year) && $year ≥ 1900 && $year ≤ 3000 &&
$mediaType = 'Vinyl' || $mediaType = 'Cassette' || $mediaType = 'CD') {
```

Implementation of FR10

```
collection_view.php
$searchField = $_GET['searchField'] ?? '';
$searchValue = $_GET['search'] ?? '';
$mediaTypeFilter = $_GET['mediaType'] ?? '';
$yearMin = $_GET['yearMin'] ?? '';
$yearMax = $_GET['yearMax'] ?? '';
$sortBy = $_GET['sort'] ?? 'mediaID';
$sortOrder = $_GET['order'] ?? 'DESC';

$query = "SELECT * FROM media_items WHERE userID = ?";
$params = [$user['user_id']];

if (!empty($searchValue) && !empty($searchField)) {
    if ($searchField === 'title') {
        $query .= " AND title LIKE ?";
        $params[] = '%' . $searchValue . '%';
    } elseif ($searchField === 'artist') {
        $query .= " AND artist LIKE ?";
        $params[] = '%' . $searchValue . '%';
    } elseif ($searchField === 'location') {
        $query .= " AND location LIKE ?";
        $params[] = '%' . $searchValue . '%';
    }
}

if (!empty($mediaTypeFilter)) {
    $query .= " AND mediaType = ?";
    $params[] = $mediaTypeFilter;
}

if (!empty($yearMin)) {
    $query .= " AND year ≥ ?";
    $params[] = (int)$yearMin;
}

if (!empty($yearMax)) {
    $query .= " AND year ≤ ?";
    $params[] = (int)$yearMax;
}

$allowedSortFields = ['mediaID', 'title', 'artist', 'year', 'mediaType'];
$allowedSortOrder = ['ASC', 'DESC'];

if (!in_array($sortBy, $allowedSortFields)) {
    $sortBy = 'mediaID';
}

if (!in_array($sortOrder, $allowedSortOrder)) {
    $sortOrder = 'DESC';
}

$query .= " ORDER BY $sortBy $sortOrder";

$stmt = $pdo->prepare($query);
$stmt->execute($params);
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

```

collection_view.php

<h2 style="cursor:pointer;" onclick="toggleFilters()">Filters</h2>
<form method="GET">
  <div class="inline-inputs">
    <label><input type="radio" name="searchField" value="title" <?php echo $searchField === 'title' ? 'checked' : ''; ?>> Title</label>
    <label><input type="radio" name="searchField" value="artist" <?php echo $searchField === 'artist' ? 'checked' : ''; ?>> Artist</label>
    <label><input type="radio" name="searchField" value="location" <?php echo $searchField === 'location' ? 'checked' : ''; ?>> Location</label>
    <input type="text" name="search" placeholder="Search ..." value="<?php echo htmlspecialchars($searchValue); ?>">
  </div>

  <div class="inline-inputs">
    <label><input type="radio" name="mediaType" value="Vinyl" <?php echo $mediaTypeFilter === 'Vinyl' ? 'checked' : ''; ?>> Vinyl</label>
    <label><input type="radio" name="mediaType" value="CD" <?php echo $mediaTypeFilter === 'CD' ? 'checked' : ''; ?>> CD</label>
    <label><input type="radio" name="mediaType" value="Cassette" <?php echo $mediaTypeFilter === 'Cassette' ? 'checked' : ''; ?>> Cassette</label>
  </div>

  <h3 style="margin:0; margin-bottom:16px;">Year Range</h3>
  <div class="inline-inputs">
    <input type="number" name="yearMin" placeholder="Min" value="<?php echo htmlspecialchars($yearMin); ?>">
    <input type="number" name="yearMax" placeholder="Max" value="<?php echo htmlspecialchars($yearMax); ?>">
  </div>

  <h3 style="margin:0; margin-bottom:16px;">Sort By</h3>
  <div class="inline-inputs">
    <select name="sort">
      <option value="mediaID" <?=$sortBy === 'mediaID' ? 'selected' : '' ?>>Default (Recently Added)</option>
      <option value="title" <?=$sortBy === 'title' ? 'selected' : '' ?>>Title</option>
      <option value="artist" <?=$sortBy === 'artist' ? 'selected' : '' ?>>Artist</option>
      <option value="year" <?=$sortBy === 'year' ? 'selected' : '' ?>>Year</option>
      <option value="mediaType" <?=$sortBy === 'mediaType' ? 'selected' : '' ?>>Media Type</option>
    </select>

    <select name="order">
      <option value="ASC" <?=$sortOrder === 'ASC' ? 'selected' : '' ?>>Ascending</option>
      <option value="DESC" <?=$sortOrder === 'DESC' ? 'selected' : '' ?>>Descending</option>
    </select>
  </div>

  <div>
    <button type="submit">Apply Filters</button>
    <a href="<?php echo $_SERVER['PHP_SELF']; ?>">Clear Filters</a>
  </div>
</form>

```

Implementation of FR11

(On next page)

```

media_delete.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$deleteMediaID = $_GET['mediaID'] ?? null;
$action = $_GET['action'] ?? null;
$error = null;
$media = null;

if ($deleteMediaID !== null) {
    $stmt = $pdo->prepare("SELECT mediaID, title, artist FROM media_items WHERE mediaID = ? AND userID = ?");
    $stmt->execute([$deleteMediaID, $_SESSION['user_id']]);
    $media = $stmt->fetch(PDO::FETCH_ASSOC);

    if (!$media) {
        $error = "Media item not found or you don't have permission to delete it.";
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST' && $action === 'delete_media' && $media) {
    $deleteStmt = $pdo->prepare("DELETE FROM media_items WHERE mediaID = ? AND userID = ?");
    $deleteStmt->execute([$deleteMediaID, $_SESSION['user_id']]);

    header("Location: collection_view.php");
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Confirm Deletion</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
<?php include 'resources/navbar.php'; ?>
<div id="main-body">
    <div class="centered-container" style="height:100%">
        <div class="content-box">
            <h2 style="color:red;">Confirm Media Item Deletion</h2>
            <?php if ($error): ?>
                <p style="color:red;"><?= htmlspecialchars($error) ?></p>
                <a href="collection_view.php"><button>Back</button></a>
            <?php elseif ($media): ?>
                <p>This action cannot be undone.<br>Are you sure you want to delete the following media item?</p>
                <p><strong>Title:</strong> <?= htmlspecialchars($media['title']) ?><br><strong>Artist:</strong> <?=
htmlspecialchars($media['artist']) ?></p>
                <form method="POST" action="<?= htmlspecialchars($_SERVER['PHP_SELF']) . "?action=delete_media&mediaID=" .
urlencode($deleteMediaID) ?>">
                    <input type="submit" value="Yes, Delete This Media" class="warning_button">
                </form>
                <a href="collection_view.php"><button style="margin-top:16px">Cancel</button></a>
            <?php else: ?>
                <p>No media item selected.</p>
                <a href="collection_view.php"><button>Back</button></a>
            <?php endif; ?>
        </div>
    </div>
</div>
</body>
</html>

```

```

media_edit.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$user = getCurrentUser();
$messageType = '';

$mediaID = $_GET['id'] ?? null;
$mediaItem = null;

if (!$mediaID) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>No media item specified. Please provide a valid media ID.</p>";
    exit;
}

$stmt = $pdo->prepare("SELECT * FROM media_items WHERE mediaID = ? AND userID = ?");
$stmt->execute([$mediaID, $_SESSION['user_id']]);
$mediaItem = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$mediaItem) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>Media item not found or you don't have permission to edit it.</p>";
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $userID = $_SESSION['user_id'];
    $title = $_POST['title'] ?? '';
    $artist = $_POST['artist'] ?? '';
    $mediaType = $_POST['mediaType'] ?? '';
    $year = $_POST['year'] ?? '';
    $location = $_POST['location'] ?? '';

    if ($userID && $mediaID && $title && $artist && $mediaType && is_numeric($year) && $year >= 1900 && $year <= 3000 &&
    $mediaType === 'Vinyl' || $mediaType === 'Cassette' || $mediaType === 'CD') {
        $stmt = $pdo->prepare("UPDATE media_items SET title = ?, artist = ?, mediaType = ?, year = ?, location = ? WHERE
        mediaID = ? AND userID = ?");
        $stmt->execute([$title, $artist, $mediaType, $year, $location, $mediaItem['mediaID'], $userID]);

        if ($stmt->rowCount() > 0) {
            header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=success");
            exit;
        } else {
            header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=error");
            exit;
        }
    } else {
        header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=missing");
        exit;
    }
}

if (isset($_GET['message'])) {
    $messageType = $_GET['message'];
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Media Item</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="content-box">
                <h1>Edit Media Item</h1>

                <p><strong>Currently editing:</strong> <?= htmlspecialchars($mediaItem['title']) ?> by <?=
                htmlspecialchars($mediaItem['artist']) ?></p>
                <form method="POST">
                    <div class="input-group">
                        <input type="text" id="title" name="title" placeholder="Title" value="<?=
                        htmlspecialchars($mediaItem['title']) ?>" required>
                        <input type="text" id="artist" name="artist" placeholder="Artist" value="<?=
                        htmlspecialchars($mediaItem['artist']) ?>" required>
                        <select id="mediaType" name="mediaType" required>
                            <option value="">Media Type</option>
                            <option value="Vinyl"><?= ($mediaItem['mediaType'] === 'Vinyl') ? 'selected' : '' ?
                            >>Vinyl</option>
                            <option value="Cassette"><?= ($mediaItem['mediaType'] === 'Cassette') ? 'selected' : '' ?
                            >>Cassette</option>
                            <option value="CD"><?= ($mediaItem['mediaType'] === 'CD') ? 'selected' : '' ?>>CD</option>
                        </select>
                        <input type="number" id="year" name="year" min="1900" max="2024" placeholder="Year" value="<?=
                        htmlspecialchars($mediaItem['year']) ?>">
                        <input type="text" id="location" name="location" placeholder="Location" value="<?=
                        htmlspecialchars($mediaItem['location']) ?>">
                        <input type="submit" value="Update Media Item">
                    </div>
                </form>

                <?php if ($messageType === 'success'): ?>
                <p style="color: green;"><strong>Media item edited successfully!</strong></p>
                <?php elseif ($messageType === 'missing'): ?>
                <p style="color: red;"><strong>Please fill in required fields (Title, Artist, and Media Type).</strong></p>
                <?php elseif ($messageType === 'error'): ?>
                <p style="color: amber;"><strong>There was an error editing the media item</strong></p>
                <?php endif; ?>
            </div>
        </div>
    </body>
</html>

```

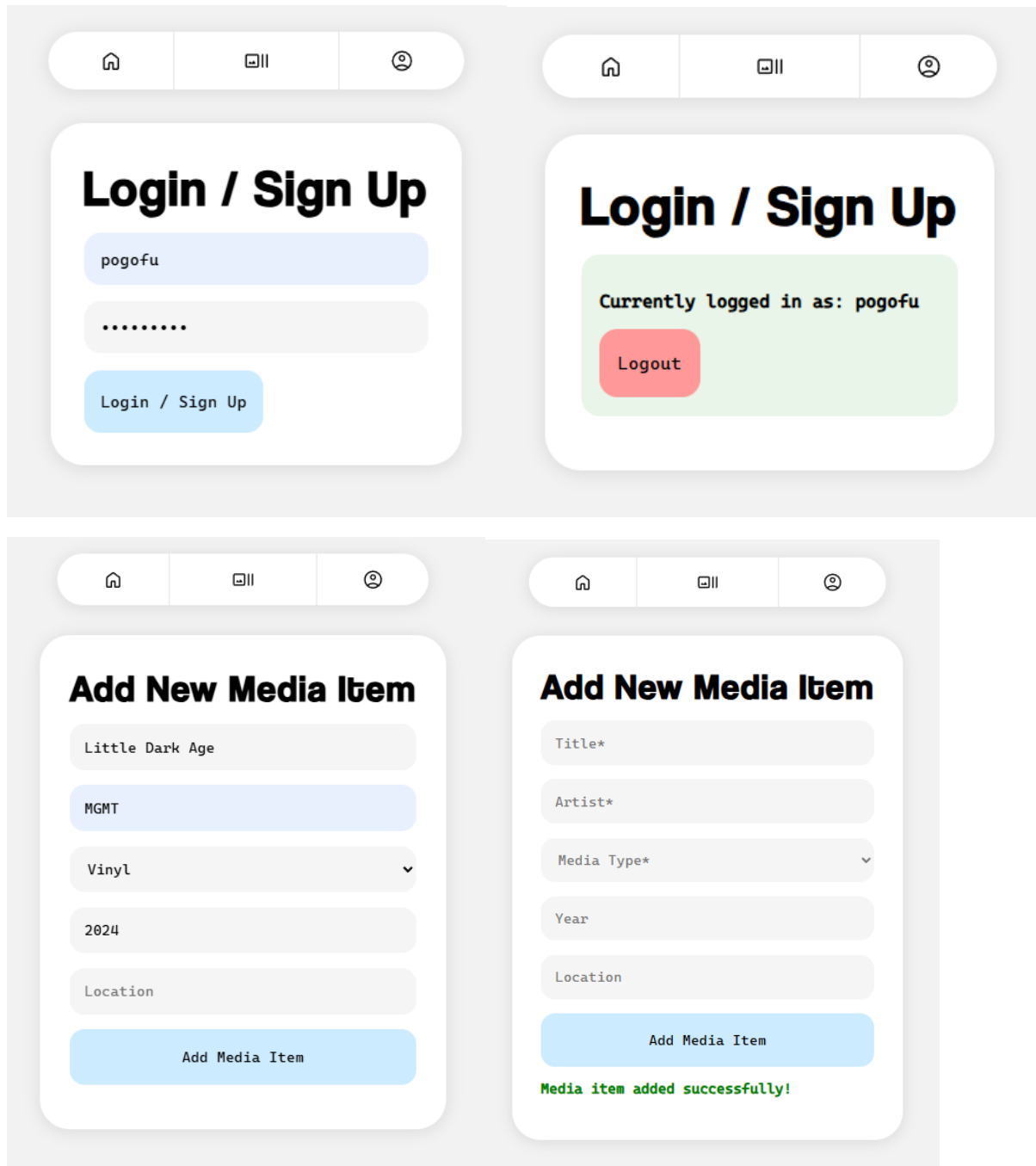
User interface implementation

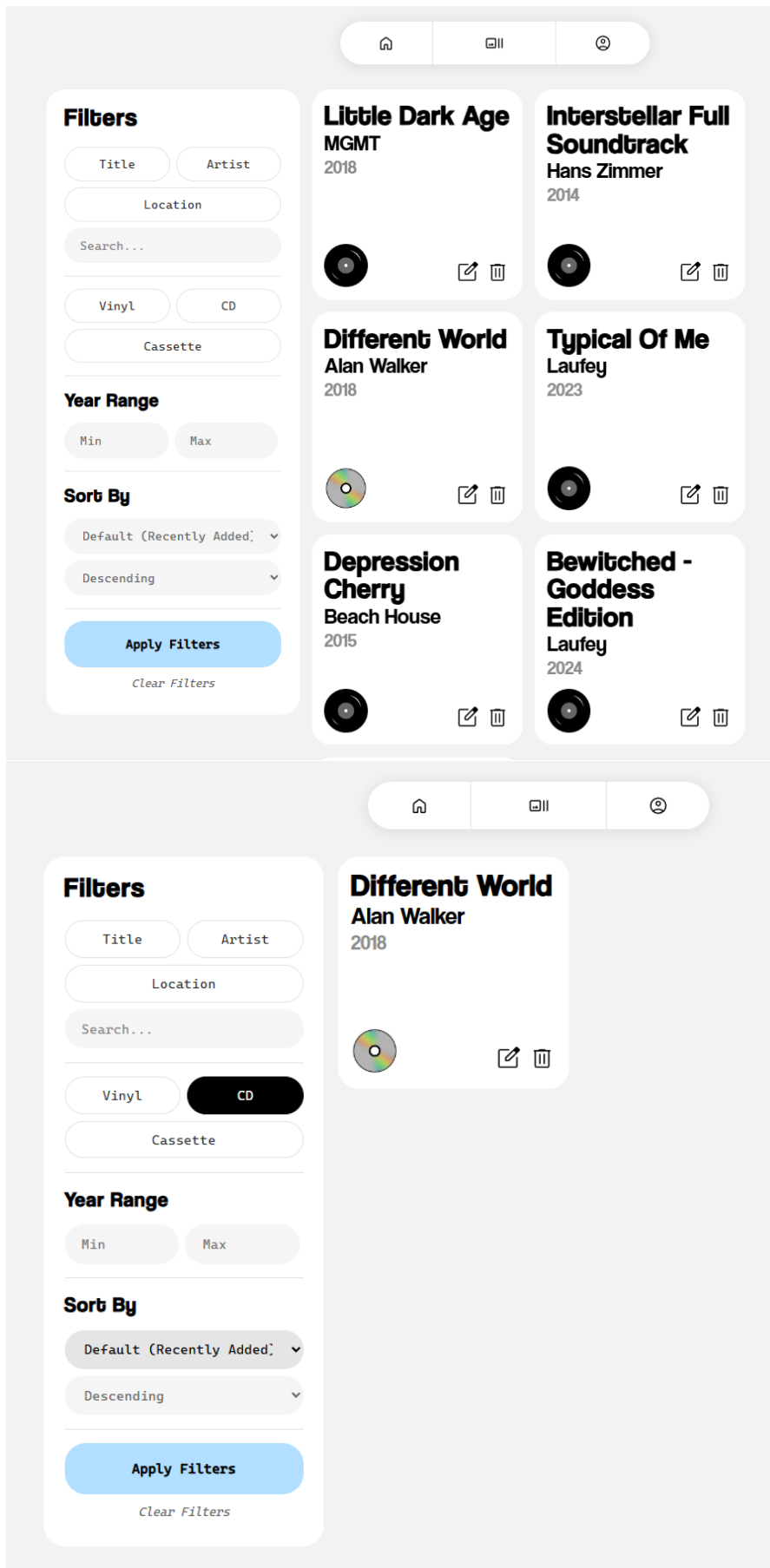
(3 marks)

Insert screenshot evidence of the implemented user interface including:

- input screens
- output showing results of processing

This evidence should be clearly labelled and should refer to relevant end-user and/or functional requirements.





Research and development of new skills and/or knowledge

(4 marks)

When implementing your functional requirements, you may need to make use of some coding that extends beyond the content of the Advanced Higher course. If this is the case, you will need to carry out some research.

Complete the table below to describe:

- ◆ any new skills and/or knowledge that you researched
- ◆ why those new skills and/or knowledge were necessary
- ◆ how you applied these new skills and/or knowledge to your project

You should reference the resources you used to research and develop these new skills and/or knowledge

New skill/knowledge	Tunnel my local host server to an online domain	
Functional Requirement(s) requiring new skill/knowledge	FR1 requires the forms and databases to be accessed from anywhere by anyone	
Why this new skill/knowledge was necessary	To allow everyone to access and use the website plus database from anywhere rather than just on my local device	
How you applied this new skill/knowledge to your project	I bought a domain through Cloudflare, set up the DNS settings, and used Cloudflared to tunnel traffic from the domain to my local server letting people access my project online.	
Resources used to research and develop this skill/knowledge (if required)	I followed Cloudflare's official documentation and online tutorials about Cloudflared and DNS setup. It took me through the auto DNS setup and Node.JS terminal control of the domain tunnelling. https://github.com/cloudflare/cloudflared	

Testing the solution (15 marks)

Once you have fully implemented your design, you must carry out final testing on your solution. This testing should be systematic and comprehensive, and based on a test plan.

This stage of your project should take around 8 hours.

Persona and test cases

(3 marks)

Describe the characteristics of one persona that you will adopt to test the end-user requirements.

A suitable persona for testing is Alex. A 24 year old music enthusiast who regularly buys physical media and wants an easy way to manage their collection. Alex has a solid grasp with everyday technology but not highly technical, so they expect simple navigation and clear instructions. They will test the ability to create and login to an account to access their personal collection, add new items using the media add form, and check that their full collection is displayed in a clear and well organised layout. Alex will also test searching by title or artist, sorting and filtering by media type such as vinyl, cassette or CD, editing or removing items when their collection changes. This ensures all end user requirements are usable for someone with average technical skills.

Describe the test cases that you will use to test each end-user requirement. Indicate the end-user requirement(s) that will be the focus of each test case.

Test case	Description each test case	End-user requirement(s) to be tested
Case 1	Test creating a new user account by entering valid details and ensuring the system logs the user in and displays their personal collection page.	EU1
Case 2	Test logging in with an existing account using correct credentials to confirm the user can access their saved collection.	EU1
Case 3	Use "Add Media" form to input a new items: title, artist, type, year and storage location. Then check that it is added to the database.	EU2
Case 4	After adding items test that the full collection can be viewed clearly in an organised layout with all details visible.	EU3
Case 5	Enter a title in the search bar and confirm that only matching items appear in the results.	EU4
Case 6	Enter an artist name in the search bar and confirm that matching items are displayed.	EU4
Case 7	Apply filters such as Vinyl, Cassette or CD and check that only items of the selected type are shown.	EU5
Case 8	Sort the collection by media type and ensure items are displayed in the correct grouped order.	EU5

Test case	Description each test case	End-user requirement(s) to be tested
Case 9	Select an existing item and change details such as title or year, then save and confirm the updated information is shown.	EU6
Case 10	Remove an item from the collection and confirm it no longer appears in the database or collection list.	EU6

Test the end-user requirements by carrying out the test cases above.

Describe the results of testing using the persona and test cases described above.

(150-300 words)

Using the persona of Alex who is a 25 year old music collector, each test case was done to evaluate how well the system meets the end user requirements. Testing started with account creation and login, where Alex successfully registered a new account logging in using the new credentials confirming that EU1 was met. When adding new physical media Alex used the input form to enter details such as: title, artist, type, year and storage location. All fields were accepted correctly, and the new item appeared in the collection showing that EU2 was met. Viewing the full collection also worked as intended as the interface organised items clearly and was easy for Alex to navigate completing EU3.

Searching the collection worked well. Searches by both title and artist returned accurate results without displaying unrelated items meeting EU4. Filtering and sorting by media type also allowed Alex to quickly narrow down the collection to specific formats such as vinyl or CD, and the sorted order displayed correctly showing that EU5 was met. Finally, Alex edited an item's details and deleted another media entry. Both changes updated in the database and collection view showing that EU6 was successfully met.

All test cases passed, and the system behaved as an average end user like Alex would expect. The solution meets all the defined end user requirements and provided a clear and usable experience for managing a physical media collection.

Comprehensive test plan

(6 marks)

Plan how you will carry out final testing of all functional requirements by completing the tables below.

Your plan should be comprehensive, to fully test each functional requirement identified at the analysis stage.

Test plan for final testing - functional requirements (AH concepts)

Test Number	Functional requirement to be tested	How it will be tested (including values if required)	Evidence required
Test 1	FR1	Navigate between all pages and submit the Add Media form using valid sample data (Title: Thriller, Artist: Michael Jackson, Type: Vinyl, Year: 1982, Location: Shelf A)	Screenshot of each page and successful form submissions
Test 2	FR2	Resize the browser to below the trigger width and check layout changes/query (font resizing, column collapse)	Screenshots of normal layout and small screen layout
Test 3	FR3	Submit the form with missing values incorrect types or empty fields. Verify server side validation messages appear.	Screenshot of error messages from PHP or HTML displayed on the page
Test 4	FR4	Log in and check that session variables (userID, username, login time) are created and persistent across pages.	Screenshots of session variable values printed for a test output

Test plan for final testing - functional requirements (integration)

Test Number	Functional requirement to be tested	How it will be tested (including values if required)	Evidence required
Test 5	FR5	Insert a new media item into the database and verify it is stored correctly with all required fields.	Database screenshot showing new row
Test 6	FR6	Do Insert, Select, Update and Delete actions on media items. Such as update “Thriller” year to 2082 or delete item	Screenshots of before and after database table and interface updates
Test 7	FR7	Use search/filter inputs and confirm that the output list is formatted and correctly filtered	Screenshot of search input and resulting organised list

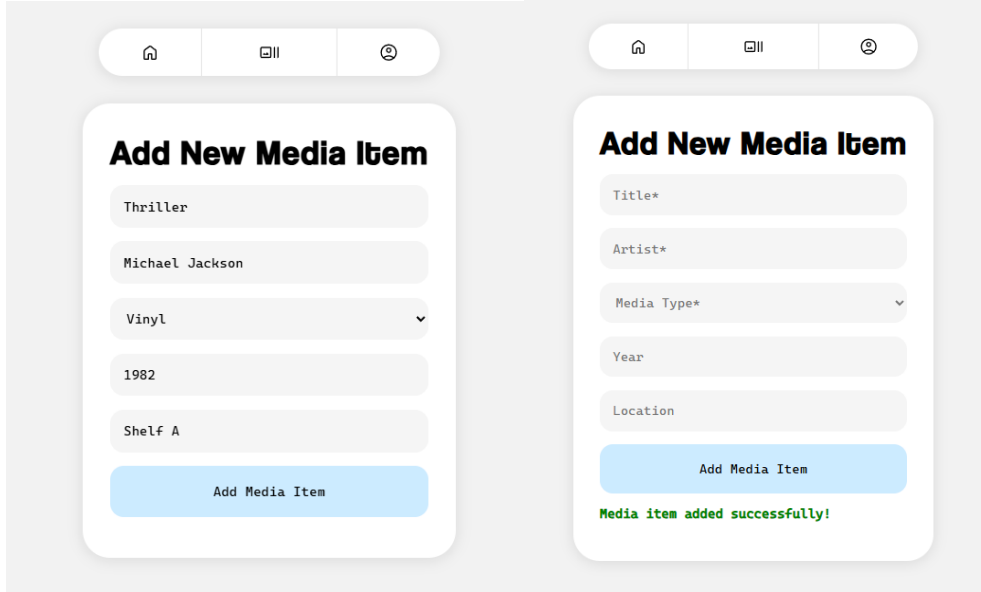
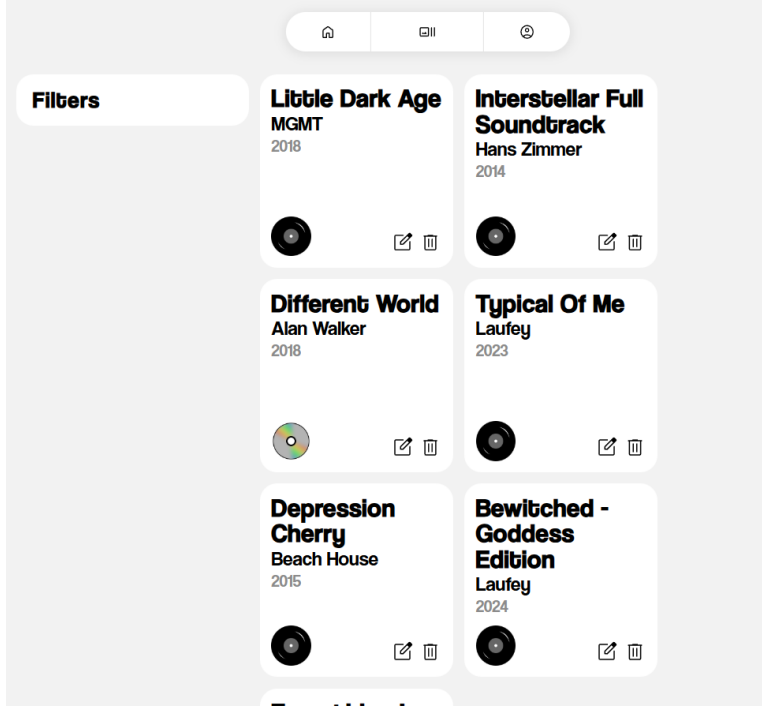
Test plan for final testing - additional functional requirements

Test Number	Functional requirement to be tested	How it will be tested (including values if required)	Evidence required
Test 8	FR8	Enter invalid years such as “abc”, “19CD” or symbols and verify the system rejects them	Error message screenshot refusing invalid input
Test 9	FR9	Try entering invalid formats such as DVD or Tape and check that only Vinyl, Cassette or CD are accepted.	Screenshot of validation message or dropdown restriction
Test 10	FR10	Search by title, artist or category and confirm accurate results verifying all matching items appear.	Screenshot of search results page
Test 11	FR11	Edit an item’s details and delete another item. Confirm the changes appear in the collection list.	Before and after screenshots showing updated or removed media

Evidence of requirements testing

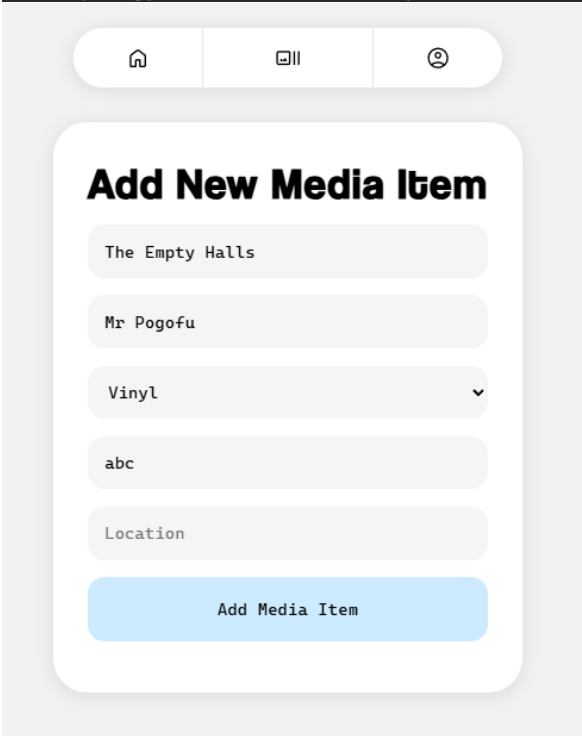
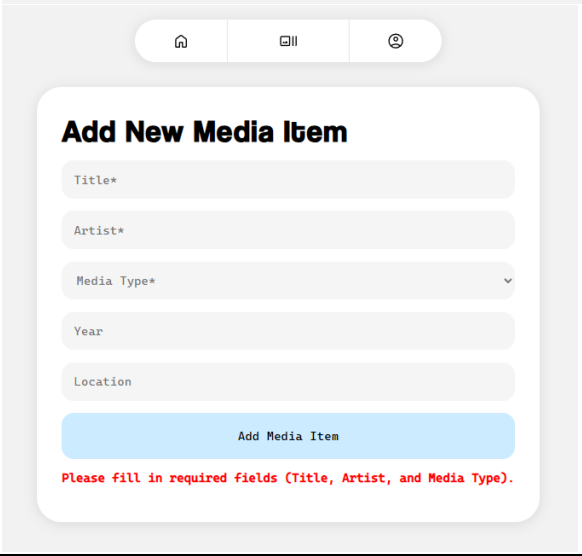
(6 marks)

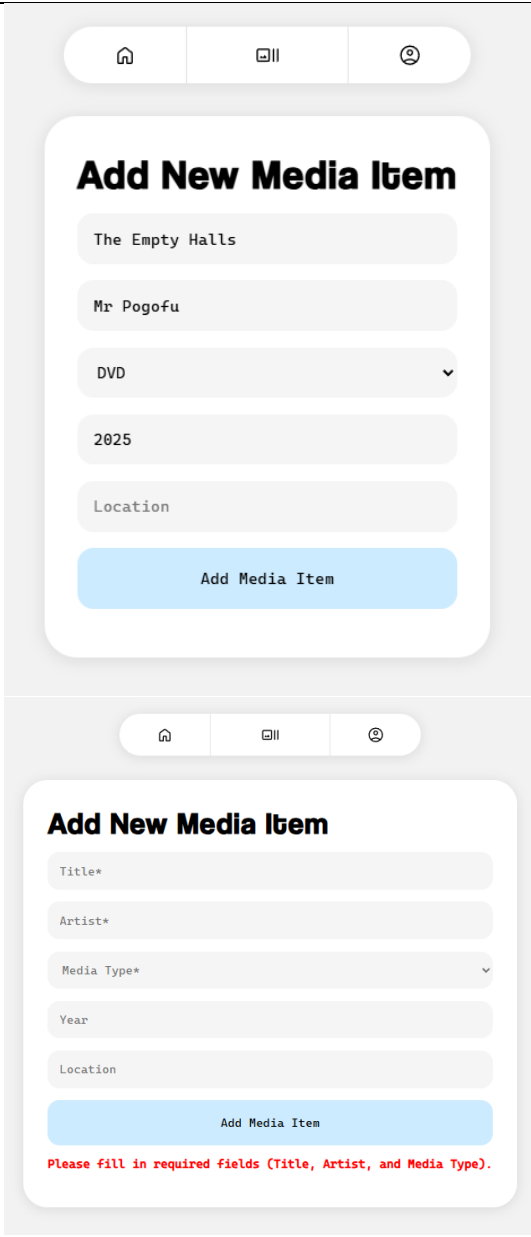
Test the functional requirements by carrying out the test plan above.
For each test, provide screenshot evidence of your testing.

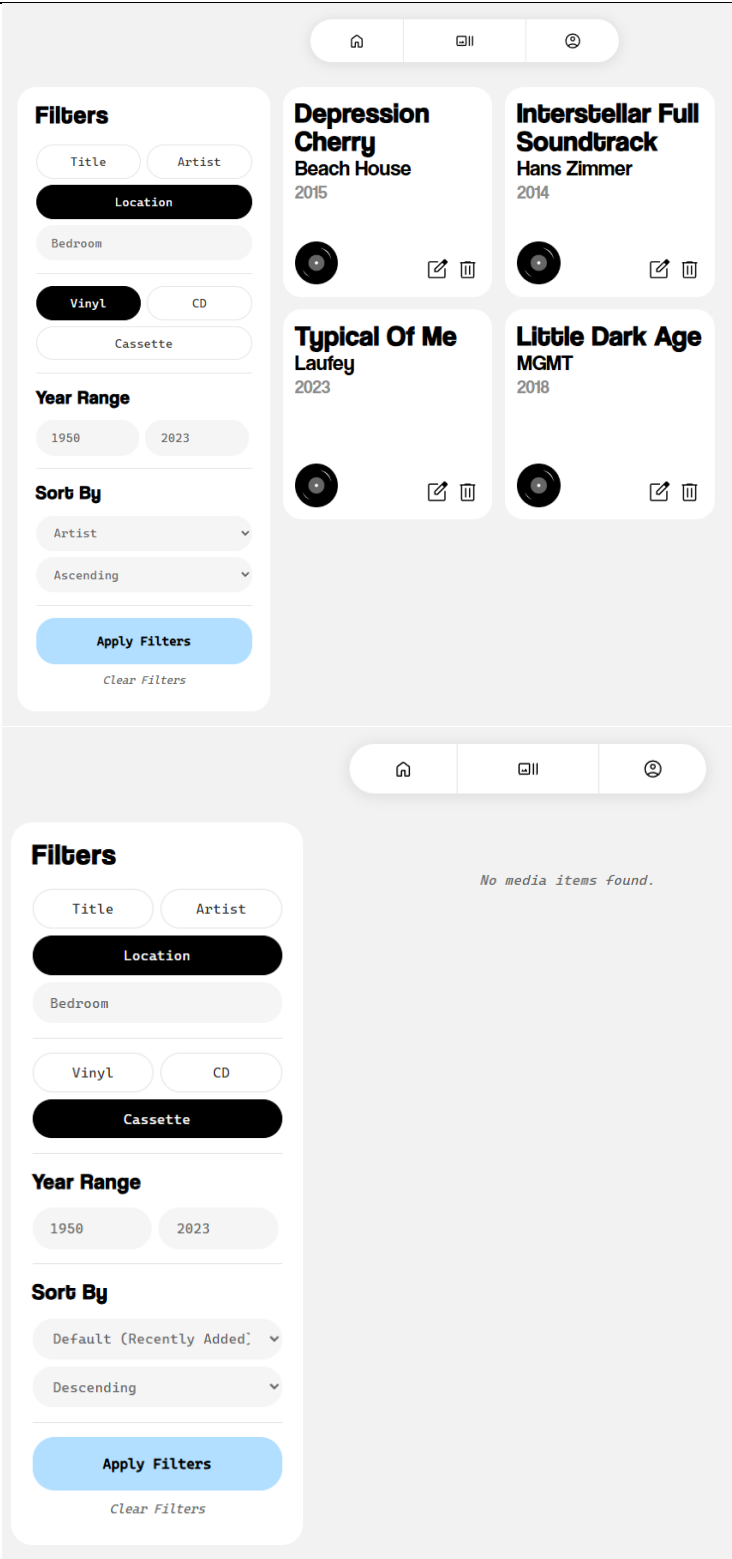
Test Number	Screenshot evidence
Test 1	
Test 2	

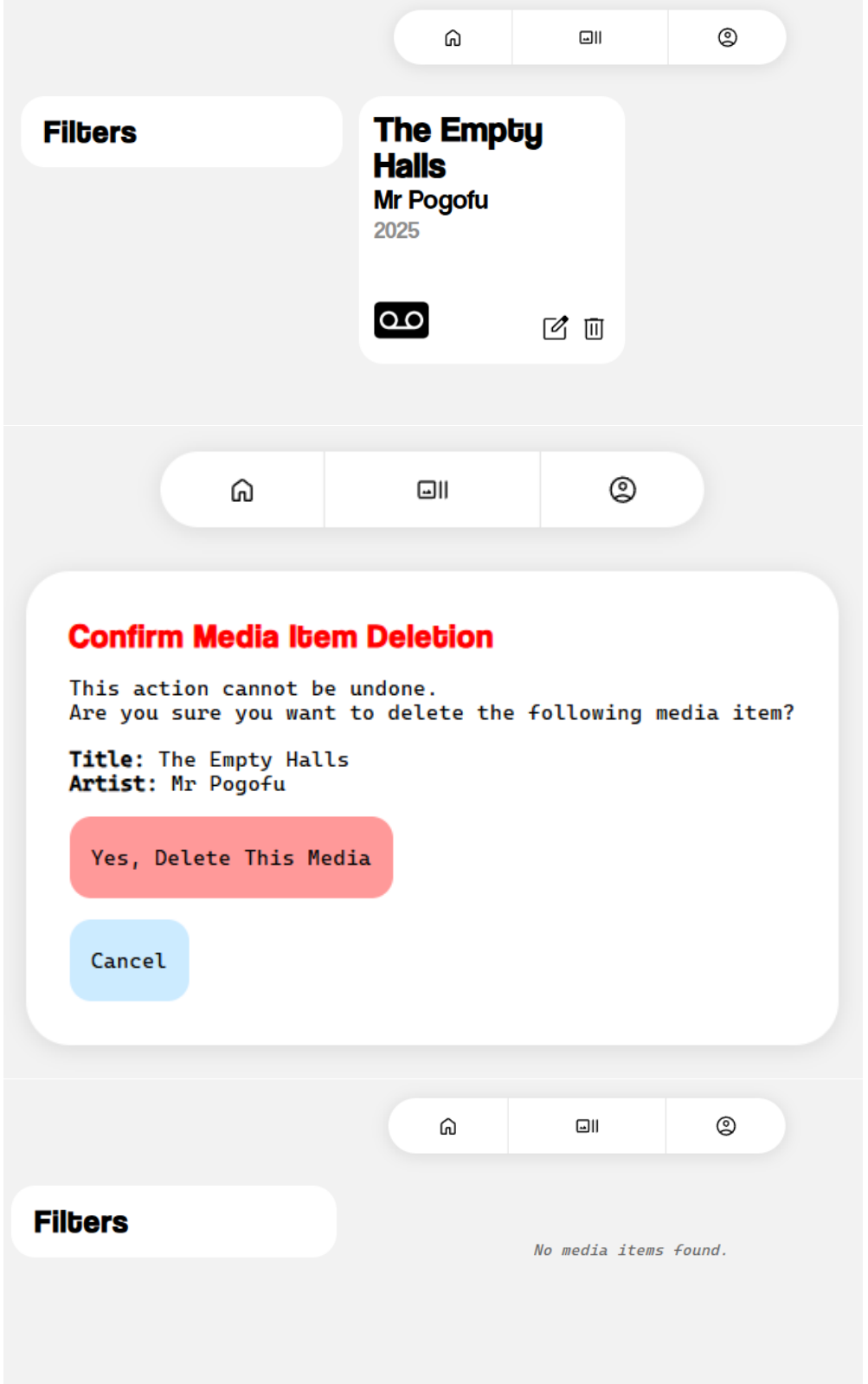
Test Number	Screenshot evidence																												
Test 3																													
Test 4																													
Test 5	<table><thead><tr><th>userID</th><th>mediaID</th><th>title</th><th>artist</th><th>mediaType</th><th>year</th><th>location</th></tr></thead><tbody><tr><td>3</td><td>26</td><td>Infinite Amethyst</td><td>Mr Pogofu</td><td>CD</td><td>2023</td><td></td></tr><tr><td>3</td><td>27</td><td>Control Within the Collapse</td><td>Mr Pogofu</td><td>Vinyl</td><td>2025</td><td></td></tr><tr><td>6</td><td>28</td><td>Thriller</td><td>Michael Jackson</td><td>Vinyl</td><td>1982</td><td>Shelf A</td></tr></tbody></table>	userID	mediaID	title	artist	mediaType	year	location	3	26	Infinite Amethyst	Mr Pogofu	CD	2023		3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025		6	28	Thriller	Michael Jackson	Vinyl	1982	Shelf A
userID	mediaID	title	artist	mediaType	year	location																							
3	26	Infinite Amethyst	Mr Pogofu	CD	2023																								
3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025																								
6	28	Thriller	Michael Jackson	Vinyl	1982	Shelf A																							

Test Number	Screenshot evidence																																																																													
Test 6	<table><thead><tr><th>userID</th><th>mediaID</th><th>title</th><th>artist</th><th>mediaType</th><th>year</th><th>location</th></tr></thead><tbody><tr><td>3</td><td>26</td><td>Infinite Amethyst</td><td>Mr Pogofu</td><td>CD</td><td>2023</td><td></td></tr><tr><td>3</td><td>27</td><td>Control Within the Collapse</td><td>Mr Pogofu</td><td>Vinyl</td><td>2025</td><td></td></tr><tr><td>6</td><td>28</td><td>Thriller</td><td>Michael Jackson</td><td>Vinyl</td><td>1982</td><td>Shelf A</td></tr></tbody></table> Edit <table><thead><tr><th>userID</th><th>mediaID</th><th>title</th><th>artist</th><th>mediaType</th><th>year</th><th>location</th></tr></thead><tbody><tr><td>3</td><td>26</td><td>Infinite Amethyst</td><td>Mr Pogofu</td><td>CD</td><td>2023</td><td></td></tr><tr><td>3</td><td>27</td><td>Control Within the Collapse</td><td>Mr Pogofu</td><td>Vinyl</td><td>2025</td><td></td></tr><tr><td>6</td><td>28</td><td>Thriller</td><td>Michael Jackson</td><td>Vinyl</td><td>2082</td><td>Shelf A</td></tr></tbody></table> Delete <table><thead><tr><th>userID</th><th>mediaID</th><th>title</th><th>artist</th><th>mediaType</th><th>year</th><th>location</th></tr></thead><tbody><tr><td>3</td><td>26</td><td>Infinite Amethyst</td><td>Mr Pogofu</td><td>CD</td><td>2023</td><td></td></tr><tr><td>3</td><td>27</td><td>Control Within the Collapse</td><td>Mr Pogofu</td><td>Vinyl</td><td>2025</td><td></td></tr></tbody></table>	userID	mediaID	title	artist	mediaType	year	location	3	26	Infinite Amethyst	Mr Pogofu	CD	2023		3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025		6	28	Thriller	Michael Jackson	Vinyl	1982	Shelf A	userID	mediaID	title	artist	mediaType	year	location	3	26	Infinite Amethyst	Mr Pogofu	CD	2023		3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025		6	28	Thriller	Michael Jackson	Vinyl	2082	Shelf A	userID	mediaID	title	artist	mediaType	year	location	3	26	Infinite Amethyst	Mr Pogofu	CD	2023		3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025	
userID	mediaID	title	artist	mediaType	year	location																																																																								
3	26	Infinite Amethyst	Mr Pogofu	CD	2023																																																																									
3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025																																																																									
6	28	Thriller	Michael Jackson	Vinyl	1982	Shelf A																																																																								
userID	mediaID	title	artist	mediaType	year	location																																																																								
3	26	Infinite Amethyst	Mr Pogofu	CD	2023																																																																									
3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025																																																																									
6	28	Thriller	Michael Jackson	Vinyl	2082	Shelf A																																																																								
userID	mediaID	title	artist	mediaType	year	location																																																																								
3	26	Infinite Amethyst	Mr Pogofu	CD	2023																																																																									
3	27	Control Within the Collapse	Mr Pogofu	Vinyl	2025																																																																									
Test 7	Filters TitleArtist Location Search... VinylCD Cassette Year Range MinMax Sort By Default (Recently Added)Descending Apply Filters Clear Filters Different World Alan Walker 2018 Filters TitleArtist Location Laufey VinylCD Cassette Year Range MinMax Sort By Default (Recently Added)Descending Apply Filters Clear Filters Typical Of Me Laufey 2023 Bewitched - Goddess Edition Laufey 2024 Everything I Know About Love Laufey 2024																																																																													
Test 8	Changed input field to accept text via the web developer tools																																																																													

Test Number	Screenshot evidence
	<pre> <select id="mediaType" name="mediaType" required>...</select> <input type="text" id="year" name="year" placeholder="Year" min="1900" max="3000"> == \$t <input type="text" id="location" placeholder="Location" name="location"> </pre>  
Test 9	Added new media types via web developer tools <pre> <datalist id="artist">...</datalist> <select id="mediaType" name="mediaType" required> == \$0 <option value="Media Type*">Media Type*</option> <option value="Vinyl">Vinyl</option> <option value="Cassette">Cassette</option> <option value="CD">CD</option> <option value="DVD">DVD</option> <option value="Tape">Tape</option> </select> <input type="number" id="year" name="year" placeholder="Ye </pre>

Test Number	Screenshot evidence
	 The image displays two screenshots of a mobile application interface for adding a new media item. Both screenshots show a form titled "Add New Media Item" with a navigation bar at the top containing icons for home, a list, and a profile. The top screenshot shows the form with the following pre-filled data: - Title: The Empty Halls - Artist: Mr Pogofu - Media Type: DVD (selected from a dropdown menu) - Year: 2025 - Location: (empty) The bottom screenshot shows the form with the following fields: - Title* - Artist* - Media Type* (selected from a dropdown menu) - Year - Location A red error message is displayed at the bottom of the form in the bottom screenshot: "Please fill in required fields (Title, Artist, and Media Type)".

Test Number	Screenshot evidence
Test 10	 The top screenshot displays a music application interface with a sidebar filter menu and a main content area showing four music items. The filter menu includes sections for 'Filters' (with buttons for Title, Artist, Location, Vinyl, CD, and Cassette), 'Year Range' (with buttons for 1950 and 2023), and 'Sort By' (with dropdowns for Artist and Ascending). The main content area shows four music items, each with a vinyl record icon, an edit icon, and a delete icon. The bottom screenshot shows the same interface but with the message 'No media items found.' displayed in the center, indicating that the filters applied in the top screenshot have resulted in no items being found.

Test Number	Screenshot evidence
Test 11	 The evidence consists of three sequential screenshots of a mobile application interface: Top Screenshot: Displays a media item card titled "The Empty Halls" by "Mr Pogofu" from the year "2025". The card includes a media icon (two eyes) and action icons for editing (pencil) and deleting (trash can). Middle Screenshot: Shows a "Confirm Media Item Deletion" dialog box. It contains the text: "This action cannot be undone. Are you sure you want to delete the following media item?" followed by the item details: "Title: The Empty Halls" and "Artist: Mr Pogofu". There are two buttons: "Yes, Delete This Media" (red) and "Cancel" (blue). Bottom Screenshot: Shows the application after deletion. A "Filters" button is visible on the left, and the text "No media items found." is displayed in the center of the screen.

Evaluation (5 marks)

The evaluation stage of your project should take around 2 hours.

Fitness for purpose

(3 marks)

Describe how your solution matches all requirements in your requirements specification. (400 to 600 words)

The final solution meets all requirements set out in the requirements specification, completing both end user needs and functional requirements.

To meet EU1 my solution includes a complete account system that allows users to register a new account and log in or out. Once logged in session variables store themselves to be used later. This also satisfies FR4 as PHP sessions are used to store the userID, username and login time maintaining login status throughout the entire website.

EU2 is completed by a dedicated “Add Media” page containing a HTML form. This form collects the required fields: title, type, year, and storage location. The form uses correct HTML attributes (method, name) meeting FR1. When submitted the PHP processes the input, validates it, and inserts it into the database. This also meets FR3 as PHP checks form variable values before processing.

The solution meets EU3 by displaying the user’s full physical media collection in a clear format. Data is dynamically retrieved from the database which completes FR6. The organisation and layout are formatted for readability and consistency.

EU4 and FR10 are achieved via a search bar that allows users to search by various attributes. The takes the user’s search query to PHP which performs a database lookup and displays matching items. This search page also meets FR7 which requires an interface to accept query inputs and display formatted results.

EU5 and part of FR7 are done through filter and sorting that allow users to view only items of a specific media type (vinyl, cassette, or CD). The solution uses SQL queries that return items of the chosen type. This relies on correct validation, which also meets FR9 ensuring the media format field only accepts valid formats.

Editing and deleting entries required by EU6 and FR11 are added via “Edit” and “Delete” buttons beside each media item. Users can update any field or remove entries. PHP processes these actions and updates the database, completing FR6.

It also has a structured physical media database table that stores all the data such as: title, artist, type, year, and storage location. Which meets FR5. PHP queries handle inserting, updating, selecting and deleting records.

The solution also meets FR8 as it validates that the year field accepts to only numerical values. This prevents invalid characters from being entered into the database. Additional validation ensures that only valid media formats are accepted preventing incorrect categories.

Finally, FR2 is satisfied through the inclusion of CSS media queries that adjust the layout for smaller screens. When the browser width drops below the set threshold, elements rearrange into a more mobile friendly layout, ensuring usability on phones and tablets. Overall the finished solution fully meets all end user requirements and functional requirements. It provides an accessible, and efficient way for users to manage their physical media collection.

Describe the results of your testing. You should reflect on what happened during final testing of functional requirements, discuss what went well and identify any tests that didn't work as expected. (200-300 words)

The final testing showed that most functional requirements operated as expected and the system behaved reliably across all scenarios. Test 1 confirmed that the website structure and navigation are fully functional and work. All the pages loaded correctly and the "Add Media" form submitted the valid sample data showing that the basic form and page met expectations. Test 2 also worked well with the media query triggering correctly when the browser width was reduced causing the layout to adapt correctly for mobile screens.

Validation tests (Tests 3, 8, 9) were largely successful. Submitting incomplete or incorrect data correctly showed error messages preventing invalid entries from the database. The year validation correctly rejected non numerical values, and the media type field only allowed input of: vinyl, cassette or CD. These tests confirmed that backend validation operated correctly, though early in implementation some data would just get let through even after being validated. These issues were fixed.

Session handling in Test 4 worked correctly with session variables being created and retained across pages after login.

Database integration tests (Tests 5, 6, 7, 10, 11) showed consistent results. New media items were added correctly, updates were done correct, and deletions removed records both from the interface and database. Search and filtering worked well, returning the correct organised results. One minor issue found during testing was that when the user reloaded the add page after submitting the item it would repeatedly add the same item. I fixed this by redirecting to the same page without "POST".

Overall, the system passed all major tests and any issues encountered were fixed showing that the solution is fit for purpose.

Future maintainability

(1 mark)

Describe how your code allows for future maintenance. (150-250 words)

The solution has been designed with future maintenance in mind making sure that new features, improvements, or fixes can be implemented without significant reconstruction of the code. The project file are organised into separate PHP files for key functions such as login, adding media, editing media, and database connecting. This structure means that changes can be made to one part of the system without affecting the others. For

example, updates to the database connection only require modifications to a single file rather than every script that connects with the database.

Consistent naming schemes are used throughout the project for: variables, functions, form fields, and database columns which makes the code easier to understand and reduces the chance of errors during future edits. SQL queries have been written in a clear way making it straight forward to add new fields or change table layouts.

Finally, the HTML forms CSS media queries and standard PHP functions ensures that the system is built on predefined functions that are documented officially. This makes the solution easier to maintain and extend.

Robustness

(1 mark)

Describe the robustness of your solution. (150-250 words)

My solution is robust because it handles incorrect inputs, user mistakes and database errors in a safe way. The system uses validation on both the HTML form and the PHP script so invalid data is stopped before it reaches the database. This includes preventing letters in the year field or rejecting formats that are not vinyl, cassette or CD. This keeps the database clean and stops broken entries from being added.

Session handling also adds to the reliability of the system. When a user logs in the session stores the userID, username and login time. These are checked on every page to make sure the user stays logged in and cannot access pages they should not see. If the session is missing or expired the system safely sends the user back to the login page.

Database actions such as adding, updating and deleting media items are checked before running so the system avoids errors. Clear feedback messages help users fix mistakes without stopping the website from working. The CSS layout also adjusts correctly on different screen sizes so the site stays usable even on unexpected devices.

Overall, the system handles errors well and continues to work smoothly in normal and unusual situations.

Appendix

```

account_settings.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$action = $_GET['action'] ?? '';

$messageType = '';

if ($action === 'change_username' && $_POST) {
    $newUsername = trim($_POST['change_username'] ?? '');
    $newUsername = substr($newUsername, 0, 25);
    if ($newUsername) {
        $stmt = $pdo->prepare("UPDATE login_details SET username = ? WHERE userID = ?");
        $stmt->execute([$newUsername, $_SESSION['user_id']]);
        if ($stmt->rowCount() > 0) {
            $_SESSION['username'] = $newUsername;
            $messageType = 'username_success';
        } else {
            $messageType = 'username_error';
        }
    }
}

elseif ($action === 'change_password' && $_POST) {
    $newPassword = $_POST['change_password'] ?? '';
    if ($newPassword) {
        $hashedPassword = password_hash($newPassword, PASSWORD_DEFAULT);
        $stmt = $pdo->prepare("UPDATE login_details SET password = ? WHERE userID = ?");
        $stmt->execute([$hashedPassword, $_SESSION['user_id']]);
        if ($stmt->rowCount() > 0) {
            $messageType = 'password_success';
        } else {
            $messageType = 'password_error';
        }
    }
}

elseif ($action === 'delete_account') {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $stmt = $pdo->prepare("DELETE FROM media_items WHERE userID = ?");
        $stmt->execute([$_SESSION['user_id']]);

        $stmt = $pdo->prepare("DELETE FROM login_details WHERE userID = ?");
        $stmt->execute([$_SESSION['user_id']]);

        session_destroy();
        header("Location: login_details.php");
        exit;
    } else {
        echo "<title>Confirm Account Deletion</title>";
        echo "<link rel='stylesheet' href='resources/main.css'>";
        echo "<link rel='stylesheet' href='resources/color-scheme.css'>";
        echo "<div class='centered-container' style='height:100%'>";
        echo "<div class='content-box'>";
        echo "<h2 style='color:red;'>Confirm Account Deletion</h2>";
        echo "<p>This action cannot be undone. Are you sure you want to delete your account?</p>";
        echo "<form method='POST' action='\" . htmlspecialchars($_SERVER['PHP_SELF']) . \"?action=delete_account'>";
        echo "    <input type='submit' value='Yes, Delete My Account' class='warning_button'>";
        echo "</form>";
        echo "<a href='\" . htmlspecialchars($_SERVER['PHP_SELF']) . \"'>";
        echo "    <button>Cancel</button>";
        echo "</a>";
        echo "</div>";
        echo "</div>";
        exit;
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Account Settings</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageTop icon">

```

```

<link rel="icon" type="image/x-icon" href="resources/pageicon.png" /
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="content-box">
                <h1>Account Settings</h1>
                <h2 style="margin-bottom:16px">Current Username: <?= htmlspecialchars($_SESSION['username']) ?></h2>

                <div class="input-group">
                    <form method="POST" action="<?= htmlspecialchars($_SERVER['PHP_SELF']) ?>?action=change_username"
class="input-button-pair">
                        <input type="text" id="change_username" maxlength="25" placeholder="Change Username"
name="change_username" required><br><br>
                        <input type="submit" value="Change">
                    </form>

                    <form method="POST" action="<?= htmlspecialchars($_SERVER['PHP_SELF']) ?>?action=change_password"
class="input-button-pair">
                        <input type="password" id="change_password" placeholder="Change Password" name="change_password"
required><br><br>
                        <input type="submit" value="Change">
                    </form>
                </div>

                <a href="<?= htmlspecialchars($_SERVER['PHP_SELF']) ?>?action=delete_account" style="color:red;">
                    <button class="warning_button" style="width:100%">Delete Account</button>
                </a>

                <?php if ($messageType === 'username_success'): ?>
                    <p style="color: green;"><strong>Username changed successfully!</strong></p>
                <?php elseif ($messageType === 'username_error'): ?>
                    <p style="color: red;"><strong>Failed to change username. It might be already taken.</strong></p>
                <?php elseif ($messageType === 'password_success'): ?>
                    <p style="color: green;"><strong>Password changed successfully!</strong></p>
                <?php elseif ($messageType === 'password_error'): ?>
                    <p style="color: red;"><strong>Failed to change password.</strong></p>
                <?php endif; ?>
            </div>
        </div>
    </div>
</body>
</html>

```

```

collection_stats.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Your Collection Stats</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body" style="display: flex; justify-content: center;">
        <div class="stats-container">
            <div class="stat-card total-card">
                <?php
                $stmt = $pdo->prepare("SELECT COUNT(*) AS total FROM media_items WHERE userID = ?");
                $stmt->execute([$SESSION['user_id']]);
                $total = $stmt->fetch(PDO::FETCH_ASSOC);
                ?>
                <div id="collection-total-title" class="stat-label">Collection Total: <div class="stat-number"><?=$total['total'] ?? 0 ?></div></div>

            </div>

            <div class="stats-row">
                <div id="average-year" class="stat-card">
                    <?php
                    $stmt = $pdo->prepare("SELECT AVG(year) AS average_year FROM media_items WHERE userID = ? AND year IS NOT
                    NULL");
                    $stmt->execute([$SESSION['user_id']]);
                    $avgYear = $stmt->fetch(PDO::FETCH_ASSOC);
                    ?>
                    <div class="stat-label">Average Year:</div>
                    <?php if ($avgYear && $avgYear['average_year']): ?>
                        <div class="stat-number blue"><? = round($avgYear['average_year']) ?></div>
                    <?php else: ?>
                        <div class="stat-number blue">N/A</div>
                    <?php endif; ?>
                </div>

                <div class="stat-card">
                    <?php
                    $stmt = $pdo->prepare("SELECT mediaType, COUNT(*) AS count FROM media_items WHERE userID = ? GROUP BY
                    mediaType ORDER BY count DESC LIMIT 1");
                    $stmt->execute([$SESSION['user_id']]);
                    $topMedia = $stmt->fetch(PDO::FETCH_ASSOC);
                    ?>
                    <div id="top-media-title" class="stat-label">Top Media:
                    <?php if ($topMedia): ?>
                        <div class="media-icon">
                            <?php
                            $mediaType = strtolower($topMedia['mediaType']);
                            if ($mediaType === 'cassette'): ?>
                                
                            <?php elseif ($mediaType === 'cd'): ?>
                                
                            <?php elseif ($mediaType === 'vinyl'): ?>
                                
                            <?php else: ?>
                                <div class="generic-icon">?</div>
                            <?php endif; ?>
                        </div>
                    <?php else: ?>
                        <div class="media-icon">
                            <div class="generic-icon">?</div>
                        </div>
                    <?php endif; ?>
                </div>
            </div>
        </div>
    </div>

```

```

        <?php echo $topArtist; </div>
    </div>
</div>

<div id="top-artist" class="stat-card">
    <?php
        $stmt = $pdo->prepare("SELECT artist, COUNT(*) AS count FROM media_items WHERE userID = ? GROUP BY artist
ORDER BY count DESC LIMIT 1");
        $stmt->execute([$$_SESSION['user_id']]);
        $topArtist = $stmt->fetch(PDO::FETCH_ASSOC);
    >
    <div class="stat-label">Top Artist:</div>
    <?php if ($topArtist): ?>
        <div class="artist-name"><?= htmlspecialchars($topArtist['artist']) ?></div>
    <?php else: ?>
        <div class="artist-name">No data</div>
    <?php endif; ?>
</div>
</div>
</body>
</html>

```



collection_view.php

```
<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$user = getCurrentUser();

$searchField = $_GET['searchField'] ?? '';
$searchValue = $_GET['search'] ?? '';
$mediaTypeFilter = $_GET['mediaType'] ?? '';
$yearMin = $_GET['yearMin'] ?? '';
$yearMax = $_GET['yearMax'] ?? '';
$sortBy = $_GET['sort'] ?? 'mediaID';
$sortOrder = $_GET['order'] ?? 'DESC';

$query = "SELECT * FROM media_items WHERE userID = ?";
$params = [$user['user_id']];

if (!empty($searchValue) && !empty($searchField)) {
    if ($searchField === 'title') {
        $query .= " AND title LIKE ?";
        $params[] = '%' . $searchValue . '%';
    } elseif ($searchField === 'artist') {
        $query .= " AND artist LIKE ?";
        $params[] = '%' . $searchValue . '%';
    } elseif ($searchField === 'location') {
        $query .= " AND location LIKE ?";
        $params[] = '%' . $searchValue . '%';
    }
}

if (!empty($mediaTypeFilter)) {
    $query .= " AND mediaType = ?";
    $params[] = $mediaTypeFilter;
}

if (!empty($yearMin)) {
    $query .= " AND year ≥ ?";
    $params[] = (int)$yearMin;
}

if (!empty($yearMax)) {
    $query .= " AND year ≤ ?";
    $params[] = (int)$yearMax;
}

$allowedSortFields = ['mediaID', 'title', 'artist', 'year', 'mediaType'];
$allowedSortOrder = ['ASC', 'DESC'];

if (!in_array($sortBy, $allowedSortFields)) {
    $sortBy = 'mediaID';
}

if (!in_array($sortOrder, $allowedSortOrder)) {
    $sortOrder = 'DESC';
}

$query .= " ORDER BY $sortBy $sortOrder";

$stmt = $pdo->prepare($query);
$stmt->execute($params);
$results = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Media Items Manager</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">

    <script>
        function toggleFilters() {
```

```

function toggleFilters() {
    const filter_cont = document.getElementById("filters-container")
    if (filter_cont.style.maxHeight === '40px'){
        filter_cont.style.maxHeight = '1000px'
    }else{
        filter_cont.style.maxHeight = '40px'
    }
}
</script>
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div id="collections-filters">
            <div id="collection-container">
                <?php if ($results): ?>
                    <div class="collection-grid">
                        <?php foreach ($results as $row): ?>
                            <div class="collection-widget">
                                <div class="media-title"><?php echo htmlspecialchars($row['title']); ?></div>
                                <div class="media-artist"><?php echo htmlspecialchars($row['artist']); ?></div>
                                <div class="media-year"><?php echo htmlspecialchars($row['year']); ?></div>

                                <div class="media-actions">
                                    <div class="media-type-container">
                                        <?php
                                            $mediaType = strtolower($row['mediaType']);
                                            if ($mediaType === 'cassette'): ?>
                                                

                                                <?php elseif ($mediaType === 'cd'): ?>
                                                    
                                                <?php elseif ($mediaType === 'vinyl'): ?>
                                                    

                                                <?php endif; ?>
                                            </div>

                                            <div class="action-links">
                                                <a href="media_edit.php?id=<?php echo htmlspecialchars($row['mediaID']); ?>"></a>
                                                <a href="media_delete.php?mediaID=<?php echo htmlspecialchars($row['mediaID']); ?>"></a>
                                            </div>
                                        </div>
                                    </div>
                                <?php endforeach; ?>
                            </div>
                        <?php else: ?>
                            <div class="no-results">
                                <p>No media items found.</p>
                            </div>
                        <?php endif; ?>
                    </div>

                    <?php
                        $filtersOpen = !empty($searchField) || !empty($mediaTypeFilter) || !empty($yearMin) || !empty($yearMax) ||
!empty($searchValue) || $sortBy === 'mediaID' || $sortOrder === 'DESC';
                    ?>

                    <div id="filters-container" style="max-height:<?=$filtersOpen ? '1000px' : '40px' ?>">
                        <h2 style="cursor:pointer;" onclick="toggleFilters()">Filters</h2>
                        <form method="GET">
                            <div class="inline-inputs">
                                <label><input type="radio" name="searchField" value="title" <?php echo $searchField === 'title' ?
'checked' : ''; ?>> Title</label>
                                <label><input type="radio" name="searchField" value="artist" <?php echo $searchField === 'artist' ?
'checked' : ''; ?>> Artist</label>
                                <label><input type="radio" name="searchField" value="location" <?php echo $searchField === 'location'
? 'checked' : ''; ?>> Location</label>
                                <input type="text" name="search" placeholder="Search ..." value="<?php echo
htmlspecialchars($searchValue); ?>">
                            </div>

                            <div class="inline-inputs">
                                <label><input type="radio" name="mediaType" value="Vinyl" <?php echo $mediaTypeFilter === 'Vinyl' ?
'checked' : ''; ?>> Vinyl</label>
                                <label><input type="radio" name="mediaType" value="CD" <?php echo $mediaTypeFilter === 'CD' ?
'checked' : ''; ?>> CD</label>

```

```

        <label><input type="radio" name="mediaType" value="Cassette" <?php echo $mediaTypeFilter ===
'Cassette' ? 'checked' : ''; ?> Cassette</label>
    </div>

    <h3 style="margin:0; margin-bottom:16px;">Year Range</h3>
    <div class="inline-inputs">
        <input type="number" name="yearMin" placeholder="Min" value="<?php echo htmlspecialchars($yearMin); ?
>">
        <input type="number" name="yearMax" placeholder="Max" value="<?php echo htmlspecialchars($yearMax); ?
>">
    </div>

    <h3 style="margin:0; margin-bottom:16px;">Sort By</h3>
    <div class="inline-inputs">
        <select name="sortBy">
            <option value="mediaID" <?= $sortBy === 'mediaID' ? 'selected' : '' ?>>Default (Recently Added)
</option>
            <option value="title" <?= $sortBy === 'title' ? 'selected' : '' ?>>Title</option>
            <option value="artist" <?= $sortBy === 'artist' ? 'selected' : '' ?>>Artist</option>
            <option value="year" <?= $sortBy === 'year' ? 'selected' : '' ?>>Year</option>
            <option value="mediaType" <?= $sortBy === 'mediaType' ? 'selected' : '' ?>>Media Type</option>
        </select>

        <select name="order">
            <option value="ASC" <?= $sortOrder === 'ASC' ? 'selected' : '' ?>>Ascending</option>
            <option value="DESC" <?= $sortOrder === 'DESC' ? 'selected' : '' ?>>Descending</option>
        </select>
    </div>

    <div>
        <button type="submit">Apply Filters</button>
        <a href="<?php echo $_SERVER['PHP_SELF']; ?>">Clear Filters</a>
    </div>
</form>
</div>
</div>
</body>
</html>

```



color-scheme.css

```
:root {  
  --accent-color: #CCEBFF;  
  --accent-color-hover: #b2dfff;  
  --accent-color-saturated: #44b4ff;  
  --accent-color-dark: #E5E5E5;  
  --accent-color-warning: #ff9999;  
  --accent-color-warning-hover: #ff6666;  
  --background-color: #F2F2F2;  
  --main-color: #fff;  
  --main-color-selected: #F5F5F5;  
  --main-color-selected-darker: #cacaca;  
  --text-color: #000000;  
  --text-color-gray: #8C8C8C;  
  --text-visited-color: #b66bfc;  
}
```



db_connection.php

```
<?php  
$host = 'localhost';  
$dbname = 'media_manager';  
$username = 'root';  
$password = '';  
  
try {  
  $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username, $password);  
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
} catch(PDOException $e) {  
  die("Connection failed: " . $e->getMessage());  
}
```



login_check.php

```
<?php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

function isLoggedIn() {
    return isset($_SESSION['user_id']) && isset($_SESSION['username']);
}

function requireLogin($loginPageUrl = 'login_details.php') {
    if (!isLoggedIn()) {
        $currentUrl = urlencode($_SERVER['REQUEST_URI']);
        header("Location: $loginPageUrl?redirect=$currentUrl");
        exit;
    }
}

function getCurrentUser() {
    if (isLoggedIn()) {
        return [
            'user_id' => $_SESSION['user_id'],
            'username' => $_SESSION['username'],
            'login_time' => $_SESSION['login_time'] ?? null
        ];
    }
    return null;
}

function logout($redirectUrl = 'index.php') {
    session_destroy();
    header("Location: $redirectUrl");
    exit;
}

//session timeout 30 minutes
function checkSessionTimeout($timeoutMinutes = 30) {
    if (isset($_SESSION['login_time'])) {
        $timeoutSeconds = $timeoutMinutes * 60;
        if (time() - $_SESSION['login_time'] > $timeoutSeconds) {
            session_destroy();
            return false;
        }
        //update login time on activity
        $_SESSION['login_time'] = time();
    }
    return true;
}
```

```

index.php
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Physical Media Manager Home</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="resources/main.css">
  <link rel="stylesheet" href="resources/color-scheme.css">
  <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
  <?php include 'resources/navbar.php'; ?>

  <?php
  $currentUser = getCurrentUser();
  ?>

  <div id="main-body">
    <div style="text-align:center; width: 100%; align-items: center; display: flex; flex-direction: column; gap: 20px;">
      <?php if ($currentUser): ?>
        <h1 style="text-align:center;">Welcome, <?php echo htmlspecialchars($currentUser['username']); ?>!</h1>
        <p style="width: 50%;">Manage your physical media collection with ease. Use the navigation bar above to view
and add to your collection, check out your profile, and explore community features like the user leaderboard.</p>
        <a href="collection_view.php">
          <button style="border-radius:50px; padding: 16px 32px;">View Collection</button>
        </a>
      <?php else: ?>
        <h1 style="text-align:center;">Welcome!</h1>
        <p style="width: 50%;">If you haven't already, sign up to get started managing your physical media today!</p>
        <a href="login_details.php">
          <button style="border-radius:50px; padding: 16px 32px;">Login/Signup</button>
        </a>
      <?php endif; ?>
    </div>
  </div>
</body>
</html>

```



login_details.php

```
<?php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}
require 'db_connection.php';

$redirectPage = $_GET['redirect'] ?? null;

$action = $_GET['action'] ?? 'login';
$message = '';
$messageType = '';

if (isset($_SESSION['user_id']) && isset($_SESSION['username'])) {
    $message = "You are already logged in as " . htmlspecialchars($_SESSION['username']) . "!";
    $messageType = 'success';
}

if (isset($_GET['action']) && $_GET['action'] === 'logout') {
    session_destroy();
    header("Location: " . $_SERVER['PHP_SELF']);
    exit;
}

if ($_POST) {
    $inputUsername = substr(trim($_POST['username'] ?? ''), 0, 25);
    $inputPassword = $_POST['password'] ?? '';
    $signup = $_POST['signup'] ?? '';

    if ($inputUsername && $inputPassword) {
        //check if username exists
        $stmt = $pdo->prepare("SELECT * FROM login_details WHERE username = ?");
        $stmt->execute([$inputUsername]);
        $existingUser = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($existingUser) {
            //username exists so check the password
            if (password_verify($inputPassword, $existingUser['password'])) {
                $_SESSION['user_id'] = $existingUser['userID'];
                $_SESSION['username'] = $existingUser['username'];
                $_SESSION['login_time'] = time();

                $message = "";
                $messageType = 'success';
                if ($redirectPage !== null) {
                    header("Location: " . $redirectPage);
                    exit;
                }
            } else {
                $message = "Incorrect password. Please try again.";
                $messageType = 'error';
            }
        } else {
            //username doesn't exist so trigger signup prompt
            if ($signup === 'yes') {
                //get the highest existing userID and add one
                $stmt = $pdo->query("SELECT MAX(userID) AS maxID FROM login_details");
                $maxID = $stmt->fetch(PDO::FETCH_ASSOC)['maxID'] ?? 0;
                $newUserID = $maxID + 1;

                $stmt = $pdo->prepare("INSERT INTO login_details (userID, username, password) VALUES (?, ?, ?)");
                $stmt->execute([$newUserID, $inputUsername, password_hash($inputPassword, PASSWORD_DEFAULT)]);

                //update login session details
                $_SESSION['user_id'] = $newUserID;
                $_SESSION['username'] = $inputUsername;
                $_SESSION['login_time'] = time();

                $message = "Account created successfully! Welcome, " . htmlspecialchars($inputUsername) . "!";
                $messageType = 'success';
                if ($redirectPage !== null) {
                    header("Location: " . $redirectPage);
                    exit;
                }
            } else if ($signup === 'no') {
                $message = "Login cancelled " .
            }
        }
    }
}
```

```

        $message = "Login cancelled.";
        $messageType = 'info';
    } else {
        $messageType = 'signup_prompt';
    }
}
} else {
    $message = "Please fill in all fields.";
    $messageType = 'error';
}
}

#Echo debug session stuff to make sure all the variables work
echo "← ";
foreach ($_SESSION as $key => $value) {
    echo "$key: $value; ";
}
echo " → ";
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login / Sign Up</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
    <script>
        function signUpYes() {
            document.getElementById('signupField').value = 'yes';
            document.getElementById('loginForm').submit();
        }

        function signUpNo() {
            document.getElementById('signupField').value = 'no';
            document.getElementById('loginForm').submit();
        }
    </script>
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="content-box">
                <h1 style="margin-bottom: 16px">Login / Sign Up</h1>

                <?php if (isset($_SESSION['username'])): ?>
                    <div style="background-color: #e8f5e8; padding: 16px; border-radius: 16px;">
                        <p><strong>Currently logged in as: <?php echo htmlspecialchars($_SESSION['username']); ?></strong>

                        <a href="?action=logout">
                            <button class="warning_button">Logout</button>
                        </a>
                    </div>
                <?php endif; ?>

                <?php if ($messageType === 'success'): ?>
                    <p style="color: green;"><strong><?php echo $message; ?></strong></p>
                <?php elseif ($messageType === 'error'): ?>
                    <p style="color: red;"><strong><?php echo $message; ?></strong></p>
                <?php elseif ($messageType === 'info'): ?>
                    <p style="color: blue;"><strong><?php echo $message; ?></strong></p>
                <?php elseif ($messageType === 'signup_prompt'): ?>
                    <p style="color: orange;"><strong>Username not found. Would you like to sign up?</strong></p>
                    <div style="margin-bottom: 16px;">
                        <button onclick="signUpYes()">Yes</button>
                        <button onclick="signUpNo()">No</button>
                    </div>
                <?php endif; ?>

                <?php if (!isset($_SESSION['username'])): ?>
                    <form method="POST" id="loginForm">
                        <div class="input-group" style="padding-right: 32px;">
                            <input type="text" id="username" name="username" maxlength="25" placeholder="Username" value="<?php
echo htmlspecialchars($_POST['username'] ?? ''); ?> required"
                            <input type="password" id="password" name="password" placeholder="Password" value="<?php echo

```

```
htmlspecialchars($_POST['password'] ?? ''); ?>" required>
    </div>

    <input type="hidden" id="signupField" name="signup" value="">

    <div>
        <input type="submit" value="Login / Sign Up">
    </div>
</form>
<?php endif; ?>
</div>
</div>
</body>
</html>
```

```

main.css
/* Leaderboard page stuff */
.leaderboard-header {
  text-align: center;
  margin-bottom: 48px;
}

.leaderboard-header h1 {
  font-size: 56px;
  color: var(--text-color);
  margin-bottom: 8px;
}

.leaderboard-header p {
  font-size: 18px;
  color: var(--text-color-gray);
  margin: 0;
}

.leaderboard-stats-row {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  grid-template-rows: auto auto;
  gap: 32px;
}

.leaderboard-card:last-child {
  grid-column: 1 / -1;
}

.leaderboard-card {
  background: var(--main-color-selected);
  border-radius: 32px;
  padding: 32px;
  position: relative;
  overflow: visible;
}

.leaderboard-card h2 {
  font-size: 28px;
  margin-bottom: 16px;
  color: var(--text-color);
  position: relative;
  padding-left: 16px;
}

.leaderboard-card h2::before {
  content: '';
  position: absolute;
  left: 0;
  top: 50%;
  transform: translateY(-50%);
  width: 4px;
  height: 24px;
  background: var(--accent-color-saturated);
  border-radius: 2px;
}

.leaderboard-card ol {
  padding: 0;
  margin: 0;
  list-style: none;
}

.leaderboard-card li {
  margin-bottom: 16px;
  font-size: 16px;
  padding: 12px 16px;
  background: var(--main-color);
  border-radius: 16px;
  transition: transform 0.2s ease, background-color 0.2s ease;
}

.leaderboard-card li:last-child {
  margin-bottom: 0;
}

.leaderboard-card li:hover {
  background: var(--accent-color-dark);
}

.rank-badge {
  display: inline-flex;
  align-items: center;
  justify-content: center;

```

```

    min-width: 28px;
    height: 28px;
    font-weight: bold;
    border-radius: 8px;
    font-size: 14px;
    background: var(--accent-color-saturated);
    color: var(--main-color);
}

.rank-gold {
  background: linear-gradient(135deg, #FFD700, #FFA500);
  color: #000;
}

.rank-silver {
  background: linear-gradient(135deg, #E8E8E8, #A0A0A0);
  color: #000;
}

.rank-bronze {
  background: linear-gradient(135deg, #CD7F32, #A0522D);
  color: #fff;
}

.leaderboard-card a {
  color: var(--text-color);
  text-decoration: none;
  font-weight: 500;
  transition: color 0.2s;
}

.leaderboard-card a.current-user {
  color: #fff;
  background: var(--text-visited-color);
  border-radius: 8px;
  padding: 4px;
  padding-left: 6px;
  padding-right: 0px;
  font-weight: 600;
  margin-right: 4px;
}

.leaderboard-card a:hover {
  color: var(--accent-color-saturated);
}

.leaderboard-card a.current-user:hover {
  color: #56287e;
}

@media (max-width: 900px) {
  .leaderboard-stats-row {
    grid-template-columns: 1fr;
  }

  .leaderboard-card:last-child {
    grid-column: auto;
  }

  .leaderboard-header {
    margin-bottom: 32px;
  }

  .leaderboard-header h1 {
    font-size: 48px;
  }
}

@media (max-width: 600px) {
  .leaderboard-card {
    padding: 24px;
  }

  .leaderboard-card h2 {
    font-size: 24px;
    margin-bottom: 24px;
  }

  .leaderboard-card li {
    padding: 10px 12px;
  }

  .leaderboard-box {
    margin: 16px;
    padding: 16px;
  }
}

```



main.css

```
/* Collection view stuff */
#collection-container {
  display: flex;
  margin-right: 256px;
  margin-bottom: 32px;
  min-width: 560px;
}

#filters-container {
  background: var(--main-color);
  border-radius: 24px;
  padding: 16px;
  padding-left: 24px;
  padding-right: 24px;
  max-width: 280px;
  align-self: flex-start;
  margin-right: 16px;
  transition: max-height 0.5s ease-in-out;
  overflow: hidden;
}

#filters-container h2 {
  font-size: 32px;
  font-weight: 700;
  margin: 0 0 20px 0;
  color: var(--text-color);
}

#filters-container h3 {
  font-size: 24px;
  font-weight: 700;
  margin: 25px 0 15px 0;
  color: var(--text-color);
}

#filters-container form > div {
  margin-bottom: 16px;
}

#filters-container input[type="radio"] {
  position: absolute;
  opacity: 0;
  pointer-events: none;
}

#filters-container label {
  display: inline-block;
  background: #fff;
  border: 2px solid var(--accent-color-dark);
  border-radius: 25px;
  padding: 12px 28px;
  cursor: pointer;
  font-size: 16px;
  font-weight: 500;
  color: #333;
  transition: all 0.2s ease;
  user-select: none;
}

#filters-container label:hover {
  border-color: var(--main-color-selected-darker);
  background: #fafafa;
}

#collections-filters {
  display: flex;
  flex-direction: row-reverse;
  margin: auto;
  justify-content: center;
}
```

```

}

.inline-inputs {
  display: flex;
  flex-wrap: wrap;
  gap: 8px;
  padding-bottom: 16px;
  border-bottom: 2px solid var(--accent-color-dark);
}

.inline-inputs label {
  flex: 1 1 0;
  text-align: center;
}

#filters-container input[type="radio"]:checked + label, #filters-container label:has(input[type="radio"]:checked) {
  background: var(--text-color);
  border-color: var(--text-color);
  color: #fff;
}

#filters-container input[type="text"], #filters-container input[type="number"] {
  width: 100%;
  padding: 14px 20px;
  border: transparent;
  border-radius: 25px;
  font-size: 16px;
  background-color: var(--main-color-selected);
  color: #666;
  box-sizing: border-box;
  outline: none;
  transition: all 0.3s ease;
  font-family: 'Cascadia Mono', Courier, monospace;
}

#filters-container input[type="text"]:focus, #filters-container input[type="number"]:focus {
  background-color: var(--accent-color-dark);
  color: #000;
}

#filters-container input[type="text"]::placeholder {
  color: #888;
}

#filters-container input[name="yearMin"],
#filters-container input[name="yearMax"] {
  width: calc(50% - 6px);
  display: inline-block;
}

#filters-container select {
  width: 100%;
  padding: 14px 20px;
  border: transparent;
  border-radius: 25px;
  font-size: 16px;
  background-color: var(--main-color-selected);
  color: #666;
  box-sizing: border-box;
  outline: none;
  transition: all 0.3s ease;
  font-family: 'Cascadia Mono', Courier, monospace;
  cursor: pointer;
}

#filters-container select:focus {
  background-color: var(--accent-color-dark);
  color: #000;
}

#filters-container button[type="submit"] {
  width: 100%;
  padding: 14px 20px;
}

```

```

    background: var(--accent-color-hover);
    color: #000;
    border: none;
    border-radius: 25px;
    font-size: 16px;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.2s ease;
}

#filters-container button[type="submit"]:hover {
    background: var(--accent-color-saturated);
    color: #fff;
}

#filters-container a {
    display: block;
    text-align: center;
    margin-top: 12px;
    color: #666;
    text-decoration: none;
    font-size: 14px;
    font-style: italic;
    transition: all 0.3s ease-in-out;
}

#filters-container a:hover {
    color: #1f1f1f;
}

.collection-grid {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    grid-auto-rows: 272px;
    gap: 16px;
    justify-content: center;
}

.collection-widget {
    background: var(--main-color);
    border-radius: 24px;
    padding: 16px;
    transition: transform 0.2s, box-shadow 0.2s;
    width: 240px;
    height: 240px;
    display: flex;
    flex-direction: column;
    overflow: hidden;
}

.media-title {
    font-family: 'Coolvetica', 'Inter', sans-serif;
    font-weight: 600;
    font-size: 36px;
    letter-spacing: 0.025em;
    line-height: 1em;
}

.media-artist {
    font-size: 28px;
    font-family: 'Coolvetica', 'Inter', sans-serif;
    font-weight: 500;
}

.media-year {
    font-size: 24px;
    font-family: 'Coolvetica', 'Inter', sans-serif;
    font-weight: 500;
    color: var(--text-color-gray);
}

.media-details {
    display: flex;

```

```

    flex-wrap: wrap;
    gap: 10px;
    margin-bottom: 10px;
}

.media-actions {
    margin-top: auto;
    display: flex;
    justify-content: space-between;
    align-items: flex-end;
    width: 100%;
}

.media-type-container {
    display: flex;
    align-items: center;
}

.action-links {
    display: flex;
    gap: 8px;
    align-items: center;
}

.edit-link:hover {
    text-decoration: underline;
}

.delete-link:hover {
    text-decoration: underline;
}

.no-results {
    text-align: center;
    color: #666;
    font-style: italic;
    margin: 40px 0;
    width: 100%;
}

@media (max-width: 1100px) {
    #collections-filters {
        flex-direction: column;
        align-items: center;
        margin: 0;
        padding: 0 16px;
    }

    #collection-container {
        margin-right: 0;
        margin-bottom: 16px;
        min-width: 100%;
        width: 100%;
        order: 2;
    }

    #filters-container {
        max-width: 100%;
        margin-right: 0;
        margin-bottom: 16px;
        order: 1;
        align-self: center;
    }

    .collection-widget {
        width: auto;
    }

    #filters-container label {
        padding: 10px 20px;
        font-size: 14px;
        margin: 4px 2px;
    }
}

```

```
.inline-inputs {
  display: flex;
  flex-wrap: wrap;
  gap: 8px;
}

.inline-inputs input[type="text"],
.inline-inputs input[type="number"] {
  flex: 1;
  min-width: 120px;
}

}

@media (max-width: 600px) {
  .collection-grid {
    grid-template-columns: repeat(1, 1fr);
  }
}
```

```

main.css

/* Navbar stuff */
#header {
  display: flex;
  padding-top: 32px;
  flex-direction: column;
  z-index: 100;
  position: fixed;
  top: 0;
  left: 50%;
  transform: translateX(-50%);
  width: auto;
  align-items: center;
  justify-content: flex-start;
  height: auto;
  min-height: 0;
  pointer-events: none;
}

#nav-bar {
  background-color: var(--main-color);
  display: inline-flex;
  width: auto;
  height: 56px;
  align-items: center;
  padding: 0px;
  margin: 0px;
  border-radius: 28px;
  overflow: hidden;
  box-shadow: 0 2px 16px rgba(0, 0, 0, 0.1);
  pointer-events: auto;
}

#nav-bar a {
  color: var(--color-text-1-light);
  padding: 16px 32px;
  margin: 0;
  height: calc(100% - 32px);
  text-decoration: none;
  justify-content: center;
  align-items: center;
  display: flex;
  border-left: 2px solid var(--accent-color-dark);
  min-width: 56px;
  position: relative;
}

#nav-bar a:hover {
  background-color: var(--main-color-selected);
}

#nav-bar a:first-child {
  border-left: none;
}

#nav-bar a img {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 1;
  transition: opacity 0.25s ease, filter 0.25s ease;
}

#nav-bar a p {
  overflow: hidden;
  width: 0%;
  z-index: 10;
  position: relative;
  text-align: center;
  transition: width 0.25s ease;
}

#nav-gap-bridge {

```

```

    position: absolute;
    top: 88px;
    left: 0;
    right: 0;
    height: 16px;
    pointer-events: auto;
    background: transparent;
}

#header #menu-dropdown {
    display: flex;
    flex-direction: column;
    background-color: var(--main-color);
    color: var(--color-text-1-light);
    border-radius: 28px;
    padding: 16px;
    margin-top: 16px;
    transition:
        all 0.3s ease-in-out,
        max-height 0.125s ease-in-out;
    opacity: 0;
    overflow: hidden;
    max-height: 0;
    pointer-events: auto;
    box-shadow: 0 2px 16px rgba(0, 0, 0, 0.1);
}

#menu-dropdown h2 {
    margin: 0;
}

#menu-dropdown p {
    margin: 0;
}

#menu-dropdown div {
    transition: opacity 0.125s ease-in-out;
}

@media (max-width: 550px) {
    #nav-bar a {
        padding: 16px 16px;
    }

    #nav-bar a:first-child {
        padding-left: 16px;
        border-left: none;
    }
}

.dropdown-item {
    display: flex;
    flex-direction: column;
    justify-content: flex-end;
    width: 100%;
    background-color: var(--main-color-selected);
    border-radius: 16px;
    padding: 20px;
    cursor: pointer;
    transition: background-color 0.3s ease-in-out;
    text-decoration: none;
    color: var(--text-color);
}

.dropdown-item:hover {
    background-color: var(--accent-color-dark);
}

.dropdown-list {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    width: 100%;
    gap: 16px;
}

/* Collection stats stuff */

```

```

.stats-container {
  max-width: 410px;
  margin: 16px;
}

.stat-card {
  background: #f8f9fa;
  border-radius: 24px;
  padding: 24px;
  margin-bottom: 1rem;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  border: 1px solid #e9ecef;
  font-family: 'Coolvetica', 'Inter', sans-serif;
  font-weight: 600;
}

.stat-number {
  margin: 0;
}

.stat-number.blue {
  color: var(--accent-color-saturated);
}

.total-card .stat-number {
  color: var(--accent-color-saturated);
}

.stats-row {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 16px;
}

.generic-icon {
  font-weight: 600;
  color: var(--accent-color-saturated);
  font-size: 48px;
}

#collection-total-title {
  font-size: 32px;
  font-weight: 500;
  display: flex;
  align-items: flex-end;
  width: 100%;
  justify-content: space-between;
}

#collection-total-title .stat-number {
  font-size: 72px;
  font-weight: 600;
  display: flex;
}

#average-year {
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  flex-direction: column;
}

#average-year .stat-label {
  font-size: 26px;
  font-weight: 500;
  width: 100%;
  text-align: center;
}

#average-year .stat-number {
  font-size: 80px;
  font-weight: 600;
}

#top-media-title {
  display: flex;

```

```
    flex-direction: column-reverse;
    align-items: center;
    font-size: 32px;
    font-weight: 500;
}

#top-artist {
    display: flex;
    flex-direction: column-reverse;
    padding: 16px;
}

#top-artist .stat-label {
    font-size: 32px;
    font-weight: 500;
}

.artist-name {
    color: var(--accent-color-saturated);
    word-break: break-word;
    float: right;
    text-align: right;
    font-size: 64px;
}
```

```

main.css

/* Public profile stuff */
.profile-content-box {
  background-color: var(--main-color);
  border-radius: 32px;
  padding: 32px;
  box-shadow: 0 2px 16px rgba(0, 0, 0, 0.1);
  max-width: 100%;
  box-sizing: border-box;
}

#recently-collected {
  display: flex;
  background-color: var(--accent-color-dark);
  flex-direction: column;
  padding: 24px;
  border-radius: 16px;
}

#recently-collected h2 {
  font-size: 32px;
  margin-bottom: 16px;
}

#profile-icon-top {
  display: flex;
  gap: 16px;
  margin-bottom: 16px;
}

#profile-icon-top #user-icon {
  flex: 1;
  width: calc(50% - 8px);
  height: auto;
  display: block;
  object-fit: cover;
  border-radius: 24px;
}

#profile-icon-top > div {
  flex: 1;
}

#profile-icon-top #top-media-profile {
  display: flex;
  flex-direction: column-reverse;
  padding: 16px;
  justify-content: center;
  align-items: center;
  background-color: var(--accent-color-dark);
  border-radius: 24px;
  text-align: center;
}

#profile-icon-top #top-media-profile h1 {
  margin-top: 16px;
  font-size: 40px;
}

.media-widgets-carousel {
  display: flex;
  gap: 10px;
  overflow-x: auto;
  padding: 10px 0;
  scroll-behavior: smooth;
  width: 512px;
}

.media-widgets-carousel::-webkit-scrollbar {
  height: 6px;
}

.media-widgets-carousel::-webkit-scrollbar-track {
  background: #f1f1f1;
  border-radius: 10px;
}

```

```

.media-widgets-carousel::-webkit-scrollbar-thumb {
  background: #c1c1c1;
  border-radius: 10px;
  transition: background 0.2s ease;
}

.media-widgets-carousel::-webkit-scrollbar-thumb:hover {
  background: #a8a8a8;
}

.media-item-widget {
  width: 150px;
  height: 150px;
  display: flex;
  flex-direction: column;
  flex-shrink: 0;
  min-width: 150px;
  background-color: var(--main-color);
  padding: 16px;
  border-radius: 16px;
  overflow: hidden;
}

.media-item-widget h3 {
  line-height: 18px;
}

.media-item-widget p {
  margin: 0;
}

.media-item-widget img {
  margin-top: auto;
  align-self: flex-start;
}

#widget-artist {
  font-family: 'Coolvetica', 'Inter', sans-serif;
  font-weight: 500;
  font-size: 18px;
}

#widget-year {
  font-family: 'Coolvetica', 'Inter', sans-serif;
  font-weight: 500;
  color: var(--text-color-gray);
}

.generic-icon {
  font-size: 80px;
  color: var(--text-color-gray);
}

@media (max-width: 750px) {
  .profile-content-box {
    margin: 0;
    max-width: 90%;
    padding: 20px;
  }

  #recently-collected {
    padding: 20px;
  }

  #profile-icon-top #user-icon {
    aspect-ratio: 1;
  }

  #profile-icon-top #top-media-profile {
    aspect-ratio: 1;
  }

  #profile-icon-top #top-media-profile img {
    max-width: 100px;
    max-height: 100px;
  }

  .media-widgets-carousel {
    width: 100%;
  }
}

```

```

        width: 100%;
    }

    .media-item-widget {
        width: 140px;
        height: 140px;
        min-width: 140px;
    }
}

@media (max-width: 480px) {
    .profile-content-box {
        padding: 16px;
    }

    #recently-collected {
        padding: 16px;
    }

    #profile-icon-top #top-media-profile img {
        max-width: 80px;
        max-height: 80px;
    }

    .media-item-widget {
        width: 130px;
        height: 130px;
        min-width: 130px;
    }
}

```

```

main.css

@font-face {
  font-family: 'Coolvetica';
  src: url('coolvetica_rg.otf') format('opentype');
  font-weight: 100;
  font-style: normal;
}

@font-face {
  font-family: 'Casadia Mono';
  src: url('CasadiaMono.ttf') format('opentype');
  font-weight: 100;
  font-style: normal;
}

body {
  background-color: var(--background-color);
  font-family: 'Casadia Mono', Courier, monospace;
  color: var(--text-color);
  padding: 0;
  margin: 0;
}

h1, h2, h3 {
  font-family: 'Coolvetica', 'Inter', sans-serif;
  margin: 0;
  letter-spacing: 0.05em;
}

h1 {
  font-size: 48px;
}

button, input[type="button"], input[type="submit"] {
  background-color: var(--accent-color);
  border: none;
  color: var(--text-color);
  padding: 16px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  border-radius: 16px;
  font-family: 'Casadia Mono', Courier, monospace;
  transition: background-color 0.3s ease-in-out;
  height: 60px;
  margin: 0;
}

button:hover, input[type="button"]:hover, input[type="submit"]:hover {
  background-color: var(--accent-color-hover);
}

button.warning_button, input[type="button"].warning_button, input[type="submit"].warning_button {
  background-color: var(--accent-color-warning);
}

button.warning_button:hover, input[type="button"].warning_button:hover, input[type="submit"].warning_button:hover {
  background-color: var(--accent-color-warning-hover);
}

#main-body {
  margin-top: 120px;
  transition: filter 0.3s ease-in-out;
}

.content-box {
  background-color: var(--main-color);
  border-radius: 32px;
  padding: 32px;
  box-shadow: 0 2px 16px rgba(0, 0, 0, 0.1);
  margin: 32px;
  margin-top: 0;
}

```

```

    margin-top: 0;
}

.input-group {
  display: flex;
  flex-direction: column;
  gap: 16px;
  margin-bottom: 16px;
  width: 100%;
}

.input-group input[type="text"], .input-group input[type="password"], .input-group input[type="number"] {
  padding: 16px;
  background-color: var(--main-color-selected);
  border: transparent;
  border-radius: 16px;
  font-size: 16px;
  font-family: 'Casadia Mono', Courier, monospace;
  color: var(--text-color);
  transition: border-color 0.3s ease-in-out, background-color 0.3s ease-in-out;
  outline: none;
}

.input-group input[type="text"]:focus, .input-group input[type="password"]:focus, .input-group input[type="number"]:focus {
  background-color: var(--accent-color-dark);
  border: transparent;
}

.input-group select {
  padding: 16px;
  background-color: var(--main-color-selected);
  border: transparent;
  border-radius: 16px;
  font-size: 16px;
  font-family: 'Casadia Mono', Courier, monospace;
  color: var(--text-color);
  transition: border-color 0.3s ease-in-out, background-color 0.3s ease-in-out;
  outline: none;
}

.input-group select:invalid {
  color: #757575;
}

.input-group option {
  border-radius: 16px;
}

.input-button-pair {
  display: flex;
  gap: 8px;
  margin: 0;
  justify-content: space-between;
}

.input-button-pair input[type="text"], .input-button-pair input[type="password"] {
  width: 100%;
}

.centered-container {
  display: flex;
  justify-content: center;
  align-items: center;
  width: 100%;
}

.warning-p {
  display: flex;
  width: 100%;
  height: 100%;
  justify-content: center;
  align-items: center;
  margin: 0;
}

```

```

media_add.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$messageType = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    reload //Changed this to stop repeated adding of item on
    $stmt = $pdo->query("SELECT MAX(mediaID) AS maxID FROM media_items");
    $maxID = $stmt->fetch(PDO::FETCH_ASSOC)['maxID'] ?? 0;
    $mediaID = $maxID + 1;

    $userID = $_SESSION['user_id'] ?? '';
    $title = $_POST['title'] ?? '';
    $artist = $_POST['artist'] ?? '';
    $mediaType = $_POST['mediaType'] ?? '';
    $year = $_POST['year'] ?? '';
    $location = $_POST['location'] ?? '';

    if ($userID && $mediaID && $title && $artist && $mediaType && is_numeric($year) && $year >= 1900 && $year <= 3000 &&
    $mediaType === 'Vinyl' || $mediaType === 'Cassette' || $mediaType === 'CD') {
        $stmt = $pdo->prepare("INSERT INTO media_items (userID, mediaID, title, artist, mediaType, year, location) VALUES (?,
        ?, ?, ?, ?, ?, ?)");
        $stmt->execute([$userID, $mediaID, $title, $artist, $mediaType, $year, $location]);

        header("Location: " . $_SERVER['PHP_SELF'] . "?message=success");
        exit;
    } else {
        header("Location: " . $_SERVER['PHP_SELF'] . "?message=error");
        exit;
    }
}

if (isset($_GET['message'])) {
    $messageType = $_GET['message'];
}

$stmt = $pdo->prepare('SELECT artist FROM media_items WHERE userID = ? GROUP BY artist');
$stmt->execute($_SESSION['user_id']);
$allArtists = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Media Items Manager</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="content-box">
                <h2 style="margin-bottom:16px; font-size:40px">Add New Media Item</h2>
                <form method="POST">
                    <div class="input-group">
                        <input type="text" id="title" placeholder="Title*" name="title" required>
                        <input type="text" list="artist" id="artist" placeholder="Artist*" name="artist" required>
                        <datalist id="artist">
                            <?php foreach ($allArtists as $artist): ?>
                                <option value="<?= htmlspecialchars($artist['artist']) ?>"></option>
                            <?php endforeach; ?>
                        </datalist>
                        <select id="mediaType" name="mediaType" required>
                            <option value="">Media Type</option>
                            <option value="Vinyl">Vinyl</option>
                            <option value="Cassette">Cassette</option>
                            <option value="CD">CD</option>
                        </select>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

        <input type="number" id="year" name="year" placeholder="Year" min="1900" max="3000">
        <input type="text" id="location" placeholder="Location" name="location">
        <input type="submit" value="Add Media Item">
    </div>
</form>

<?php if ($messageType === 'success'): ?>
    <p style="color: green;"><strong>Media item added successfully!</strong></p>
<?php elseif ($messageType === 'error'): ?>
    <p style="color: red;"><strong>Please fill in required fields (Title, Artist, and Media Type).</strong>
</p>

    <?php endif; ?>
</div>
</div>
</body>
</html>

```

```

media_delete.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$deleteMediaID = $_GET['mediaID'] ?? null;
$action = $_GET['action'] ?? null;
$error = null;
$media = null;

if ($deleteMediaID !== null) {
    $stmt = $pdo->prepare("SELECT mediaID, title, artist FROM media_items WHERE mediaID = ? AND userID = ?");
    $stmt->execute([$deleteMediaID, $_SESSION['user_id']]);
    $media = $stmt->fetch(PDO::FETCH_ASSOC);

    if (!$media) {
        $error = "Media item not found or you don't have permission to delete it.";
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST' && $action === 'delete_media' && $media) {
    $deleteStmt = $pdo->prepare("DELETE FROM media_items WHERE mediaID = ? AND userID = ?");
    $deleteStmt->execute([$deleteMediaID, $_SESSION['user_id']]);

    header("Location: collection_view.php");
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Confirm Deletion</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
<?php include 'resources/navbar.php'; ?>
<div id="main-body">
    <div class="centered-container" style="height:100%">
        <div class="content-box">
            <h2 style="color:red;">Confirm Media Item Deletion</h2>
            <?php if ($error): ?>
                <p style="color:red;"><?= htmlspecialchars($error) ?></p>
                <a href="collection_view.php"><button>Back</button></a>
            <?php elseif ($media): ?>
                <p>This action cannot be undone.<br>Are you sure you want to delete the following media item?</p>
                <p><strong>Title:</strong> <?= htmlspecialchars($media['title']) ?><br><strong>Artist:</strong> <?=
htmlspecialchars($media['artist']) ?></p>
                <form method="POST" action="<?= htmlspecialchars($_SERVER['PHP_SELF']) . "?action=delete_media&mediaID=" .
urlencode($deleteMediaID) ?>">
                    <input type="submit" value="Yes, Delete This Media" class="warning_button">
                </form>
                <a href="collection_view.php"><button style="margin-top:16px">Cancel</button></a>
            <?php else: ?>
                <p>No media item selected.</p>
                <a href="collection_view.php"><button>Back</button></a>
            <?php endif; ?>
        </div>
    </div>
</div>
</body>
</html>

```

```

media_edit.php

<?php
require 'db_connection.php';
require 'login_check.php';
requireLogin();

$user = getCurrentUser();
$messageType = '';

$mediaID = $_GET['id'] ?? null;
$mediaItem = null;

if (!$mediaID) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>No media item specified. Please provide a valid media ID.</p>";
    exit;
}

$stmt = $pdo->prepare("SELECT * FROM media_items WHERE mediaID = ? AND userID = ?");
$stmt->execute([$mediaID, $_SESSION['user_id']]);
$mediaItem = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$mediaItem) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>Media item not found or you don't have permission to edit it.</p>";
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $userID = $_SESSION['user_id'];
    $title = $_POST['title'] ?? '';
    $artist = $_POST['artist'] ?? '';
    $mediaType = $_POST['mediaType'] ?? '';
    $year = $_POST['year'] ?? '';
    $location = $_POST['location'] ?? '';

    if ($userID && $mediaID && $title && $artist && $mediaType && is_numeric($year) && $year ≥ 1900 && $year ≤ 3000 &&
    $mediaType === 'Vinyl' || $mediaType === 'Cassette' || $mediaType === 'CD') {
        $stmt = $pdo->prepare("UPDATE media_items SET title = ?, artist = ?, mediaType = ?, year = ?, location = ? WHERE
mediaID = ? AND userID = ?");
        $stmt->execute([$title, $artist, $mediaType, $year, $location, $mediaItem['mediaID'], $userID]);

        if ($stmt->rowCount() > 0) {
            header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=success");
            exit;
        } else {
            header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=error");
            exit;
        }
    } else {
        header("Location: " . $_SERVER['PHP_SELF'] . "?id=" . urlencode($mediaID) . "&message=missing");
        exit;
    }
}

if (isset($_GET['message'])) {
    $messageType = $_GET['message'];
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Media Item</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="content-box">

```

```

<h1>Edit Media Item</h1>

<p><strong>Currently editing:</strong> <?= htmlspecialchars($mediaItem['title']) ?> by <?=
htmlspecialchars($mediaItem['artist']) ?></p>
<form method="POST">
  <div class="input-group">
    <input type="text" id="title" name="title" placeholder="Title*" value="<?=
htmlspecialchars($mediaItem['title']) ?>" required>
    <input type="text" id="artist" name="artist" placeholder="Artist*" value="<?=
htmlspecialchars($mediaItem['artist']) ?>" required>
    <select id="mediaType" name="mediaType" required>
      <option value="">Media Type</option>
      <option value="Vinyl" <?= ($mediaItem['mediaType'] === 'Vinyl') ? 'selected' : '' ?
>>Vinyl</option>
      <option value="Cassette" <?= ($mediaItem['mediaType'] === 'Cassette') ? 'selected' : '' ?
>>Cassette</option>
      <option value="CD" <?= ($mediaItem['mediaType'] === 'CD') ? 'selected' : '' ?>>CD</option>
    </select>
    <input type="number" id="year" name="year" min="1900" max="3000" placeholder="Year" value="<?=
htmlspecialchars($mediaItem['year']) ?>">
    <input type="text" id="location" name="location" placeholder="Location" value="<?=
htmlspecialchars($mediaItem['location']) ?>">
    <input type="submit" value="Update Media Item">
  </div>
</form>

<?php if ($messageType === 'success'): ?>
  <p style="color: green;"><strong>Media item edited successfully!</strong></p>
<?php elseif ($messageType === 'missing'): ?>
  <p style="color: red;"><strong>Please fill in required fields (Title, Artist, and Media Type).</strong>
</p>

<?php elseif ($messageType === 'error'): ?>
  <p style="color: amber;"><strong>There was an error editing the media item</strong></p>
<?php endif; ?>
</div>
</div>
</body>
</html>

```



navbar.js

```
const navBar = document.getElementById("nav-bar");
const menuDropdown = document.getElementById("menu-dropdown");
var menuDropdownOpen = false;
var currentMenuType = null;
const menuSectionHeights = {};
var dropdownEnabled = false;
const isMobileDevice = isMobile();
const clickedLinks = new Set();

setTimeout(() => {
  dropdownEnabled = true;
}, 500);

if (navBar && menuDropdown) {
  const syncWidths = () => {
    menuDropdown.style.width = `${navBar.offsetWidth - 32}px`;
  };

  const resizeObserver = new ResizeObserver(syncWidths);
  resizeObserver.observe(navBar);

  syncWidths();
}

function setupNav() {
  const navBarDivs = navBar ? navBar.querySelectorAll("a") : [];
  const headingCenterTop = document.getElementById("heading-center-top");

  const menuSections = document.querySelectorAll(".dropdown-list");
  menuSections.forEach((section) => {
    menuSectionHeights[section.id] = section.offsetHeight;
    section.style.display = "none";
  });

  navBarDivs.forEach((div) => {
    const menuType = div.getAttribute("data-page");

    if (isMobileDevice) {
      div.addEventListener("click", function (event) {
        if (menuType !== "home") {
          if (!dropdownEnabled) return;

          const linkHref = div.getAttribute("href");

          if (menuDropdownOpen && currentMenuType === menuType) {
            closeDropdown();
            clickedLinks.clear();
            hideLinkAnimation(div);

            setTimeout(() => {
              if (linkHref) window.location.href = linkHref;
            }, 150);
            return;
          }

          event.preventDefault();

          showLinkAnimation(div);
          clickedLinks.add(menuType);
          dropdownMenu(menuType);

          const navBarDivs = navBar ? navBar.querySelectorAll("a") : [];
          navBarDivs.forEach(otherDiv => {
            if (otherDiv !== div) hideLinkAnimation(otherDiv);
          });
        }
      });
    }
  });
}
```

```

div.addEventListener("touchstart", function(event) {
  div.style.transform = "scale(0.95)";
});

div.addEventListener("touchend", function(event) {
  div.style.transform = "scale(1)";
});

} else {
  div.addEventListener("mouseover", function (event) {
    const img = div.querySelector("img");
    const p = div.querySelector("p");
    p.style.width = "100%";
    img.style.opacity = "0";
    img.style.filter = "blur(8px)";

    if (dropdownEnabled) {
      if (menuType !== "home") {
        dropdownMenu(menuType);
      } else {
        closeDropdown();
      }
    }
  });

  div.addEventListener("mouseout", function (event) {
    const relatedTarget = event.relatedTarget;
    const img = div.querySelector("img");
    const p = div.querySelector("p");
    p.style.width = "0%";
    img.style.opacity = "1";
    img.style.filter = "blur(0)";
    if (dropdownEnabled &&
      !menuDropdown.contains(relatedTarget) &&
      (!headingCenterTop || !headingCenterTop.contains(relatedTarget))
    ) {
      closeDropdown();
    }
  });
}
});

if (headingCenterTop && !isMobileDevice) {
  headingCenterTop.addEventListener("mouseout", function (event) {
    const relatedTarget = event.relatedTarget;
    if (dropdownEnabled &&
      !menuDropdown.contains(relatedTarget) &&
      !headingCenterTop.contains(relatedTarget)
    ) {
      closeDropdown();
    }
  });

  headingCenterTop.addEventListener("mouseover", function () {
    if (dropdownEnabled && menuDropdownOpen) {
      dropdownMenu();
    }
  });
}

if (isMobileDevice) {
  document.addEventListener("click", function(event) {
    if (menuDropdownOpen &&
      !navBar.contains(event.target) &&
      !menuDropdown.contains(event.target)) {
      closeDropdown();
      clickedLinks.clear();
    }
  });
}

const dropdownItems = document.querySelectorAll(".dropdown-item");

```

```

        dropdownItems.forEach(item => {
            item.addEventListener("click", function() {
                closeDropdown();
                clickedLinks.clear();
            });
        });
    }
}
setupNav();

function showLinkAnimation(linkElement) {
    const img = linkElement.querySelector("img");
    const p = linkElement.querySelector("p");
    if (img && p) {
        p.style.width = "100%";
        img.style.opacity = "0";
        img.style.filter = "blur(8px)";
    }
}

function hideLinkAnimation(linkElement) {
    const img = linkElement.querySelector("img");
    const p = linkElement.querySelector("p");
    if (img && p) {
        p.style.width = "0%";
        img.style.opacity = "1";
        img.style.filter = "blur(0)";
    }
}

function dropdownMenu(menu_type) {
    if (!dropdownEnabled) return;

    const mainBody = document.getElementById("main-body");
    if (mainBody) {
        document.getElementById("main-body").style.filter = "blur(8px)";
    }
    menuDropdown.style.opacity = "1";
    menuDropdown.style.filter = "blur(0)";
    menuDropdown.style.pointerEvents = "all";

    if (menu_type) {
        const menuSections = document.querySelectorAll(".dropdown-list");
        menuSections.forEach((section) => (section.style.display = "none"));
    }

    if (menu_type) {
        const targetSection = document.getElementById(menu_type);
        if (targetSection) {
            if (currentMenuType !== menu_type && menuDropdownOpen) {
                const pastSection = document.getElementById(currentMenuType);
                if (pastSection) {
                    transitionToPage(pastSection, targetSection);
                } else {
                    targetSection.style.opacity = "0";
                    targetSection.style.display = "flex";
                    setTimeout(() => {
                        targetSection.style.opacity = "1";
                    }, 50);
                }
            } else {
                targetSection.style.display = "flex";
                setTimeout(() => {
                    targetSection.style.opacity = "1";
                }, 50);
            }
            menuDropdown.style.maxHeight =
                menuSectionHeights[targetSection.id] + "px";
            currentMenuType = menu_type;
        }
    }
}

```

```

    menuDropdownOpen = true;
}

function transitionToPage(past_page, new_page) {
    past_page.style.display = "flex";
    past_page.style.transition = "opacity 0.25s ease-out";
    new_page.style.transition = "opacity 0.25s ease-in";
    past_page.style.opacity = "0";
    new_page.style.opacity = "0";

    setTimeout(() => {
        new_page.style.display = "flex";
        setTimeout(() => {
            new_page.style.opacity = "1";
        }, 25);
    }, 125);

    setTimeout(() => {
        past_page.style.display = "none";
    }, 125);
}

function closeDropdown(menu_type) {
    const mainBody = document.getElementById("main-body");
    if (mainBody) {
        document.getElementById("main-body").style.filter = "blur(0)";
    }
    menuDropdown.style.opacity = "0";
    menuDropdown.style.filter = "blur(8px)";
    menuDropdown.style.pointerEvents = "none";
    menuDropdownOpen = false;

    if (isMobileDevice) {
        const navBarDivs = navBar ? navBar.querySelectorAll("a") : [];
        navBarDivs.forEach(hideLinkAnimation);
    }
}

closeDropdown();

//allows users to deselect radio buttons when they double click plus auto submit form on radio button change
document.addEventListener("DOMContentLoaded", () => {
    document.querySelectorAll('label').forEach(label => {
        label.addEventListener("dblclick", () => {
            const radio = label.querySelector('input[type="radio"]');
            if (radio && radio.checked) {
                radio.checked = false;
                const form = label.closest('form');
                if (form) {
                    form.submit();
                }
            }
        });
    });
});

const form = document.querySelector('#filters-container form');
if (form) {
    form.querySelectorAll('input[type="radio"]').forEach(radio => {
        radio.addEventListener("change", () => {
            form.submit();
        });
    });
}
});

function isMobile() {
    const nav_user_agent = navigator.userAgent.toLowerCase();
    const mobile_tags = ["mobile", "android", "webos", "iphone", "ipad", "blackberry", "iemobile", "opera mini"];

    const isTouchDevice = ('ontouchstart' in window) || (navigator.maxTouchPoints > 0);
    const isSmallScreen = window.innerWidth <= 600;

```

```

X navbar.php

<?php
require_once __DIR__ . '/../login_check.php';
$currentUser = getCurrentUser();
?>

<div id="header">
  <div id="heading-center-top">
    <div id="nav-bar">
      <a href="index.php" data-page="home">
        
        <p>Home</p>
      </a>
      <a href="collection_view.php" data-page="collection">
        
        <p>Collection</p>
      </a>
      <a href="<?php echo $currentUser ? 'login_details.php' : 'login_details.php'; ?>" data-page="profile">
        
        <p><?php echo $currentUser ? htmlspecialchars($currentUser['username']) : 'Login'; ?></p>
      </a>
    </div>
    <div id="nav-gap-bridge"></div>
    <div id="menu-dropdown">
      <div id="collection" class="dropdown-list">
        <a href="collection_stats.php" class="dropdown-item">
          <h2>Collection Stats</h2>
          <p>View interesting stats about your physical media collection</p>
        </a>
        <a href="media_add.php" class="dropdown-item">
          <h2>Add New Item</h2>
          <p>Quickly add new item to your collection</p>
        </a>
        <a href="userLeaderboard.php" class="dropdown-item">
          <h2>User Leaderboard</h2>
          <p>See top users by collection stats</p>
        </a>
      </div>
      <div id="profile" class="dropdown-list">
        <?php if ($currentUser): ?>
          <a href="account_settings.php" class="dropdown-item">
            <h2>Account Settings</h2>
            <p>Change account settings such as username or password</p>
          </a>
          <a href="profile.php" class="dropdown-item">
            <h2>Public Profile</h2>
            <p>View your public profile</p>
          </a>
          <a href="login_details.php?action=logout" class="dropdown-item">
            <h2>Logout</h2>
            <p>Sign out of your account</p>
          </a>
        <?php else: ?>
          <a href="login_details.php" class="dropdown-item">
            <h2>Login/Sign-up</h2>
            <p>Login or sign up to your account</p>
          </a>
        <?php endif; ?>
      </div>
    </div>
  </div>
</div>

<script src="resources/navbar.js"></script>

```

```

profile.php

<?php
require 'db_connection.php';
require 'login_check.php';
if (isLoggedIn()) {
    $user = getCurrentUser();
}

$publicUsername = $_GET['user'] ?? null;

if (!$publicUsername) {
    $username = $user['username'] ?? null; // get the username safely
    if ($username) {
        $redirectUrl = $_SERVER['REQUEST_URI'];
        $redirectUrl = strtok($redirectUrl, '?');
        $redirectUrl .= '?user=' . urlencode($username);

        header("Location: $redirectUrl");
        exit;
    }
}

if (!$publicUsername) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>No user specified. Please provide a username.</p>";
    exit;
}

$stmt = $pdo->prepare("SELECT * FROM login_details WHERE username = ?");
$stmt->execute([$publicUsername]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$user) {
    echo "<link rel='stylesheet' href='resources/main.css'>";
    echo "<p class='warning-p' style='color: red;'>User not found</p>";
    exit;
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><?= htmlspecialchars($user['username']) ?>'s Public Profile</title>
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">

    <script>
        function shareSite() {
            const url = window.location.href;
            const your_username = <?php echo json_encode($user['username']); ?>;

            const shareData = {
                title: `${your_username} Physical Media Profile`,
                text: `View all of ${your_username}'s physical media`,
                url: url,
            };

            if (navigator.share) {
                navigator.share(shareData);
            }
        }
    </script>
</head>
<body>
    <?php include 'resources/navbar.php'; ?>

    <div id="main-body">
        <div class="centered-container">
            <div class="profile-content-box">
                <div style="display:flex; justify-content:space-between;">
                    <h1 style="color:var(--accent-color-saturated); margin-bottom:16px"><?=

```



```

userLeaderboard.php

<?php
require 'db_connection.php';

session_start();

$currentUser = $_SESSION['username'] ?? null;

//top collection
$topUsersStmt = $pdo->query("
    SELECT l.username, COUNT(m.mediaID) AS total
    FROM login_details l
    JOIN media_items m ON l.userID = m.userID
    GROUP BY l.userID
    ORDER BY total DESC
    LIMIT 10
");
$topUsers = $topUsersStmt->fetchAll(PDO::FETCH_ASSOC);

//top average year
$avgYearStmt = $pdo->query("
    SELECT l.username, AVG(m.year) AS avg_year, COUNT(m.mediaID) AS total
    FROM login_details l
    JOIN media_items m ON l.userID = m.userID AND m.year IS NOT NULL
    GROUP BY l.userID
    HAVING total > 0
    ORDER BY avg_year DESC
    LIMIT 10
");
$topAvgYears = $avgYearStmt->fetchAll(PDO::FETCH_ASSOC);

//top collected artists
$topArtistStmt = $pdo->query("
    SELECT l.username, t.artist, t.artist_count
    FROM login_details l
    JOIN (
        SELECT userID, artist, COUNT(*) AS artist_count
        FROM media_items
        GROUP BY userID, artist
    ) t ON l.userID = t.userID
    WHERE (t.userID, t.artist_count) IN (
        SELECT userID, MAX(artist_count)
        FROM (
            SELECT userID, artist, COUNT(*) AS artist_count
            FROM media_items
            GROUP BY userID, artist
        ) sub
        GROUP BY userID
    )
    ORDER BY t.artist_count DESC
    LIMIT 10
");
$topArtists = $topArtistStmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User Leaderboard</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="resources/main.css">
    <link rel="stylesheet" href="resources/color-scheme.css">
    <link rel="icon" type="image/x-icon" href="resources/pageIcon.ico">
</head>
<body>
    <?php include 'resources/navbar.php'; ?>
    <div id="main-body">
        <div class="centered-container">
            <div class="content-box leaderboard-box" style="max-width:900px;">
                <div class="leaderboard-header">
                    <h1>User Leaderboards</h1>
                    <p>See the top collectors and stats across the community</p>
                </div>
                <div class="leaderboard-stats-row">
                    <div class="leaderboard-card">
                        <h2>Top Collections</h2>

```

```

<ol>
<?php foreach ($stopUsers as $i => $user): ?>
    <li>
        <span class="rank-badge
            <?php if($i==0) echo 'rank-gold';
                elseif($i==1) echo 'rank-silver';
                elseif($i==2) echo 'rank-bronze'; ?>">
            <?= $i+1 ?>
        </span>
        <a href="profile.php?user=<?= urlencode($user['username']) ?>"
            class="<?= ($currentUser["username"] && $user['username'] === $currentUser["username"]) ?
'current-user' : '' ?>">
            <?= htmlspecialchars($user['username']) ?>
        </a>
        <span style="color:var(--text-color-gray); font-size:15px;"><?= $user['total'] ?> items)
    </span>
    </li>
<?php endforeach; ?>
</ol>
</div>
<div class="leaderboard-card">
    <h2>Highest Average Year</h2>
    <ol>
    <?php foreach ($stopAvgYears as $i => $user): ?>
        <li>
            <span class="rank-badge
                <?php if($i==0) echo 'rank-gold';
                    elseif($i==1) echo 'rank-silver';
                    elseif($i==2) echo 'rank-bronze'; ?>">
                <?= $i+1 ?>
            </span>
            <a href="profile.php?user=<?= urlencode($user['username']) ?>"
                class="<?= ($currentUser["username"] && $user['username'] === $currentUser["username"]) ?
'current-user' : '' ?>">
                <?= htmlspecialchars($user['username']) ?>
            </a>
            <span style="color:var(--text-color-gray); font-size:15px;"><?= round($user['avg_year']) ?>
        </span>
        </li>
    <?php endforeach; ?>
    </ol>
</div>
<div class="leaderboard-card">
    <h2>Most Collected Artist</h2>
    <ol>
    <?php foreach ($stopArtists as $i => $row): ?>
        <li>
            <span class="rank-badge
                <?php if($i==0) echo 'rank-gold';
                    elseif($i==1) echo 'rank-silver';
                    elseif($i==2) echo 'rank-bronze'; ?>">
                <?= $i+1 ?>
            </span>
            <a href="profile.php?user=<?= urlencode($row['username']) ?>" class="<?=
($currentUser['username'] && $row['username'] === $currentUser['username']) ? 'current-user' : '' ?>">
                <?= htmlspecialchars($row['username']) ?>
            </a>
            <span style="color:var(--accent-color-saturated); font-size:15px;">
                <?= htmlspecialchars($row['artist']) ?>
            </span>
            <span style="color:var(--text-color-gray); font-size:15px;"><?= $row['artist_count'] ?>
        </span>
        </li>
    <?php endforeach; ?>
    </ol>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```