

**Group Name:** Project SQL: Some Quality Library

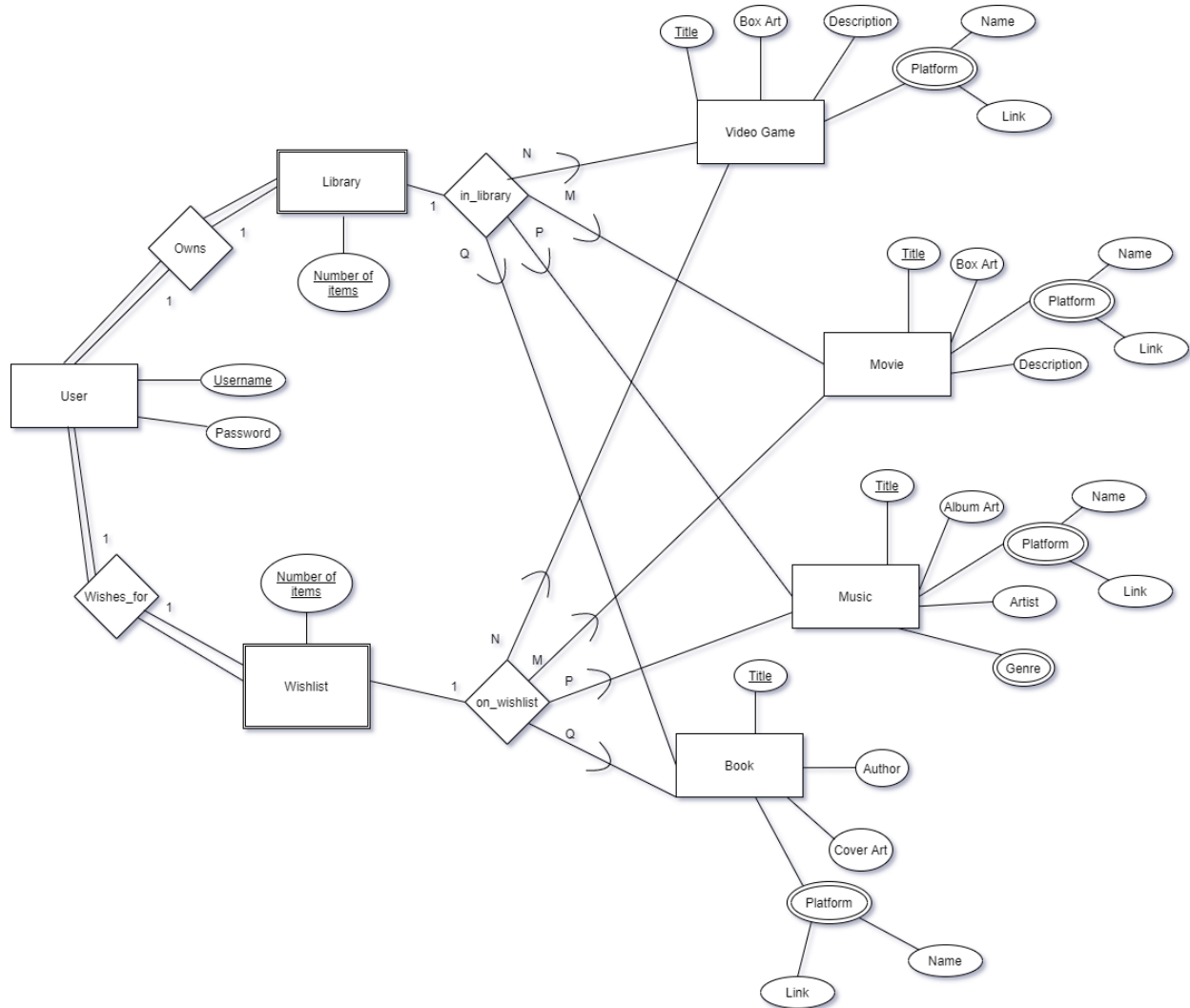
**Group Members:** Cameron Falls, Alex Fahnestock, Henri Evjen

## Problem Statement

Project Idea: Cross Platform Media Manager

Description: This web application provides the ability for someone to store all of their media information with links to the media in one place. Users would have their own accounts and would be able to add, modify, and delete media. Users can also add media to a wishlist that displays media that the user wants but does not currently own. Each media entity would have a general description and a link to where the media is stored at (Steam/Netflix/Spotify/etc.). A database would be necessary for this application because with a database the users could log in and access their media library anywhere. Also a database would allow this application to be scalable. If a user adds a large amount of media entities, it would not take up any storage on the user's local machine since all of the data would be stored on a database.

# Conceptual Database Design



Each user has a library and a wishlist. Each library or wishlist can have different types of media. Each media relationship must have a wishlist or library and one media entity. Each title is unique for each media entity type.

## Changes

- Changed image attributes to be data type string. These attributes now store a link to the image rather than the image itself.
- Changed all string types to CHAR(80)
- Added a modify media box that allows a user to modify media in their library or wishlist, as well as to delete the media from their library or wishlist.

## Entities

- Video Game:
  - Title
  - Box Art
  - Description
  - Platform
  - Link to store page (if applicable)
- Movie:
  - Title
  - Box Art
  - Platform
  - Description
  - Link to movie (if applicable)
- Music
  - Title
  - Album Art
  - Platform
  - Artist
  - Genre
  - Link to Spotify (if applicable)
- Book
  - Title
  - Author
  - Cover Art
  - Platform
  - Link to Amazon (if applicable)
- User
  - Username
  - Password
- Library
  - Number of Items
- Wishlist
  - Number of Items

## Functional Requirements

- Add media to library
- Remove media from library
- Add/remove media from wishlist
- Edit media
- Different views of libraries for different users
  - Library entity owned by users
- User login to view specific library
- Create user
- Delete user
- Each media has page with all its information on it
- Search function to find media in your library/wishlist

## Logical Database Design

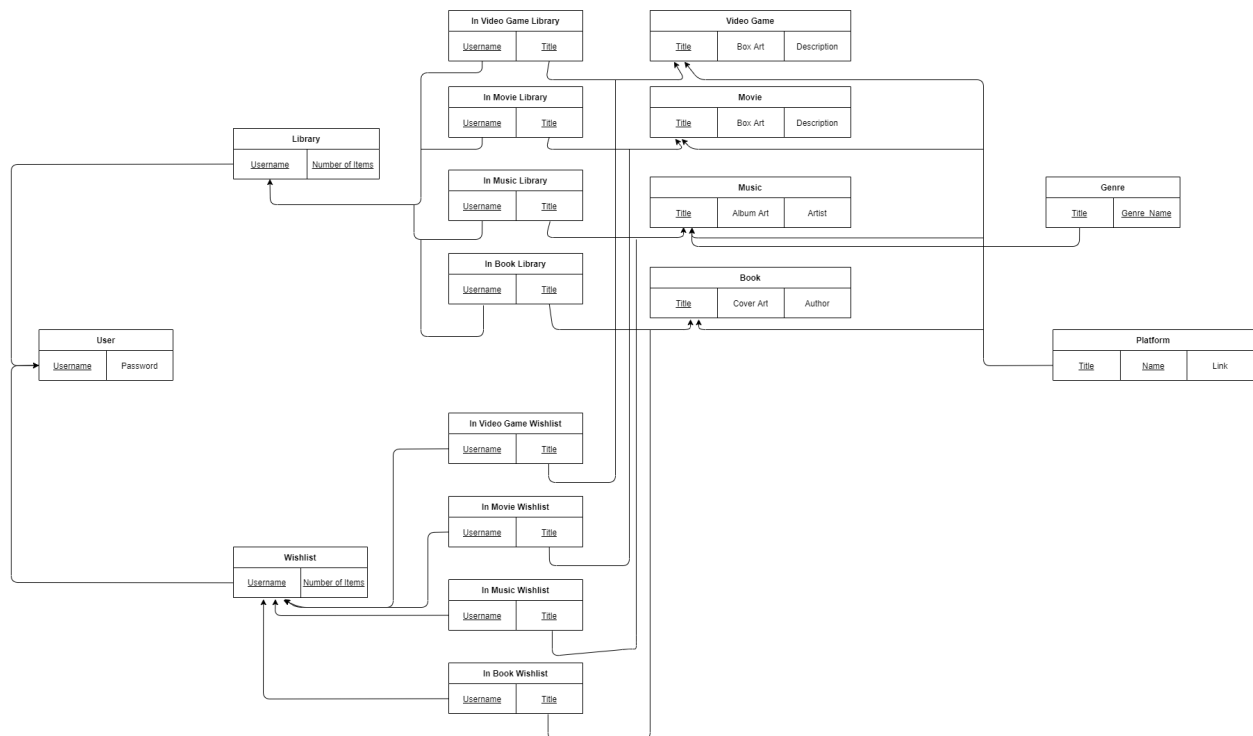


Table	Attribute	Type	Constraint
User	Username	CHAR(255)	Primary Key
User	Password	CHAR(255)	NOT NULL
Library	Username	CHAR(255)	Foreign Key
Library	Number of Items	INTEGER	Primary Key
Wishlist	Username	CHAR(255)	Foreign Key
Wishlist	Number of Items	INTEGER	Primary Key
In Video Game Library	Username	CHAR(255)	Foreign Key
In Video Game Library	Title	CHAR(255)	Foreign Key
Video Game	Title	CHAR(255)	Primary Key
Video Game	Box Art	CHAR(255)	
Video Game	Description	CHAR(255)	
In Movie Library	Username	CHAR(255)	Foreign Key
In Movie Library	Title	CHAR(255)	Foreign Key
Movie	Title	CHAR(255)	Primary Key
Movie	Box Art	CHAR(255)	
Movie	Description	CHAR(255)	
In Music Library	Username	CHAR(255)	Foreign Key
In Music Library	Title	CHAR(255)	Foreign Key
Music	Title	CHAR(255)	Primary Key
Music	Album Art	CHAR(255)	
Music	Artist	CHAR(255)	
Genre	Title	CHAR(255)	Foreign Key
Genre	Genre Name	CHAR(255)	Foreign Key

In Book Library	Username	CHAR(255)	Foreign Key
In Book Library	Title	CHAR(255)	Foreign Key
Book	Title	CHAR(255)	Primary Key
Book	Cover Art	CHAR(255)	
Book	Author	CHAR(255)	
Platform	Title	CHAR(255)	Foreign Key
Platform	Name	CHAR(255)	Primary Key
Platform	Link	CHAR(255)	
In Video Game Wishlist	Username	CHAR(255)	Foreign Key
In Video Game Wishlist	Title	CHAR(255)	Foreign Key
In Movie Wishlist	Username	CHAR(255)	Foreign Key
In Movie Wishlist	Title	CHAR(255)	Foreign Key
In Music Wishlist	Username	CHAR(255)	Foreign Key
In Music Wishlist	Title	CHAR(255)	Foreign Key
In Book Wishlist	Username	CHAR(255)	Foreign Key
In Book Wishlist	Title	CHAR(255)	Foreign Key

# Application Program Design

```
login(username, password)
```

```
    if (username == query users table for match && password == user from user's  
table password)
```

```
        logged_in = true
```

```
    else
```

```
        logged_in = false
```

```
create_user(username, password)
```

```
    if (username not in query users table for match)
```

```
        Execute query to insert into users table (username, password)
```

```
    else
```

```
        create_user = false
```

```
delete_user(username)
```

```
    if (username == query users table for match)
```

```
        Execute query to delete user from users table (username)
```

```
    else
```

```
        delete_user = false
```

```
add_media_to_library(media_obj)
```

```
    // Check to make sure data constraints are met
```

```
    Execute query to add media_obj to corresponding media library table
```

```
remove_media_from_library(media_title)
```

```
    If (media_title not in query media table)
```

```
        Execute query to remove media_title from corresponding media table
```

```
    Else
```

```
        Remove_media = false
```

```
add_media_to_wishlist(media_obj)
```

```
    // Check to make sure data constraints are met
```

```
    Execute query to add media_obj to corresponding media wishlist table
```

```
remove_media_from_wishlist(media_title)
```

```
    If (media_title not in query media table)
```

```
        Execute query to remove media_title from corresponding media table
```

```
    Else
```

```
        Remove_media = false
```

```
modify_media(media_obj)
    // Execute query to find media by media_title
    // Set media_obj in corresponding media table = local variable media_obj

find_media(media_title)
    // Execute query to find media by media_title
    return media_obj
```

## Data Aggregation Functions

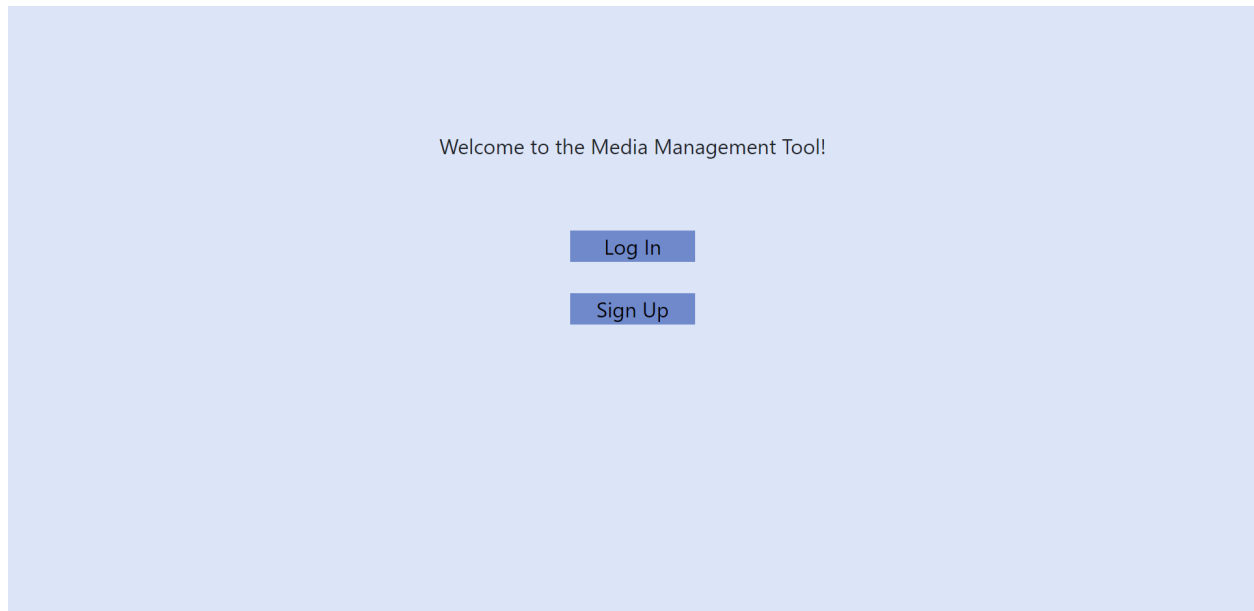
```
sum_of_all_library_media()
    // Iterate over all the media in the library, adding to a total count variable. Once it
    has iterated over all the media in the library, return the count variable
```

```
sum_of_all_wishlist_media()
    // Iterate over all the media in the wishlist, adding to a total count variable. Once it
    has iterated over all the media in the wishlist, return the count variable
```



# User Interface Design

## Welcome Page:



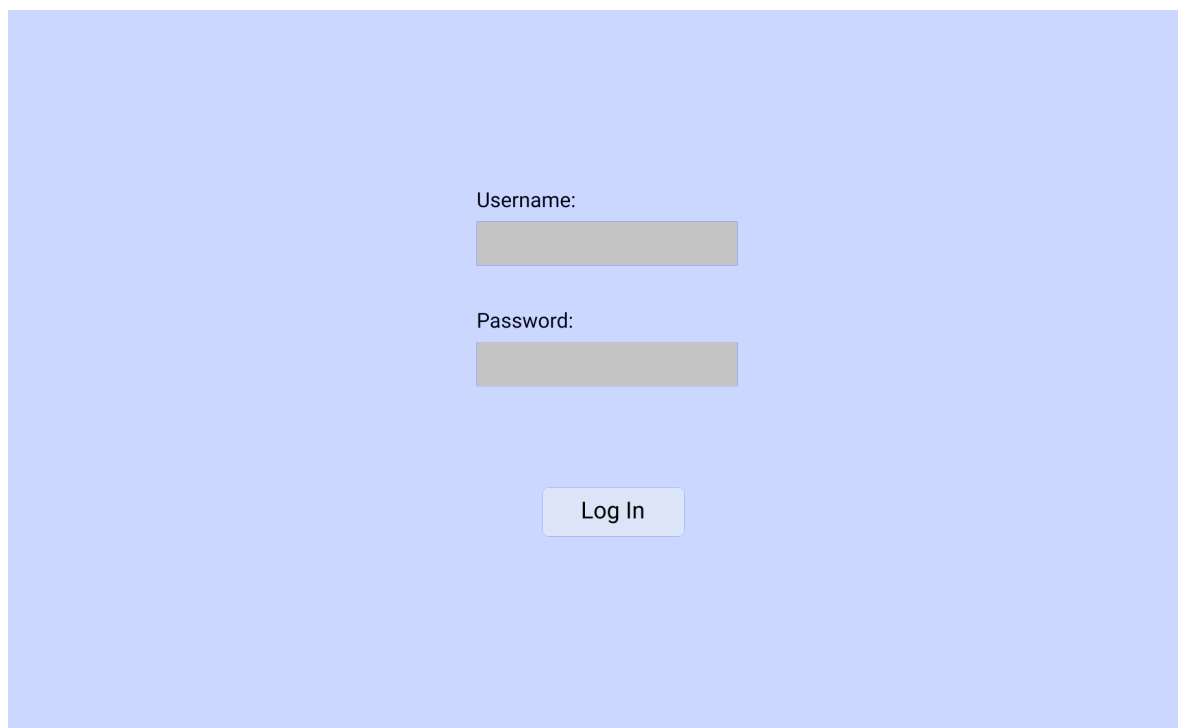
Welcome to the Media Management Tool!

Log In

Sign Up

The welcome page features a light blue background. At the top center, the text "Welcome to the Media Management Tool!" is displayed. Below this text, there are two blue buttons with white text: "Log In" and "Sign Up", stacked vertically.

## Log In:



Username:

Password:

Log In

The login page features a light blue background. It contains two labels, "Username:" and "Password:", each followed by a corresponding input field. The input fields are light gray with a darker gray border. Below the input fields, there is a light blue button with white text that says "Log In".

## Sign Up:

Username:

Password:

Repeat Password:

Sign Up

Library/Wishlist View:

Library

Wishlist

Delete Account

Sign Out

All Types

Sort

+

New Item

?

Title

?

Title

?

Title

?

?

?

?

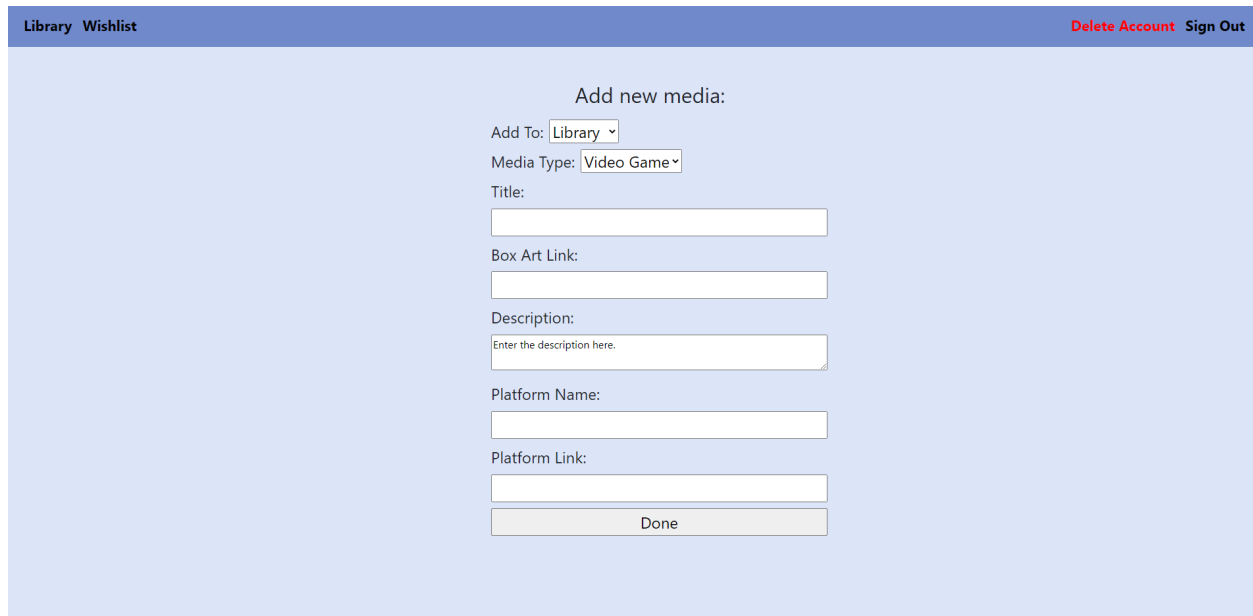
## Delete Account Page:

Are you sure you wish to delete your account? This cannot be undone.

Yes

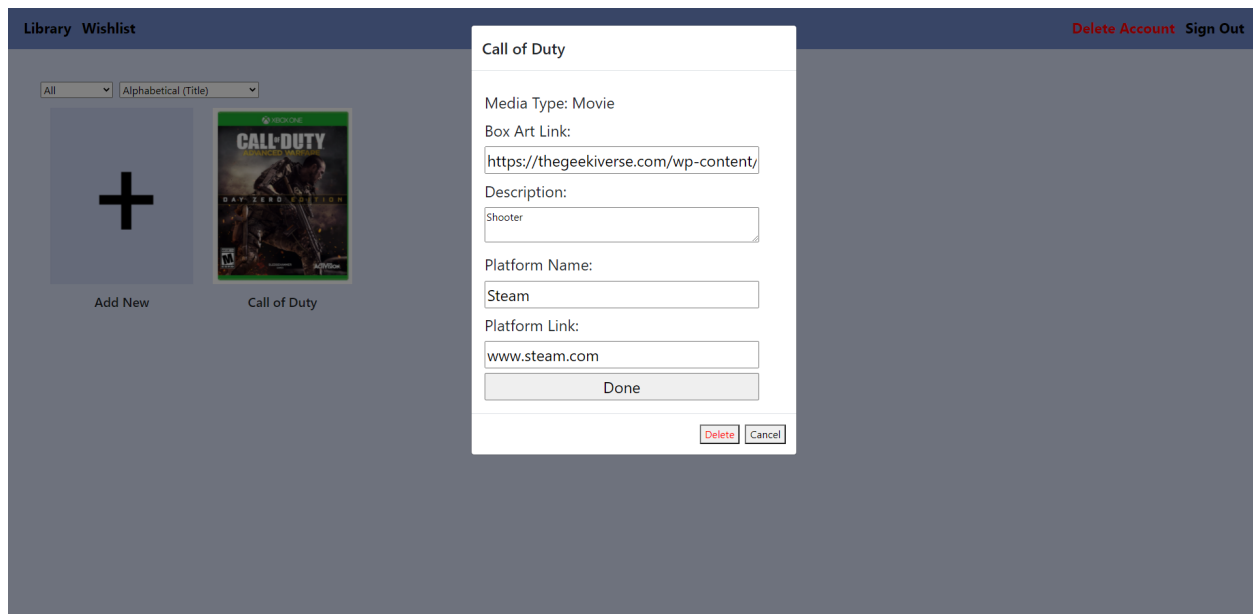
No

## Add New Media Type:



The screenshot shows a web application interface with a dark blue header bar containing 'Library' and 'Wishlist' on the left, and 'Delete Account' and 'Sign Out' on the right. The main content area is light blue and contains a form titled 'Add new media:'. The form includes the following fields: 'Add To:' with a dropdown menu set to 'Library', 'Media Type:' with a dropdown menu set to 'Video Game', 'Title:' with a text input field, 'Box Art Link:' with a text input field, 'Description:' with a text area containing the placeholder 'Enter the description here.', 'Platform Name:' with a text input field, and 'Platform Link:' with a text input field. At the bottom of the form is a 'Done' button.

## Modify Media Box:



The screenshot shows a web application interface with a dark blue header bar containing 'Library' and 'Wishlist' on the left, and 'Delete Account' and 'Sign Out' on the right. The main content area is dark grey and contains a modal dialog titled 'Call of Duty'. The dialog has a 'Media Type: Movie' label. It includes the following fields: 'Box Art Link:' with a text input field containing 'https://thegeekiverse.com/wp-content/', 'Description:' with a text input field containing 'Shooter', 'Platform Name:' with a text input field containing 'Steam', and 'Platform Link:' with a text input field containing 'www.steam.com'. At the bottom of the dialog are 'Delete' and 'Cancel' buttons. In the background, a library view is visible with a dropdown menu set to 'All' and 'Alphabetical (Title)', and two items: 'Add New' (represented by a plus sign) and 'Call of Duty' (represented by a game cover).

## Installation Instructions (works on any OS)

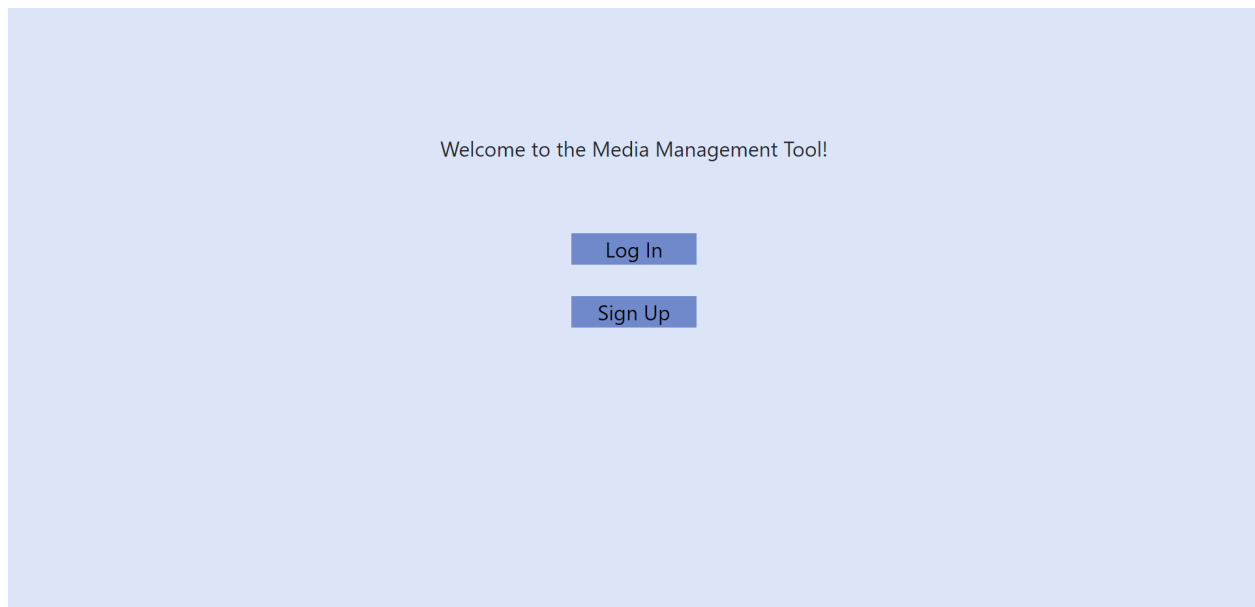
- 1) Install MySQL on computer using the instruction PDF that was provided to us
- 2) Install the latest version of Node.js from <https://nodejs.org/en/download/>

- 3) Unzip project folder
- 4) In Database\_project/app/config/db.config.js, put your MySQL password where it says "<put\_password\_here>"
- 5) In Database\_Project directory, type "npm install"
- 6) In Database\_Project directory, type "npm start" to run the backend with the MySQL database
- 7) In Database\_Project/front-end directory, type "npm install" (may take a few minutes)
- 8) In Database\_Project/front-end directory, type "npm start" to start the client front end

## User Manual

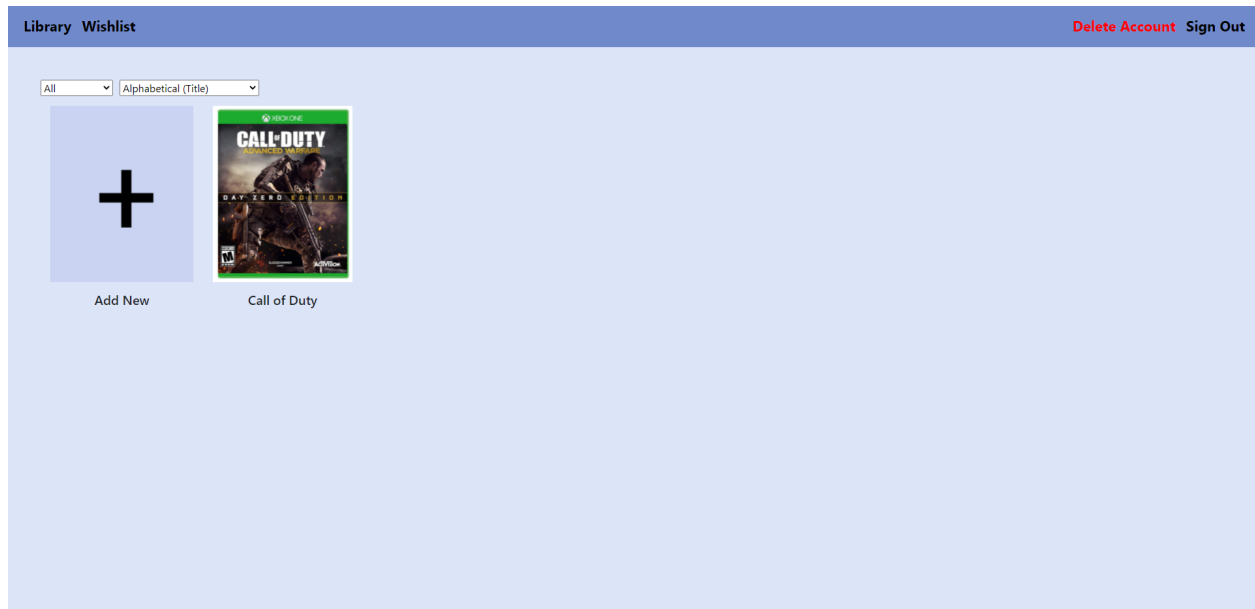
### Pages:

Welcome Page: On this page, the user is displayed a welcome message and is prompted to log in or sign up. This is the landing page for the site. At the top of all pages on this site is the navigation bar that allows quick access to the library, wishlist, delete account, and sign out pages.

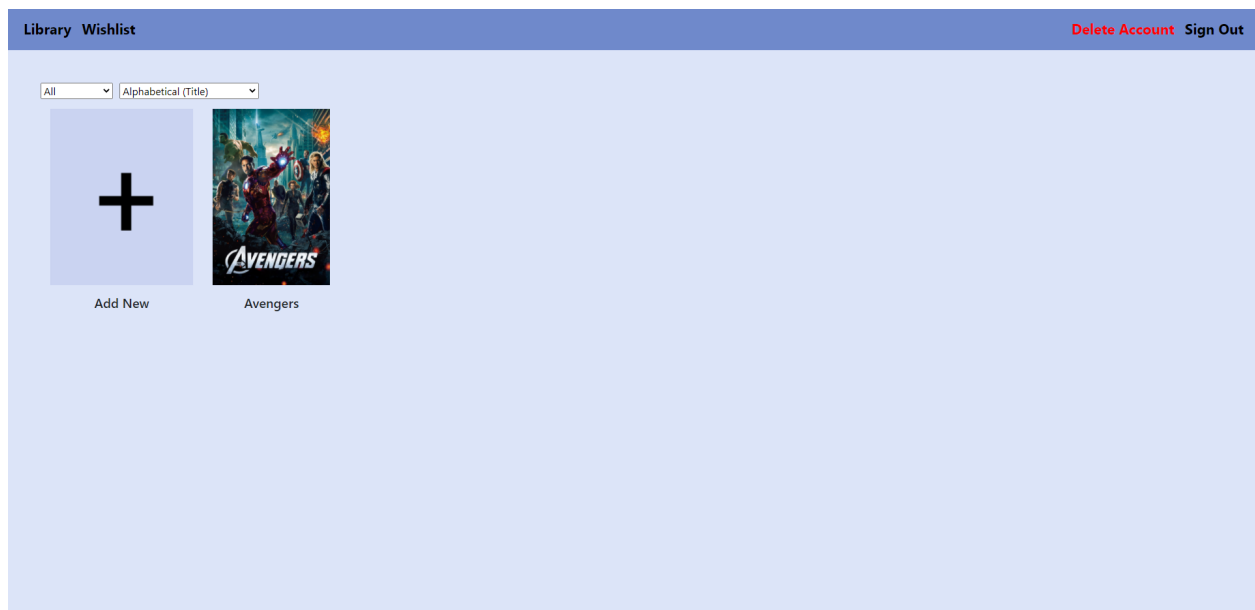


Library Page: On the library page, the user can view all the media in their library. The user can also choose to add a new piece of media by clicking the "add new" button. The

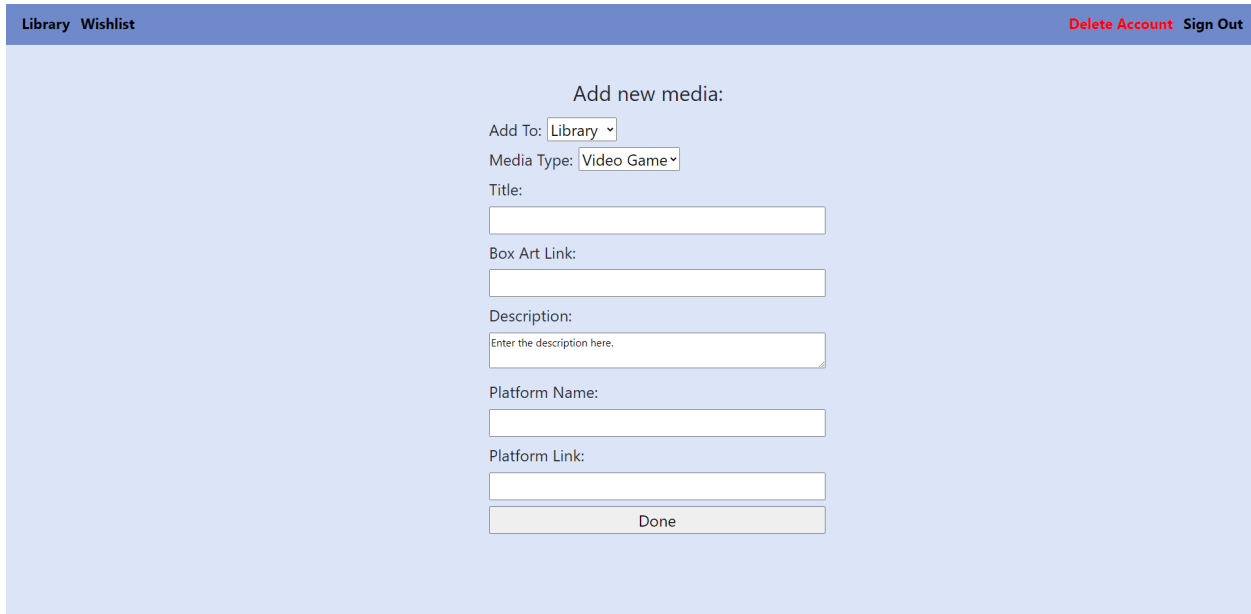
media in the library can be filtered by type and sorted by alphabetical or reverse-alphabetical order of the title.



Wishlist Page: On the wishlist page, the user can view all the media on their wishlist. The user can also choose to add a new piece of media by clicking the “add new” button. The media in the wishlist can be filtered by type and sorted by alphabetical or reverse-alphabetical order of the title.



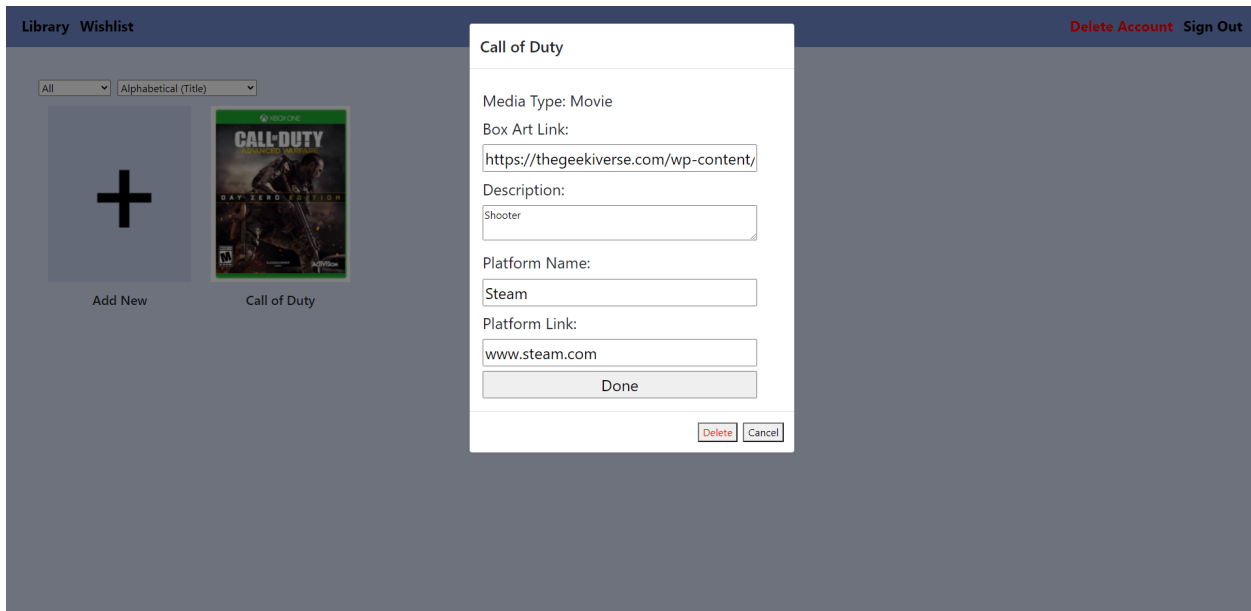
**Add New Media Page:** On the add new media page, the user can fill out a form to add a new piece of media to their library or wishlist. This form contains all the information about the media.



The screenshot shows the 'Add new media' form on the 'Library Wishlist' page. The form is centered on a light blue background. At the top right of the page, there are links for 'Delete Account' and 'Sign Out'. The form fields are as follows:

- Add To:** A dropdown menu with 'Library' selected.
- Media Type:** A dropdown menu with 'Video Game' selected.
- Title:** A text input field.
- Box Art Link:** A text input field.
- Description:** A text input field with the placeholder text 'Enter the description here.'
- Platform Name:** A text input field.
- Platform Link:** A text input field.
- Done:** A button at the bottom of the form.

**Modify Media Box:** By clicking on a media item you can open the edit media popup modal. Here you can edit or delete your media item.



The screenshot shows the 'Call of Duty' media item edit modal. The modal is a white box with a title bar 'Call of Duty'. It contains the following fields:

- Media Type:** Movie
- Box Art Link:** <https://thegeekiverse.com/wp-content/>
- Description:** Shooter
- Platform Name:** Steam
- Platform Link:** [www.steam.com](http://www.steam.com)
- Buttons:** 'Done', 'Delete', and 'Cancel'.

In the background, the 'Library Wishlist' page is visible, showing a grid with an 'Add New' button (a plus sign) and a 'Call of Duty' media item.

**Delete User page:** On this page, the user is displayed a message asking them to confirm whether they wish to delete their account. If they choose yes, they are taken to the welcome page. If they choose no, they are taken back to their library.



[Library](#) [Wishlist](#) [Delete Account](#) [Sign Out](#)

Are you sure you wish to delete your account? This cannot be undone!

Yes

No

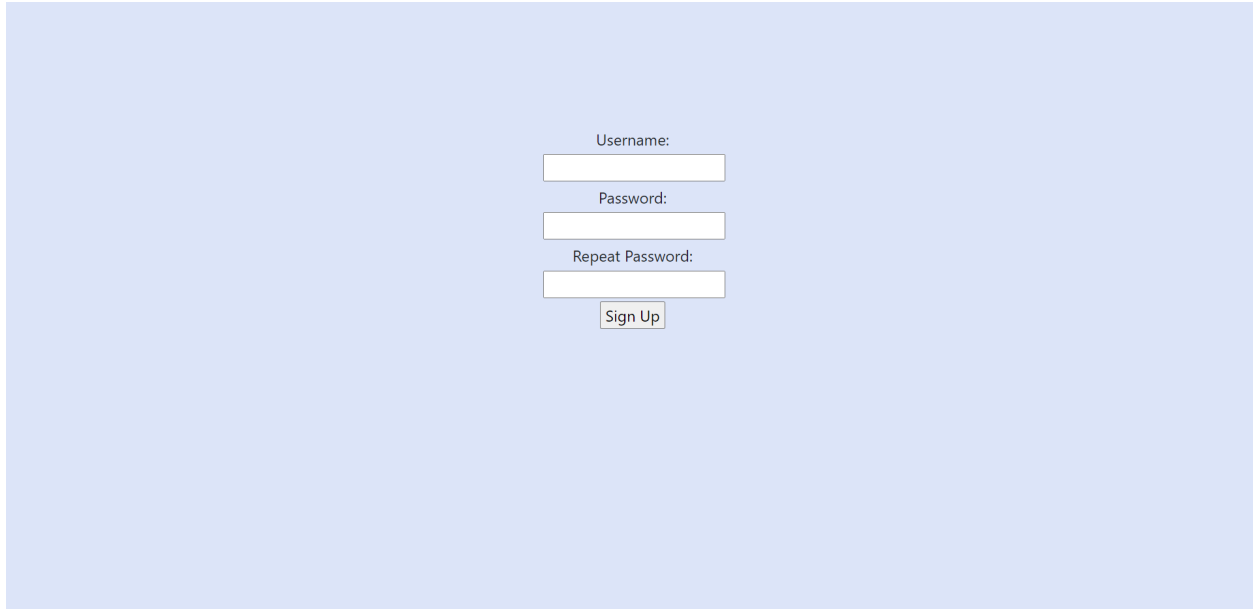
Login Page: On the login page, the user can enter their username and password to login. Once the user submits successfully, they are taken to their library page.

Username:

Password:

Log In

Sign Up Page: On the sign up page, the user can enter their username and password to sign up. The password must match with the repeat password. Once the user submits successfully, they are taken to their library page.



Username:

Password:

Repeat Password:

## Data Aggregation

To find the count of a particular media type in a library or wishlist, use the api link:  
[http://localhost:8080/api/{media\\_type}/{library\\_or\\_wishlist}/count](http://localhost:8080/api/{media_type}/{library_or_wishlist}/count)

## Usage

To use the application (after completing installation instructions):

1. From the Database\_Project directory, issue the command `npm start` to start the database service.
2. From the Database\_Project/front-end directory, issue the command `npm start` to start the web application.