

# **Final Project**

# **Proposal**

The Album Archive

04/06/2025

**Prepared For**

COM S 3190 – Construction of User Interfaces

Iowa State University Computer Science Department

**Prepared By**

Owen Gilbertson

William Frommelt

# Table of Contents

1. Introduction
2. Purpose of the Proposal
3. Goals and Objectives
4. Project
5. Project Path Selection
6. Ownership/Responsibility
7. Resources
8. File Structure
9. Data Sources and Management
10. User Experience Views
11. Final Comments

[\*\*1. Introduction\*\*](#)

We, Will Frommelt and Owen Gilbertson, are on team AB7, and are Sophomore students majoring in Software Engineering. Throughout the course we will continue to learn valuable skills and languages in web development such as HTML, CSS, JavaScript, React, MySQL, and CRUB usage. Because we have known each other for 4 semesters now, we have had the privilege of building a friendship through some of the courses we have taken together. We believe that friendship is valuable for communicating and working through difficulties in this project.

## 2. Purpose of the Proposal

We aim to develop a website where users can log in or sign up to view “The Album Archive.” “The Album Archive” is a space where an extensive list of albums, not limited to what is shown initially, are displayed with their respective cover art, information/details, ratings, and links to them. Along with this display, there will be a “my collections” page within our website where users can custom pick which albums/tracks they want in their collection. The user will be left with a space they can easily interact with, like the look of, and keep wanting to return to.

## 3. Goals and Objectives

Our project's goals and objectives include:

- Creating an aesthetically pleasing and easy-to-use website
- Requiring credentials to log in or sign up to view our webpage
- Make a list of albums users can browse through with their respective cover art, information/details, ratings, and links to them.
- Including a “my collections” tab in the navigation bar where the user can see which albums they have added or removed from their collection.
- Including an additional to add albums to the website for Developers to update the webpage if needed.
- Creating a navbar that complements the webpage and is easy to use.
- Include buttons users can interact with to navigate or contribute to functionality.
- Include a confirmation page for removing/adding an album to their selection
- Include an about us page for us authors, so that users can reach out with ideas or feedback
- Maybe add filter to search for certain genres so the user doesn't have to scroll through unwanted parts of the webpage.

## 4. Project

Our website, “The Album Archive,” will initially take the users to a login page; if they don’t have one, it will take them to a sign-up page. The sign-up page will be in Form format and filled out using the user’s name, new username, and password. Once they submit this form, their credentials will be stored in a MySQL database so the user can log in instead of signing up the next time they visit our page. Once users have logged in, they will be presented with a catalog-style variety of albums. They will be sorted based on the genre. Each album will display a small image of their respective cover art, name of album, artist/s, and general rating, along with links to them/or integrate a listening feature if possible. There will also be a navbar included on this page that has the options: “About us” and “My collection” and an additional “Add album” page if it’s an Admin logged in so that the albums can be updated. “My collection” shows the album users selected to add to their collection. “Add album” will make the user fill out a form that updates the webpage in submission to include the new album. The last page we will have is a confirmation page that displays the album a user is trying to add or remove to their collection. To recap, our six pages are a login page, a sign-up page, a main page listing albums, a “My collections” page displaying the user’s selected albums, an “About Us” page with information about the authors of the webpage, and a confirmation page for adding and removing albums to the user’s collection.

### Key Components:

- A login page that asks for a username and password. Using CRUD functionality, the credentials will be looked for in MySQL. The initial page will be displayed in React.
- A sign-up page that asks users to create a username and password. Using CRUD functionality, the credentials will be added in MySQL. The initial page will be displayed in React.
- A main page that displays a variety of different albums. Each album display includes cover art, information/details, ratings, and links to them. Album information will be retrieved using CRUD functionality connected to the MySQL database. Then, the data will be displayed in React.
- A “My Collection” page that displays the albums users have selected to add. Album information will be retrieved using CRUD functionality connected to the MySQL database. Then, the data will be displayed in React.
- An “About Us” page that displays the authors and information about them. Displayed in React.

- A confirmation page that confirms what album is being added or removed from a user's collection. The information displayed on this page is dynamically allocated using useState() or useEffect(). Displayed in React.
- A navbar used to navigate back and forth through the pages smoothly. Using React.
- useState() and useEffect() functions that update the list of albums when an Admin adds to it
- Updating variable's states as users interact with the website
- Using Node.js with Express to store routes for the CRUD requests, being used to interact with the MySQL database.
- Button functionality to add albums to user's collections
- An easy-to-use and aesthetically pleasing design

## 5. Project Path Selection

We decided to go with Option 2 and start a new Project. We did this to have a fresh start with the latest material and wouldn't have to worry about changing previous functionality within our midterm project. It will also give us a better understanding of what starting a project like this from scratch will look like. Starting over, we decided to shift our focus to a different topic. We figured it would be different enough to make it enjoyable to work with and even added a few more openings for new features. So, we came up with the idea of "The Album Archive."

A website that displays music albums with cover art, information/details, ratings, and links to them, along with some features. The first feature is when users open the website. They are prompted to enter their login; if they don't have one, they are redirected to a sign-up page. Once they have gotten into the website, a navbar with "My Collection," "About Us," and "Add album," depending on if they have Admin status, will be displayed along with several albums and their respective details. They will be separated by genre; if the user wants, they can add the album to their collection by clicking the add button. The user will then get a confirmation screen to confirm that they wish to add the album to their collection. Once they confirm that, it will be displayed in the "My Collection" section of the navbar. The "About Us" section displays information about the authors. If the user is an Admin, they will have the additional "Add album" selection to input the corrective information to add another album to the website. More in-depth descriptions can be found in the Project section of this document.

## 6. Ownership/Responsibility

**NOTE:** We plan on conversing on all different pages and responsibilities, to keep each other up with what is going on our parts of the project. We also will not be afraid to help one another if one of us is having problems with what we are working on. These responsibilities can be subject to change as well, but this is our original outline.

- Login + Sign Up Page – Handled by Will (Frontend + Backend): Allows Users to log in or sign up for an account that will save the users data on to their account, and they can access their own music through it.
- Add an Album Page– Handled by Owen (Frontend + Backend): Allows for a user to add an album to their list by filling out a form on this page that will contain details of the music. Important because this is the first step in adding music to the main page.
- Confirmation Page – Handled by Owen (Frontend + Backend): This confirms all of the data is correct from the Add an Album page, and allows you to officially to add it to the website.
- Main Page – Handled by Both: will be the start of the project, and the part we will work together most, and will take the most communication design wise, and everything else will be stemmed off this design. One might work mainly on Nav Bar of the website, while the other sets up the albums for the website.
- About Us Page – Handled by Will (Frontend + Backend): Will contain information about the authors of the website, including images and details.
- My Collection Page – Handled by Owen (Frontend + Backend): Displays only the albums selected by the user on that particular profile.

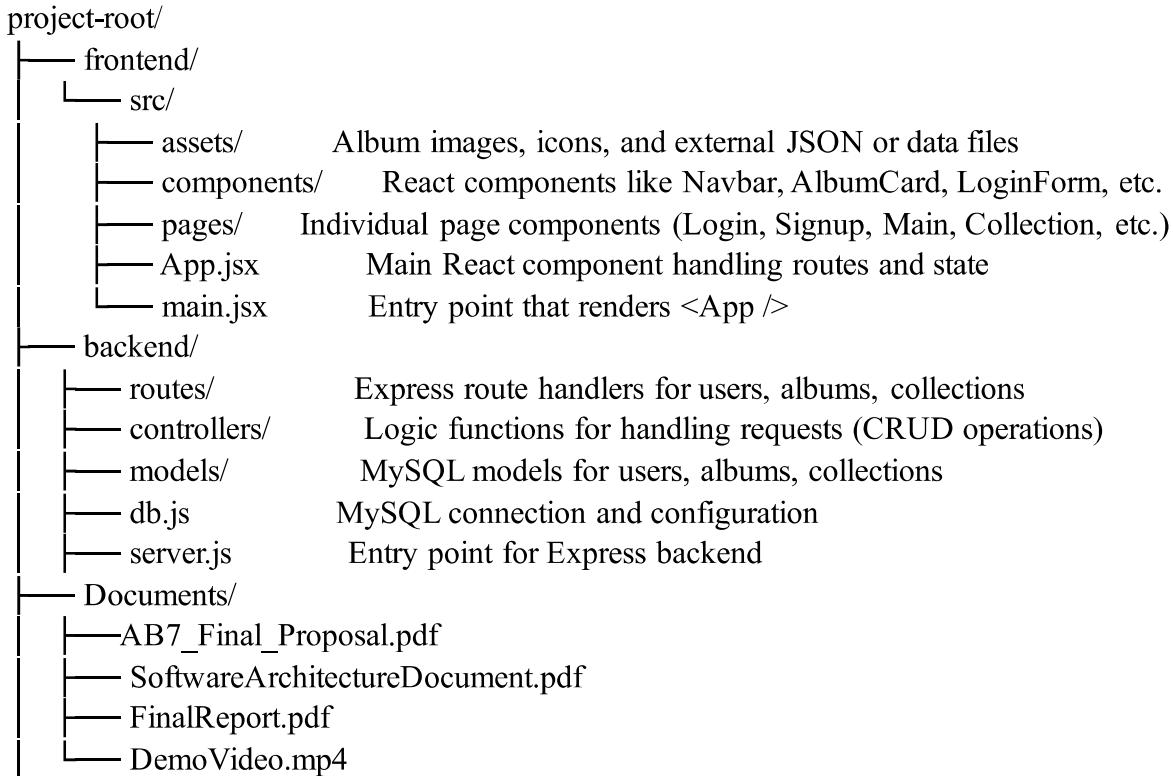
## 7. Resources

We will utilize the following tools, libraries, and frameworks to build our Website:

- Gitlab – For source control, issue tracking, and team collaboration
- React – for building all frontend components and creating interactive pages
- Node.js and Express – to create the backend server and API endpoints for CRUD operations
- CRUD – using crud to access and interact MySQL database
- MySQL – for managing user data, album info, and user collections in a relational database
- useState() / useEffect() - Used for managing state of main after adding albums
- Excalidraw – used for planning site structure and architecture visually

Weekly Time commitments: Not super specified per week, because it depends on the workload of the week: Will probably be around 10 hours a week per person, and more if needed on some of the more difficult implementation and design processes.

## 8. File Structure



## 9. Data Sources and Management

Our application uses three main sources of data:

### 1. User Input

- Users enter data during account creation, login, and when adding/removing albums from their collection.

### 2. Local Seed Data

- A preloaded set of music albums is stored in the database on initialization.
- This seed data populates the homepage album feed for all users.

### 3. Database (MySQL)

- Persistent storage for users, albums, and user collections.
- MySQL is managed through a backend Node.js/Express API.

## CRUD Operations

### Create

- **User signup:** POST /api/signup
  - Adds a new user to the database.
- **Add album to collection:** POST /api/users/:id/collection
  - Adds a row in the collections table linking user ID and album ID.

### Read

- **Login validation:** POST /api/login
  - Checks credentials against stored users.
- **Fetch albums:** GET /api/albums
  - Returns all albums from the database.
- **Fetch user collection:** GET /api/users/:id/collection
  - Returns list of albums a user has saved.

### Update

- Optional: Could support editing user info or admin editing album info via:
  - PUT /api/users/:id
  - PUT /api/albums/:id

### Delete

- **Remove album from collection:** DELETE /api/users/:id/collection
  - Removes the user-album link in the collections table.

## 10. User Experience Views

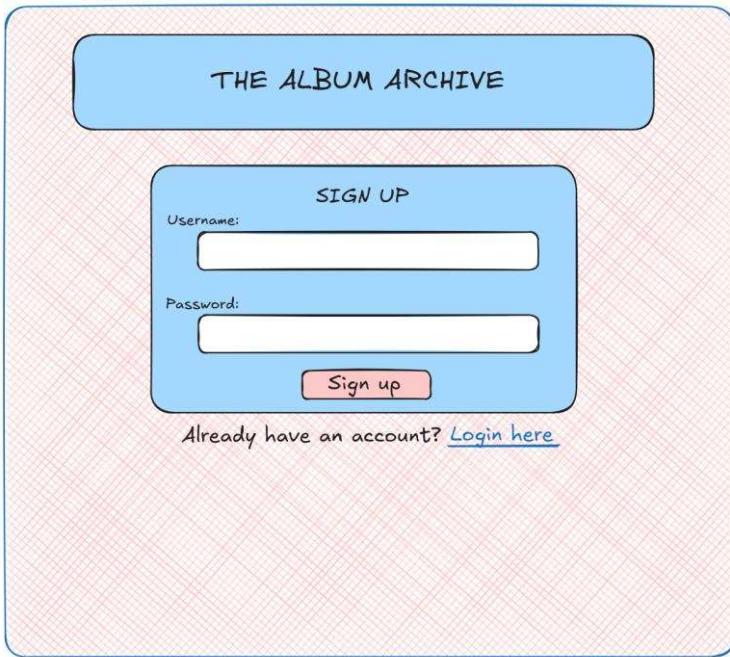
Login page:

On this page users enter their username and password to login into the website. Once they have entered that information, they may then enter the website by clicking the Login button. However, if they do not have a username and password, they must click the Sign-up here link which takes them to the Sign Up page.



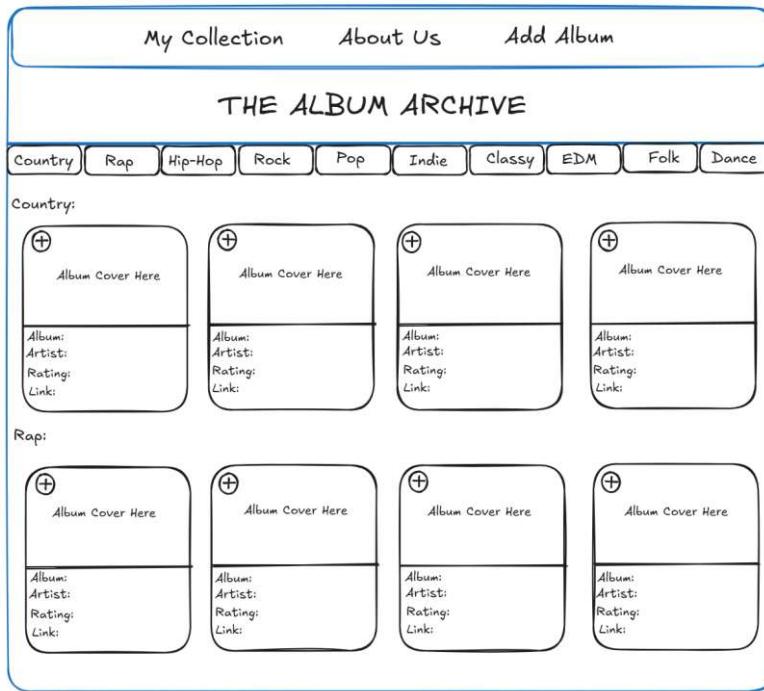
### Sign Up page:

On this page users enter a username and password that they want to use for their new account. Once they have filled in the fields, they can click the Sign up button to add the account to the websites database, giving their credentials access to the webpage when they try and login next time. If they already have an account and are on this page, they then can click the Login here link to get back to the Login page.



Main page:

On this page there is a navigation bar with 3 different buttons that lead to three different pages. Clicking My Collection takes the user to their collection of albums that they have added. Clicking About Us takes the user to the About Us page that talks about the authors. The Add Album text only appears if the user logged in is an Admin. If the Admin clicks Add Album they are taken to a screen where they are given the option to add an album to the archive options. The different genres below the text "THE ALBUM ARCHIVE" are buttons that navigate the user to different genres of albums. The last button on the screen is the + button on the top left of every album, this takes the user to a screen where they can confirm that they want to add the album to their collection.



### Add Album page:

On this page the user/Admin may enter information about an album they want to add to The Album Archive. Once every field is filled out the user can then click the Add Album button, and it will successfully add the new album selection to the Album Archive. If the user would not like to add an album, they may click the cancel button, and it will take them back to The Album Archive.

### Add an Album to the Archive

Enter the Album information

Genre:

Select a genre

Link to Album Cover:

Album Name:

Artist Name:

Rating:

Select a rating

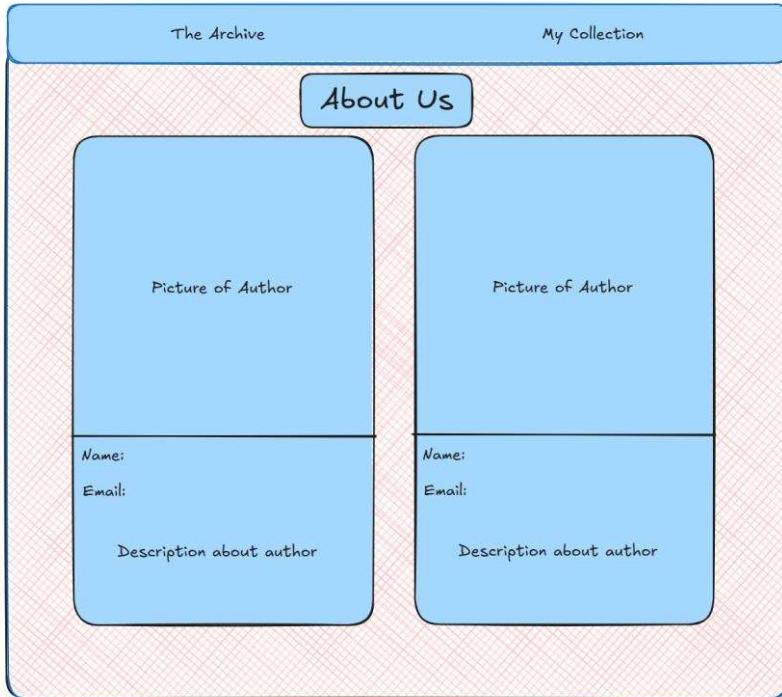
Link to the Album:

[Cancel](#) [Add Album](#)

This is a wireframe representation of a form titled "Enter the Album information". The form contains several input fields: a dropdown menu for "Genre" with the placeholder "Select a genre"; a text input field for "Link to Album Cover"; a text input field for "Album Name"; a text input field for "Artist Name"; a dropdown menu for "Rating" with the placeholder "Select a rating"; and a text input field for "Link to the Album". At the bottom of the form are two buttons: a pink "Cancel" button and a pink "Add Album" button.

#### About Us page:

On this page the user is presented with information about the authors of the website. The user can click The Archive text, and it will take them back to The Album Archive or the user can click the My Collection text, and it will take them to their collection of albums.



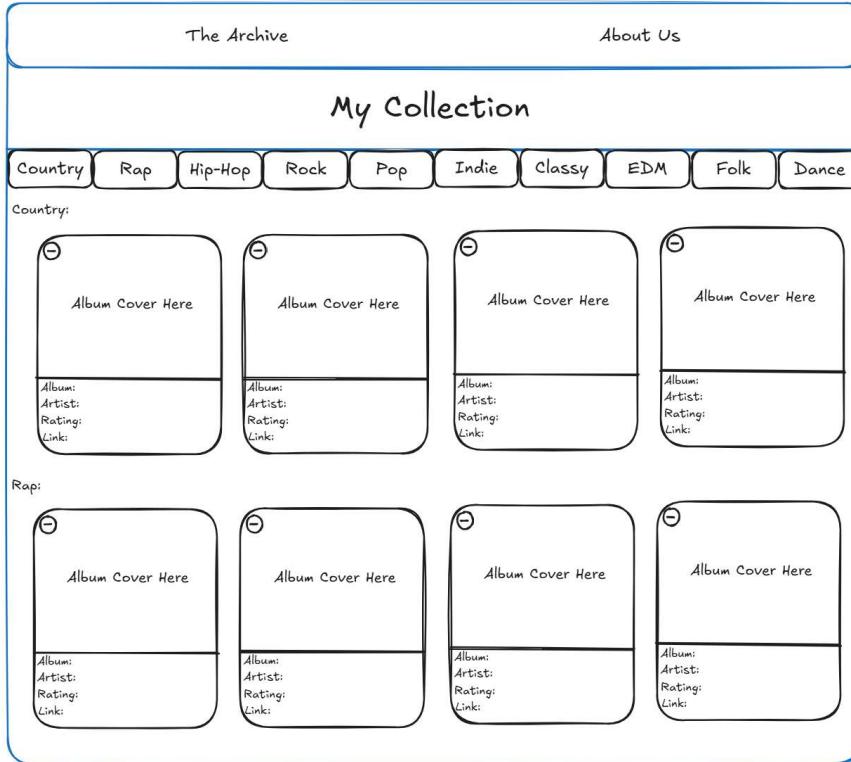
### Confirmation Page:

On this page the user can click the Add button if they want to add the album that is being presented to them to their collection, or they can click the cancel button which will take them back to The Album Archive. If the user is removing an album from their collection, then the Add button will change to a Remove button, and once that is clicked the album will be removed from their collection. Otherwise, if they hit the cancel button, they will be taken back to their collection.



### My Collection page:

On this page the user may click The Archive text, and it will take them back to The Album Archive. Clicking the About Us text will take the user to the About Us page. The user also has a row of different genre buttons they can click that will navigate them to the corresponding genre in their collection. Clicking the – button in the top left of an album cover will take the user to the Confirmation page where they can decide if they want to remove the album from their collection or not.



## 11. Final Comments

This is our proposal for The Album Archive. We aim to improve our knowledge while making a website that nicely displays albums and is easy for people to use. In the future, after we learn more about the construction of user interfaces, we could expand on the functionality, improving the website even more. Thank you for considering our proposal.

We look forward to starting development and taking suggestions along the way.