CrossTune

Team Members

Shane Cao
Github: Cao-Shane
shca6876@colorado.edu

James Bowen
Github: jabo2204
jabo2204@colorado.edu

Kendrix Henderson
Github: Jackkhuncho
elhe7727@colorado.edu

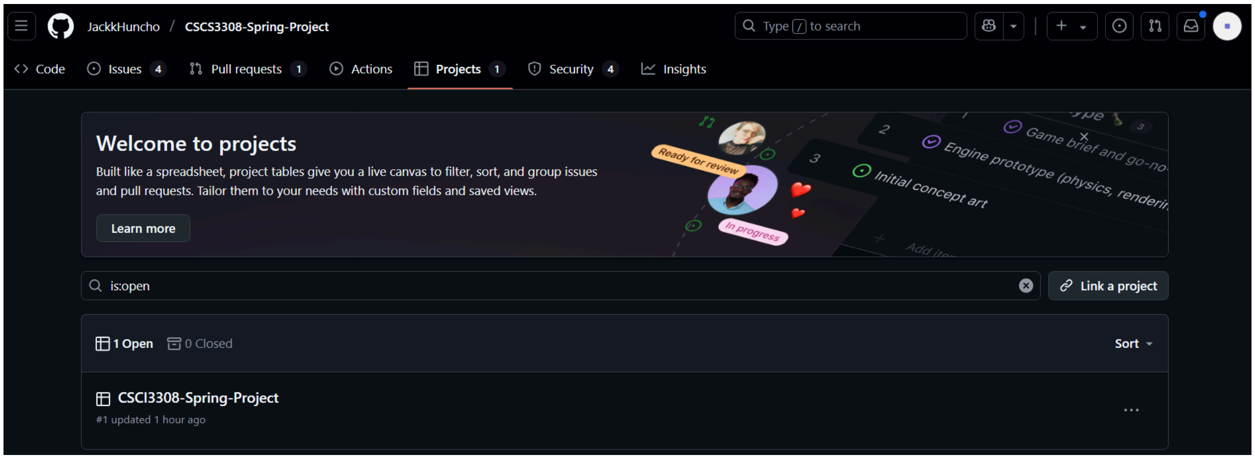
Renner Skidmore
Github: rennerskidmore
resk7605@colorado.edu

Project Description

CrossTune makes it extremely simple to share Spotify and Apple Music playlists across platforms bridging the gap between Spotify and Apple Music users. CrossTune allows its users to share their new playlists to other users with ease. This application will convert cross platform playlists to their respective formats, ensuring that songs are matched accurately between Spotify and Apple Music. Users may connect their Apple Music or Spotify accounts through authentication allowing them to convert or add another user's playlist with a click of a button. The social aspect of our application allows our users to post their playlists with their respective platforms to share to other CrossTune user's who can view each other's profiles and post history. CrossTune's sleek user interface and well developed database opens the door in the music industry by encouraging the celebration of music. The divide between platforms that companies have created for consumers will be lifted with CrossTune, allowing music enthusiasts to share their love for music, regardless of platform differences. Whether they're discovering new music or sharing their favorite tracks with friends, CrossTune makes the process seamless.

Project Tracker

https://github.com/users/JackkHuncho/projects/1



Welcome to projects

Built like a spreadsheet, project tables give you a live canvas to filter, sort, and group issues and pull requests. Tailor them to your needs with custom fields and saved views.

is:open

1 Open 0 Closed

CSCI3308-Spring-Project

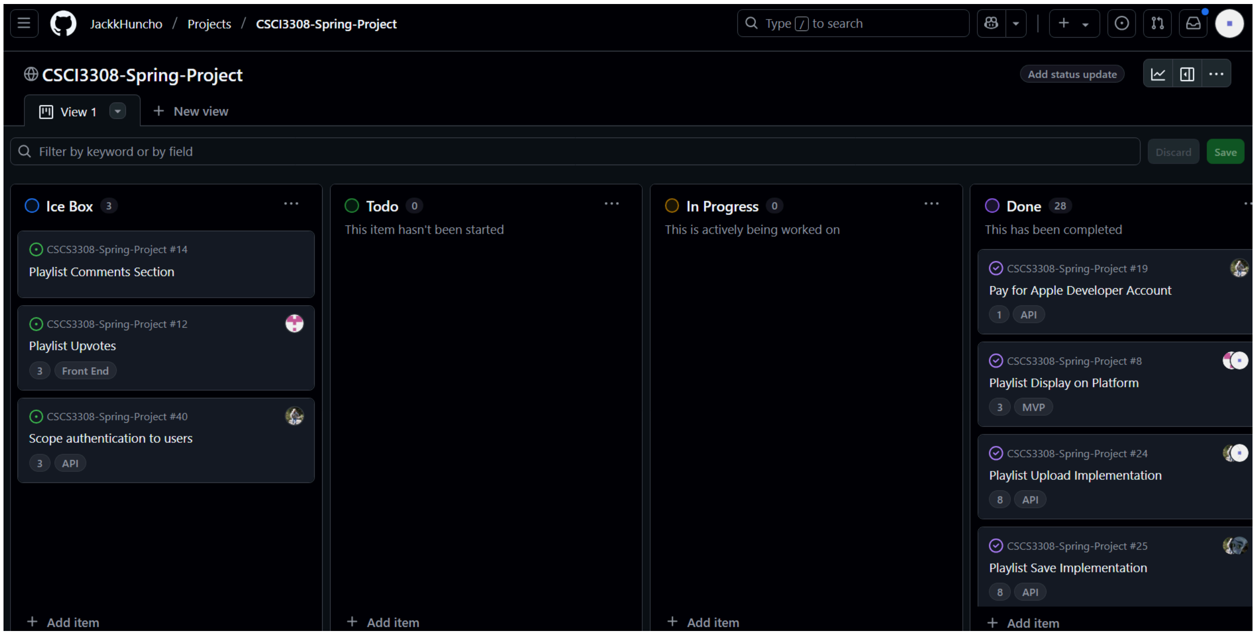
#1 updated 1 hour ago

Video:

https://drive.google.com/file/d/11ewuKFlQmOLD0_eBxt4W0jO83fyuNAsy/view?usp=sharing

Version Control: https://github.com/JackkHuncho/CSCS3308-Spring-Project.git

Contribution Resources



CSCI3308-Spring-Project

Ice Box 3

- CSCS3308-Spring-Project #14
  Playlist Comments Section
- CSCS3308-Spring-Project #12
  Playlist Upvotes
  3 Front End
- CSCS3308-Spring-Project #40
  Scope authentication to users
  3 API

Todo 0

This item hasn't been started

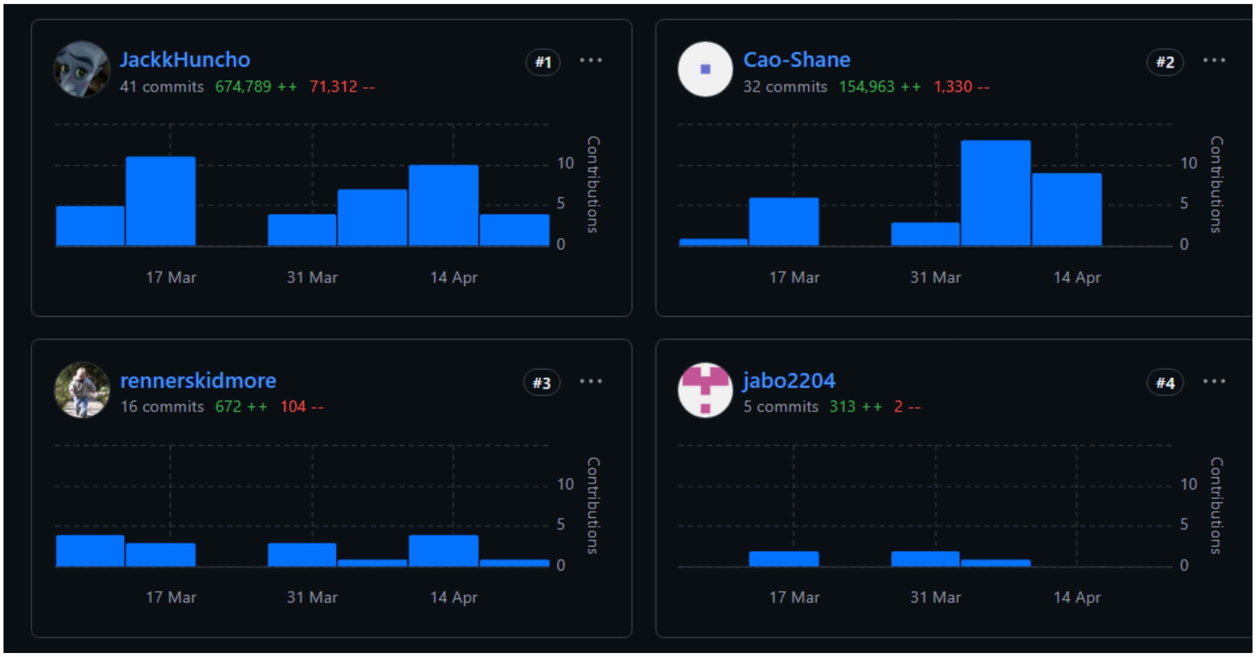
In Progress 0

This is actively being worked on

Done 28

This has been completed

- CSCS3308-Spring-Project #19
  Pay for Apple Developer Account
  1 API
- CSCS3308-Spring-Project #8
  Playlist Display on Platform
  3 MVP
- CSCS3308-Spring-Project #24
  Playlist Upload Implementation
  8 API
- CSCS3308-Spring-Project #25
  Playlist Save Implementation
  8 API



JackkHuncho
41 commits 674,789 ++ 71,312 --

Cao-Shane
32 commits 154,963 ++ 1,330 --

rennerskidmore
16 commits 672 ++ 104 --

jabo2204
5 commits 313 ++ 2 --

Team Contributions

Shane Cao

Focused on the front end of the application ensuring the user interface worked smoothly and was intuitive from a user perspective. Utilized css, html, bootstrap, and JavaScript to ensure the database worked well with the front end and the application was visually appealing on both mobile and desktop. Added features such as allowing users to post their playlist and it appearing on the homepage with all their songs and the original poster's info. Ability to click on user's profiles and see their live posts as well as their own while also being able to click on posts into a modal.

Kendrix Henderson

Worked mainly on the back end of the application dealing with database design and functional routes interfacing with API integrations. Learning tools such as axios, multer, Spotify/Apple API's, and asynchronous JavaScript techniques, features such as profile picture changing, and playlist conversion were able to be implemented. In addition, Kendrix helped overall with the codebase and making sure the feature integration was well maintained. Outside of the back end, Kendrix was in charge of cloud deployment, and maintaining the structure of the application.

James Bowen

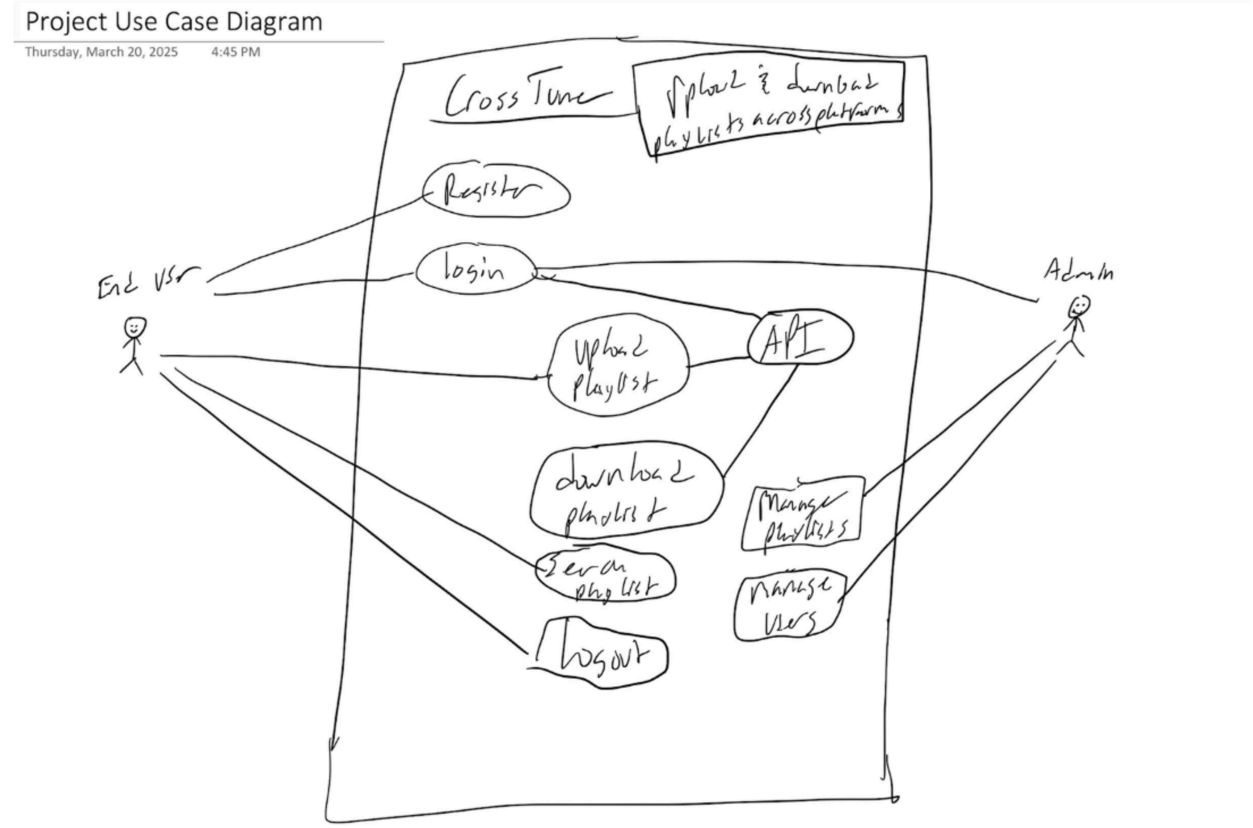
Worked on creating partials to display people's playlists and information when posted on the site. Made routes to update databases with user id and post id information when posts are created so posts can be matched and displayed with a user's profile. Made test cases for the login route using mocha and chai. Experimented with an upvote and playlist sorting feature that ultimately wasn't able to make it to the final product.

Renner Skidmore

Renner was scrum master, team lead, and focused on API connections. During the project Renner assigned stories, helped other team members determine story points, and created epics. In meetings he also updated the project board to be up to date. As team lead Renner supported the rest of the team with the stories they were focused on through expediting communication, scheduling meetings, testing code, and approving pull requests. When working on APIs Renner started the foundational understanding, purchased the Apple developer account, and implemented test branches that had base functionality of the API uses that the team later pulled code from to implement features later in the project.

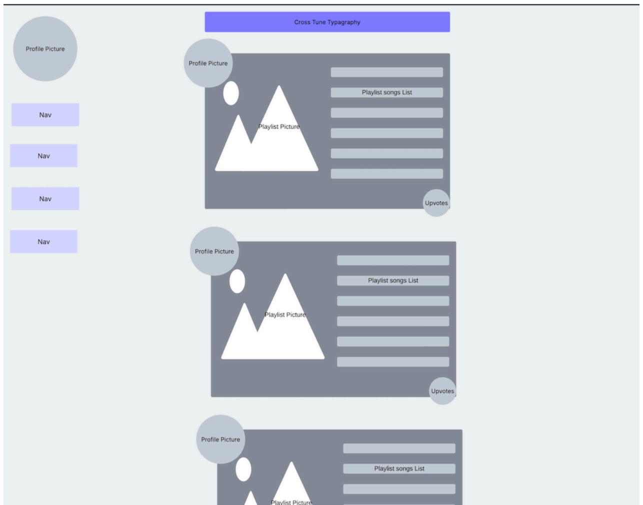
Use Case Diagram

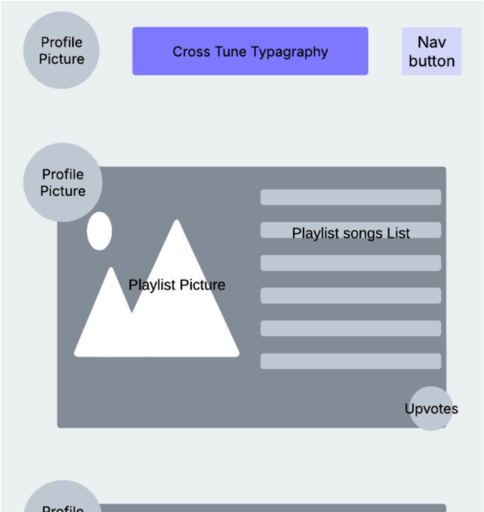
This is our use case diagram from the lab 9 submission. We found it extremely helpful and it was later referenced to reconcile the differences between our architecture diagram, how we experienced the platform, and our initial use case diagram that Renner drew on his Ipad.

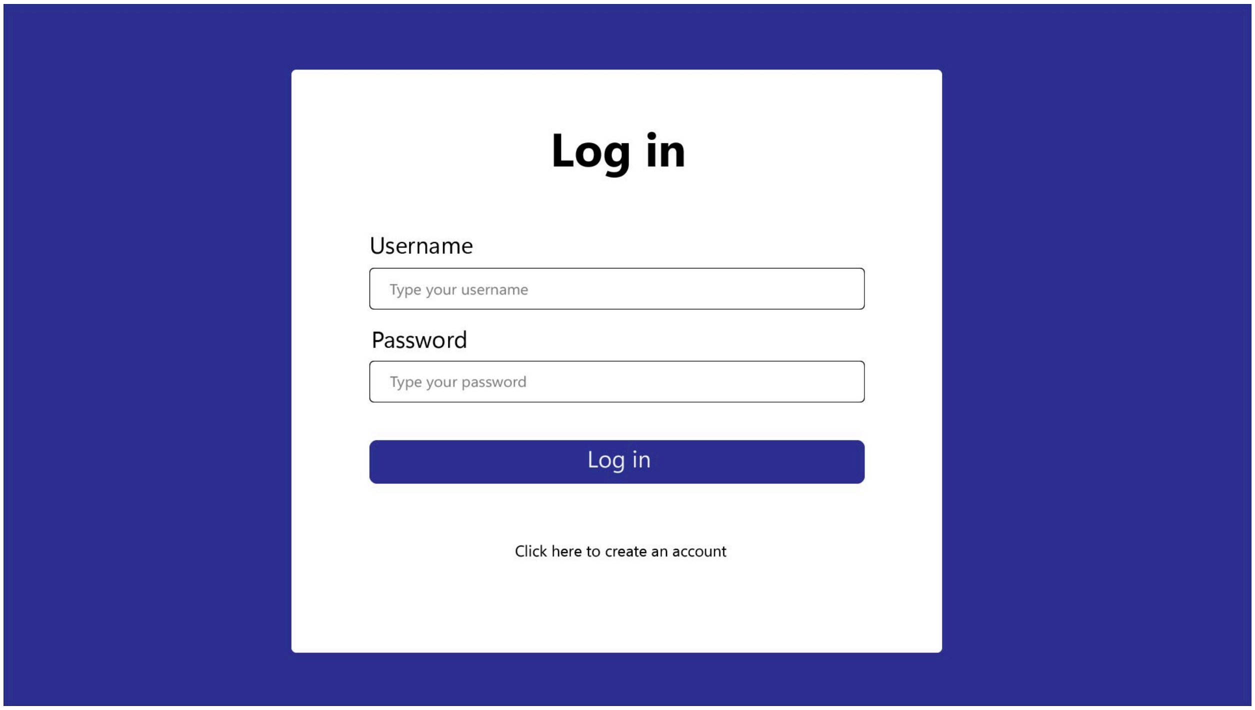


Wireframes

In our wireframe submissions from lab 9 were not done well. After we got our grades back we did high fidelity wireframes of other pages without functionality of our login page (same as registration), settings page, and we added our high fidelity wireframes for the profile and home page that we already had done too.







Log in

Username

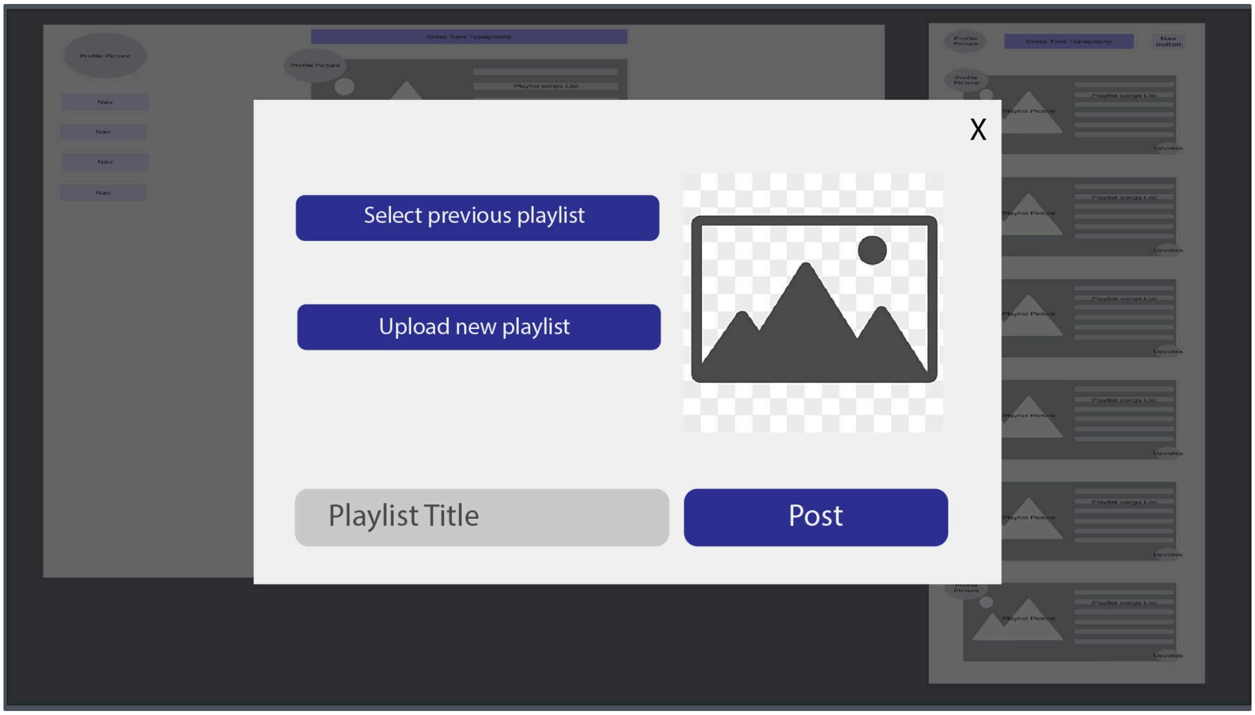
Type your username

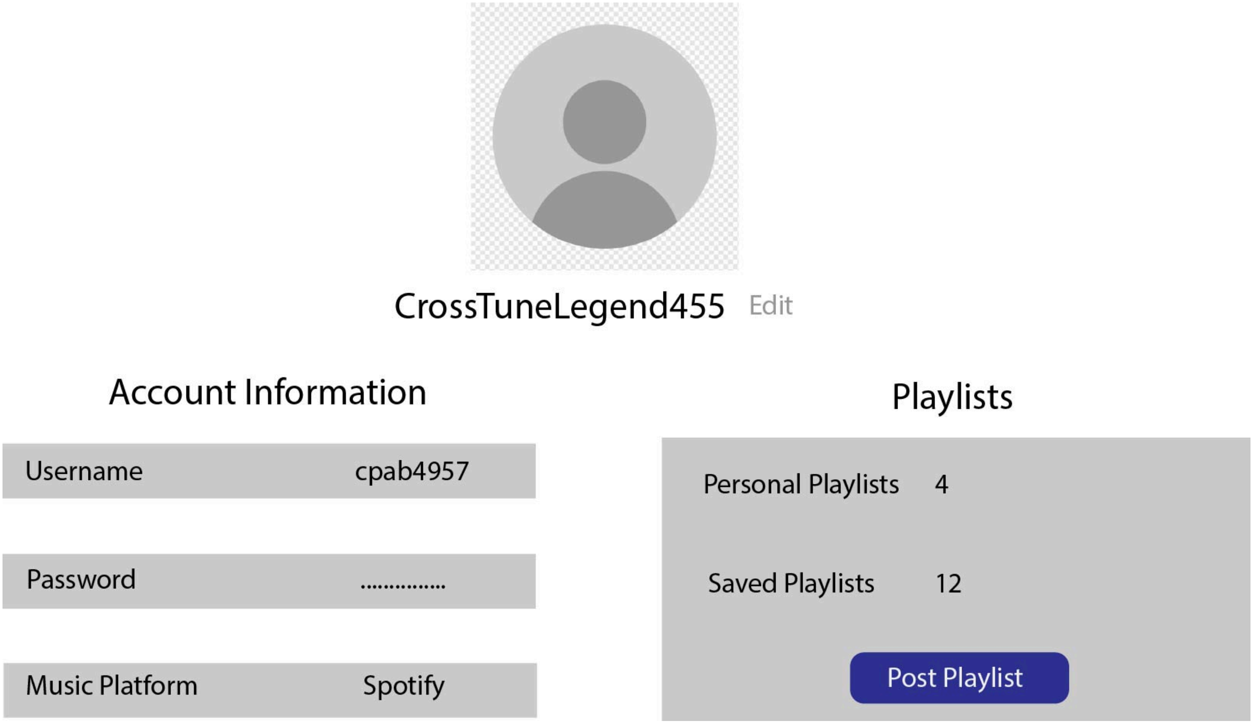
Password

Type your password

Log in

Click here to create an account





CrossTuneLegend455 Edit

Account Information

Username	cpab4957
Password	....
Music Platform	Spotify

Playlists

Personal Playlists	4
Saved Playlists	12

Test results

Feature 1: Convert Spotify Playlist to Apple Music

Test Case: Successful Conversion

- **Action**: User logs in, goes to settings and connects to Apple Music, clicks "CrossTune"
- **Observation**:
  - The settings page is very clear and easy to access and navigate
  - The user went straight to the correct authentication for Apple Music
  - The user logged in through apple and went to a post and clicked CrossTune
  - The user expects an Apple Music playlist in his account as a result of clicking
- **Result**: Conversion completes. The Apple Music playlist shows up in their account with the proper songs.
- **Change Needed**: None.

Test Case: Not logged in through Apple Music or Spotify

- **Action**: User logs in, clicks "CrossTune" on a post, connects to Apple Music, clicks "CrossTune" again.
- **Observation**:
  - The crosstune button was easily in sight and seemed intuitive to go through the process to convert
  - There was a proper message regarding the lack of authentication and redirected the user to the settings page
  - The user logged in through apple and returned to the post and clicked CrossTune
  - The user expects an Apple Music playlist in his account as a result of clicking
- **Result**: Conversion completes. The Apple Music playlist shows up in their account with the proper songs.

Change Needed: None.

Feature 2: Post a Playlist

Test Case: Successful Post

- **Action**: The User logs in, opens the post modal, enters title/caption/link, clicks “Post.”
- **Observation**:
    - He fills all fields quickly.
    - Waits for the post to appear without reloading.
    - The process doesn’t seem difficult at all and is all very intuitive
- **Result**: Playlist appears in the feed properly.
- **Change Needed**: Consider fade-in animation to highlight the new post.

Test Case: Missing Required Field

- **Action**: Leaves title blank, tries to post.
- **Observation**:
    - He clicks "Post" too quickly.
    - Confused when the button does nothing.
- **Result**: Error message under title field.
- **Change Made**: Mention that a Title and Caption are required

Feature 3: Change Username

Test Case: Change Username

- **Action**: The user navigates to Settings, updates username to “musicLover,” saves.
- **Observation**:
    - It seems very intuitive that settings is in the dropdown of their profile picture
    - It’s rejected due because there was no password filled in
    - Retries by entering their password
- **Result**: Username updated and visible on all pages.
- **Change Needed**: Make it more clear you need to enter your password to change your username

Test Case: Reject Duplicate Username

- **Action**: Attempts changing to “renner” (an existing user)
- **Observation**:
  - Looks like it updates but maintains the same username
  - The user seems a little confused
  - No error message
- **Result**: Username is not changed
- **Change Made**: Plan to add “Username Taken” with potential suggestions based on their input.

Deployment

https://cscs3308-spring-project.onrender.com/