

# Final Report: Language Learning Music Finder

CS 228: Human Computer Interaction

Maya Griffith, Alyn Kirsch, and Drew Jepsen

December 11, 2022

## 1) Problem statement

Language learning can be a time-consuming process, but music can make it enjoyable. Songs can be a helpful tool in picking up another language (Mizener 2008), but music recommended to you by teachers or a search on the internet can be boring to listen to and not your style. Also, if you are learning a new language with a new character set, searching for songs on search engines could be a challenge if unfamiliar with the characters or unable to access the correct keys. But if you were able to find the right song, music can convey much more to a listener and improve retention of what you hear (Lake 2002). Now, what if you were able to generate recommendations or create a playlist in the language you were learning based on your specific music tastes?

Our goal is to create a website where language learners could input their music-streaming account, and based on their previous listens, this new product will generate a playlist of recommendations that the user will enjoy in their selected language(s).

This way, users will be able to find music in languages they are learning that they may like, making it a more enjoyable experience for them while finding some great songs they can listen to regularly by making content easily transportable to your streaming platform. Our user groups consist of people who are trying to learn a language and people who enjoy music and want new songs that would appeal to their specific tastes. These two user groups do have a lot of overlap, and we were able to find many people to interview for our study. As far as this group has researched, there is no specific UI of this function. Our sources cited contain information on how music can be a helpful tool in learning a language, and how music that is enjoyable to the user can result in increased motivation and understanding of the language.

## 2) User Research and Results

To further gather requirements for our project, we decided to conduct user research with 3 people, who are currently learning a language and who listen to music on a streaming platform for at least 30 minutes a day. We choose interviews as our method of research because it was the most easily accessible option for us because of time constraints and lack of user tasks available and allowed us to understand potential users on a deeper level. We tried to ask open-ended questions to elicit more insight from potential users. Some examples of questions we asked them are:

- “Have you learned a language in the past or are you currently learning a language?”
- “What strategies have you employed in learning your languages?”
- “Do you have trouble finding new music?”

Some of the results that we summarized from these interviews are shown in figure 1 below.

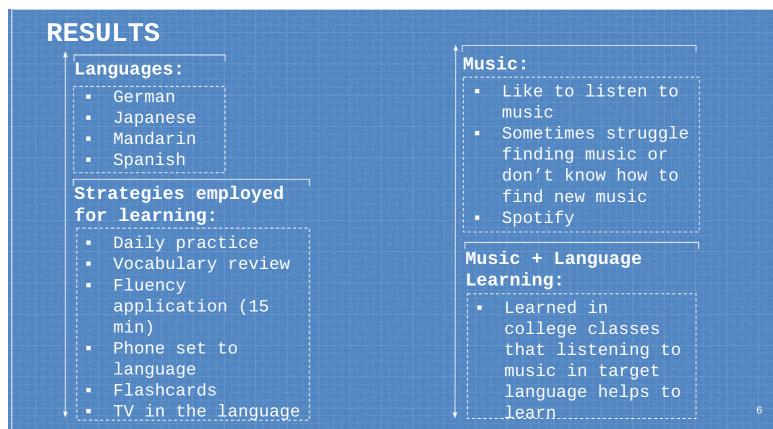


Figure 1) Summarization of User Research Results

Our user research results were successful in many ways: we updated our requirements for our product to match what was said by our users in the interviews conducted. For example, we revisited our user input and data requirements, as interviewees put emphasis on the ease of logging into a potential product through many different streaming platforms, whether it be Spotify, Apple Music, or Pandora.

We were also able to gauge interest and need for the app from potential users, which turned out to be very high. Most users were aware of the connection between language learning and things that interest them (Kao, 2014), and were already implementing it in their everyday

lives through things like setting their phone or TV to their target language. Because of this reasoning, users were very excited to hear about our potential website.

Although our research was successful, we could have improved it in the following ways: ask fewer leading questions, insight more conversations about their experience and their thoughts, include more follow-up questions, and interview more people to increase the variety of answers. If we would have put more emphasis on those points in our research, we would have had more robust results.

### 3) Personas, Scenarios, and User-Flow

As a team of three, we created a total of six personas and scenarios, and together we chose the best three as we believe they encompass our main user groups: students who want to supplement their learning, people who want to their music and people who are too busy to take traditional language classes. Our goal with creating these personas was to better understand use cases for our product and a basic user flow. Our three personas are as follows<sup>1</sup>:

- Mark Smith, 20 male, engineering student, who is too busy with his engineering major to focus on his foreign language studies
- Delia Sanchez, 32 female, hairdresser, who wants to diversify the music heard in her salon
- Justin Case, 39 male, civil engineer, who wants to improve communications with his Spanish speaking clients

These three personas then had realistic scenarios built around them in order to better understand in what areas a solution to our problem would be needed. Simplified scenarios for each respective persona are as follows:

- Mark is unable to register for German 101 but is still determined to learn the basics of German in a less intensive way that fits around his already busy schedule.
- Delia is requested by one of her clientele to play Korean music. Delia confesses that she is unable due to not knowing how to search for music not in the Latin Alphabet<sup>2</sup>.

---

<sup>1</sup> The full personas and their scenarios are located in Appendix, [Section 9.1](#).

<sup>2</sup> This is the alphabet used in Latin-derived languages like English, Spanish, and German

- Justin is bothered by how much time is wasted in construction jobs due to language barriers and wants to learn Spanish in response. But his requirement to constantly travel prevents him from taking a traditional class.

After creating and examining the personas and scenarios, we discovered that our user flow chart<sup>3</sup> needs to be fluid and very easy to navigate. The playlist creation should be customizable but also quick to insight meaningful and repeated usage. Whatever the solution should be, it should have the ability to work for those who have hectic and busy lifestyles. A preferable feature would be continued functionality while offline as a user may use this while traveling on a plane or in low-service areas.

#### 4) Iterative Design Process

When exploring possible solutions for this problem, one will find that the ones that already exist are lacking in at least one of three major areas: specificity, quantity, or discoverability. For example, learning songs through word of mouth may lack all three of the aforementioned. A person who shares music with their friend may be on an entirely different level of understanding the language or the song they recommend may not be appealing/enjoyable to the target user. Finding new music through word of mouth may be inconsistent also as the recommender only may know a few songs, limiting the quantity of music to practice with and the user may not know from who or where they might discover a new song.

In the creation stage, we explored alternative solutions<sup>4</sup> that we wanted to either integrate the song finder with or have it coexist with preexisting software. We also explored solutions separate from the streaming platform. From all of this, our group created two major groups of designs: revisions of existing software and standalone websites that connected to the streaming platform.

Existing software integration mainly focused on a revision of Spotify's user interface. We had envisioned a version of Spotify where you could easily swap between languages on the main UI in order to test the user's comprehension during app navigation. Using Spotify's user data, the playlists created would be able to be tailored to a specific user's preferred type of music, improving their enjoyability of the playlist and in turn improving their learning experience. We

---

<sup>3</sup> The final user flow chart is located in Appendix, [Section 9.2](#)

<sup>4</sup> Alternate Design Wirerames are located in Appendix, [Section 9.3](#)

also explored the idea of using Spotify's song recommendation algorithms to generate different playlists for the user in their targeted language based on song difficulty. This difficulty would be determined using Spotify's classifications<sup>5</sup> of songs' tempo, length, and energy. For example, a beginner Spanish playlist would have songs with slower tempos and fewer words per minute and not high-energy, high-speed hip-hop songs.

Creating a standalone website would offer a variety of benefits in comparison to a user interface revision including, access to multiple streaming services, controlled user flow, and shareability. Having the song recommender disconnected for a specific streaming platform would allow a wider range of users to be able to use the service. The user would be able to log in to the site with their preferred platform and on their preferred device and be able to save the generated playlists to that account as well. A website can be made simple and have artificial constraints that lead the user down an easy user path. It would have the user log in, select their target language and what type of music they like, and then be shown suggested playlists and songs. This control over the user experience would prevent errors and promote a good user experience. Finally, the website being its own thing would allow for it to be easily shared. If it was to be integrated into a platform like Spotify it may be lost in the clutter of features that modern services have. Instead, a user would just need to send the link to the website.

<sup>6</sup>Once we had considered the alternative designs, as a group we decided on developing a website that could use a user's data from any streaming platform they chose, in combination with any language or languages that they wanted to learn, and create suggested playlists and songs separated into three levels of difficulty that the user could then save to whichever streaming platform account they were currently logged in with. Additionally, any playlist that is created on this site can be further customized by the user to suit their specific needs. We found it critical that the song recommender is seen as its own product, if we were to revise an already existing platform there is a high likelihood that it would be lost in the clutter. Also having it be a website defines its purpose as a learning tool and increases its ease of discoverability.

---

<sup>5</sup> Spotify's API has many numeric values that express aspects of songs. For example, Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.

<sup>6</sup> The Final Wireframe Design is located in Appendix, [Section 9.3](#)

Any of our alternative designs are improvements over existing designs. The implementation of modern song-searching algorithms allows for a generated playlist to hit those three crucial design requirements: specificity, quantity, and discoverability.

## 5) Evaluation and Improvement

After creating several different wireframes that followed the specifications of our user research and concluding on a final design as referenced in the appendix<sup>7</sup>, we conducted both individual and group evaluations of our final design and identified areas that needed improvement and/or violated the 10 usability heuristics (Nielsen 1990). By looking at these design heuristics and comparing them to the solution that we had, we were able to find important errors in our design that we would not have caught otherwise. Through design-review was needed in order to make our proposed design the best it could be. After identifying these weaknesses, we were able to fix these usability problems in our final implementation of the website.

Some examples of usability heuristics that were violated in our final design were the visibility of system status. Our final wireframe did not have titles on the pages, which would have led to confusion for users about where they are in the user flow. We also improved the visibility of system status by highlighting the certain page that the user is on on the navigation bar in a visible color like green, so they have something else to reference to determine where they are in the user flow. We also improved the match between the system and the real world, by displaying the languages in their natural alphabets on the selection page and in the playlists page, while still having their English translation, so users can eventually start to learn these words, instead of them written in Roman character translations that are not common. Another improvement we made to the final implementation from reviewing our final design is making the error pages more descriptive, by telling you what your specific error was so the user can fix it. For example, when a user gets the username or password wrong, the error description will match what the user inputted wrong. In the current implementation of our final wireframe, it was very hard to backtrack and understand where the user made a mistake. Also in order to combat this, we also added a help and documentation page in order to help users determine how to flow through the program and recover from errors quickly.

---

<sup>7</sup> The final design is located in Appendix, [Section 9.3](#)

Through evaluation, before we started building our final product, we were able to save time and effort by catching these usability mistakes that would eventually add up and make our website unfriendly to users.

## 6) Solution/Final Outcome

Our team's proposed solution is a website separate from any specific music streaming service that a user can log into using any account from any streaming service, select a preferred language or languages, and then will be giving playlists in their target language based on their user data that they can either directly add to their logged in account or directly edit. Our prototype was coded using mainly HTML, Javascript, and CSS and is all locally hosted using python's flask API. We decided to use these languages in order to capture most of the core functionality of the website and to also make it mobile-friendly. The prototype passes the provided accessibility<sup>8</sup> test.

<sup>9</sup>The implementation roughly follows the final wireframe<sup>10</sup> while also correcting critical issues discovered in the evaluation stage of development. The main user flow has been boiled down to its simplest features in order to maintain our goal of having an easy-to-understand user experience through constricting user options. The average user experience is as follows:

- User logs in using their preferred streaming service account
- User selects target language(s)
- User is brought to a loading screen as the system generates their playlists
- User is shown five playlists of varying difficulty and five suggested songs
- User can further inspect a specific playlist where they can add, remove, like and rearrange songs
- Alternatively the user can navigate to the “add playlist” button to create a playlist that can be a mix of all suggested songs from every playlist.
- Finally the user can save any desired songs or playlists to the account logged in

To assist with this main usage, there are many features implemented with accessibility in mind. Notably at every stage, a help option is displayed that links the user to the help page that houses instructions and general information about the entire site. Also, we purposely made every button

---

<sup>8</sup> Provided Accessibility tester: <https://www.webaccessibility.com/>

<sup>9</sup> Screenshots of the final prototype are located in Appendix, [Section 9.4](#)

<sup>10</sup> The Final Wireframe Design is located in Appendix, [Section 9.3](#)

and icon large to allow the user to find them easily. In terms of feedback, after clicking on any link, a red border appears to indicate action. We have an error page to catch any unexpected actions that explain to the user that they have hit an error along with a more in-depth error message and redirect options to previous pages.

The were a few limitations that we ran into, most of which stemmed from the format of the class. The most noticeable limitation is the lack of a real back end in the implementation. All the selected songs are not pulled from the user's data but instead are chosen by us. Due mainly to a time limitation, in our prototype, we only have three suggested playlist pages, one for Spanish, one for Hindi, and one for the combination of both. Additionally, the lack of the API would make an actual working create playlist option very time-consuming. We have no access to an API's database and querying so we would have had to hard code in search results, so our search feature only brings up one song: Dura by Daddy Yankee. A final result of no backend is the lack of account validation and connection. We only have one "work" account<sup>11</sup> hard coded in and everything else brings you to the error screen. Consequently, a playlist cannot actually be saved to a real account. Also, everything coded is framed in terms of Spotify as the main streaming service being used. It has the largest user base of the main music platforms and if we were to implement a backend, it has the most developer-friendly API.

Our team believes that our prototype even with its limitations successfully addresses our original problem: it is difficult to find relevant and enjoyable music to help learn and practice foreign languages. We aimed to create a product that was simple to use, offering a plethora of personalized playlists that helped anyone practice foreign language skills no matter the level. The site is simple to use as it asks a few questions for the user to input and shows them personalized playlists in as little as three pages. The suggested playlist<sup>12</sup> has five unique playlists for the user, individual songs, and the option for the user to make their own.

## 7) Conclusion

In conclusion, our solution provides a new, usable UI for students of all foreign languages to find music that they would like in a selected language. We believe this UI fills a gap in the market that can allow users to have personalized playlists and songs for their educational needs.

---

<sup>11</sup> Image of the Login Screen located in Appendix, [Section 8.4](#)

<sup>12</sup> Image of the Suggested Playlists Pages for Spanish is located in Appendix, [Section 8.4](#)

We first started designing this product by collecting requirements, conducting user research through interviews, designing several UI alternatives and converging on a single final design, and then iterating through evaluation and feedback to finally come up with our final coded UI design. Through this process, we were able to delve deep into how and why our user would interact with our foreign language music finder, and use those insights to repetitively make our final product better. Through this process of research into the user, we were able to catch several usability problems and fix them so our final website was as user-friendly as possible. Music and songs should be a motivating factor in the learning process, enjoyable music leads to a want to learn (Israel 2013).

The final implementation of our website met our standards of what we determined to be the core components of our product, but could still be improved upon. For example, more emphasis can be added to the education aspect of our website, and we could have features to help the user learn the song or help improve their vocabulary by displaying the lyrics of a recommended song and testing them on it, as reading lyrics can be very important in language learning through music (Boothe 2015). Also, another implementation that may be of use is allowing the user to sign in to several of their streaming accounts to combine recommendations and make the songs displayed by our website more accurate. But even without these features, our high-level attention to the user's needs through our in-depth research allowed us to create a final product that could be of great use to our user group and solve the original problem of searching for songs in different languages.

## 8) References

Boothe, D., & West, J. (2015, July). English language learning through music and song lyrics—The performance of a lifetime. In Proceedings of the future of education conference (pp. 248-254).

Israel, Hilda F. "Language learning enhanced by music and song." Literacy information and computer education journal (LICEJ) 2.1 (2013): 1360-1365.

Jakob Nielsen and Rolf Molich. (1990). Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90). Association for Computing Machinery, New York, NY, USA, 249–256. <https://doi.org/10.1145/97243.97281>

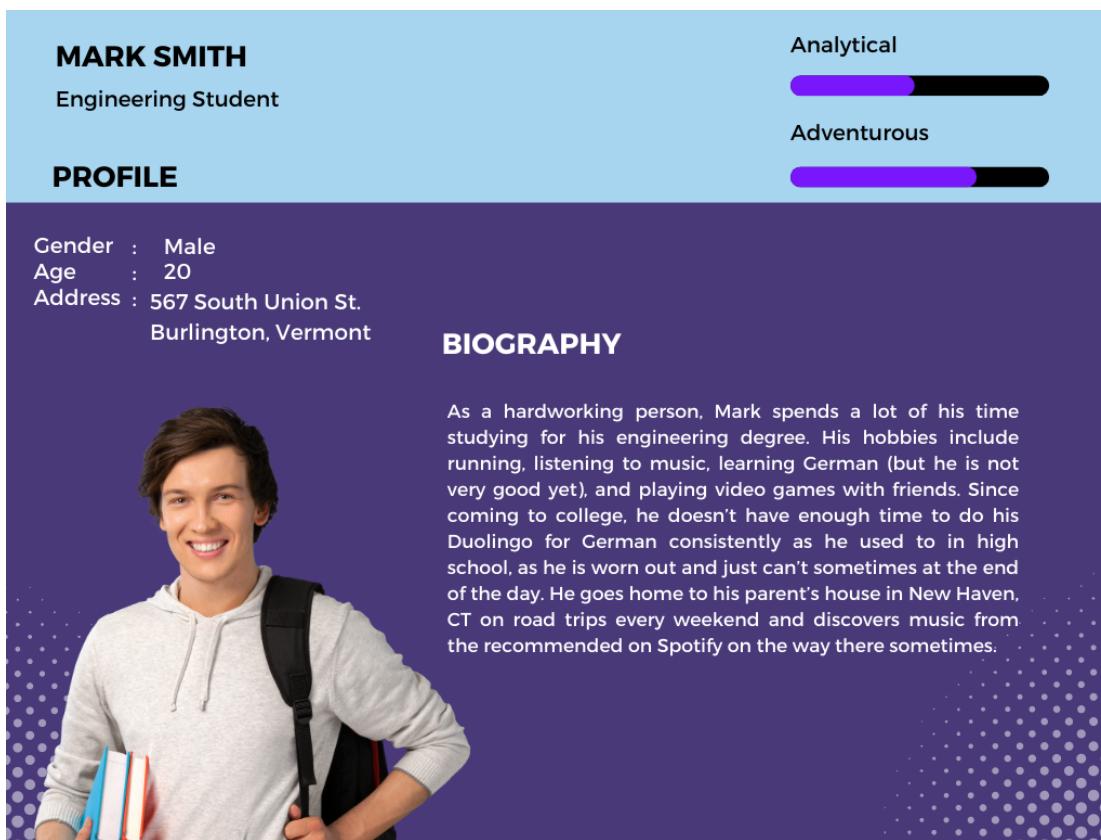
Kao, T.-A., & Oxford, R. L. (2014). Learning language through music: A strategy for building inspiration and motivation. System, 43, 114–120. doi:10.1016/j.system.2014.01.003

Lake, R. (2002). Enhancing acquisition through music. Journal of the Imagination in Language Learning and Teaching, 7, 98-106.

Mizener, C. P. (2008). Enhancing Language Skills Through Music. General Music Today, 21(2), 11–17. <https://doi.org/10.1177/1048371308316414>

## 9) Appendix

### 9.1) Personas + Scenarios



#### Scenario 1:

As soon as Mark tries to register for classes this year: German 101 fills up. Oh no! The only time he could fit it into his schedule, the class fills up before he can even add it. Disappointed, that night he looks for other ways he can incorporate learning German without a class and not that intensely because he has so much homework that he can't stray away from for too long.

Stumbling across several articles, he learns that catchy music in another language could be a great way to learn without that much effort, and tries to search up “popular German songs”, but only finds things he could find are definitely not what he would listen to while studying or driving, the only times he turns on his speaker. Finally, he stumbles across the website language learning music finder.

**DELIA SANCHEZ**

Hairdresser

**PROFILE**

Gender : Female  
Age : 32  
Address : 216 14th Ave, New York, NY



Creative

Impatient

**BIOGRAPHY**

As a hairdresser, Delia 28, loves music and playing her new finds in her salon for her clients. She speaks Spanish and English and lives with her boyfriend. Her hobbies include watching movies, skating, science, and also listening to music, especially in different languages, as you can see from her playlist, even if it is in a language she doesn't know. She uses Spotify to play music all the time, coming to and from work in NYC, and often even playing it out loud on the salon speaker.

19	 Qué Más Pues?	J Balvin, Maria Becerra
▶	 Te Quiero	Maluma
21	 Yet To Come	BTS
22	 On ne s'aimera plus jamais	Larusso
23	 Aaminin	Tiara Shaye
24	 Un Ratito	Bad Bunny

### Scenario 2:

Working in a hair salon making small talk, her clients often recommend her songs, as she controls the speaker in the hair salon. One day her Korean client confessed to her as she sat down for a quick haircut: “I love your music taste! I just wish you would play more Korean songs, I feel like they would fit in so nicely here.”

“Of course”, Cara replied. She wanted to make all her clients happy. And she was totally down to expanding her music palette, as she is an adventurous person. “I would love to - it’s just that I don’t have a Korean keyboard on my phone, it is tough to search up songs, especially when you don’t know what any of the characters mean.”

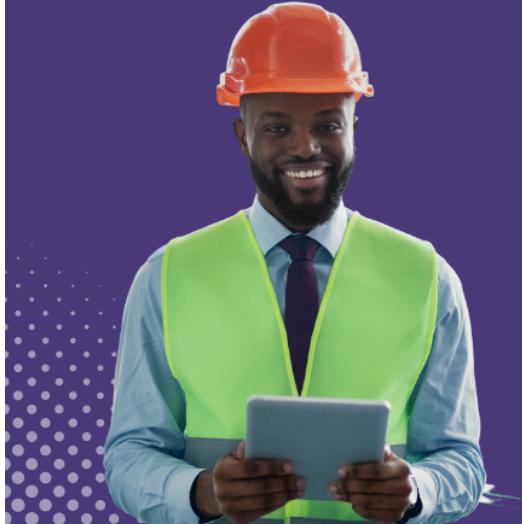
“I could give you a couple of recommendations of course”, she replied. “But to do it on your own - I see how that could be really hard. Especially when the first “popular Korean songs” google search doesn’t bring up desirable results. Good thing I know of a website that can give you recommendations based on your music streaming platform listens!”

**Persona 3:**

**JUSTIN CASE**  
Traveling Civil Engineer

**PROFILE**

Gender : Male  
Age : 39  
Address : 29 Harriot St, Boulder CO



**BIOGRAPHY**

Justin Case is a traveling civil engineer who works in North America. He lives in Boulder Colorado with his wife and two kids. He is the primary income earner for his family. Justin's job requires him to interact with many Spanish communities in Texas and California. Justin wants to strengthen his Spanish comprehension in order to interact better with his clients. A big problem preventing Justin from taking more traditional classes in Spanish is his requirement to travel around the continent constantly for his job.

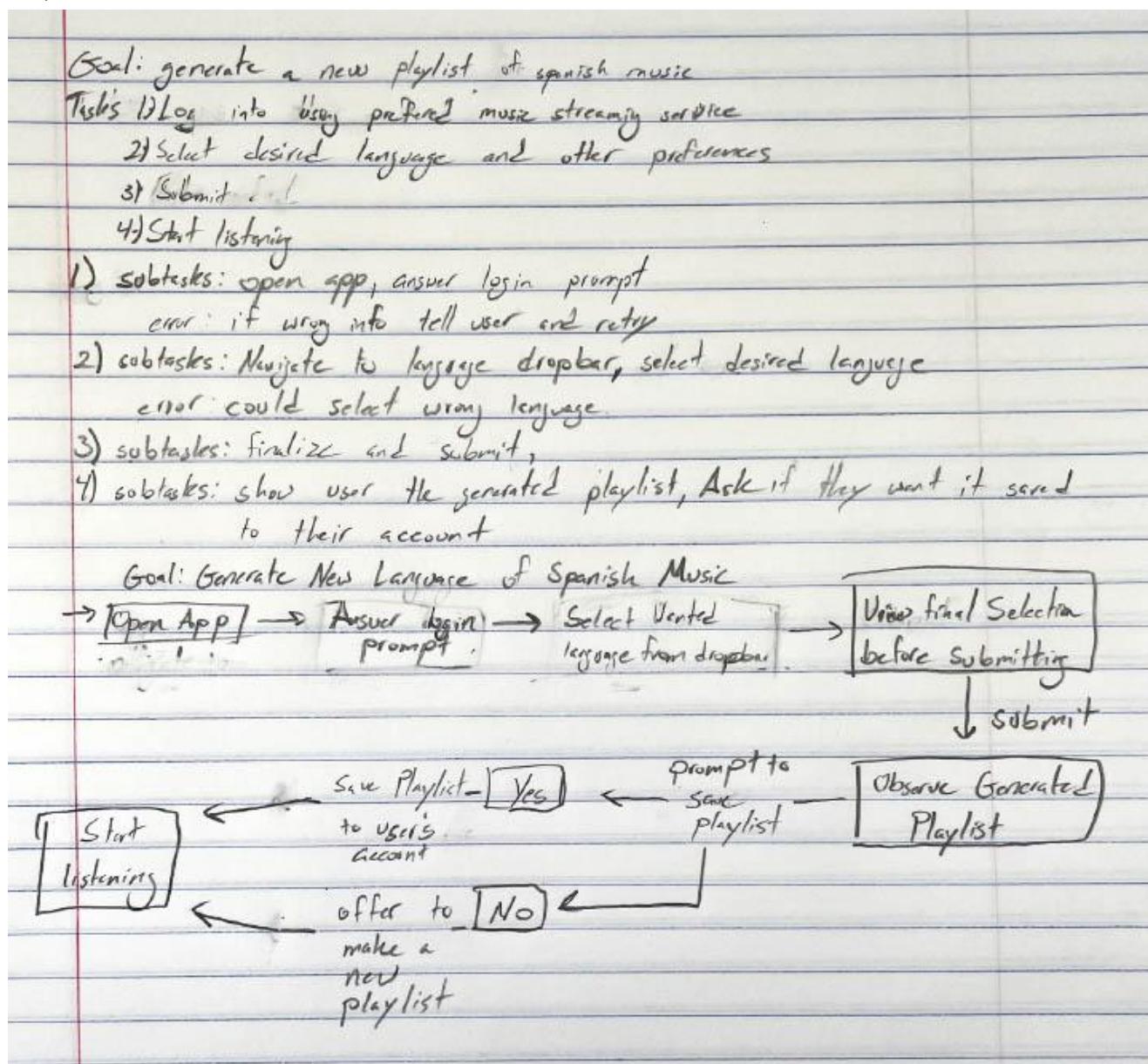
**Scenario 3:**

Justin is working on a site in Southern California. He is working with the local construction workers, many of whose first language is Spanish. This disconnect in communication leads to slowed production and time wasted on calcification and translation.

On his own time, Justin has been trying to improve his Spanish but due to his hectic lifestyle, he does not have enough time to commit to any traditional classes. While driving to a sight, Justin is listening to the radio when Spanish music comes on. This gives him the idea that maybe music can help him practice while on the move.

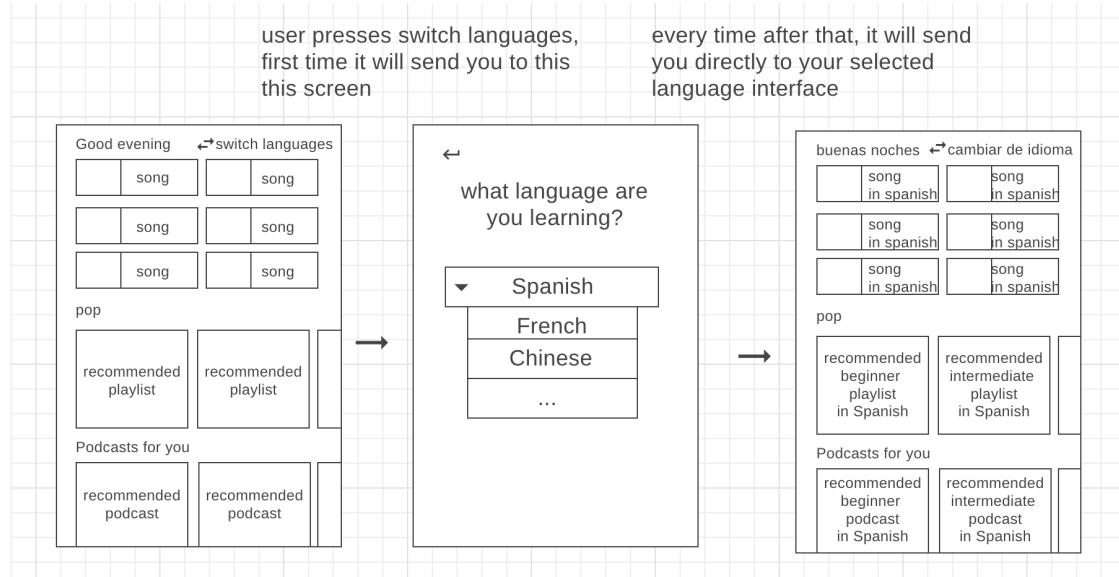
At home, Justin tries to create a playlist of Spanish music that would be enjoyable to listen to and help him become familiarized with spoken Spanish. But in his search, the music is either not enjoyable for him, too fast, and difficult to understand any of the words, or is not even in Spanish.

## 9.2) User Flow

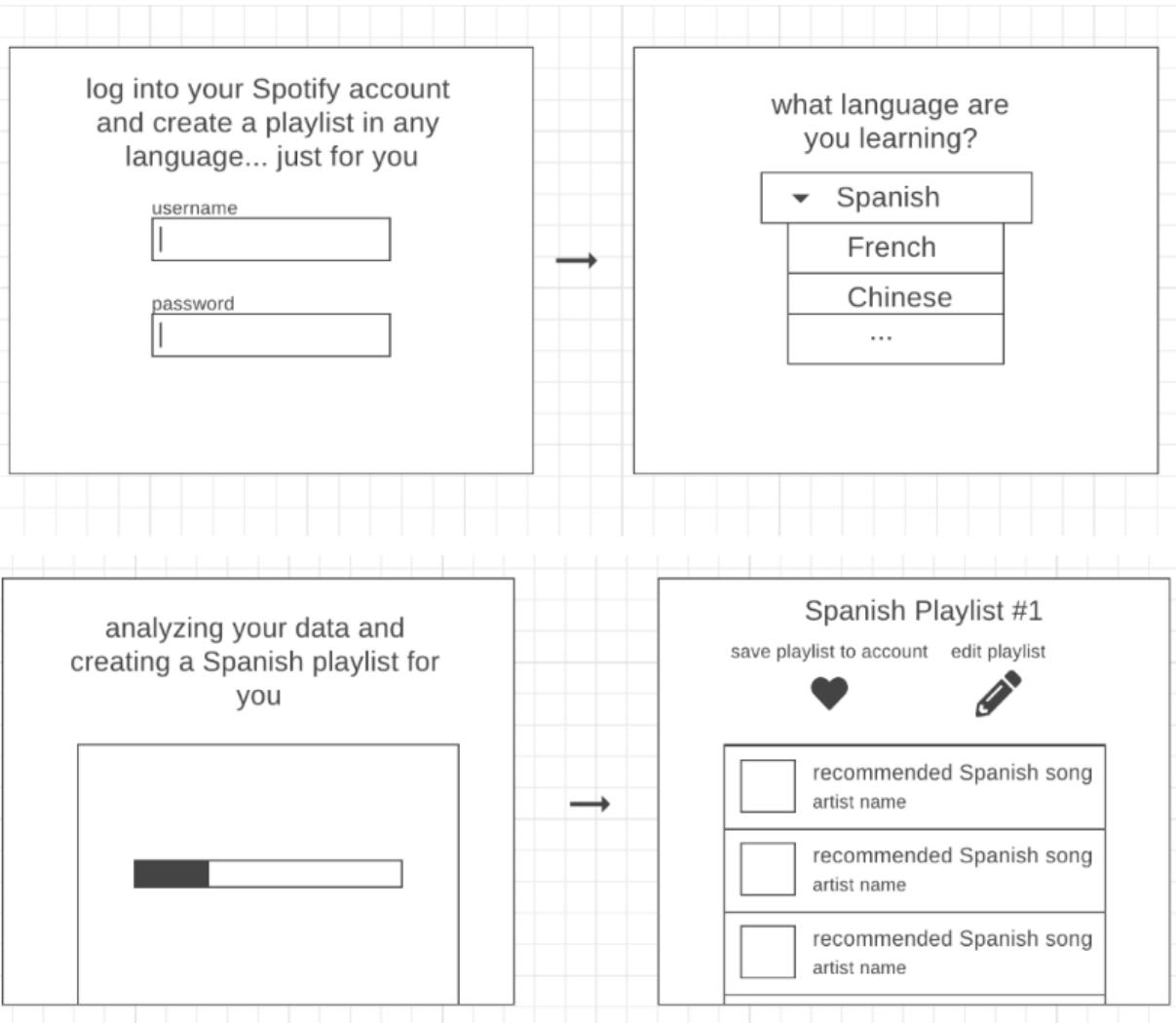


### 9.3) Wireframes

1)



2)



3)

1)

2)

- Search for specific languages
- can scroll through entire list
- can select multiple languages

3)

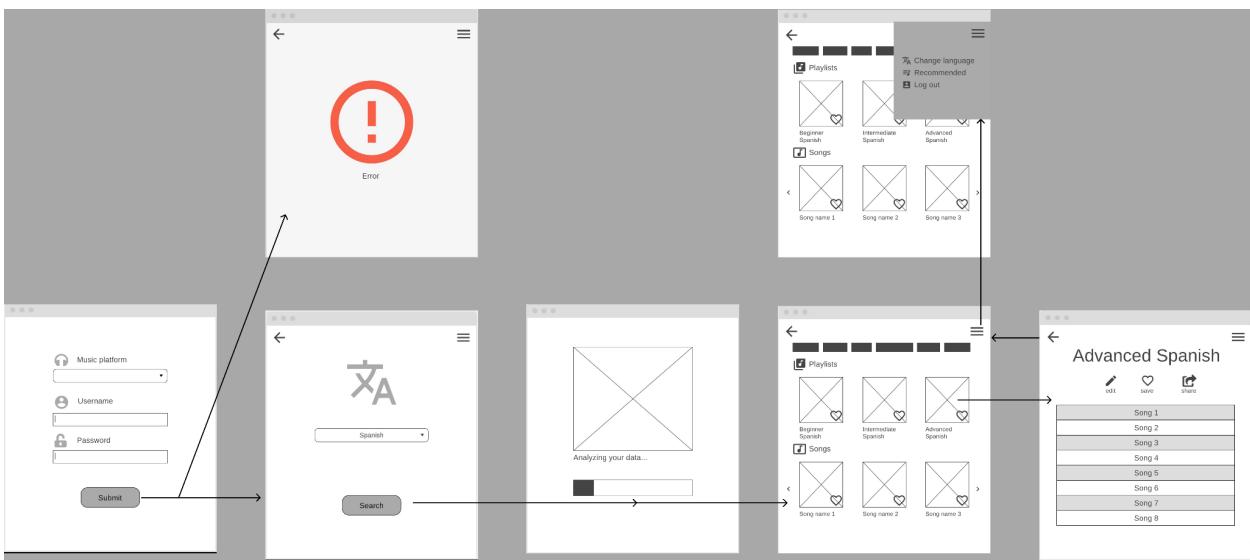
4)

- Hitting the  lets you inspect each playlist
- Hitting  adds the playlist to your Spotify account

5)

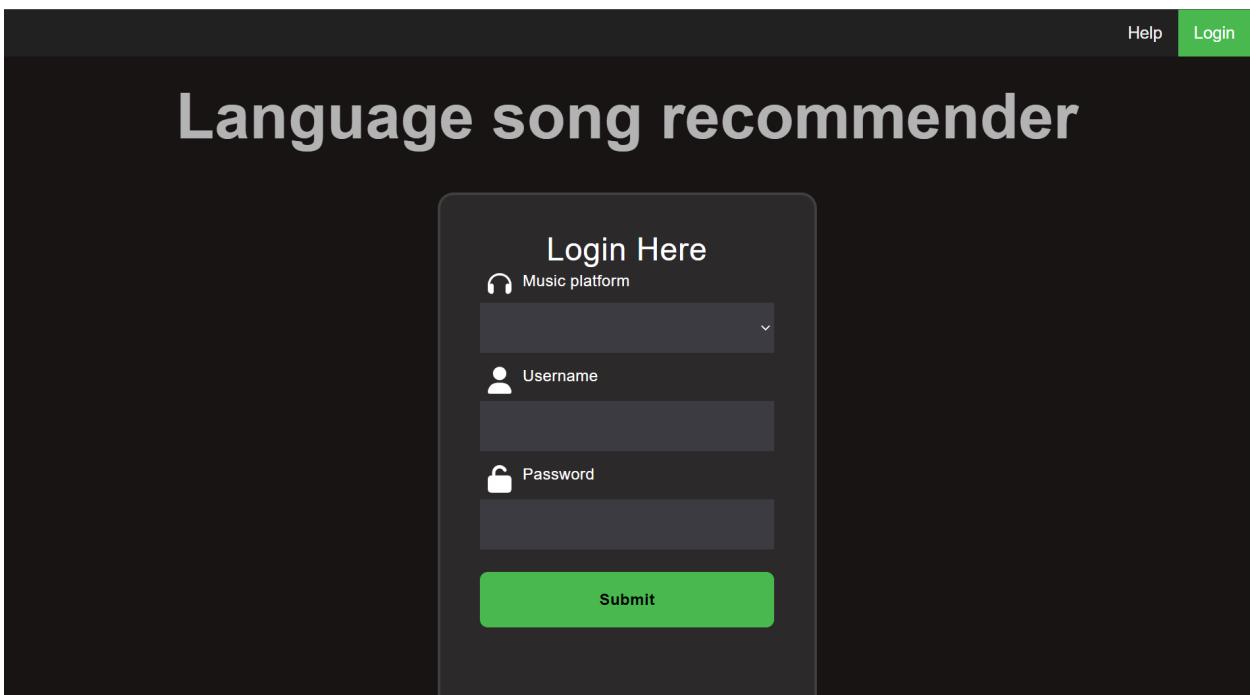
- Hitting  plays a portion of that song
- Hitting  removes that song from the playlist

## Final Wireframe:

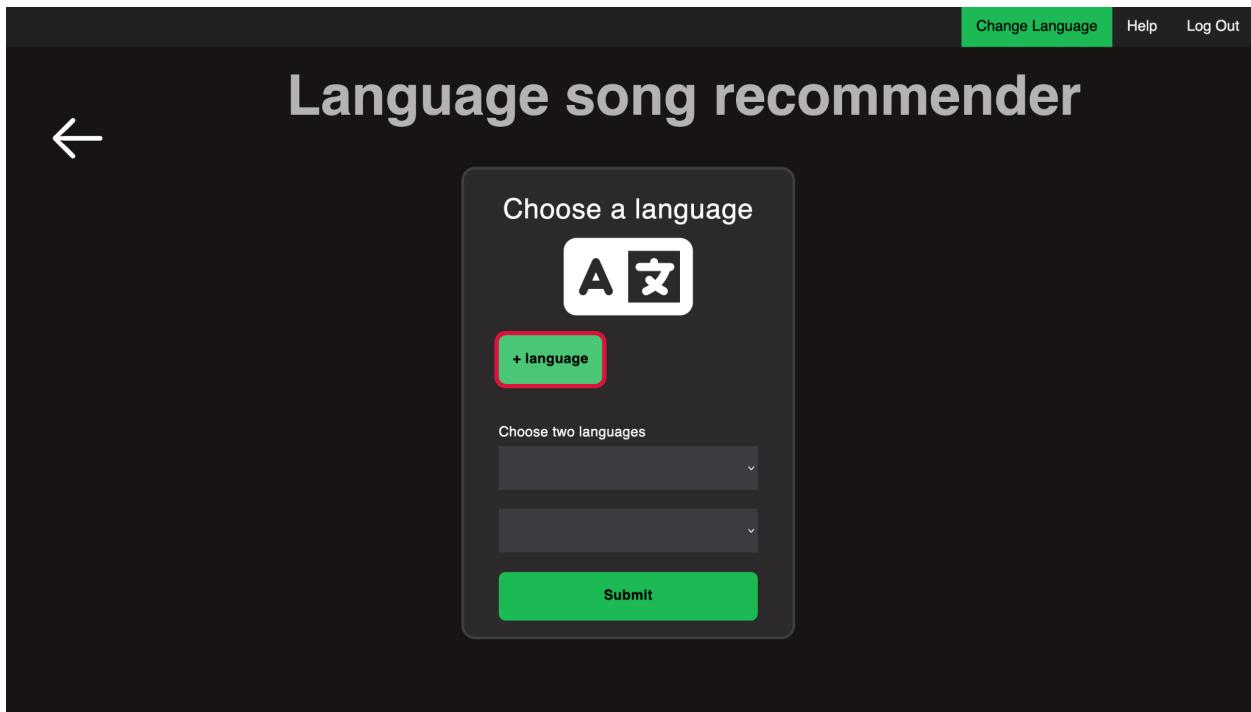


## 9.4) Final Prototype Images

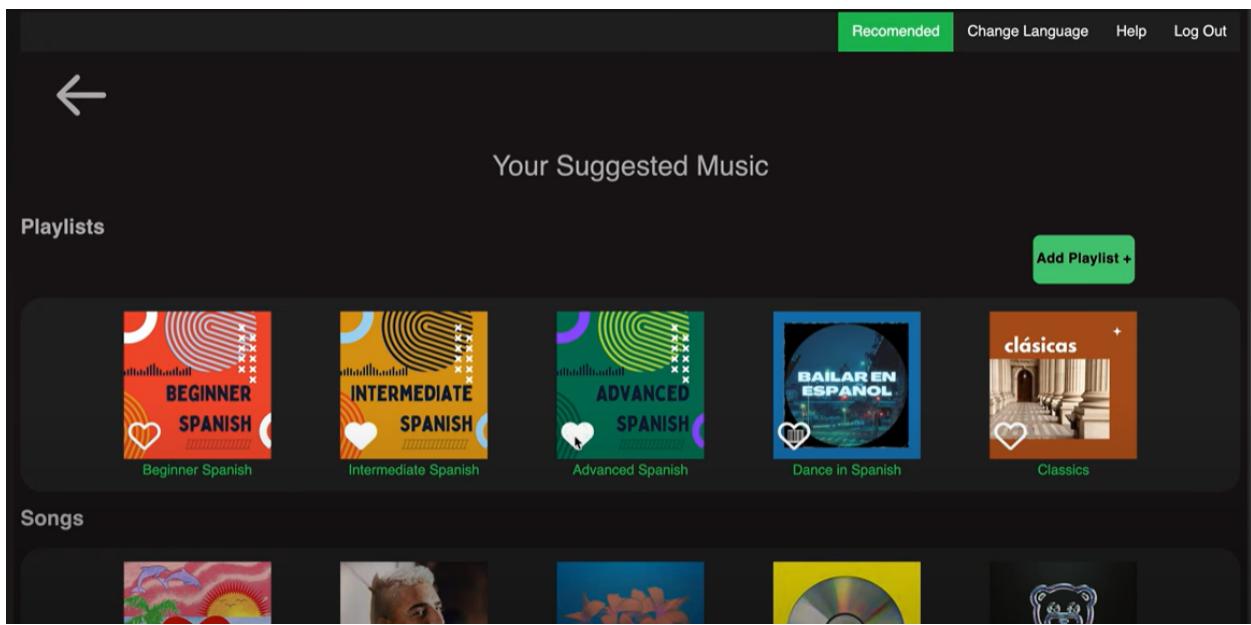
### Login Page:



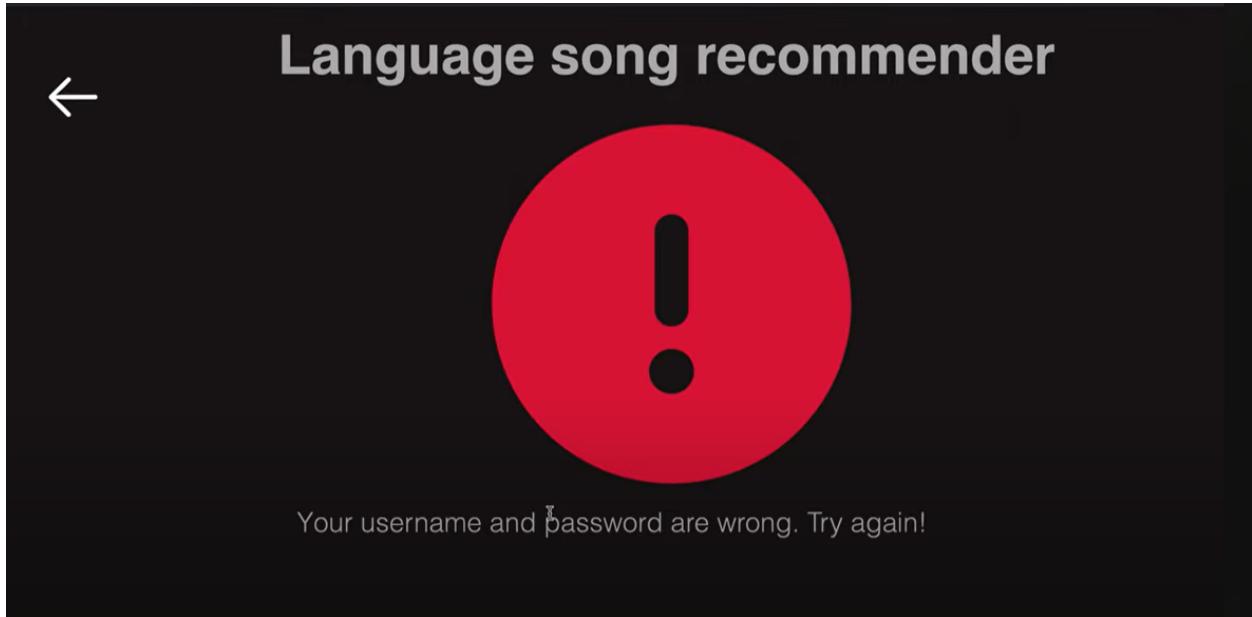
Language Selection Page:



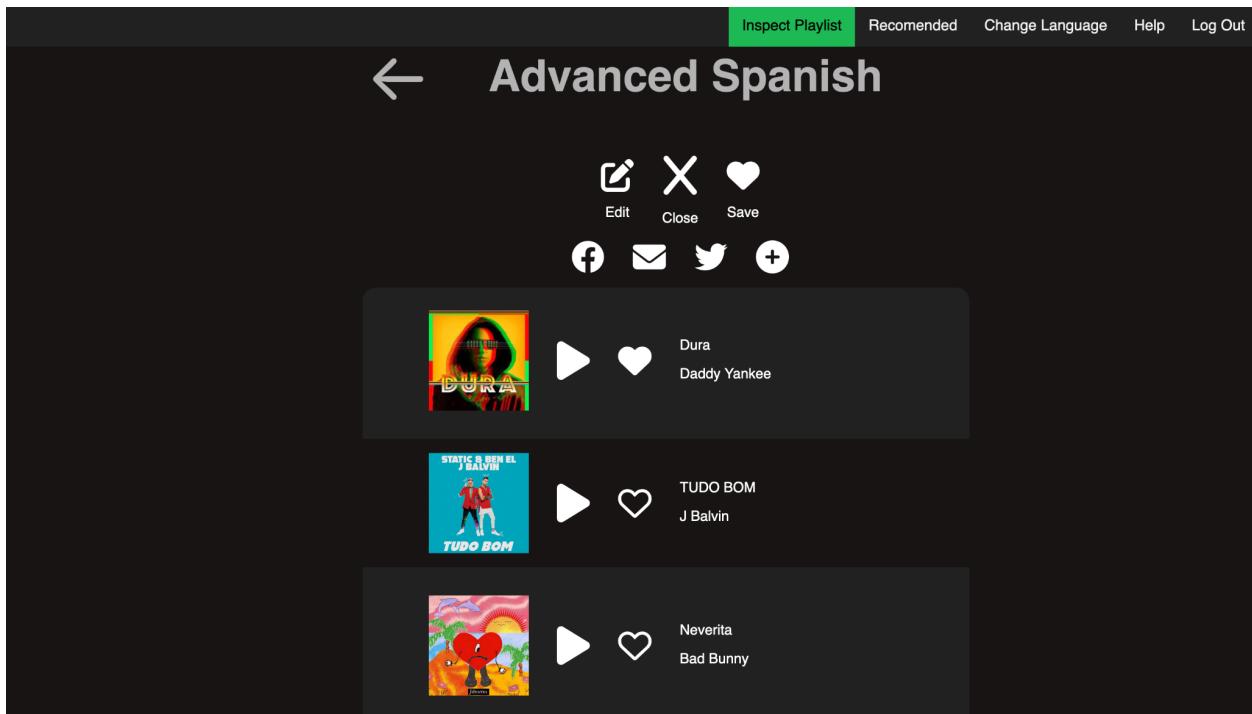
Suggested Playlists Page:



Error Page:



Inspect Playlist Page:



The image shows a playlist page titled "Advanced Spanish". The navigation bar includes "Inspect Playlist" (highlighted in green), "Recomended", "Change Language", "Help", and "Log Out". The main content displays three songs in a dark-themed interface:

- Dura** by Daddy Yankee
- TUDO BOM** by J Balvin
- Neverita** by Bad Bunny

Each song entry includes a play button, a heart icon, and the song title and artist name. There are also social sharing icons (Facebook, Email, Twitter) and a plus sign icon at the top right of the list.

## Help Page:

This site is aimed to help create a usable playlist for those looking to learn new languages.

Language learning can be a time-consuming process and hard, but music can make it enjoyable. Songs can be a helpful tool in picking up another language, but music recommended to you by teachers or the search bar on Spotify can be boring to listen to and not your style. Now, what if you were able to generate recommendations or create a playlist in the language you were learning based on your specific music tastes? Welcome to Music Language Recomender, where language learners could input their music streaming platform, and based on their previous listens, this new website will generate a playlist of recommendations in the user's most listened to genre and in their selected language. This way, users will be able to find music in what they are learning that they may like, making it a more enjoyable experience for them, while finding some great songs they would like. Music can be a helpful tool in learning a language, and music that is enjoyable to the user can result in increased motivation and understanding of the language.

## Instructions

Log in to your preferred music streaming platform, choose a language (or two!), and start listening to your own personalized recommendations. Choose from curated playlists and songs for you, and create your own playlists with recommendations from the site. You also have the ability to listen to a preview of the songs, "like" it in your preferred streaming platform, and remove it from our "made for you" playlists. Happy listening!

## About Us

All code authored by Alyn Kirsch, Maya Griffith and Drew Jepsen

## Create new Playlist:

Add song:

Dura Search

Edit Share Save

Dura  
Daddy Yankee