Terptune

Overview

Overview describing the motivation and high-level concept of the app

Popular music streaming apps like Spotify and Apple Music are the modern cornerstone for *most* of your music needs. You can listen, rate, share and curate your own taste in music over time. Where these apps lack is a public forum where users can give more feedback than just a like. More broadly speaking, these apps are missing community. Terptune seeks to address this need by offering a more social approach to listening to music, while maintaining some of the listening capabilities of more traditional platforms.

Terptune users are exposed to thousands of popular songs and encouraged to listen to samples. Users will review and rate songs they like or dislike, and the reviews will accumulate for each song as time goes by. Once you review some songs, Terptune will gauge what you do and do not enjoy and curate some new recommendations for you.

Given that our goal is to blend social media with music, we were greatly inspired by the popular **Explore / For You** model that is used by many social media giants today. The explore page will expose the user to new content, and the for you page will expose them to content tailored to their preferences. Over time, the user will curate a profile full of reviews. The open-sourced nature of the app will also give songs a plethora of feedback that is available if you just click on the song. In other words, you will be able to see what others think, and make your voice heard as well.

Development Process

Discussion of the development process, including when/how tasks were completed; any setbacks encountered; decisions and changes made to the initial plan; and what you learned, both about the development process and the features you used that were not covered in class

This project began back at the beginning of the semester. We agreed collectively that we wanted to make an app that involved music and community, and with that in mind, we looked for inspiration. Letterboxd is a film review website that is a very sophisticated, parallel implementation of our app's general concept. It was a forum for those with a shared interest to be social, give their opinions, and interact deeper with films and other art they may have enjoyed (or not enjoyed). We took this as a starting point for Terptune, our new music reviewing app.

Within the guidelines of the assignment, we brainstormed the basic functionality for the app. These first thoughts are listed in our initial proposal, where we discussed basic features such as liking songs, reviewing songs, rating songs and a UI concept. We set a broad framework that would be easily amenable in the future once we became more familiar with our own intentions for the app. Our first concept included three tabs: Explore, Top Songs and Profile.

We were certain about a couple of things at the beginning. An online database would be used to manage user data, there would be authentication, and there would be absolutely no cached data about songs. All music information displayed would come from an API, but which one was still a question. The group decided the Spotify API would offer the widest range of options, functionality, and features.

Now that we had some logistical decisions out of the way, and an app with three tabs, it was time to get programming. Our first milestone encompassed all that came before a single line of code was written, as well as configuring everyone's environments and git repositories for a sustainable workflow throughout the semester.

Now that we had a UI template, it was time to add some of the core functionality of the app. We began by playing with Spotify API calls in an attempt to populate the explore page with songs. It can't be overstated how little we wanted to rely on cached song data, so the Spotify API would provide everything we needed regarding song data. This is where we had our first real setback.

In essence, the Spotify API requires spotify credentials to make an API call. Can we expect every user of our app to provide us with their credentials to a completely different app? Probably not. This turned our heads to the notion of user authentication, which would be crucial to our functionality, so our focus for milestone 2 became a fully functioning user authentication (register, login, logout) system using Firebase Cloud Firestore. Firebase was the perfect option for us, since it would seamlessly handle user authentication AND manage user data in the Firestore Cloud Database. This process was challenging. As a group, we have very little prior experience using a cloud database service like Firebase, so implementing this user registration system on top of our core app was a laborious process that encompassed our second milestones progress.
Finally, we have a UI template, a functioning user registration system, and a connected cloud database ready for reading and writing. It was now time to attack the Spotify API and bring our app to life with songs, reviews and a profile page. We focused first on the explore page, displaying popular songs as a scrollable list which were derived from a 'Top 100' playlist we found on Spotify. The Spotify API proved to be a challenging API to use. The format of the responses were extremely difficult to parse, and often inconsistent, so we had to do a lot of data cleaning and manipulation so our app could digest the responses and display songs and their associated information. Our explore page now contains a list of 100 songs, with titles, artist names and a detailed view when you click on it.

As a group, at this moment, we took a step back to evaluate the other tab 'Top Songs'. Today in the social media industry, a very popular model for apps that facilitate social interactions AND content use an **Explore/For You** model. For example, twitter and tiktok use the explore page to

gauge what a user enjoys, and then uses a recommendation algorithm to funnel content curated specifically to the user through the for you page. This was our first major shift in thinking, where we would now have a For you page instead of a Top songs page. Luckily, the Spotify API provides a recommendation algorithm where you can use 'seed songs' as input, and get recommended songs as output. This made it clear that we needed to utilize a review system to gather seed songs, so we could curate a for you page full of music based on songs they positively reviewed previously.

The review feature is available when you tap on a song through the explore page. You will see a more detailed view of the song you chose, along with a review form for submitting your thoughts as well as a score 1-5. Upon completion of the review, we connected the database to store your review in a specific collection. Now, we have a fully functioning explore page with a review system. The For you page, visually, is nearly identical to the explore page. However, the content is tailored precisely to what you had positively reviewed, rather than a broad selection that you would see on the explore page. In theory, this recommendation algorithm would only get more and more accurate as the user provides more reviews.

Let it be known that this Youtube series served as a basis for learning Firestore. Episodes 1-17 were followed for connecting the database to our flutter app, although the nature of the example app in the videos is a polar opposite of what we are trying to do. It was very useful and taught us what we needed to get started.
https://www.youtube.com/playlist?list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC

Once our group reached a point where we had a basic explore page, for you page, and user registration, we took some time to polish the UI. First, the registration and sign up pages had many different kinds of errors that had to be dealt with. We needed validators, error prompts, and significant UI improvements that aligned with our decided branding (light purple and grey, modern fonts, minimalistic). Then, we added album covers to our explore/for you pages, and the app really started to come to life. There were colors, moving parts, and most importantly a cloud database and no cached song data that left the app itself very 'light'.

This is the point in the project where everything we decided to do was an improvement or upgrade, besides the Profile page, which was yet to be implemented. We split up our efforts to make an interesting and functional profile page AND improve the experience with the Explore/For You tabs (UI/UX speaking). Well… what is a music app without some music? The Spotify API provides a 30 second preview in their json response! Now, it was a matter of adding a play button next to every single song with a preview available, and allowing the user to actually listen to the music within the app! This is one of our favorite features, and was one of the most fun to implement as a group. It turned out to be one of the most crucial features, because how can you review a song without hearing it? There was a lot of effort put into the play and pause button to ensure that any user input (spamming play buttons, switching tabs mid song, etc) would be appropriately dealt with without causing errors.

The profile page slowly took shape, including a basic UI design and the ability to change your profile picture. Now that we had a review system, it was natural to decide to show a user's reviews on their profile page. This was a surprisingly difficult task, as we had to reconfigure our database collections to work in a relational manner between each other. The profile page now listed all of the reviews a user had left on different songs, really tying the app together.

After milestone 3, we reflected on the progress and our initial goals. In summary, our most significant changes to our initial proposal were the model of the app changing from Explore/Top Songs -> Explore/For You (for better UX), and the decision to use Firebase above all other options. We realized we were missing a mobile dependent feature, so we brought back one of the stretch goals that we ditched earlier in the semester, the **Shake and Rate**. In our app, you can shake your phone, you'll be prompted to review a random song that has started playing. It's a neat feature, but it's a bit mundane (for now).

The app we spent months slowly developing provided a plethora of learning opportunities along the way. Most importantly, we experienced a long process of developing in a group setting, with collaboration, differing ideas and unique opinions. It's worth noting that with this comes the adoption of some type of version control, so we embraced Git and it served us well not only as a tool but as a vehicle for learning the tool itself, which is crucial in industry. On the technical side, it's clear that we had to go an extra mile to familiarize ourselves with a cloud database that was not covered in our class material. Firebase is extremely well-documented and proved to be a great decision, in retrospect. Those in the group that managed the database and interacted with it in their contributions will no doubt say that they learned a lot about relational databases, NoSQL databases, and even configuring databases themselves. The Spotify API was another key learning opportunity for those of us who hadn't really gotten dirty with JSON. The tedious process of extracting exactly what you need from a really beefy response is a valuable learning experience, if you want to think of it that way.

Goals

List of goals, where each is described, not just a bullet list

The primary goal of this app is to have a list of songs, such that users can sign into their profiles and offer ratings and reviews to these songs, as well as browse reviews from other users.

For You Page

The user's For You page is generated after they review at least three songs. This page utilizes Spotify's song recommendation algorithm to supply the user with songs they are likely to enjoy based on their past review history. From this page, a user can demo a song, and also tap on the song to review it. There is also the ability to "like" the song.

Explore Page

The Explore page shows the user songs that are currently popular/trending on Spotify. These are songs that are charting highly, and have received a lot of attention in recent times. From the explore page, the user can demo a song, and also tap on the song to review it. There is also the ability to "like" the song.

Tap That Like Button!

The most intuitive feature for our app is clicking the like button.

Play/Pause Button

The play button can be tapped on for each song displayed on the app. The song begins to play, and the user can pause the song once they are done hearing it.

Login/Logout Button

Upon starting the app, the user is prompted to create an account. This involves supplying a username, password, and email address. Once the account is created, the user may click the login button. Once the user is logged into the app, there is a logout button at the bottom of the screen that will log the user out and bring them back to the login menu.

What Do You Think?

"What Do You Think" is the name of our review feature, allowing the user to offer their opinions on songs, in the format of a text review and also a score out of 10. Once the user offers the review and rating, the information is stored using Google Firestore. These reviews will be visible when one clicks on a song, and for individual users, all of their own reviews are visible from the profile tab.

Future Directions

Potential future directions

- API tokens need to be refactored so that they utilize fresh tokens, and do not rely on someones spotify credentials
- QR Code feature to follow a user
- Explore page including more genres, eras of music, previous ratings etc. Better filter options
- Follow other users, have a feed where you can see peoples thoughts on new music
- Upvote, downvote features

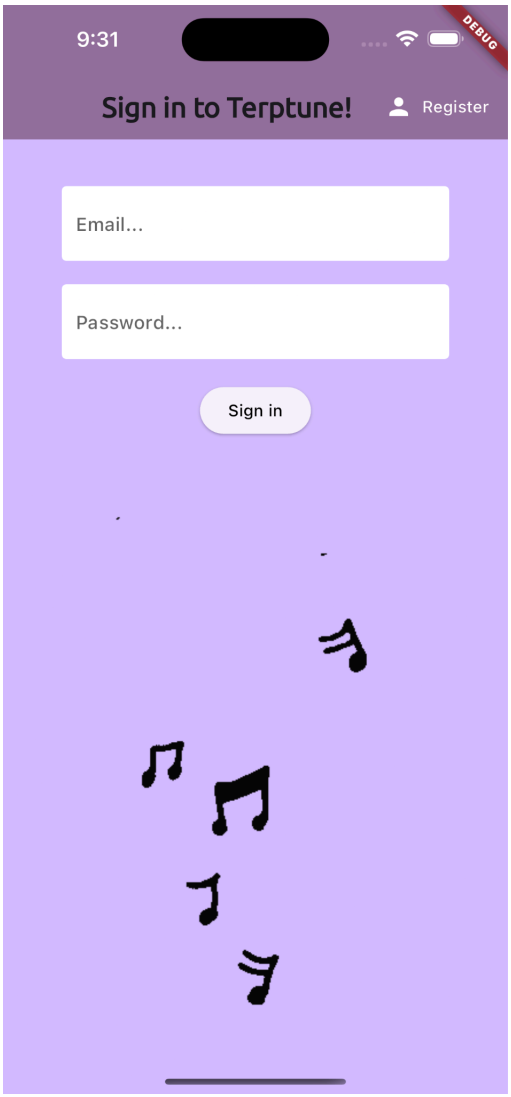
We are dedicated to prioritizing the user experience, aiming to provide the best possible journey for our users. To achieve this, we have several ideas that aim to foster a community while expanding the app's functionality. First, we recognize the importance of security, so we plan to refactor our API tokens to ensure they consistently use fresh tokens, rather than consistently relying on one user's Spotify credentials. Additionally, we aim to implement a QR code feature that allows users to easily follow each other, which could help to build a sense of community within the app and enable users to discover music recommendations from other users.

Moreover, we plan to enhance the Explore page so that it includes a wider range of genres, music eras, previous ratings, and more, which would provide users with more diverse and tailored music options. Currently, the Explore tab only displays the "top songs," so allowing users to go beyond that would be a big step. In line with our community-building efforts, we're also considering the development of a Feed tab. This feature would allow users to share their thoughts on new music, which would continue to develop the community feel within the app. Finally, we aspire to implement upvote and downvote features on music, reviews, posts, etc, which enables users to express their preferences and contribute to the curation of music content on the platform.

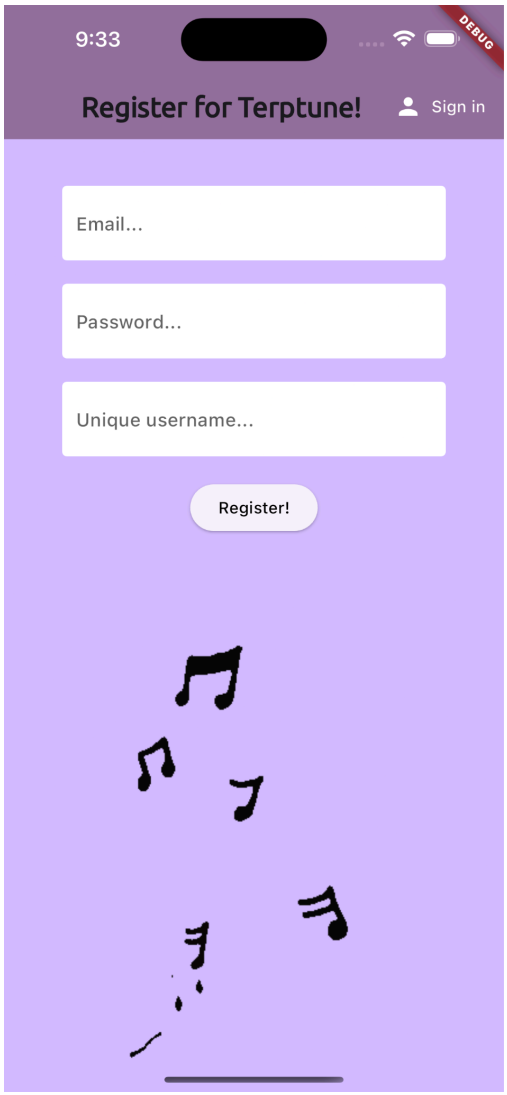
Walkthrough

App is run on IOS 17.4. The emulator is an Iphone 15 Pro Max

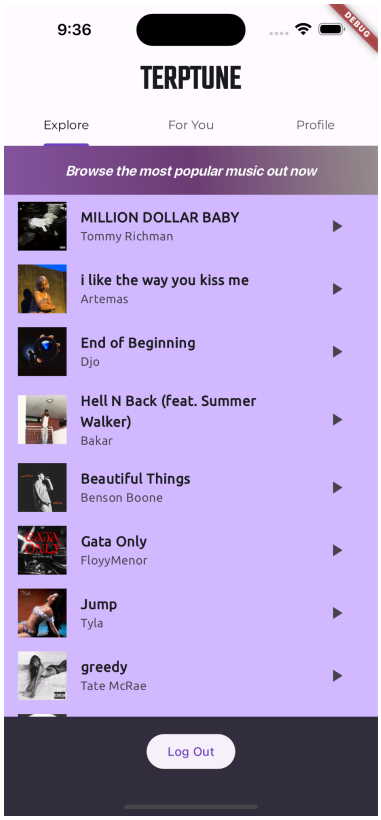
On startup our application shows a “sign-in” page.



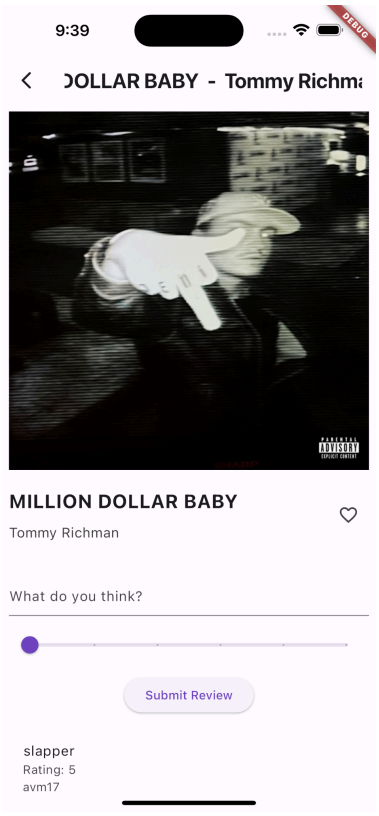
As well as a “register” page, if you do not have an account

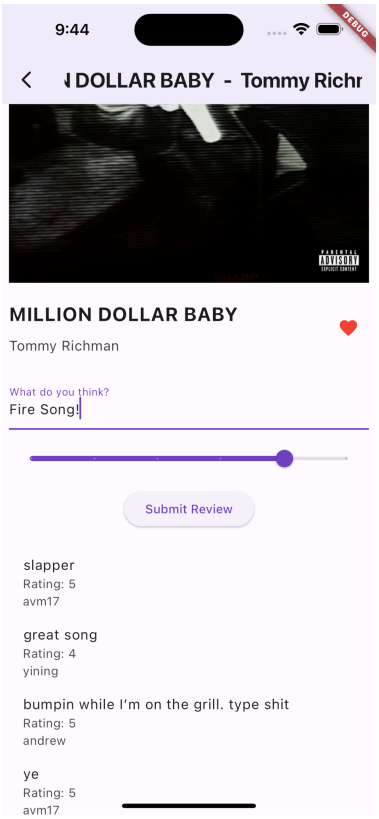


After logging in you are greeted with the explore screen of the application. That shows the current top 100 songs on Spotify. You have the ability to play previews of tracks that have an associated play button next to them!

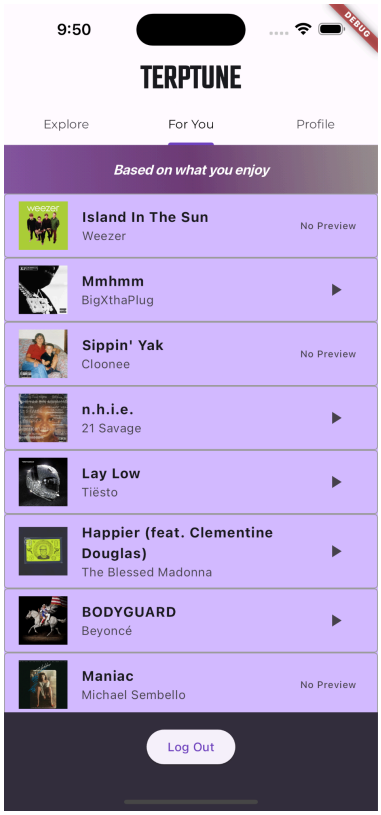


If you tap on a song for example, MILLION DOLLAR BABY by Tommy Richman. You'll be taken to the review page. Within the review page you can like a song, write a review, rate the song, and see other reviews from other users.

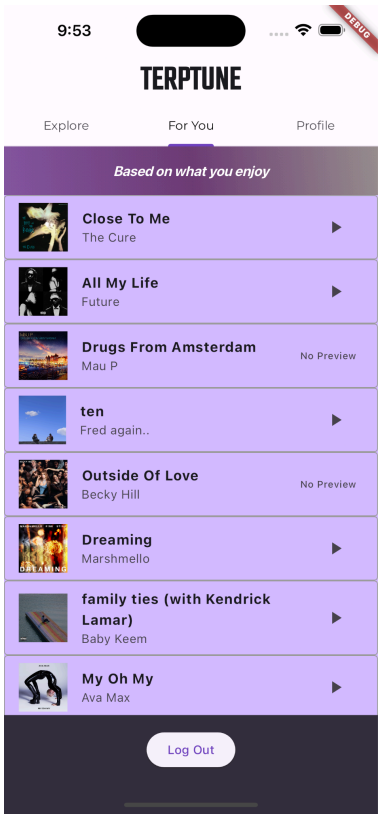


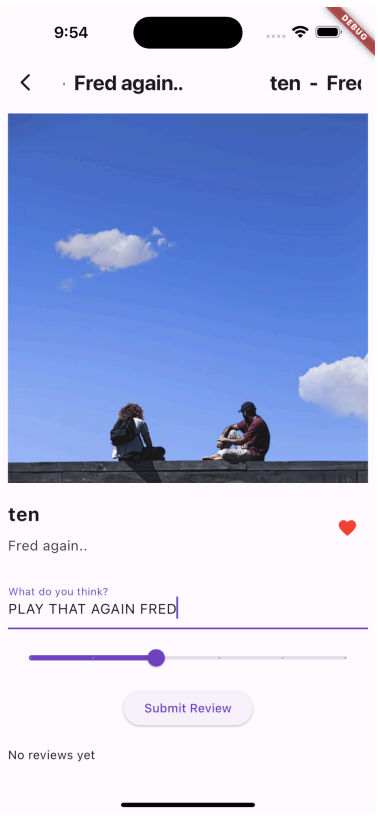


Next let's go to the for you page! After you have reviewed 3 different songs. The for you page will reveal recommended songs. This recommendation algorithm works by recommending you songs based on your high reviews.

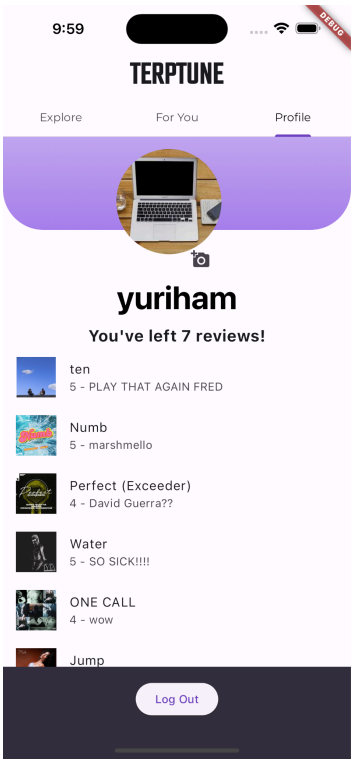


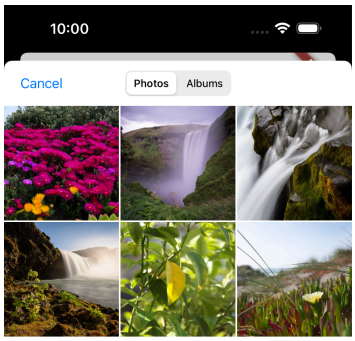
From within the review page you have the same ability to review songs that have been recommended to you as well!

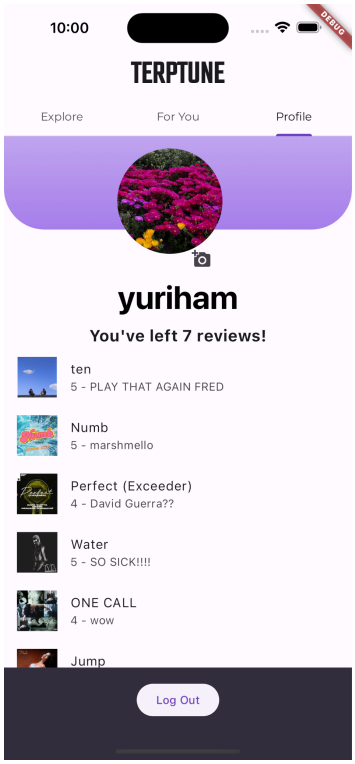




Let's move onto the profile Page! From the profile page you have the ability to see your name, how many songs you've reviewed, and have the ability to change your profile image!







Now the Shake and Rate Feature. If you shake your phone from any page an alert will pop up that will play a completely random song. From this alert you'll have the chance to rate this random song!