DJ Assistant: A Song Recommendation System for DJs

Human Computer Interaction Exam 2022

Johan Kresten Horsmans

201810219@post.au.dk

School of Communication and Culture

Aarhus University

Langelandsgade 139, 8000, Aarhus C, Denmark

June 10^{th}, 2022

Abstract

There are many different factors one needs to consider when curating songs for a DJ set. Since songs need to be crossfaded into a continuous and uninterrupted stream of music, it is required that tracks have congruent tempos and keys while, simultaneously, complementing each other aesthetically. Meanwhile, the amount of amateur DJs are on the rise and even most professionals are not trained in music theory. Followingly, these concepts can be difficult to relate to for a substantial portion of the DJ community. The following synopsis presents a solution to this problem, namely, the DJ Assistant app, which allows users to get song recommendations that are congruent with a prespecified track on the aforementioned parameters. Firstly, the theoretical and practical background behind the product is presented. This is followed by a thorough guide of different use cases and an analysis of how various theories from the field of Human Computer Interaction have been utilized and implemented in the UI. Lastly, the limitations and potential future prospects of DJ Assistant are discussed.

Keywords: DJing, Spotify API, UX, UI, Recommendation system.

Regarding character count and code:

The product is designed using Python 3.8.9 (Van Rossum & Drake, 2009) and the open-source web application framework, Streamlit (“Streamlit”, n.d.). The recommendation algorithm is based on the Spotify API (“Web API — Spotify for Developers”, n.d.). All code can be found in the following GitHub repository.

Character count: 16,795

Contents

1 Introduction

In its most basic essence, a *disc jockey*, or *DJ*, is a term for someone who plays recordings of music for an audience (Katz, 2012). Nonetheless, today it mostly covers the idea of someone who not only curates and plays music but also mixes songs together, through what is known as *crossfading*, into a continuous and uninterrupted stream of music to induce a hedonically pleasing state of mind for a group of listeners (Brewster & Broughton, 2014; Katz, 2012). It is in the latter sense that the term will be used for the remainder of this synopsis, which seeks to describe and discuss a novel tool for creating DJ sets, namely, *DJ Assistant* (henceforth referred to as *DJA*). DJA works by taking an input track and various user-specified filters whereafter it will propose a similar track that one can use as the next song in one's DJ set. Before expanding on the practicalities and uses of the tool, we need first to understand the justification and motivation behind its creation.

1.1 Background and motivation

The discipline of DJ-ing, while seemingly simple to the uninitiated, is oftentimes described as being difficult and, at the very least, harder than it looks (Attias & van Veen, 2011; Ishizaki et al., 2009; Katz, 2012). This is in large part due to the substantial amount of conventions and best practices that one needs to follow in order to create a pleasurable experience for the listeners (Steventon, 2014; Veire & De Bie, 2018). These conventions, amongst others, include that tracks need to have the same tempo (measured in *beats per minute*, or *BPM*), *key* and *'feel'* in order to be successfully crossfaded into each other (Cliff, 2000). These practices are regarded as common sense in the DJ community but their importance is also supported by theories and experimental findings in cognitive neuroscience.

First and foremost, the theory of *predictive coding*, which is central in modern neuroscience, suggests that a key component of musical enjoyment is rooted in listeners' ability to predict what will happen next (Salimpoor et al., 2015). In brief terms, this is because the predictive coding framework argues that the brain's key objective is to predict future events. This creates a rewarding sense of anticipation which is heightened if the outcome matches (or exceeds) what was expected whereas, vice versa, it is unrewarding if the actual outcome fails to match what was predicted (Salimpoor et al., 2015). In relation to music, it is hypothesized that this process of generating anticipatory predictions and subsequently having these expectations be confirmed or violated as the composition unfolds is what gives rise to musical pleasure (and displeasure) (Salimpoor et al., 2015). This theory is supported by both self-reported-, neurophysiological-, and psychophysiological data indicating that violated musical predictions are associated with negative emotional responses (Koelsch et al., 2008; Salimpoor et al., 2015; Steinbeis et al., 2006). This mechanism is suggested to be regulated by dopamine neurons in the midbrain that both signal prospective rewards and encode how well the actual outcome matches what was predicted (Peciña & Berridge, 2013; Rohrmeier & Koelsch, 2012; Salimpoor et al., 2015; Saunders et al., 2013; Schultz, 2013).

There are numerous different theories on what constitutes the fundamental origins of such musical expectations. One of the proposed sources is what is referred to as *implicit knowledge*, which covers the idea of one's fundamental understanding of the general rules of music (Huron, 2008; Salimpoor et al., 2015). Such implicit expectations for temporal, tonal and harmonic patterns are believed to arise due to systematic rules which are statistically learned through various exposures, cultural norms and musical education (Salimpoor et al., 2015). This theory is supported by numerous studies showcasing that implicit knowledge influences predictions concering both temporal synchronicity and harmonic congruency which, as stated before, are two of the main concerns of a DJ (Hallam et al., 2011; Huron, 2008; Marmel et al., 2010; Pearce & Wiggins, 2012; Rohrmeier & Koelsch, 2012; Salimpoor et al., 2015; Tillmann et al., 2008). As such, people who are exposed to mainly conventional Western music will predict most music to follow similar Western rules of music theory – such as songs being harmoniously consonant within a 12-tone system and all instruments being based around a synchronous and congruent tempo.

The findings and theories highlighted above underline that there are certain cognitive and hedonic benefits associated with harmoniously consonant and rhythmically synchronous crossfades (two songs in the same key and BPM) as opposed to harmoniously and rhythmically incongruent crossfades (two songs with different keys and BPM). In light of this, I argue that it is of utmost importance that DJs play songs that are matching both in key and tempo. Nonetheless, finding a sufficiently large selection of pleasurable music allowing one to create a coherent set of songs fulfilling the aforementioned criteria is no simple task. Especially given that a single DJ set can often last as long as 6 hours (Cliff, 2000).

Another aspect to take into consideration in relation to creating DJ sets in the modern DJing landscape is the rise of amateur DJs and digital tools. Traditionally DJ equipment consisted of two analog turntables for playing records and a mixer that could be used to crossfade between the two songs (Katz, 2012; MacCutcheon, 2014). Since the early 2000s, this has, for the most part, been replaced by digital solutions merged into a single and cheap module which has made DJing a much more accessible enterprise with a significantly lower cost of entry (MacCutcheon, 2014). As such, DJing has become increasingly more popular with many amateur *bedroom DJs* (Hall & Zukic, 2013; Katz, 2012; MacCutcheon, 2014; Sansom, 1998). Many professional DJs do not even know conventional music theory and one can expect even less from the rising amount of hobbyists, meaning that the musical concepts alluded to previously, such as BPM and keys, can be hard to fathom and, followingly, make it harder to curate good DJ sets (W. Marshall, 2009).

This is sought accommodated by the aforementioned digital solutions where, with a single click, one can transpose and stretch audio files to transform two songs to the same tempo and key despite not originally being congruent on those parameters (MacCutcheon, 2014). Nonetheless, there are various quirks associated with such transformations that can create strange artefacts that can make a song unpleasurable to listen to. As such, it is still preferable to link songs that are, if not identical, then at least similar in the starting point of their key and tempo. Furthermore, it is also important that songs fit the profile of the event that the DJ is playing at and, followingly, they have to be consistent in accordance with the stylistic music preferences – which only increases the difficulty of the task.

1.2 Proposed solution

The previous section illustrates the challenges that DJs are faced with when putting together coherent DJ sets. In light of these points, I argue that there is a need for a tool where DJs can easily find new songs for their sets while being rest assured that they will transition smoothly from whatever track is specified to come beforehand. Such a solution does, to the author's knowledge, currently not exist but DJA aims to solve the aforementioned challenges by creating an easy and effective tool for finding songs for one's DJ set that are all matching in musical style, tempo, key and other auditory features.

2 Product

2.1 Use case and functionality

For a video demonstration of the product, please see the following YouTube video. When users first launch DJA they are met with the UI seen in *figure 1.* Immediately visible at the bottom of the page, is an expandable section titled, *How to use*, containing a thorough guide to all the different features of the tool. Users are prompted to write the composing artist and name of a song that they want their recommendation based on. Once specified, DJA retrieves the given song data and displays them to the user in a table, as depicted in *figure 2.* Users are also allowed to specify advanced search features, by expanding an *advanced features* section (seen in *figure 1*).

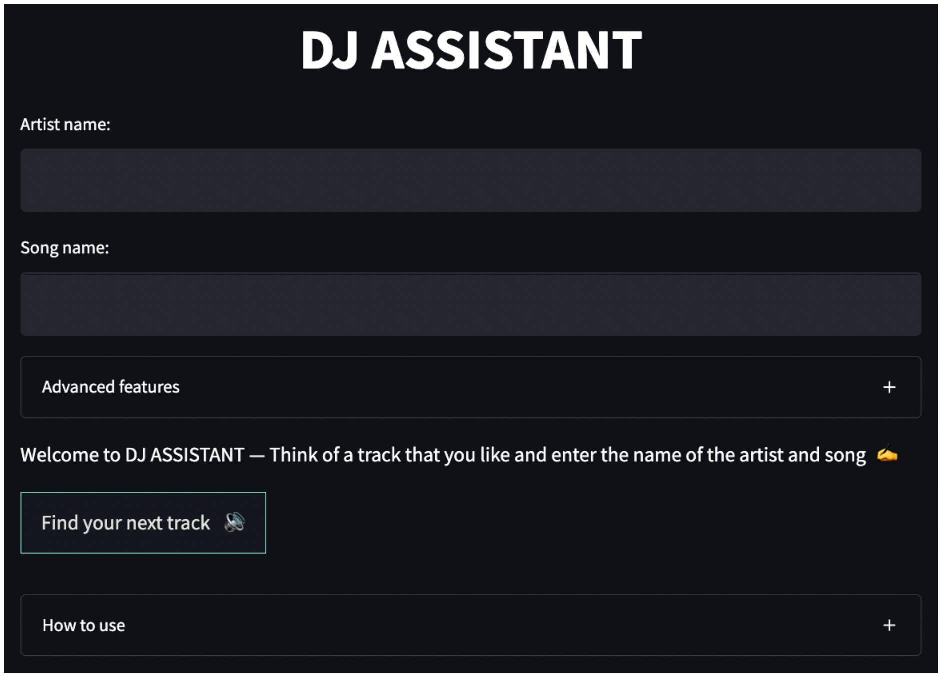


Figure 1: *Illustration of the landing page of DJA (see appendix A for full size figure).*

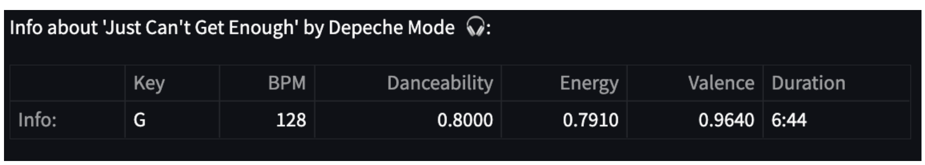


Figure 2: *Example of presented song data (see appendix B for full size figure).*

When this is pressed, users are met with checkboxes for activating each of the available advanced search filters. By ticking the different boxes, the specific customizable parameters for the respective features will appear on the screen (except for the "same key" parameter which takes a binary value). If users tick the *Minimum auditory feature* checkbox, there will appear another checkbox titled *Show description of audio features* which will show the description of the three available auditory features that can be used for search filtering, namely, *danceability, energy* and *valence.* An example of all advanced features and customizability interfaces can be seen in *figure 3.* When users have specified their song of choice and, optionally, advanced parameters, they can press the "Find your next track"-button and by doing so, DJA will generate a song recommendation as depicted in *figure 4*. This button can be pressed repeatedly and will each time generate a new recommendation. DJA will also display the artwork of the recommended song as a clickable hyperlink that directs the user directly to the song on Spotify.

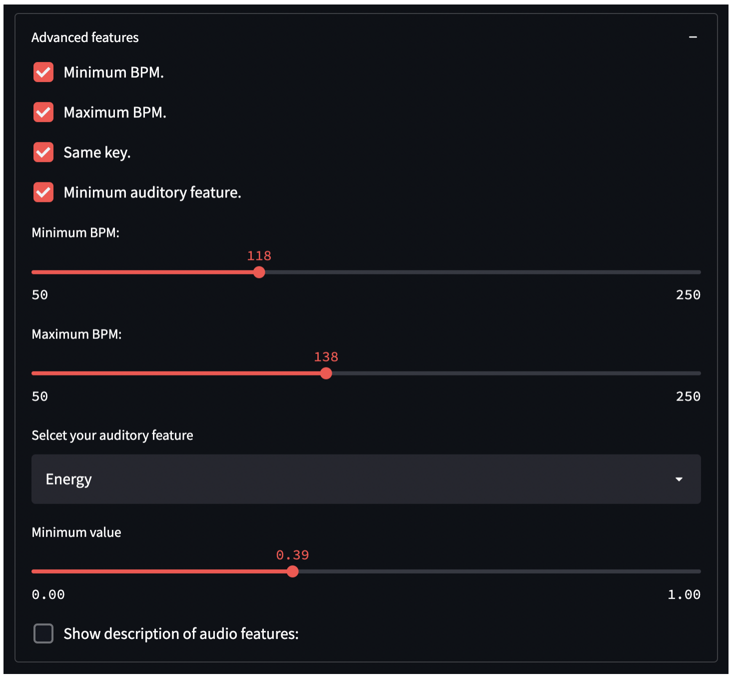


Figure 3: *Illustration of advanced features (see appendix C for full size figure).*

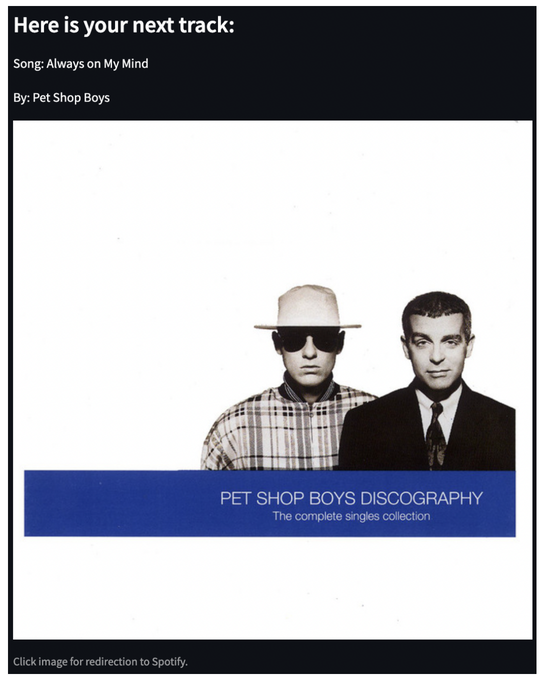


Figure 4: *Example of a recommended song output (see appendix D for full size figure).*

2.2 Design choices and theory

Several considerations, both based on theories of Human Computer Interaction (HCI) and concrete use-cases, were put into the UI design. Firstly, the app is designed with a 'dark' layout to accommodate dark clubs and concert venues where the product should be able to be used. By giving the app a dark interface it prevents potentially blinding the DJ when shifting between the utilized music-playing software and DJA, which would lead to a worse user experience. By keeping the interface dark, it is also less straining for the eye over time, which I also argue leads to an improved quality of the user experience as opposed to the interface being designed with a 'light' layout.

Another vital design consideration was concerning how many filtering features there should be and whether they should be optional or not. This talks into a prominent discussion within HCI, namely, the tradeoff between *functionality* and *usability* (Goodwin, 1987; Korhan & Ersoy, 2016). Because, even though increased functionality might make a product more flexible and powerful, it comes with a cost of usability due to the increased complexity of the interface – especially for novice users. As stated earlier, the problem that DJA tries to solve is relevant for both professional– and bedroom DJs meaning that the balance between usability and functionality should take both user groups into account. Followingly, the app features the expandable "advanced features" menu that allows professional DJs to have an additional level of customization based on their expertise while still allowing novice DJs to use the most basic functionality without disturbances.

Nonetheless, for the purpose of usability, there has still been excluded a large array of possible customizability features from the 'advanced'-section. Through the Spotify API, the default amount of data for retrieved songs consists of 13 auditory features. These include, e.g., *acousticness, liveness* and *loudness*. The three included features were selected on the basis of being the most relevant when curating a DJ set – as evaluated through a series of DJ handbooks (see, Attias and van Veen (2011), Katz (2012), and Steventon (2014)). When filtering the auditory features, users are only allowed to specify a minimum value since, I argue, it would not be a very useful feature to filter max values of the selected filters – i.e., based on the aforementioned DJ handbooks, few DJs would care to specify that their songs should not be *too* energetic or danceable. Therefore this feature would mainly be a distraction and, as such, this is a clear case where concerns about usability have led to a limitation of functionality. BPM and key were included as possible filter features due to the reasons mentioned in *Background and motivation*.

Furthermore, the UI features continuous *feedback* from the system to the user. This feedback is designed around the idea of *contingency*, meaning that the messages received by the user are contingent upon the actions sent by the user (Sundar et al., 2016). For instance, when launching the app, the user is met with the message *"[...] enter the name of the artist and song"* and when, e.g., the artist's name is entered, the message changes to *"Good, now you just need to enter the name of the song"*. Such contingency-based feedback is shown to improve user experience (Sundar et al., 2016). The feedback is deliberately devised to appear as if sent by an actual agent. This is also known as the *anthropomorphic approach* and it facilitates a more intuitive and user-friendly UX since such messages are easier to interpret and, followingly, easier to act upon for the users (Eberts, 1994). For an example of such communication, see *figure 5* which illustrates an error message resulting from not being able to locate a song in the Spotify database. To make messages even more anthropomorphic, emojis are used since they are usually reserved for human-to-human communication. Furthermore, these emojis are primarily included in places where actions are required from the user. The chosen emojis are relevant to the given task and, followingly, they act as *metaphors* that can efficiently facilitate the task that they are mapped to (P. Marshall & Hornecker, 2013).

Lastly, users can be redirected to the recommended songs on Spotify by clicking on the displayed album artwork. This is done since many modern DJ tools have Spotify integration, thus allowing direct loading from Spotify into the DJ tool of choice which facilitates a more efficient workflow.

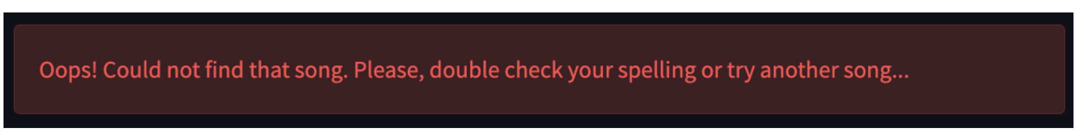


Figure 5: *Example of an anthropomorphic error message (see appendix E for full size figure).*

3 Discussion

3.1 Limitations

A key weakness of DJA is that the underlying recommendation algorithm of the Spotify API is a complete black box. This means that there are certain flaws in DJA that are not amendable from a developer's perspective. For instance, sometimes, the recommended song for a given track will be the radio-edit version of the same track. This is obviously not optimal since few DJs would ever want to crossfade between two nearly identical versions of the same song. Another limitation caused by the inflexibility of the Spotify API is that it bases its recommendations on behavioral user data. As such, DJA will only recommend music that already has a certain degree of popularity, which potentially creates a feedback loop where the algorithm will never recommend completely new music that does not have many listeners yet. As such, DJA is at risk of diminishing the spread of new underground music. This is a problem since a large part of the DJ profession is to always be ahead of the curve and play the hits of tomorrow rather than only the hits of yesterday (Katz, 2012; Steventon, 2014).

Another problem with DJA is that the Spotify API only provides 'root' keys - i.e. it does not differentiate between, e.g., *C# major* and *C# minor* but will, simply, instead classify both as *C#*. This is a substantial problem since it will not always be possible to harmoniously crossfade a track in C# major into a track in C# minor. Furthermore, DJA currently only allows users to search for songs in the same root key but, c.f. the music theory concept known as *the circle of fifths*, it is also possible to smoothly crossfade into a new song if the key changes to the *subdominant*–, *dominant*– or *relative key* (Katz, 2012; Veire & De Bie, 2018).

3.2 Future prospects

The problem of DJA only retrieving root keys, could be solved by merging the Spotify API with key-data from music theory databases such as NoteDiscover. Furthermore, it could be beneficial to add filtering options that allowed for recommendations in other congruent keys. Moreover, oftentimes DJs already have more than a single song they want to include in their set. Followingly, it would be relevant for users to be able to generate recommendations based on entire playlists. This could most likely also improve the quality of the generated recommendations since there would be a much larger basis for the recommendation.

Furthermore, it would be relevant to add an embedded Spotify player in the app, so users could listen to a preview of the recommended song without having to alternate between apps. In this case, users could easily opt for a new recommendation if the current one was not to their liking – all while staying in the DJA app. Lastly, it would also be relevant to conduct a user test of the software on actual DJs in order to expose potentially amendable flaws relating to the UI and UX.

References

Attias, B. A., & van Veen, T. C. (2011). Off the record: Turntable and controllerism in the 21st century (part 1).

Brewster, B., & Broughton, F. (2014). *Last night a DJ saved my life: The history of the disc jockey*. Open Road+ Grove/Atlantic.

Cliff, D. (2000). Hang the DJ: Automatic sequencing and seamless mixing of dance-music tracks [Publisher: Hewlett-Packard Laboratories]. *Hp Laboratories Technical Report Hpl*, *104*.

Eberts, R. E. (1994). *User interface design*. Prentice-Hall, Inc.

Goodwin, N. C. (1987). Functionality and usability [Publisher: ACM New York, NY, USA]. *Communications of the ACM*, *30*(3), 229–233.

Hall, M. M., & Zukic, N. (2013). The DJ as electronic deterritorializer. *DJ Culture in the Mix: Power, Technology, and Social Change in Electronic Dance Music. New York: Bloomsbury Publishing*, 103–122.

Hallam, S., Cross, I., & Thaut, M. (2011, May 26). *Oxford handbook of music psychology* [Google-Books-ID: 74LnCwAAQBAJ]. Oxford University Press.

Huron, D. (2008, January 25). *Sweet anticipation: Music and the psychology of expectation*. MIT Press.

Ishizaki, H., Hoashi, K., & Takishima, Y. (2009). Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort. *ISMIR*, 135–140.

Katz, M. (2012). *Groove music: The art and culture of the hip-hop DJ*. Oxford University Press on Demand.

Koelsch, S., Fritz, T., & Schlaug, G. (2008). Amygdala activity can be modulated by unexpected chord functions during music listening [Publisher: LWW]. *Neuroreport*, *19*(18), 1815–1819.

Korhan, O., & Ersoy, M. (2016). Usability and functionality factors of the social network site application users from the perspective of uses and gratification theory [Publisher: Springer]. *Quality & quantity*, *50*(4), 1799–1816.

MacCutcheon, D. (2014). Two studies investigating the value of DJing for contemporary music education.

Marmel, F., Tillmann, B., & Delbé, C. (2010). Priming in melody perception: Tracking down the strength of cognitive expectations. [Publisher: American Psychological Association]. *Journal of Experimental Psychology: Human Perception and Performance*, *36*(4), 1016.

Marshall, P., & Hornecker, E. (2013). Theories of embodiment in HCI [Publisher: Sage Thousand Oaks]. *The SAGE handbook of digital technology research*, *1*, 144–158.

Marshall, W. (2009). Unlocking the groove: Rhythm, meter, and musical design in electronic dance music.

Pearce, M., & Wiggins, G. (2012). Auditory expectation: The information dynamics of music perception and cognition. *Topics in cognitive science*, *4*, 625–52. https://doi.org/10.1111/j.1756-8765.2012.01214.x

Peciña, S., & Berridge, K. C. (2013). Dopamine or opioid stimulation of nucleus accumbens similarly amplify cue-triggered 'wanting'for reward: Entire core and medial shell mapped as substrates for PIT enhancement [Publisher: Wiley Online Library]. *European Journal of Neuroscience*, *37*(9), 1529–1540.

Rohrmeier, M. A., & Koelsch, S. (2012). Predictive information processing in music cognition. a critical review. *International Journal of Psychophysiology*, *83*(2), 164–175. https://doi.org/10.1016/j.ijpsycho.2011.12.010

Salimpoor, V. N., Zald, D. H., Zatorre, R. J., Dagher, A., & McIntosh, A. R. (2015). Predictions and the brain: How musical sounds become rewarding [Publisher: Elsevier]. *Trends in cognitive sciences*, *19*(2), 86–91.

Sansom, M. (1998). Jungle DJs: Valuing music in dance culture [Publisher: Citeseer].

Saunders, B. T., Yager, L. M., & Robinson, T. E. (2013). Cue-evoked cocaine "craving": Role of dopamine in the accumbens core [Publisher: Soc Neuroscience]. *Journal of Neuroscience*, *33*(35), 13989–14000.

Schultz, W. (2013). Updating dopamine reward signals. *Current Opinion in Neurobiology*, *23*(2), 229–238. https://doi.org/10.1016/j.conb.2012.11.012

Steinbeis, N., Koelsch, S., & Sloboda, J. A. (2006). The role of harmonic expectancy violations in musical emotions: Evidence from subjective, physiological, and neural responses. *Journal of Cognitive Neuroscience*, *18*(8), 1380–1393. https://doi.org/10.1162/jocn.2006.18.8.1380

Steventon, J. (2014). *Steventon: DJing for dummies - google scholar*. Retrieved June 6, 2022, from https://scholar.google.com/scholar_lookup?title=DJing%20for%20dummies&publication_year=2014&author=Steventon%2CJ.

Streamlit. (n.d.). Retrieved June 9, 2022, from https://docs.streamlit.io/

Sundar, S. S., Bellur, S., Oh, J., Jia, H., & Kim, H.-S. (2016). Theoretical importance of contingency in human-computer interaction: Effects of message interactivity on user engagement [Publisher: SAGE Publications Inc]. *Communication Research*, *43*(5), 595–625. https://doi.org/10.1177/0093650214534962

Tillmann, B., Janata, P., Birk, J., & Bharucha, J. J. (2008). Tonal centers and expectancy: Facilitation or inhibition of chords at the top of the harmonic hierarchy? [Publisher: American Psychological Association]. *Journal of Experimental Psychology: Human Perception and Performance*, *34*(4), 1031.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

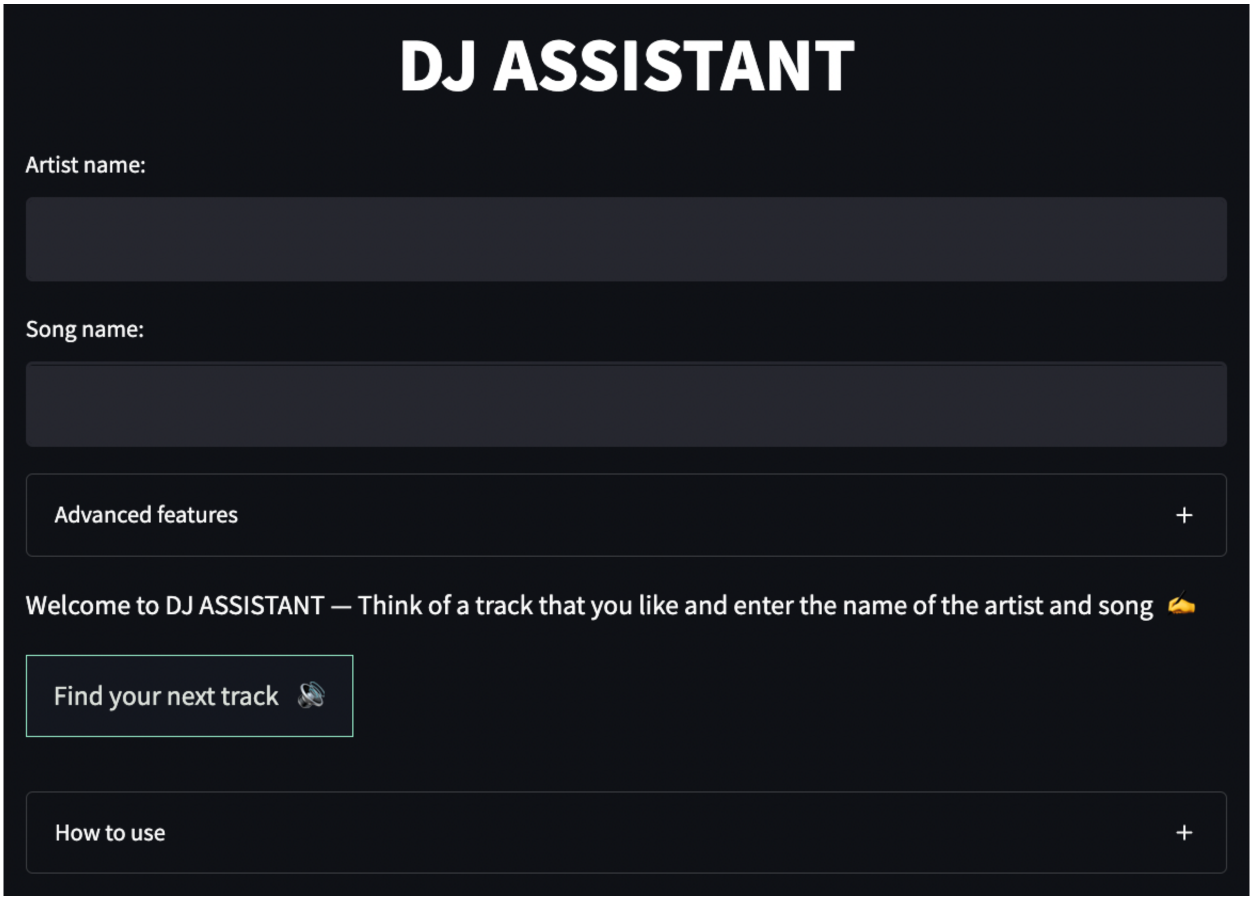
Veire, L. V., & De Bie, T. (2018). From raw audio to a seamless mix: Creating an automated DJ system for drum and bass [Publisher: SpringerOpen]. *EURASIP Journal on Audio, Speech, and Music Processing*, *2018*(1), 1–21.

Web API — spotify for developers. (n.d.). Retrieved June 7, 2022, from https://developer.spotify.com/documentation/web-api/

4 Appendix

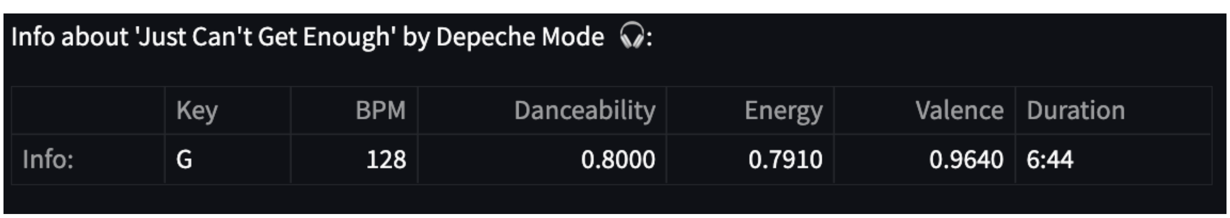
A

Figure 1:



B

Figure 2:



Info about 'Just Can't Get Enough' by Depeche Mode 🎧:

	Key	BPM	Danceability	Energy	Valence	Duration
Info:	G	128	0.8000	0.7910	0.9640	6:44

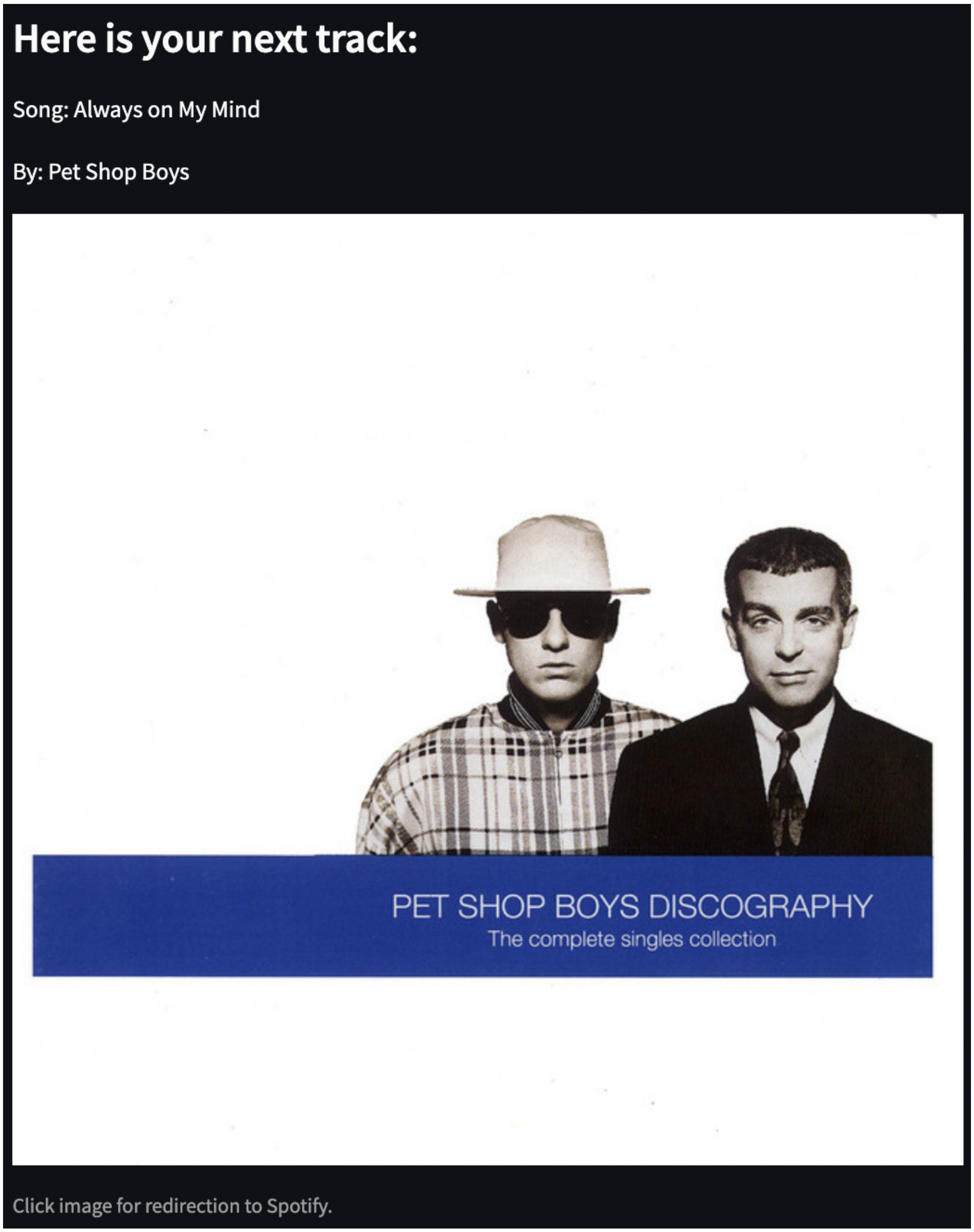
C

Figure 3:


Advanced features
Minimum BPM.
Maximum BPM.
Same key.
Minimum auditory feature.
Minimum BPM:
118
50
250
Maximum BPM:
138
50
250
Selcet your auditory feature
Energy
Minimum value
0.39
0.00
1.00
Show description of audio features:

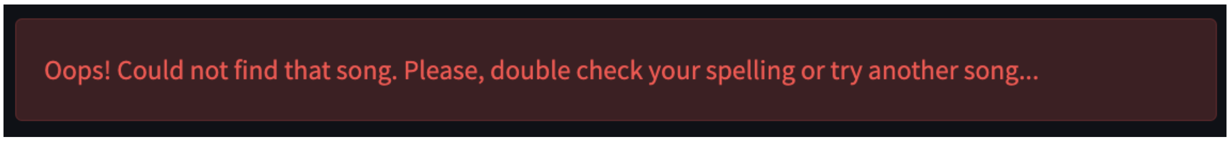
D

Figure 4:


Here is your next track:
Song: Always on My Mind
By: Pet Shop Boys
PET SHOP BOYS DISCOGRAPHY
The complete singles collection
Click image for redirection to Spotify.

E

Figure 5:


Oops! Could not find that song. Please, double check your spelling or try another song...