

# PlayBack: A New Approach to Music Engagement

A Personalized and Uninfluenced Listening Experience

Gaetano Panzer II, Max Collins, Ryan Ferrell, RJ Walsh

February 2025

# The Problem with Traditional Music Review Platforms

- Traditional review sites shape opinions before listeners form their own.
- Numeric ratings and analytical reviews dominate perception.
- Community-driven platforms still rely on aggregate scores.
- Streaming services reinforce past listening patterns, limiting discovery.

ALBUMS

# *Reputation*

Taylor Swift

2017



6.5

By Jamieson Cox

## Influence of Traditional Reviews

- **Pitchfork:** Numeric rating displayed before the review.

## Joji Nectar



### CRITIC SCORE

67

Based on 5 reviews  
2020 Ratings: #801 / 875

### USER SCORE

72

Based on 5,239 ratings  
2020 Rank: #435

MORE

### DETAILS

[SUBMIT CORRECTION](#)

[September 25, 2020](#) / release date

[LP](#) / format

[88rising](#), [12Tone](#) / label

[Alternative R&B](#), [Alt-Pop](#)

[Art Pop](#), [Pop Rap](#), [Trap](#) / genre

[Joji](#), [The Donuts](#), [Bekon](#), [west1ne](#), [Isaac Sleator](#), [+22 more...](#) / producer


[Joji](#), [Daniel Wilson](#), [Tyler Reese](#), [Bekon](#), [Chakra](#), [+39 more...](#) / writer

[FULL CREDITS](#)

# Influence of Traditional Reviews

- **Album Of The Year:** Aggregate scores shape expectations.


RYM | new music genres charts lists | Search...



**GNX**

Artist: **Kendrick Lamar**  
 Type: **Album**  
 Released: **22 November 2024**  
 RYM Rating: **3.71** / 5.0 from **27,615** ratings  
 Ranked: **#14** for 2024, **#1,773** overall  
 Genres: **Hip Hop**  
 Conscious Hip Hop, Ratchet Music, Nervous Music, Hardcore Hip Hop, Hyphy, Trap, Pop Rap  
 Scenes: **West Coast Hip Hop**  
 Descriptors: boastful, male vocalist, aggressive, energetic, rhythmic, conscious, introspective, sampling, eclectic, playful, humorous, dark, quirky, party, lush, hateful, sarcastic, raw, manic, Christian  
 Language: English

AD (0:09)



Previous: Mr. Morale & The Big Stepp... Kendrick Lamar EP

RYM | new music genres charts lists | Search...

Include all languages Page 1 2 3 .. 7 .. 11 .. 14 .. 18 .. 22 .. 25 .. 2

**neckpickup** Feb 25 2025

**decent, but underwhelming**

first listen I thought this was pretty median (kendrick's median), just a simple mixtape more 4fun than any of masterpieces (kendrick always suffers from success, as in he set up the bar so high we do not expect him to solid good album for fun, we always expect something mind blowing, at least I kind off do).

But then, it's all about perspective.  
 I do not think this is an album to listen like TPAB, it's an album to turn up high in some good speakers (car, etc) yourself to. I do not go to the gym but I could imagine myself using this as hype, on in a party, etc. It's a hype simple and easy to listen and more commercial, but still with some of his artistic touch and some soul.

With this said and with more listens, it still doesn't quite satisfy me much. With a different perspective, I stay listen thoughts. It has something near 3 nice songs and some decent ones, but overall it's a bit underwhelming not really much substance and essence here, a bit too raw and lots of "filler" songs. I'm giving it a 3.5 probably Kendrick bias and off the good songs I'll be relistening in a playlist rotation sometimes.

Also, it's curious the impact of this new Kendrick side has had on social media, pop culture, in "fancams" music videos, etc... Very interesting, fun, and deserving. Proof he can do it all. For the people who said they don't Kendrick regularly and don't put him in their car, there's not even that argument anymore.

Probably the most complete and with the biggest peak artist in hip-hop and even in others. Shoutout Kendrick Lamar

This album made me go back do untitled unmastered and TPAB, so, I guess it was not that disappointing if that.

Published

# Influence of Traditional Reviews

# Rate Your Music (RYM): Endless opinions before forming one's own.

# Streaming Algorithms and the Filter Bubble

- Spotify & Apple Music dominate the industry.
- Algorithms reinforce listening habits, limiting discovery.
- Fragmented listening history across platforms.
- Offline listening is not captured.

# Introducing PlayBack

- **No public numeric ratings:** Private ratings for personal reference.
- **Post first, then view others' thoughts:** Encourages unbiased impressions.
- **Manual tracking of listening habits:** Provides a complete music journey.

# How PlayBack Stands Out

- **Personal Reflection over Consensus.**
- **No Aggregate Ratings or External Influence.**
- **User-Driven Tags ('Vibes') for Discovery.**
- **Concise, Accessible 'Thoughts' System.**



# Who Benefits from PlayBack?

- **Casual Listeners:** No pressure from traditional reviews.
- **Seasoned Enthusiasts:** Encourages concise and intentional reflections.
- **All Music Fans:** A personal, uninfluenced, and exploratory music

# Looking Ahead

- PlayBack sets the foundation for a new way to engage with music.
- The following sections will explore technical implementation and project scope.
- Understanding these elements will highlight how PlayBack achieves its vision

# Resources Needed for Project

- Music API access
  - Way to retrieve artist/song/album data to be displayed on site.
  - Spotify's API is free to use, and while it has some limitations on usage frequency it likely won't be an issue for our project.
- Database
  - Storing posts, user information, and all relevant data to be used in the project.
  - MySQL Community Edition is free and easy to use and will satisfy the requirements of our project.

# Resources Needed for Project

- Backend
  - Handling API connection, database connection, and site logic.
  - Flask is a Python framework that is simple, lightweight, and is free to use and should meet the needs of our project.
- Frontend
  - Libraries that will control what and how data is displayed.
  - React is a free JavaScript library and has everything we need for the site. Emphasis on being easy to use.
- Graphing/ Planning
  - Allows us to plan site structure and inner workings, and create and reference a primary schedule to keep us on track.
  - Drawio.com is a free and easy to use website that has fulfilled our needs thus far and will likely continue to do so later in development.

# Resources Needed for Project

- Authentication
  - Controlling user sign in and security.
  - Node.js libraries, those of which we will be using are free and easy to learn (maybe Passport.js?).
- Server Hosting
  - Way to keep site active without having to constantly run on one of our devices.
  - Will cost money but is easy to set up. The service we use will likely be determined by which has the lowest cost at the time of completion.

# Work Breakdown Structure (WBS)

Items are dependent on sub-bulleted points completion before their own completion.

- Design
  - Database schema creation
  - Website file directory flow and page layout mapping
  - Deciding which resources to use for each section of development
    - Finalization of app scope (what features to implement)
- Front-end
  - Setting up user authentication method for sign in
    - Setting up connection between front-end and back-end
  - Reviewing appearance of website and usability
    - UI design and implementation
- Back-end
  - Testing back-end functionality
    - User creation and user data storage
      - Setting up connection with database and database implementation
    - API querying and results formatting
    - Storing comments, reviews, and other posts and returning them
      - Setting up connection with database and database implementation

# Cost Estimation Based on Lines of Code (LOC)

Database Schema Creation	0
Website file directory flow and page layout	0
Deciding which resources to use	0
Finalization of PlayBack scope	0
Setting up user authentication (Javascript, Python)	~300
Setting up front-end back-end connection (Javascript, Python)	~300
Reviewing appearance of website	0
UI design and implementation (Javascript, HTML, CSS, Python)	~2000
Testing back-end functionality (Python)	~100
User creation and data storage (Python, SQL)	~700
Setting up connection between back end and database implementation (Python, SQL)	~500
API querying and results formatting (Python)	~700
Storing post data and returning data (Python, SQL, Javascript)	~100
Total	4800

Figure 1: Lines of Code Estimation Table

## Things to Consider

- The average junior developer can write 100 lines of code a day (McEnry, 2020).
- The average salary of a junior developer in Wilmington is \$83,913 a year according to ZipRecruiter.
- Using these two values, a cost of \$11,055 was produced.

# Cost Estimation Based on Function Points

Information Domain Value	Optimistic	Likely	Pessimistic	Est. Count	Weight	FP count
External Inputs	4	5	6	5	4	20
External Outputs	1	1	2	1	1	1
External Inquiries	3	4	5	4	4	16
Internal Logic Files	2	2	3	2	2	4
External Interface Files	0	1	1	1	1	1
Count Total						42

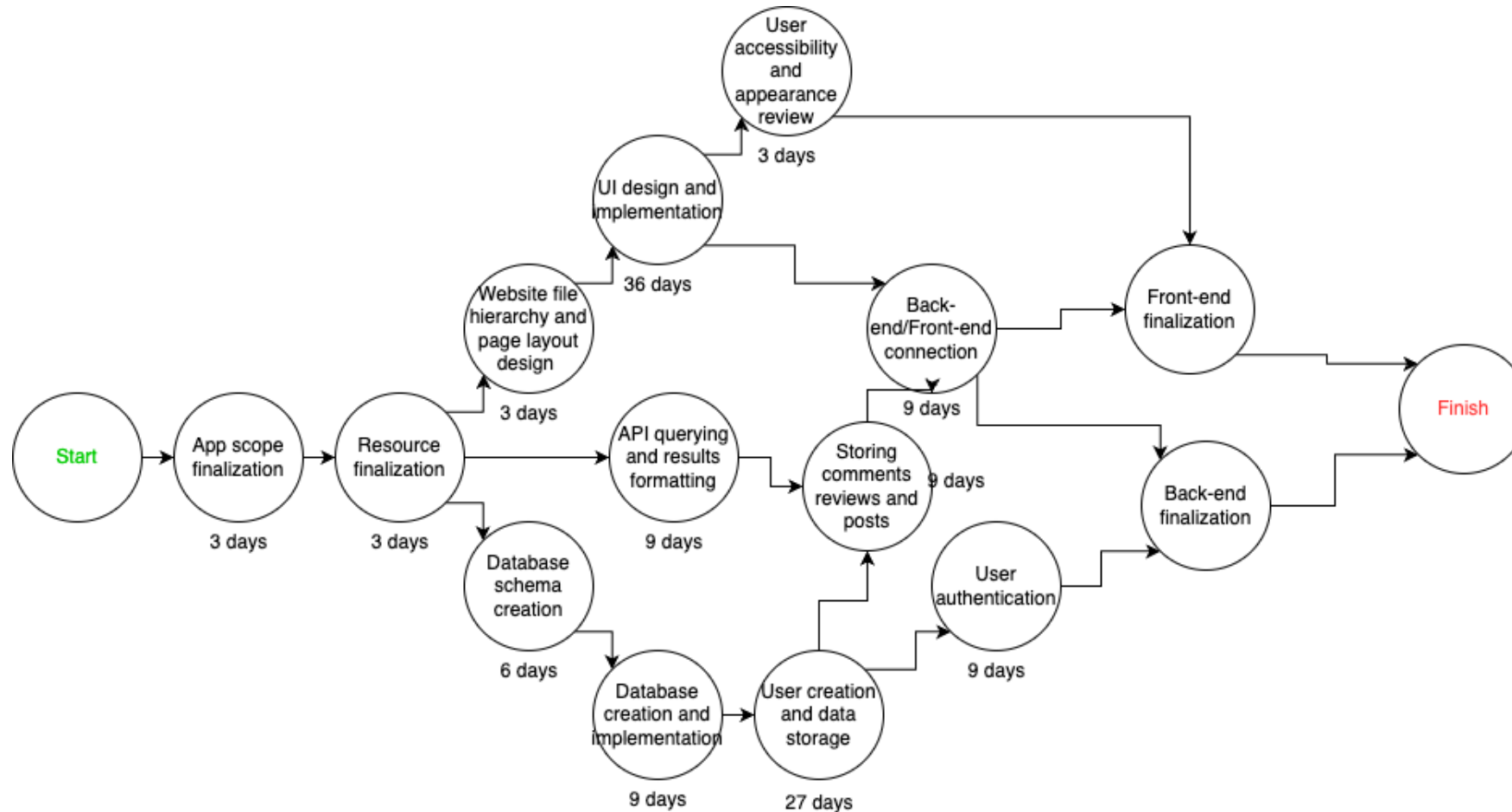
Figure 2.1: Function Point Computations

## Things to consider

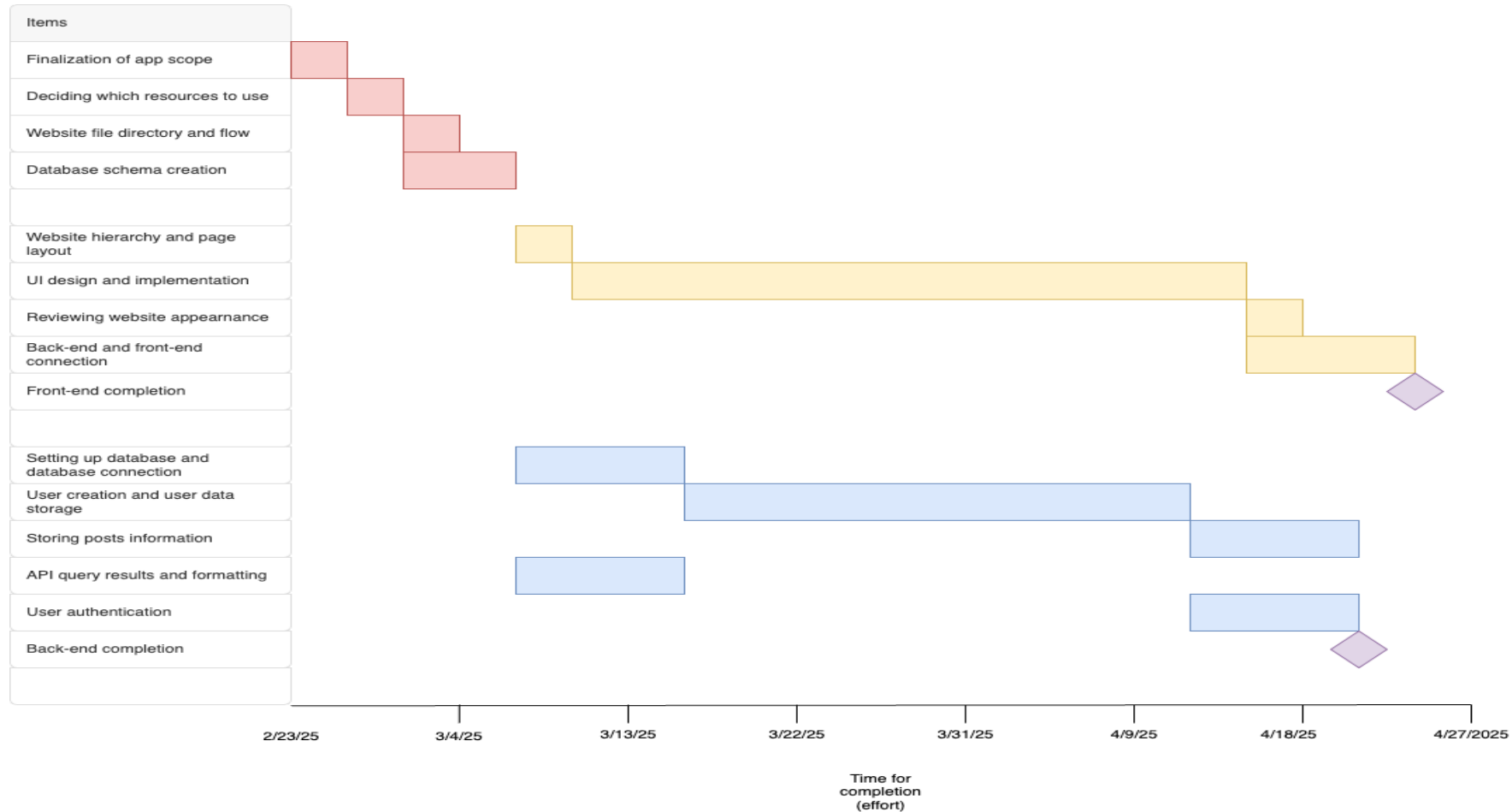
- Salary estimate of \$83,913 was reused.
- Value adjustment factor is 0.78.
- Assumes it takes a month to complete 10 function points.
- Estimated cost is \$22,801



# Scheduling Estimation Graphs (PERT)



# Scheduling Estimation Graphs (GANTT)



# Data Dictionary

## Post Data



Data Name	Type	Data Format	Num. Of Bytes	Description	Example	Limitations
User Post	String	<u>255 character string</u>	256	Basic user post, references user	"I like this song!"	Less than 255 characters.
User reply	String	<u>255 character string</u>	256	<u>User</u> reply to post, references user	"You're wrong!"	Less than 255 characters.

## Song Data

Data Name	Type	Data Format	Num. Of Bytes	Description	Example	Limitations
Genre Tags	String Array	6 element <u>string array</u>	~	List of song tags	Rock, rap...	Less than 6 elements
Song interactions	INT		4	Tracks engagement	Posts: 10	

# Data Dictionary

User Data						
Data Name	Type	Data Format	Num. Of Bytes	Description	Example	Limitations
Username	String		20	User identifier	CatLover96	UTF characters, less than 20 characters, must be unique
User Password	String		20	User validation key	Password!	UTF characters, more than 5 characters, less than 20 characters
Date Joined	Float	MM/DD/YYYY	4	Tracks when user joined	02/10/2025	
Post History	Array		~	Points to all user posts		

# Responsibility Matrix

PlayBack Responsibility Matrix - RACI Model					
Tasks	Gaetano Panzer II	Max Collins	Ryan Farrell	RJ Walsh	
Webpage Design	C	R	I	I	
Front-end Implementation	R	C	I	I	
Database Construction	C	R	C	C	
Back-end Implementation	I	I	R	R	
Spotify API Implementation	A	A	R	R	
Authentication	R	I	I	I	I - Informed
Server Hosting	R	I	I	I	

# Risk Management

Risk	Category	Probability	Impact	Priority	RMMM
Spotify API cannot be properly implemented into project	Technology Risk	1	5	5	All team members will make sure they know how to use the Spotify API. If the Spotify API cannot be used, a suitable alternative will need to be found
Issues with crashing or poor load times occur on the website	Technology Risk	5	4	4	Performance issues will be inevitable when working with new technology, the software will need to be extensively tested and good practices should be followed when programming to optimize the code and user experience
Server hosting cost	Business Risk	4	4	4	PlayBack will be a website that will need to be hosted for users to access it. The cheapest hosting option should be considered as to not waste too much money during development. If it is possible to host for free, that should be the option we go with
Team members will not be able to complete their tasks	People Risk	2	3.5	4	If, for any reason, a team member is unable to complete a task, other team members may need to step in to assist in or take over the task at hand
A software functionality cannot be implemented due to time constraints	Product Size Risk	2.5	3	3.5	Functionalities of the software should be ranked by how important they are, when time constraints begin to set in lower priority functions may be dropped to focus on more important tasks

# Anticipated Issues and Constraints

- Introduction to Spotify API
  - Learning curve for team members
- General development issues
  - Bugs in code
  - Optimization issues
- Real world constraints
  - Semester time limit
  - Workload from other courses

# Project Control

- GitHub
  - Version Control
- Discord
  - Voice/Text Communication
  - Progress Updates
- Microsoft Word
  - Planning/Idea Documents
  - Requirements Engineering Report



# Citations

- McEnery, Sage. “How Much Computer Code Has Been Written?” *Modern Stack*, 18 July 2020, [medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459](https://medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459).