

Branti
A PROJECT REPORT
In partial fulfillment for the award of the degree
Of
INTEGRATED CERTIFICATE CUM DIPLOMA
IN
COMPUTER SCIENCE & ENGG., DATA ENTRY & WORD PROCESSING



Under the guidance of
Er. Rahul Gautam
Assistant Professor
CSE Department

Submitted by
Siddhant Kumar(1912505)
Rishabh Raj(1912010)

**SANT LONGOWAL INSTITUTE OF ENGINEERING AND TECHNOLOGY,
LONGOWAL – 148106, DISTRICT- SANGRUR, PUNJAB, INDIA**

May, 2022

CERTIFICATE

Certified that this project report "Branti" submitted by **Siddhant Kumar (cde-1912505)** and **Rishabh Raj (cde-1912010)** in the partial fulfillment of the requirements for the award of Diploma.

This report has not been submitted to any other University or Institute for the award of any degree.

Siddhant Kumar
Siddhant Kumar
(cde-1912505)

Rishabh Raj
Rishabh Raj
(cde-1912010)

This is to certify that the above statement made by the candidate(s) is correct to the best of their knowledge.

Er. Rahul Gautam
Assistant Professor
(CSE DEPARTMENT)

The viva-voce examination of Siddhant Kumar and Rishabh Raj held on

Table of Contents

- i) Certificate
- ii) Acknowledgement
- iii) Abstract

- 1) Chapter 1. Introduction
- 2) Chapter 2. Problem Definition
- 3) Chapter 3. Objectives
- 4) Chapter 4. List of Figures
- 5) Chapter 5. Methodology Used
- 6) Chapter 6. Results and Screenshots

Acknowledgment

We would like to thank our supervisor, Er. Rahul Gautam who helped us a lot to work on the Branti. His suggestions and directions have helped in the completion of this project.

We would also like to express our gratitude to our Class Counselor, Preetpal Kaur Buttar who provided us with the information that helped us coordinate the development of the Branti.

We would also like to express our gratitude towards the free and open source libraries/frameworks that helped in the successful completion of this project. We used Ruby on Rails (<https://rubyonrails.org>) for our project.

Abstract

Branti is a music streaming platform that provides creators with a streamlined process of getting their music released.

Branti gives listeners an enjoyable experience by providing seamless streaming of music. The music uploaded by artists quickly becomes available on the platform for other users/artists to listen to.

Anyone can become an artist on Branti and can publish their music on this platform for other users/artists to enjoy.

Branti serves also as the showcase of challenges we faced during the development of a full stack web application.

Simple and easy to use interface that is well documented to be used by anyone. The User Interface primitives used in Branti are inspired from well known design systems.

Extensible so that it can later be ported to other devices. The backend of Branti, built using Ruby on Rails, can be easily converted to a RestAPI so that it can be consumed by clients from various platforms (Mobile, TVs, and embedded systems).

Introduction

Branti is a music streaming platform that provides creators with a streamlined process of getting their music released while also giving listeners an enjoyable experience.

Branti gives listeners an enjoyable experience by providing seamless streaming of music. The music uploaded by artists quickly becomes available on the platform for other users/artists to listen to.

On Branti, anyone can publish their music on this platform for other users/artists to enjoy.

Current music platforms are not free, those that are free, show ads on their platform that degrades the user experience. **Branti** on the other hand, tries to be a completely free and independent music platform where artists get to own their content.

The simple and easy to use interface of Branti is well documented so that it can also be used by beginners. The User Interface primitives used in Branti are inspired from well known design systems like Material design (<https://material.io/design>).

On the technical side of things, this project is also built to experience the real life software development problems that one can face while developing a full stack web application. Primarily, we dealt with database modeling, User Interface development, and Software Security.

Problem Definition

The goal is to build a music streaming platform that provides creators with a streamlined process of getting their music released while also giving listeners an enjoyable experience.

The app should give listeners an enjoyable experience by providing seamless streaming of music.

The music uploaded by artists should quickly become available on the platform for others to listen to.

Build a music player application with the following features:

- Anyone should be able to register as an Artist.
- Registration process should be easy and should not involve too many steps.
- Users should be able to browse through the existing library of music.
- Users should be able to stream music content.
- Users should be able to play/pause, mute/unmute, and adjust the volume of the music that is currently playing.
- Artists should be able to upload their music content.
- Artists should be able to group their music content in Albums.
- Users/Artists should be able to create public/private playlists.
- Public playlists should be accessible by any other user/artist on the platform.
- Static genres of music to organize songs globally.
- Genres should only be created by special users (admins).

Objectives

The objective of this project is to build a platform for everyone where they'll be able to publish their music content and listen to published music content of other artists.

Users with the goal of just content consumption should also get an enjoyable experience out of our platform.

The platform will have organized sections like **public playlists**, **genres**, and **albums** that'll help the user to easily find the content they love.

Users who want to become Artists will have to go through a simple authentication process. This authentication process helps us to track and better serve their content on our platform.

To become an artist, a user will be given multiple options to sign up that'll include basic **email/password** signup and **google oauth** based login.

The published content on this platform will be marked for review by special users before it becomes available for listening by end users.

List of Figures

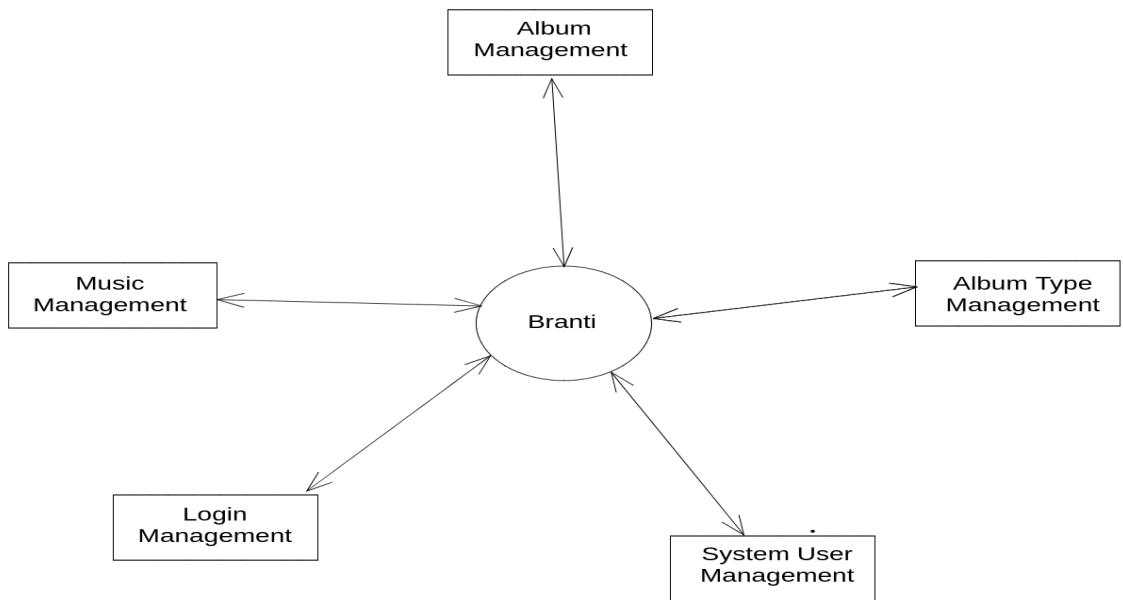


fig 1: Zero level DFD

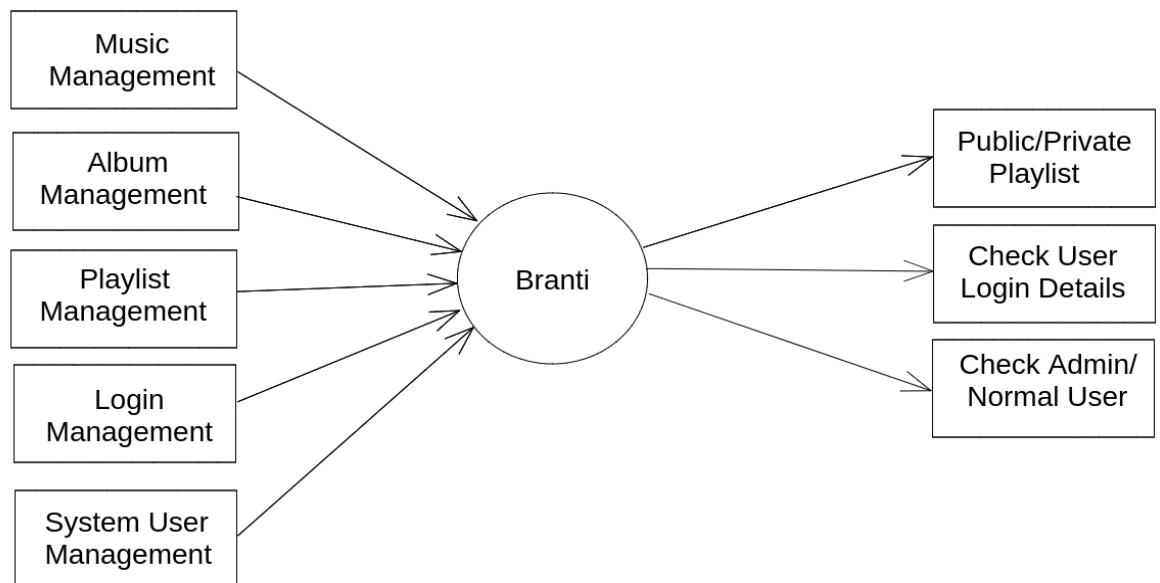


fig 2: First level DFD

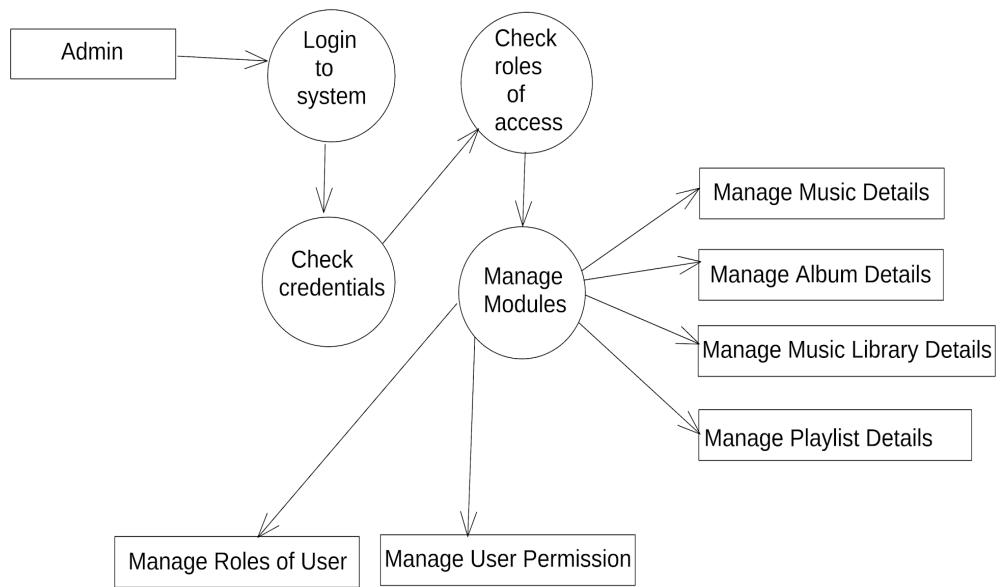


fig 3: Second level DFD

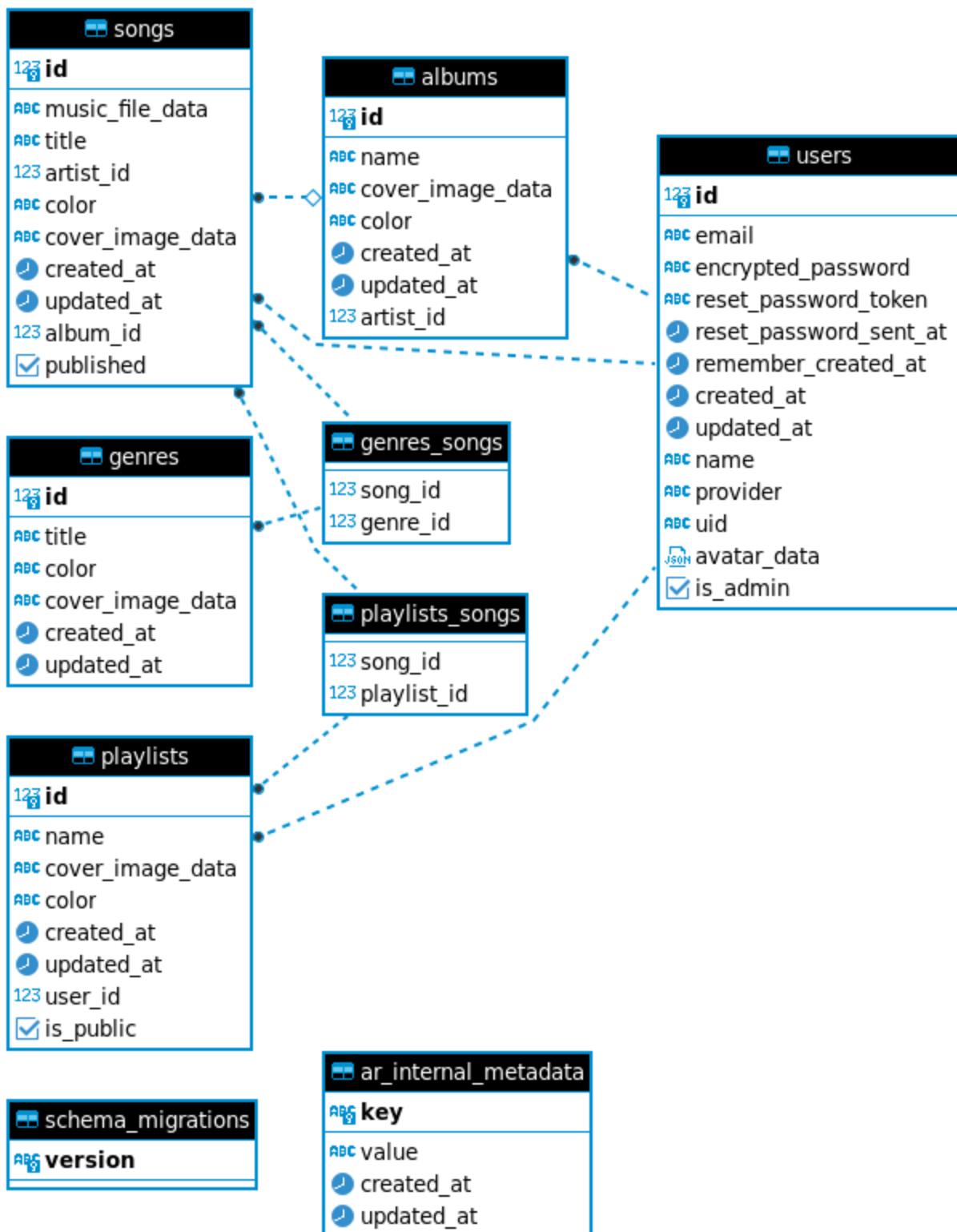


fig 4: ER Diagram

Methodology Used

➤ HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using the markup language.

HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that defines the structure of web pages.

HTML is the language in which most websites are written. HTML is used to create pages and make them functional.

➤ HTML Basic Format Page Structure:-

- **<DOCTYPE html>** – A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. It is not an element or tag. The doctype declaration is not case-sensitive.
- **<html>** – This tag is used to define the root element of an HTML document. This tag tells the browser that it is an HTML document. It is the second outer container element that contains all other elements within it.
- **<head>** – This tag is used to define the head portion of the HTML document that contains information related to the document. Elements within the head tag are not visible on the front-end of a webpage.
- **<body>** – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.

➤ CSS

CSS (Cascading Style Sheets) is a stylesheet language used to design the webpage to make it attractive.

The reason for using this is to simplify the process of making web pages presentable. It allows us to apply styles to web pages. More importantly, it enables us to do this independent of the HTML that makes up each web page.

CSS3 is the current version of Cascading Style Sheet.

➤ Why use CSS?

- **Base for web development:** HTML and CSS is the basic skill that every web developer should know. It is the basic skill that is required for building a website.
- **Makes your website look attractive:** A website that's dull and plain will not attract the user most probably, so adding some style would surely make your website presentable to the user.
- **Makes the design come live:** A web developer is responsible for making the design given to him as a live product. It is used for styling to develop the design of the website.
- **Increases user experience of the website:** A website with a simple yet beautiful UI would help the users to go through the website easily. It is used to make the user interface better.
- **More career opportunities:** Since CSS is a basic requirement while learning Web Development, therefor there are abundant career opportunities for it. As a freelancer, you can land up to many projects.

➤ JavaScript

JavaScript is the world's most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages.

It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments.

Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.

➤ Why Use JavaScript?

- No need of compilers: Since JavaScript is an interpreted language, therefore it does not need any compiler for compilations.
- Used both Client and Server-side: Earlier JavaScript was used to build client-side applications only, but with the evolution of its frameworks namely Node.js and Express.js, it is now widely used for building server-side applications too.
- Helps to build a complete solution: As we saw, JavaScript is widely used in both client and server-side applications, therefore it helps us to build an end-to-end solution to a given problem.
- Used everywhere: JavaScript is so loved because it can be used anywhere. It can be used to develop websites, games or mobile apps, etc.
- Huge community support: JavaScript has a huge community of users and mentors who love this language and take its legacy forward.

➤ Ruby on Rails

Ruby on Rails or also known as rails is a server-side web application development framework that is written in the Ruby programming language, and it is developed by David Heinemeier Hansson under the MIT License.

It supports MVC(model-view-controller) architecture that provides a default structure for database, web pages, and web services, it also uses web standards like JSON or XML for transfer data and HTML, CSS, and JavaScript for the user interface.

It allows us to write less code than other languages and frameworks. It includes everything needed to create database-backed web applications according to the MVC pattern.

➤ Why Ruby on Rails?

- Ruby on Rails is best used for creating online stores with decent and sophisticated browsing and purchasing options.
- Ruby on Rails can also be used for creating efficient stock marketing platforms.
- It is also beneficial to use Ruby on Rails to create Social Networking sites.
- Ruby on Rails can be a good choice while creating Non-standard complex projects.
- Due to Ruby on Rails' easy building features, using it for creating SaaS solutions is the optimum choice.



Compress the complexity of modern web apps.

Learn just what you need to get started, then keep leveling up as you go. **Ruby on Rails scales from HELLO WORLD to IPO.**

Rails 7.0.2.4 — released April 26, 2022

The screenshot shows a Mac OS X desktop environment. In the foreground, a code editor window displays the file `comments_controller.rb` with the following content:

```
1 class CommentsController < ApplicationController
2   before_action :set_post
3
4   def create
5     comment = @post.comments.create! params.required(:comment).permit(:content)
6     CommentsMailer.submitted(comment).deliver_later
7     redirect_to @post
8   end
9
10  private
11  def set_post
12    @post = Post.find(params[:post_id])
13  end
14end
```

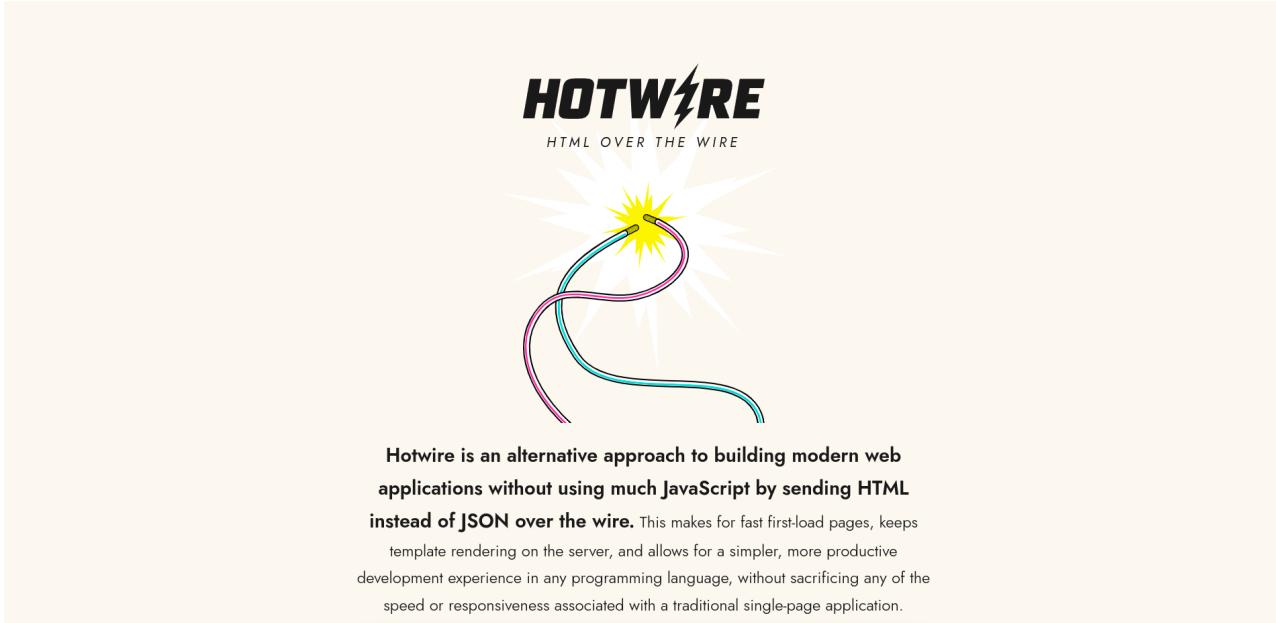
In the background, a file browser window titled "demo" shows the directory structure of the application:

- models
- views
 - active_storage
 - comments
 - _comment.html.erb
 - _new.html.erb
 - comments_mailer
 - submitted.html.erb
 - submitted.text.erb

➤ Hotwire

Hotwire is an alternative approach to building modern web applications without using much JavaScript by sending HTML instead of JSON over the wire.

It is made for fast first-load pages, keeps template rendering on the server, and allows for a simpler, more productive development experience in any programming language, without sacrificing any of the speed or responsiveness associated with a traditional single-page application.



Hotwire is an alternative approach to building modern web applications without using much JavaScript by sending HTML instead of JSON over the wire. This makes for fast first-load pages, keeps template rendering on the server, and allows for a simpler, more productive development experience in any programming language, without sacrificing any of the speed or responsiveness associated with a traditional single-page application.

➤ Turbo

The heart of Hotwire is Turbo. A set of complementary techniques for speeding up page changes and form submissions, dividing complex pages into components, and stream partial page updates over WebSocket. All without writing any JavaScript at all. And designed from the start to integrate perfectly with native hybrid applications for iOS and Android.



➤ Stimulus

While Turbo usually takes care of at least 80% of the interactivity that traditionally would have required JavaScript, there are still cases where a dash of custom code is required.

Stimulus makes this easy with a HTML-centric approach to state and wiring.



Results and Screenshots

This platform is built for everyone where they're able to publish their music content and listen to published music content of other artists.

Users with the goal of just content consumption will get an enjoyable experience out of our platform.

The platform has organized sections like **public playlists**, **genres**, and **albums** that help the user to easily find the content they love.

Users who want to become Artists have to go through a simple authentication process. This authentication process helps us to track and better serve their content on our platform.

To become an artist, a user has been given multiple options to sign up that includes basic **email/password** signup and **google oauth** based login.

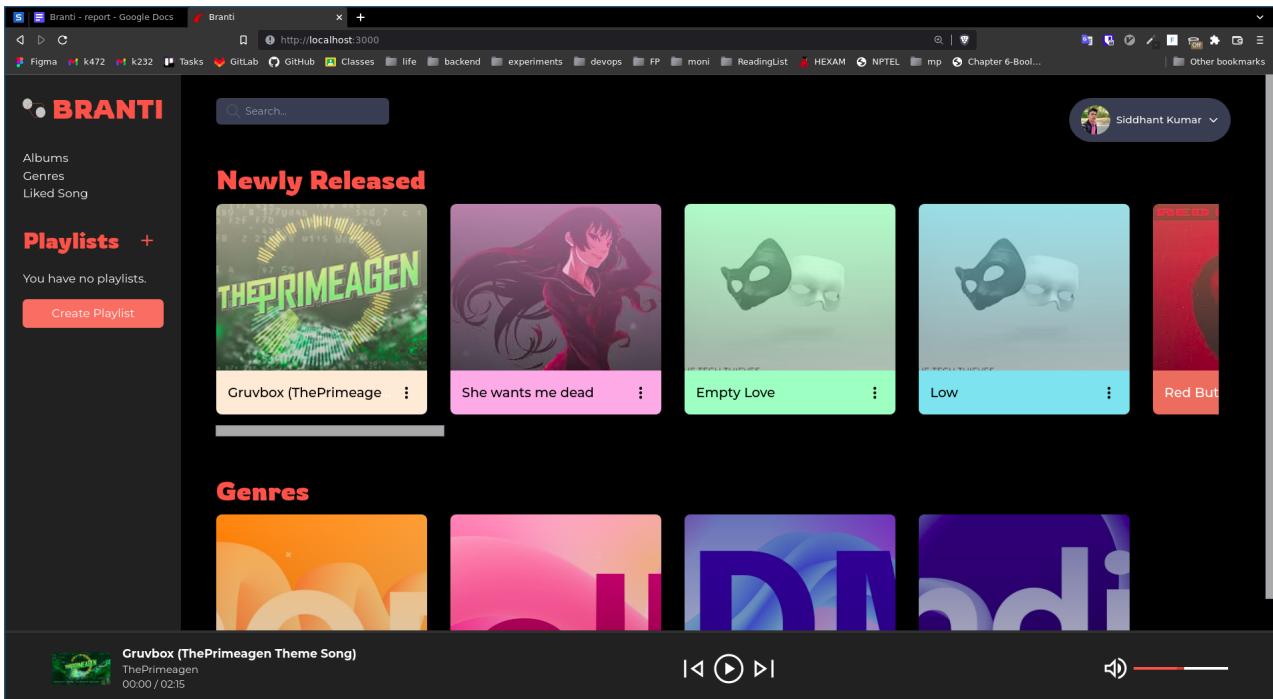
The published content on this platform will be marked for review by special users before it becomes available for listening by end users.

Special users (admins) can only be created by existing admins.

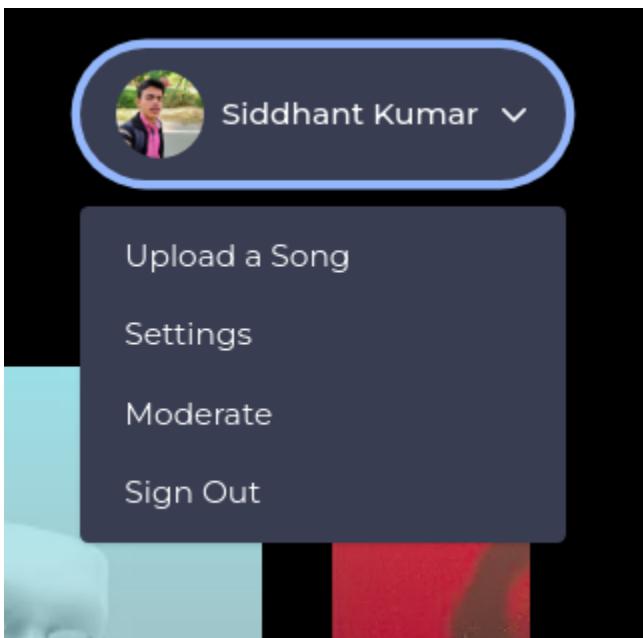
There is one default admin (project owner).

Screenshot

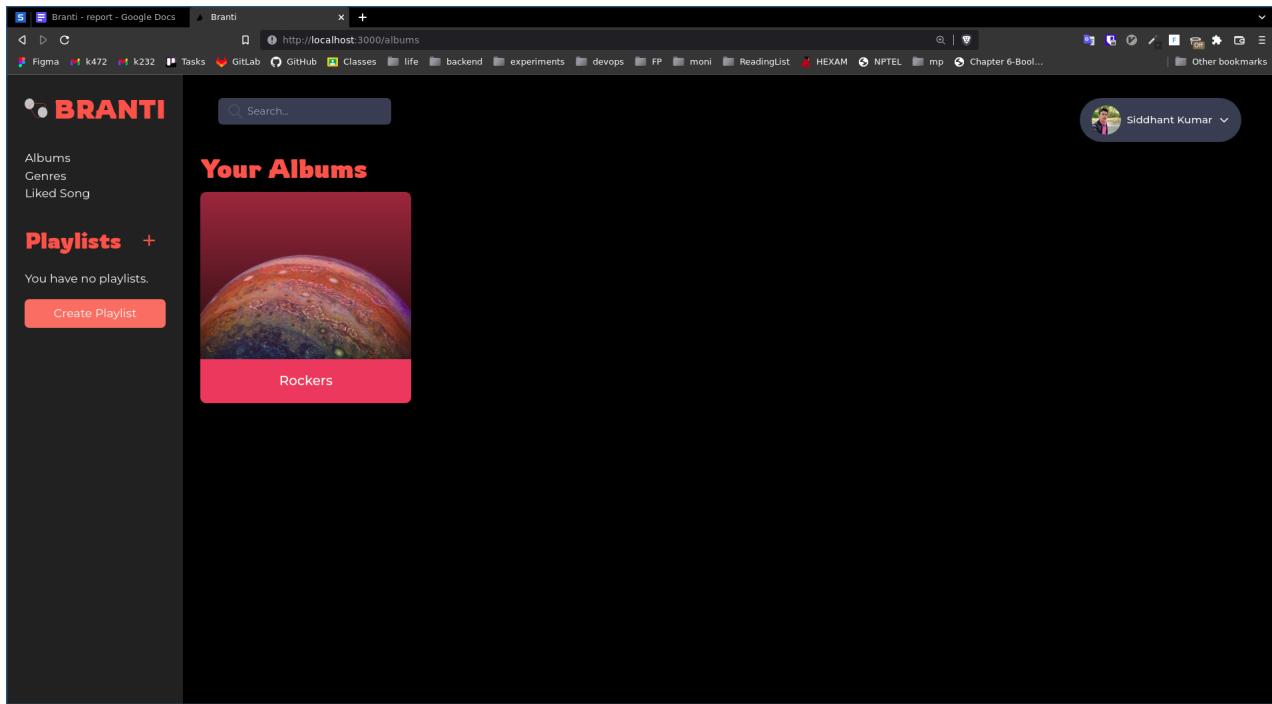
Home Page



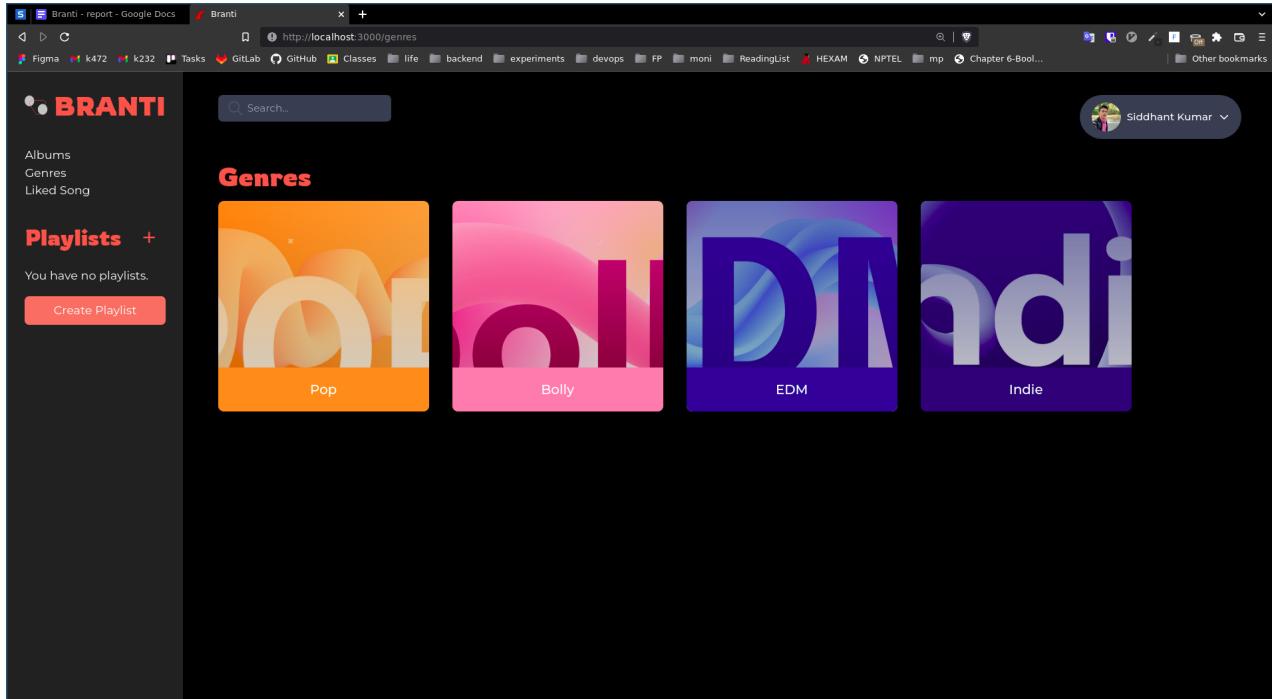
Admin Options



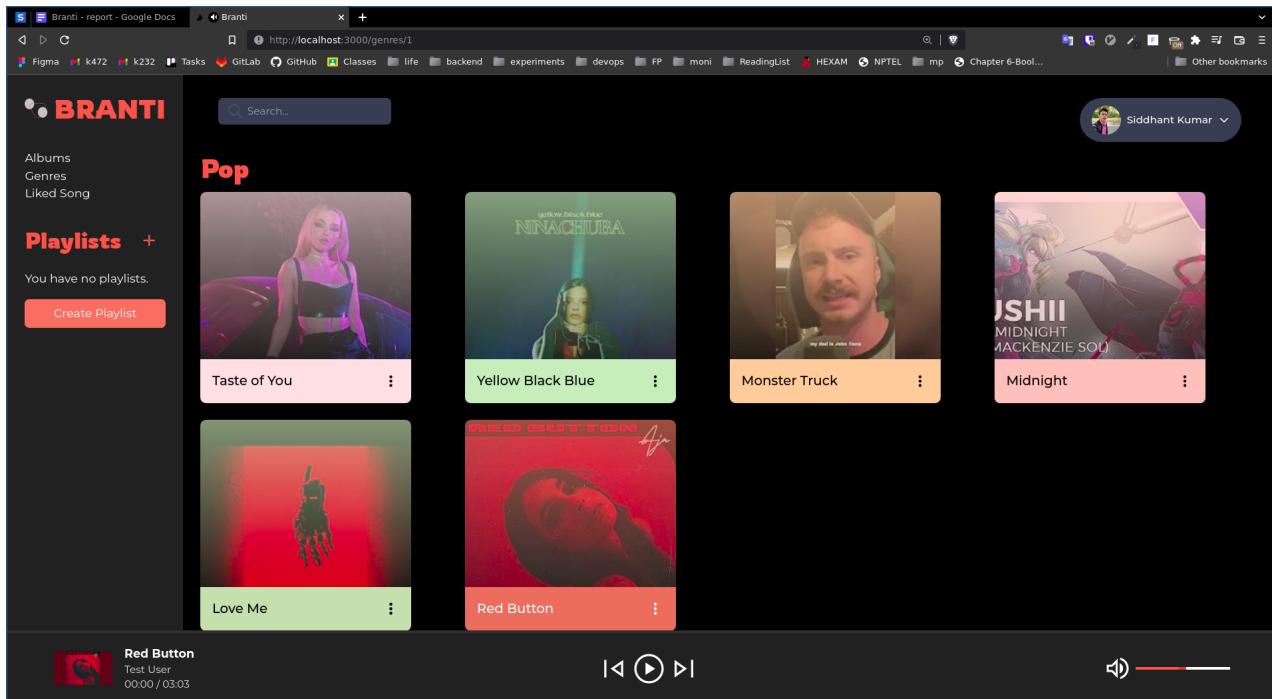
Albums Page



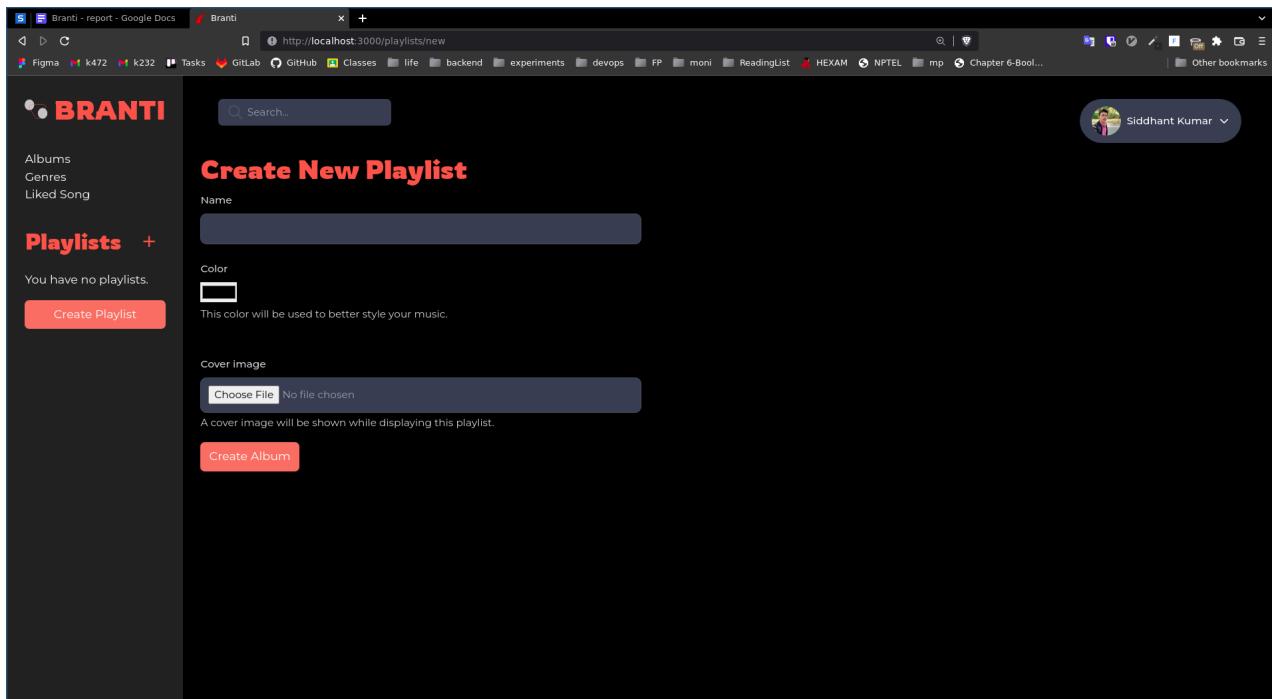
Genres Page



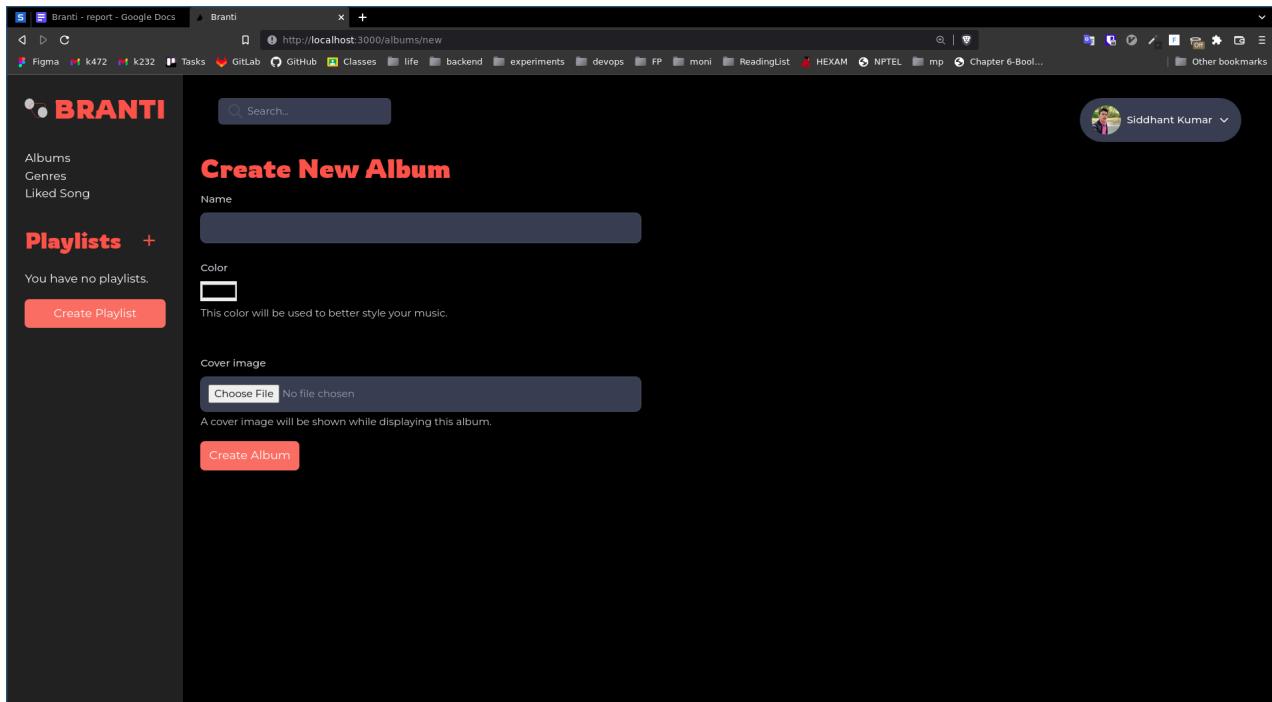
Songs categorised by genre



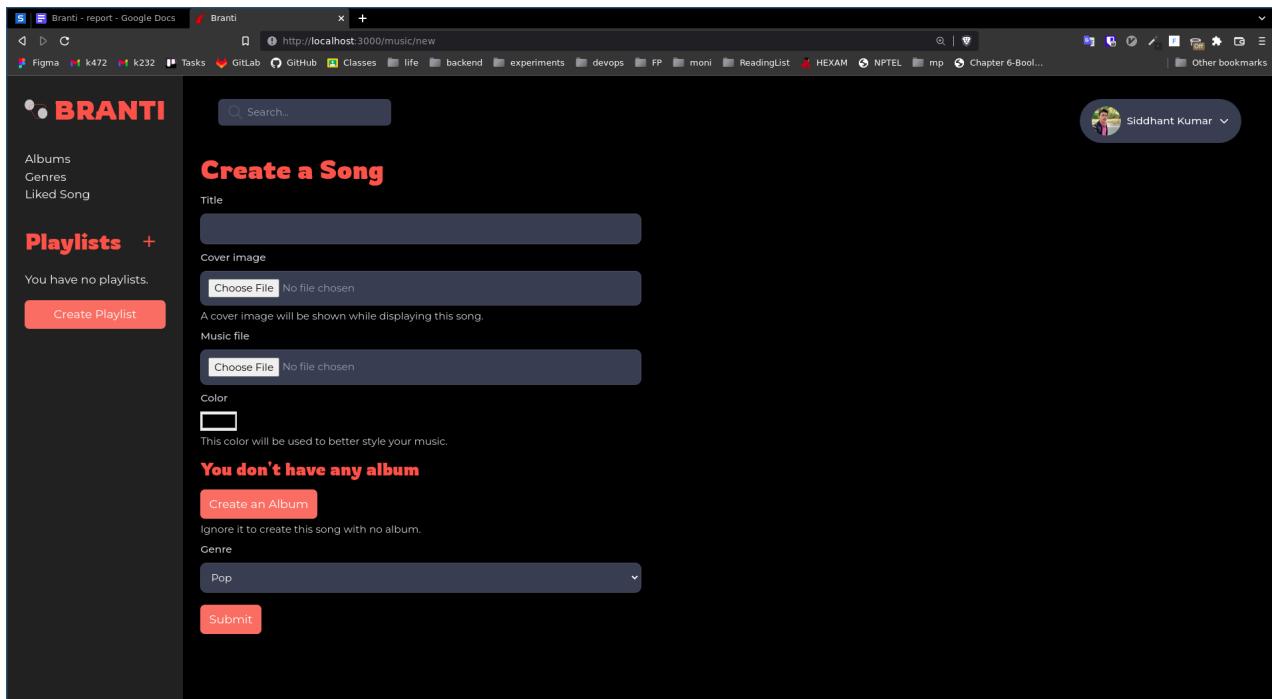
Create Playlist Page



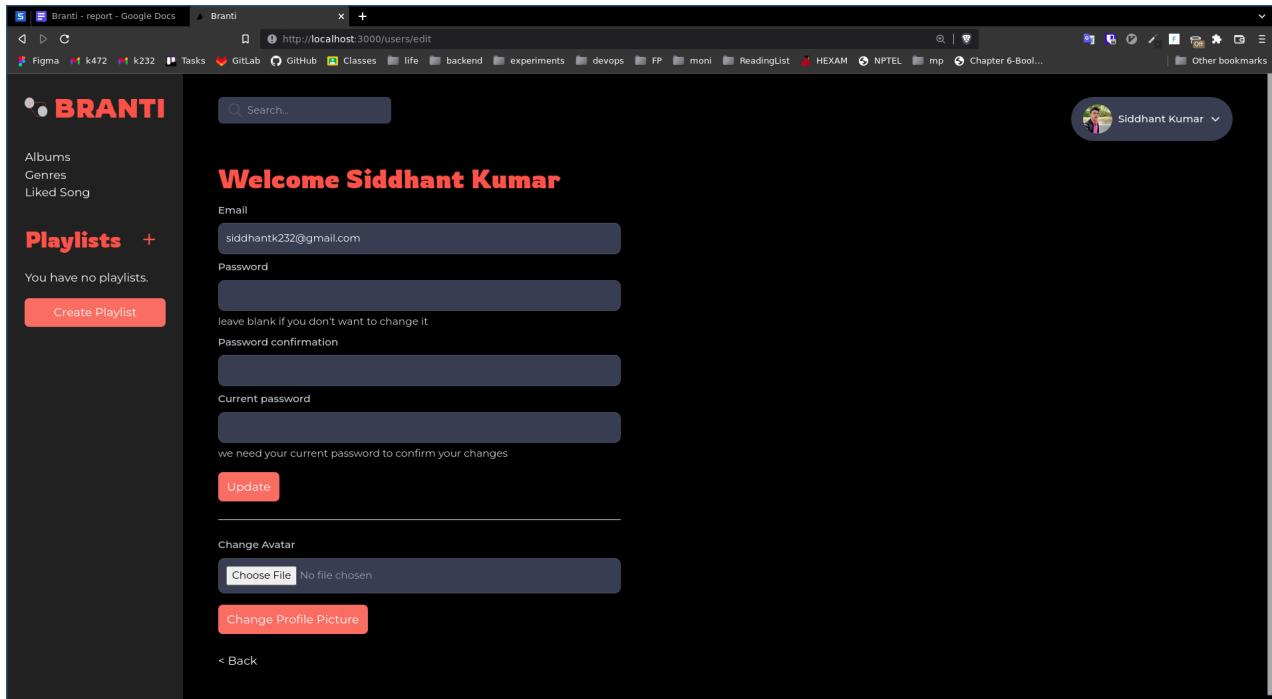
Create Album Page



Create and Upload Song page

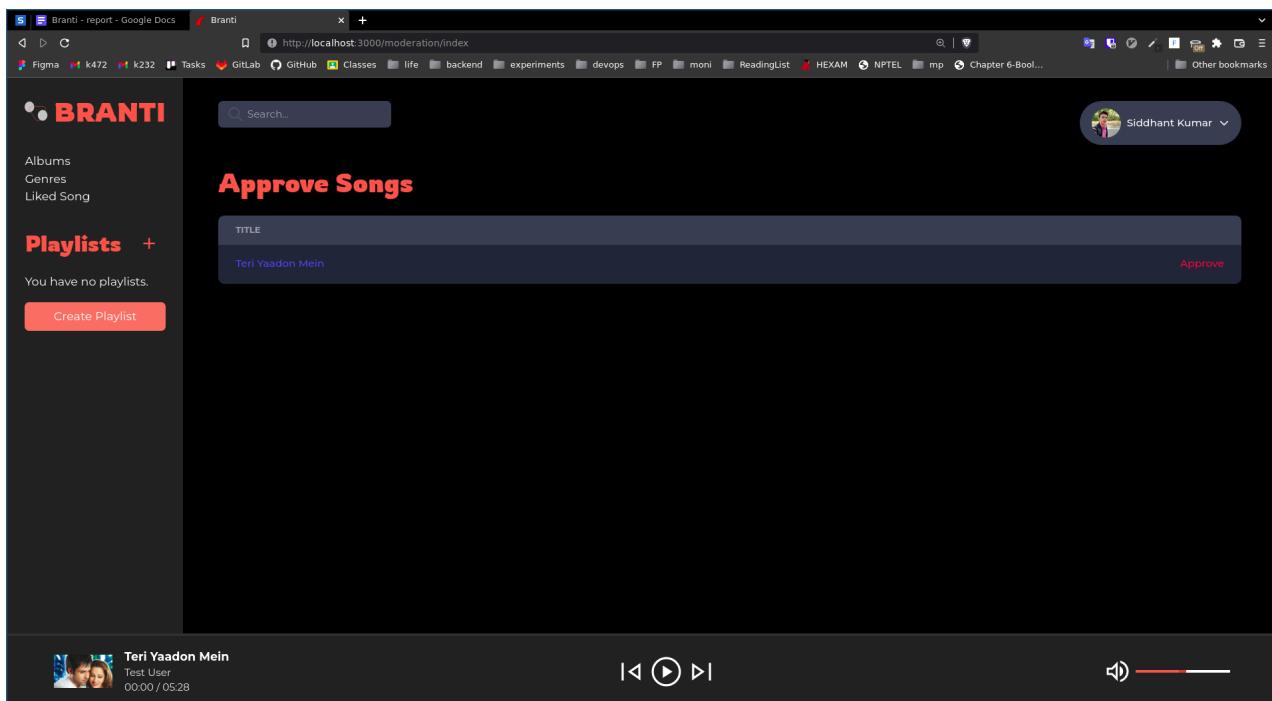


User Settings Page



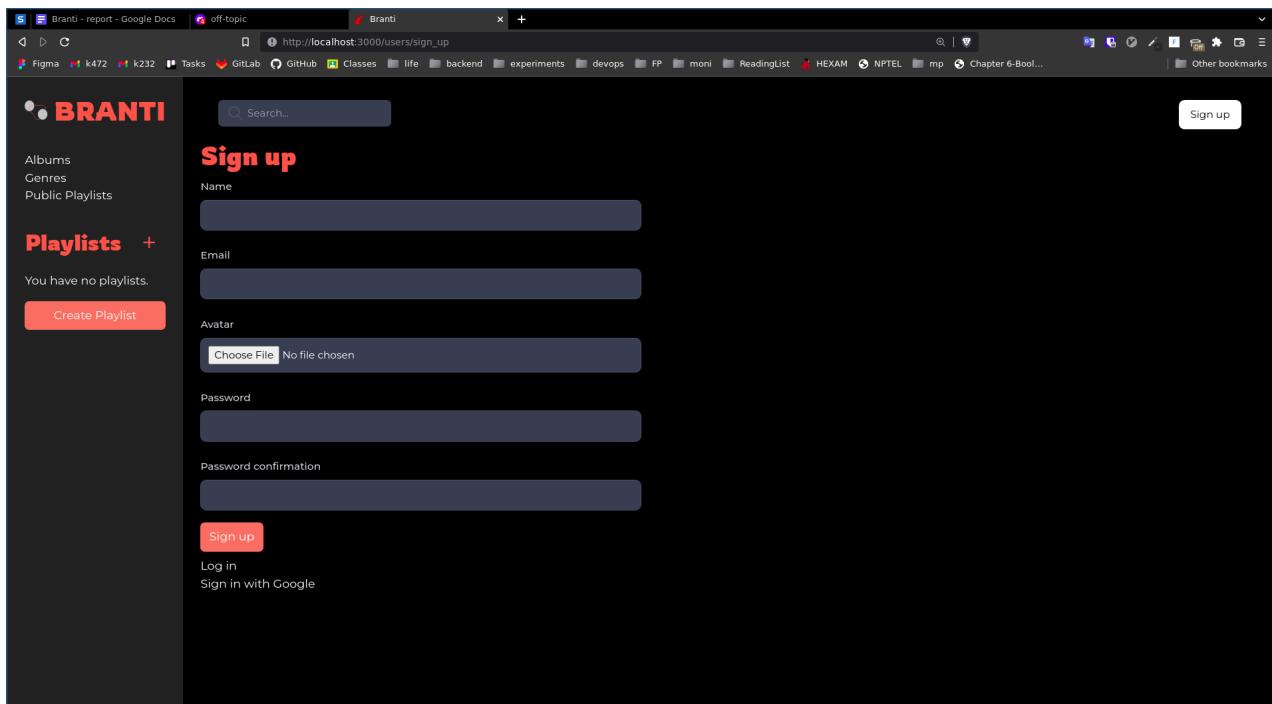
The screenshot shows a dark-themed user settings page for a service named 'BRANTI'. On the left sidebar, there are links for 'Albums', 'Genres', 'Liked Song', and a 'Playlists' section which indicates 'You have no playlists.' Below this is a 'Create Playlist' button. The main content area has a header 'Welcome Siddhant Kumar'. It contains several input fields: 'Email' (siddhantk232@gmail.com), 'Password' (empty), 'Password confirmation' (empty), and 'Current password' (empty). A note below the password fields says 'we need your current password to confirm your changes'. There is a red 'Update' button. Below this is a 'Change Avatar' section with a 'Choose File' button (No file chosen) and a 'Change Profile Picture' button. At the bottom left is a 'Back' link.

Moderation Page

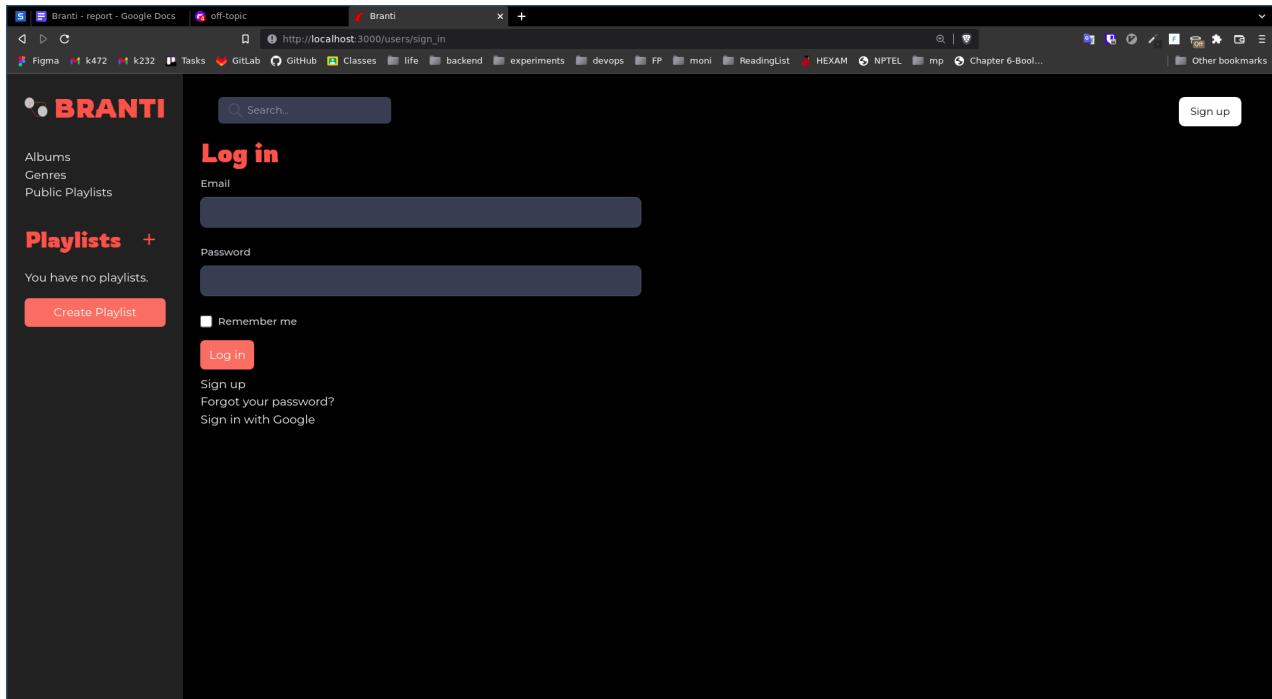


The screenshot shows a dark-themed moderation page for songs. The left sidebar is identical to the User Settings page. The main content area has a header 'Approve Songs'. It displays a single song entry with the title 'Teri Yaadon Mein'. To the right of the title is an 'Approve' button. At the bottom left, there is a thumbnail for the song 'Teri Yaadon Mein' by 'Test User', showing a duration of '00:00 / 05:28'. At the bottom center is a media control bar with icons for play, previous, next, and volume.

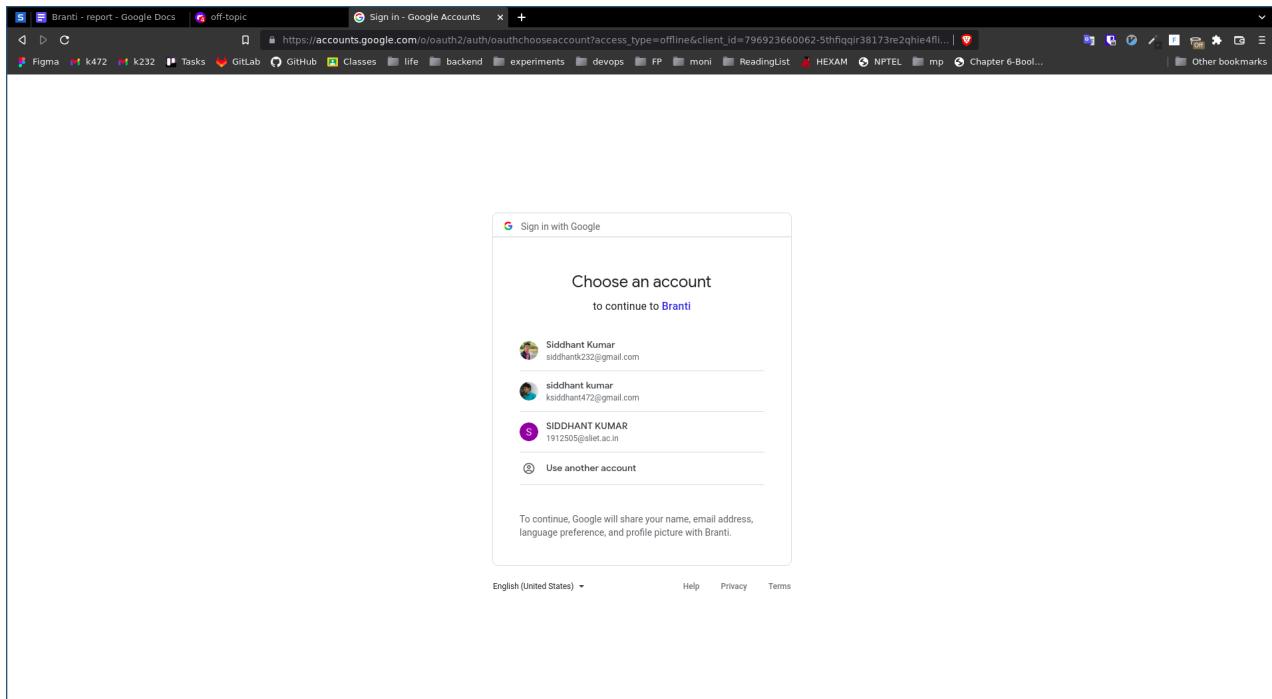
Sign Up Page



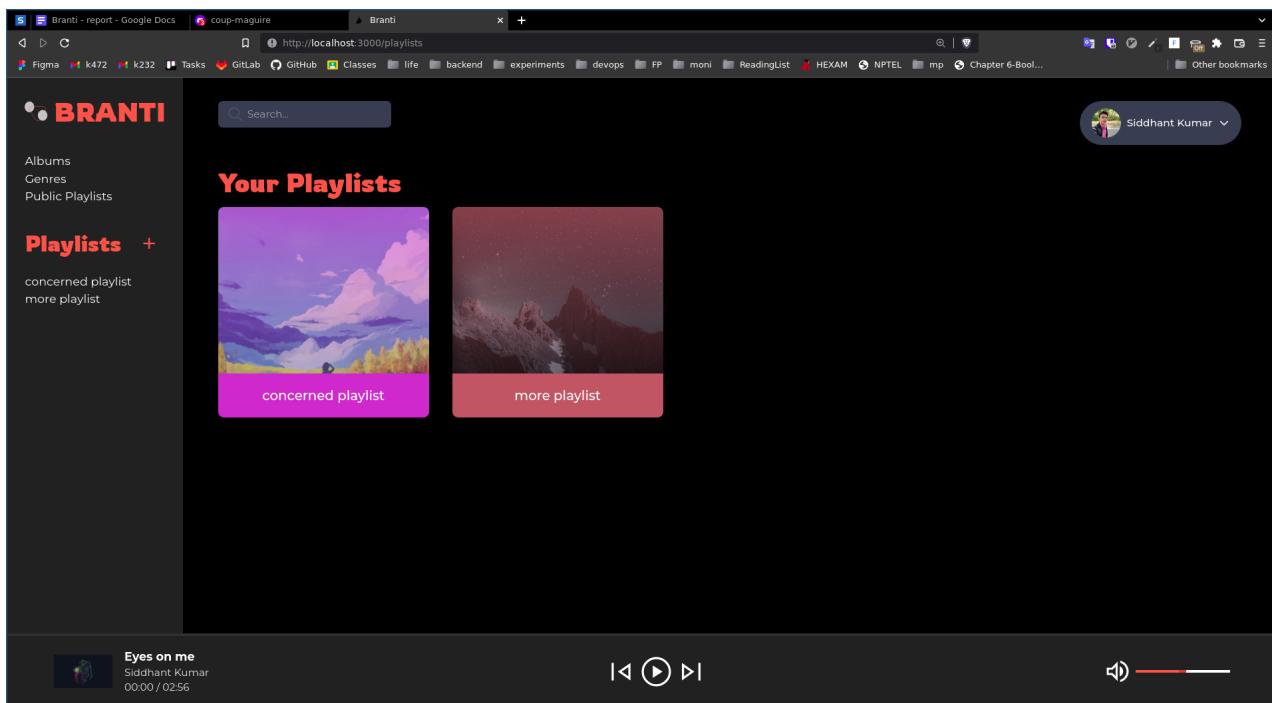
Log in Page



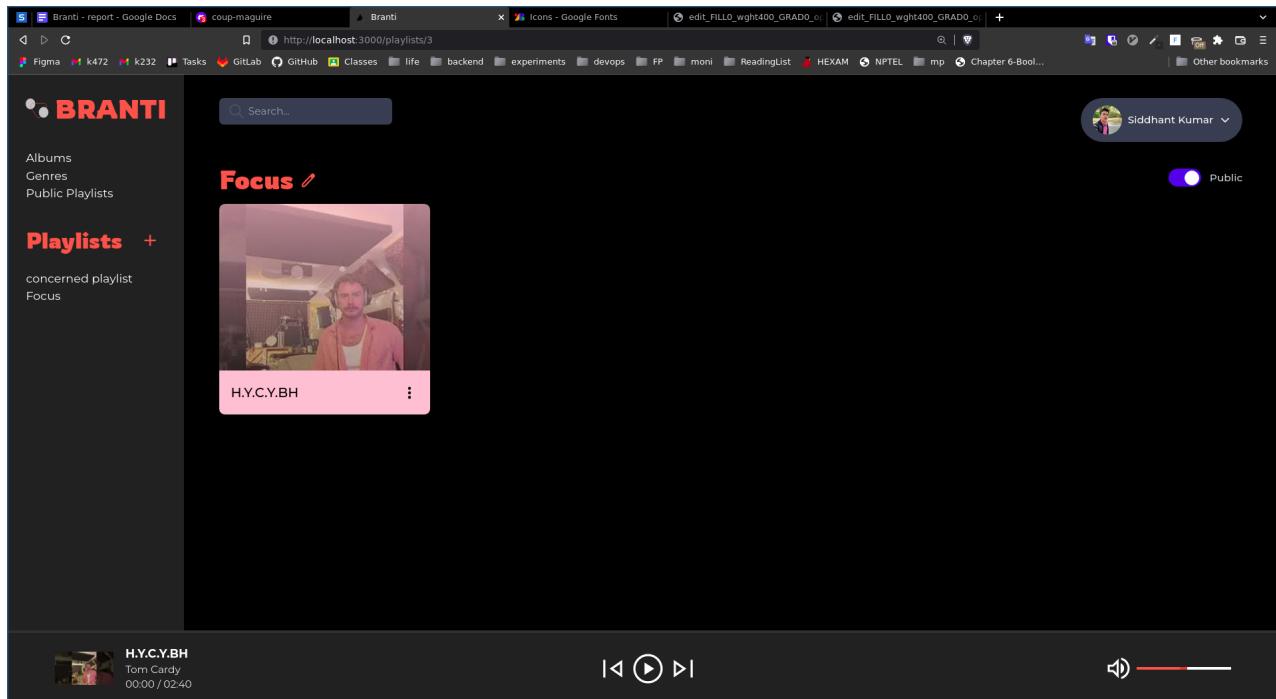
Google Sign in (OAuth) Page



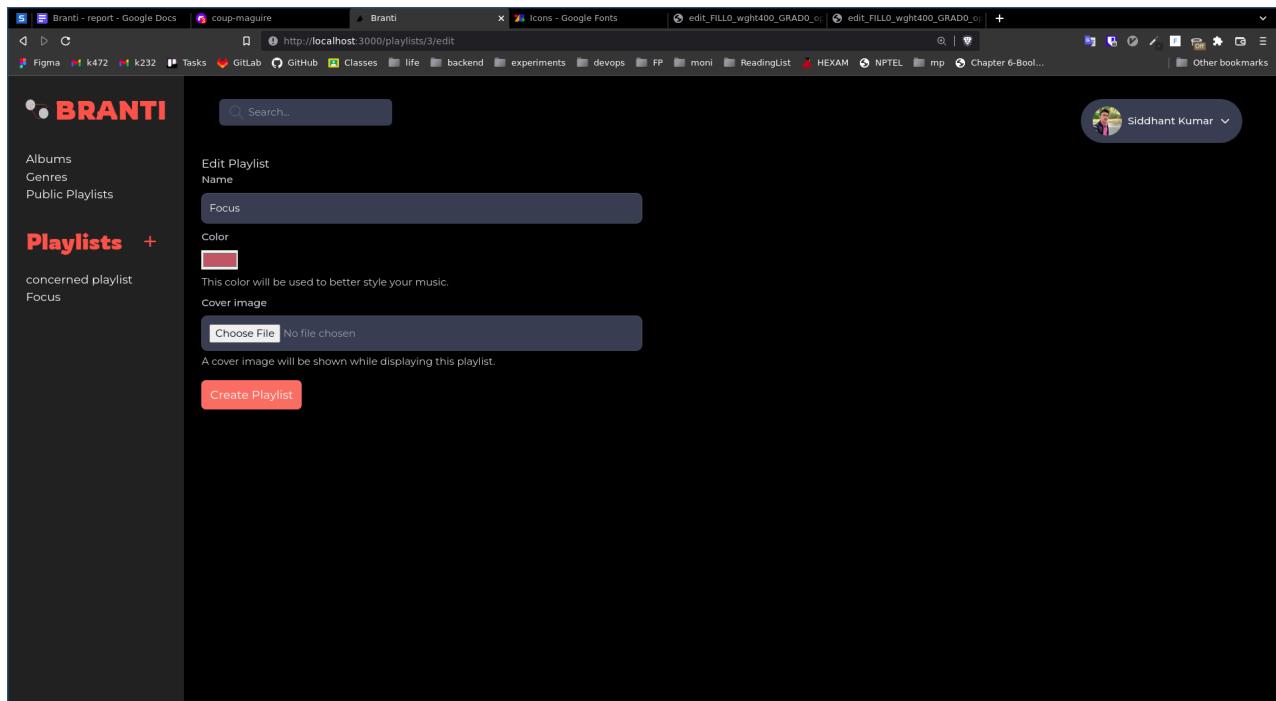
Playlists Page



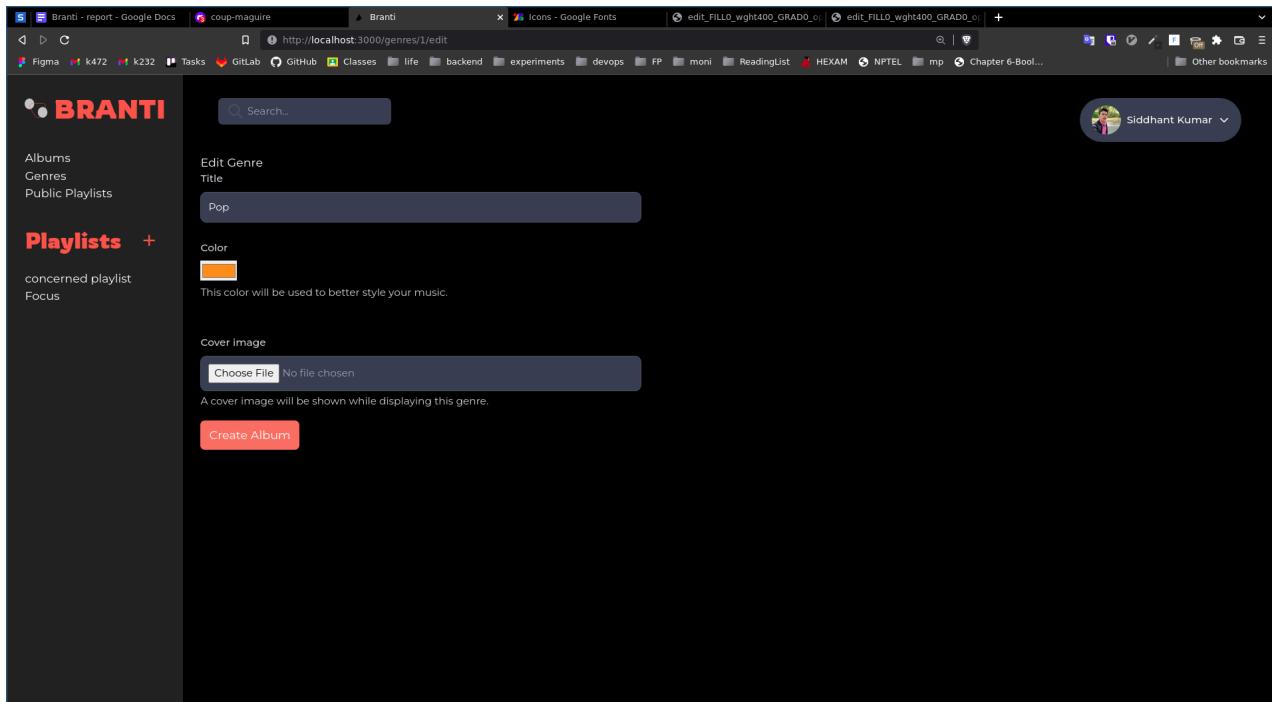
Songs Categorised by Playlists



Edit Playlists Page



Edit Genres Page



Search Results Page

