DESIGN AND DEVELOPMENT OF MUSIC STREAMING APP

A

MINOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 24

ADITYA PRAKASH 0187CS171012

Under the guidance of

RAJESH RAI

(Assistant Professor)

Apr-2020

Department of COMPUTER SCIENCE & ENGINEERING
Sagar Institute of Science & Technology (SISTec)
Bhopal (M.P.)
Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

Sagar Institute of Science & Technology (SISTec), Bhopal
Department of COMPUTER SCIENCE & ENGINEERING
Bhopal (M.P.)

Apr-2020

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project-II Report entitled **Design and Development of Music Streaming App,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science & Engineering*** and submitted to the Department of Computer Science & Engineering, *Sagar Institute of Science & Technology (SISTec)***,** Bhopal (M.P.) is an authentic record of my own work carried out during the period from Jan-2020 to Apr-2020 under the supervision of **Rajesh Rai Sir(Assistant Professor).**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Signature

Aditya Prakash
0187CS171012

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Project Guide	***HOD***	***Principle***
(Profe.Rajesh Rai)	***(Profe.Ujjawal Nigam)***	***(*Dr. Keshavendra Choudhary*)***

ABSTRACT

Project Topic:- Design and Development of Music Streaming App using Flutter and Dart.

Mobile applications have made life much easier and smoother than before. It has improved the general human lifestyle in many ways from ordering food to buying dresses and what not!. The custom Android app development has started playing a great role when it comes to music streaming as the development of music streaming applications has become much more smooth, simple and advantageous than before.Mobile apps have been successful in changing the mindset of people about music and music listener's consumer market.

The project will aim to do this by critically analysing the streaming apps for Musics and also how the another music apps like Spotify, Jio Saavn etc works. The references used in this report consist of academic findings alongside website articles which offer insights into the music streaming services, its associated artists and the music industry surrounding them.

NAME :- ADITYA PRAKASH
ENROLLMENT NO.:- 0187CS171012
GROUP NO.:- 24

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide **Prof. Rajesh Rai** and our Project Coordinator **Prof. Gajendra Gaekwad**, *Department of Computer Science And Engineering, SISTec Gandhi Nagar Bhopal*, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and cooperative behaviour are sincerely acknowledged.

We would like to express our sincere thanks to **Dr. Keshavendra Choudhary**, *Principal, SISTec, Gandhi Nagar, Bhopal* for giving us an opportunity to undertake this project. We also wish to express our gratitude to **Prof. Ujjawal Nigam**, *Head, Department of Computer Science and Engineering*, for his kind hearted support.

At the end, we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

NAME	**ENROLLMENT NO.**	**SIGNATURE**
ADITYA PRAKASH	0187CS171012	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
SDLC	Software Development Life Cycle
NoSQL	Non-Relational Structured Query Language
JSON	JavaScript Object Notation
UML	Unified Modeling Language
YAML	**YAML** Ain't Markup Language

CHAPTER 1

INTRODUCTION

CHAPTER-1
INTRODUCTION

Music streaming services have become the most popular method for consumers to listen to music. Streaming services offer consumers unlimited access to large catalogue of music. These services store the music in a server that users can connect to via their laptops and mobile devices. Whilst connected to the internet users can listen to any song they wish, by selecting it on the application from which it can then be played. On many services the consumer can choose to store the music locally, by making it available offline. This means that the music is stored on the consumer’s device within the memory of the streaming service app. The user does not own any of the music in the catalogue, in this way it is similar to how people rent books from a library. Users are also unable to access the individual MP3 files within the catalogue and therefore they cannot make copies for their own use.

Through a monthly subscription whether paid or free, users are able to stream their favourite artists easily on their own devices. The technology of streaming services whether music or video has been growing consistently over the past ten years. Most streaming services allow users to download an application for free from which they are able to access the full service, rather than just using the services main website. Music streaming has grown in popularity due to the fact that many consumers have access to a smart phone with internet on a regular basis. Thus showing that the rise of music streaming services has been dependent on the technology which has been created alongside it. Music streaming has offered the music industry and its artists a new method of sharing and distributing music to consumers across the world.

1.1 ABOUT PROJECT

This project is basically belongs to the android platform for sharing and listen musics online by the process of Streaming. It provides the basic functionality of musics app like spotify, Saavn etc. The module of app is divided into two phases such as client side and server side commonly known as admin of an app. This app is run on two platform at same time Android as well as Ios devices.

1.2 PROJECT OBJECTIVES

- Get Info About What You Hear.
- Enjoy Music Offline.
- Downloads your favourite songs.
- Watch your current listen song on youtube.
- Easy UI for controlling different songs.
- Background Services available.

CHAPTER 2

SOFTWARE & HARDWARE
REQUIREMENTS

CHAPTER-2
SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE REQUIREMENTS

- Good GPU such as Ardeno 640 etc.
- Storage - depending on your use.
- RAM - Minimum 2GB.
- High Speed Server for streaming.
- Server Storage minimum 10GB.

SOFTWARE REQUIREMENTS

- Visual Studio (latest version).
- IntelliJ Studio or Android Studio.
- SDK API > 28.0.
- Flutter SDK
- Dart 2.7 Modules and Libraries.
- Figma Web application tool.
- Rive animation tool.
- POSTMAN Application

CHAPTER 3

PROBLEM DESCRIPTIONS

CHAPTER-3
PROBLEM DESCRIPTIONS

Fluctron music streaming app is provide the collections of songs online and you can listen your favourite songs through streaming and also download it.

The app comprises of various model such as all mixed songs, categorized into albums, separate artists and singers, various podcasts and much more.You can see the song video instantly on same screen of your currently listen songs.This app is built for both Android and ios devices with some system compatibility.The app also includes advertisements for newly release movies song, albums and podcasts etc.

We can also try to achieve the playlists recommendations according to users choice , regularly listen songs.You can also create a playlists and add songs to favourite list.

Here are some Key features of this app:-

There are two modules in this app :-

1) USERS

- Users can get the recent updates regularly.
- They can download the songs and stored it into your mobile storage for offline mode.
- Add new playlists
- Add songs to Favourite
- Switch between Dark and Light mode.
- Share the Songs ,albums,podcasts with other social media platforms.
- Watch the current listen songs on PIP mode.
- Songs and Playlists recommendations.

2) ADMIN

- Updates the advertisements.
- Adding new release songs, albums and much more.
- Bugs fixings regularly
- Manage users data.

CHAPTER 4

LITERATURE SURVEY

CHAPTER-4
LITERATURE SURVEY

Now a days everything is going online then why not domestic helps also provided by online. To maintain different registers for various activities such as storing & searching data which is tedious and time consuming task.

The Fluctron Music Streaming App totally built in Flutter and Dart, so firstly we have to understand the architecture of Flutter based application and cross-platform services.

FLUTTER ARCHITECTURE

WHY WE USE FLUTTER?

- Be highly productive

1. Develop for iOS and Android from a single codebase.
2. Do more with less code, even on a single OS, with a modern, expressive language and a declarative approach.
3. Prototype and iterate easily

   I. Experiment by changing code and reloading as your app runs (with hot reload).

   II. Fix crashes and continue debugging from where the app left off.

- Create beautiful, highly-customized user experiences

1. Benefit from a rich set of Material Design and Cupertino (iOS-flavor) widgets built using Flutter's own framework.
2. Realize custom, beautiful, brand-driven designs, without the limitations of OEM widget sets.

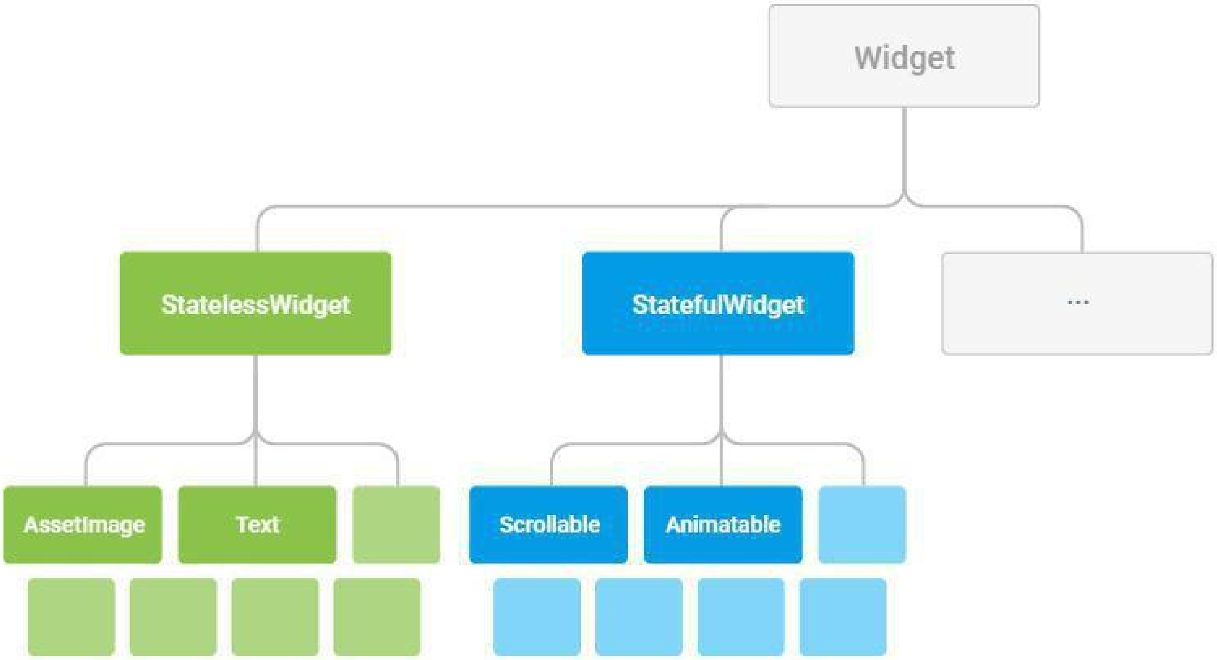
CORE PRINCIPLES

Everything's a widget

Widgets are the basic building blocks of a Flutter app's user interface. Each widget is an immutable declaration of part of the user interface. Unlike other frameworks that separate views, view controllers, layouts, and other properties, Flutter has a consistent, unified object model: the widget.

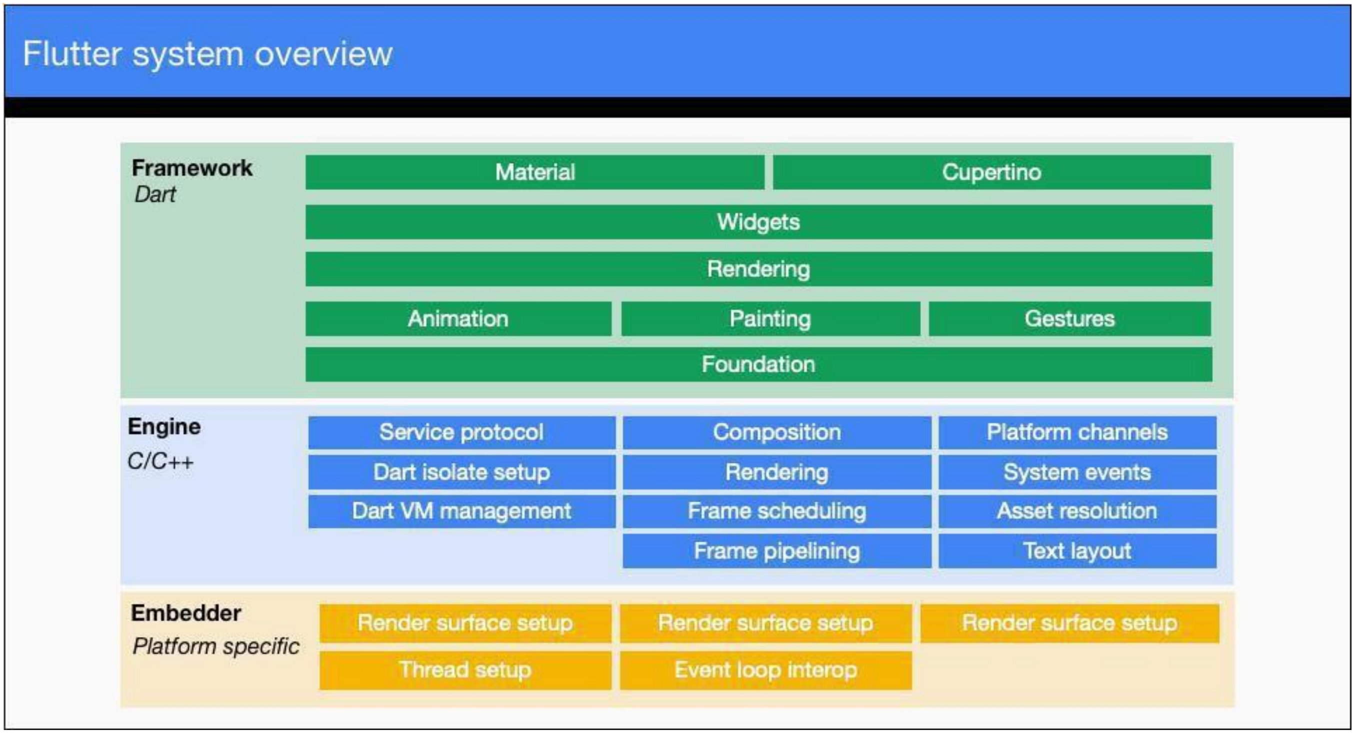
CLASS HEIRARCHY

FIG.1.1



FLUTTER SYSTEM AND ARCHITECTURE

FIG.1.2



Flutter system overview

Framework
Dart

Material | Cupertino
Widgets
Rendering
Animation | Painting | Gestures
Foundation

Engine
C/C++

Service protocol | Composition | Platform channels
Dart isolate setup | Rendering | System events
Dart VM management | Frame scheduling | Asset resolution
Frame pipelining | Text layout

Embedder
Platform specific

Render surface setup | Render surface setup | Render surface setup
Thread setup | Event loop interop

CHAPTER 5

SOFTWARE REQUIREMENTS SPECIFICATIONS

CHAPTER-5
SOFTWARE REQUIREMENTS SPECIFICATIONS

5.1 FUNCTIONAL REQUIREMENTS

5.1.1 CLIENTS

The client-side of the system will be an application with a user interface that is integrated into a music listening application. This application gathers the information from users, investigates some actions of the users, and provides the connection with the server. This application is the client-side interface of the Music Recommender, so it does not include the functionalities of the host music environment such as playing music etc.

- **Requesting recommendations**

The client-side application must allow user to request recommendations manually, and interact with the server to receive recommendations.

- **Evaluating songs**

It must be able to evaluate songs and send appropriate information to the server.

- **Investigating user**

The system will follow the interaction of users with music items., and send these obtain back to the server.

- **Display recommendations**

The application must display the recommendations that are obtained from the server to the user in a proper way by providing a GUI.

5.1.2 SERVER

The server-side system will hold the entire data in a graph database, and must include all functionality to perform operations on this database, receive requests from the clients, evaluate, create and send recommendations etc.

- **Handle recommendation requests**

The server application shall obtain and handle requests for recommendations.

- **Store evaluations**

The server application shall receive and store music evaluations.

- **Data storing**

The server application shall be able to store the newly retrieved data to the database.

- **Recommend using content based filtering**

The server application shall be capable of producing recommendations by interpreting the content and evaluations by actual user.

- **Recommend using collaborative filtering**

The server application shall be capable of producing recommendations by interpreting the evaluations given by actual user and other similar users.

- **Recommend using contextual collaborative filtering**

The server application shall be capable of producing recommendations by interpreting contextual information given by the users and evaluations given by the actual user and other similar users.

5.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements of the system are explained below as performance requirements and design constraints.

5.2.1 Performance requirements

- **Accuracy**

Since we will give the priority to the accuracy of the software, the performance of the Music Recommender will be based on its accuracy on recommendations.

- **Failure handling**

System components may fail independently of others. Therefore, system components must be built so they can handle failure of other components they depend on.

- **Openness**

The system should be extensible to guarantee that it is useful for a reasonable period of time.

- **Security**

Sensitive information should be kept in safe.

5.2.2 Design constraints

- **Languages**

The product should be integratable with any music websites. Also we will handle with a lot of parameter and object.

- **Hardware Constraints**

The system will be integrated with a website. To use recommendation system, user should enter from a personal computer, mobile device with internet connection, tablet etc.

- **Software System Attributes**

Usability

The software will be embedded in a website. It should be scalable designed to be easily adopted by a system.

Reliability

The system should have accurate results and fast responses to user’s changing habits.

Security

User profile information will be used, so data security is one of the most important concern of the system.

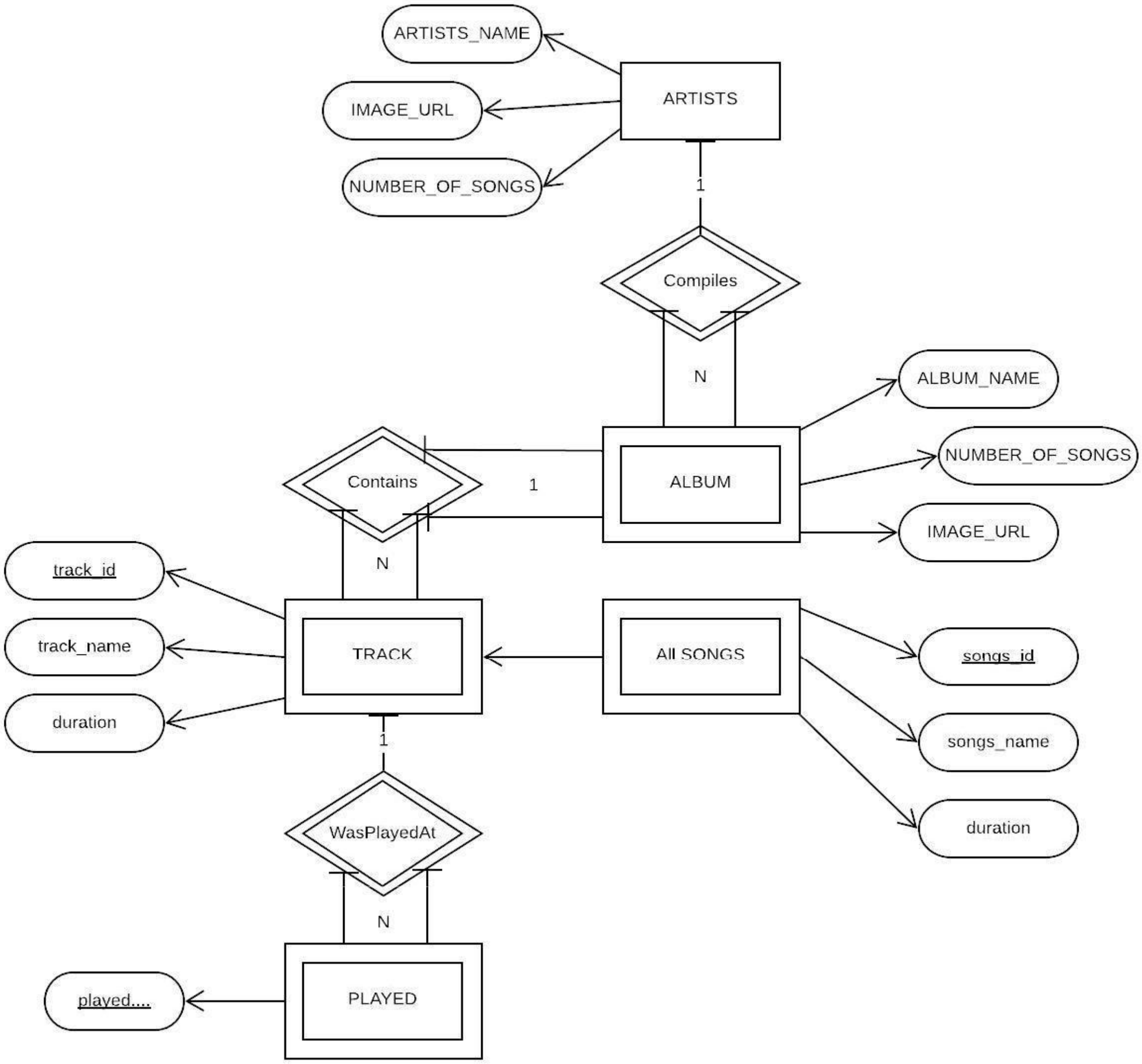
CHAPTER 6

SOFTWARE DESIGN

CHAPTER-6
SOFTWARE DESIGN

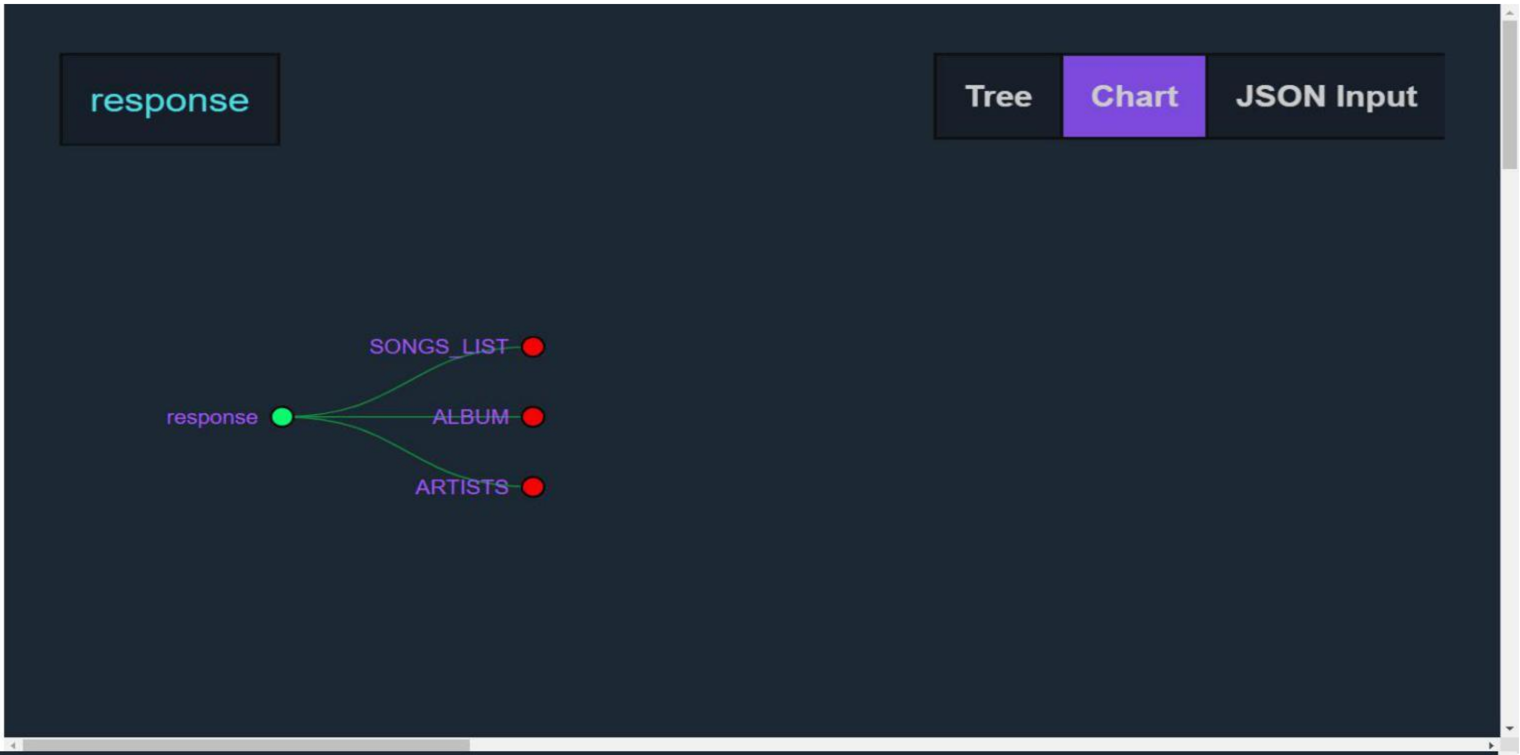
6.1 ENTITY-RELATIONSHIP DIAGRAM

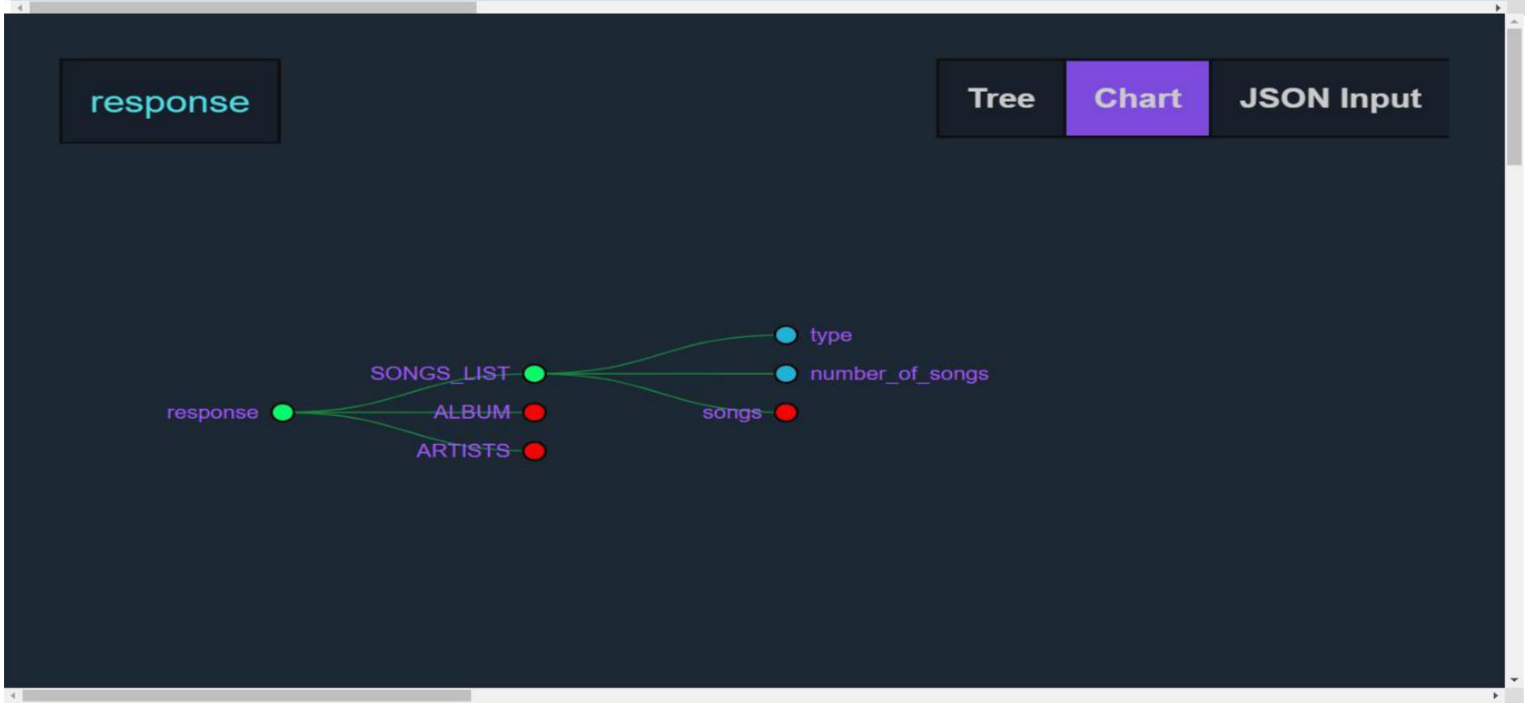
FIG.2.1

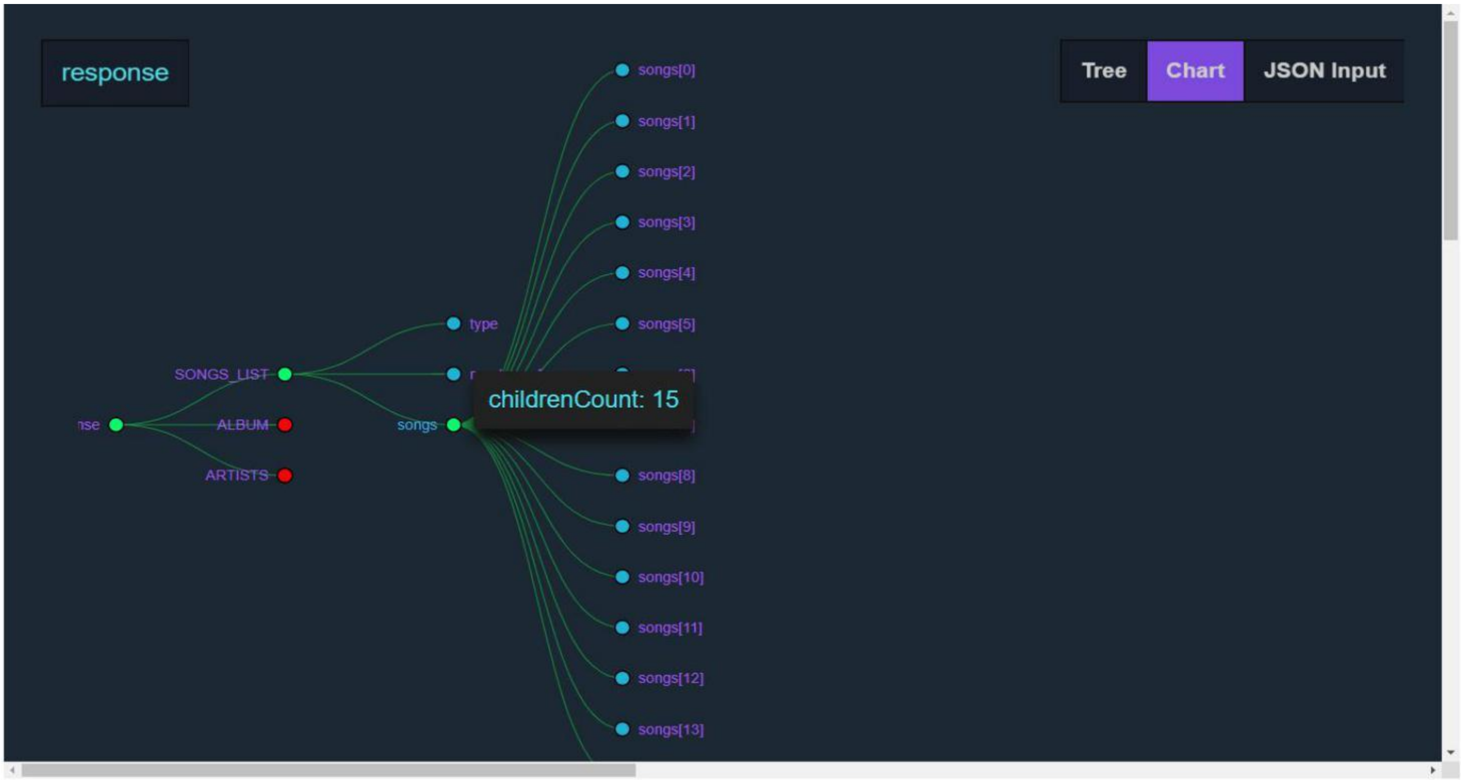


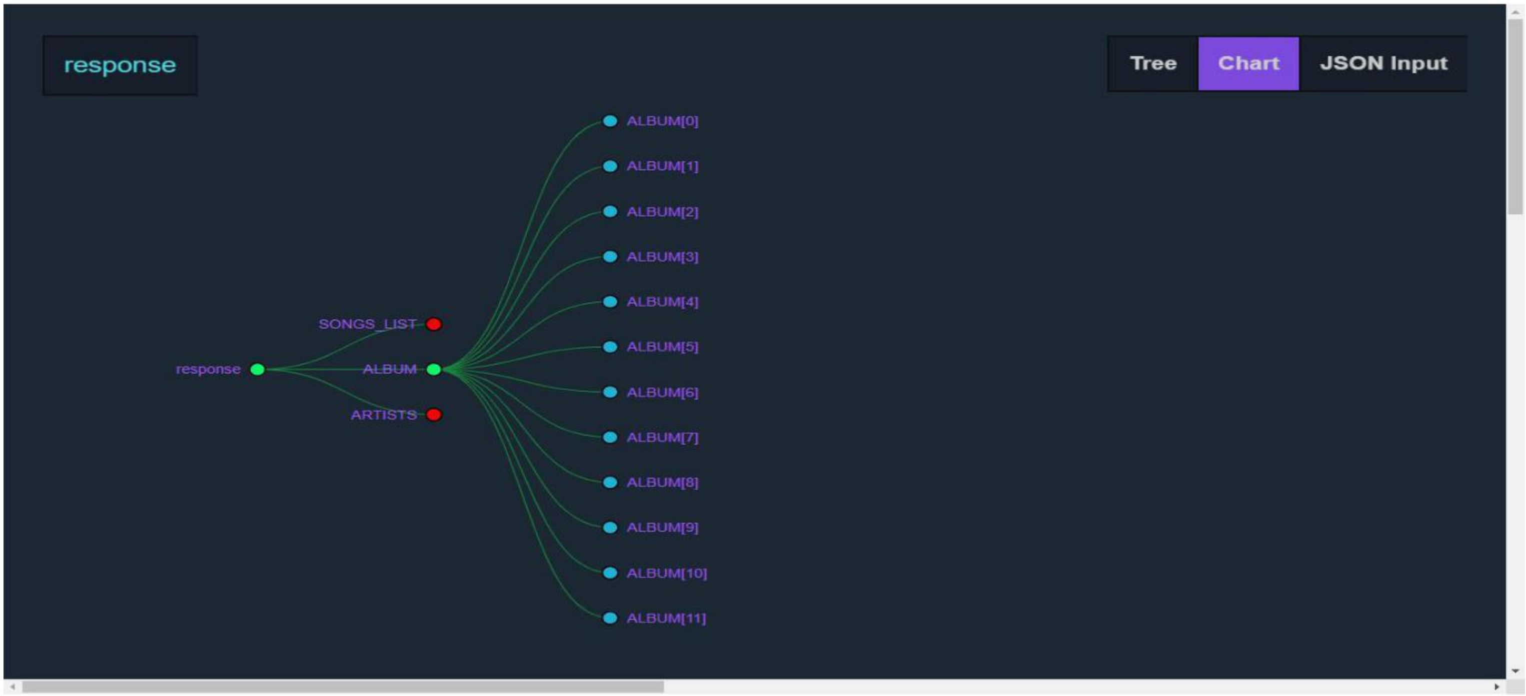
6.2 JSON STRUCTURE

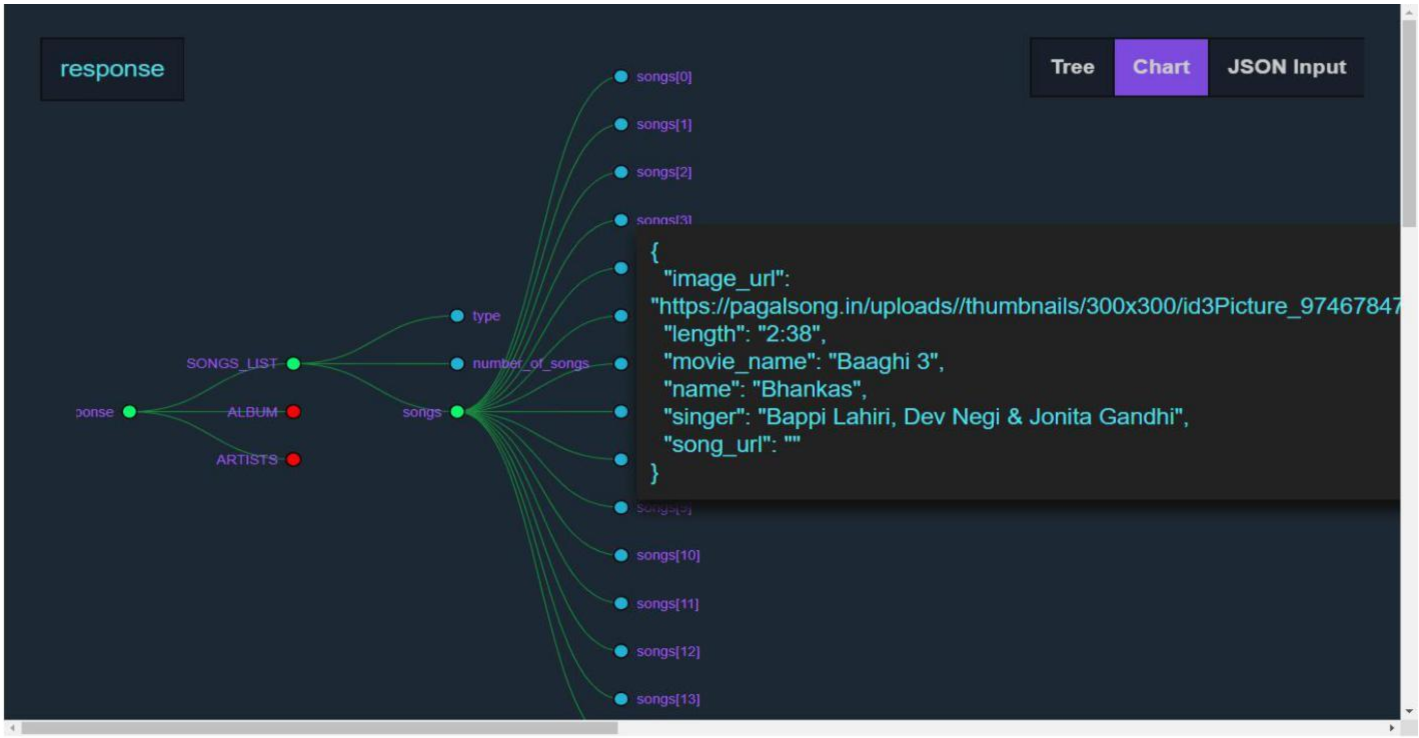
DATA URL: - http://thehacktivist.tech/api_data/music_dataset.json



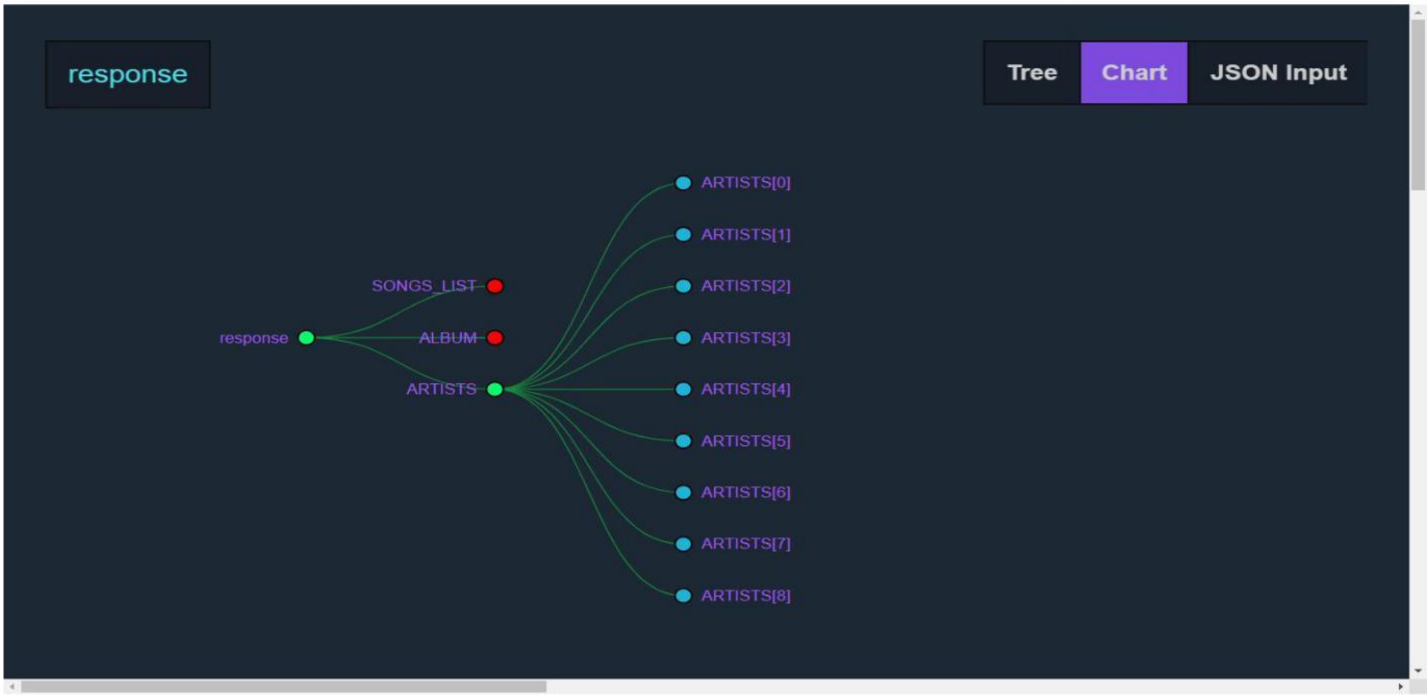


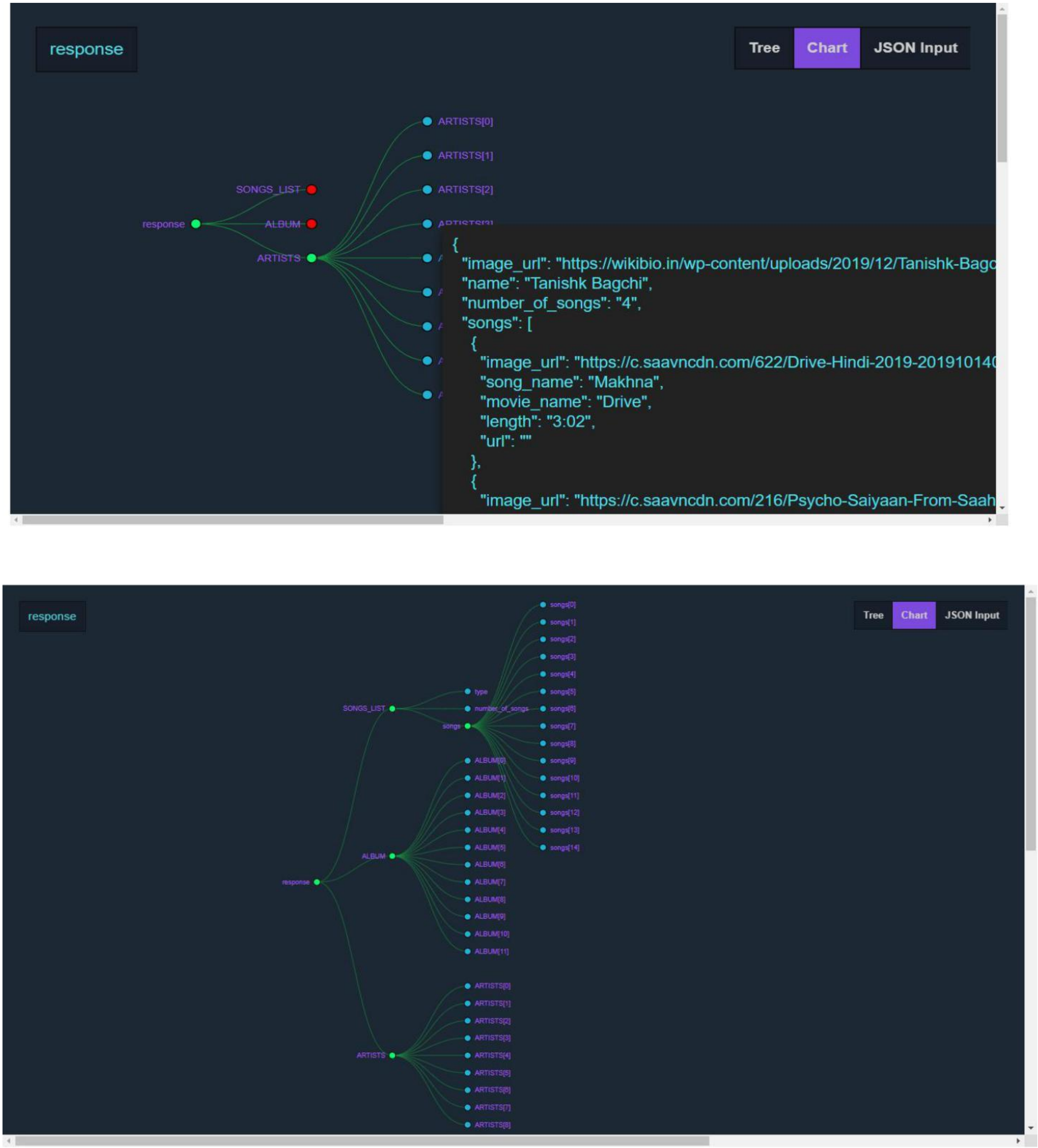

response
Tree
Chart
JSON Input
songs[0]
songs[1]
songs[2]
songs[3]
songs[4]
songs[5]
type
SONGS_LIST
nse
ALBUM
songs
childrenCount: 15
ARTISTS
songs[8]
songs[9]
songs[10]
songs[11]
songs[12]
songs[13]


response
Tree
Chart
JSON Input
ALBUM[0]
ALBUM[1]
ALBUM[2]
ALBUM[3]
ALBUM[4]
SONGS_LIST
ALBUM[5]
response
ALBUM
ALBUM[6]
ARTISTS
ALBUM[7]
ALBUM[8]
ALBUM[9]
ALBUM[10]
ALBUM[11]



```
{
  "image_url":
"https://pagalsong.in/uploads//thumbnails/300x300/id3Picture_97467847
  "length": "2:38",
  "movie_name": "Baaghi 3",
  "name": "Bhankas",
  "singer": "Bappi Lahiri, Dev Negi & Jonita Gandhi",
  "song_url": ""
}
```





```
{
  "image_url": "https://wikibio.in/wp-content/uploads/2019/12/Tanishk-Bagc
  "name": "Tanishk Bagchi",
  "number_of_songs": "4",
  "songs": [
   {
    "image_url": "https://c.saavncdn.com/622/Drive-Hindi-2019-20191014(
    "song_name": "Makhna",
    "movie_name": "Drive",
    "length": "3:02",
    "url": ""
   },
   {
    "image_url": "https://c.saavncdn.com/216/Psycho-Saiyaan-From-Saah
```

FIG.2.2

CHAPTER 7

OUTPUT SCREENS

CHAPTER-7
OUTPUT SCREENS

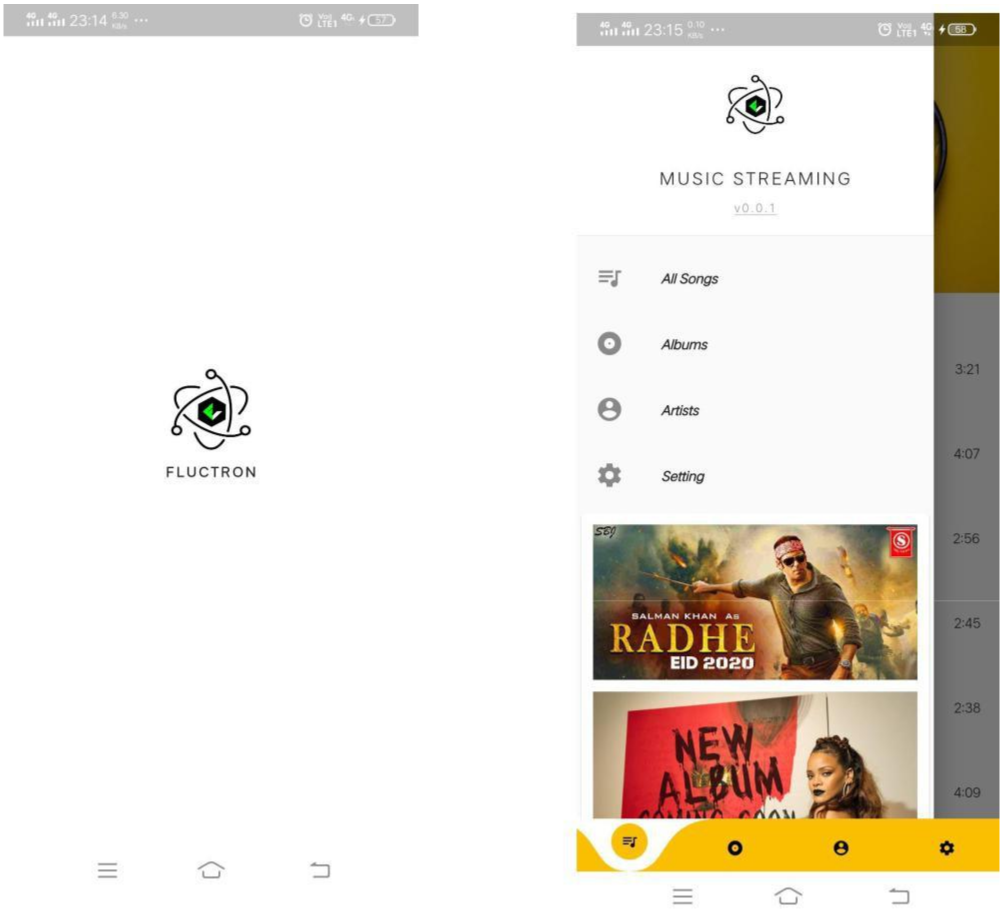
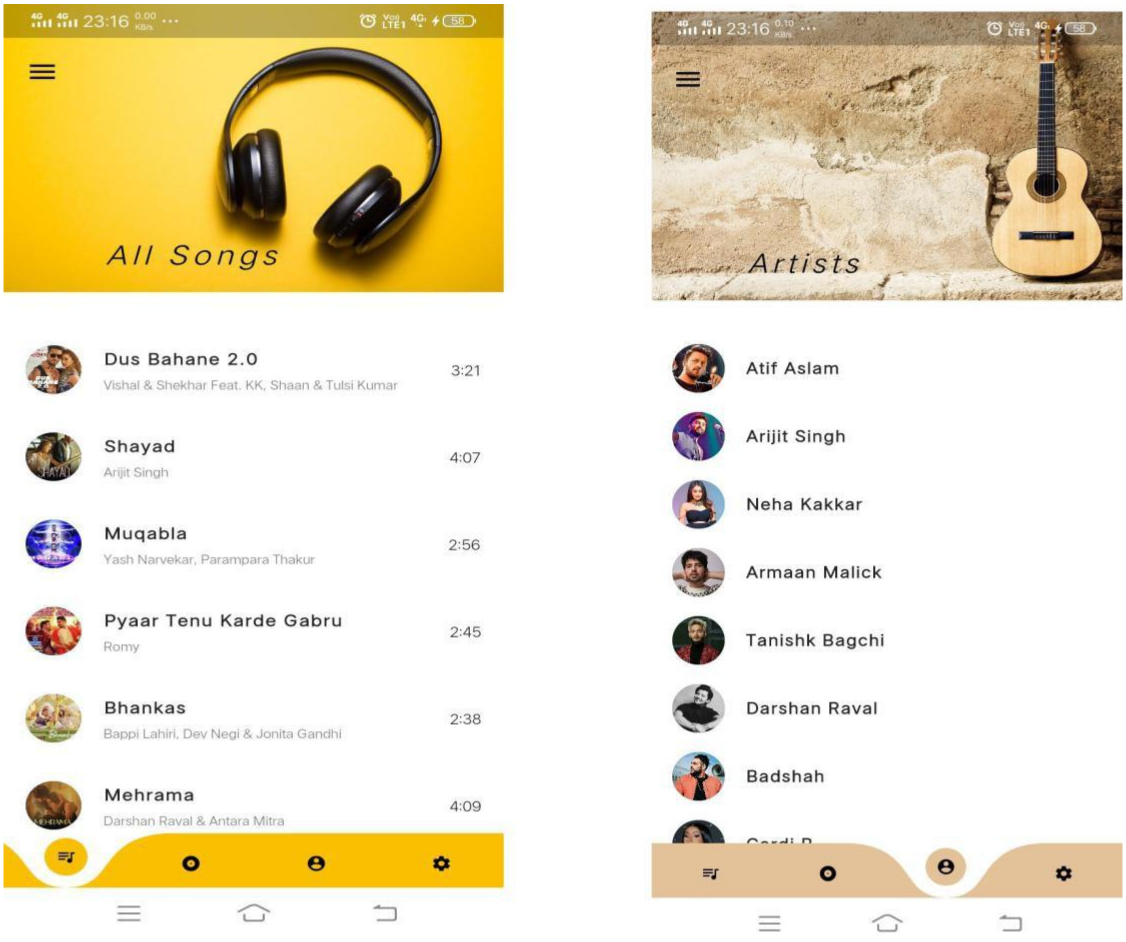


FIG.3.1 and FIG.3.2



All Songs

Dus Bahane 2.0
Vishal & Shekhar Feat. KK, Shaan & Tulsi Kumar
3:21

Shayad
Arijit Singh
4:07

Muqabla
Yash Narvekar, Parampara Thakur
2:56

Pyaar Tenu Karde Gabru
Romy
2:45

Bhankas
Bappi Lahiri, Dev Negi & Jonita Gandhi
2:38

Mehrama
Darshan Raval & Antara Mitra
4:09

Artists

Atif Aslam

Arijit Singh

Neha Kakkar

Armaan Malick

Tanishk Bagchi

Darshan Raval

Badshah

Cardi B

FIG.3.3 and FIG.3.4

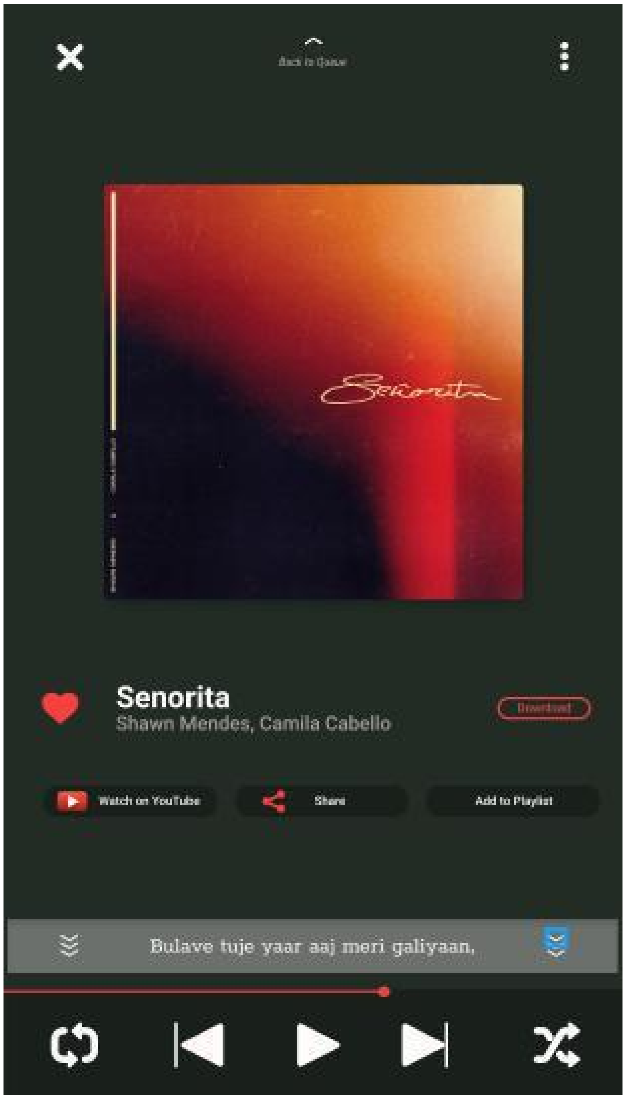

Back to Queue
Senorita
Senorita
Shawn Mendes, Camila Cabello
Download
Watch on YouTube
Share
Add to Playlist
Bulave tuje yaar aaj meri galiyaan,

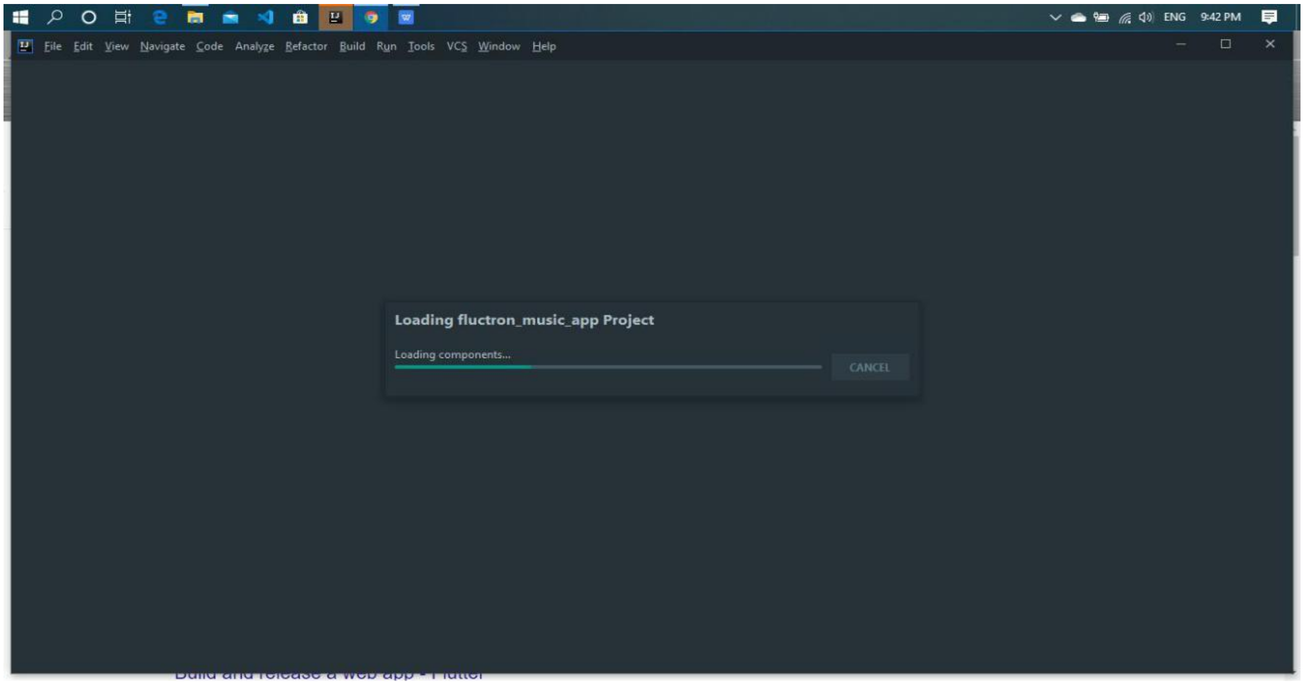
FIG.3.5

CHAPTER 8

DEPLOYMENT

CHAPTER-8
DEPLOYMENT

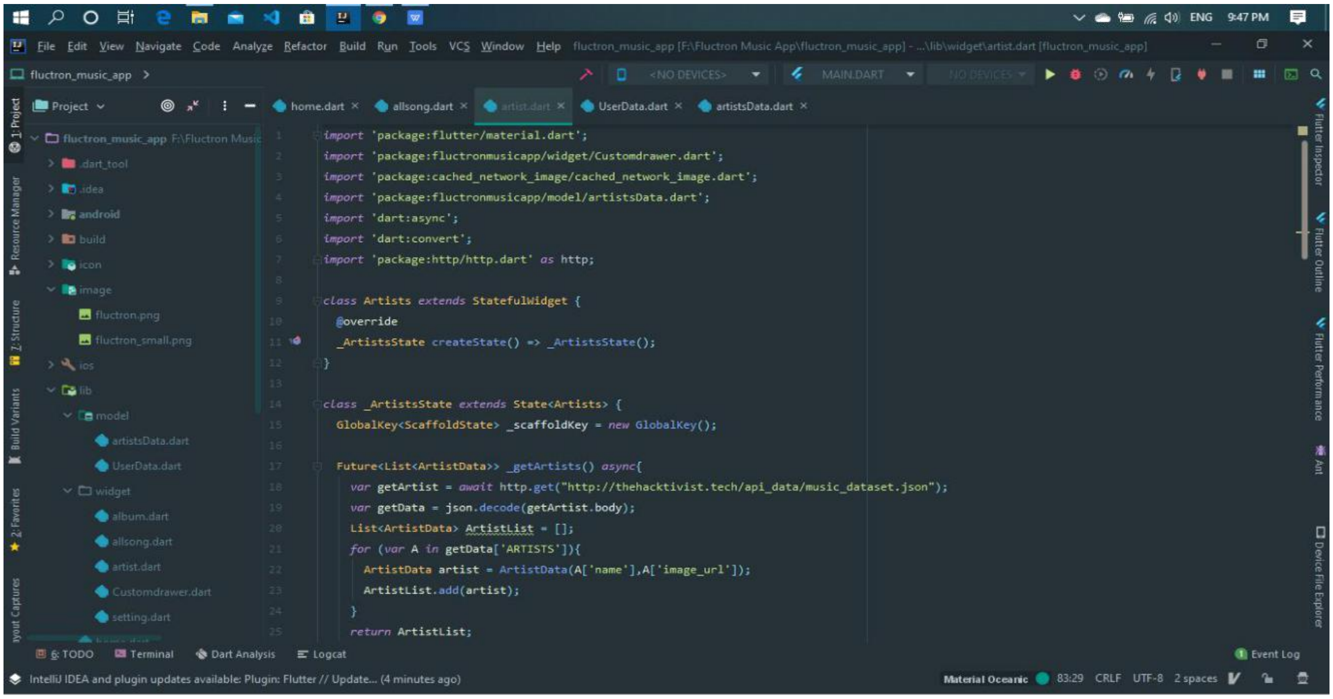
FIG.4.1


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ENG 9:42 PM
CANCEL

Loading fluctron_music_app Project

Loading components...

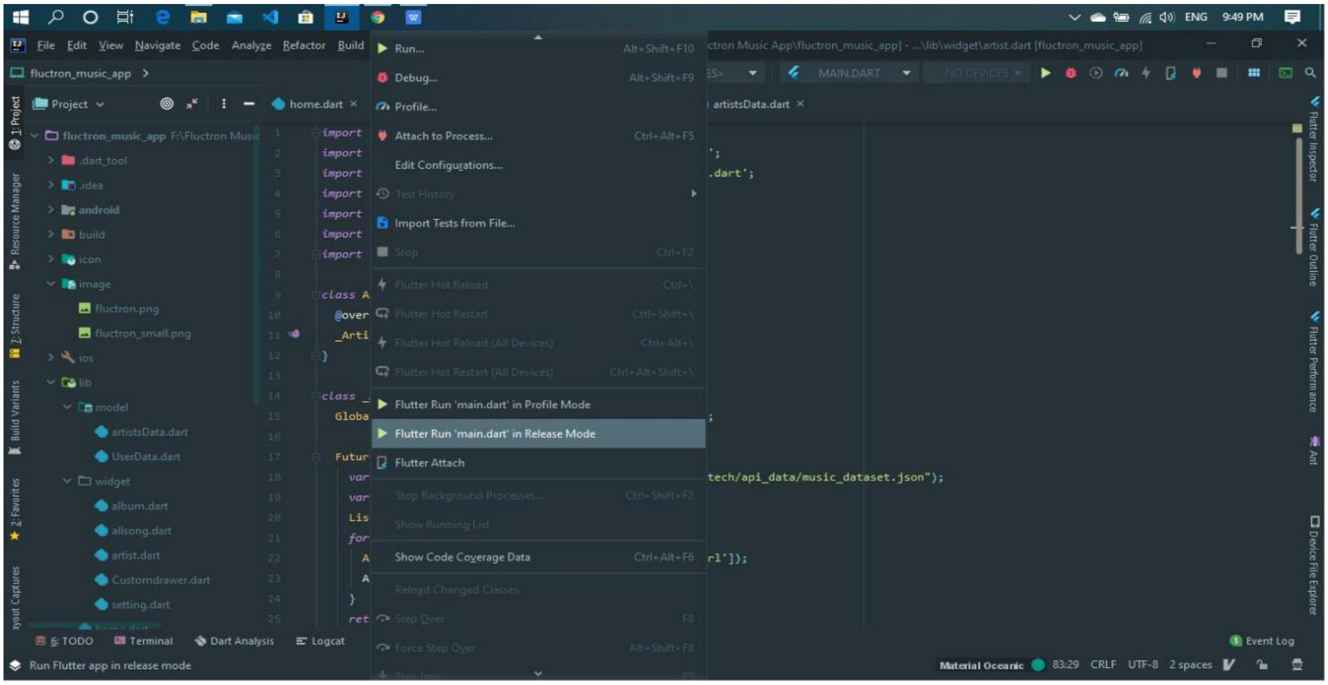
FIG.4.2



```dart
import 'package:flutter/material.dart';
import 'package:fluctronmusicapp/widget/Customdrawer.dart';
import 'package:cached_network_image/cached_network_image.dart';
import 'package:fluctronmusicapp/model/artistsData.dart';
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;

class Artists extends StatefulWidget {
  @override
  _ArtistsState createState() => _ArtistsState();
}

class _ArtistsState extends State<Artists> {
  GlobalKey<ScaffoldState> _scaffoldKey = new GlobalKey();

  Future<List<ArtistData>> _getArtists() async{
    var getArtist = await http.get("http://thehacktivist.tech/api_data/music_dataset.json");
    var getData = json.decode(getArtist.body);
    List<ArtistData> ArtistList = [];
    for (var A in getData['ARTISTS']){
      ArtistData artist = ArtistData(A['name'],A['image_url']);
      ArtistList.add(artist);
    }
    return ArtistList;
```

IntelliJ IDEA and plugin updates available: Plugin: Flutter // Update... (4 minutes ago)

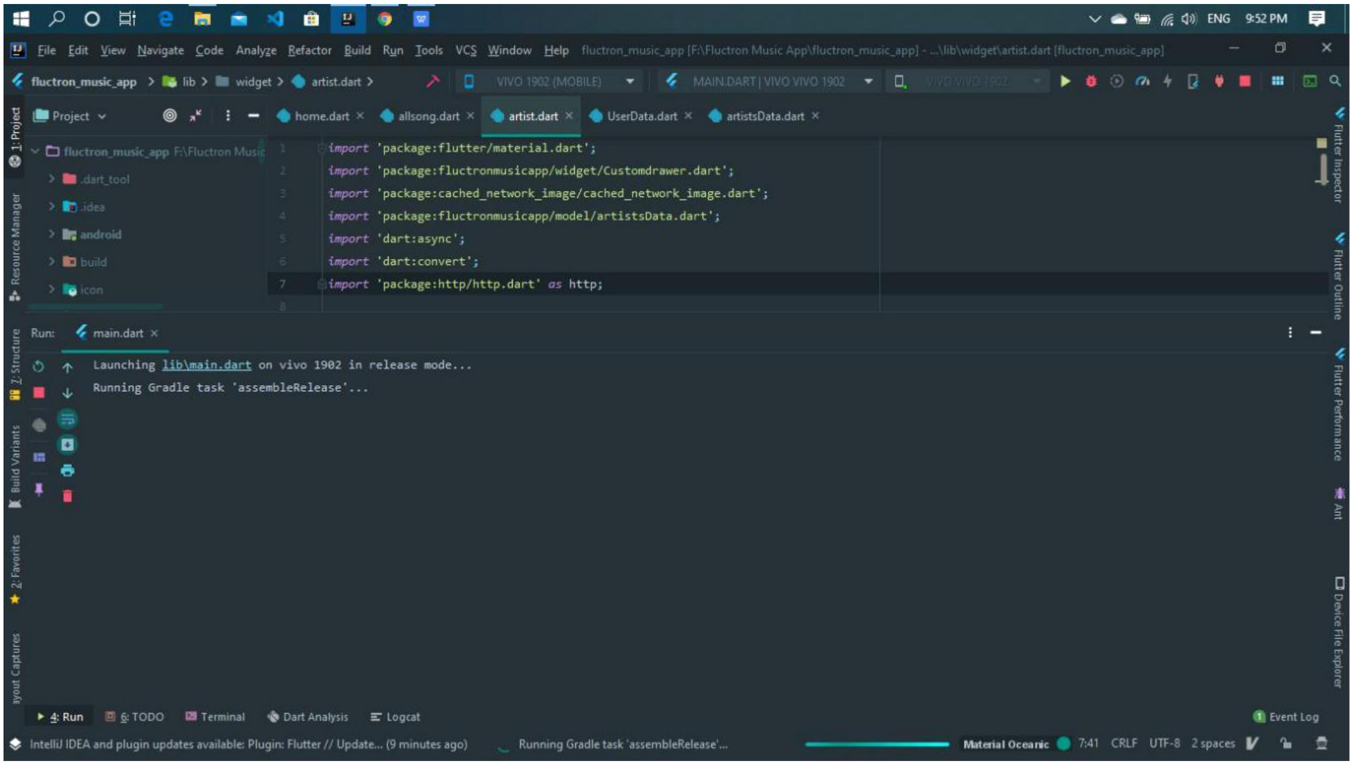
FIG.4.3



Run...	Alt+Shift+F10
Debug...	Alt+Shift+F9
Profile...
Attach to Process...	Ctrl+Alt+F5
Edit Configurations...
Test History
Import Tests from File...
Stop	Ctrl+F2
Flutter Hot Reload	Ctrl+\
Flutter Hot Restart	Ctrl+Shift+\
Flutter Hot Reload (All Devices)	Ctrl+Alt+\
Flutter Hot Restart (All Devices)	Ctrl+Alt+Shift+\
Flutter Run 'main.dart' in Profile Mode
Flutter Run 'main.dart' in Release Mode
Flutter Attach
Stop Background Processes...	Ctrl+Shift+F2
Show Running List
Show Code Coverage Data	Ctrl+Alt+F6
Reload Changed Classes
Step Over	F8
Force Step Over	Alt+Shift+F8

Run Flutter app in release mode

FIG.4.4



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
fluctron_music_app [F:\Fluctron Music App\fluctron_music_app] - ...\lib\widget\artist.dart [fluctron_music_app]
fluctron_music_app > lib > widget > artist.dart >
VIVO 1902 (MOBILE)
MAIN.DART | VIVO VIVO 1902
Project
home.dart
allsong.dart
artist.dart
UserData.dart
artistsData.dart
fluctron_music_app F:\Fluctron Music
.dart_tool
.idea
android
build
icon
Run:
main.dart
4: Run
6: TODO
Terminal
Dart Analysis
Logcat
Event Log
Material Oceanic
7:41
CRLF
UTF-8
2 spaces
ENG
9:52 PM

```
import 'package:flutter/material.dart';
import 'package:fluctronmusicapp/widget/Customdrawer.dart';
import 'package:cached_network_image/cached_network_image.dart';
import 'package:fluctronmusicapp/model/artistsData.dart';
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
```

```
Launching lib\main.dart on vivo 1902 in release mode...
Running Gradle task 'assembleRelease'...
```

IntelliJ IDEA and plugin updates available: Plugin: Flutter // Update... (9 minutes ago)   Running Gradle task 'assembleRelease'...

FIG.4.5

REFERENCES

WEBSITES (with exact URL up to page)

1. https://pub.dev/
2. https://github.com/prakashaditya13?tab=repositories
3. https://dart.dev/tools/dart-devtools