GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD - 382424
GOVERNMENT ENGINEERING COLLEGE
GANDHINAGAR
(Approved by AICTE, Affiliated to GTU)
Sector - 28 – 382028, Gandhinagar

Project Report on

"REAL TIME BEHAVIOR ANALYSIS"

Project (2170001)

B.E. (Semester – VIII)

Team No. 127518

Submitted By:

Name of Student	Enrollment No.
PINKY CHAUDHARY	170130116007
HONEY DARJI	170130116009
POOJA THAKKAR	170130116053

Professor C.M. KAPADIYA

(Faculty Guide)

Professor D.A. PARIKH

(Head of Department)

Academic Year (2020-2021)

GUJARAT TECHNOLOGICAL UNIVERSITY
GOVERNMENT ENGINEERING COLLEGE
Sector - 28, Gandhinagar -382028
(Approved by AICTE, Affiliated to GTU)

CERTIFICATE

This is to certify that project titled "**Real Time Behavior Analysis**" has been carried out by PINKY CHAUDHARY (170130116007), HONEY DARJI (170130116009) and POOJA THAKKAR (170130116053) has completed Project Report in a group under guidance of the faculty guide Prof. C. M. KAPADIYA in fulfillment of the degree of Bachelor of Engineering in Information Technology (8th semester) of Gujarat Technical University, Ahmedabad during the academic year 2020-2021.

Head of Department

ACKNOWLWDGEMENT

We would like to take this opportunity to express our sincere thanks and deep sense of gratitude to all those people without whose support, encouragement and co-operation this project would not have been a success.

We would like to thank our institute for giving us the opportunity to have some feel about the Industrial environment. We would like to thank our H.O.D Prof. D. A. PARIKH, our Faculty guide Prof. C. M. Kapadiya for constantly guiding and showing us the correct path to reach towards our desired goal.

This acknowledgement will be incomplete if we do not convey our heartfelt gratitude to other faculty members for their guidance and moral support during the preparation of this project.

PINKY CHAUDHARY

DARJI HONEY

POOJA THAKKAR

Originality of Work

We hereby certify that we are the sole authors of this UDP project report and that neither any part of this UDP project report nor the whole of the UDP Project report has been submitted for a degree by other student(s) to any other University or Institution.

We certify that, to the best of our knowledge, the current UDP Project report does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our IDP/UDP Project report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current UDP Project report and have included copies of such copyright clearances to our appendix.

We have checked the write up of the present UDP Project report using anti-plagiarism database and it is in the allowable limit. In case of any complaints pertaining to plagiarism, we certify that we shall be solely responsible for the same and we understand that as per norms, University can even revoke BE degree conferred upon the student(s) submitting this UDP Project report, in case it is found to be plagiarized.

Team ID: 127518

Name	Enrollment number
Pinky Chaudhary	170130116007
Honey Darji	170130116009
Pooja Thakkar	170130116053

Place: Gandhinagar

Faculty Guide: C. M. Kapadiya

Table of Content

Chapter	Title	Page No
Chapter 1	Introduction	**1**
	1.1 Problem Summary	2
	1.2 Purpose	2
	1.3 Objective	3
	1.4 Problem Specification	3
	1.5 Brief Literature Review and PAS about project	4
	1.6 Project Planning	6
	1.7 Materials / Tools required	8
Chapter 2	Design	9
	2.1 System Architecture	10
	2.2 System Overview	12
	2.3 Ideation Canvas	14
	2.4 Product Development Canvas	15
	2.5 Business Canvas	16
Chapter 3	Implementation	17
	3.1 Methodology and Dataset	18
	3.2 Data Dictionary	20
	3.3 Emotion Extraction Module	22
	3.4 Playlist Generation Module	25

	3.5 Actual Implementation	27
	3.6 Application Result	30
	3.7 Testing and Verification	32
Chapter 4	Summary of the Result	36
	4.1 Summary	37
	4.2 Advantages of Result	37
	4.3 Scope	37
Appendix		38
	5.1 PPR	39
	5.2 PDE	41
	5.3 Plagiarism Search Report	44
	5.4 References	45

Chapter 1

INTRODUCTION

1.1 Problem Summary

Recommending music based on a user’s music preference is a way to improve user listening experience. Finding the correlation between the user data (e.g., location, time of the day, music listening history, emotion, etc.) and the music is a challenging task. There are a lot of automated music recommendation available on a payable subscription. But they don’t consider internal emotion of user as their key feature.

We have driven music recommendation to the next level, generating playlist which suits with user’s mood of listening music. Here, we propose a personalized music recommendation system based on the facial expression of user. Using front camera of your device, facial emotion is classified in one of pre specified classes (angry, contempt, happy, excited, sad, fear, disgust, surprise) and music playlist is requested using API then songs are played through the music player.

We are Considering human Facial Points (Landmark) which will give an accurate description of the Data Points for Emotion Classification. For music recommendation we are using songs features (energy and valence) and correlating that with the emotion category to Segregate and Generate a new Playlist.

1.2 Purpose

The purpose of project to eliminate the time-consuming and tedious task of manually segregating or grouping songs into different playlists. Our Project purposed designed and develop an emotion-based playlist generation using facial recognition system. Playlists are generated using user’s data from Spotify API. It aims at scanning and interpreting the data and accordingly creating a playlist based the parameters provided.

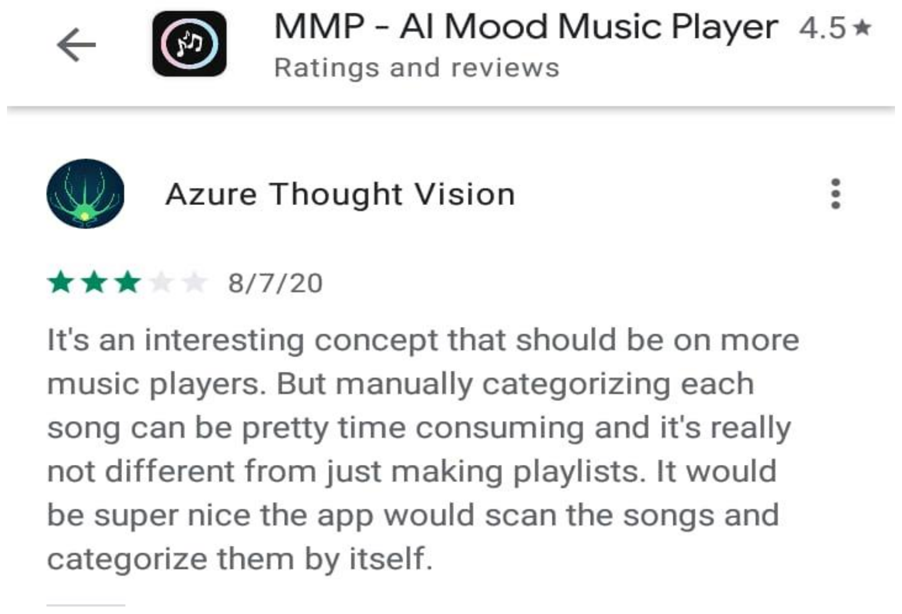
1.3 Objective

- ❖ To provide an interface between music player and human emotion.
- ❖ To provide a very good entertainment for the users.
- ❖ To implement Machine Learning for Emotion Classification.
- ❖ To provide a new age platform for music lovers.

1.4 Problem Specification

1. The music player available in market don't consider the actual human biometric feature for personalized recommendation.

2. The MMP -AI mood Music Player uses but users have to manually create playlist and group songs in a playlist and there is no streaming option from other platforms is available in that.



MMP - AI Mood Music Player 4.5★
Ratings and reviews

Azure Thought Vision

★★★☆☆ 8/7/20

It's an interesting concept that should be on more music players. But manually categorizing each song can be pretty time consuming and it's really not different from just making playlists. It would be super nice the app would scan the songs and categorize them by itself.

1.5 Brief Literature Review and Prior Art Search About Project

1.5.1 Existing Research

Generating Music Playlist Based on facial Expression

Markus Mans Folke Andreasson et al [2] purposed a system which plays a song on a device, and then for next song recommendation it captures an image of a user, perform facial expression recognition of the user based on the image. It states that image of the user may be captured based on at least one of the user's interaction with the device or a periodic timing mechanism. It needs user interaction and plays music available in the device only.

Using Animated Mood Pictures in Music Recommendation

Arto Lehtiniemi and Jukka Holm et al [3] proposed system on animated mood picture in music recommendation. On this system the user interacts with a collection of images to receive music recommendation with respect to genre of picture. This music recommendation system is developed by Nokia researched center. This system uses textual meta tags for describing the genre and audio signal processing.

Human-computer interaction using emotion recognition from facial expression.

F. Abdat, C. Maaoui et al and A. Pruski et al. They proposed a system fully automatic facial expression and recognition system based on three step face detection, facial characteristics extraction and facial expression classification [4]. This system proposed anthropometric model to detect the face feature point combined to shi and Thomasi method. In this method the variation of 21 distances which describe the facial feature from neutral face and the classification base on SVM (Support Vector Machine).

Enhancing Music Recommender Systems with Personality Information and Emotional States

Bruce Ferwerda et al and Markus Schedl et [5] al proposed that the initial research assumptions to improve music recommendations by including personality and emotional states. By including these psychological factors, the accuracy of the recommendation can be enhanced. The system gives attention to how people use music to regulate their emotional states, and how this regulation is related to their personality.

1.5.2 Existing System

Currently, there are many existing music player applications. Some of the interesting applications among them are:

1) Saavan: These application gives good user accessibility features to play songs and recommends user with other songs of similar genre

2) Moodfuse: In this application, user should manually enter mood and genre that wants to be heard and mood fuse recommends the songs-list

3) Steromood: User should select his mood manually by selecting the moods from the list and the application plays music from YouTube.

4) Musicovery: This application has High quality songs and music recommendations. It also suggests predefined playlist for the user.

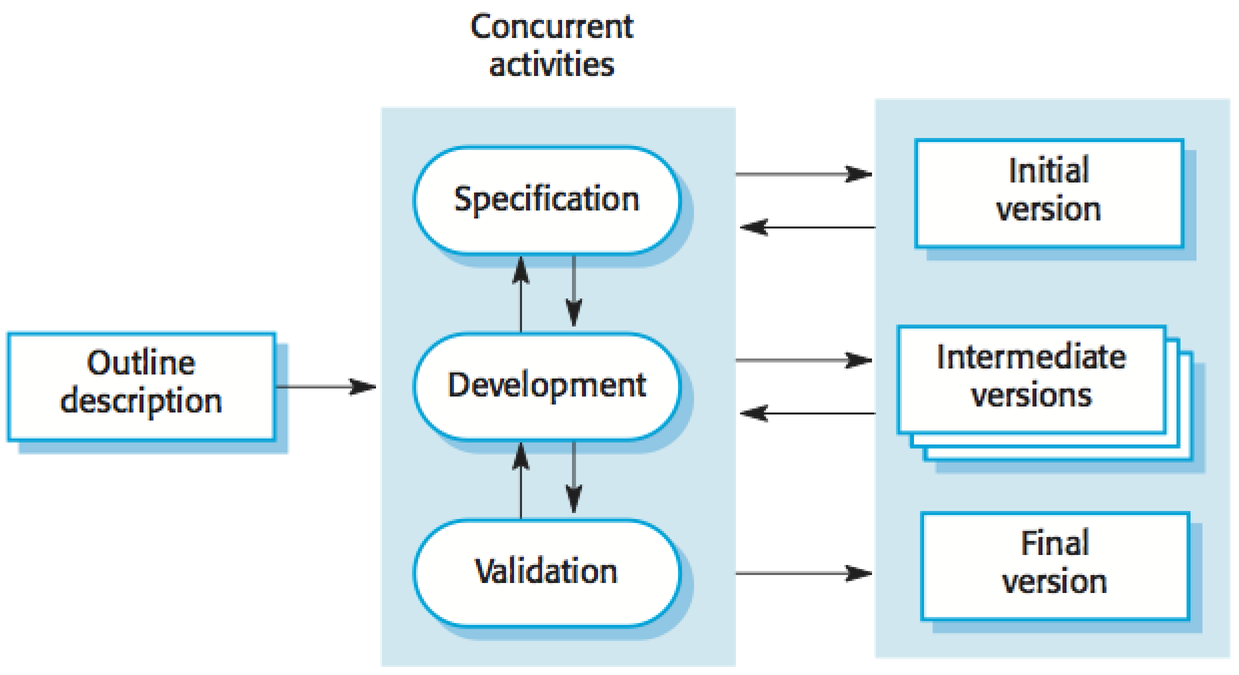
5) Gaana: It features music from 21 languages, user friendly and it allows users to make their playlists public so that they can be seen by other users. Supports all OS.

1.6 Planning

Our Model is developed using Incremental Software Development Life Cycle Model

Incremental Model is a used where requirements divided into multiple standalone Iterations of the software development cycle. Every subsequent version of product adds function to the previous release. The process continues until the complete system achieved.

Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.



Incremental Model

Advantage of Incremental Model:

Lower cost of changes

The cost of accommodating changing customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.

Frequent feedback

It is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented.

Faster delivery

More rapid delivery and deployment of useful software to the customer is possible. Customers are able to use and gain value from the software earlier than is possible with a waterfall process.

Disadvantage of Incremental Model

The process is not visible

Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system.

System structure tends to degrade as new increments are added

Unless time and money are spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly

1.7 Materials / Tools/Software required

The Following are the minimum requirements to develop the application:

1. HARDWARE REQUIREMENT:

- ❖ Processor: 2GHz
- ❖ RAM: 1GB

2. SOFTWARE AND FRAMEWORK

- ❖ TensorFlow >=2
- ❖ Flask==1.1.2
- ❖ Python >=3.7
- ❖ PostgreSQL 13.2
- ❖ Python IDE
- ❖ Spotify API

CHAPTER 2

DESIGN

2.1 SYSTEM ARCHITECTURE

The system architecture of Semotion is shown in Fig 2.1. The application is built using the architectural pattern of Model-View-Controller. It is also widely used architecture. Here, the application is divided into three main logical components: the model, the view and the controller.

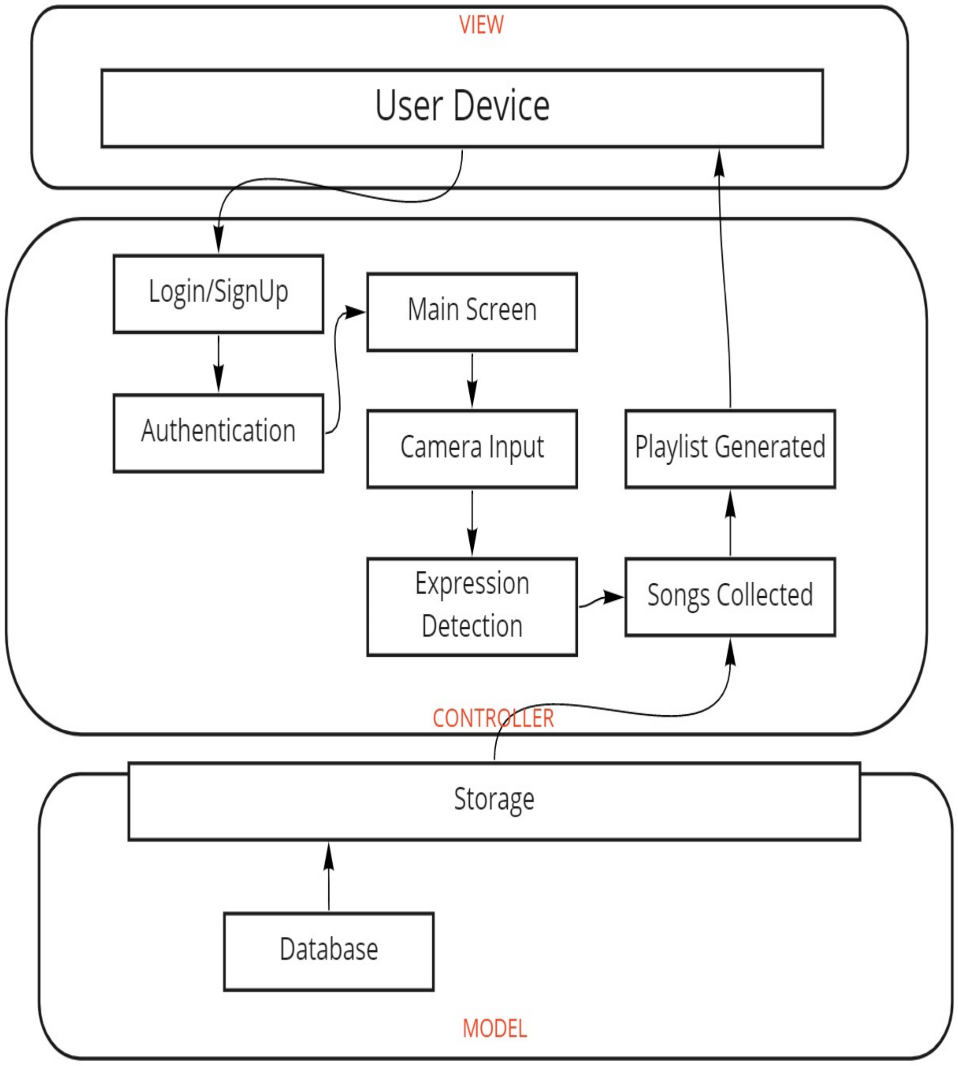


Fig. 2.1. System Architecture of Semotion

• **View**: The top layer is where the end-user communicates with the application through clicking buttons, typing details, accessing camera, clicking play button, chosing previous playlist, etc. This layer is responsible for displaying all data or a portion of data to user based on the requirement of the application. This layer also acts as a bridge between the user and application itself. HTML and Javascript is used in this application for displaying the output or response of the system to the user.

• **Controller**: This middle layer of the application contains the business logic, and the main functionality of the application. As soon as the user interacts with the application, the response is processed in this layer. From log-in to displaying play-list, all the functions that run in background belong to this layer. This mainly consists of all the functions and Main model (Face detector and Emotion recognizer) interaction which helps in segregating songs and sending output to view layer.

• **Model**: This layer is responsible for maintaining the user's data. Semotion uses Postgres SQL database to store user data temprorily and Flask-SQLAlchemy is used to interact with that database. Postgres SQL data base stores data of user and their music prefrence till the user is loged in as soon as the user logs out all their information is also removed from database to provide user security.

Here both the face detector and emotion classifier are the model converted in for JavaScript integration and is used in client-side only. TensorflowJs is used to convert the Keras models created in python to convert them in Browser Compatible.

2.2 SYSTEM OVERVIEW

This section illustrates the design and functional phase of the application. Semotions is a web application, user can access the app using https://semotions.herokuapp.com and play songs based on their emotions.

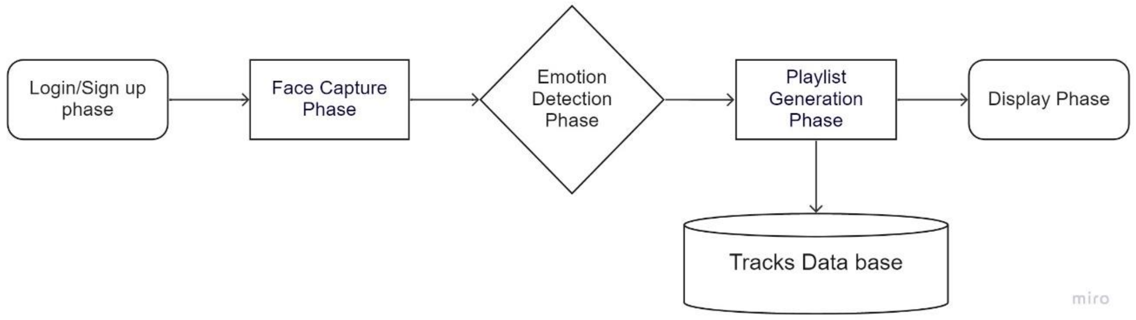


Fig 2.2 depicts the overview of the application

1. Login/signUp phase: Our Application uses users spotify data so they have to login to access customized play-lists as well as songs. Once user logs-in, their temporly data is stored in User and User Track Tables in our database ,until they manually log-out. Whenever a new playlist is created that is added in spotify Account.

2. Face Capture phase: As soon as the authentication phase is done, user is provided with two options either to create a new playlist or to play from previoulsy created playlists, when user chooses to create a new playlist the application will ask user's permission to access camera to and loads the model in browser.

3. Emotion Detection Phase: After the model is loaded the real time video of user is processed and face deretion and emotion capture models start predtion once user is satisified with the captured emotion he/she can submit the prediction and name their custom playlist.

4. Playlist-Generation phase: This is performed in backend using spotify API to fetch tracks features (energy and valence) and correlate that to the value of mood classified. We have used Thayers' Method to create the coorelation matrix for music featue and user mood value. Songs are added in playlist.

5. Display-Phase: Here, the songs are displayed in client browser and user can play them by cliacking on each song individually or by using the embedded Spotify web player.

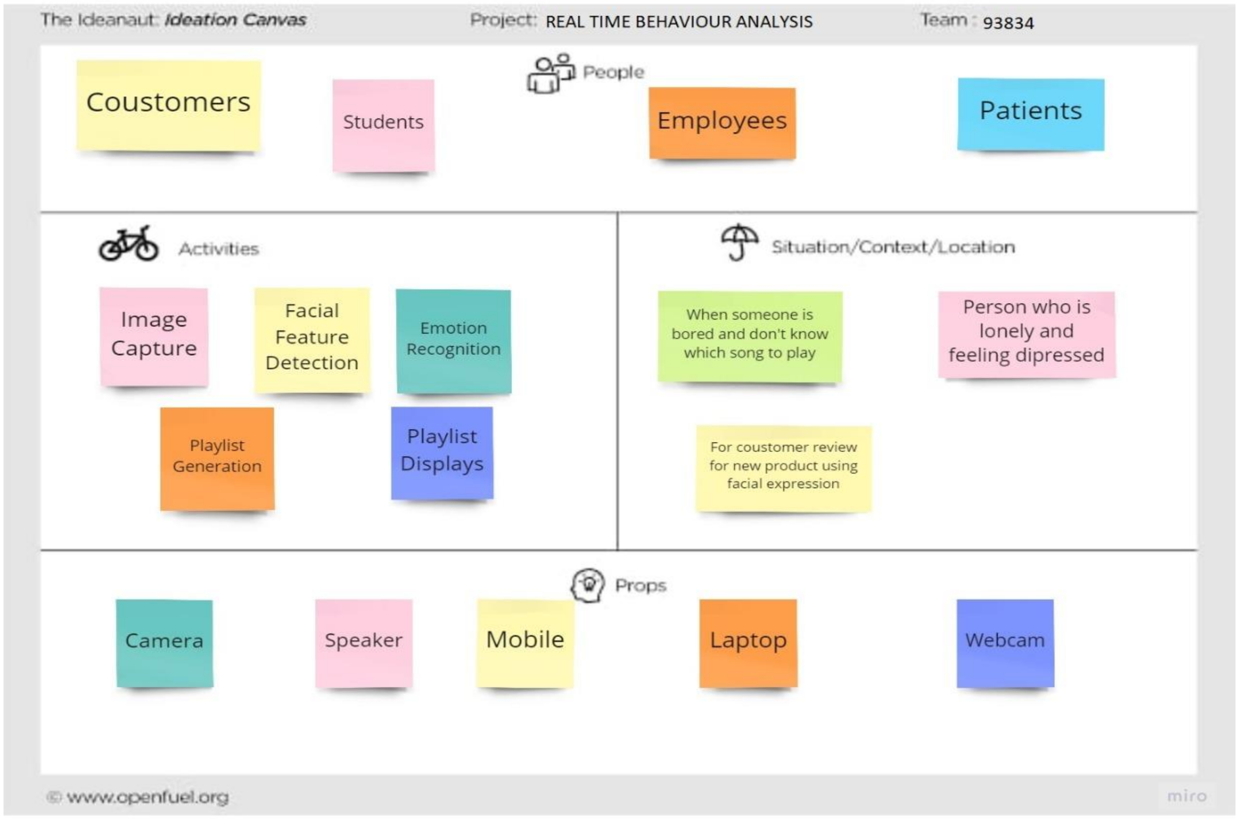
2.3 Ideation Canvas

Ideation Canvas where first we started with people, where we simply thought about the people for whom we want to solve the problem.

Then list out whatever activity every segment of people does. Then thought for context/location/situation and finally for possible solutions.

Ideation is the creative process of generating, developing and communicating new ideas, where an idea is understood as a basic element of thought that can be visual, concrete or abstract. Ideation compromises all stages of thought cycle, from innovation, to development, to actualization. As such, it is an essential part of the design process, both in education and practice.

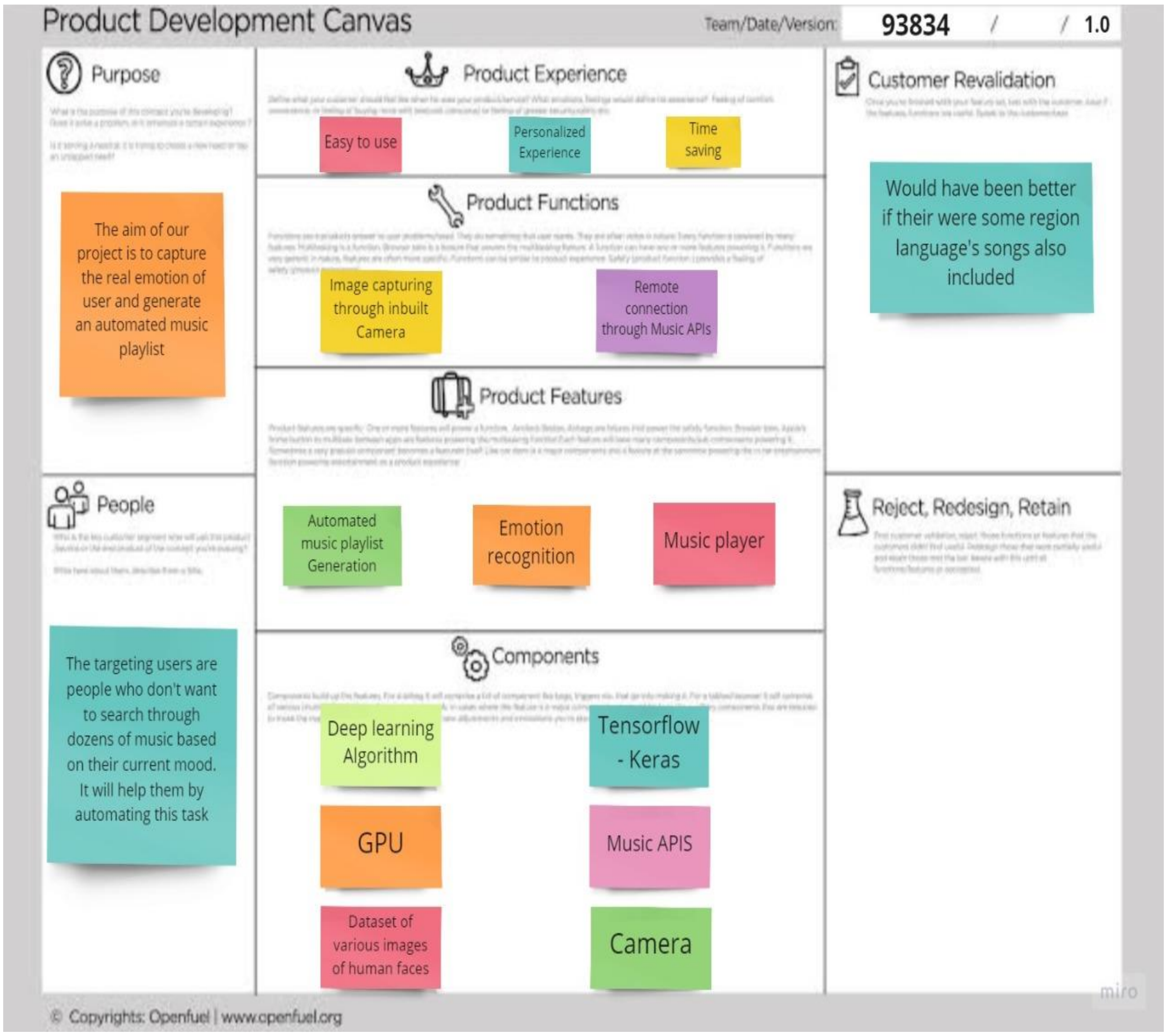
An Ideation canvas is sheet where ideas can be stretched into any limits or dimensions


The Ideanaut: Ideation Canvas
Project: REAL TIME BEHAVIOUR ANALYSIS
Team : 93834
People
Coustomers
Students
Employees
Patients
Activities
Image Capture
Facial Feature Detection
Emotion Recognition
Playlist Generation
Playlist Displays
Situation/Context/Location
When someone is bored and don't know which song to play
Person who is lonely and feeling dipressed
For coustomer review for new product using facial expression
Props
Camera
Speaker
Mobile
Laptop
Webcam
© www.openfuel.org
miro

2.4 Product Development Canvas

This exercise is meant for giving strategic orientation to the project of each team so that it achieves its true goal as defined by previous canvas exercises.

This exercise is more about developing strategy for the proposed product/solution design, after the team has successfully attempted the ideation process and has incorporated inputs from all Stakeholders.



Product Development Canvas

Team/Date/Version: 93834 / / 1.0

Purpose

The aim of our project is to capture the real emotion of user and generate an automated music playlist

People

The targeting users are people who don't want to search through dozens of music based on their current mood. It will help them by automating this task

Product Experience

Easy to use

Personalized Experience

Time saving

Product Functions

Image capturing through inbuilt Camera

Remote connection through Music APIs

Product Features

Automated music playlist Generation

Emotion recognition

Music player

Components

Deep learning Algorithm

Tensorflow - Keras

GPU

Music APIS

Dataset of various images of human faces

Camera

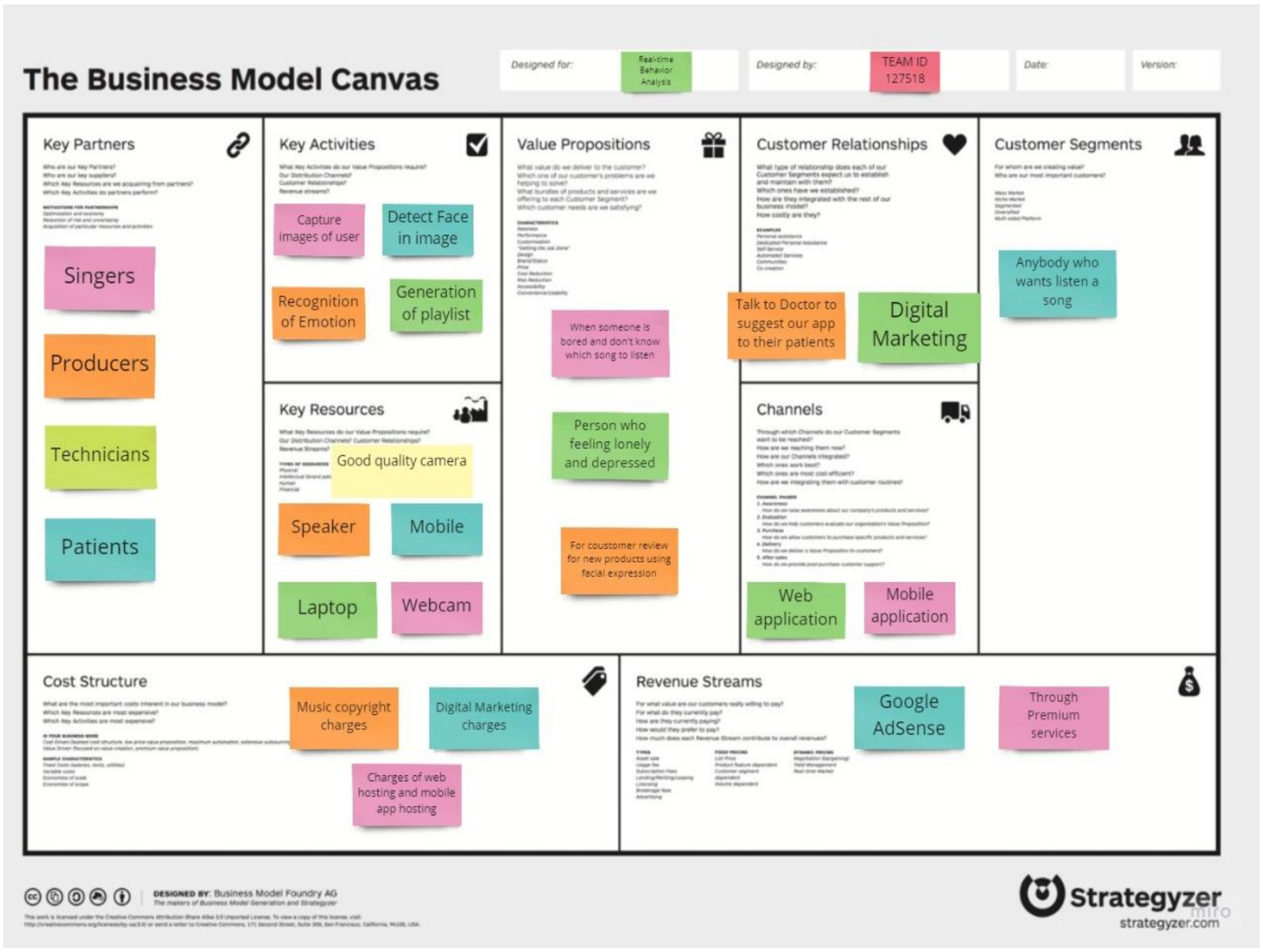
Customer Revalidation

Would have been better if their were some region language's songs also included

Reject, Redesign, Retain

© Copyrights: Openfuel | www.openfuel.org

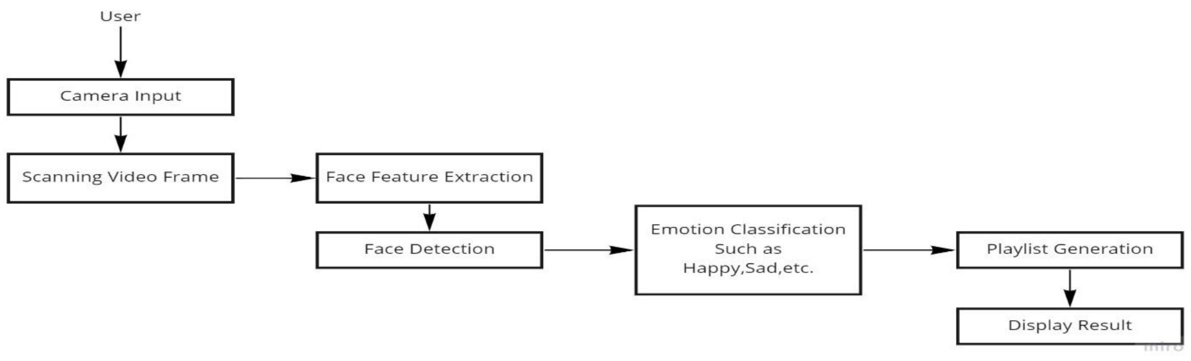
2.5 Business Canvas



SECTION 3

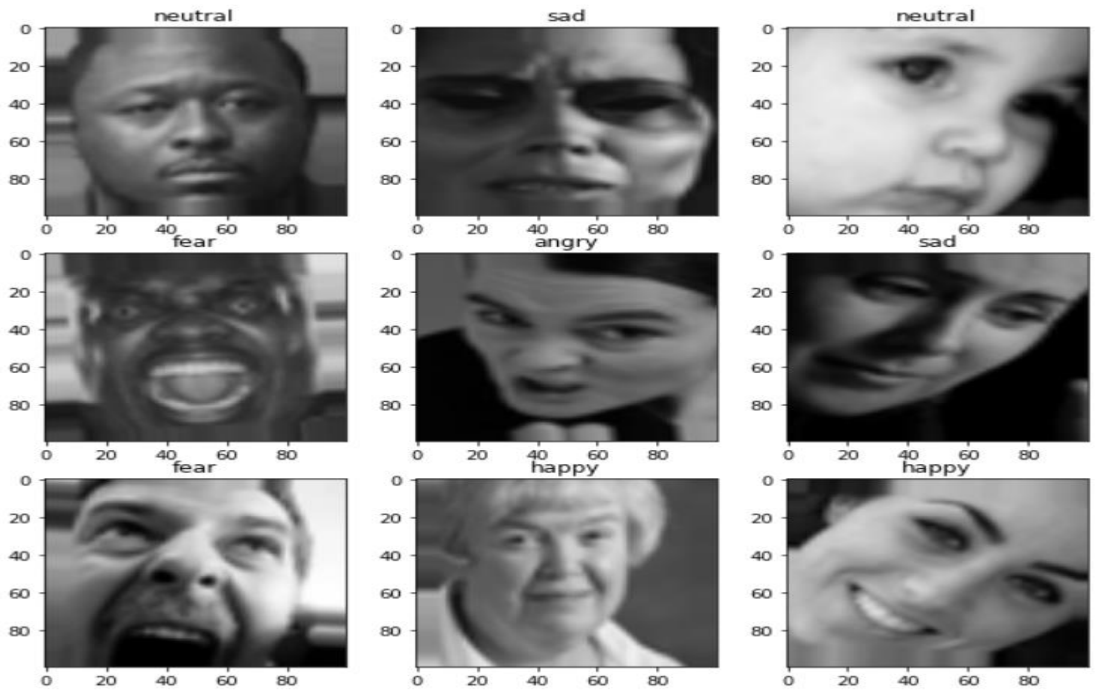
IMPLEMENTATION

3.1 Methodology and Data-Set

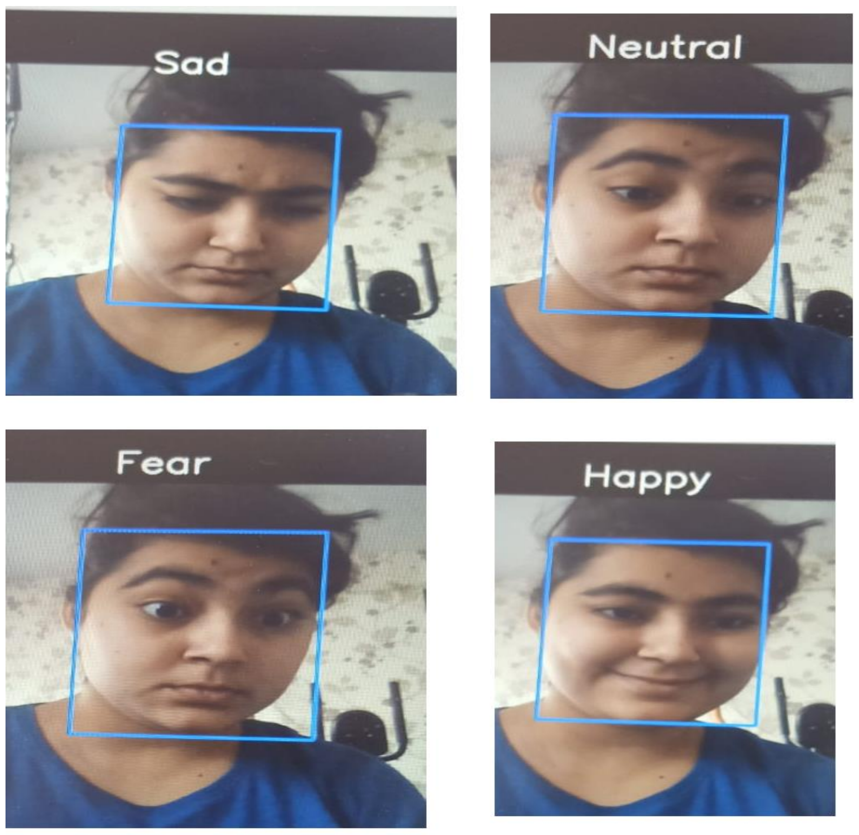


The system purposes automated music recommendation system that plays song according to the mood or current emotion of the person. The person's photo is captured whenever the application user wishes to create a new playlist, hence current emotion is captured and detected. According to the information given by the image, the song is played related to the emotion. The songs are fetching from user-track database (as shown below) and are classified based on their valence and feature value and emotion category such as happy, sad, anger, surprise, fear, disgust and neutral. For New user who don't have any information based on artist we have used the top 50 artist playlist to create an artist list for them and overcome the cold start problem in recommendation system. Application is composed of three modules. Emotion Recognition module, Playlist Generation module and System Integration module. Facial feature extraction and expression recognition are combined in video capture phase and Playlist generation is mutually exclusive modules from them. Hence, system integration module maps two modules to find the correct match of detected emotion. For Expression Classifier Model we have created a Raw dataset by downloading images one by one from google for seven expressions. Extra Data Set is taken from Kaggle datasets for Facial Expression detection. The FER2013 dataset is images of 48,48 grayscale pixel we converted them in 100*100 RGB images for better accuracy and to create a sample as close as possible to real world images.

Sample Images for Training Data



Testing Through Real Time Video Frames



3.2 Data Dictionary

A data dictionary is a catalog – a repository – of the elements in a system as the name suggests, these elements center on data the way they are structured to meet user requirements and organization needs. In a data dictionary you will find a list of all the elements composing the data flowing through a system. The major elements are data flows, data stores, and processes. The data dictionary stores details and descriptions of these elements.

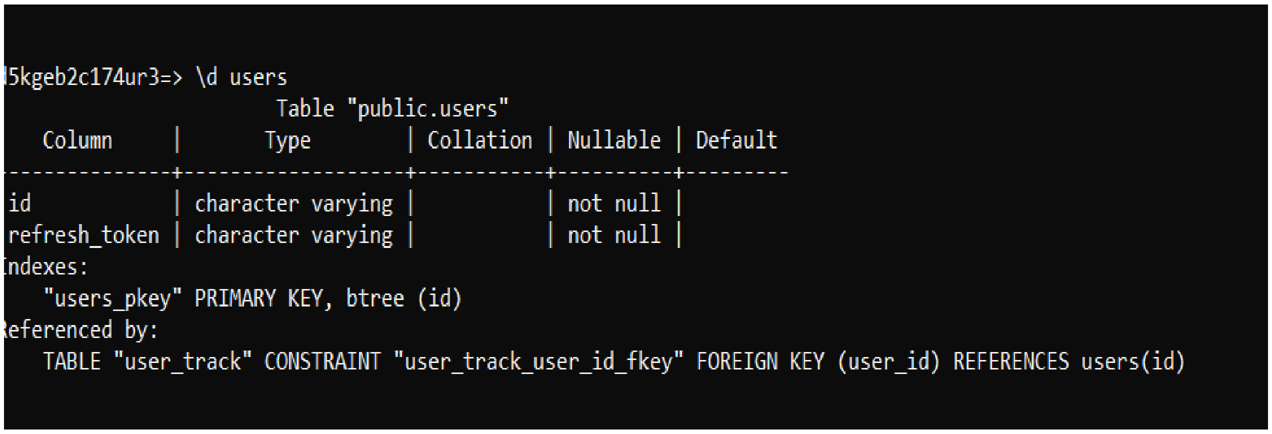
❖ **Features of using Data Dictionary**

The volume of data in most information systems applications is substantial – more than a single analyst can easily keep track of.

Diagrams by themselves do not fully describe the subject of the investigation. The data dictionary provides additional information about the system

1) User Detail Table

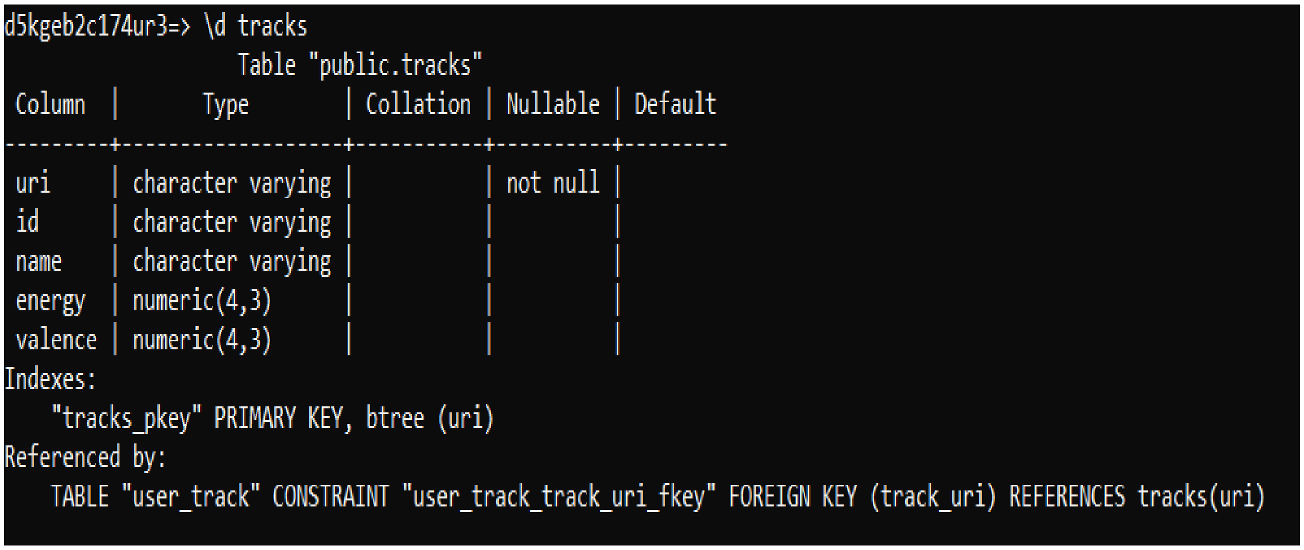
To store temporary data for user in database we have created a User Table as shown below. When user Logs-in in our application a new entry is created in table with their refresh token for future authentication in Spotify account.



```
5kgeb2c174ur3=> \d users
                       Table "public.users"
   Column     |       Type        | Collation | Nullable | Default
--------------+-------------------+-----------+----------+---------
 id           | character varying |           | not null |
 refresh_token | character varying |           | not null |
ndexes:
   "users_pkey" PRIMARY KEY, btree (id)
eferenced by:
   TABLE "user_track" CONSTRAINT "user_track_user_id_fkey" FOREIGN KEY (user_id) REFERENCES users(id)
```

2) Tracks Table

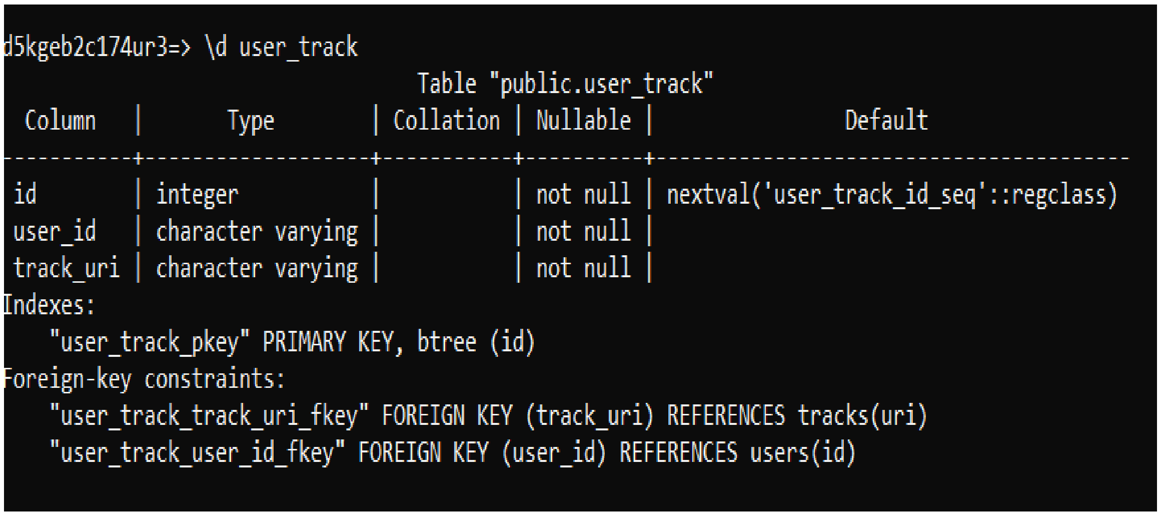
Track table is permanent table and store only track related information Track_uri, energy and valency feature also.



```
d5kgeb2c174ur3=> \d tracks
                  Table "public.tracks"
 Column  |       Type        | Collation | Nullable | Default
---------+-------------------+-----------+----------+---------
 uri     | character varying |           | not null |
 id      | character varying |           |          |
 name    | character varying |           |          |
 energy  | numeric(4,3)      |           |          |
 valence | numeric(4,3)      |           |          |
Indexes:
    "tracks_pkey" PRIMARY KEY, btree (uri)
Referenced by:
    TABLE "user_track" CONSTRAINT "user_track_track_uri_fkey" FOREIGN KEY (track_uri) REFERENCES tracks(uri)
```

3) User-Track Table

This is also a temporary table which correlates the Track and User information and collects the tracks related to specific user. When user Logs-out form our application entry of user tracks and user is removed form database.



```
d5kgeb2c174ur3=> \d user_track
                                   Table "public.user_track"
  Column   |       Type        | Collation | Nullable |                Default
-----------+-------------------+-----------+----------+----------------------------------------
 id        | integer           |           | not null | nextval('user_track_id_seq'::regclass)
 user_id   | character varying |           | not null |
 track_uri | character varying |           | not null |
Indexes:
    "user_track_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "user_track_track_uri_fkey" FOREIGN KEY (track_uri) REFERENCES tracks(uri)
    "user_track_user_id_fkey" FOREIGN KEY (user_id) REFERENCES users(id)
```

3.3 Emotion Extraction Module

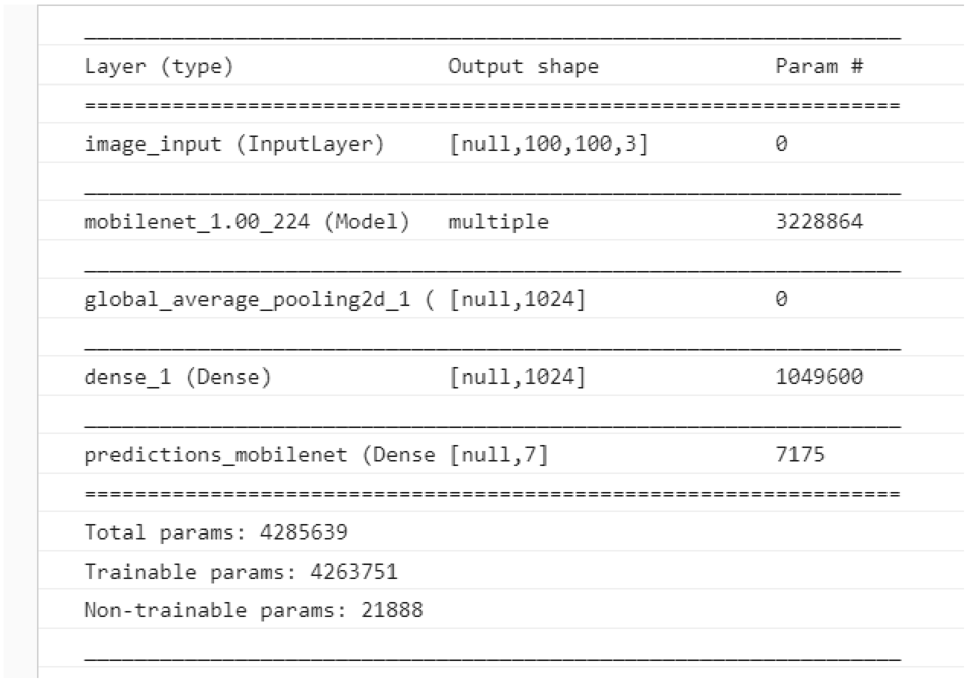
We are using user's camera to capture real time video frames feeding that in preprocessor.

The acquired video frames are converted in 100*100-pixel array of 3 color channel and face is detected. The convolution neural network with transfer learning which uses mobilenetv1 for facial emotion detection. MobileNet model uses depthwise separable convolution layer which helps in extracting information from pixels more accurately. For combining result from all the depthwise separable convolution layer we have used globalaverage pooling layer followed by fully connected layer of 1024 neurons. Final output layer is dense layer of 7 output neurons with softmax activation function to get prediction from among given 7 classes of emotions.

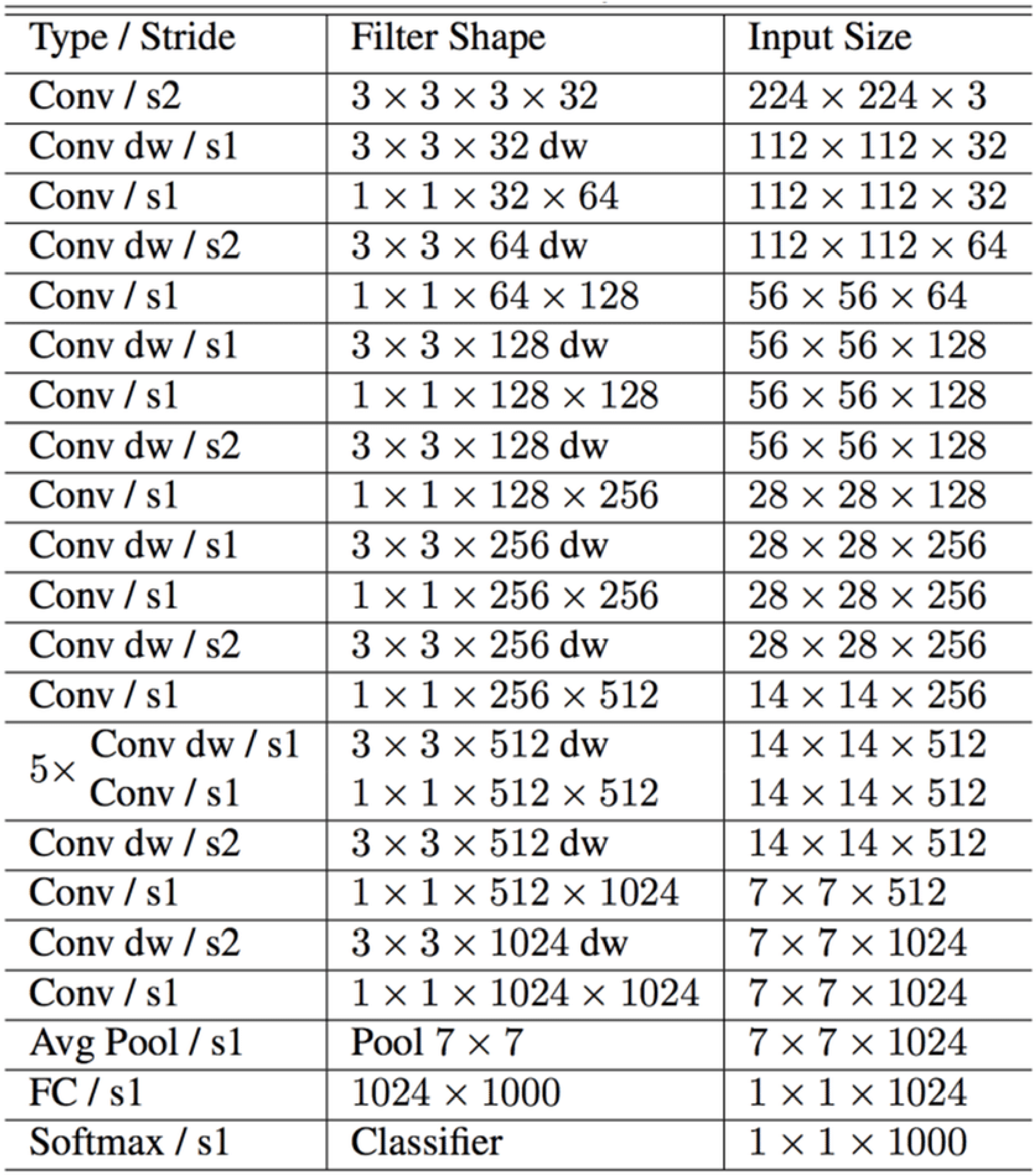
We have used mobile net as base model because of its faster execution and better results on edge devices.

Mobile net architecture is shown below it has 3228864 trainable parameters due to which the model is quite big and may take some time to load first time.

We have converted keras hdf5 model in TensorFlow layers model to provide in browser prediction and users data safety.

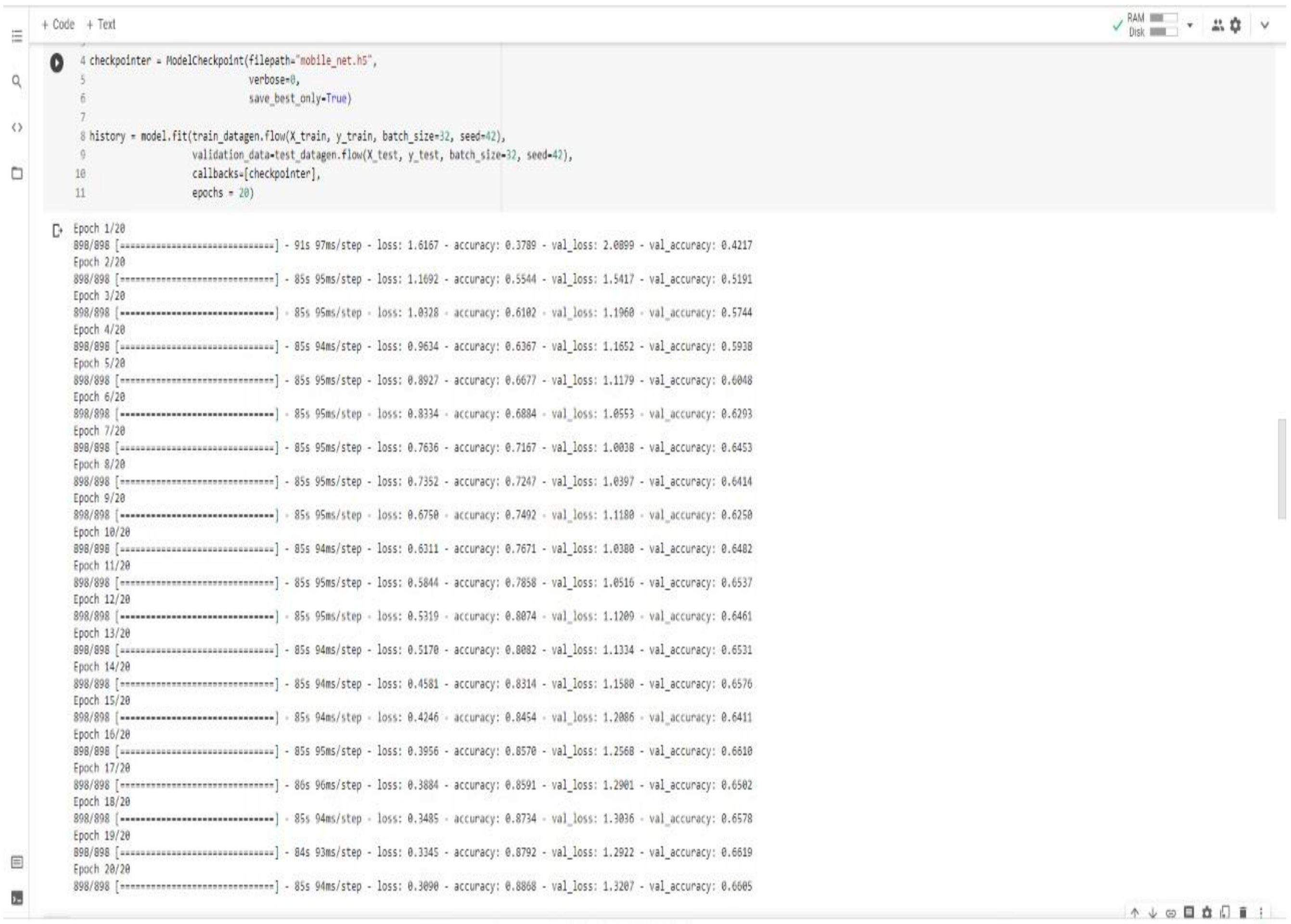

```
_________________________________________________________________
Layer (type)                 Output shape              Param #
=================================================================
image_input (InputLayer)     [null,100,100,3]          0
_________________________________________________________________
mobilenet_1.00_224 (Model)   multiple                  3228864
_________________________________________________________________
global_average_pooling2d_1 ( [null,1024]               0
_________________________________________________________________
dense_1 (Dense)              [null,1024]               1049600
_________________________________________________________________
predictions_mobilenet (Dense [null,7]                  7175
=================================================================
Total params: 4285639
Trainable params: 4263751
Non-trainable params: 21888
_________________________________________________________________
```

Model Summary



Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Mobile Net V1 Internal Architecture (dw stands for Depth Wise Separable Layers)



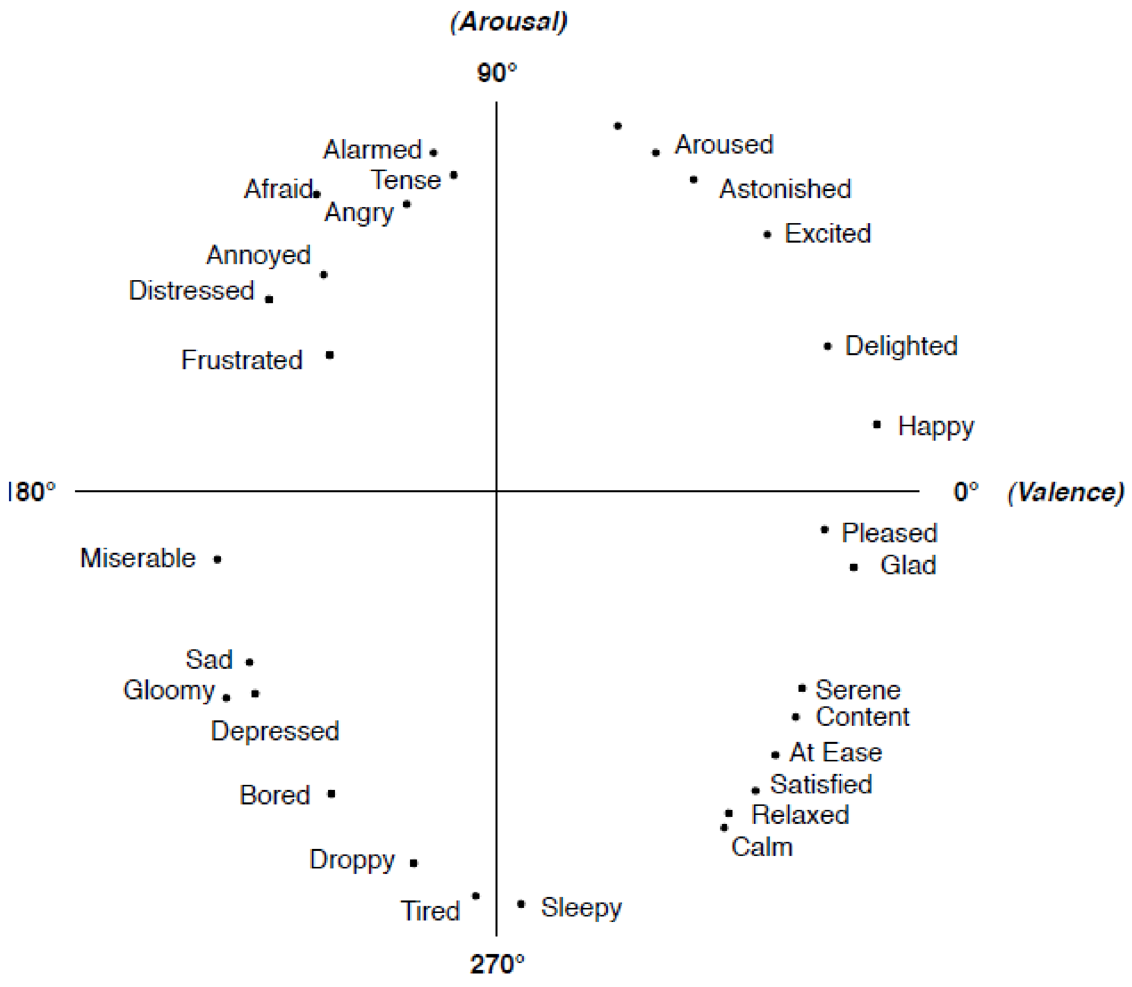
```python
checkpointer = ModelCheckpoint(filepath="mobile_net.h5",
                               verbose=0,
                               save_best_only=True)

history = model.fit(train_datagen.flow(X_train, y_train, batch_size=32, seed=42),
                    validation_data=test_datagen.flow(X_test, y_test, batch_size=32, seed=42),
                    callbacks=[checkpointer],
                    epochs = 20)
```

```
Epoch 1/20
898/898 [==============================] - 91s 97ms/step - loss: 1.6167 - accuracy: 0.3789 - val_loss: 2.0899 - val_accuracy: 0.4217
Epoch 2/20
898/898 [==============================] - 85s 95ms/step - loss: 1.1692 - accuracy: 0.5544 - val_loss: 1.5417 - val_accuracy: 0.5191
Epoch 3/20
898/898 [==============================] - 85s 95ms/step - loss: 1.0328 - accuracy: 0.6102 - val_loss: 1.1960 - val_accuracy: 0.5744
Epoch 4/20
898/898 [==============================] - 85s 94ms/step - loss: 0.9634 - accuracy: 0.6367 - val_loss: 1.1652 - val_accuracy: 0.5938
Epoch 5/20
898/898 [==============================] - 85s 95ms/step - loss: 0.8927 - accuracy: 0.6677 - val_loss: 1.1179 - val_accuracy: 0.6048
Epoch 6/20
898/898 [==============================] - 85s 95ms/step - loss: 0.8334 - accuracy: 0.6884 - val_loss: 1.0553 - val_accuracy: 0.6293
Epoch 7/20
898/898 [==============================] - 85s 95ms/step - loss: 0.7636 - accuracy: 0.7167 - val_loss: 1.0038 - val_accuracy: 0.6453
Epoch 8/20
898/898 [==============================] - 85s 95ms/step - loss: 0.7352 - accuracy: 0.7247 - val_loss: 1.0397 - val_accuracy: 0.6414
Epoch 9/20
898/898 [==============================] - 85s 95ms/step - loss: 0.6750 - accuracy: 0.7492 - val_loss: 1.1180 - val_accuracy: 0.6250
Epoch 10/20
898/898 [==============================] - 85s 94ms/step - loss: 0.6311 - accuracy: 0.7671 - val_loss: 1.0380 - val_accuracy: 0.6482
Epoch 11/20
898/898 [==============================] - 85s 95ms/step - loss: 0.5844 - accuracy: 0.7858 - val_loss: 1.0516 - val_accuracy: 0.6537
Epoch 12/20
898/898 [==============================] - 85s 95ms/step - loss: 0.5319 - accuracy: 0.8074 - val_loss: 1.1209 - val_accuracy: 0.6461
Epoch 13/20
898/898 [==============================] - 85s 94ms/step - loss: 0.5170 - accuracy: 0.8082 - val_loss: 1.1334 - val_accuracy: 0.6531
Epoch 14/20
898/898 [==============================] - 85s 94ms/step - loss: 0.4581 - accuracy: 0.8314 - val_loss: 1.1580 - val_accuracy: 0.6576
Epoch 15/20
898/898 [==============================] - 85s 94ms/step - loss: 0.4246 - accuracy: 0.8454 - val_loss: 1.2086 - val_accuracy: 0.6411
Epoch 16/20
898/898 [==============================] - 85s 95ms/step - loss: 0.3956 - accuracy: 0.8570 - val_loss: 1.2568 - val_accuracy: 0.6610
Epoch 17/20
898/898 [==============================] - 86s 96ms/step - loss: 0.3884 - accuracy: 0.8591 - val_loss: 1.2901 - val_accuracy: 0.6502
Epoch 18/20
898/898 [==============================] - 85s 94ms/step - loss: 0.3485 - accuracy: 0.8734 - val_loss: 1.3036 - val_accuracy: 0.6578
Epoch 19/20
898/898 [==============================] - 84s 93ms/step - loss: 0.3345 - accuracy: 0.8792 - val_loss: 1.2922 - val_accuracy: 0.6619
Epoch 20/20
898/898 [==============================] - 85s 94ms/step - loss: 0.3090 - accuracy: 0.8868 - val_loss: 1.3207 - val_accuracy: 0.6605
```

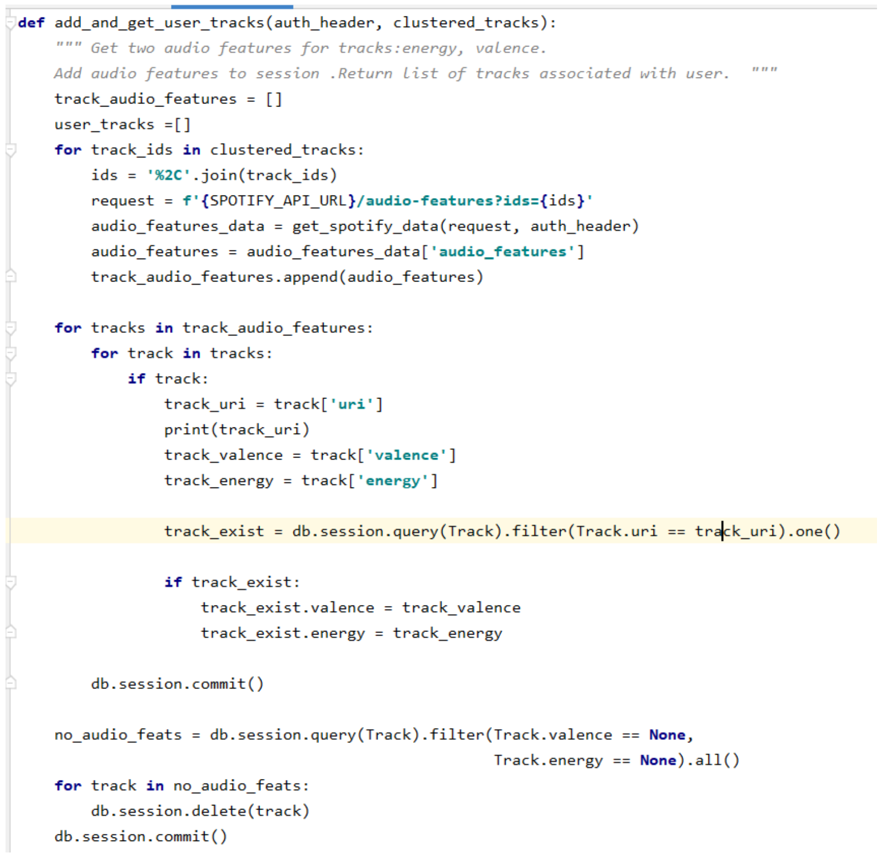
Training the custom model using Pretrained Model as Base

3.4 Playlist Generation

We have used Russell's emotion model to map the emotion classes to the song's audio feature energy and valence, collected using Spotify API. We have standardized the audio feature values to map them in a particular range. Energy and Valence feature is now inn [0,1] range so we have mapped the negative and positive values in this range. Then the emotion class is used to create the playlist dynamically and displayed to user.

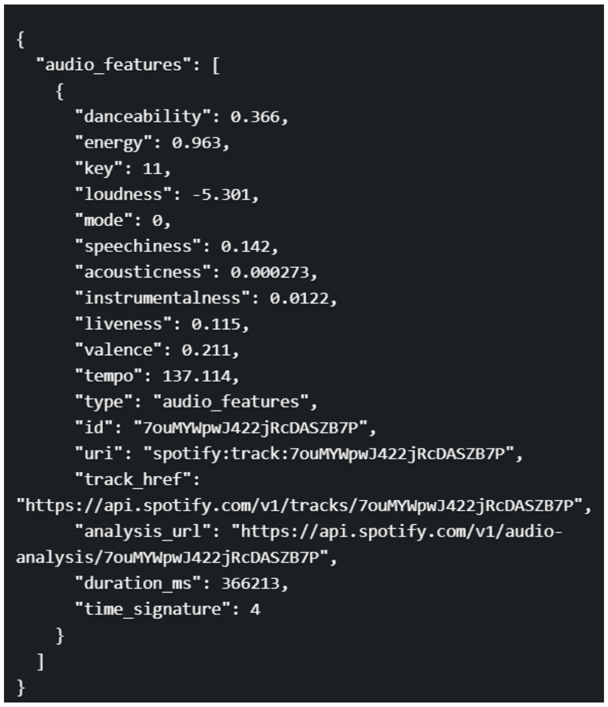


Russell's emotion model



```python
def add_and_get_user_tracks(auth_header, clustered_tracks):
    """ Get two audio features for tracks:energy, valence.
    Add audio features to session .Return list of tracks associated with user.  """
    track_audio_features = []
    user_tracks =[]
    for track_ids in clustered_tracks:
        ids = '%2C'.join(track_ids)
        request = f'{SPOTIFY_API_URL}/audio-features?ids={ids}'
        audio_features_data = get_spotify_data(request, auth_header)
        audio_features = audio_features_data['audio_features']
        track_audio_features.append(audio_features)

    for tracks in track_audio_features:
        for track in tracks:
            if track:
                track_uri = track['uri']
                print(track_uri)
                track_valence = track['valence']
                track_energy = track['energy']

                track_exist = db.session.query(Track).filter(Track.uri == track_uri).one()

                if track_exist:
                    track_exist.valence = track_valence
                    track_exist.energy = track_energy

        db.session.commit()

    no_audio_feats = db.session.query(Track).filter(Track.valence == None,
                                                    Track.energy == None).all()
    for track in no_audio_feats:
        db.session.delete(track)
    db.session.commit()
```

Getting Audio Feature and Adding them in Tracks Table

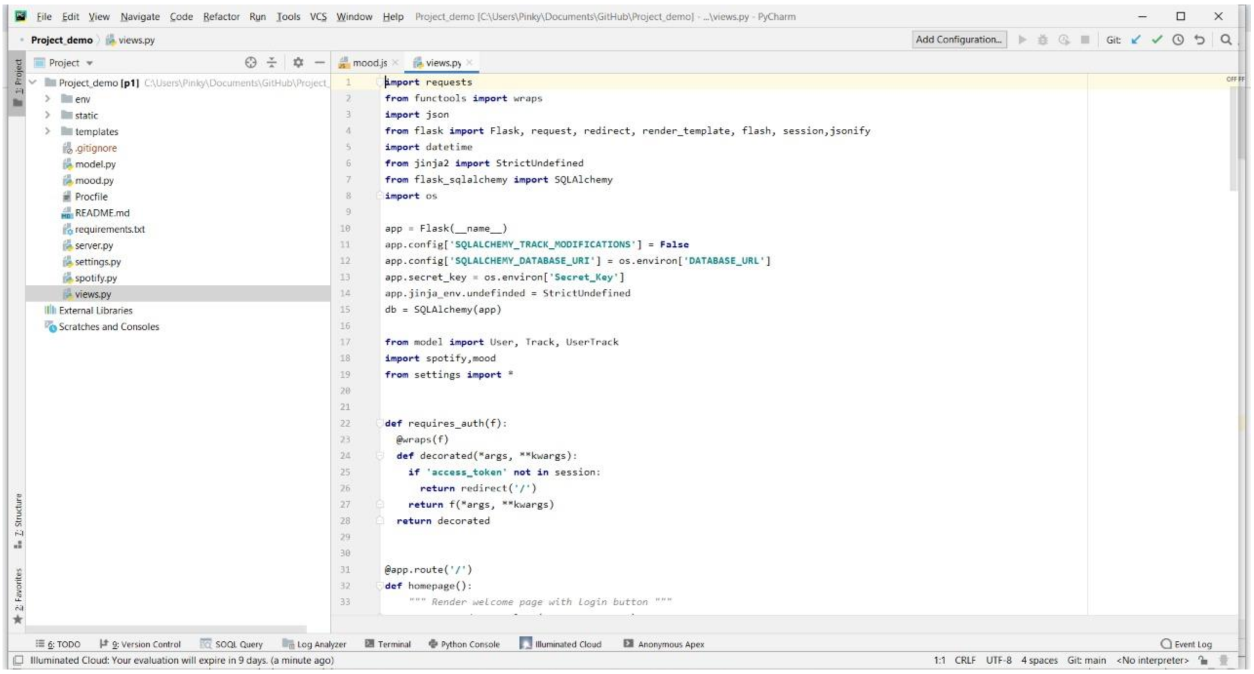


```
{
  "audio_features": [
    {
      "danceability": 0.366,
      "energy": 0.963,
      "key": 11,
      "loudness": -5.301,
      "mode": 0,
      "speechiness": 0.142,
      "acousticness": 0.000273,
      "instrumentalness": 0.0122,
      "liveness": 0.115,
      "valence": 0.211,
      "tempo": 137.114,
      "type": "audio_features",
      "id": "7ouMYWpwJ422jRcDASZB7P",
      "uri": "spotify:track:7ouMYWpwJ422jRcDASZB7P",
      "track_href":
"https://api.spotify.com/v1/tracks/7ouMYWpwJ422jRcDASZB7P",
      "analysis_url": "https://api.spotify.com/v1/audio-
analysis/7ouMYWpwJ422jRcDASZB7P",
      "duration_ms": 366213,
      "time_signature": 4
    }
  ]
}
```

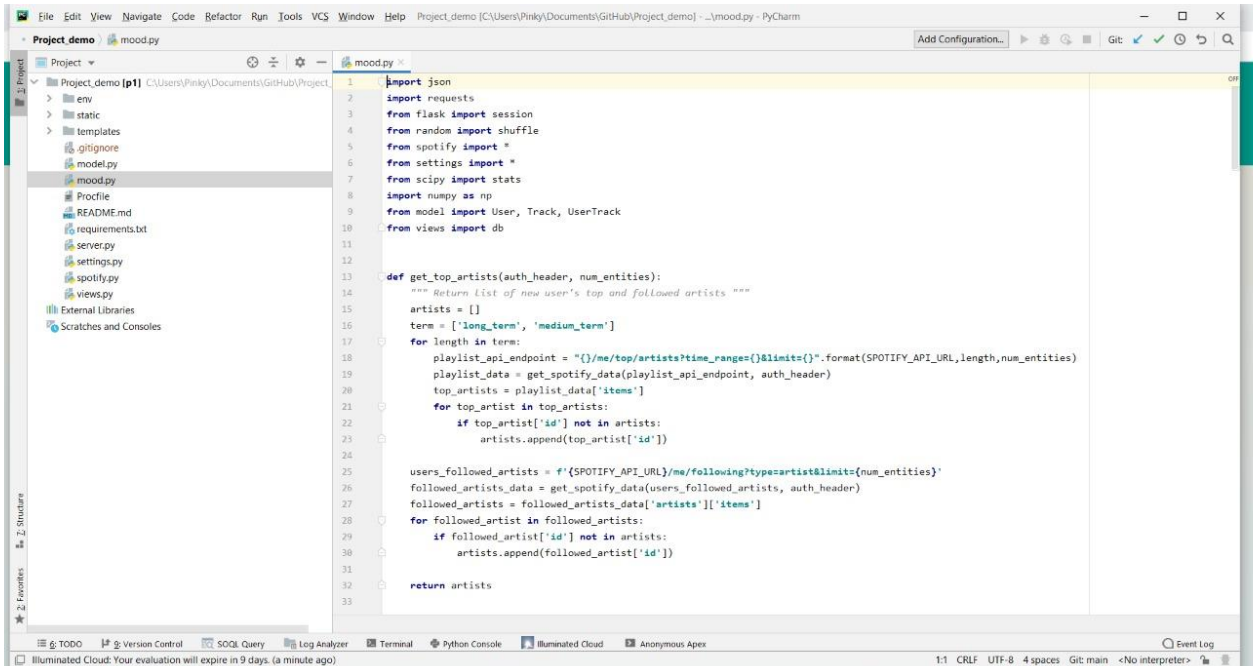
All the features of Audio

3.5 Actual Implementation

Flask Application

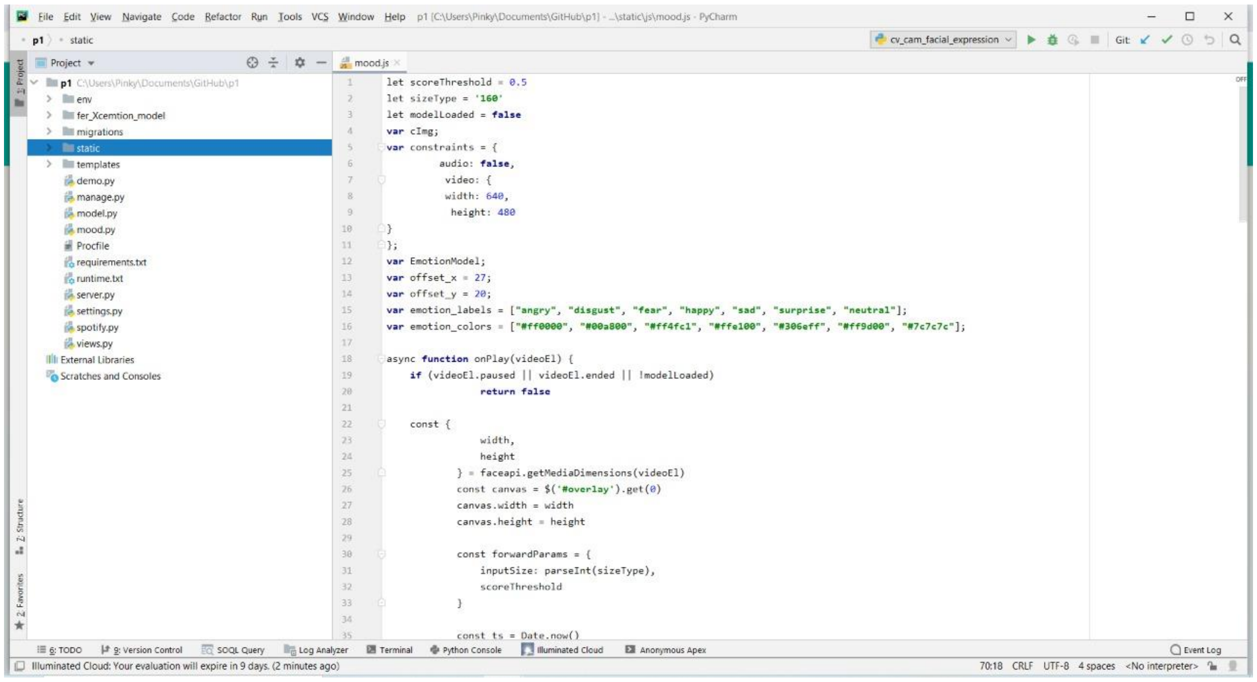


```python
import requests
from functools import wraps
import json
from flask import Flask, request, redirect, render_template, flash, session,jsonify
import datetime
from jinja2 import StrictUndefined
from flask_sqlalchemy import SQLAlchemy
import os

app = Flask(__name__)
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SQLALCHEMY_DATABASE_URI'] = os.environ['DATABASE_URL']
app.secret_key = os.environ['Secret_Key']
app.jinja_env.undefinded = StrictUndefined
db = SQLAlchemy(app)

from model import User, Track, UserTrack
import spotify,mood
from settings import *


def requires_auth(f):
  @wraps(f)
  def decorated(*args, **kwargs):
    if 'access_token' not in session:
      return redirect('/')
    return f(*args, **kwargs)
  return decorated


@app.route('/')
def homepage():
    """ Render welcome page with login button """
```

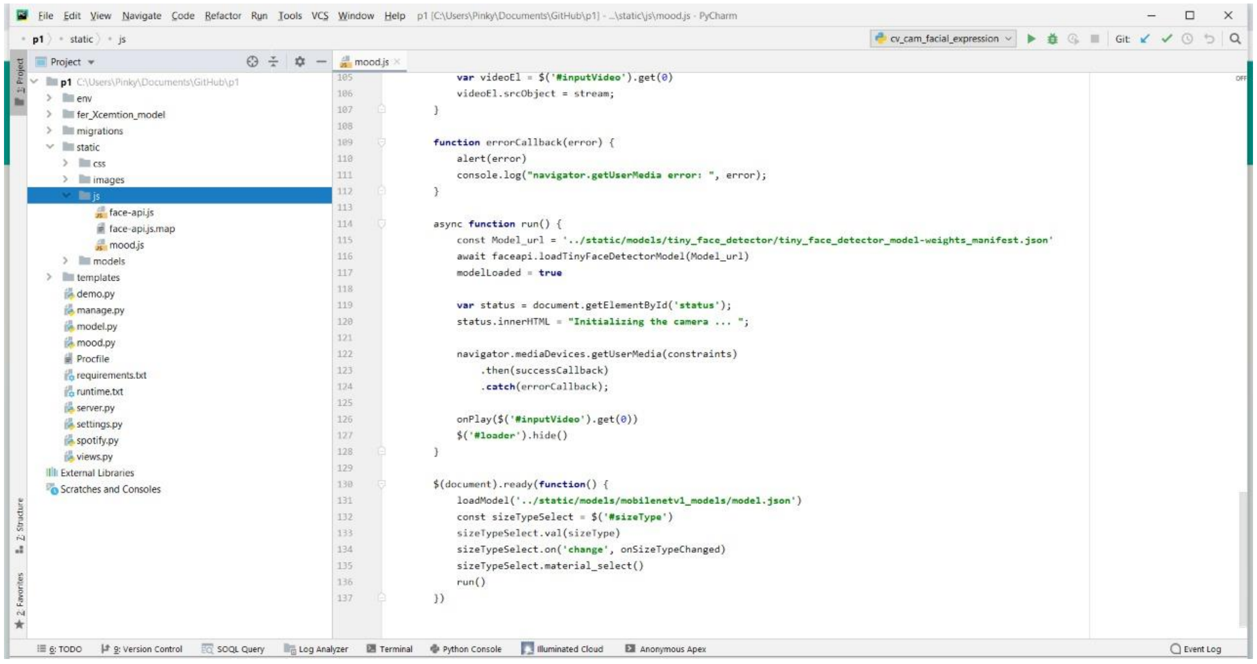


```python
import json
import requests
from flask import session
from random import shuffle
from spotify import *
from settings import *
from scipy import stats
import numpy as np
from model import User, Track, UserTrack
from views import db


def get_top_artists(auth_header, num_entities):
    """ Return list of new user's top and followed artists """
    artists = []
    term = ['long_term', 'medium_term']
    for length in term:
        playlist_api_endpoint = "{}/me/top/artists?time_range={}&limit={}".format(SPOTIFY_API_URL,length,num_entities)
        playlist_data = get_spotify_data(playlist_api_endpoint, auth_header)
        top_artists = playlist_data['items']
        for top_artist in top_artists:
            if top_artist['id'] not in artists:
                artists.append(top_artist['id'])

    users_followed_artists = f'{SPOTIFY_API_URL}/me/following?type=artist&limit={num_entities}'
    followed_artists_data = get_spotify_data(users_followed_artists, auth_header)
    followed_artists = followed_artists_data['artists']['items']
    for followed_artist in followed_artists:
        if followed_artist['id'] not in artists:
            artists.append(followed_artist['id'])

    return artists

```

JavaScript code For Expression Classification

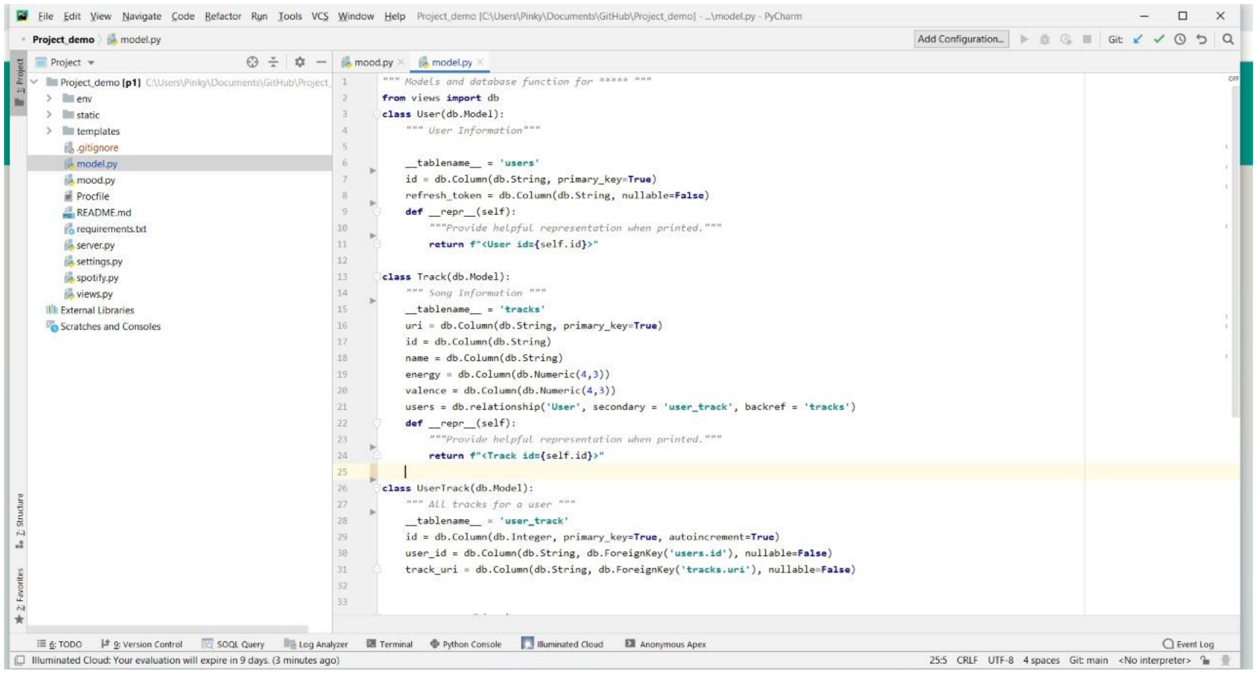


```
let scoreThreshold = 0.5
let sizeType = '160'
let modelLoaded = false
var cImg;
var constraints = {
        audio: false,
         video: {
         width: 640,
          height: 480
}
};
var EmotionModel;
var offset_x = 27;
var offset_y = 20;
var emotion_labels = ["angry", "disgust", "fear", "happy", "sad", "surprise", "neutral"];
var emotion_colors = ["#ff0000", "#00a800", "#ff4fc1", "#ffe100", "#306eff", "#ff9d00", "#7c7c7c"];

async function onPlay(videoEl) {
    if (videoEl.paused || videoEl.ended || !modelLoaded)
            return false

    const {
            width,
            height
        } = faceapi.getMediaDimensions(videoEl)
        const canvas = $('#overlay').get(0)
        canvas.width = width
        canvas.height = height

        const forwardParams = {
            inputSize: parseInt(sizeType),
            scoreThreshold
        }

        const ts = Date.now()
```

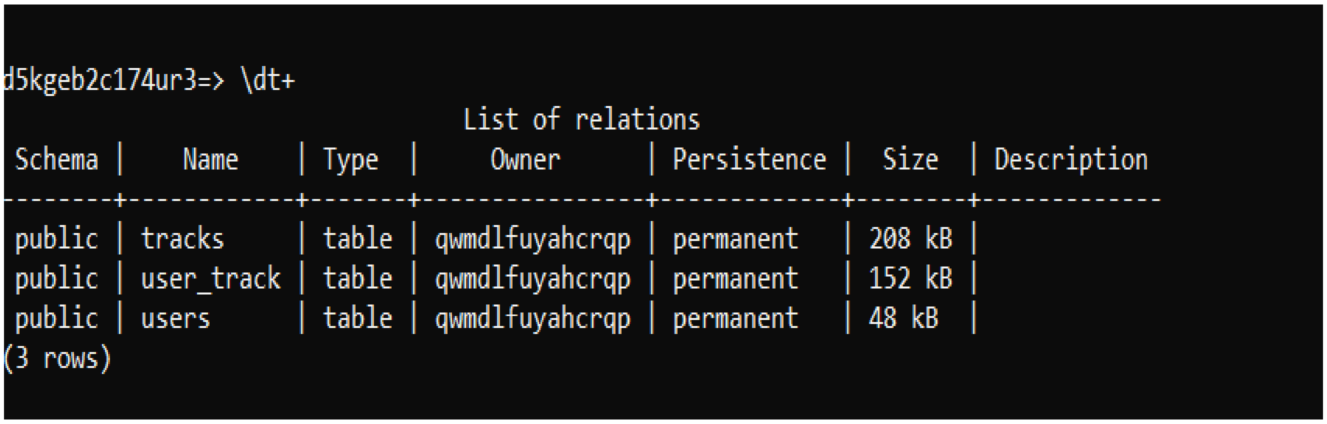


```
        var videoEl = $('#inputVideo').get(0)
        videoEl.srcObject = stream;
    }

    function errorCallback(error) {
        alert(error)
        console.log("navigator.getUserMedia error: ", error);
    }

    async function run() {
        const Model_url = '../static/models/tiny_face_detector/tiny_face_detector_model-weights_manifest.json'
        await faceapi.loadTinyFaceDetectorModel(Model_url)
        modelLoaded = true

        var status = document.getElementById('status');
        status.innerHTML = "Initializing the camera ... ";

        navigator.mediaDevices.getUserMedia(constraints)
            .then(successCallback)
            .catch(errorCallback);

        onPlay($('#inputVideo').get(0))
        $('#loader').hide()
    }

    $(document).ready(function() {
        loadModel('../static/models/mobilenetv1_models/model.json')
        const sizeTypeSelect = $('#sizeType')
        sizeTypeSelect.val(sizeType)
        sizeTypeSelect.on('change', onSizeTypeChanged)
        sizeTypeSelect.material_select()
        run()
    })
```

Database Creation Class

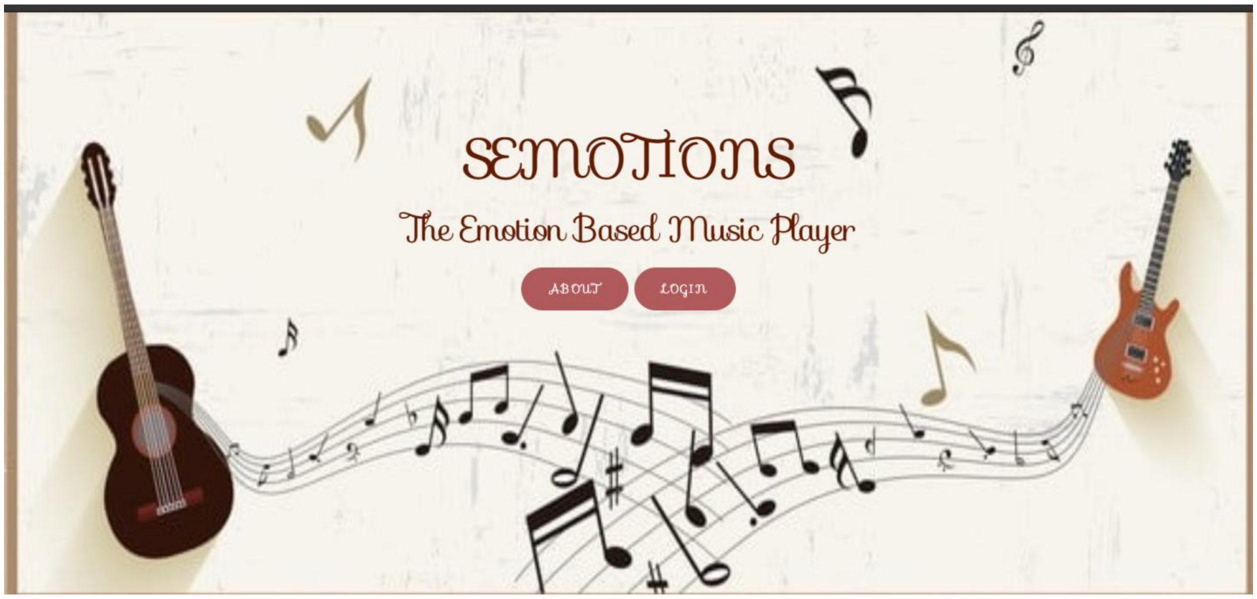


```
""" Models and database function for ***** """
from views import db
class User(db.Model):
    """ User Information"""

    __tablename__ = 'users'
    id = db.Column(db.String, primary_key=True)
    refresh_token = db.Column(db.String, nullable=False)
    def __repr__(self):
        """Provide helpful representation when printed."""
        return f"<User id={self.id}>"

class Track(db.Model):
    """ Song Information """
    __tablename__ = 'tracks'
    uri = db.Column(db.String, primary_key=True)
    id = db.Column(db.String)
    name = db.Column(db.String)
    energy = db.Column(db.Numeric(4,3))
    valence = db.Column(db.Numeric(4,3))
    users = db.relationship('User', secondary = 'user_track', backref = 'tracks')
    def __repr__(self):
        """Provide helpful representation when printed."""
        return f"<Track id={self.id}>"

class UserTrack(db.Model):
    """ All tracks for a user """
    __tablename__ = 'user_track'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    user_id = db.Column(db.String, db.ForeignKey('users.id'), nullable=False)
    track_uri = db.Column(db.String, db.ForeignKey('tracks.uri'), nullable=False)
```

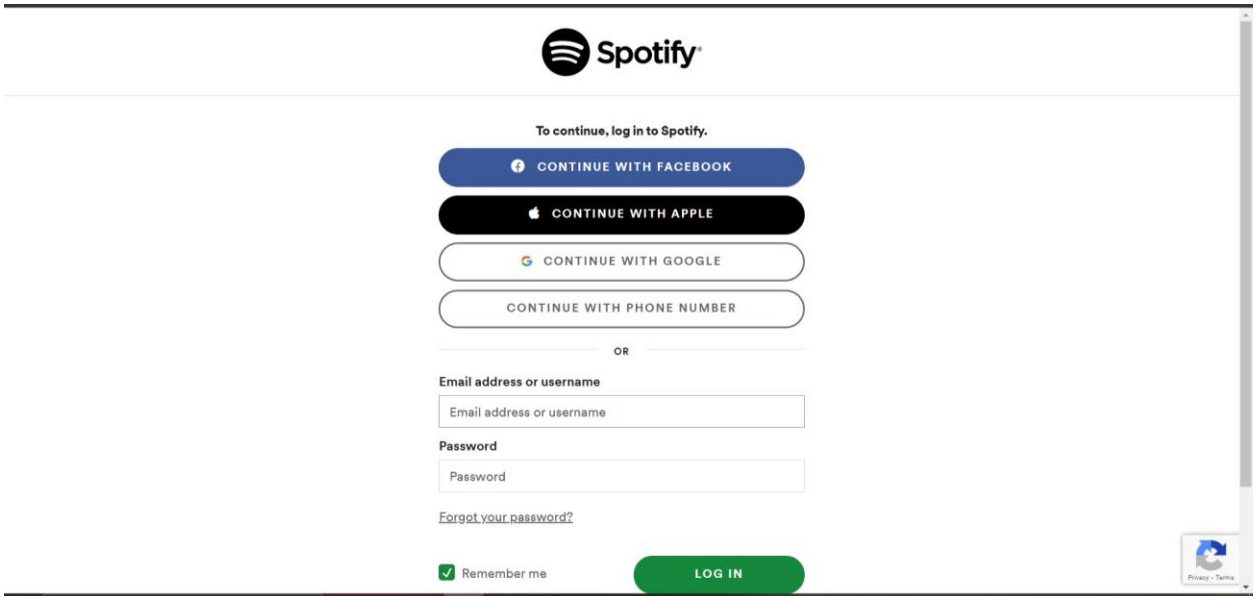


```
d5kgeb2c174ur3=> \dt+
                              List of relations
 Schema |    Name    | Type  |     Owner      | Persistence |  Size  | Description
--------+------------+-------+----------------+-------------+--------+-------------
 public | tracks     | table | qwmdlfuyahcrqp | permanent   | 208 kB |
 public | user_track | table | qwmdlfuyahcrqp | permanent   | 152 kB |
 public | users      | table | qwmdlfuyahcrqp | permanent   | 48 kB  |
(3 rows)
```

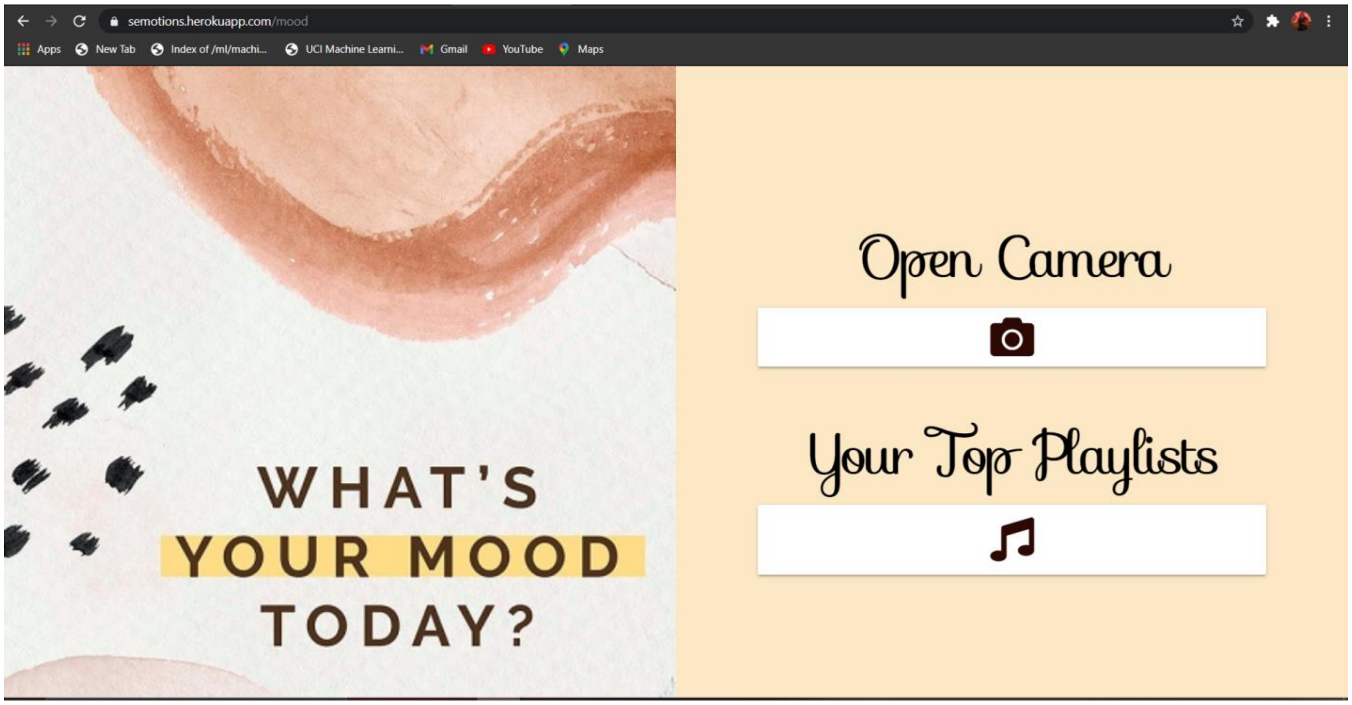
3.6 Application Result

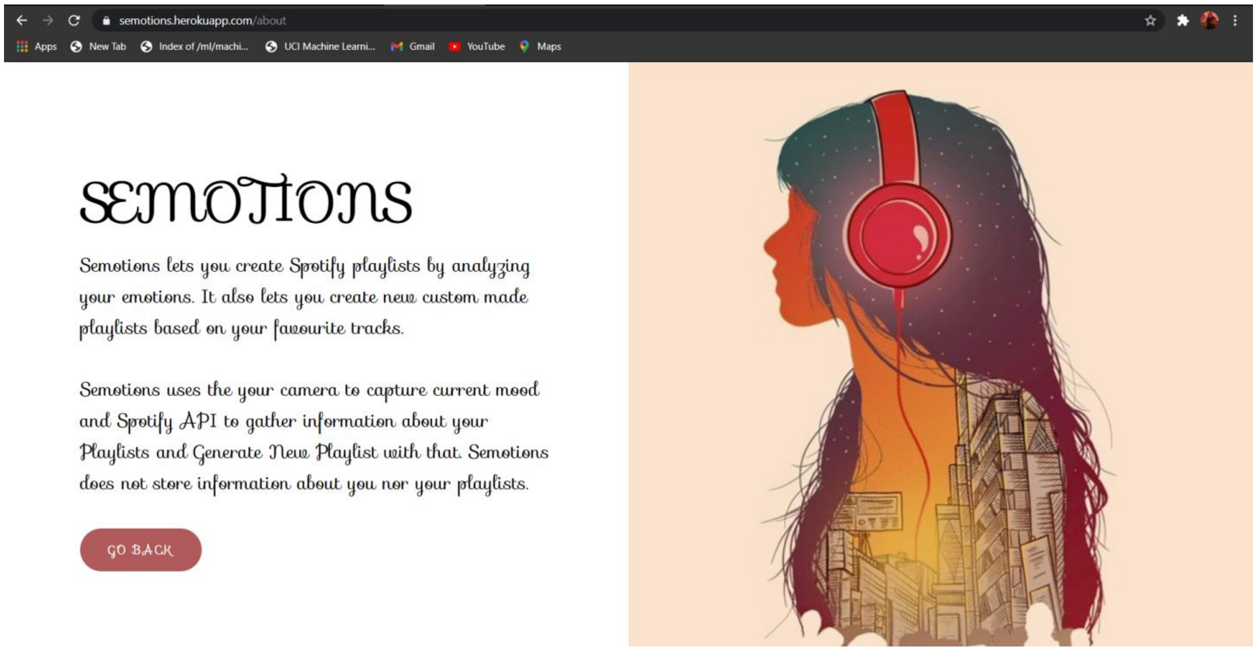


Home Page



User Login





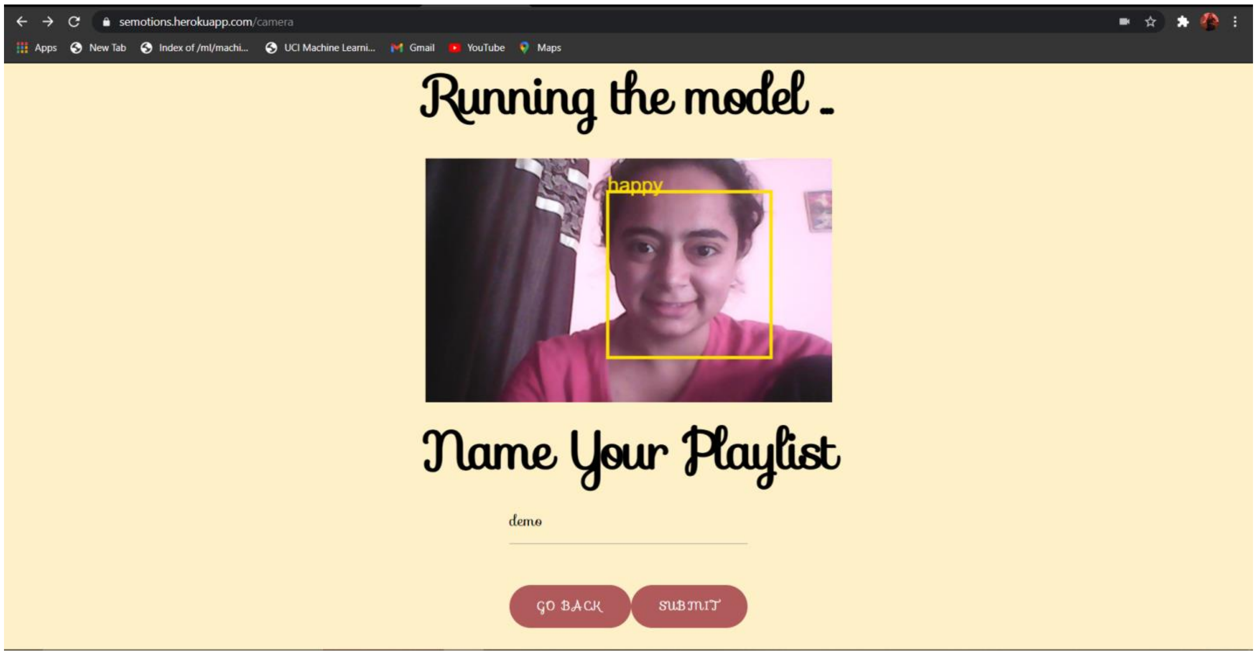
Semotions lets you create Spotify playlists by analyzing your emotions. It also lets you create new custom made playlists based on your favourite tracks.

Semotions uses the your camera to capture current mood and Spotify API to gather information about your Playlists and Generate New Playlist with that. Semotions does not store information about you nor your playlists.

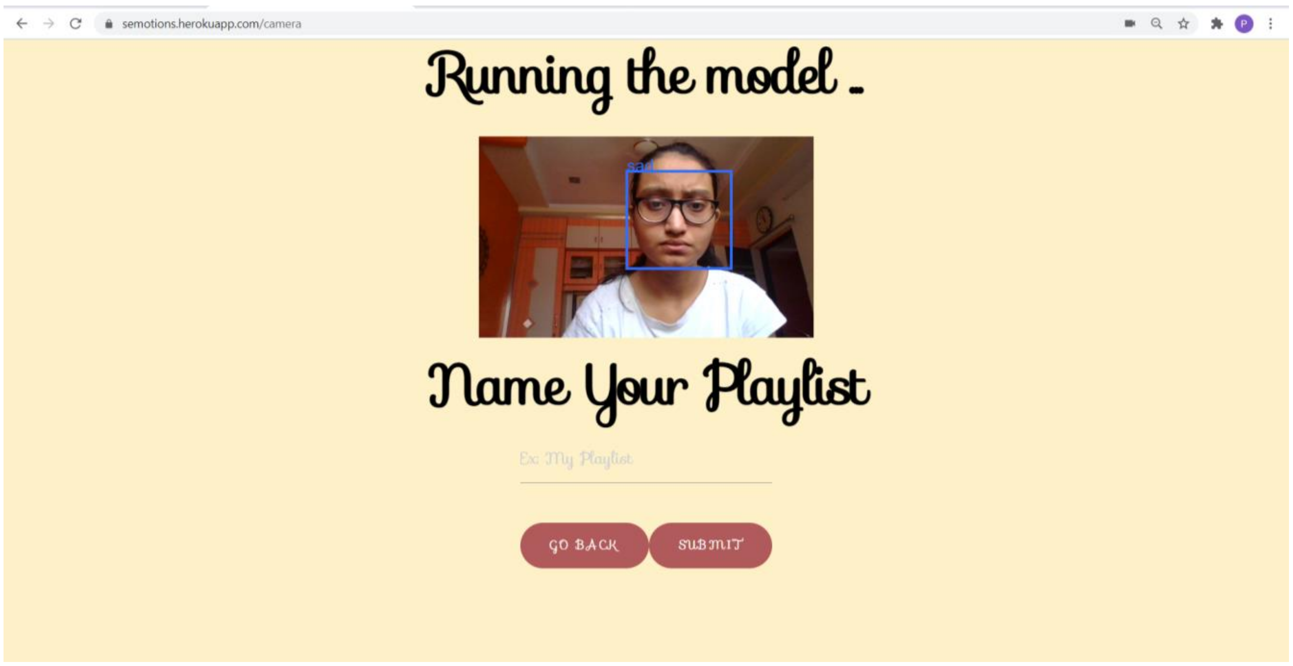
Application Information

3.7 Testing and Verification

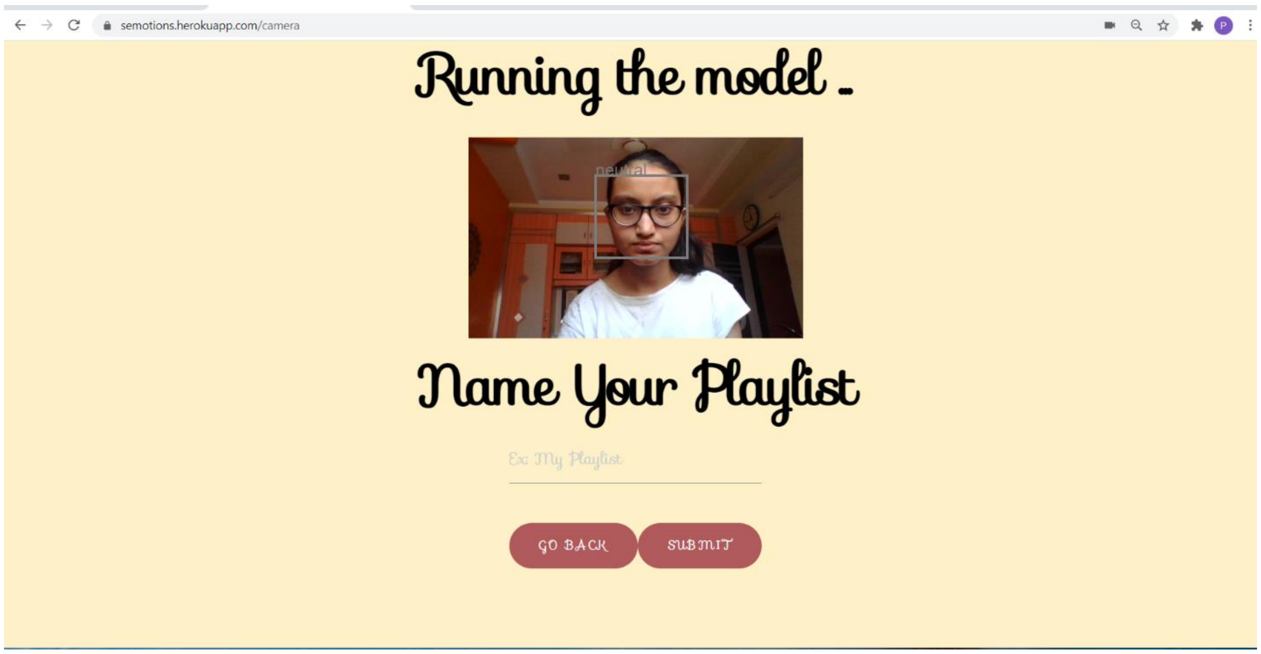
Emotion Classifier Results



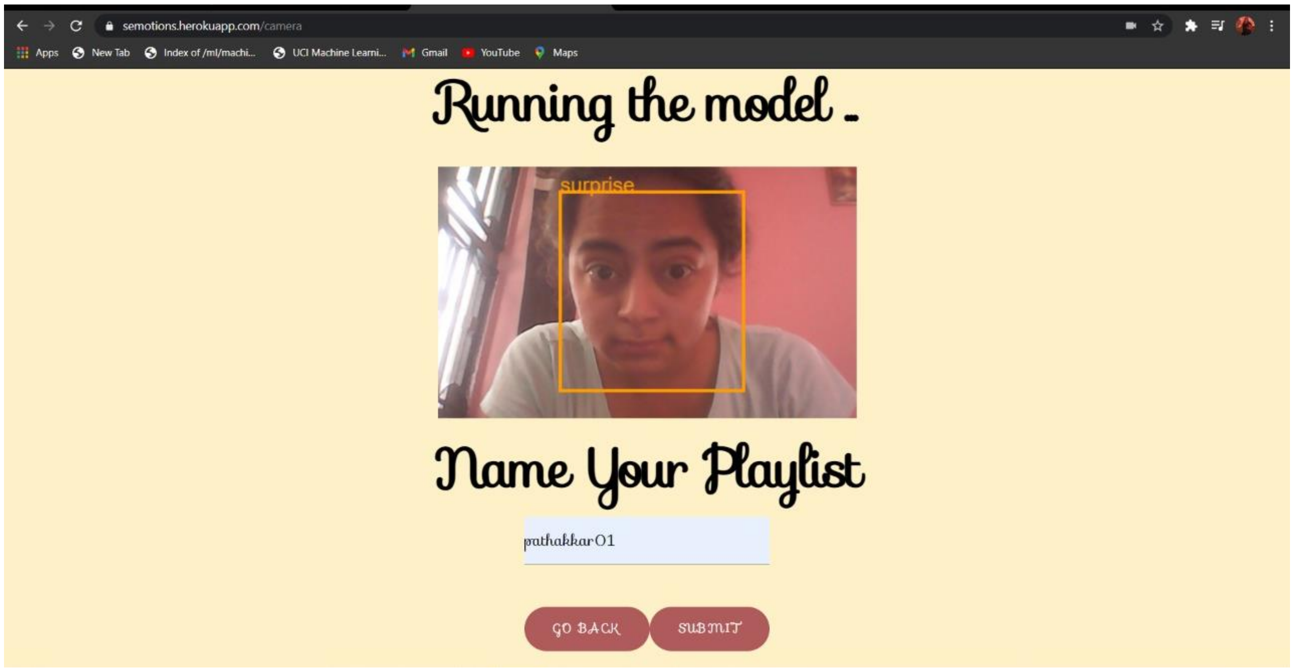
Predicted **Happy** Emotion Correctly



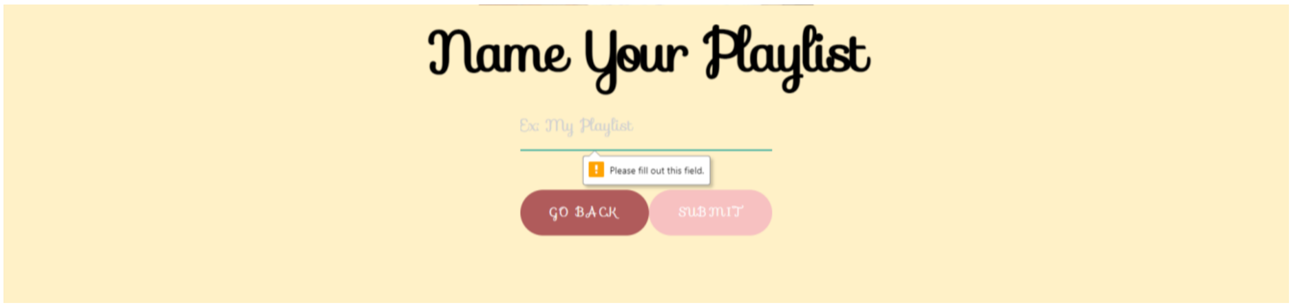
Predicted **Sad** Emotion correctly



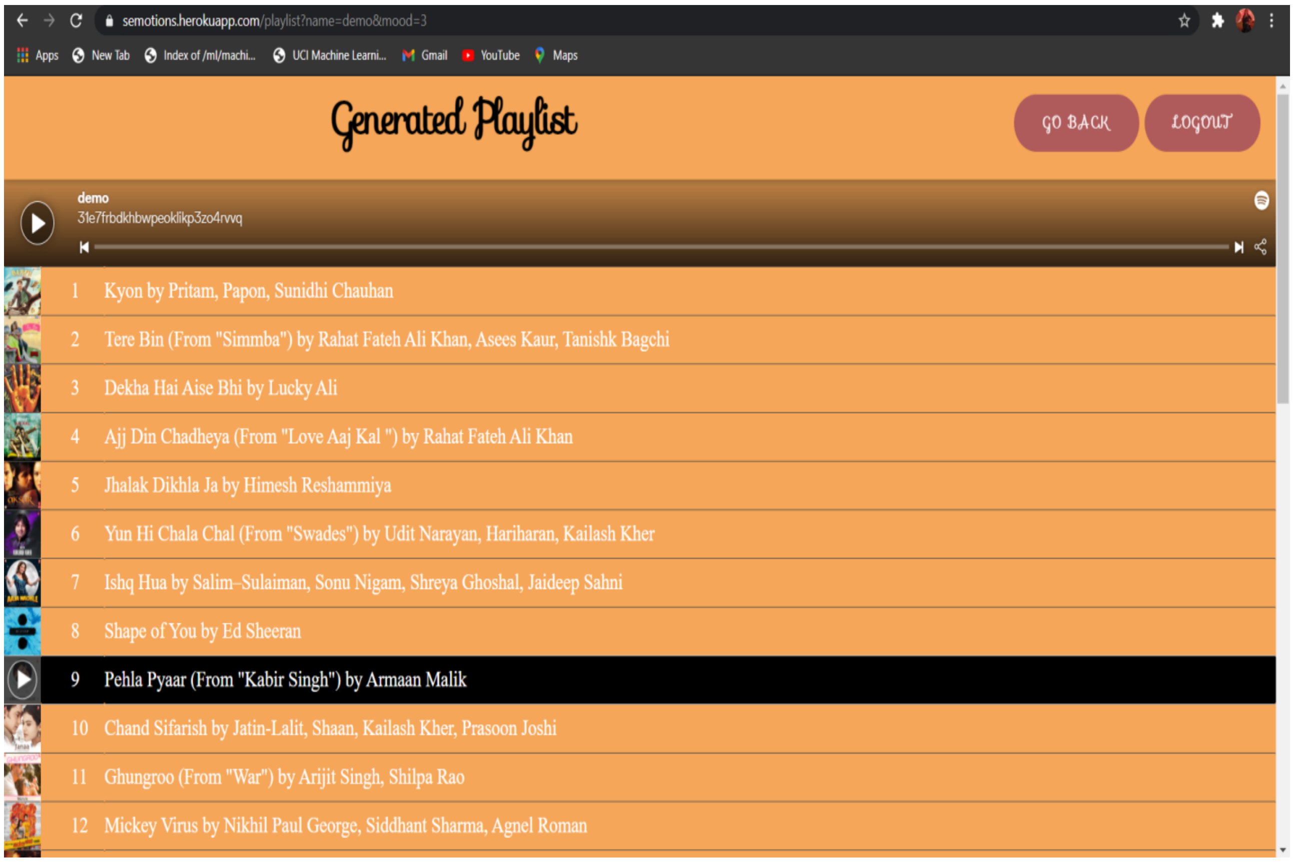
Predicted **Neutral** Emotion correctly



Predicted **Surprise** Emotion Correctly

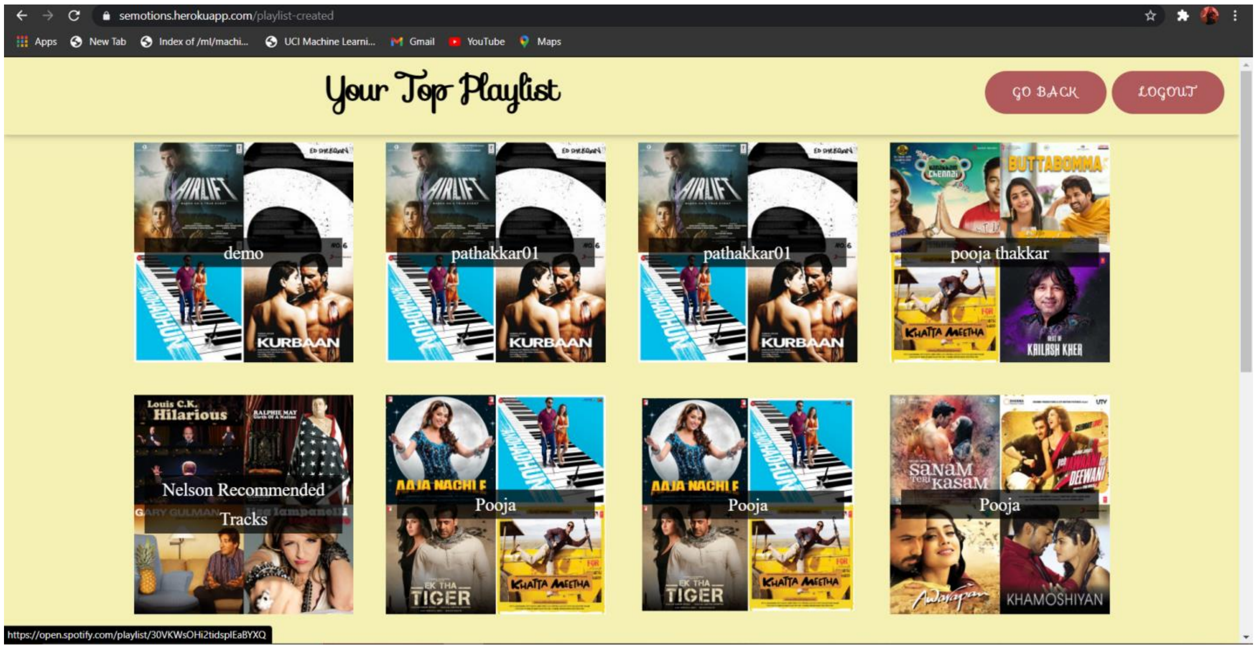


Missing Field Validation

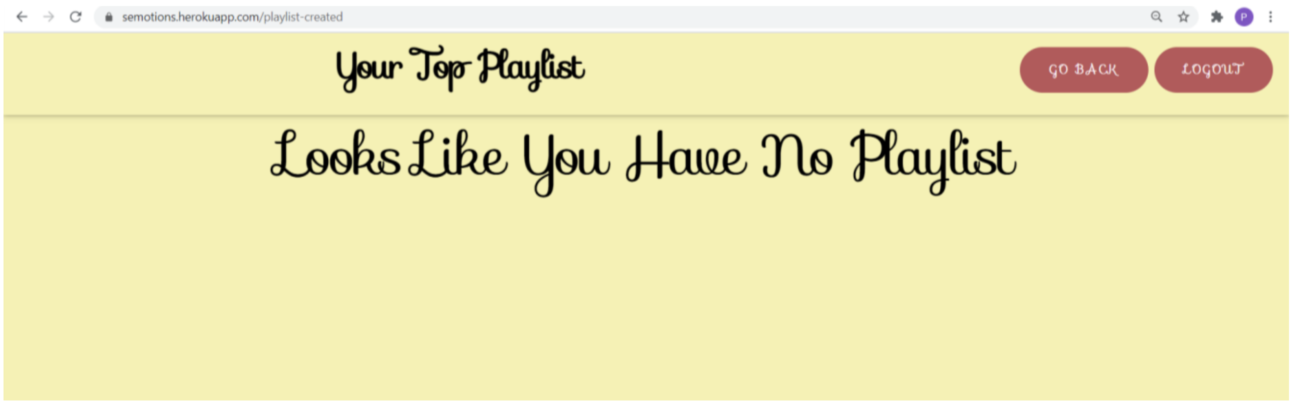


1 Kyon by Pritam, Papon, Sunidhi Chauhan
2 Tere Bin (From "Simmba") by Rahat Fateh Ali Khan, Asees Kaur, Tanishk Bagchi
3 Dekha Hai Aise Bhi by Lucky Ali
4 Ajj Din Chadheya (From "Love Aaj Kal ") by Rahat Fateh Ali Khan
5 Jhalak Dikhla Ja by Himesh Reshammiya
6 Yun Hi Chala Chal (From "Swades") by Udit Narayan, Hariharan, Kailash Kher
7 Ishq Hua by Salim–Sulaiman, Sonu Nigam, Shreya Ghoshal, Jaideep Sahni
8 Shape of You by Ed Sheeran
9 Pehla Pyaar (From "Kabir Singh") by Armaan Malik
10 Chand Sifarish by Jatin-Lalit, Shaan, Kailash Kher, Prasoon Joshi
11 Ghungroo (From "War") by Arijit Singh, Shilpa Rao
12 Mickey Virus by Nikhil Paul George, Siddhant Sharma, Agnel Roman

Recommendation Songs Playlist with Embedded Player Generated Successfully


semotions.herokuapp.com/playlist-created
Your Top Playlist
GO BACK
LOGOUT
demo
pathakkar01
pathakkar01
pooja thakkar
Nelson Recommended Tracks
Pooja
Pooja
Pooja
https://open.spotify.com/playlist/30VKWsOHi2tidspIEaBYXQ

Displaying Top Playlist for Regular User


semotions.herokuapp.com/playlist-created
Your Top Playlist
GO BACK
LOGOUT
Looks Like You Have No Playlist

For New user who just Sign-Up Displaying this due to no previous record of recently playlists

SECTION 4

SUMMARY OF THE RESULT

4.1 Conclusion

The Semotions is used to automate and give a better music player experience for the end user. The application solves the basic needs of music listeners without troubling them as existing applications do: it uses technology to increase the interaction of the system with the user in many ways. It eases the work of the end-user by capturing the image using a camera, determining their emotion, and suggesting a customized play-list through a more advanced and interactive system. The user did not need to store any song in their system as everything will be done by our application.

4.2 Advantage of Result

- It is helpful in emotion capturing and generating insight from that.
- It will take less time than manual playlist generation.
- Emotion Classification model can be used for any other application also.
- Personalized Music Playlist without human interaction.
- A smooth interface between Spotify web player and user for emotion-based song playing.

4.3 Scope of Future work

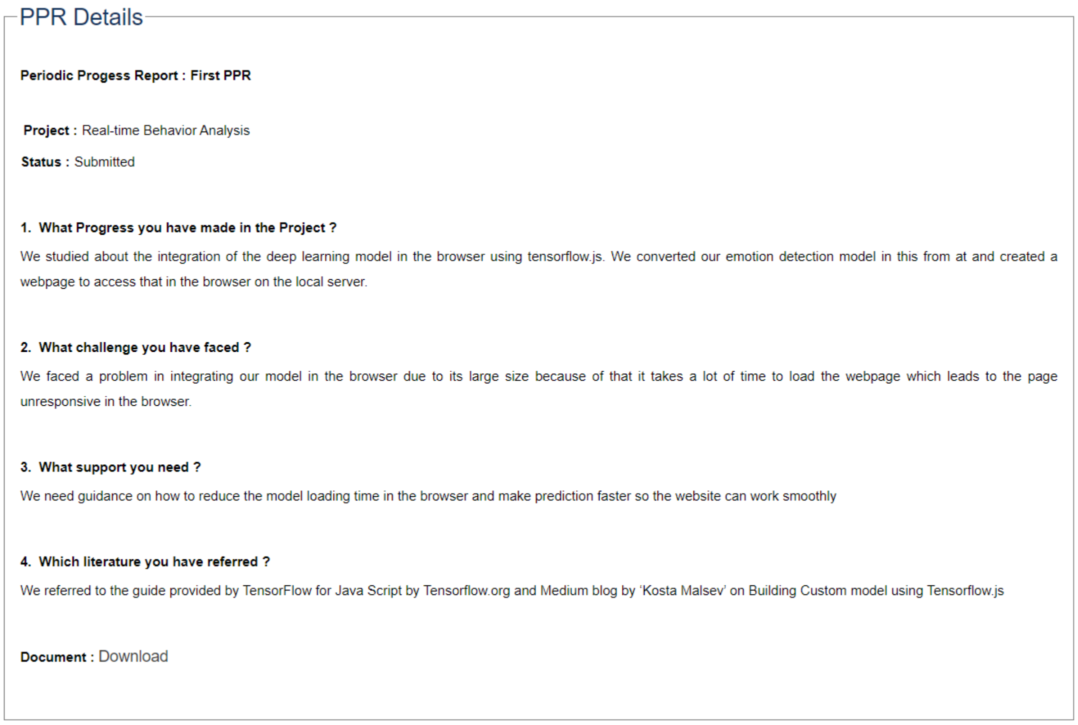
The application can be improved by modifying and adding few functionalities.

- Use other Light Weight model.
- Optimizing the Emotion Detection Algorithm by including additional features which helps system to categorize user based on many other factors like location and suggesting the user to travel to that location and play songs accordingly
- Improve the accuracy in detected emotion without any bias towards any particular feature.

SECTION 5

Appendix

5.1 PPR



PPR Details

Periodic Progess Report : First PPR

Project : Real-time Behavior Analysis

Status : Submitted

1. What Progress you have made in the Project ?

We studied about the integration of the deep learning model in the browser using tensorflow.js. We converted our emotion detection model in this from at and created a webpage to access that in the browser on the local server.

2. What challenge you have faced ?

We faced a problem in integrating our model in the browser due to its large size because of that it takes a lot of time to load the webpage which leads to the page unresponsive in the browser.

3. What support you need ?

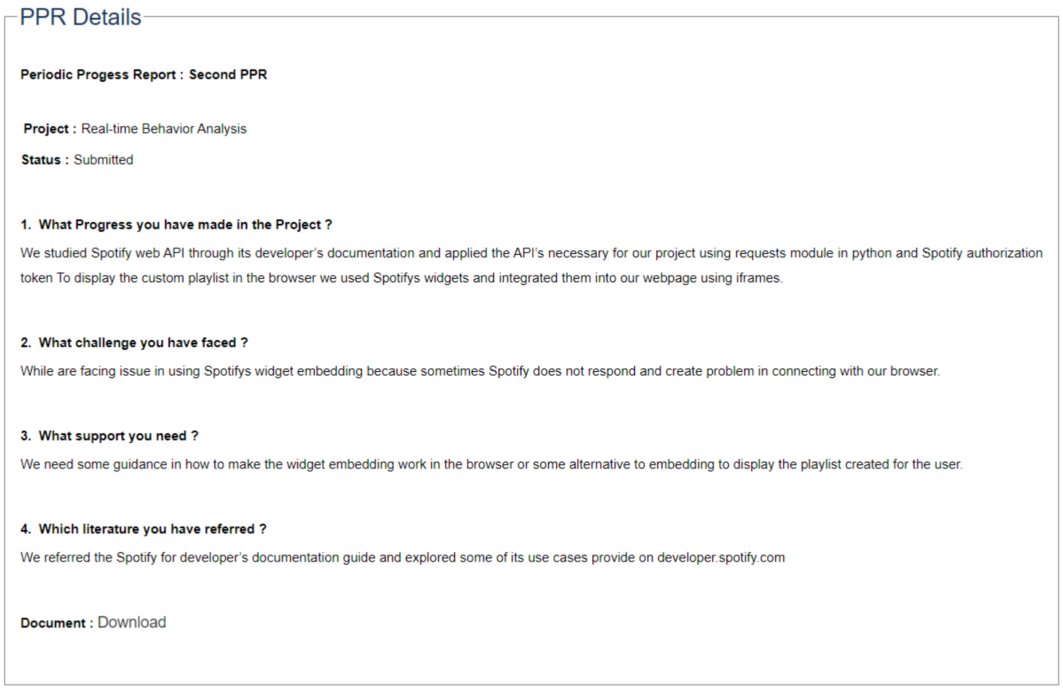
We need guidance on how to reduce the model loading time in the browser and make prediction faster so the website can work smoothly

4. Which literature you have referred ?

We referred to the guide provided by TensorFlow for Java Script by Tensorflow.org and Medium blog by 'Kosta Malsev' on Building Custom model using Tensorflow.js

Document : Download

PPR 1



PPR Details

Periodic Progess Report : Second PPR

Project : Real-time Behavior Analysis

Status : Submitted

1. What Progress you have made in the Project ?

We studied Spotify web API through its developer's documentation and applied the API's necessary for our project using requests module in python and Spotify authorization token To display the custom playlist in the browser we used Spotifys widgets and integrated them into our webpage using iframes.

2. What challenge you have faced ?

While are facing issue in using Spotifys widget embedding because sometimes Spotify does not respond and create problem in connecting with our browser.

3. What support you need ?

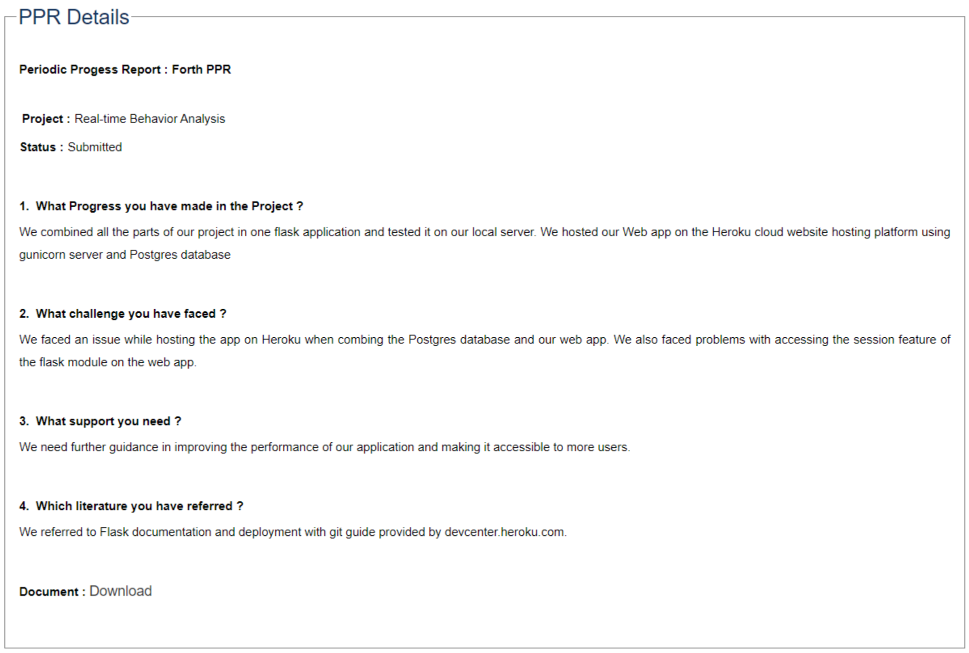
We need some guidance in how to make the widget embedding work in the browser or some alternative to embedding to display the playlist created for the user.

4. Which literature you have referred ?

We referred the Spotify for developer's documentation guide and explored some of its use cases provide on developer.spotify.com

Document : Download

PPR 2



PPR Details

Periodic Progess Report : Forth PPR

Project : Real-time Behavior Analysis

Status : Submitted

1. **What Progress you have made in the Project ?**

We combined all the parts of our project in one flask application and tested it on our local server. We hosted our Web app on the Heroku cloud website hosting platform using gunicorn server and Postgres database

2. **What challenge you have faced ?**

We faced an issue while hosting the app on Heroku when combing the Postgres database and our web app. We also faced problems with accessing the session feature of the flask module on the web app.

3. **What support you need ?**

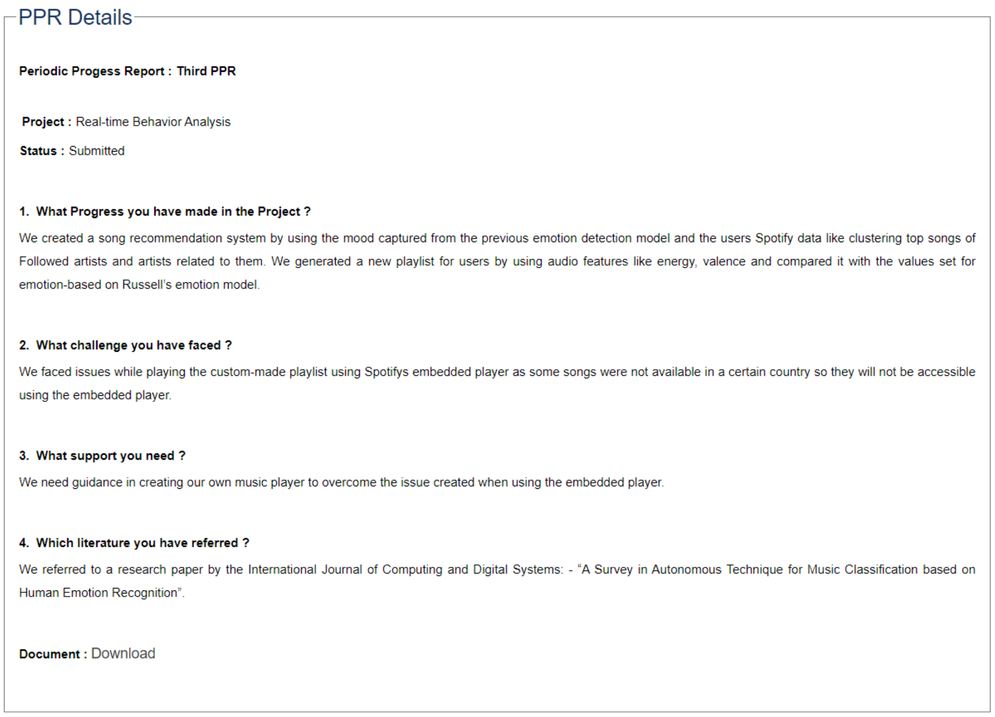
We need further guidance in improving the performance of our application and making it accessible to more users.

4. **Which literature you have referred ?**

We referred to Flask documentation and deployment with git guide provided by devcenter.heroku.com.

Document : Download

PPR 3



PPR Details

Periodic Progess Report : Third PPR

Project : Real-time Behavior Analysis

Status : Submitted

1. **What Progress you have made in the Project ?**

We created a song recommendation system by using the mood captured from the previous emotion detection model and the users Spotify data like clustering top songs of Followed artists and artists related to them. We generated a new playlist for users by using audio features like energy, valence and compared it with the values set for emotion-based on Russell's emotion model.

2. **What challenge you have faced ?**

We faced issues while playing the custom-made playlist using Spotifys embedded player as some songs were not available in a certain country so they will not be accessible using the embedded player.

3. **What support you need ?**

We need guidance in creating our own music player to overcome the issue created when using the embedded player.

4. **Which literature you have referred ?**

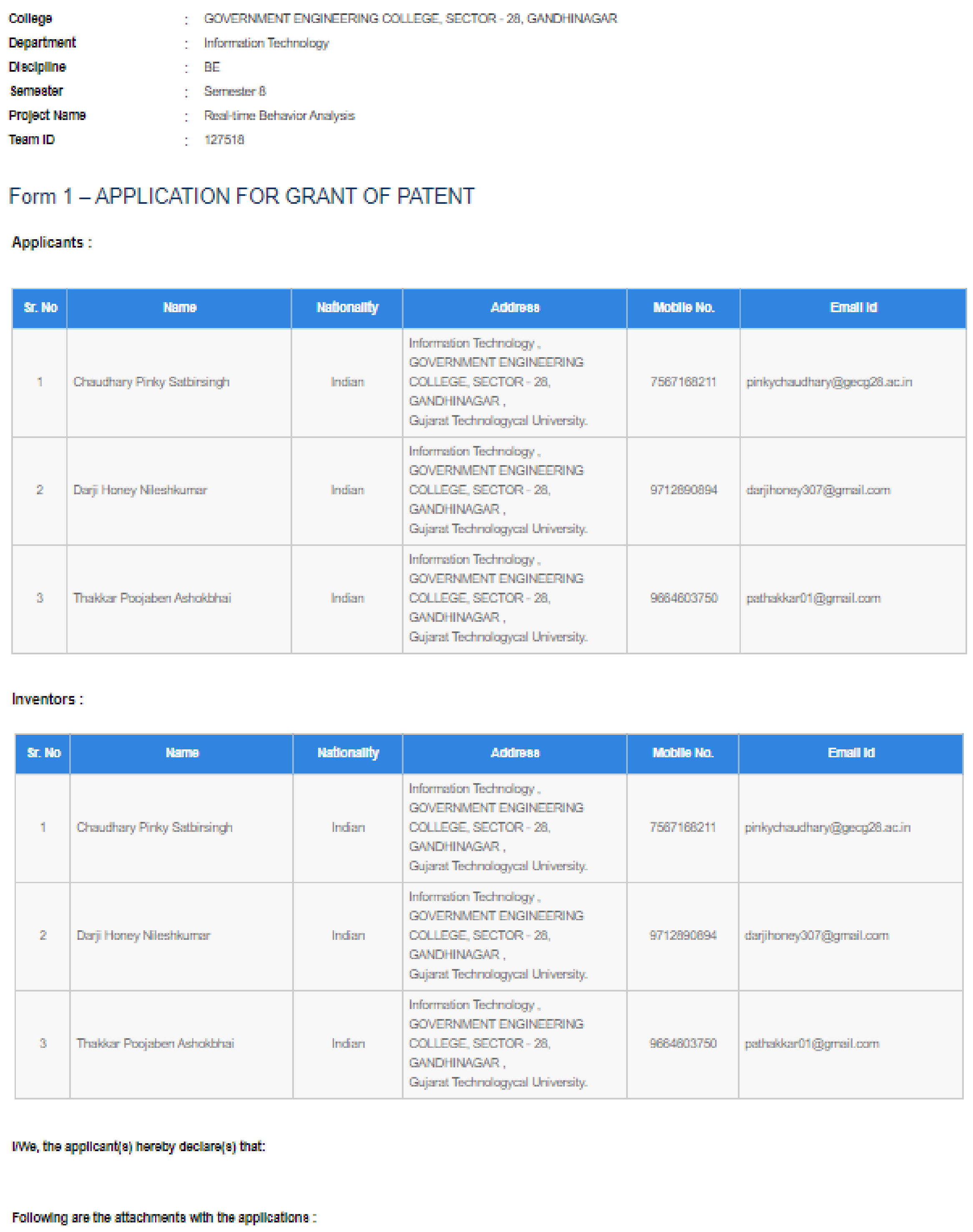
We referred to a research paper by the International Journal of Computing and Digital Systems: - "A Survey in Autonomous Technique for Music Classification based on Human Emotion Recognition".

Document : Download

PPR4

5.2 PDE

Form-1



College	:	GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR
Department	:	Information Technology
Discipline	:	BE
Semester	:	Semester 8
Project Name	:	Real-time Behavior Analysis
Team ID	:	127518

Form 1 – APPLICATION FOR GRANT OF PATENT

Applicants :

Sr. No	Name	Nationality	Address	Mobile No.	Email Id
1	Chaudhary Pinky Satbirsingh	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	7567168211	pinkychaudhary@gecg28.ac.in
2	Darji Honey Nileshkumar	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	9712890894	darjihoney307@gmail.com
3	Thakkar Poojaben Ashokbhai	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	9664803750	pathakkar01@gmail.com

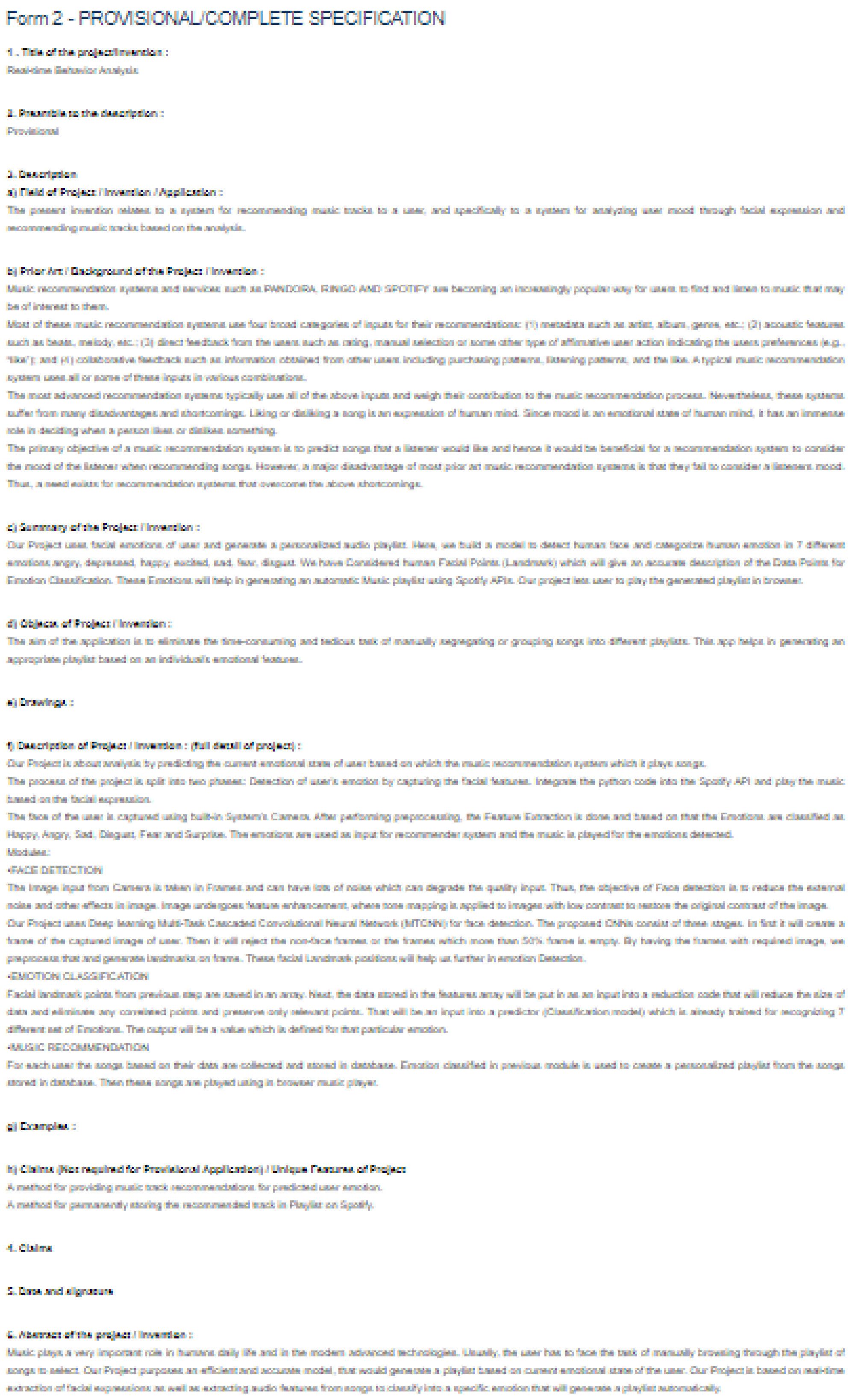
Inventors :

Sr. No	Name	Nationality	Address	Mobile No.	Email Id
1	Chaudhary Pinky Satbirsingh	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	7567168211	pinkychaudhary@gecg28.ac.in
2	Darji Honey Nileshkumar	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	9712890894	darjihoney307@gmail.com
3	Thakkar Poojaben Ashokbhai	Indian	Information Technology , GOVERNMENT ENGINEERING COLLEGE, SECTOR - 28, GANDHINAGAR , Gujarat Technologycal University.	9664803750	pathakkar01@gmail.com

I/We, the applicant(s) hereby declare(s) that:

Following are the attachments with the applications :

FORM-2



Form 2 - PROVISIONAL/COMPLETE SPECIFICATION

1 . Title of the project/invention :
Real-time Behavior Analysis

2. Preamble to the description :
Provisional

3. Description
a) Field of Project / Invention / Application :
The present invention relates to a system for recommending music tracks to a user, and specifically to a system for analyzing user mood through facial expression and recommending music tracks based on the analysis.

b) Prior Art / Background of the Project / Invention :
Music recommendation systems and services such as PANDORA, RINGO AND SPOTIFY are becoming an increasingly popular way for users to find and listen to music that may be of interest to them.
Most of these music recommendation systems use four broad categories of inputs for their recommendations: (1) metadata such as artist, album, genre, etc.; (2) acoustic features such as beats, melody, etc.; (3) direct feedback from the users such as rating, manual selection or some other type of affirmative user action indicating the users preferences (e.g., "like"); and (4) collaborative feedback such as information obtained from other users including purchasing patterns, listening patterns, and the like. A typical music recommendation system uses all or some of these inputs in various combinations.
The most advanced recommendation systems typically use all of the above inputs and weigh their contribution to the music recommendation process. Nevertheless, these systems suffer from many disadvantages and shortcomings. Liking or disliking a song is an expression of human mind. Since mood is an emotional state of human mind, it has an immense role in deciding when a person likes or dislikes something.
The primary objective of a music recommendation system is to predict songs that a listener would like and hence it would be beneficial for a recommendation system to consider the mood of the listener when recommending songs. However, a major disadvantage of most prior art music recommendation systems is that they fail to consider a listeners mood. Thus, a need exists for recommendation systems that overcome the above shortcomings.

c) Summary of the Project / Invention :
Our Project uses facial emotions of user and generate a personalized audio playlist. Here, we build a model to detect human face and categorize human emotion in 7 different emotions angry, depressed, happy, excited, sad, fear, disgust. We have Considered human Facial Points (Landmark) which will give an accurate description of the Data Points for Emotion Classification. These Emotions will help in generating an automatic Music playlist using Spotify APIs. Our project lets user to play the generated playlist in browser.

d) Objects of Project / Invention :
The aim of the application is to eliminate the time-consuming and tedious task of manually segregating or grouping songs into different playlists. This app helps in generating an appropriate playlist based on an individual's emotional features.

e) Drawings :

f) Description of Project / Invention : (full detail of project) :
Our Project is about analysis by predicting the current emotional state of user based on which the music recommendation system which it plays songs.
The process of the project is split into two phases: Detection of user's emotion by capturing the facial features. Integrate the python code into the Spotify API and play the music based on the facial expression.
The face of the user is captured using built-in System's Camera. After performing preprocessing, the Feature Extraction is done and based on that the Emotions are classified as Happy, Angry, Sad, Disgust, Fear and Surprise. The emotions are used as input for recommender system and the music is played for the emotions detected.
Modules:
•FACE DETECTION
The image input from Camera is taken in Frames and can have lots of noise which can degrade the quality input. Thus, the objective of Face detection is to reduce the external noise and other effects in image. Image undergoes feature enhancement, where tone mapping is applied to images with low contrast to restore the original contrast of the image.
Our Project uses Deep learning Multi-Task Cascaded Convolutional Neural Network (MTCNN) for face detection. The proposed CNNs consist of three stages. In first it will create a frame of the captured image of user. Then it will reject the non-face frames or the frames which more than 50% frame is empty. By having the frames with required image, we preprocess that and generate landmarks on frame. These facial Landmark positions will help us further in emotion Detection.
•EMOTION CLASSIFICATION
Facial landmark points from previous step are saved in an array. Next, the data stored in the features array will be put in as an input into a reduction code that will reduce the size of data and eliminate any correlated points and preserve only relevant points. That will be an input into a predictor (Classification model) which is already trained for recognizing 7 different set of Emotions. The output will be a value which is defined for that particular emotion.
•MUSIC RECOMMENDATION
For each user the songs based on their data are collected and stored in database. Emotion classified in previous module is used to create a personalized playlist from the songs stored in database. Then these songs are played using in browser music player.

g) Examples :

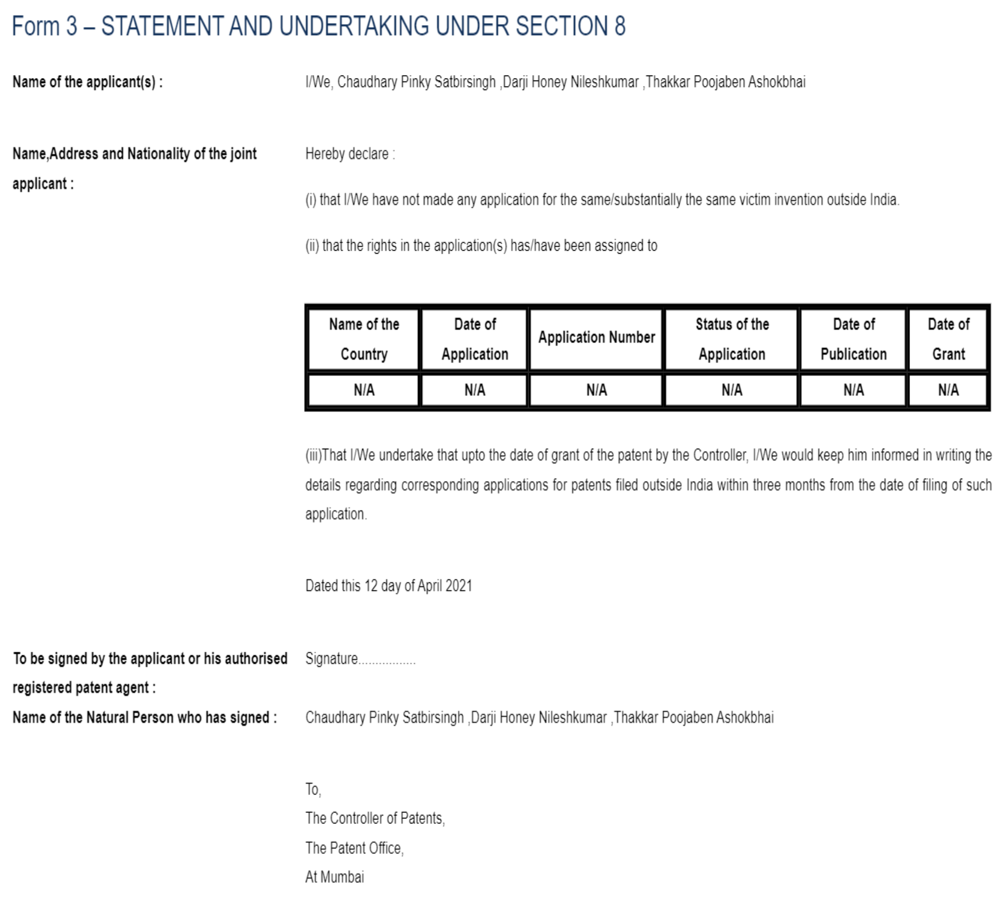
h) Claims (Not required for Provisional Application) / Unique Features of Project
A method for providing music track recommendations for predicted user emotion.
A method for permanently storing the recommended track in Playlist on Spotify.

4. Claims

5. Date and signature

6. Abstract of the project / Invention :
Music plays a very important role in humans daily life and in the modern advanced technologies. Usually, the user has to face the task of manually browsing through the playlist of songs to select. Our Project purposes an efficient and accurate model, that would generate a playlist based on current emotional state of the user. Our Project is based on real-time extraction of facial expressions as well as extracting audio features from songs to classify into a specific emotion that will generate a playlist automatically.

FORM-3



Form 3 – STATEMENT AND UNDERTAKING UNDER SECTION 8

Name of the applicant(s) : I/We, Chaudhary Pinky Satbirsingh ,Darji Honey Nileshkumar ,Thakkar Poojaben Ashokbhai

Name,Address and Nationality of the joint applicant : Hereby declare :

(i) that I/We have not made any application for the same/substantially the same victim invention outside India.

(ii) that the rights in the application(s) has/have been assigned to

Name of the Country	Date of Application	Application Number	Status of the Application	Date of Publication	Date of Grant
N/A	N/A	N/A	N/A	N/A	N/A

(iii)That I/We undertake that upto the date of grant of the patent by the Controller, I/We would keep him informed in writing the details regarding corresponding applications for patents filed outside India within three months from the date of filing of such application.

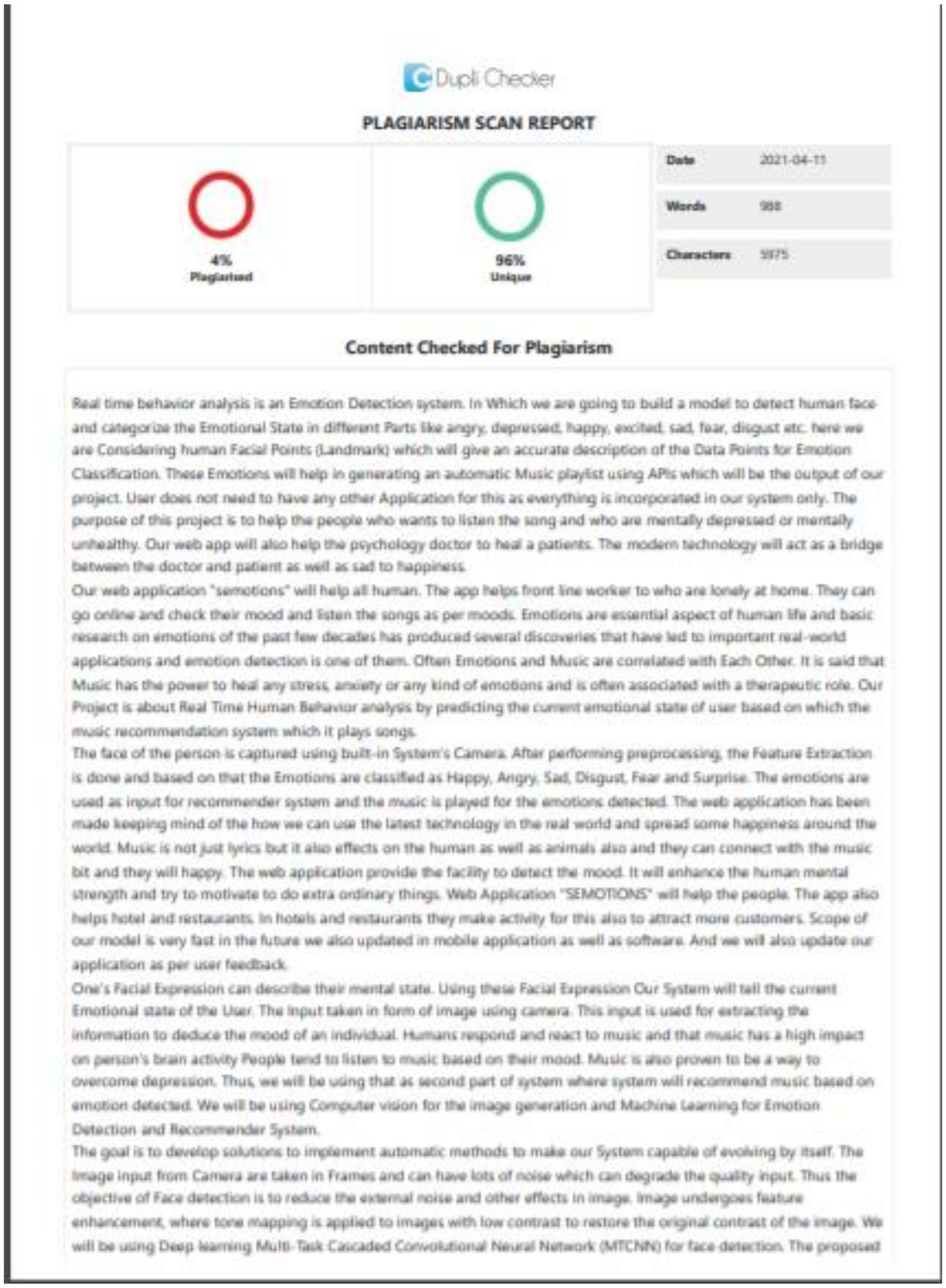
Dated this 12 day of April 2021

To be signed by the applicant or his authorised registered patent agent : Signature.................

Name of the Natural Person who has signed : Chaudhary Pinky Satbirsingh ,Darji Honey Nileshkumar ,Thakkar Poojaben Ashokbhai

To,
The Controller of Patents,
The Patent Office,
At Mumbai

5.3 Plagiarism Report Result



Dupli Checker

PLAGIARISM SCAN REPORT

4%
Plagiarised

96%
Unique

Date	2021-04-11
Words	988
Characters	5975

Content Checked For Plagiarism

Real time behavior analysis is an Emotion Detection system. In Which we are going to build a model to detect human face and categorize the Emotional State in different Parts like angry, depressed, happy, excited, sad, fear, disgust etc. here we are Considering human Facial Points (Landmark) which will give an accurate description of the Data Points for Emotion Classification. These Emotions will help in generating an automatic Music playlist using APIs which will be the output of our project. User does not need to have any other Application for this as everything is incorporated in our system only. The purpose of this project is to help the people who wants to listen the song and who are mentally depressed or mentally unhealthy. Our web app will also help the psychology doctor to heal a patients. The modern technology will act as a bridge between the doctor and patient as well as sad to happiness.

Our web application "semotions" will help all human. The app helps front line worker to who are lonely at home. They can go online and check their mood and listen the songs as per moods. Emotions are essential aspect of human life and basic research on emotions of the past few decades has produced several discoveries that have led to important real-world applications and emotion detection is one of them. Often Emotions and Music are correlated with Each Other. It is said that Music has the power to heal any stress, anxiety or any kind of emotions and is often associated with a therapeutic role. Our Project is about Real Time Human Behavior analysis by predicting the current emotional state of user based on which the music recommendation system which it plays songs.

The face of the person is captured using built-in System's Camera. After performing preprocessing, the Feature Extraction is done and based on that the Emotions are classified as Happy, Angry, Sad, Disgust, Fear and Surprise. The emotions are used as input for recommender system and the music is played for the emotions detected. The web application has been made keeping mind of the how we can use the latest technology in the real world and spread some happiness around the world. Music is not just lyrics but it also effects on the human as well as animals also and they can connect with the music bit and they will happy. The web application provide the facility to detect the mood. It will enhance the human mental strength and try to motivate to do extra ordinary things. Web Application "SEMOTIONS" will help the people. The app also helps hotel and restaurants. In hotels and restaurants they make activity for this also to attract more customers. Scope of our model is very fast in the future we also updated in mobile application as well as software. And we will also update our application as per user feedback.

One's Facial Expression can describe their mental state. Using these Facial Expression Our System will tell the current Emotional state of the User. The input taken in form of image using camera. This input is used for extracting the information to deduce the mood of an individual. Humans respond and react to music and that music has a high impact on person's brain activity People tend to listen to music based on their mood. Music is also proven to be a way to overcome depression. Thus, we will be using that as second part of system where system will recommend music based on emotion detected. We will be using Computer vision for the image generation and Machine Learning for Emotion Detection and Recommender System.

The goal is to develop solutions to implement automatic methods to make our System capable of evolving by itself. The Image input from Camera are taken in Frames and can have lots of noise which can degrade the quality input. Thus the objective of Face detection is to reduce the external noise and other effects in image. Image undergoes feature enhancement, where tone mapping is applied to images with low contrast to restore the original contrast of the image. We will be using Deep learning Multi-Task Cascaded Convolutional Neural Network (MTCNN) for face detection. The proposed

5.4 References

Patents and Research papers and Reference Books

- An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach by Ashu Abdul 1 ID, Jenhui Chen 2,3, 4, † ID, Hua-Yuan Liao 2 and Shun-Hao Chang 2

- Kołakowska A., Landowska A., Szwoch M., Szwoch W., Wróbel M.R. (2014) Emotion Recognition and Its Applications.

- Arto Lehtiniemi and Jukka Holm, "Using Animated Mood Pictures in Music Recommendation", 2012 16th International Conference on Information Visualization.

- F. Abdat, C. Maaoui and A. Pruski, "Human-computer interaction using emotion recognition from facial expression", 2011 UKSim 5th European Symposium on Computer

- Park SH., Ihm SY., Jang WI., Nasridinov A., Park YH. (2015) A Music Recommendation Method with Emotion Recognition Using Ranked Attributes

- Bruce Ferwerda and Markus Schedl "Enhancing Music Recommender Systems with Personality Information and Emotional States": A Proposal: 2014

- Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras Book by Dipanjan Sarkar, Raghav Bali, and Tamoghna Ghosh

- Digital Image Processing book by rafael c gonzalez.

- Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python Book by Md. Rezaul Karim, MohitSewak, and Pradeep Pujari

- https://worldwide.espacenet.com/patent/search/family/043427165/publication/US9818024B2?q=pn%3DUS9818024B2

- https://patents.google.com/patent/US8094891
- https://patents.google.com/patent/US20060143647?oq=Facial+Emotion+based+Music+playlist
- https://worldwide.espacenet.com/patent/search/family/054210027/publication/US2015286858A1?q=pn%3DUS2015286858A1
- https://patents.google.com/patent/US20030133599A1/en?q=Emotions+detection+facial+expressions&oq=Emotions+detection+using+facial+expressions