CHAPTER 1

INTRODUCTION

1.1 PROJECT TITLE:

EMOTICBEAT
(Emotion Based Music Player)

1.2 PROJECT DESCRIPTION:

The human face is an important part of an individual's body and it especially plays an important role in knowing an individual's mood. Extracting the required input from the human face can now be done directly using a camera. This input can then be used in many ways. One of the applications of this input can be for extracting the information to deduce the mood of an individual. This data can then be used to get a list of songs that comply with the mood derived from the input provided earlier. This eliminates the time consuming and tedious task of manually Segregating or grouping songs into different lists and helps in generating an appropriate playlist based on an individual's emotional features. Various algorithms have been developed and proposed for automating the playlist generation process.

Facial Expression Based Music Player aims at scanning and interpreting the data and accordingly creating a playlist based the parameters provided. The scanning and interpreting include audio feature extraction and classification to get a list of songs belonging to a similar genre or to get a list of similar sounding songs. Human emotions are meant for mutual understanding and sharing feelings and intentions. The emotions are manifested in verbal and facial expressions. Face detection has been around for ages. Taking a step forward, human emotion displayed by face and felt by brain, captured in either video, electric signal (EEG) or image form can be approximated. Human emotion detection is the need of the hour so that modern artificial intelligent systems can emulate and gauge reactions from face. This can be helpful to make informed decisions be it regarding identification of intent, promotion of offers or security related threats. Recognizing emotions from images or video is a trivial task for human eye, but proves to be very challenging for machines and requires many image processing techniques for feature extraction. Several machine learning algorithms are suitable for this job. Any detection or recognition by machine learning requires training algorithm and then testing them on a suitable dataset.

People of today’s world are very indecisive even when it comes to the simplest of things like music. This project’s purpose is to design a classifier that could be used to eliminate this confusion exhibited by people. This project is about building a music recommendation system for users. Such a system can not only be used to brighten up one's mood on a rainy weekend; especially in hospitals, other medical clinics, or public locations such as restaurants, this classifier could be used to spread positive mood among people.

The project has to be implemented in 2 phases-facial emotion recognition and then suggestion of music based on the detected emotion in the previous phase. The

emotional reaction to music is different for every person, so analyzing it will not likely yield perfect results. The method used then is to decide upon certain base songs that very closely embody a certain mood, and to match songs to these specific categories. We can also study the efficacy of music stimuli to modify (reduce rather than induce) naturally occurring negative mood states (like sad, depressed, etc.) and to gradually bring about a change in this negative mood state turning it into a positive one. It could also simply be used just to prevent certain songs from having a negative impact on a person's mood.

CHAPTER 2

LITERATURE SURVEY

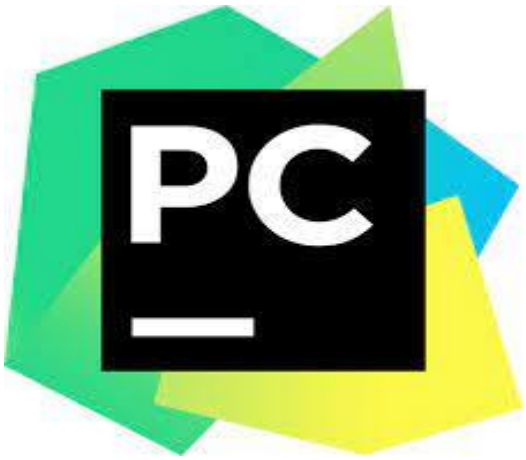
2.1 EXSTING SYSTEM AND PROPOSED SYSTEM:

Music plays a very important role in enhancing an individual ‘s life as it is an important medium of entertainment for music lovers and listeners. In today ‘s world, with ever increasing advancements in the field of multimedia and technology, various music players have been developed with features like fast forward, reverse, variable playback speed (seek & time compression), local playback, streaming playback with multicast streams and including volume modulation, genre classification etc. Although these features satisfy the user ‘s basic requirements, yet the user has to face the task of manually browsing through the playlist of songs and select songs based on his current mood and behavior.

This project work helps the user to automatically play songs based on the emotions of the user. It recognizes the facial emotions of the user and plays the songs according to their emotion. The process of detecting moods is quite challenging. We are using the FER & Haar Cascade Algorithm to recognize the face expression of user: here we are using haar cascade to recognize the face & FER for Expression Recognition. By using FER Model We can Classify Expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

2.2 TOOLS AND TECHNOLOGY USED:

Pycharm IDE:



PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

Flask:

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where preexisting third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Opencv

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

Keras & Tensorflow:

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

Keras is:

Simple -- but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.

Flexible -- Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.

Powerful -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

Tensorflow is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

Efficiently executing low-level tensor operations on CPU, GPU, or TPU.

Computing the gradient of arbitrary differentiable expressions.

Scaling computation to many devices, such as clusters of hundreds of GPUs.

Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

FER (Face Expression Recognition)

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

train.csv contains two columns, "emotion" and "pixels". The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column and your task is to predict the emotion column. The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples. This dataset was prepared by Pierre-Luc Carrier and Aaron Courville, as part of an ongoing research project. They have graciously provided the workshop organizers with a preliminary version of their dataset to use for this contest.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements:

• Intel i3 or higher processor

• Webcam

• Monitor

• 500GB HDD

Software Requirements:

• Windows 7 Or Higher Windows Os

• Pycharm Ide

• Python 3

• Flask

• Tensorflow

• Keras

Programming Languages:

- Python
- Html, Css, Javascript, Bootstrap

CHAPTER 3

SYSTEM DESIGN

3.1 USE CASE DIAGRAMS

A use case diagram as its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different use cases and will often be accompanied by other types of diagrams as well.

While a use case itself might drill into a lot of detail about every possibility, the use case diagram can help provide a higher-level view of the system. It has been said before that "Use Case diagrams are the blueprints for the system" They provide the specified and graphical representation of what the system must actually do.

A use case diagram contains four components:

The boundary, which defines the system of interest in relation to the world around it.The actors, usually individuals involved with the systems defined according to their roles.

The use cases in which the specific roles are played by the actors within and around the system.

The relationships between and among the actors and the use cases.

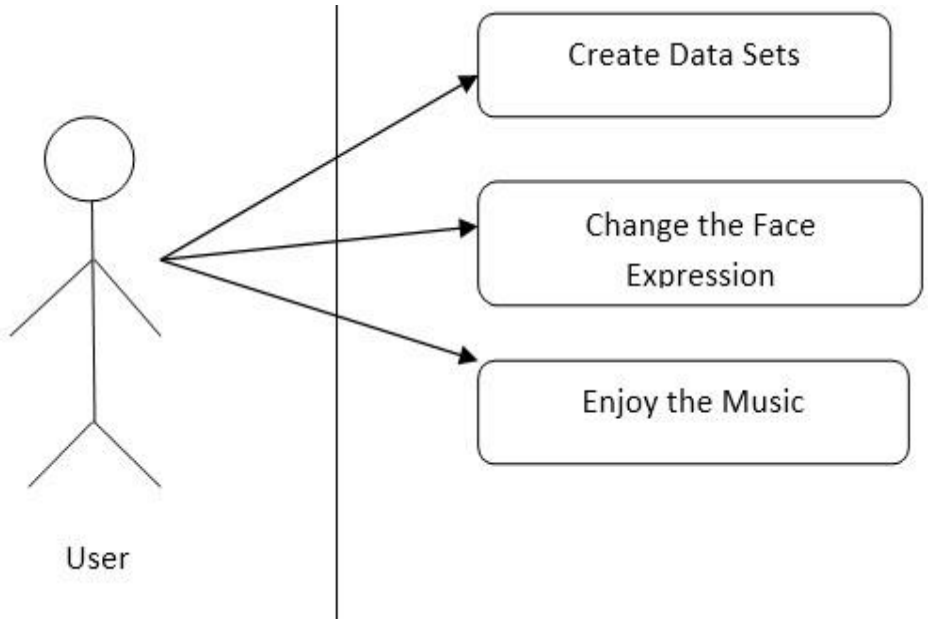


Fig: Use Case Diagram of EmoticBeat

3.2 BLOCK DIAGRAM

It is represented as a diagram of a system in which the principal parts or functions are represented in the form of blocks connected by lines that shows the relationship of the blocks.

They are heavily used in hardware design, electronic design, software design, and process flow diagrams.

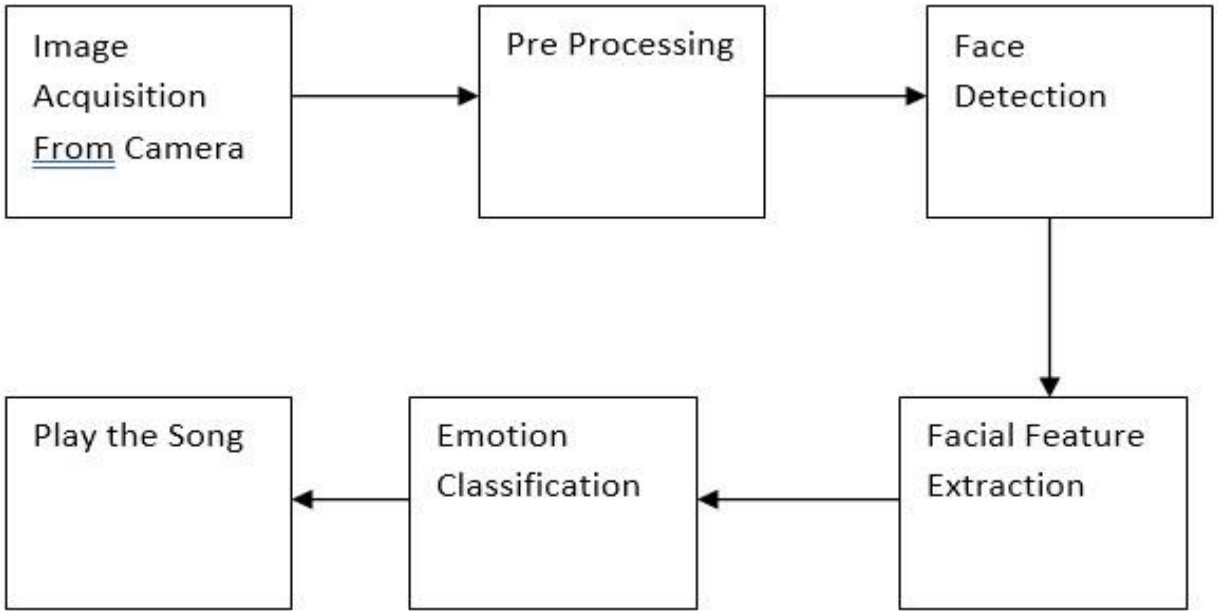


Fig: Block Diagram of EmoticBeat

3.3 DATA FLOW DIAGRAMS:

Data flow indicates the passage of data in the system, from where the data flows. It is indicated by an arrow which indicates the direction of flow.The arrow is labeled by the name of the data flow

Flow of data in the system could be any one of the following:

From a data store to a process

From source to process

From a process to a sink

Zero Level

First Level

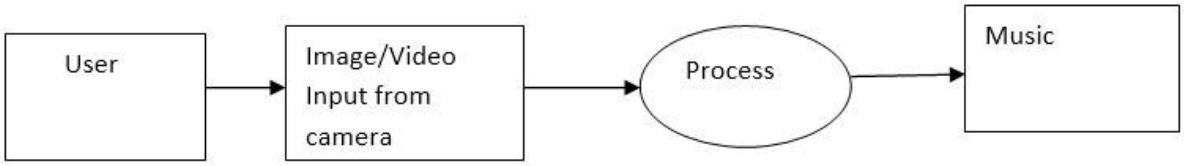


Fig: Level 0 and 1 data flow diagram of EmoticBeat

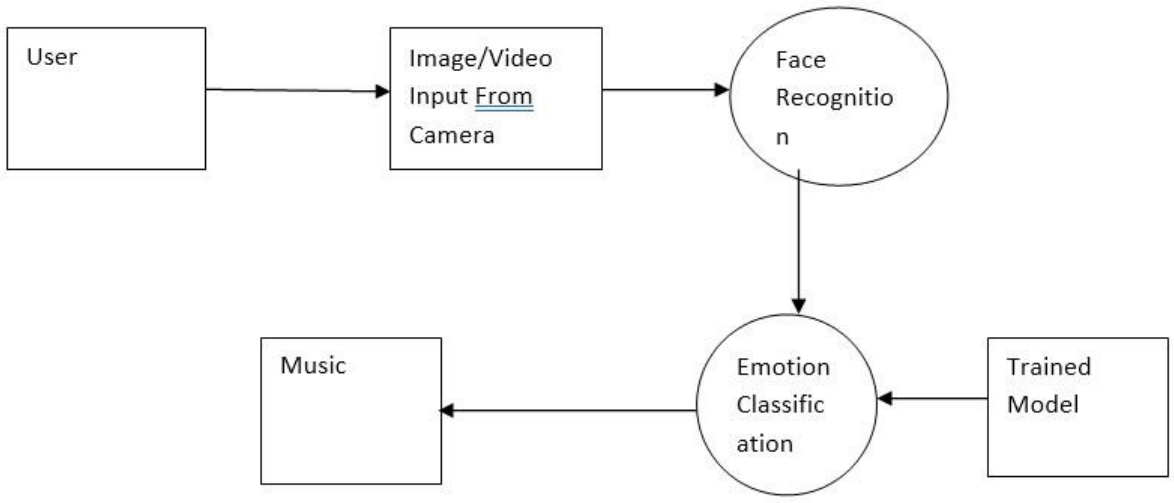


Fig: Level 2 data flow diagram of EmoticBeat

3.4 ER DIAGRAM:

ER (Entity-Relationship) diagram is a popular high-level conceptual data model and its variations are frequently used for the conceptual design of the database design tools to employ its concepts. It is used in the design of conceptual schemas for database applications. Entities and Attributes:

The basic object that the ER model represents is an entity, which is a "thing" in the real world with an independent existence. An entity may be an object with a physical existence a particular person car or an employee or it may be an object with conceptual existence particular properties that describes it.

Ex: A client entity may be described by the client name, client project name, etc. A particular entity will have a value for each of its attributes.

These are the following types of attributes:

- Simple
- Composite
- Single valued
- Stored
- Derived
- Multi-valued

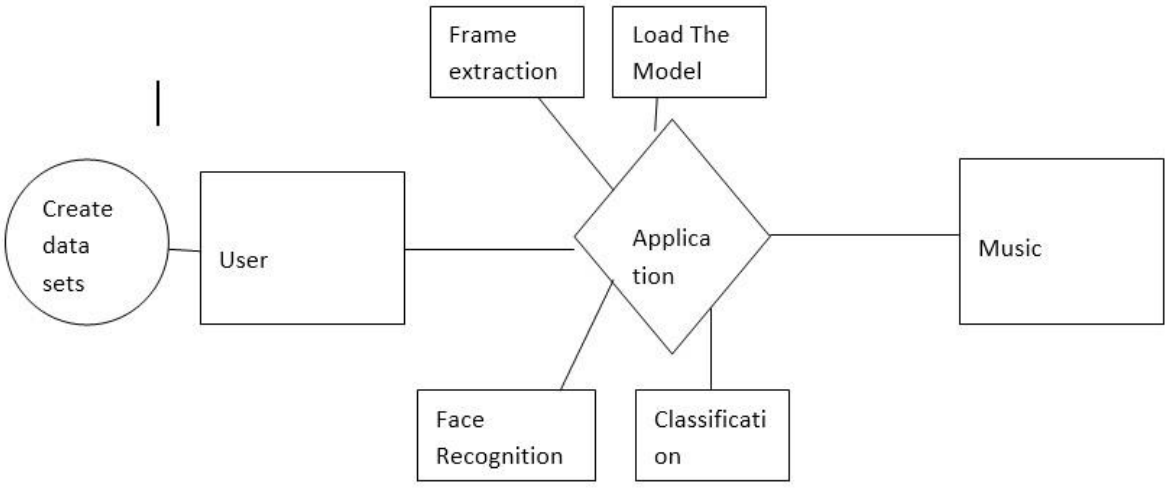


Fig: ER Diagram of EmoticBeat

CHAPTER 4

CODING

4.0 Coding:

Front End:

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>EMOTICBEAT</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
     <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
</head>

<body style="background-image:url('/static/backg.jpg');background-size:1600px
1200px;background-repeat:no-repeat">
<div class="text-center">
    <h1 class="display-1">EMOTICBEAT MUSIC PLAYER</h1>

</div>

<div class="text-center display-3">
    <button class="btn btn-warning btn-lg"> <a href="/start"><h1 class="text-
dark">Start</h1></a> </button>
</div>
```

```
</body>
</html>
```

happy.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>EMOTICBEAT</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
</head>

<body style="background-image:url('/static/backg.jpg');background-size:1600px
1200px;background-repeat:no-repeat">
<div class="text-center">
      <h1 class="display-1">EMOTICBEAT MUSIC PLAYER</h1>

</div>
<br>

<br>
<div class="text-left">
```

```
   <img src="/static/happy/happy.jpg" height="300" width="330">
</div>
<br>
<table class="table font-weight-bold" style="text-size:30px;">
<thead>
<tr>
<th>SONG NAME</th>
<th>TRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>All_Is_Well_3 IDIOTS</td>

<td><audio controls autoplay>
<source src="/static/happy/All_Is_Well.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Illahi_YEH_JAWANI_HAI_DEEWANI</td>

<td><audio controls>
<source src="/static/happy/Illahi.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Manwa_Laage_HAPPY_NEW_YEAR</td>

<td><audio controls>
<source src="/static/happy/Manwa_Laage.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Param_Sundari_MIMI</td>
<td><audio controls>
<source src="/static/happy/Param_Sundari.mp3" type="audio/mpeg">
</audio>
```

```
</td>
</tr>
<tr>
<td>Raatan_Lambiyan_SHERSHAAH</td>
<td><audio controls>
<source src="/static/happy/Raatan Lambiyan.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Shab_Tum_Ho_DARSHAN_RAVAL</td>
<td><audio controls>
<source src="/static/happy/Shab_Tum_Ho.mp3" type="audio/mpeg">
</audio>
</td>
</tr>

</tbody>
</table>


</body>
</html>
```

neutral.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>EMOTICBEAT</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
```

```
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
</head>

<body style="background-image:url('/static/backg.jpg');background-size:1600px
1200px;background-repeat:no-repeat">
<div class="text-center">
    <h1 class="display-1">EMOTICBEAT MUSIC PLAYER</h1>

</div>
<br>

<br>
<div class="text-left">
    <img src="/static/neutral/neutral.jpg" height="300" width="330">
</div>
<br>
<table class="table font-weight-bold" style="text-size:30px;">
<thead>
<tr>
<th>SONG NAME</th>
<th>TRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Krishna flute</td>
<td><audio controls autoplay>
<source src="/static/neutral/Krishna flute.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Main Agar Kahoon Remix</td>
```

```
<td><audio controls>
<source src="/static/neutral/Main_Agar_Kahoon_Bol_Do_Na_Zara.mp3"
type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Tera Hone Laga Hoon</td>
<td><audio controls>
<source src="/static/neutral/Tera_Hone_Laga_Hoon.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Tu Jaana Na</td>
<td><audio controls>
<source src="/static/neutral/Tu_Jaane_Na.mp3" type="audio/mpeg">
</audio>
</td>
</tr>


</tbody>
</table>


</body>
</html>
```

fear.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>EMOTICBEAT</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
     <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
</head>

<body style="background-image:url('/static/backg.jpg');background-size:1600px
1200px;background-repeat:no-repeat">
<div class="text-center">
    <h1 class="display-1">EMOTICBEAT MUSIC PLAYER</h1>

</div>
<br>

<br>
<div class="text-left">
    <img src="/static/fear/fear.jpg" height="300" width="330">
</div>
<br>
<table class="table font-weight-bold" style="text-size:30px;">
<thead>
<tr>
<th>SONG NAME</th>
<th>TRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lost_Without_You_HALF_GIRLFRIEND</td>
<td><audio controls autoplay>
<source src="/static/fear/Lost_Without_You.mp3" type="audio/mpeg">
</audio>
</td>
```

```
</tr>
<tr>
<td>Darkside_ALAN_WALKER</td>
<td><audio controls>
<source src="/static/fear/Alan_Walker_-_Darkside.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Marethu Hoyithe_AMAR</td>
<td><audio controls>
<source src="/static/fear/Marethuhoyithe.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Mussanje_Veleli_ADDHURI</td>
<td><audio controls>
<source src="/static/fear/Mussanje_veleli.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Phir Bhi Tumko Chaahungi_HALF GIRLFRIEND</td>
<td><audio controls>
<source src="/static/fear/Phir_Bhi_Tumko_Chaahungi.mp3" type="audio/mpeg">
</audio>
</td>
</tr>


</tbody>
</table>


</body>
</html>
```

surprise.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>EMOTICBEAT</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script src='https://kit.fontawesome.com/a076d05399.js'
crossorigin='anonymous'></script>
</head>

<body style="background-image:url('/static/backg.jpg');background-size:1600px
1200px;background-repeat:no-repeat">
<div class="text-center">
    <h1 class="display-1">EMOTICBEAT MUSIC PLAYER</h1>

</div>
<br>

<br>
<div class="text-left">
    <img src="/static/surprise/surprise.jpg" height="300" width="330">
</div>
<br>
<table class="table font-weight-bold" style="text-size:30px;">
<thead>
<tr>
```

```
<th>SONG NAME</th>
<th>TRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dhadak_Title_Track</td>
<td><audio controls autoplay>
<source src="/static/surprise/Dhadak_Title_Track.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Bekhudi_Tera Surroor</td>
<td><audio controls>
<source src="/static/surprise/Bekhudi.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Heartless_BADSHAH</td>
<td><audio controls>
<source src="/static/surprise/Heartless.mp3" type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Meri Aashiqui Ab Tum Ho_FEMALE</td>
<td><audio controls>
<source src="/static/surprise/Meri_Aashiqui_Ab_Tum_Hi_Ho_Female.mp3"
type="audio/mpeg">
</audio>
</td>
</tr>
<tr>
<td>Nazm Nazm</td>
<td><audio controls>
<source src="/static/surprise/Nazm_Nazm.mp3" type="audio/mpeg">
</audio>
</td>
```

```
</tr>
<tr>
<td>Blank Space_TAYLOR SWIFT</td>
<td><audio controls>
<source src="/static/surprise/Taylor_Swift_-_Blank_Space.mp3"
type="audio/mpeg">
</audio>
</td>
</tr>
</tbody>
</table>


</body>
</html>
```

Back End:

main.py

```
from flask import *
import cv2
import numpy as np
from keras.models import model_from_json
from keras.preprocessing import image

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/start')
def start():
    # load model
    model = model_from_json(open("fer.json", "r").read())
    # load weights
    model.load_weights('fer.h5')
```

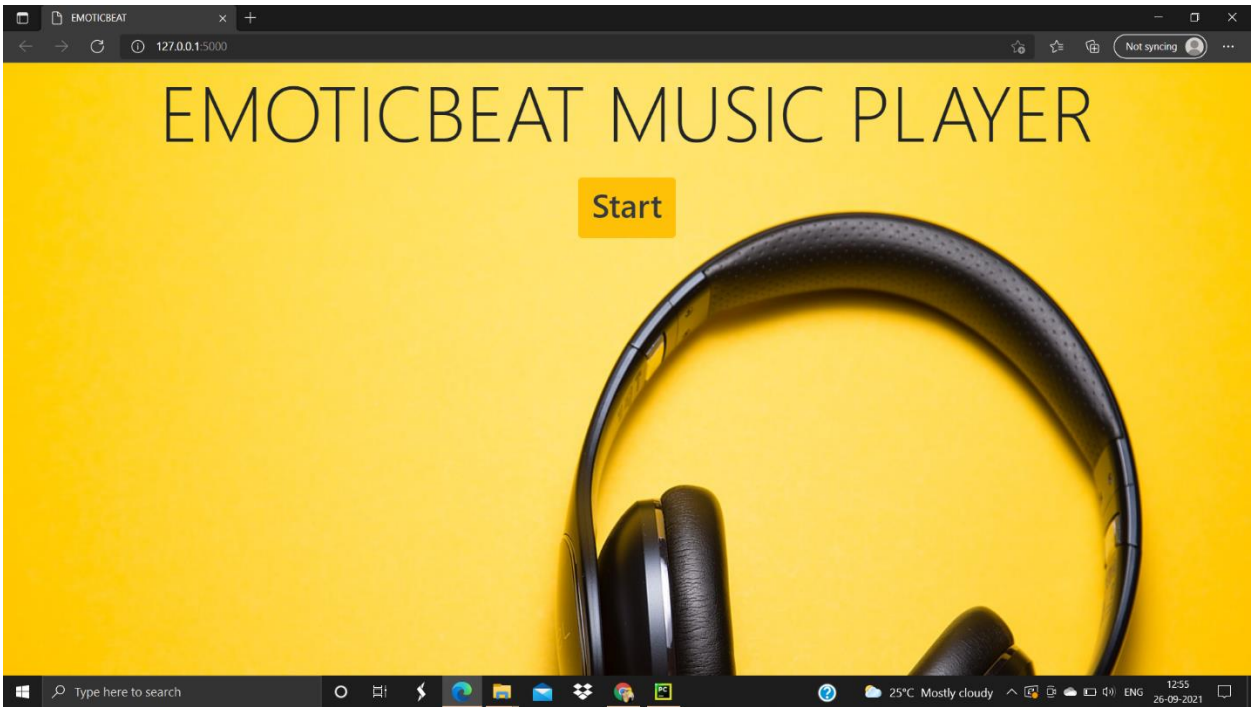
```
   face_haar_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
   i = 0 #happy
   j = 0 #neutral
   k=0 #fear
   l=0 #surprise
   cap = cv2.VideoCapture(0)

   while True:
      ret, test_img = cap.read()
      if not ret:
         continue
      gray_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)

      faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)
      resized_img = test_img
      for (x, y, w, h) in faces_detected:
         cv2.rectangle(test_img, (x, y), (x + w, y + h), (255, 0, 0), thickness=7)
               roi_gray = gray_img[y:y + w, x:x + h]
         roi_gray = cv2.resize(roi_gray, (48, 48))
         img_pixels = image.img_to_array(roi_gray)
         img_pixels = np.expand_dims(img_pixels, axis=0)
         #img_pixels
         img_pixels /= 255
         predictions = model.predict(img_pixels)
         max_index = int(np.argmax(predictions))
         emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
         predicted_emotion = emotions[max_index]

         cv2.putText(test_img, predicted_emotion, (int(x), int(y)),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
         resized_img = cv2.resize(test_img, (1000, 700))
         cv2.imshow('Facial emotion analysis ', resized_img)
         if predicted_emotion == 'happy':
            i = i + 1
            j = k=l = 0
            if i > 20:
                     print(i)
               cv2.imwrite('static/happy/happy.jpg',resized_img)
               return render_template('happy.html')
```

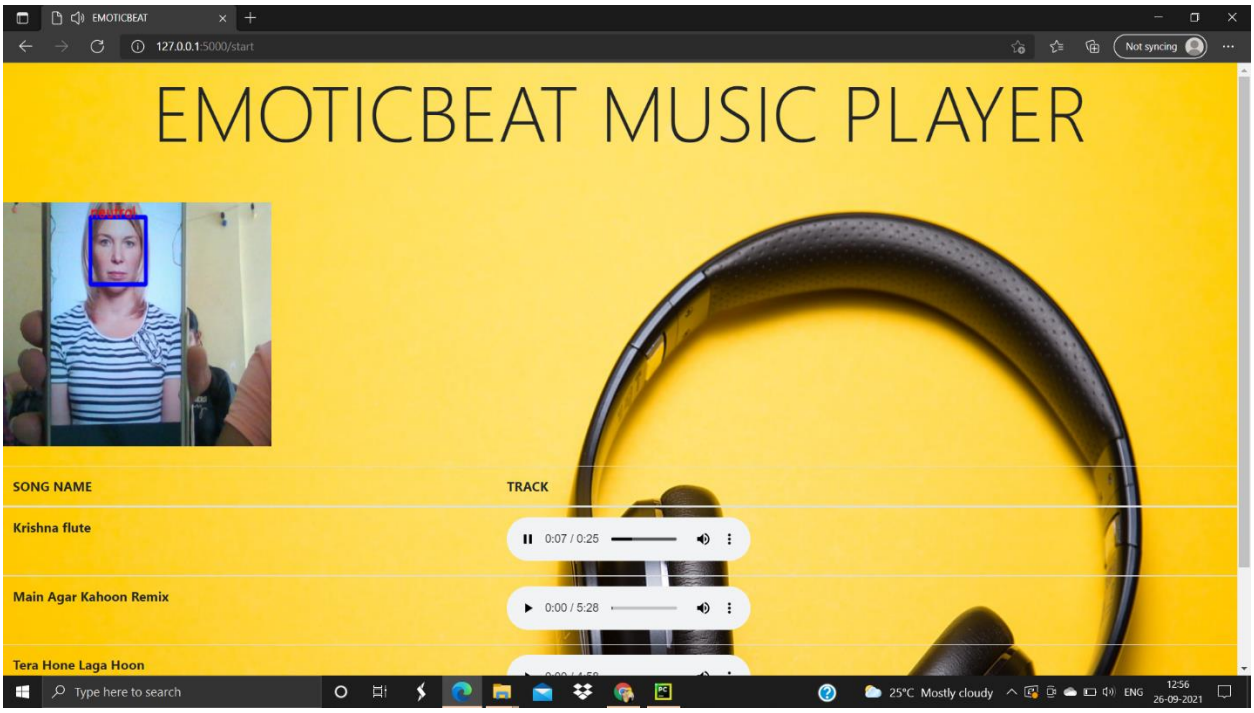
```
        if predicted_emotion == 'neutral':
            print(j)
            j = j + 1
            i=k=l = 0
            if j > 10:
                print(j)
                cv2.imwrite('static/neutral/neutral.jpg', resized_img)
                return render_template('neutral.html')
        if predicted_emotion == 'fear':
            print(k)
            k = k + 1
            j=i=l = 0
            if k > 10:
                print(k)
                cv2.imwrite('static/fear/fear.jpg', resized_img)
                return render_template('fear.html')
        if predicted_emotion == 'surprise':
            print(k)
            l = l + 1
            j=i = 0
                    if l> 10:
                print(l)
                cv2.imwrite('static/surprise/surprise.jpg', resized_img)
                return render_template('surprise.html')
        #cv2.putText(test_img, predicted_emotion, (int(x), int(y)),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
      #resized_img = cv2.resize(test_img, (1000, 700))
        #cv2.imshow('Facial emotion analysis ', resized_img)
      if cv2.waitKey(10) == ord('q'):
        break
   cap.release()
   cv2.destroyAllWindows()
if __name__ == '__main__':
   app.run()
```

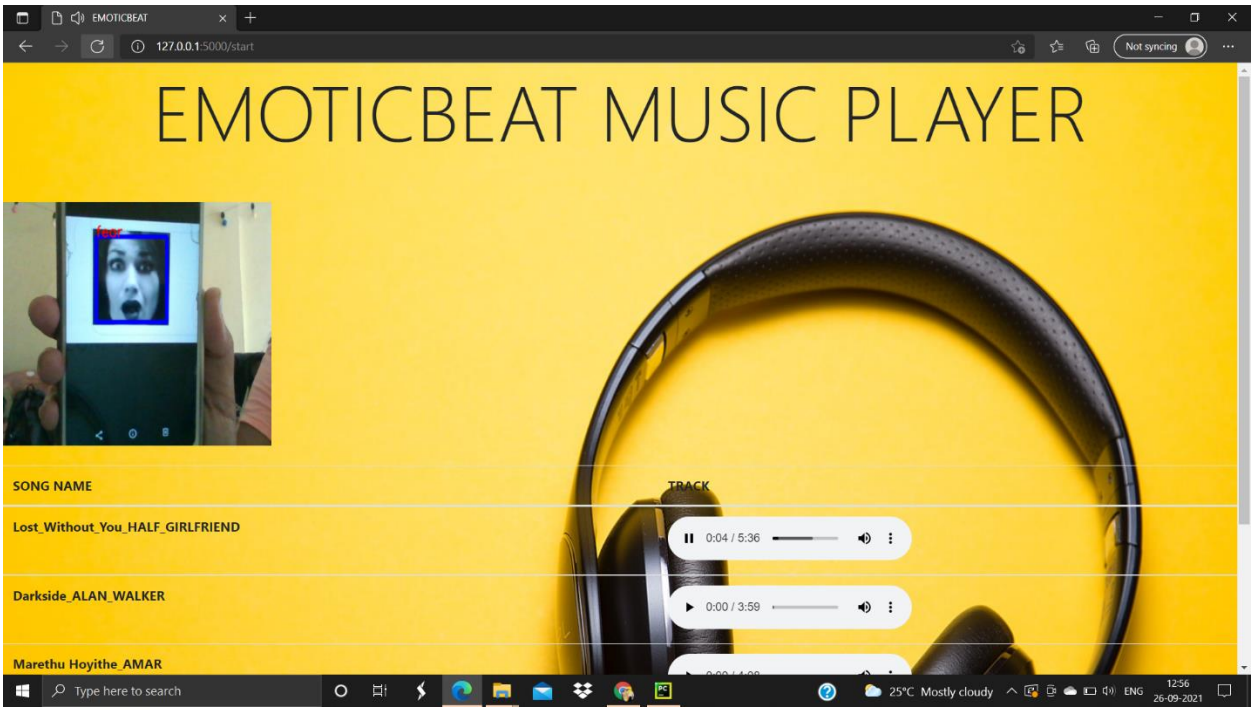
CHAPTER 5

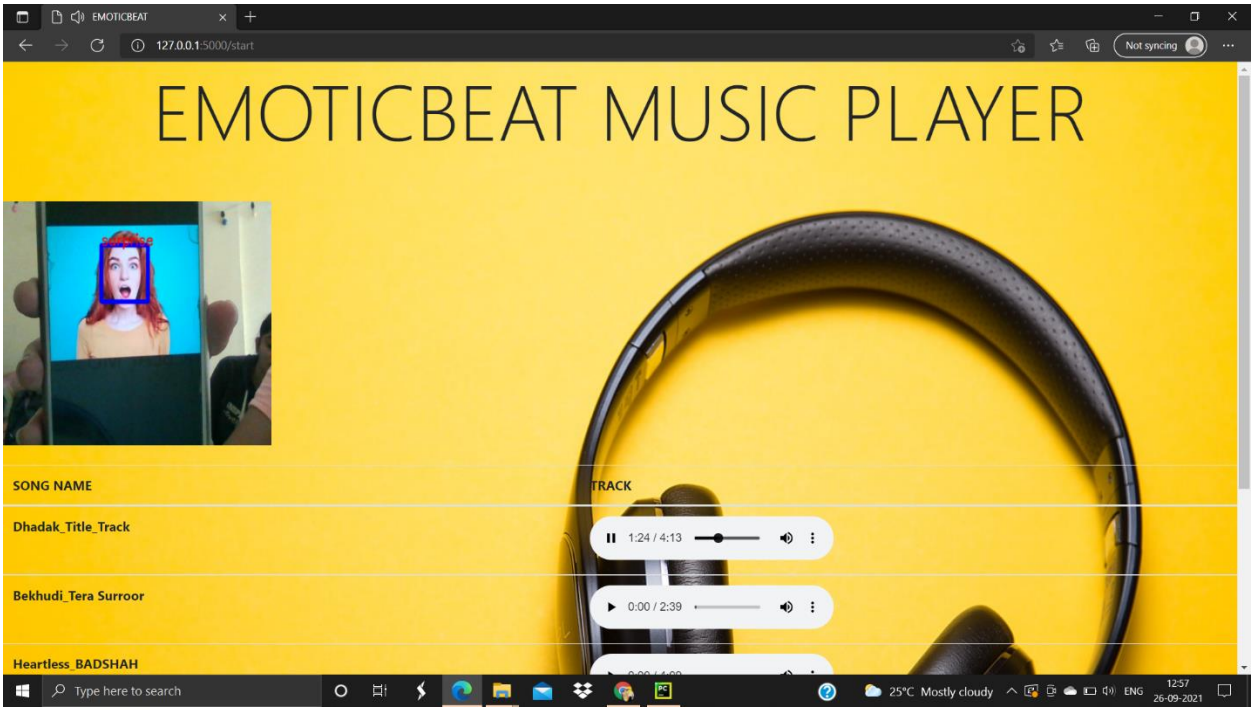
IMPLEMENTATION

5.1 SCREEN SHOTS:






EMOTICBEAT
127.0.0.1:5000/start
Not syncing
EMOTICBEAT MUSIC PLAYER
fear
SONG NAME
TRACK
Lost_Without_You_HALF_GIRLFRIEND
0:04 / 5:36
Darkside_ALAN_WALKER
0:00 / 3:59
Marethu Hoyithe_AMAR
Type here to search
25°C Mostly cloudy
ENG
12:56
26-09-2021


EMOTICBEAT
127.0.0.1:5000/start
Not syncing
EMOTICBEAT MUSIC PLAYER
surprise
SONG NAME
TRACK
Dhadak_Title_Track
1:24 / 4:13
Bekhudi_Tera Surroor
0:00 / 2:39
Heartless_BADSHAH
Type here to search
25°C Mostly cloudy
ENG
12:57
26-09-2021

CHAPTER 6

SOFTWARE TESTING

6.SOFTWARE TESTING:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Software testing can be stated as the process of validating and verifying that a computer program meets the requirements that guided its design and development, application works as expected, product can be implemented with the same characteristics and satisfies the needs of customers.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. When an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment.

Testing Strategies

Software testing methods are traditionally divided into white- box testing and black-box testing.

These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

White-box testing (also known as clear box testing, glass box testing and transparent-box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. This testing can be applied at unit level, integration level and system level.

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. It is also known as Release testing, functional testing. This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. The one who tests is only aware of what the software is supposed to do, not how it does it.

Unit Testing

Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other. Depending on the organization's expectations for software development, unit testing might include static code analysis, data flow analysis metrics analysis, peer code reviews, code coverage analysis and other software verification practices.

Example: After designing the login page, we tested the page code using dummy.User name and password to check if it works correctly.

Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system. Integration testing checks that the components actually work together and are called correctly and transfer the right data at the right time across their interfaces.

Example: Payments page and User Page are interlinked and are tested using dummy inputs.

System Testing

System testing tests a completely integrated system to verify that it meets its requirements. The concern of this testing is to check the behaviour of the whole system as defined by the scope of the project. The main concern of system testing is to verify the system against the specified requirements. While carrying out the test, the one who tests is not concerned with the internals of the system, but checks if the system behaves as per expectations.

All the modules and pages have been tested after the whole system is built.

Acceptance Testing

This testing is done while the product is completely ready to be delivered. Here the tester especially has to literally think like the client and test the software with respect to user needs, requirements and business processes and determine, whether the software can be handed over to the client. At this stage, often a client representative is also a part of the testing team, so that the client has confidence in the system. There are different types of acceptance testing:

- Operational Acceptance Testing
- Compliance Acceptance Testing
- Alpha testing
- Beta Testing

These tests are carried out for both the functional and non-functional

Attributes of the software.

Validation Testing

Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Validation is confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled.

Validation testing is an important testing, which is carried out before the software is handed over to the customer. The aim of validation testing is to ensure that the software product is made according to the requirements of the client and does indeed fulfill the intended purpose.

So that the software product is tested thoroughly without any bias, often the job of validation testing may also be given to third party services. Therefore, it is an important part of software quality assurance procedures and standards.

- For all pages, modules validation controls are inserted and are verified.

Verification Testing

Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Verification is confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled.

Verification testing can be used by the Unit Test component to provide coverage metrics for test cases that execute at various levels, including function/method, module/class and component/system levels. Verification testing provides test planning documents and color-coded graphs to help you achieve the desired coverage, ranging from procedure/function calls to safety-critical modified condition/ decision coverage.

7. CONCLUSION:

The system works using the principle of FER to play song based on recognized mood of the individual. It makes use of Haar Cascade algorithm alongside PCA approach to extract essential facial features. The system eradicates the need of browsing through the Web for desired song and ensures correct song is played for current mood of user.

8. FUTURE ENHANCEMENTS:

The future scope for this system would be to implement it on Mobile platforms like Android or iOS, as there has been increase in the number of Mobile phone application users worldwide. Also in the work, at a time only one emotion is recognized so system design can be extended to a mixed mood recognition system which will further enhance the functionality and accuracy of the overall system. This system is applicable in various fields like Medical Science and Psychology as it will be helpful in Music Therapy treatment for music therapists to treat their patients suffering from depression, mental stress, fatigue and trauma. This can be also useful to identify the mood of a physically disabled person as we can

understand their needs and help them accordingly.

9. BIBLIOGRAPHY:

Website references:

a. W3Schools :-https://www.w3schools.com/
b. http://www.ijreat.org/Papers%202020/Issue47/IJREATV8I5010.pdf
c. https://www.irjet.net/archives/V5/i4/IRJET-V5I4830.pdf
d. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
e. https://www.ijert.org/research/mood-based-music-recommendation-system-IJERTV10IS060253.pdf