**VIBING MUSIC**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**M ASHWINI- 1602-19-737-068**

**V SHWETHA - 1602-19-737-106**

**P ASHWINI - 1602-19-737-311**

**A close-up of a logo

Description automatically generated with medium confidence**

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2021**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

**DECLARATION BY THE CANDIDATE**

              We, **M. ASHWINI, V SHWETHA, P ASHWINI** bearing hall ticket numbers, **1602-19-737-068, 1602-19-737-106, 1602-19-737-311**, hereby declare that the project report entitled **“VIBING MUSIC”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

               This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**M. ASHWINI**

**1602-19-737-068**

**V. SHWETHA**

**1602-19-737-106**

**P. ASHWINI**

**1602-19-737-311**

(Faculty In-Charge)                   (Head,Dept of IT)

MINI PROJECT

           ON

                 VIBING MUSIC

TEAM MEMBERS GUIDES

M. Ashwini(1602-19-737-068) Dr.B. Kezia Rani

V. Shwetha(1602-19-737-106) Haseeba Yaseen

P. Ashwini(1602-19-737-311)                 Mukesh Kumar Tripathi

**ACKNOWLEDGEMENT**

A project is never the outcome of a single person’s effort. It is a confluence of varied thought processes harmoniously integrated into a resourceful product.

We are immensely grateful to our HOD Sir Mr. Ram Mohan Rao, Mrs. B. Kezia Rani for the constant guidance, support and sagacious suggestions from the beginning of our training till we complete our project.

We are thankful to Mrs. Haseeba Yaseen, Mr. Mukesh Kumar Tripathi for guiding us throughout.

**ABSTRACT**

The most expressive way humans express their moods and emotions is through their facial expressions and to cheer up our mood or to enjoy the current mood ‘Music’ helps in a great way. So our project recognises mood through one’s facial expressions and suggests the best playlist it can to cheer up if we are in a bad mood or to enjoy the best mood one is in. “Vibing Music” scans the facial expressions of a person and recognises the mood using Machine Learning.

“Music can change your mood in a second. Listen to music that makes you smile”

1. **INTRODUCTION**

  1.1 Motivation

There are many styles of music and most of us like some or other types of music. Music reduces heart rate, lower blood pressure, decreases cortisol (stress hormone) levels and increases serotonin and endorphin levels in the blood. It elevates the mood. Music can boost the brain’s production of the hormone dopamine. This increased dopamine production helps relieve feelings of anxiety and depression. Music is processed directly by the amygdala, which is the part of the brain involved in mood and emotions.

A person would love to listen to songs based on the mood to lift up his/her mood. So the web application that we built helps people to listen to the songs based on their moods.

1.2 PROBLEM STATEMENT

People of all age groups love to listen to music. Music helps people uplift their mode. If the person is in a good mood then he/she would like to enjoy their current mood. So by finding the closest emotion of the person and suggesting them a playlist based on their mood can help them enjoy the music more easily and interestingly.

This project is based on Machine Learning libraries(OpenCV and FER) to detect the face and the emotion of the person by his facial expression.

1.3 PROJECT OBJECTIVE

The main objective of the project is to detect the mood of the person based on facial expressions accurately by using the Machine Learning algorithms.

Based on the mood detected, the playlist will be suggested to uplift the mood of the person. The user can add or delete the songs based on his/her choice from the set of available songs.

**2. TECHNICAL LITERATURE**

We have used HTML, CSS, JavaScript and BootStrap for the user interface. Python is used for backend. The database used is DB Browser Sqlite and the framework used is flask. The python libraries containing Machine Learning algorithms are used which are OpenCV and FER. SQLAlchemy is used,  which is a SQL toolkit used to manipulate the database.

**3. EXISTING SYSTEMS**

There are many existing systems which suggest playlists.

* Suggest playlist based on languages: Different languages have different playlists. So based on our preference of language, a playlist is suggested.
* Suggest playlist based on beats: Different music beats like slow beat songs, pop songs, hip hop songs, etc. have different playlist. So based on our music type preference a playlist is suggested.
* Suggest playlist based on Occasions: Playlists based on different occasions like devotional events, party time, etc.

**3.1** DRAWBACKS

There is no application that detects the emotion and suggests the playlist.

For the user to listen to the songs to uplift his/her mood he needs to search for the songs.

* + - other users blogs.

1. PROPOSED SYSTEM:
   1. System Requirements and Specifications
      1. **: Software Specifications** FrontEnd:HTML,CSS,Javascript and bootstrap Backend:Python

Database:Sqlite3 Framework:Flask

#### : FUNCTIONAL REQUIREMENTS

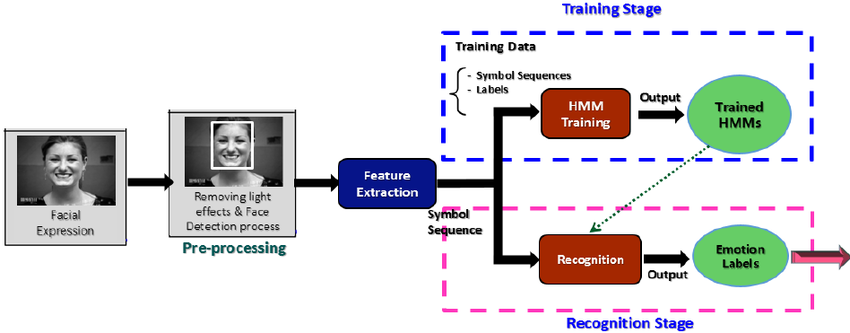
Functional requirements deal with the functionality of the software in the engineering view. The component flow and the structural flow of the same are enhanced and described by it.

The functional requirements for our project include detecting an emotion by capturing an image and suggesting a playlist.

#### :NON-FUNCTIONAL REQUIREMENTS

In order to meet the functional requirements while operating in constrained environment the system had to meet several non-functional requirements that are critical and core to its performance. The non-functional requirements mainly touch on the systems abilities in embedded environment

* 1. **: ARCHITECTURE**



Diagram

Description automatically generated

1. **DESIGN**
   1. UML diagrams/UX diagrams

#### USE CASE DIAGRAM:

Diagram

Description automatically generated

Diagram

Description automatically generated

**FLOWCHART :**

Diagram

Description automatically generated

**TESTING**

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done either manually or using automated tools.

**TEST REPORT**

The testing of the system is performed for various test cases under different conditions considering most of the possible scenarios. To list a few, some of the test reports have been listed below.

Unit Testing:

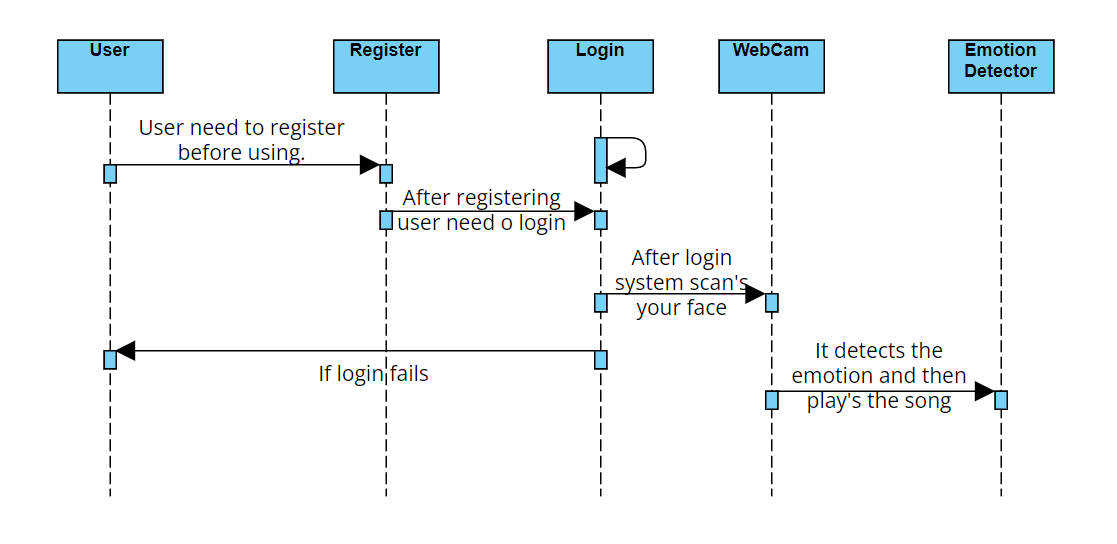
|  |  |
| --- | --- |
| **Test case:** | **UTC-1** |
| Name of the Test: | Face Scanning |
| Item Tested : | Face |
| Sample Input: | Photo of the person with face included |
| Expected Output: | Face Scanning in the frame |
| Actual Output: | Same as expected output |
| Remarks: | Successful |

Functional Testing:

|  |  |
| --- | --- |
| **Test case:** | **FTC-1** |
| Name of the Test: | Emotion Detection |
| Item Tested : | Image captured before |
| Sample Input: | Image of the person |
| Expected Output: | Emotion recognized in the picture |
| Actual Output: | Same as expected output |
| Remarks: | Successful |

|  |  |
| --- | --- |
| **Test case:** | **UTC-3** |
| Name of the Test: | Operating a Play List |
| Item Tested : | Play List |
| Sample Input: | Emotion which is detected earlier |
| Expected Output: | Particular playlist based on emotion |
| Actual Output: | Same as expected output |
| Remarks: | Successful |

**SEQUENCE DIAGRAM:**



* 1. **Database Schema**

1. Login table

|  |  |
| --- | --- |
| USERNAME | PASSWORD |

1. Implementation and Testing

### Screenshots and Testcases

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

A picture containing text, electronics, screenshot, display

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

CODE SNIPPETS:

Emotion Detection Module

from flask import Flask, render\_template  
import base64  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.image as mpimg  
import cv2  
from fer import FER  
import playsound  
  
def save\_img(img\_base64):  
 #binary <- string base64  
 img\_binary = base64.b64decode(img\_base64)  
 print("Hello world")  
 #jpg <- binary  
 img\_jpg = np.frombuffer(img\_binary, dtype=np.uint8)  
 #raw image <- jpg  
 img = cv2.imdecode(img\_jpg, cv2.IMREAD\_COLOR)  
 #Path to save the decoded image  
 image\_file = "./img.jpg"  
 print(type(img))  
  
 cv2.imshow('Image', img)  
 cv2.waitKey()  
 #Save image  
 cv2.imwrite(image\_file, img)  
 input\_image = mpimg.imread("img.jpg")  
 imgplot = plt.imshow(input\_image)  
 # plt.show()  
 emotion\_detector = FER(mtcnn=True)  
 # Output image's information  
 result = emotion\_detector.detect\_emotions(input\_image)  
 print(emotion\_detector.detect\_emotions(input\_image))  
 emotion = {}  
 emotion = result[0]['emotions']  
 Keymax = max(emotion, key=lambda x: emotion[x])  
 print(Keymax)  
 return Keymax

return Keymax

Playlist Module

<!doctype html>  
<html lang="en">  
  
<head>  
 <!-- Required meta tags -->  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
  
 <!-- Bootstrap CSS -->  
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">  
 <style>  
 body {  
 background: darkcyan;  
 height: 100%;  
 }  
  
 img {  
 border-radius: 30px;  
 }  
  
 .container {  
 height: 100%;  
  
 justify-content: center;  
 align-items: center;  
 }  
  
 </style>  
 <script>  
 var ***playlist*** = ["{{url\_for('static', filename='Songs/s1.ogg')}}", "{{url\_for('static', filename='Songs/s2.ogg')}}"];  
 var ***neutral***=["{{url\_for('static', filename='Songs/One-Dance(PagalWorld).mp3')}}", "{{url\_for('static', filename='Songs/Akon - feat. Kat Deluna - Right now (Na na na).mp3')}}", "{{url\_for('static', filename='Songs/Dua Lipa - New Rules.mp3')}}", "{{url\_for('static', filename='Songs/Levitating---Dua-Lipa(pagolworld.nl).mp3')}}", "{{url\_for('static', filename='Songs/Baby(PagalWorld).mp3')}}"];  
 var ***happy*** = ["{{url\_for('static', filename='Songs/s1.ogg')}}", "{{url\_for('static', filename='Songs/s2.ogg')}}", "{{url\_for('static', filename='Songs/s3.ogg')}}", "{{url\_for('static', filename='Songs/s4.ogg')}}", "{{url\_for('static', filename='Songs/s5.ogg')}}"];  
 var ***angry***=["{{url\_for('static', filename='Songs/Harry Potter Theme.mp3')}}", "{{url\_for('static', filename='Songs/Anyone Justin Bieber 128 Kbps.mp3')}}", "{{url\_for('static', filename='Songs/Baila Conmigo Selena Gomez 128 Kbps.mp3')}}", "{{url\_for('static', filename='Songs/s2.ogg')}}", "{{url\_for('static', filename='Songs/s3.ogg')}}"];  
 var ***disgust***=["{{url\_for('static', filename='Songs/One-Dance(PagalWorld).mp3')}}", "{{url\_for('static', filename='Songs/Akon - feat. Kat Deluna - Right now (Na na na).mp3')}}", "{{url\_for('static', filename='Songs/Dua Lipa - New Rules.mp3')}}", "{{url\_for('static', filename='Songs/Levitating---Dua-Lipa(pagolworld.nl).mp3')}}", "{{url\_for('static', filename='Songs/Baby(PagalWorld).mp3')}}"];  
 var ***fear***=["{{url\_for('static', filename='Songs/Harry Potter Theme.mp3')}}", "{{url\_for('static', filename='Songs/Anyone Justin Bieber 128 Kbps.mp3')}}", "{{url\_for('static', filename='Songs/Baila Conmigo Selena Gomez 128 Kbps.mp3')}}", "{{url\_for('static', filename='Songs/s2.ogg')}}", "{{url\_for('static', filename='Songs/s3.ogg')}}"];  
 var ***sad***=["{{url\_for('static', filename='Songs/s1.ogg')}}", "{{url\_for('static', filename='Songs/s2.ogg')}}", "{{url\_for('static', filename='Songs/s3.ogg')}}", "{{url\_for('static', filename='Songs/s4.ogg')}}", "{{url\_for('static', filename='Songs/s5.ogg')}}"];  
 var ***surprise***=["{{url\_for('static', filename='Songs/One-Dance(PagalWorld).mp3')}}", "{{url\_for('static', filename='Songs/Akon - feat. Kat Deluna - Right now (Na na na).mp3')}}", "{{url\_for('static', filename='Songs/Dua Lipa - New Rules.mp3')}}", "{{url\_for('static', filename='Songs/Levitating---Dua-Lipa(pagolworld.nl).mp3')}}", "{{url\_for('static', filename='Songs/Baby(PagalWorld).mp3')}}"];  
 var ***i*** = 0;  
 var ***available***=[];  
 function select(x){  
 ***console***.log(x);  
 if(x=="happy")  
 {  
 ***available***=***Array***.from(***happy***);  
 ***console***.log(***available***);  
 }  
 else if(x=="neutral")  
 {  
 ***available***=***Array***.from(***neutral***);  
 ***console***.log(***available***);  
 }  
 else if(x=="angry")  
 {  
 ***available***=***Array***.from(***angry***);  
 ***console***.log(***available***);  
 }  
 else if(x=="disgust") {  
 ***available***=***Array***.from(***disgust***);  
 ***console***.log(***available***);  
 }  
 else if(x=="fear")  
 {  
 ***available***=***Array***.from(***fear***);  
 ***console***.log(***available***);  
 }  
 else if(x=="sad")  
 {  
 ***available***=***Array***.from(***sad***);  
 ***console***.log(***available***);  
 }  
 else if(x=="surprise")  
 {  
 ***available***=***Array***.from(***surprise***);  
 ***console***.log(***available***);  
 }  
 else {  
 ***available***=***Array***.from(***disgust***);  
 ***console***.log(***available***);  
 }  
 }  
 function add() {  
 let val = parseInt(prompt("1.s1.mp3\n2.s2.mp3\n3.s3.mp3\n4.s4.mp3\n5.s5.mp3\nEnter song number:"));  
 //console.log(val);  
 ***playlist***.push(***available***[val]);  
 //console.log(playlist);  
 var a = ***document***.getElementById("list");  
 ***console***.log(a)  
 var li = ***document***.createElement("li");  
 ***console***.log(li)  
 li.setAttribute('id', ***available***[val]);  
 li.appendChild(***document***.createTextNode(***available***[val - 1]));  
 a.appendChild(li);  
 }  
  
 function del() {  
 let val = parseInt(prompt("Enter song number from the displayed playlist on the page:"));  
 var a = ***document***.getElementById("list");  
 //var candidate = document.getElementById("candidate");  
 ***console***.log(***playlist***[val - 1]);  
 var item = ***document***.getElementById(***playlist***[val - 1]);  
 //console.log(item);  
 a.removeChild(item);  
 if (val != -1) {  
 ***playlist***.splice(val - 1, 1);  
 }  
 }  
  
 function next() {  
 if (***i*** >= ***playlist***.length)  
 ***i*** = 0;  
 ***document***.getElementById("d3").src = ***playlist***[***i***];  
 ***i***++;  
  
 function playAudio() {  
 ***x***.play();  
 }  
  
 function pauseAudio() {  
 ***x***.pause();  
 }  
 }  
  
 </script>  
</head>  
  
  
<body>  
<!-- <script>select({{emotion|tojson}});</script>-->  
 <div class="container">  
 <div class="row no-gutters">  
 <div class="col-lg-6">  
 <div class="row no-gutters">  
 <h1>Hey! We got you &#128516;.</h1>  
 </div>  
 <div class="row">  
 <img src="https://media.istockphoto.com/vectors/vector-paper-cut-craft-style-music-composition-for-jazz-concert-vector-id1266921478?k=20&m=1266921478&s=612x612&w=0&h=EOaSX76xpg5brpCSbFiLGTSnJuHDF-NSgYf4-tFXgY4=" class="img-fluid" alt="">  
 </div>  
 <br>  
<!-- <var>x</var>=<var>{{emotion}}</var>-->  
<!-- <script >play(x);</script> -->  
 <div class="row">  
 <audio controls id="d3">  
 <source src={{url\_for('static', filename='Songs/s1.ogg')}} type="audio/ogg">  
 </audio>  
 <button class=" btn"onclick="next()">NEXT</button>  
 </div>  
 {% if emotion=="happy" %}  
 <script>select("happy")</script>  
 {% elif emotion=='surprise' %}  
 <script>select("surprise")</script>  
 {% elif emotion=='neutral' %}  
 <script>select("neutral")</script>  
 {% elif emotion=='fear' %}  
 <script>select("fear")</script>  
 {% elif emotion=='angry' %}  
 <script>select("angry")</script>  
 {% elif emotion=='sad' %}  
 <script>select("sad")</script>  
 {% else %}  
 <script>select("disgust")</script>  
 {% endif %}  
 </div>  
 <div class="col-lg-1"></div>  
 <div class="col-lg-5">  
 <div class="row no-gutters">  
 <h3>Wanna create your own playlist? Go for it &#128521;</h3>  
 </div>  
 <br>  
 <div class="row">  
 <ol id="list">  
 <li id="s1.mp3">harry-porter-theme-song.mp3</li>  
 <li id="s2.mp3">ninnele.mp3</li>  
 </ol>  
 </div>  
 <div class="row"><button onclick="add()" class="btn">ADD</button>  
 <button onclick="del()" class="btn">DELETE</button></div>  
 </div>  
 </div>  
 </div>  
  
</body>

### Results

We were able to execute all the modules successfully. And also able to register and login . After logging in we were able to detect an emotion and suggest a playlist.

**CONCLUSION**

The current project developed is a web application. It is a website where you can vibe on a music based on your mood. Many people listen to the songs based on there mood, it may be happy, sad or more. So in this project we play a song according to your mood. Firstly system scans the user’s face and takes a snapshot. Secondly, it detects the emotion captured on the image. Finally, it plays a song accordingly.

**FUTURE SCOPE**

# The Vibing Music is an emotion based music suggesting application which is really interesting to use. It can be followed by different age groups. It can be extended by adding interesting features like giving dynamic information to the user i.e., How many songs a user listened on that particular day or which is the most listened song. We can also collect different information of a user to make it interesting. In database we can also add personalized playlist for every emotion of a user.

References

[1]<https://towardsdatascience.com/face-detection-recognition-and-emotion-detection-in-8-lines-of-code-b2ce32d4d5de>

[2]<https://www.geeksforgeeks.org/facial-expression-recognizer-using-fer-using-deep-neural-net/>

[3] <https://www.geeksforgeeks.org/how-to-capture-a-image-from-webcam-in-python/>

[4] H. Immanuel James, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, R. Saranya” EMOTION BASED MUSIC RECOMMENDATION SYSTEM”: pISSN: 2395-0072 , IRJET 2019

[5] https://www.irjet.net/archives/V8/i7/IRJET-V8I7457.pdf