

**FACE EMOTION BASED MUSIC PLAYER**

**A Project Report**

**In Partial fulfilment for the award of the degree**

***Of***

**BACHELOR OF TECHNOLOGY AND ENGINEERING**

***IN***

**COMPUTER SCIENCE AND ENGINEERING**

## *At*

**Aravali College of Engineering and Management**

**Jasana, Tigaon Rd, Neharpar Faridabad, Haryana, 121002**

**(AFFILATED TO YMCA UNIVERSITY OF SCIENCE AND TECHNOLOGY, FARIDABAD(INDIA)**

**August-2022**

**Submitted To: SUBMITTED BY:**

Ms. Shivani Pahuja Shubham Bharti (19011004084)

Ankush (20011004505)

**DECLARATION**

I hereby declare that the project work entitled “**Face Emotion Based Music Player”** submitted to **YMCA, Faridabad (Haryana)**, is a record of an original work done by me under the guidance of **Ms. Shivani Pahuja Ma’am** in Computer Science and Engineering, ARAVALI COLLEGE OF ENGINEERING AND MANAGEMENT, Faridabad.

This project is submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Shubham Bharti Date: Aug 2022

Ankush

**Acknowledgment**

Presentation inspiration and motivation have always played a key role in the success of any venture.

We express our sincere thanks to **Respected Dean sir, Aravali College of Engineering & Management.**

We pay our deep sense of gratitude to **Prof. Sakshi Sachdeva Ma’am (HOD) of CSE department, Aravali College of Engineering & Management**, Faridabad to encourage me to highest peak and to provide me the opportunity to prepare the project. We are immensely obliged to my friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of my project. We feel to acknowledge my indebtedness and deep sense of gratitude to my guide and subject teacher **Ms.Shivani Pahuja Ma’am** whose valuable guidance and kind supervision given to me throughout the course which shaped the present project as it is shown.

Last but not the least, **Our Parents** are also an important inspiration for me. So, with due regards, we express my gratitude to them.

Shubham Bharti

Ankush

**CERTIFICATE**

This is to certify that the project titled is “**FACE EMOTION BASED MUSIC PLAYER**” is the bona fide work carried out by Shubham Bharti, Ankush, student of B Tech (CSE) of **Aravali College of Engineering and Management** in Faridabad. Affiliated to YMCA University of Science and Technology, Faridabad during the academic year 2021-22, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Ms. Shivani Pahuja Signature

Project Guide(Internal)

Date:

**COUNTER SIGNED BY**

Name: Signature

Date:

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE OF CONTENTS** | | | |
|  | | | **Page No** |
| Declaration | | | i |
| Acknowledgement | | | ii |
| Certificate | | | iii |
|  | | |  |
| **Chapter 1** | | **INTRODUCTION** |  |
|  | 1.1 | Abstract | 2 |
|  | 1.2 | Scope of the Project | 3 |
|  | 1.3 | Dataflow Chart | 4 |
|  | 1.4 | Merits of the system proposed | 5 |
|  | 1.5 | Application of the system proposed | 6 |
|  | | |  |
| **Chapter 2** | | **TECHNOLOGY USED** |  |
|  | 3.1 | PYTHON | 8 |
|  | 3.2 | HTML | 9 |
|  | 3.3 | CSS | 11 |
|  | 3.4 | PYCHARM | 13 |
|  | 3.5 | OPENCV | 14 |
|  | 3.6 | KERAS | 14 |
|  | 3.7 | NUMPY | 15 |
|  | 3.8 | HARDWARE USED | 15 |
|  | | |  |
| **Chapter 3** | | **SCREENSHOTS** | 16-20 |
|  | | **CONCLUSION** | 21 |
|  | | **REFERENCE** | 22 |
|  | | **LINKS** | 23 |

**CHAPTER – 1**

**EMOTION BASED MUSIC PLAYER**

**ABSTRACT**

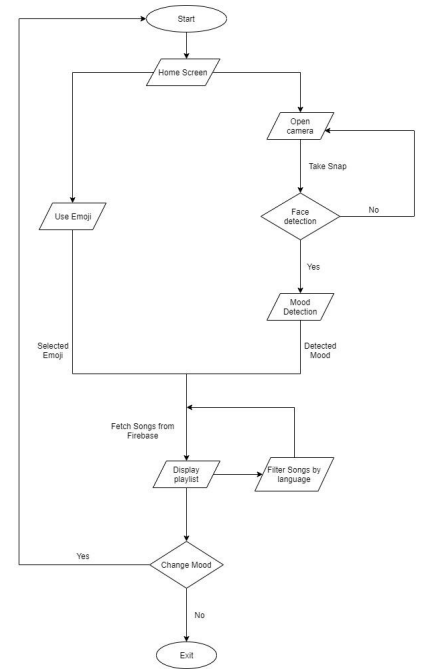
Recent studies confirm that humans respond and react to music and that music has a high impact on person’s brain activity. Many people likes to listen music often. Many of people have hobby to listen music. So music plays an important role in one’s life. People react to music by their emotions. People tend to listen to music based on their mood and interests. This project focuses on creating an application to suggest songs for user based on their emotions by differentiating different kind of moods. This application is only based on moods and emotion depending on user’s mood. For developing this application we are using python.

Music is an important medium of life. Everyone likes to listen music but what if we made more easier for user to listen music by differentiating their mood? Now a days we have grown up with advanced technology. Many mobile applications have come up with many functions like emotion capturing with artificial intelligence, gesture, sleeper time, lyrics, shuffle play and many more. This application is randomly similar that off but here we came up with new function called Emotion Based Music Player with many different Moods. Here user can select music list depending on their current emotion or we can also call as mood so after clicking to different types of mood a list of songs would be displayed. And user can also search their favourite songs by searching in-search tab. User can also create their playlist. This application is made for online as well as offline too. User can easily handle this application because many of us have been using different kind of music players day-by-day.

**SCOPE OF THE PROJECT**

Project Emotion Based Music Player is a novel approach which helps user to play the songs according to their emotion / mood. In android we can apply many features. This application is made for android users only from seventh version which is (Nought) and further more. Firstly, user have to sign up or login to access the application and then can according search songs or by just clicking on moods user can also access songs play list. user can also access this application on desktop too. To access in desktop user, need windows7 or further version. User can also upload his / her playlist from device. Also, can download songs and also can share from our application. In this application dark as well as light theme are provided for user eye comfort. Users can boost their sound in equalizer. Our application does not contain any add because we have observed in many famous applications they put add in middle and we get disturbed so just to make user happy we haven’t collaborated with any type of article add. And we believe in equality so we haven’t added any premium package this application is same for all user.

**DATAFLOW CHART**

****

### **MERITS OF THE SYSTEM PROPOSED**

* The system is easy to use and is fully authorized.
* It can be used to listen music in every type of mood.
* It is not an electronic based system so one can easily make use of laptops to use this system.
* It is free of cost.
* It also saves the time to find different types of music.
* It maximizes accuracy and reduces storage space usage.
* More feature such as
  + Find every language of songs
  + Listen your favourite singer
  + much more...

**APPLICATION OF THE SYSTEM PROPOSED**

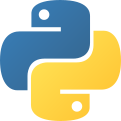
* Plays songs according to user mood and emotions.
* Act as plugins for website.
* Recommended for YouTube.

**CHAPTER – 2**

**TECHNOLOGY USED**

**1. Python:** Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.



**Figure 2.1 of PYTHON**

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages.

**History**

[](https://en.wikipedia.org/wiki/File:Guido_van_Rossum_OSCON_2006_cropped.png)

**Figure 2.2 of The designer of Python, Guido van Rossum, at OSCON 2006**

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent

dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x[]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-3000-47) and 2.7.x. Releases of Python 3 include the utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No further security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later were supported. Later, support for 3.6 was also discontinued. In 2021, Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues leading to possible remote code execution and web cache poisoning.

In 2022, Python 3.10.4 and 3.9.12 were expedited and so were older releases including 3.8.13, and 3.7.13 because of many security issues. Python 3.9.13 is the latest 3.9 version, and from now on 3.9 (and older; 3.8 and 3.7) will only get security updates.

**2. HTML: HTML**(HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)) or functionality/behavior ([JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head), [<title>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title), [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body), [<header>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header), [<footer>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer), [<article>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article), [<section>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section), [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p), [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div), [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span), [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img), [<aside>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/aside), [<audio>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio), [<canvas>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas), [<datalist>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist), [<details>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/details), [<embed>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/embed), [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav), [<output>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/output), [<progress>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress), [<video>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video), [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul), [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol), [<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li) and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

**Beginner's tutorials**

Our HTML Learning Area features multiple modules that teach HTML from the ground up — no previous knowledge required.

*Introduction to HTML*

This module sets the stage, getting you used to important concepts and syntax such as looking at applying HTML to text, how to create hyperlinks, and how to use HTML to structure a web page.

*Multimedia and embedding*

This module explores how to use HTML to include multimedia in your web pages, including the different ways that images can be included, and how to embed video, audio, and even entire other webpages.

*HTML tables*

Representing tabular data on a webpage in an understandable, accessible way can be a challenge. This module covers basic table markup, along with more complex features such as implementing captions and summaries.

*HTML forms*

Forms are a very important part of the Web — these provide much of the functionality you need for interacting with websites, e.g. registering and logging in, sending feedback, buying products, and more. This module gets you started with creating the client-side/front-end parts of forms.

*Use HTML to solve common problems*

Provides links to sections of content explaining how to use HTML to solve very common problems when creating a web page: dealing with titles, adding images or videos, emphasizing content, creating a basic form, etc.

**Advanced topics**

*CORS enabled image*

The crossorigin attribute, in combination with an appropriate CORS header, allows images defined by the <img> element to be loaded from foreign origins and used in a <canvas> element as if they were being loaded from the current origin.

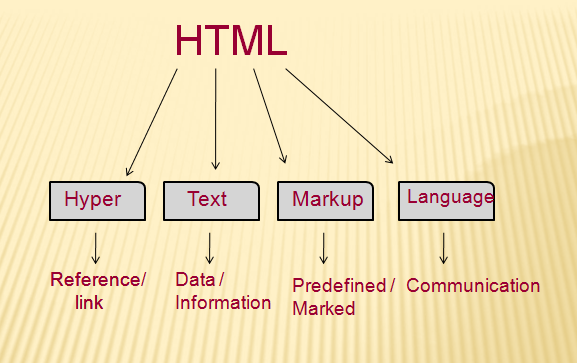
*CORS settings attributes*

Some HTML elements that provide support for CORS, such as <img> or <video>, have a crossorigin attribute (crossOrigin property), which lets you configure the CORS requests for the element's fetched data.

*Preloading content with rel="preload"*

The preload value of the [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) element's [rel](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link" \l "attr-rel) attribute allows you to write declarative fetch requests in your HTML [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head), specifying resources that your pages will need

very soon after loading, which you therefore want to start preloading early in the lifecycle of a page load, before the browser's main rendering machinery kicks in. This ensures that they are made available earlier and are less likely to block the page's first render, leading to performance improvements. This article provides a basic guide to how preload works.



**Figure 2.3 of HTML**

**3. CSS: Cascading Style Sheets** (**CSS**) is a [stylesheet](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) language used to describe the presentation of a document written in [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) or [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction) (including XML dialects such as [SVG](https://developer.mozilla.org/en-US/docs/Web/SVG), [MathML](https://developer.mozilla.org/en-US/docs/Web/MathML) or [XHTML](https://developer.mozilla.org/en-US/docs/Glossary/XHTML)). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the **open web** and is standardized across Web browsers according to [W3C specifications](https://www.w3.org/Style/CSS/#specs). Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to [develop and release recommendations separately per module](https://www.w3.org/Style/CSS/current-work). Instead of versioning the CSS specification, W3C now periodically takes a snapshot of [the latest stable state of the CSS specification](https://www.w3.org/TR/css/).



**Figure 2.4 of CSS**

Our [CSS Learning Area](https://developer.mozilla.org/en-US/docs/Learn/CSS) features multiple modules that teach CSS from the ground up — no previous knowledge required.

[*CSS first steps*](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps)

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

[*CSS building blocks*](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks)

This module carries on where [CSS first steps](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps) left off — now you've gained familiarity with the language and its syntax, and got some basic experience with using it, it's time to dive a bit deeper. This module looks at the cascade and inheritance, all the selector types we have available, units, sizing, styling backgrounds and borders, debugging, and lots more.

The aim here is to provide you with a toolkit for writing competent CSS and help you understand all the essential theory, before moving on to more specific disciplines like [text styling](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text) and [CSS layout](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout).

[*CSS styling text*](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text)

With the basics of the CSS language covered, the next CSS topic for you to concentrate on is styling text — one of the most common things you'll do with CSS. Here we look at text styling fundamentals, including setting font, boldness, italics, line and letter spacing, drop shadows, and other text features. We round off the module by looking at applying custom fonts to your page, and styling lists and links.

[*CSS layout*](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout)

At this point we've already looked at CSS fundamentals, how to style text, and how to style and manipulate the boxes that your content sits inside. Now it's time to look at how to place your boxes in the right place in relation to the viewport, and to each other. We have covered the necessary prerequisites so we can now dive deep into CSS layout, looking at different display settings, modern layout tools like flexbox, CSS grid, and positioning, and some of the legacy techniques you might still want to know about.

[*Use CSS to solve common problems*](https://developer.mozilla.org/en-US/docs/Learn/CSS/Howto)

This module provides links to sections of content explaining how to use CSS to solve common problems when creating a web page.

**3. PYCHARM:** PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects

* Code completion and inspection
* Advanced debugging
* Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −

### **Code Completion**

PyCharm enables smoother code completion whether it is for built in or for an external package.

### **SQLAlchemy as Debugger**

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

### **Git Visualization in Editor**

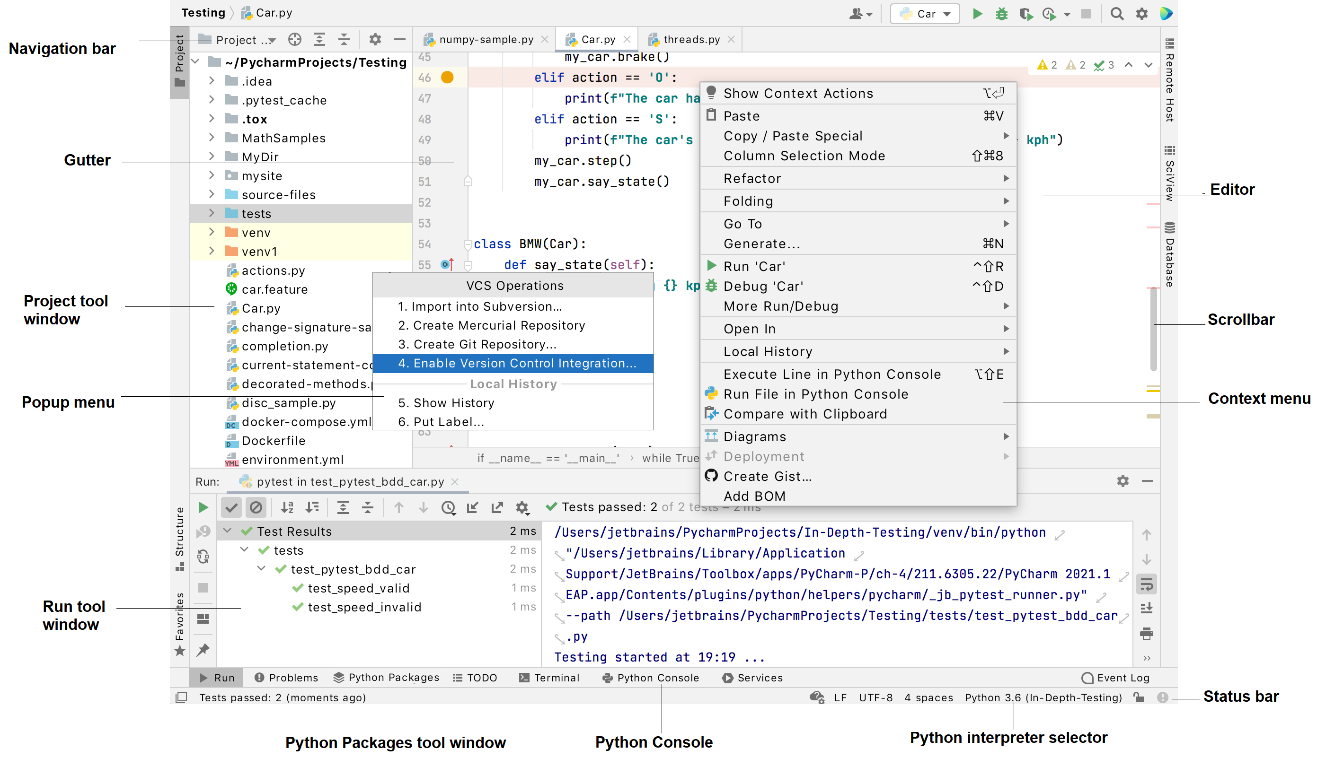
When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

### **Code Coverage in Editor**

You can run **.py** files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

### **Package Management**

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.



**Figure 2.5 of PYCHARM Interface**

**4. OPENCV:** OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

**5. KERAS:** Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user-friendly, extensible, and modular for

facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

It cannot handle low-level computations, so it makes use of the **Backend** library to resolve it. The backend library act as a high-level API wrapper for the low-level API, which lets it run on TensorFlow, CNTK, or Theano.

**5. NUMPY:** NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

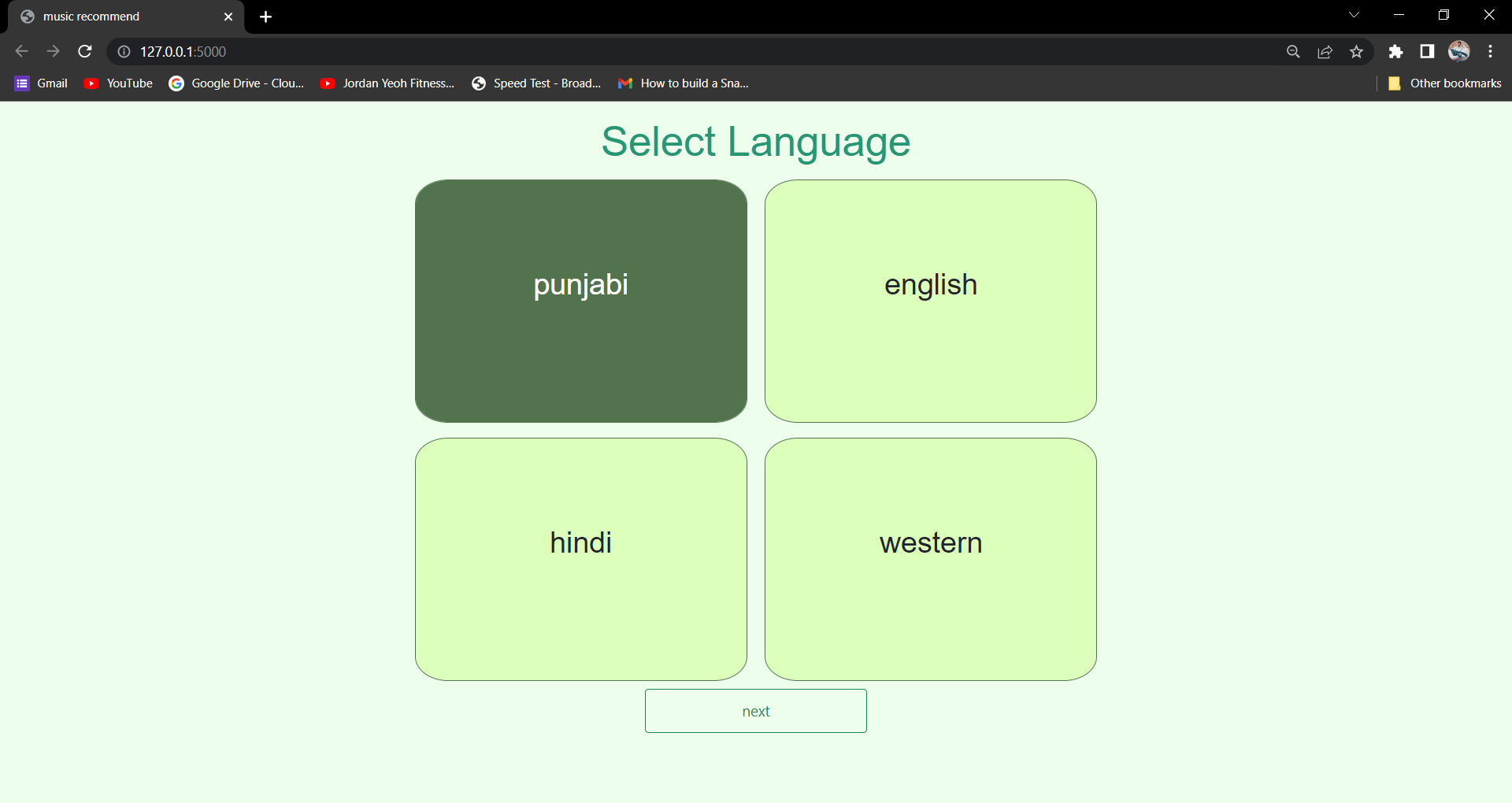
**6. HARDWARE USED:** In terms of hardware requirements there is not much required at all but still below requirements are must:

* Working PC or Laptop
* Processor: - Ryzen 5 3450U.
* RAM: - 8 GB RAM.
* Space Required: - 500MB.
* Webcam with drivers installed
* Lan connection/ Internet service
* Flashlight/ LED if using this at night.

**CHAPTER – 3**

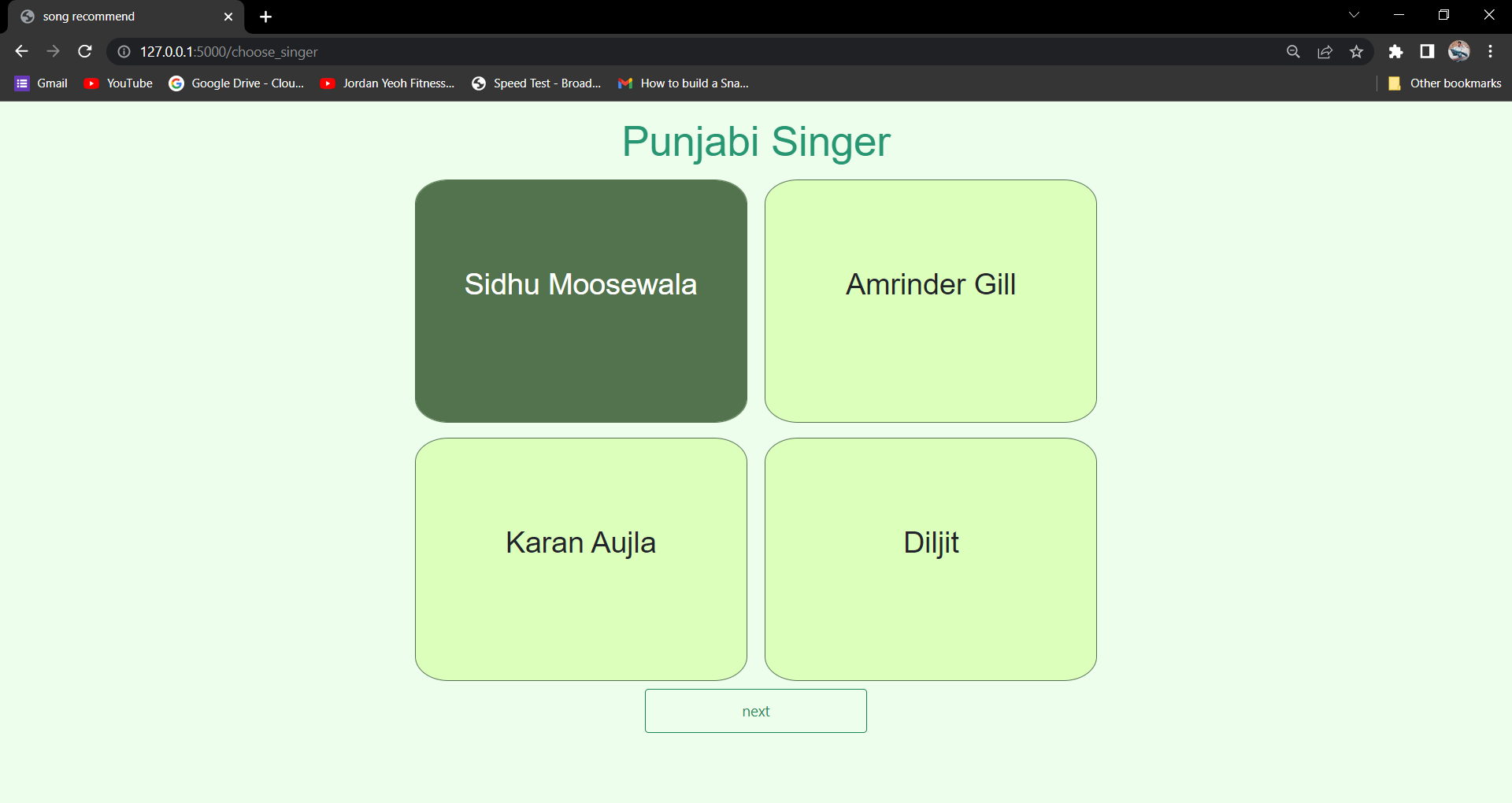
**SCREENSHOT**

* **Language Selection**

******

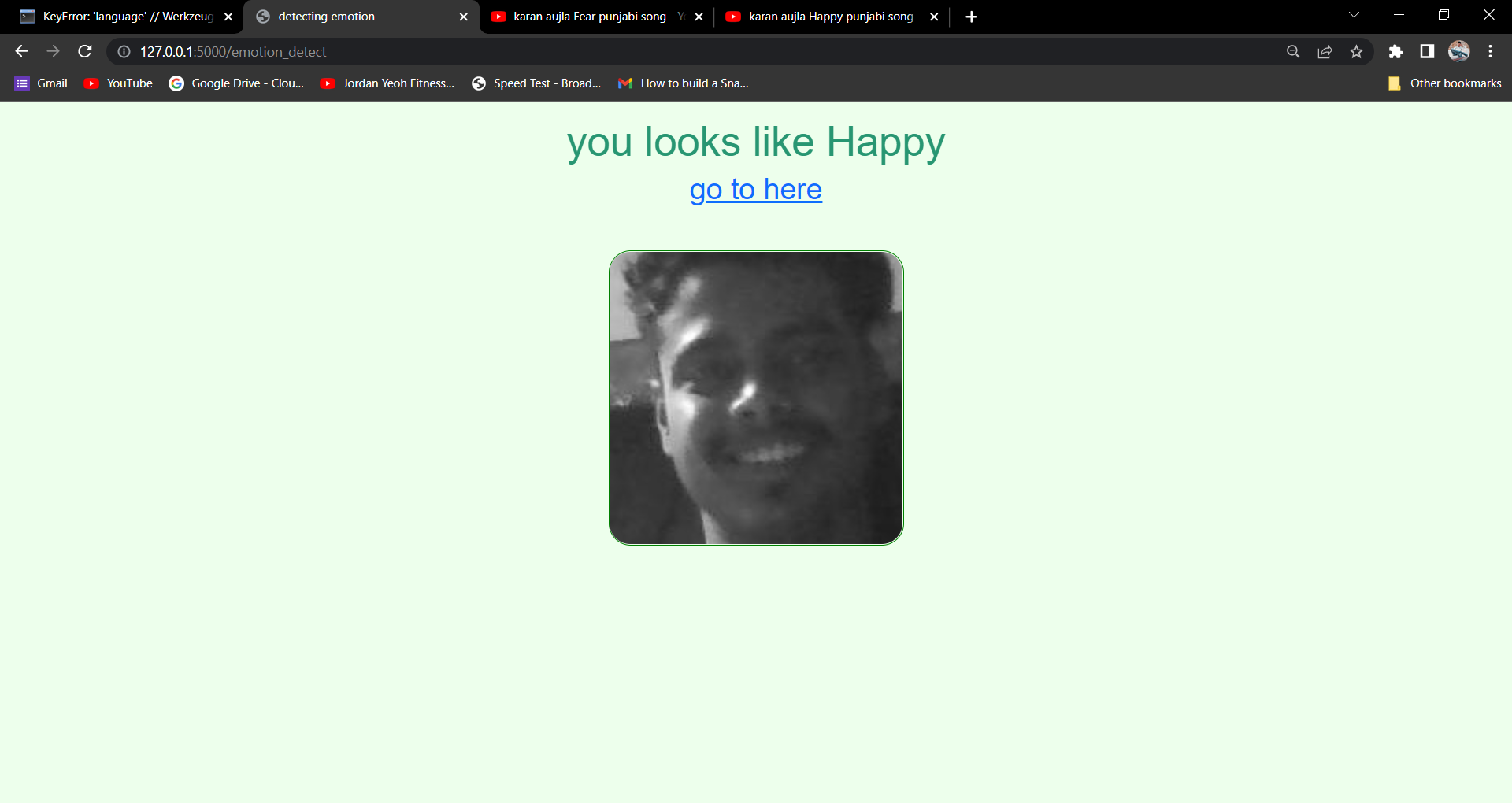
**Figure 3.1 of select the language of the music which wants to listen.**

* **Favourite Singer Selection**



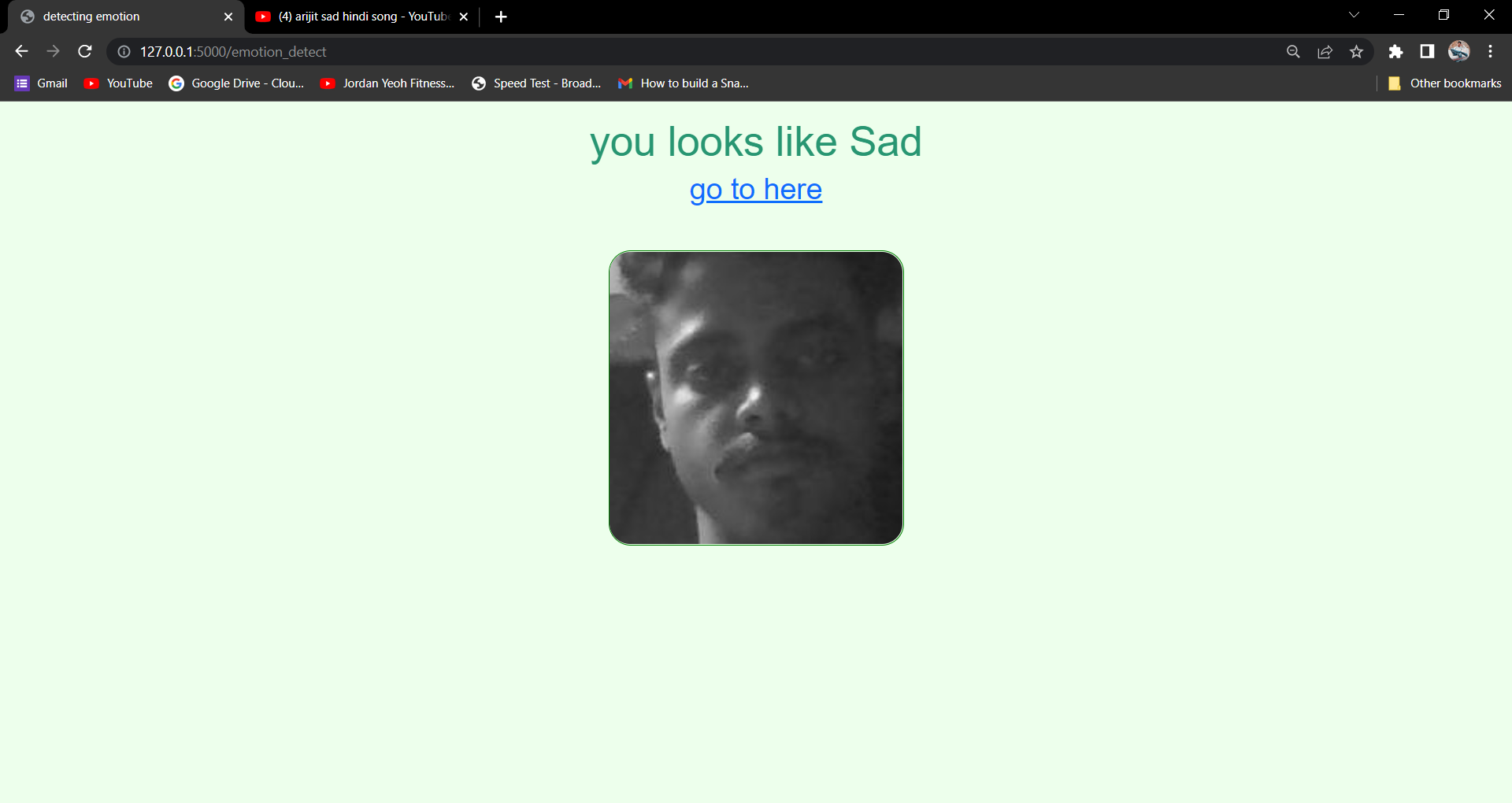
**Figure 3.2 of Select your favourite singer.**

* **Face Detection**
  + *Happy Face Detection*



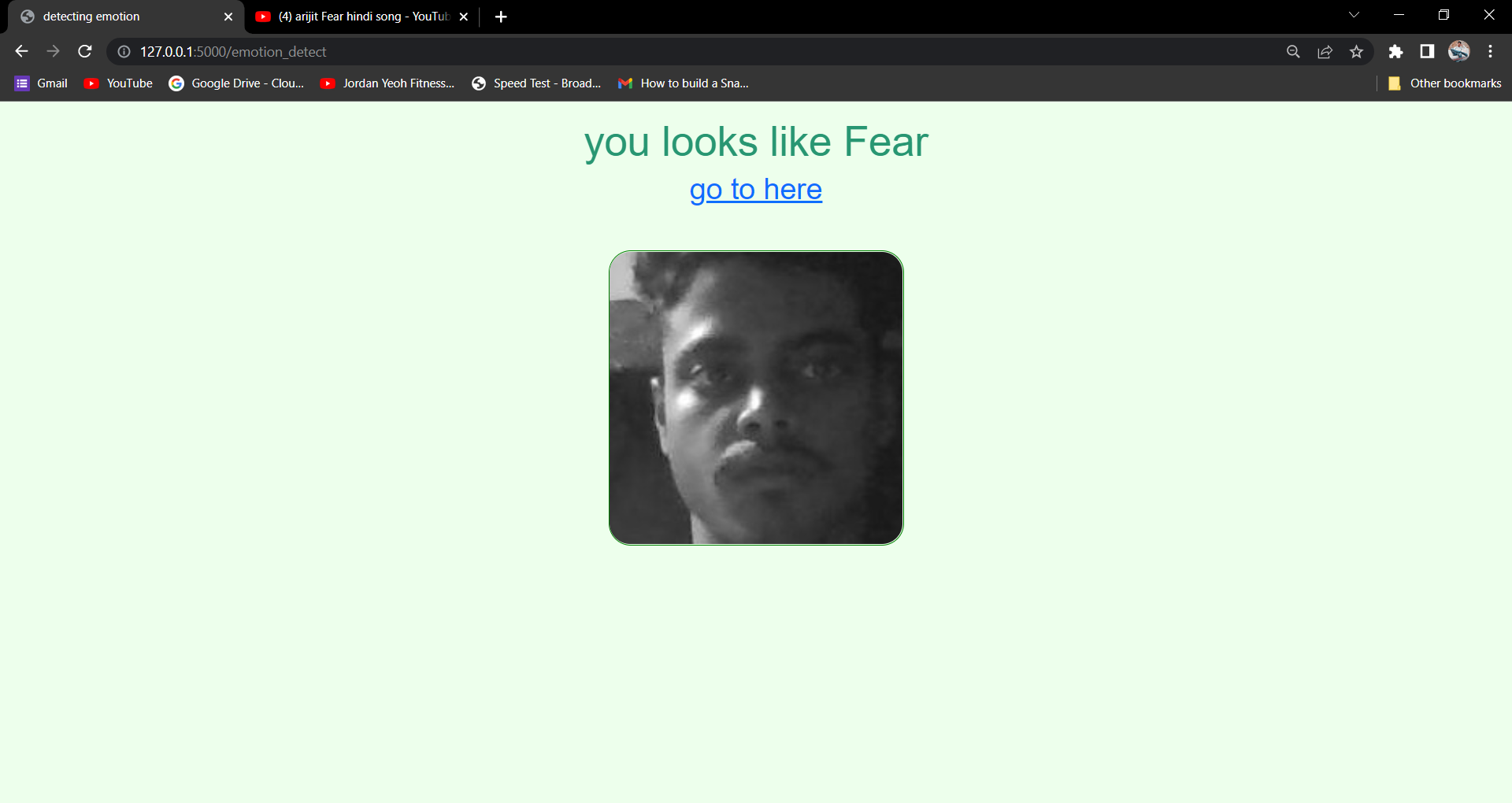
**Figure 3.3 of Happy Face Detected**

* + *Sad Face Detection*



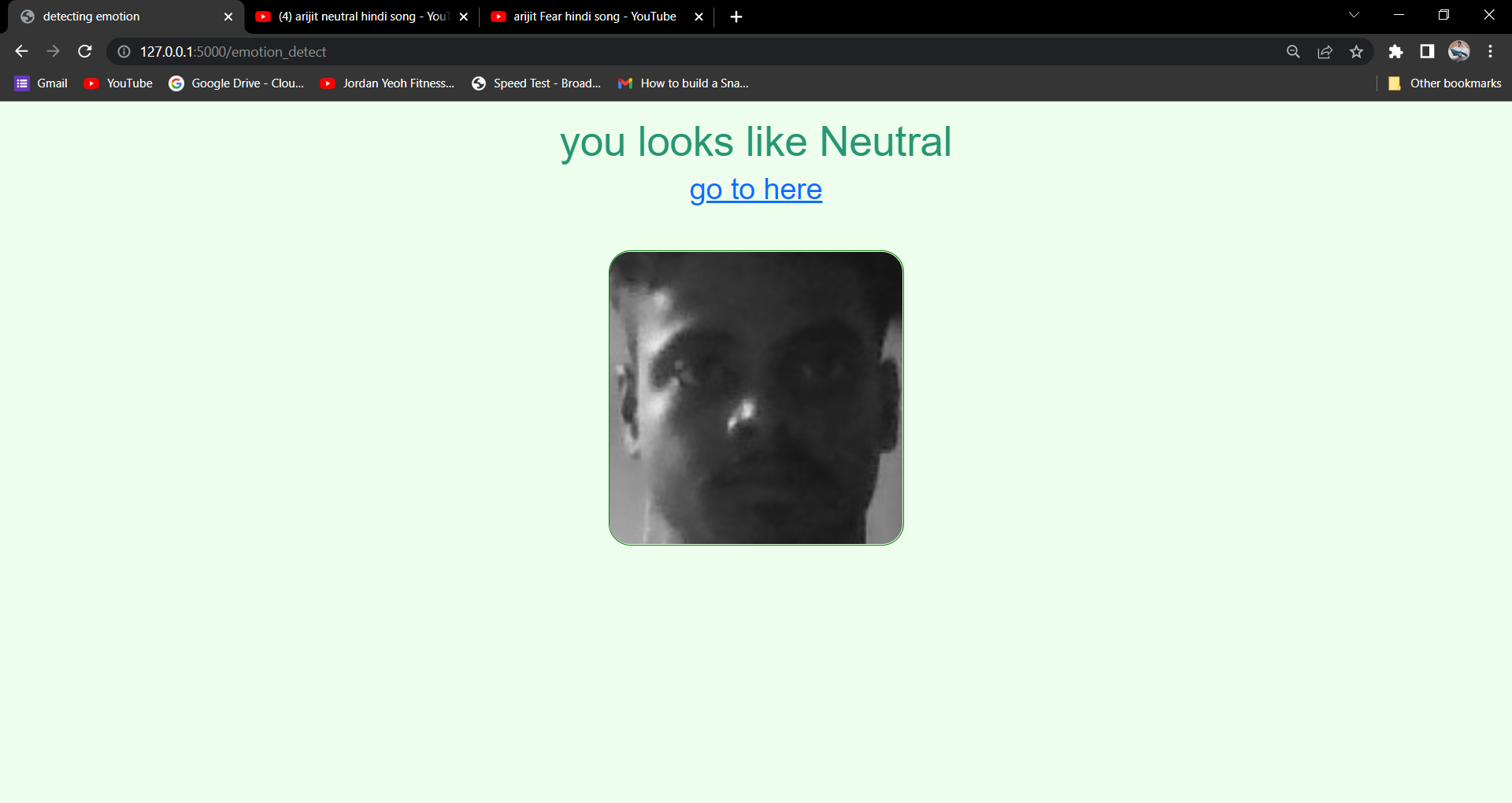
**Figure 3.4 of Sad Face Detected**

* + *Fear Face Detection*



**Figure 3.5 of Fear face Detected**

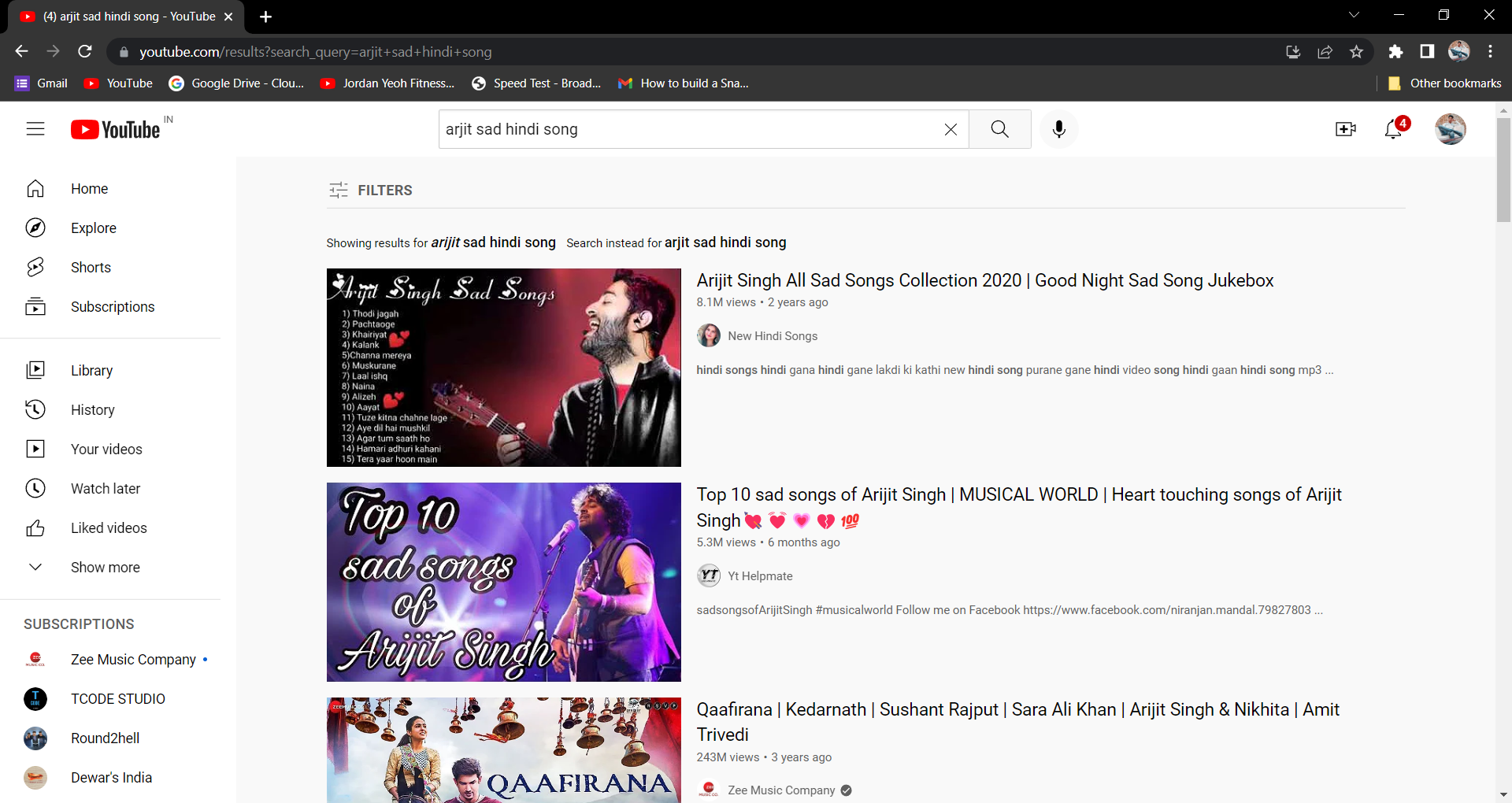
* + *Neutral Face Detection*



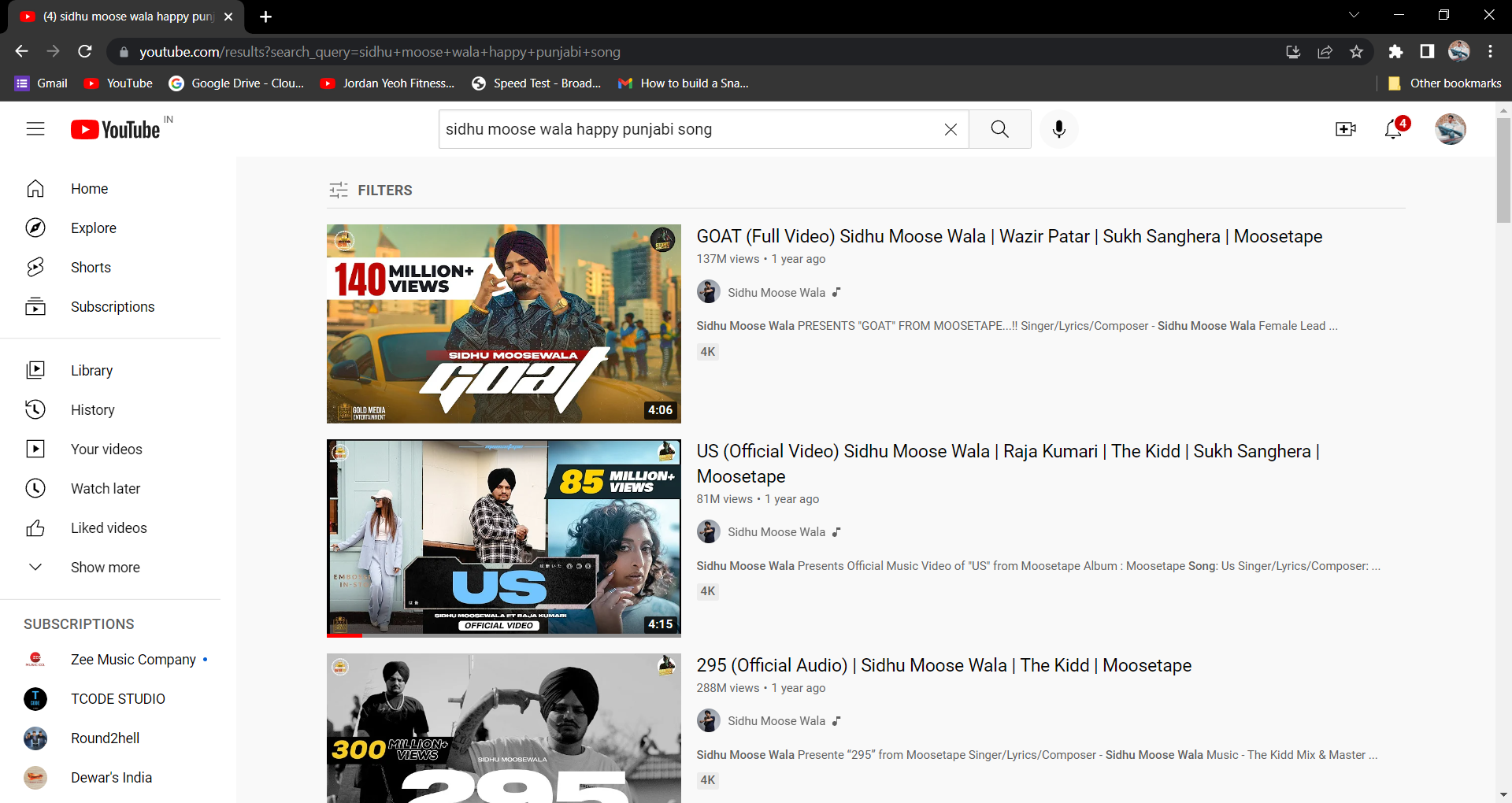
**Figure 3.6 of detect Neutral.**

And many more emotions….

* **YouTube Screen**



**Figure 2.5 of Shows playlist according to the Language, Singer and Emotion of the User’s. Here it shows Hindi Songs of singer Arijit and according to the Emotion here it detects Sad face.**



**Figure 2.5 of Here it shows Punjabi Songs of singer Sidhu Moose Wala and according to the Emotion here it detects Happy Face.**

**CONCLUSION**

This Project has been developed to give us a great advancement in the field of human behaviour. In this project we have used more than 9 moods which can be ordinarily found in humans. Emotion Based Music Player fulfil to sort out music based on moods. Our main aim is to consume user’s time and to satisfy them. We have designed this platform in such a manner that it can run in Web browser.

**REFERENCES**

* S, P. J., Rajashree, S. M., S. H., & .J, S. S. (2018). Review on Facial Expression Based Music Player. International Journal of Engineering Research & Technology ( IJERT ), 6(15), 1-3.
* Cherry, K. (2018). The 6 Types of Basic Emotions and Their Effect on Human Behavior. Retrieved from Very well Mind: https://www.verywellmind.com/about-us.
* Developer, A. (n.d.). Create an Android project. Retrieved from developers: https://developer.android.com/training/basics/firstapp/creating-project
* ED, B. (2015). Hello, Android. In S. D. Pfalzer (Ed.), Hello, Android (Vol. 4, p. 319). Pragmatic Bookshelf.
* H. P., Adarsh, .B, A. C., & Kumar, V. A. (2018, April). Emotion Based Music Player. International Research Journal of Engineering and Technology ( IRJET ), 05(04), 1-6.

**LINKS**

* [www.google.com](http://www.google.com)
* [www.pypi.org](http://www.pypi.org)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [www.github.com](http://www.github.com)
* [www.youtube.com](http://www.youtube.com)