

Project Report
on
EMOTION BASED RECOMMEDATION
SYSTEM USING DEEP LEARNING

Submitted by
Sk.Yasmin(R170546)

Under the guidance of

A.Mahendra
Assistant Professor

Department of Computer Science and Engineering



**Rajiv Gandhi University of Knowledge and Technologies(RGUKT),
R.K.Valley, Kadapa, Andra Pradesh.**



Rajiv Gandhi University of Knowledge Technologies

RK Valley, Kadapa (Dist), Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled “**EMOTION BASED RECOMMENDATION SYSTEM USING DEEPFACE**” is a bonafied Project work submitted by Sk.Yasmin in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfillment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering for the year 2022-2023 carried out the work under the supervision.

GUIDE

A.MAHENDRA

HEAD OF THE DEPARTMENT

N.SATYANANDARAM

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be Incomplete without the mention of the people who made it possible and whose constant and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.N.SATYANANDARAM for his encouragement , overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Mr.A.MAHENDRA for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

S.NO	INDEX	Page No
1	Abstract	5
2	Introduction	6
2.1	Motivation	6
2.2	Problem Statement	6
2.3	Objective of project	6
2.4	Limitations of project	6
3	Literature Survey	7
3.1	Introduction	7
3.2	Existing System	7
3.2.1	Disadvantages	7
3.3	Proposed System	8
3.3.1	Advantages	8
4	System Analysis	9
5	Design	10-16
6	Implementation	17-20
7	System Testing	21-22
8	Source code	23-28
9	Results	29-32
10	conclusion	33
11	Reference	34

ABSTRACT

Emotion are an integral part of human psychology and are of various kinds. Emotions arise as a function of physiological arousal state which is both subjective and private to an individual. Facial expressions play an important role in detecting human emotions and current state of mind. Machine Learning can detect the emotions by capturing facial expressions and learning what each expression means. It also applies that knowledge to new information presented. In recent years, machine learning has achieved success in computer vision and recommendation systems.

It has benefits on machine learning-based recommendation algorithms and has made remarkable progress in the following three aspects:

- Powerful representation learning capability
- Collaborative / Content-based filtering
- Deep interaction between features

Recommender systems have traditionally relied on data centric descriptors for content and user modeling. In recent years we have witnessed an increasing number of attempts to use emotions in different ways to improve the quality of recommender systems. The Automatic recommendation is one of those considerable techniques. Various media sharing platforms are the experimental areas that are automatic machine to collect human-like characteristics. Humans express non-verbal and involuntary channels in many ways, mainly facial expressions. By calculating human facial expressions, we are going to develop an emotion-based recommendation system.

2.INTRODUCTION

2.1 MOTIVATION

Music is important in everyone's life. It plays a important role in enhancing a person's life. Most music-loving users find themselves in an odd situation when they do not find songs to suit their mood in the situation. Ever since computers were developed, scientists and engineers have thought of artificially intelligent systems that that are mentally and/or physically equivalent to humans.

In today's world, with the development in technology and multimedia, there are many music players which have various features like fast forward, variable playback speed, local playback, streaming playback with multicast stream. Despite the fact that these features meet the basic needs of the user, the user still faces the task of manually selecting the songs through the playlist of songs based on their current mood and behavior. So we have came up with an idea of emotion based music player using Facial Attribute Analysis

2.2 PROBLEM STATEMENT

Music listeners have a tough time creating and segregating the play-list manually when they have hundreds of songs. It is also difficult to keep track of all the songs, sometimes songs that are added and never used, wasting a lot of device memory and forcing the user to find and delete songs manually. Users have to manually select songs every time based on interest and mood. Users also have difficulty to re-organize and play music when play-style varies.

Currently in existing application, music is organized using manual list, and songs cannot be modified or altered in one click. Users have to manually change or update each song in their play-list every time. The sequence of songs in a play-list might not be the same every time, and songs that a user wants to listen frequently might not be given priority or might be left out from the list. Currently, there are no applications that allows users to play.

2.3 OBJECTIVE OF PROJECT

Emotion based music player using deep face(facial attribute analysis) requires the application to take a picture and analyze the facial expressions, it then detects the emotion and generates a playlist. It also displays the mood of the person along with the picture. It displays the playlist of songs. The user needs to select a song from that list and play it. The user can avail the functionality of volume controls.

2.4 LIMITATIONS OF PROJECT

- This project doesn't give the user to create a new playlist every time he/she doesn't like the songs.
- It doesn't keep the track record of the songs being played by the users. Keeping a track record helps determine the songs being played the most.
- This further could be helpful in creating Machine Learning based automatic playlists.
- It allows the user to play a song from a pre-determined playlist. The user could change the song but can't edit the current playlist.

3. LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

Literature survey is something when you look at a literature (publications) in a surface level, or an Ariel view. It incorporates the study of place people and productions are setting of research.

It is phase where the analyst tries to know about what is all the literature related with one range of interest. Also, the relevant literature works are short-listed. Moreover, literature survey guides or helps the researcher to define/find out/identify a problem.

3.1 INTRODUCTION

The main purpose of the literature review work was to survey previous studies on knowledge sharing and intranets. In this, we look into the details about the existing system and try to reduce the disadvantages of the existing system. We try to improve the performance and the efficiency of the new proposed system and also learn the advantages of proposed system.

3.2 EXISTING 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 and sometimes even imparts a therapeutic approach. Listening to music has to be in a facilitated way, that the player will be able to play the song in accordance to the person's mood.

The existing system has a system for emotion recognition that is capable of detecting the user emotions and plays a song that can improve the user's mood.

Implementation of the existing recommender system is performed using Viola-Jonse algorithm , Point detection algorithm , Support vector machine algorithm , Principal Component Analysis (PCA) and other techniques. The existing system uses these algorithms and plays a song for the user.

3.2.1 DISADVANTAGES

There are few disadvantages identified in the existing system and are defined below:

- The existing system has a low accuracy in face detection.
- The existing system doesn't produce a list of songs for the user to select from.
- It doesn't suggest a list of movies based on the mood detected.
- It doesn't give a list of books to read based on the mood detected.

3.3 PROPOSED SYSTEM

The proposed system uses OpenCV to capture a picture of the user and then using DeepFace-Facial Attribute Analysis it recognizes the mood of the user. The proposed system can recognise moods like happy,sad, angry, surprise and fear.The proposed system also generates a playlist of songs , suggests list of movies, list of bookssuggestions according to the mood detected. The user can browse the playlist from which a song can be selected to play. The user can also change the song , pause the song and perform other operations.

3.3.1 ADVANTAGES

There are advantages in the proposed system which could overcome the drawbacks of the existing system and are defined below:

- Playlist Generation.
- Better Accuracy.
- Music controls.
- List of movies
- List of books

4. SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

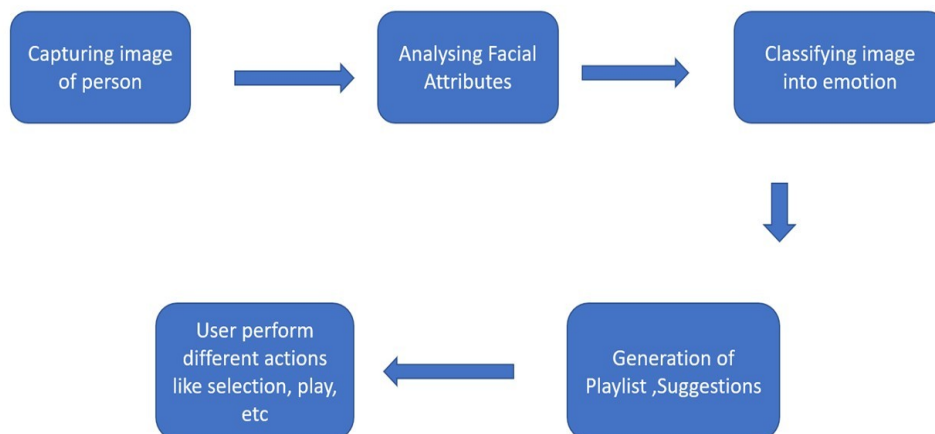
4.1 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 / Linux/
- Back End Language : Python
- Front End languages: Html ,CSS, Bootstrap, Javascript
- Framework : Django
- Packages: Opencv, Pillow, Deep face, Tensorflow

4.2 HARDWARE REQUIREMENTS

- Processor : 2GHZ
- RAM : 4GB
- Hard Disk : Minimum 16GB

4.3 CONTENT DIAGRAM OF PROJECT



The brief overview of the system is shown in the Figure 4.3. An image is captured. Then the picture is used to detect the emotion of the person. Upon clicking desired option, a list is generated for the user to select from. User can perform various operations like select , pause etc.

5.DESIGN

5.1 INTRODUCTION

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of the requirements into data.

UML Diagram:

Actor:

A coherent set of roles that users of use cases play when interacting with the use cases an observable result of value of an actor.

Use case:

A description of sequence of actions, including variants, that a system performs yields an observable result of value of an actor. actor diagram is drawn in a eclipse shape.

Purpose of Use Case Diagram:

Use case diagrams are typically developed in the early stage of development and are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts.

Usage of Use Case Diagrams:

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

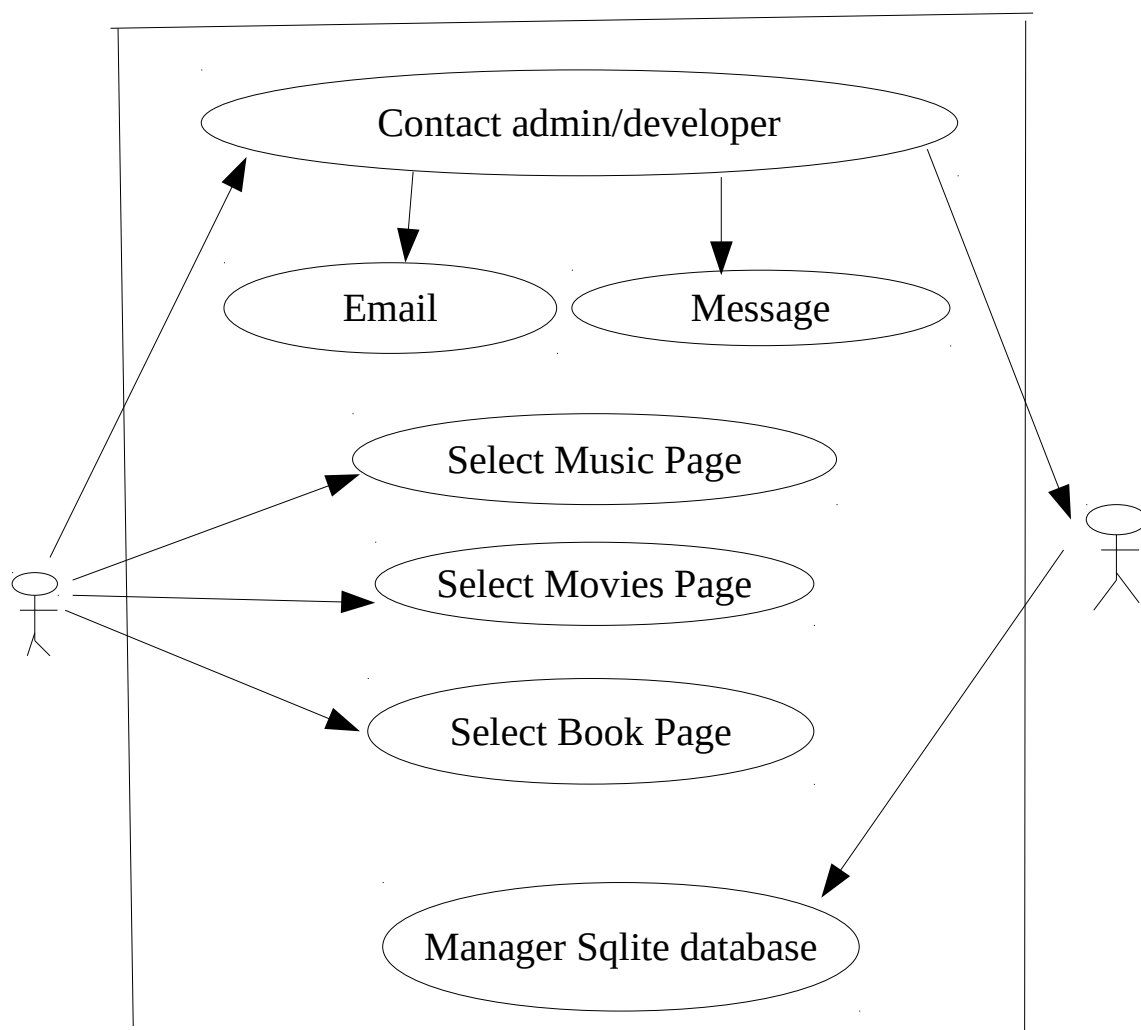
These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering.

Use case diagram is used differently to make it suitable for reverse engineering. In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for –

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.



Emotion Based Recommendation System

ACTIVITY DIAGRAM

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

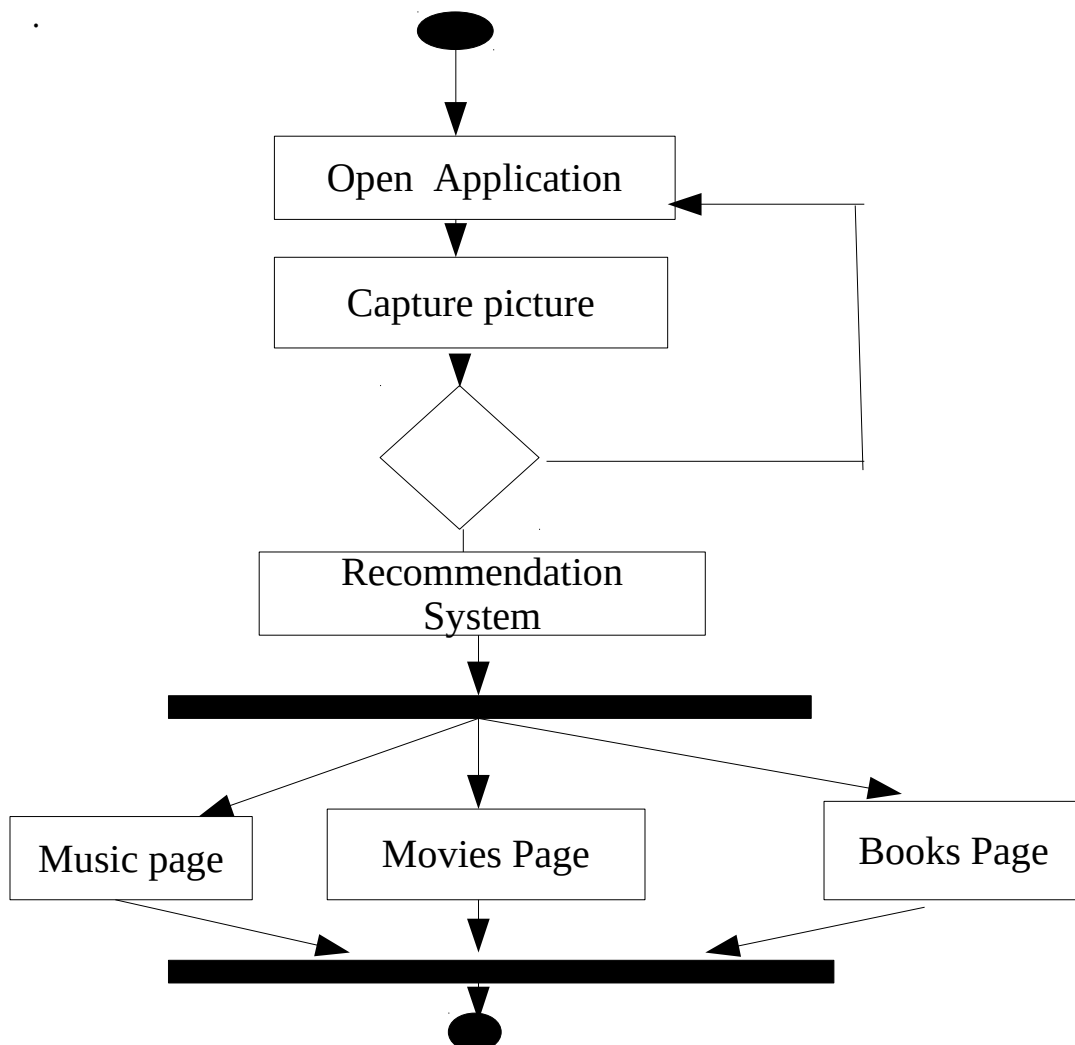


Figure 4.2 : Activity Diagram for system overview

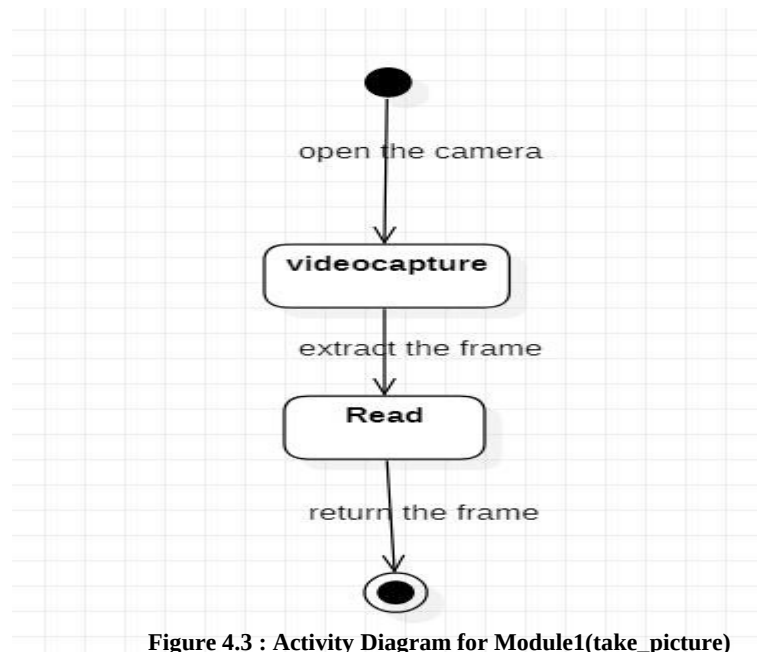


Figure 4.3 : Activity Diagram for Module1(take_picture)

The Activity Diagram for the Module1- take_picture is presented in the Figure 4.3. When the user clicks on Get image, the system opens the camera. Videocapture is used to extract the frame. Read returns the single frame from live camera.

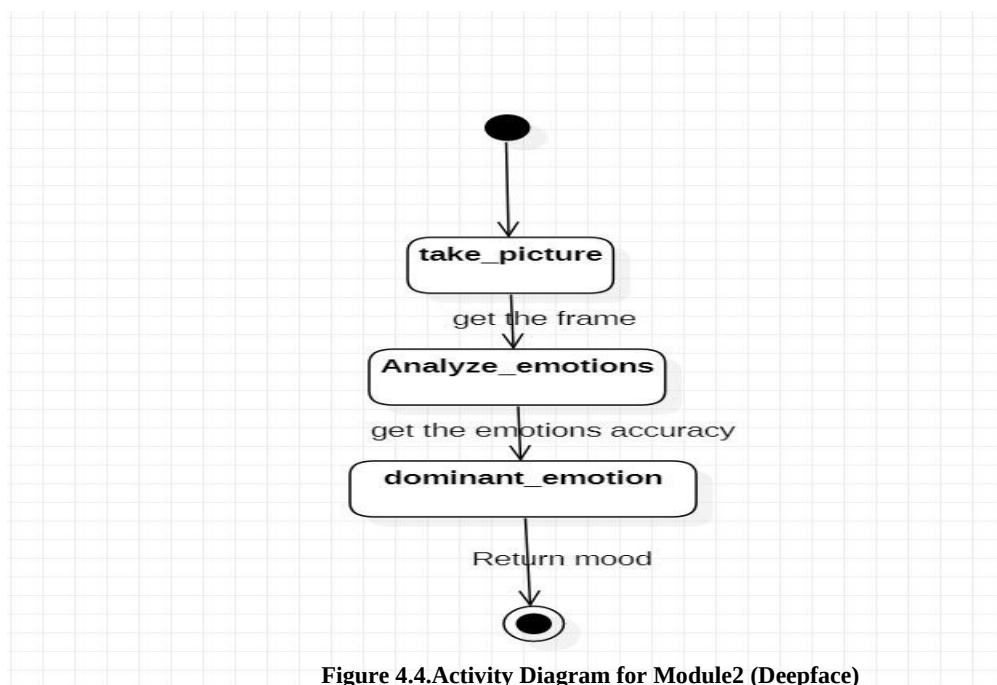
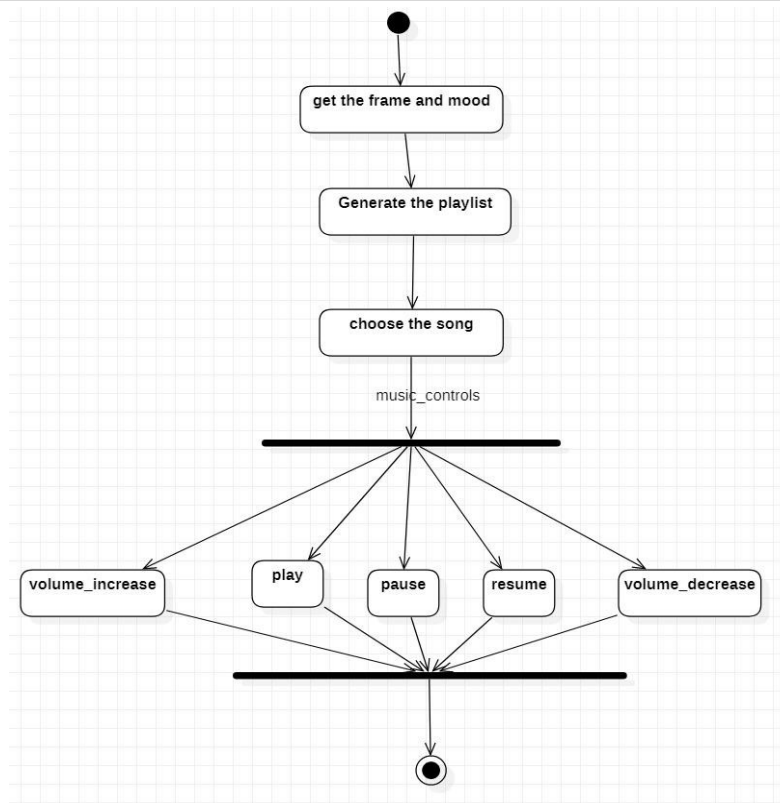


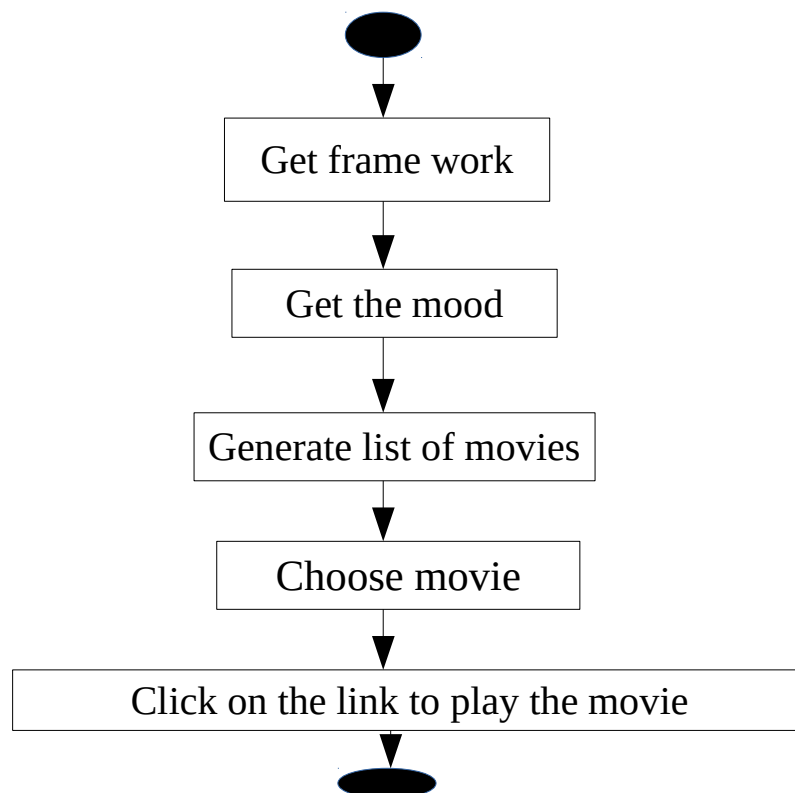
Figure 4.4. Activity Diagram for Module2 (Deepface)

The Activity Diagram for the Module2- Deepface is presented in the Figure 4.4. Here, take_picture is used to get the frame. The emotion from the frame is analyzed using deepface. The dominant emotion is selected and returned.

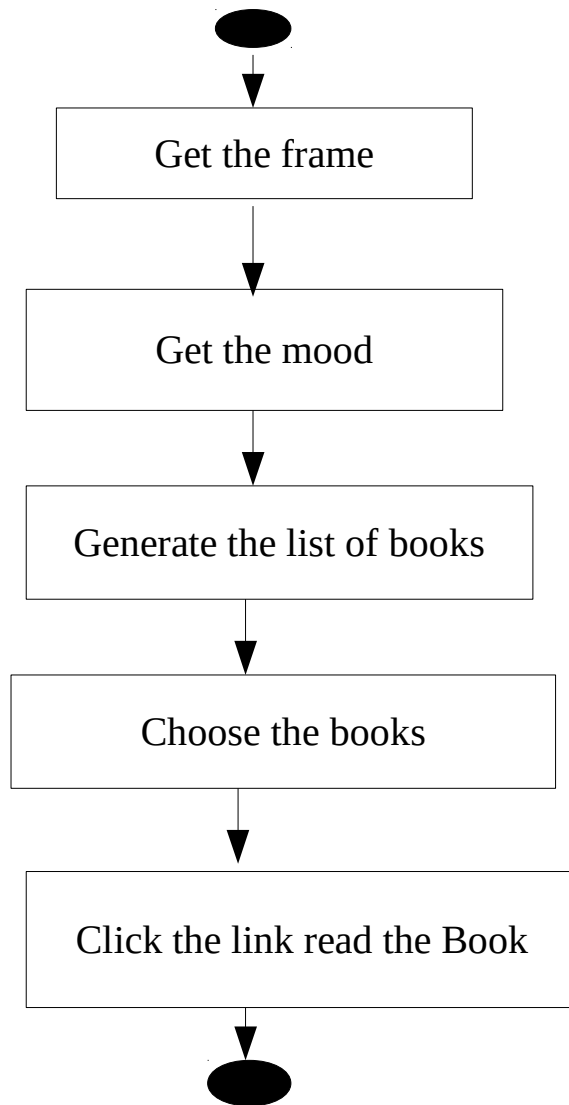


Activity Diagram for Module-Music Page

The Activity Diagram for the Module-Music Page is presented in the Figure 4.5. After detecting the emotion, when the user selects Music in the menu, a playlist is generated. The user selects a song to play. Music operations can also be performed.



Activity Diagram for Module-Book Page



Activity Diagram for Module-Book Page

CLASS DIAGRAM

Purpose of Class Diagram

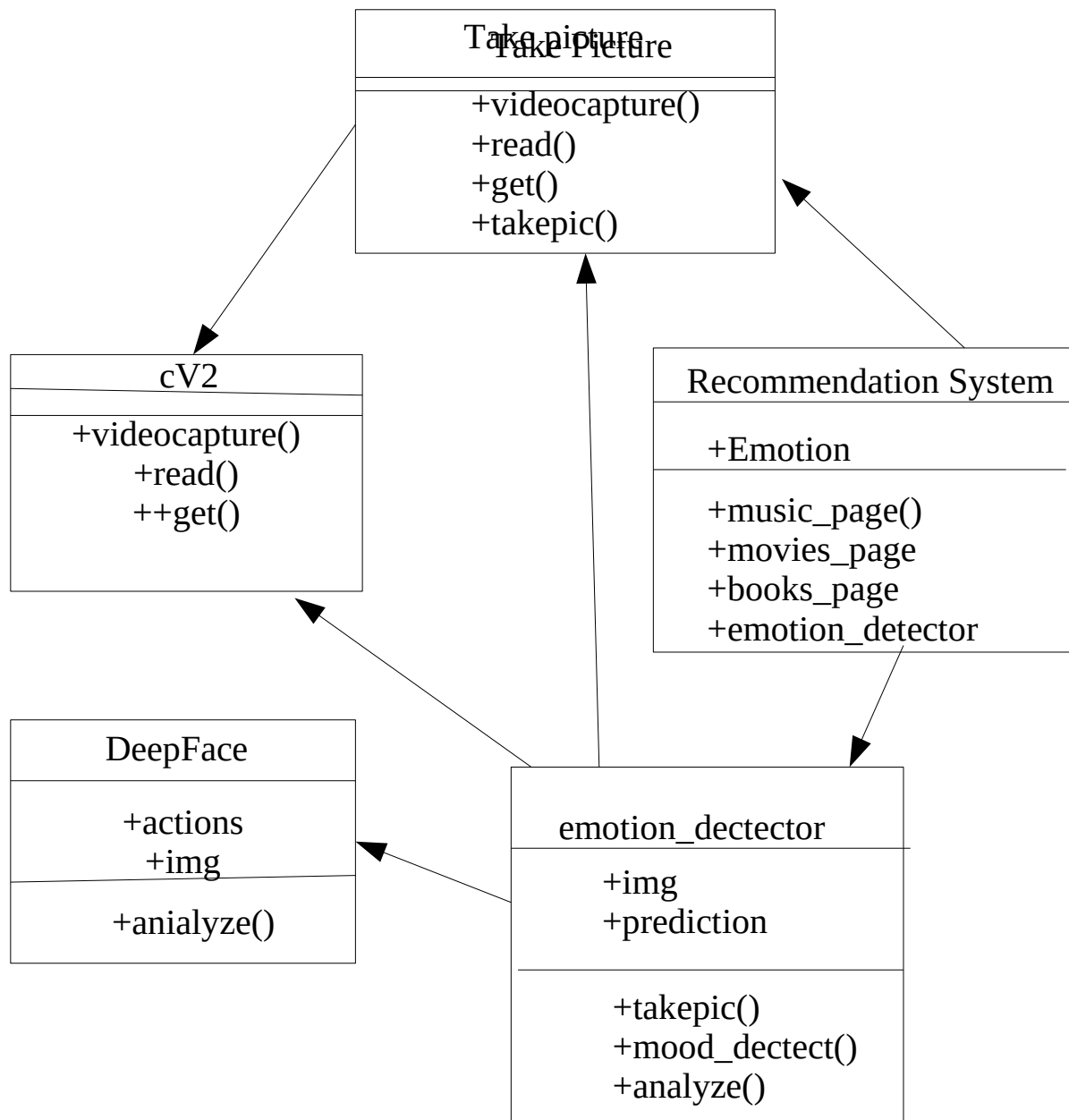
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application; however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as

- This is the only UML which can appropriately depict various aspects of OOPs concept.
- Analysis and design of the static view of an application.

- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



Class Diagram

In Figure 4.8, the Class diagrams gives us a static view of the system. The videocapture() function is used to extract the frames. read() function is used to get the single frame from the live video. get() function is used to return the frame. The analyze() function analyses the frame returned by the Take_picture module to detect the emotion and returns it to the recommendation system. The system in return displays a menu containing three pages, Music page, Movies Page and Book page. When the user selects a certain option, the respective page is displayed.

6. IMPLEMENTATION

6.1 INTRODUCTION

Functions are used for placing or storing the code which is to be repeated several times. For example, if we need same code, then we must have to write that code again and again. So in order to remove this we use functions. Implementation is the stage where the theoretical design is turned into a working system.

The most crucial stage in achieving a new successful system is giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on Implementation, design of methods to achieve the change over an evaluation of change over methods apart from planning. Two major tasks for preparing the implementation are education and training of the users and testing of the system.

6.2 METHOD OF IMPLEMENTATION

The more complex the system being implemented, the more involved will be the systems analysis and design efforts required for implementation. The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then change over to his new fully tested system and the old system is discontinued.

The technologies that are used in the project are:

- Machine Learning
- Python
- Open cv
- Django
- HTML & CSS
- Javascript
- Deep face
- Tensorflow
- SQLite

MACHINE LEARNING:

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. Machine Learning is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Working With Machine Learning In This Project:

Classification algorithm is being used here. In this, OpenCV is used for capturing the image of the user. The captured image is taken as the input and the emotion of the user is detected using DeepFace model.

PYTHON:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

OpenCV:

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process Images and videos to identify objects, faces, or even the handwriting of a human.

Working With OpenCV:

OpenCV provides the “videocapture” class which is used to work with the camera. It is used to do the following tasks:

- Read video, display video, and save the Video.
- Capture from the Camera and display it.
- ❖ We define a function called “takepic” which extracts the frame and returns this to the calling function.
- ❖ Create an object of VideoCapture class to capture a video. It accepts either a device index or the name of a video file.
- ❖ A number which is specifying to the Camera is called device index. We can select a Camera bypassing the 0 or 1 as an argument. After that, we can capture the video frame-by-frame.

TensorFlow:

The first in the list of python libraries for data science is TensorFlow. TensorFlow is a library for high-performance numerical computations with around 35,000 comments and a vibrant community of around 1,500 contributors. It is used across various scientific fields. TensorFlow is basically a framework for defining and running computations that involve tensors, which are partially defined computational objects that eventually produce a value.

Working With TensorFlow- Keras:

Keras is compact, easy to learn, high-level Python library run on top of TensorFlow framework. It is made with focus of understanding deep learning techniques, such as creating layers for neural networks maintaining the concepts of shapes and mathematical details

DeepFace:

DeepFace is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python. It is a hybrid face recognition framework wrapping state-of-the-art models: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace and Dlib. The library is mainly based on Keras and TensorFlow.

Deepface also offers facial attribute analysis including age, gender, facial expression (including angry, fear, neutral, sad, disgust, happy and surprise) and race (including asian, white, middle eastern, indian, latino and black) predictions. Analysis function under the DeepFace interface is used to find demography of a face.

Working With DeepFace:

In this, we define a function called mood_detect. In this function, we call another function called “takepic” from module1 (take picture). The frame extracted is stored in a temporary variable. A function called “analyze” is used to predict the emotion. We pass two arguments to this function. They are ‘image’ and ‘actions’. We take the emotion with highest accuracy and return it to the calling function.

Generating The List:

Once the emotion is detected, the path of the current working directory is stored in a temporary variable. The detected emotion is given is appended to the path. This is used to get the folder containing songs or movies or books, based on the emotion detected. Using that a list is generated.

Django:

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

- Ridiculously fast: Django was designed to help developers take applications from concept to completion as quickly as possible.
- Reassuringly secure: Django takes security seriously and helps developers avoid many common security mistakes.
- Exceedingly scalable: Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

Working With Django:

We develop an emotion based recommendation system application using Django framework. The Django framework is used for back end development. It also has SQLite which is used as a database. SQLite is used to store the data and perform CRUD operations. We use HTML, CSS and Javascript for the front end design with Django Framework.

PSEUDO CODE:

while window is not closed

if capture button is clicked
 ▪ Call mood detect function from module2(dp) and store the image, emotion in variables.
 ▪ Convert the image into PNG format and display it to the front end using HTML.

if face detected

 Display the emotion.

if emotion is not null

- Display three different options- Music , Movies and Books.
- Select one of the three options from the menu.

if Music option is selected

- Display the list of songs.
- Music Controls are allowed.

elif Movies option is selected

- Display the list of movies
- Select the movie by clicking on the link to watch the movie.

else

- Display the list of books
- Click on the desired option to view the book.

else

 Display “Face Not Detected”.

When the application is opened, we get an index page with generate button. When the button is clicked , the request is sent to respective view class in in “views.py”. In “views.py”, the mood_detect function is imported from "dp.py" module. This function detects the emotion of the person and returns to the calling program. We use this emotion to generate list of music, movies and books. The music class in "views.py" generates the playlist from the SQLite database. The movies class in "views.py" generates list of movies for the detected emotion from the SQLite database. In the same way the Books class also generates list of books from the database. The admin can add songs, movies and books using admin interface provided by Django framework.

SYSTEM TESTING

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing
- 2 .Integration testing

Unit Testing

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses . A procedure to call the functions of the unit under test with appropriate parameters.

Integration Testing

In the Integration testing we test various combination of the project module by providing the Input. The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other modu

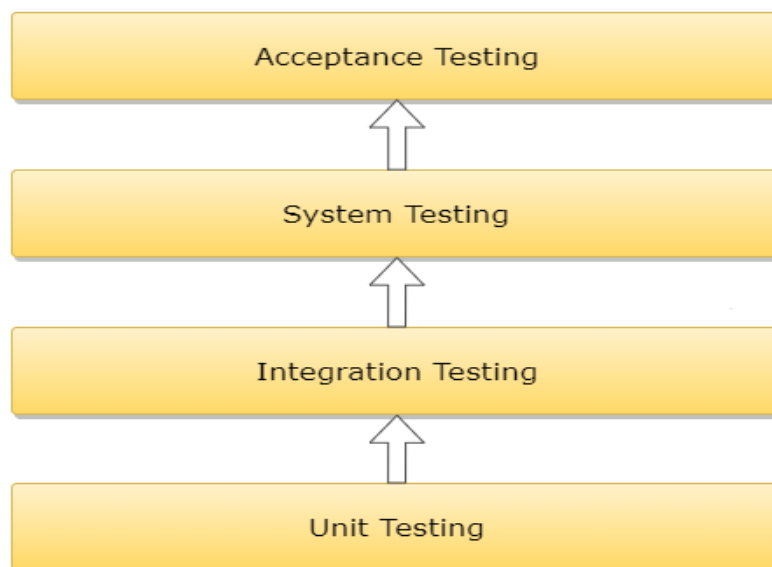


Figure 6.1 Levels Of Testing

S.NO	Description	Detected Stage	Result	Remarks
1	Click on capture Button and take photo of user using OpenCV	Unit Testing	Exception to show face properly.	Result dissatisfied
2	Click on capture button and take photo of user using OpenCV	Unit Testing	Face shown correctly.	Result satisfied
3	Detecting mood of the person using DeepFace model	Unit Testing	Detects mood of person with good accuracy and displays to the user	Result satisfied
4	Click the music option on menu	Unit Testing	Displays a list of songs for the user to select from.	Result satisfied
5	Click on Play song	Unit Testing	Shows exception of empty playlist	Result dissatisfied
6	Click on Play song	Unit Testing	Loads and plays any one Song from playlist of detected mood.	Result satisfied
7	Click on Resume song	Unit Testing	Resumes the current playing song.	Result satisfied
8	Click on volume up and down	Unit Testing	Increases and decreases volume by one level.	Result satisfied
9	Click on Pause	Unit Testing	Pauses the current playing song.	Result satisfied
10	Click on the Movies option on menu	Unit Testing	Displays a list of movies for the user to select.	Result satisfied
11	Click on the Books option on men	Unit Testing	Displays a list of books for the user to select.	Result satisfied

SOURCE CODE

Views.py

```
from django.http import *
from django.shortcuts import render
from .dp import mood_detect
from recommendationapp.models import Song,Movie,Book,Error,Contact
from django.core.paginator import Paginator
# Create your views here.
from django.shortcuts import render
global context
def index(request):
    global context
    return render(request,'index.html')
def about(request):
    return render(request,'about.html')
def contact(request):
    return render(request,'contact.html')
def cssdemo(request):
    return render(request,"cssdemo.html")
def movies(request):
    global context
    context={}
    if "movobj" not in context:
        emotion=mood_detect()
        movobj1=Movie.objects.all()
        movobj=[]
        for i in movobj1:
            if i.mood==emotion:
                movobj.append(i)
        #context["movobj"]=movobj
        context={"emo":emotion,"movobj":movobj}
    return render(request,"movies.html",context)
def books(request):
    global context
    context={}
    if "bokobj" not in context:
        emotion=mood_detect()
        bokobj1=Book.objects.all()
        bokobj=[]
        for i in bokobj1:
            if i.mood==emotion:
                bokobj.append(i)
        #context["movobj"]=movobj
        context={"emo":emotion,"bokobj":bokobj}
    return render(request,"book.html",context)
def music(request):
    global context
    #if request.method=='POST':
    if True:
        emotion=mood_detect()
        songobj1=Song.objects.all()
        songobj=[]
        for i in songobj1:
            if i.mood==emotion:
                songobj.append(i)
        context={"emo":emotion,"songobj":songobj}
    return render(request,"music.html",context)
```

```

def pdf(request):
    global context
    dict=request.POST
    bks=Book.objects.all()
    for i in bks:
        #print(i.name,dict)
        if i.name in dict:
            path1=i.book_file
path1="C:/Users/praveenraj/Documents/emotion_based_recommendation_system_using_django/
EMOTION_RECOMMENDATIO_SYS/media/"+str(path1)
        #print(path1,"hi")
        break
    #print(bks)
#pdf1=open(r"C:\Users\praveenraj\Documents
emotion_based_recommendation_system_using_django\EMOTION_RECOMMENDATIO_SYS\
media\media\books_dir\praveenresume.pdf", "rb").read()
    pdf1=open(path1,"rb").read()
    return HttpResponse(pdf1,content_type="application/pdf")

def Error1(request):
    message=request.POST
    try:
        msg_obj=Error.objects.all()
        msg_len=len(msg_obj)
        print(message)
        msg=Error(Errorid=msg_len+1,Error_desc=message["E_msg"])
        msg.save()
    except Exception:
        msg=Error(Errorid=1,Error_desc=message)
        msg.save()
        print(message)
    #code to store messages in Error table
    return render(request,"index.html")

def contact(request):
    m=request.POST
    mail=Contact(name=m["name"],email=m["email"],phone_number=m["phone"],message=m[
"message"])
    mail.save()
    return render(request,"contact.html")
=====

```

urls.py

""""EMOTION_RECOMMENDATIO_SYS URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
<https://docs.djangoproject.com/en/3.1/topics/http/urls/>
Examples:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path("", views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path("", Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

""""


```

from django.urls import path,include
from django.conf.urls.static import static
from recommendationapp import views
urlpatterns = [
    path("", views.index ,name='index'),
    path('about', views.about ,name='about'),
    path('contact', views.contact,name='contact'),
    path('movies',views.movies,name='movies'),
    path('music',views.music,name="music"),
    path('cssdemo', views.cssdemo,name='cssdemo'),
    path('book',views.books,name='book'),
    path('pdf',views.pdf,name='pdf'),
    path('Error1',views.Error1,name="Error1")
    #path('recommendationapp/', include('recommendationapp.urls')),
]

```

models.py

```

from django.db import models
# Create your models here.
class Song(models.Model):
    title= models.TextField(max_length=100)
    artist= models.TextField(max_length=100)
    image= models.ImageField(upload_to='media/songs_images')
    audio_file = models.FileField(upload_to='media/songs_dir')
    audio_link = models.CharField(max_length=200,blank=True,null=True)
    mood=models.TextField(max_length=10)
    def __str__(self):
        return self.title
class Movie(models.Model):
    name=models.TextField(max_length=150)
    artists=models.TextField(max_length=400)
    image=models.ImageField(upload_to="media/movie_images")
    movie_link=models.URLField(max_length=2000)
    mood=models.TextField(max_length=10)
    def __str__(self):
        return self.name
class Book(models.Model):
    name=models.TextField(max_length=150)
    author=models.TextField(max_length=400)
    image=models.ImageField(upload_to="media/book_images")
    book_link=models.URLField(max_length=2000)
    book_file = models.FileField(upload_to='media/books_dir')
    mood=models.TextField(max_length=10)
    def __str__(self):
        return self.name
class Error(models.Model):
    Errorid=models.TextField(max_length=30,unique=True)
    Error_desc=models.TextField(max_length=500)
    def __str__(self):
        return self.Errorid
class Contact(models.Model):
    name=models.TextField(max_length=100)
    email=models.EmailField(max_length=300,unique=True)
    phone_number=models.IntegerField()
    message=models.TextField(max_length=5000)
    def __str__(self):
        return self.email

```

dp.py

```
from deepface import DeepFace
import cv2
def takepic():
    cap = cv2.VideoCapture(0)
    #width=cap.get(cv2.CAP_PROP_FRAME_WIDTH)
    #height=cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
    _, frame = cap.read()
    return frame
def mood_detect():
    img=takepic()
    #img=cv2.imread(r"C:\Users\praveenraj\Downloads\livetest4.jpg")
    cv2.imwrite(r"recommendationapp\static\recommendationapp\personpic.jpg",img)
    try:
        predictions=DeepFace.analyze(img,actions=['emotion'])
        predictions=predictions['emotion']
        del predictions['neutral']
        del predictions['disgust']
        print(predictions)
        temp1=0
        for i in predictions:
            if predictions[i]>temp1:
                mod1=i
                temp1=predictions[i]
        #print(mod1)
    except Exception:
        mod1=None
    #print(mod1,w,h)
    return mod1
```

Index Page

```
<!doctype html>
<html lang="en">
{% load static %}
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbZKgwra6"
crossorigin="anonymous">
<link rel="stylesheet" href="{% static 'recommendationapp/assets/css/chat.css'%}">
<link rel="stylesheet" href="{% static 'recommendationapp/ch_win_style.css'%}">
<link rel="stylesheet" href="{% static 'recommendationapp/assets/css/typing.css'%}">
<title>Hello, world!</title>
</head>
<body style="background-color: rgba(187, 177, 177, 0.63);position: relative;">
{% load static %}
<nav class="navbar navbar-expand-lg bg-dark navbar-dark">
<div class="container-fluid">
<a class="navbar-brand" href="\ ">PSK Developers</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
```

```

target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="/">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/about">About</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/contact">Contact</a>
</li>
</ul>
</div>
</div>
</nav>
<div class="container" style="margin-top: 60px;margin-bottom: 60px;">
<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
<div class="carousel-indicators">
<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-
label="Slide 2"></button>
<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-
label="Slide 3"></button>
</div>
<div class="carousel-inner">
<div class="carousel-item active">

</div>
<div class="carousel-item">

</div>
<div class="carousel-item">

</div>
</div>
<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators"
data-bs-slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators"
data-bs-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="visually-hidden">Next</span>
</button>
</div>
</div>
<!--<div class="label">{{emo}}</div>-->
<div style="text-align: center;
background-color: orange ;
width:100%;height: auto;">

```

```

<h1 style="font-weight:bold ;margin-top: 50px;margin-bottom:0px;">Click here to Generate Playlist</h1>
<hr style="width: 100%;color:black; height: 3px;size: 4;" >
<form action="\music" method="POST">
  {% csrf_token %}
  <button type="submit" style="margin-bottom:60px;
  background-image: url('/static/recommendationapp/facedetectbg.png') ;
  width: 400px;height:400px;

background-size:contain;background-repeat: no-repeat;
border-radius: 5%;">
</button>
</form >
</div>
<div class="container">
<div class="chatbox">
<div class="chatbox__support">
<div class="chatbox__header">
<div class="chatbox__image--header">

</div>
<div class="chatbox__content--header">
<h4 class="chatbox__heading--header">Error reporting</h4>
</div>
</div>
<div class="chatbox__footer">
<form action="\Error1" method="POST">
  {% csrf_token %}
  <!---->
  <!---->
  <input type="text" name="E_msg" placeholder="Write a message...">
  <input type="submit" class="send" value="Send"/>
  <!-- -->
</form>
</div>
</div>
<div class="chatbox__button">
<button>chat</button>
</div>
<script src="{% static 'recommendationapp/assets/js/Chat.js'%}"></script>
<script src="{% static 'recommendationapp/chat_app.js'%}"></script>
</div>
</div>
<footer>
<!-- Copyright -->
<div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">
  © 2021 Copyright | All Rights Reserved by PSK Developers</h4>
</div>
<!-- Copyright -->
</footer>
<!-- Footer -->
<!-- Optional JavaScript; choose one of the two! -->
<!-- Option 1: Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
crossorigin="anonymous"></script>
<!-- Option 2: Separate Popper and Bootstrap JS -->
<!--
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"
</body>
</html>

```

9.RESULTS



Figure 9.1:Home Page

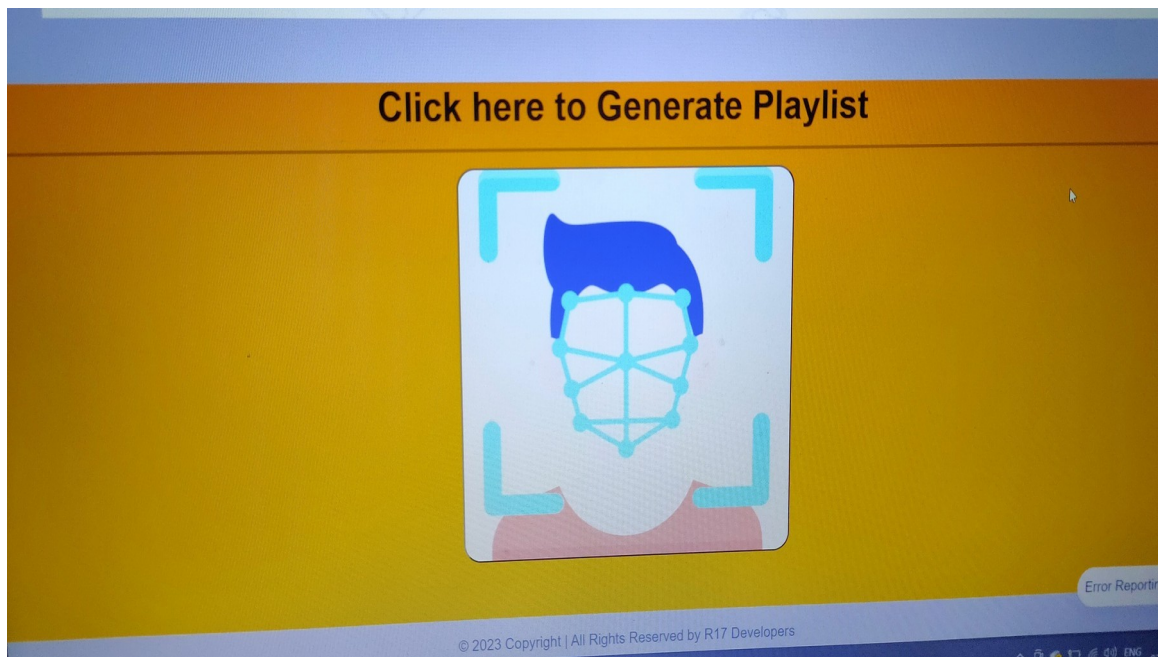


Figure 9.2:Image Capturing Page

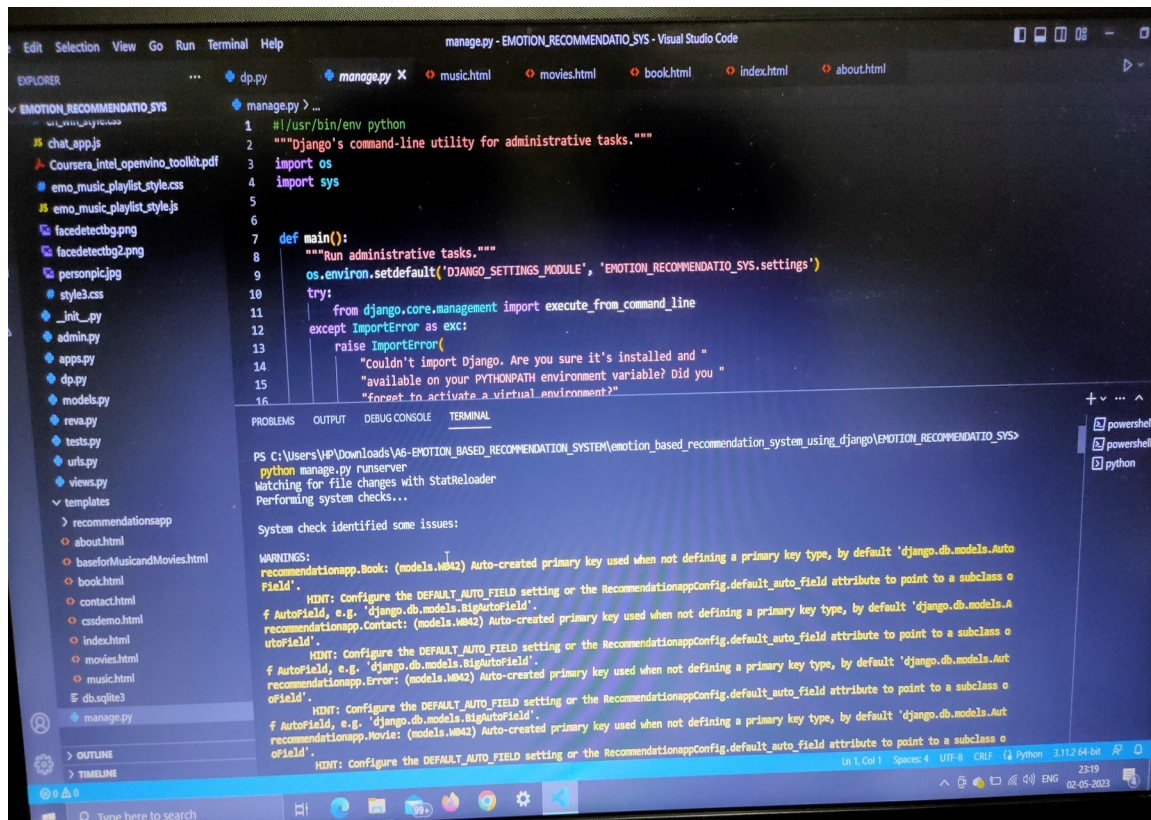


Figure 9.3:Terminal Page

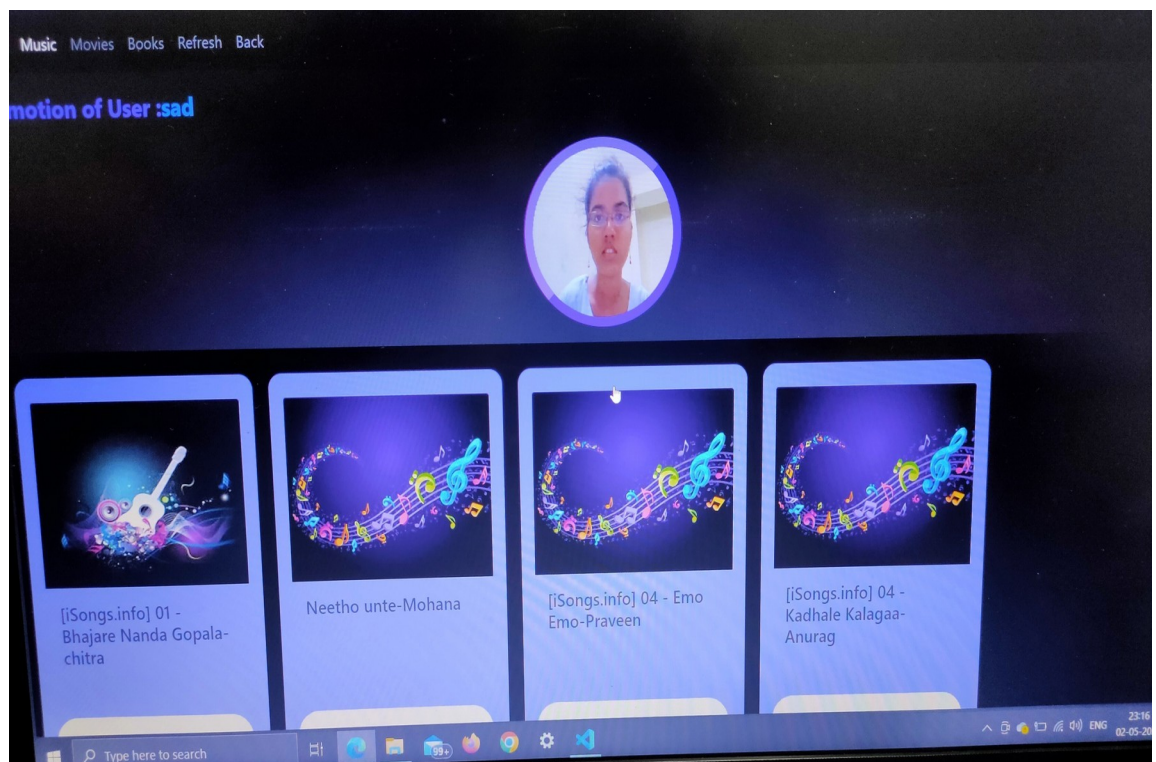


Figure 9.4:Music Page

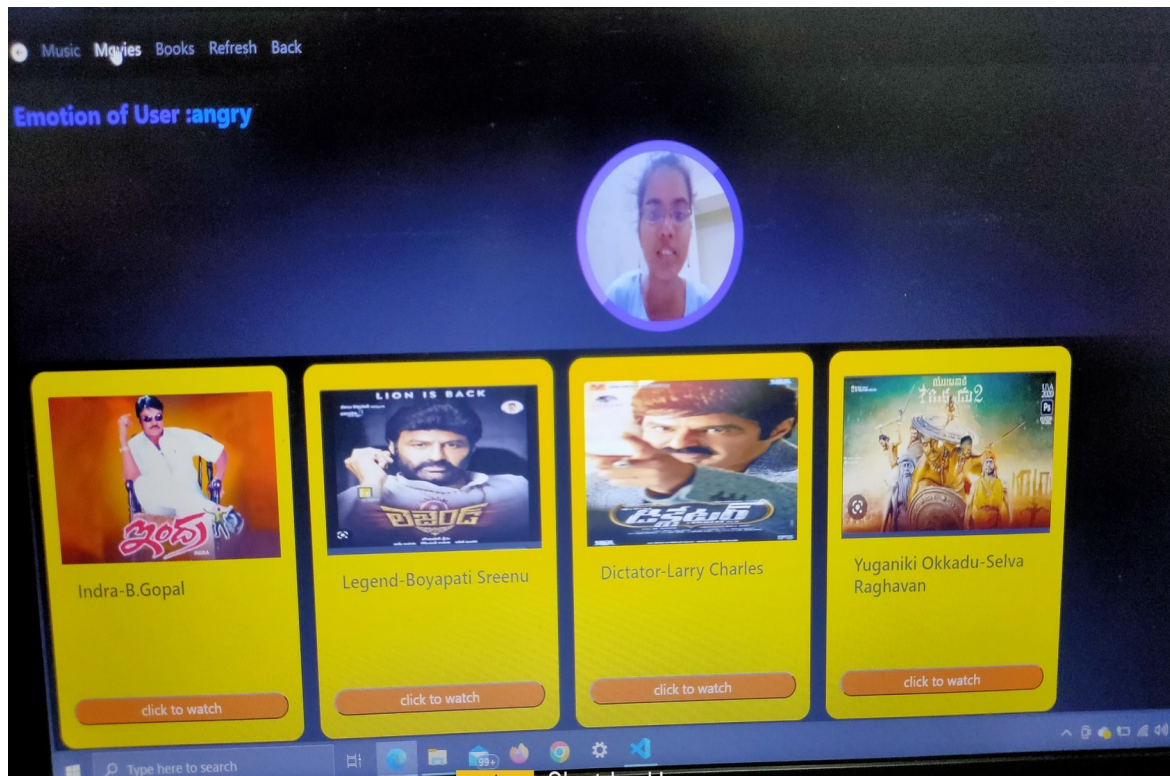


Figure 9.5:Movie Page

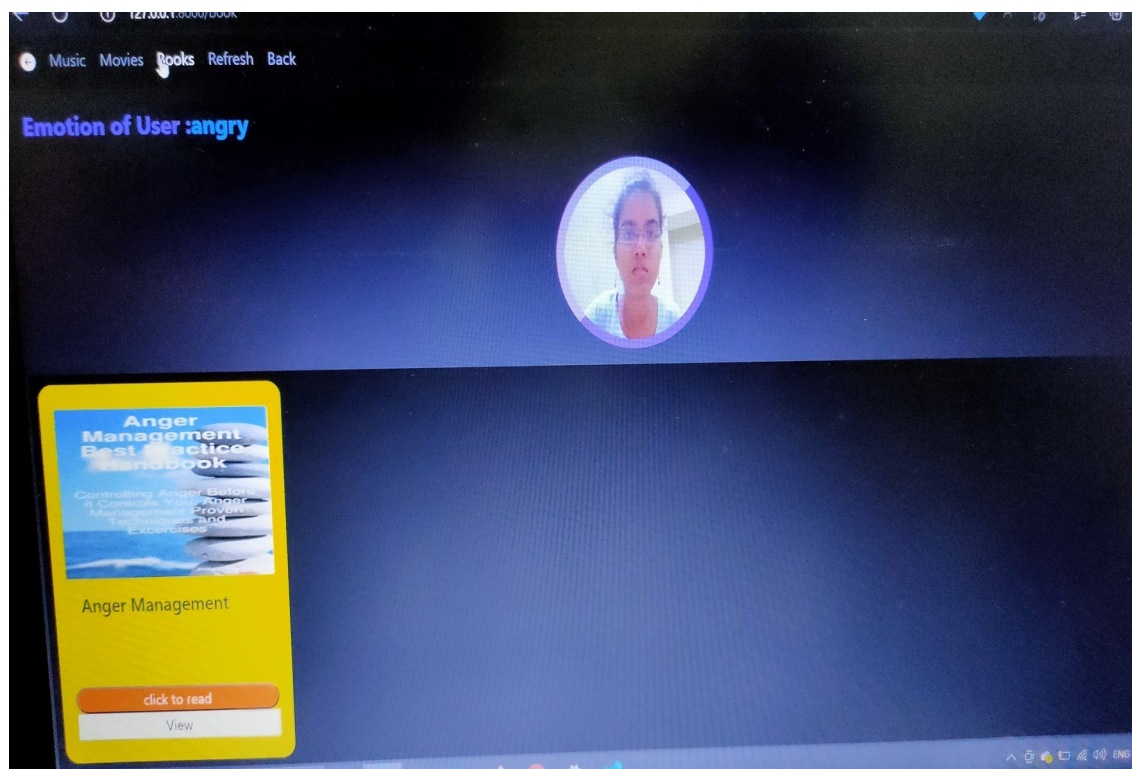


Figure9.6:Books Page

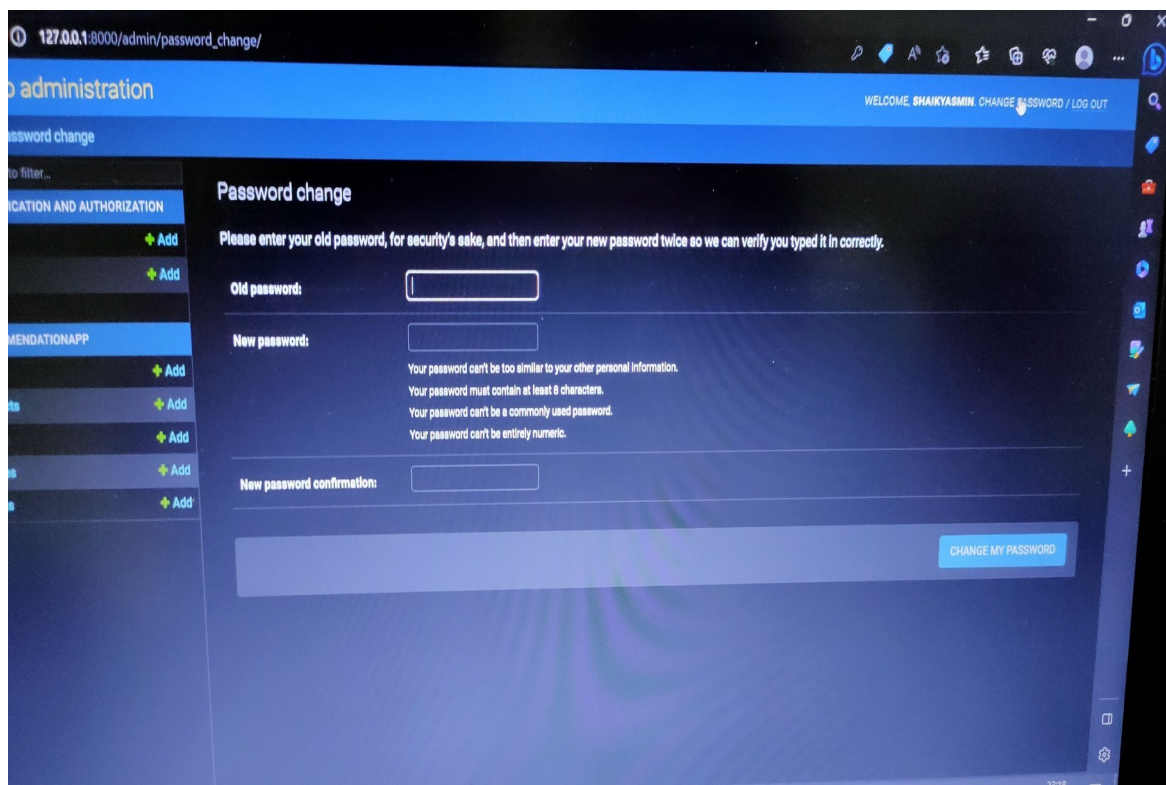
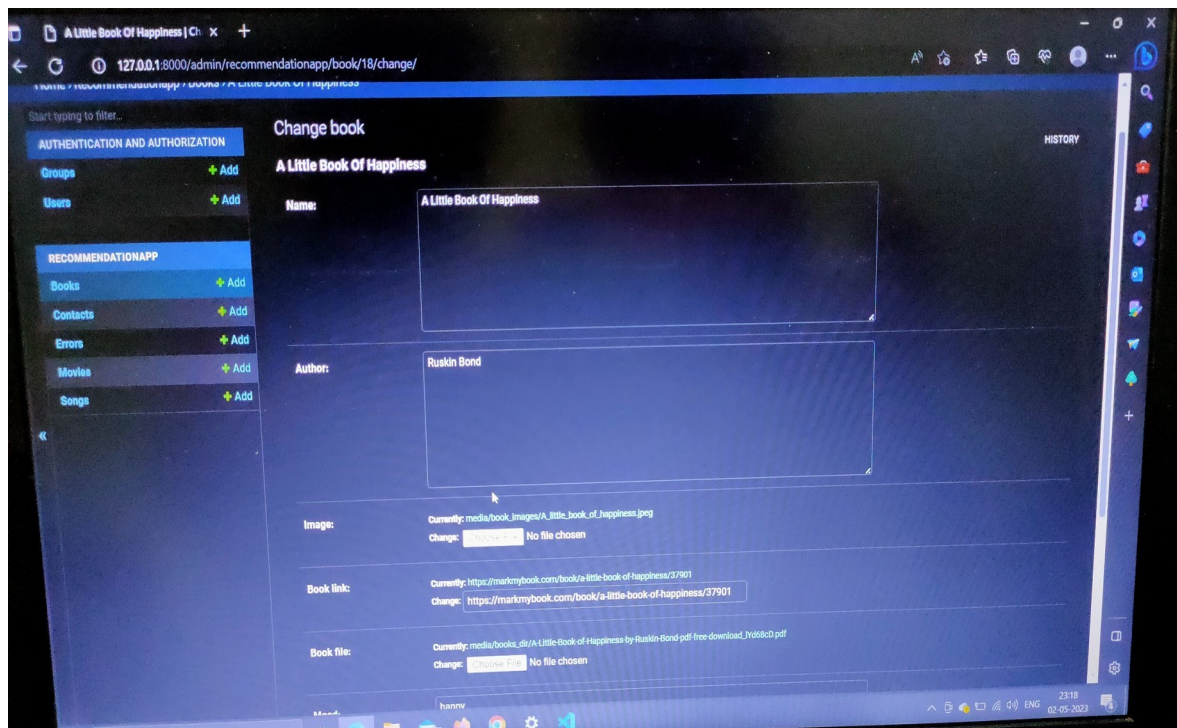


Figure 9.7:Admin Pages

10. CONCLUSION

The Emotion-Based Recommendation System is used to automate and give a better experience for the end user. The application solves the basic needs of music listeners , movie watchers and bibliophiles 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 allows the user to select an option from the menu. The options being Music, Movies and Books. Upon selectin g one of the three options, a list is generated based on the emotion detected. The user can select any song or movie or book from the list generated respectively.

10.1 FUTURE ENHANCEMENT:

In future we can enhance following functionalities:

- It keeps track record of the songs being played by the users. Keeping a track record helps to determine the songs being played the most.
- The background will be changed as per the mood detected.
- It allows the user to edit the playlists.
- It allows used to watch short clips instead of a whole length movie.
- The book section could be further expanded into subsections like Novels, Short stories etc.
- It allows the user to give voice based commands to run the application.

11.REFERENCES

11.1 TEXT BOOKS:

- Machine Learning with Python: Design and Develop Machine Learning and Deep Learning Technique using real world code examples, Abhishek Vijayvargia, 1 st Edition, 2019.
- Django for Professionals: Production Websites with Python & Django, William S. Vincent, Published 2019.

11.2 WEBSITES:

- www.python.org
- www.stackoverflow.com
- www.tutorialspoint.com
- www.geeksforgeeks.org
- www.javatpoint.com
- ieeexplore.ieee.org
- www.djangoproject.com
- en.wikipedia.org/wiki/Machine_learning
- www.pypi.org/project/deepface/