EMOTION DETECTION AND MUSIC PLAYER SYSTEM

MOODMELODY!

FINAL DOCUMENT

BY

SOFTWARE SPRINTERS

PRESENTED TO – DR. HUBERT A. JOHNSON

CSIT 515 – 02 SOFTWARE ENGINEERING & RELIABILTY

SPRING 2024

Team members-

LAXMI KEERTHI REDDY GANUTHALA

LOKESHWAR REDDY VARRA

SHIVA REDDY ONTELA

VAISHNAVI LOLLA

TEAM MEMBERS

NAME	CONTACT NO.	MAIL
Laxmi Keerthi Reddy Ganuthala	(201) 892 - 7352	ganuthalal1@montclair.edu
Lokeshwar Reddy Varra	(346) 575 - 8454	varral1@montclair.edu
Shiva Reddy Ontela	(203) 706 - 5367	ontelas1@montclair.edu
Vaishnavi Lolla	(201) 423 - 0395	lollav1@montclair.edu

ACKNOWLEDGEMENT

We are pleased to present "Mood Melody: Emotion Recognition and Music player Sytem" project and take this opportunity to express our profound gratitude to all those people who helped us in the completion of this project.

We express our deepest gratitude towards our Prof. Hubert A. Johnson who gave a valuable and timely advice during the various phases in our project. We would also like to thank him for providing us with all proper facilities and support. We would like to thank him for support, patience, and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.

ABSTRACT

This project aims to develop an integrated system combining facial recognition technology with a music player to deliver a personalized music experience based on detected mood. Utilizing Convolutional Neural Networks (CNN) for image processing, along with TensorFlow and OpenCV, the system identifies users and assesses their emotional state to play music that matches their mood. The system is built on a Flask framework with a MySQL database backend to handle data operations, including user registration, login, and music management. This innovative approach leverages machine learning to enhance user interaction with digital media, aiming to provide a responsive and intuitive user experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVATIONS

S.no	Abbreviation	Full Form	Page No.
1	CNN	Convolutional Neural Network	1
2	HTML	Hyper Text Markup Language	4
3	CSS	Cascading Style Sheets	4
4	OpenCV	Open-Source Computer Vision Library	4
5	IDE	Integrated Development Environment	10
6	GDPR	General Data Protection Regulation	11
7	CCPA	California Consumer Privacy Act.	11
8	UI	User Interface	22
9	RDBMS	Relational Database Management System	40
10	API	Application Programming Interface	53

1. INTRODUCTION

1.1 Introduction of the project

Today's technology focuses heavily on personalization, adapting to each user's preferences and emotional states. Our project taps into this trend by developing a web-based application that utilizes facial recognition and machine learning to revolutionize the way users interact with music players. This system is capable of not only recognizing individual users through their webcam but also determining their current mood. Using advanced image processing libraries such as TensorFlow and OpenCV, the system employs a Convolutional Neural Network (CNN) to analyze facial expressions and ascertain emotions like happiness or sadness. Based on this mood detection, the system selects and plays music from a predefined library that suits the user’s emotional state. This functionality is supported by Pandas for efficient data management, ensuring songs and user data are handled securely. The entire application runs within a Flask framework, offering a responsive web interface for user interactions like registration and music playback. Hosted on an Anaconda server, the environment ensures all components work seamlessly together, managed through Visual Studio Code, which provides a robust platform for coding, testing, and deploying our application. This integration of facial recognition with a mood-responsive music player aims to create a deeply personalized and engaging user experience.[1]

1.2 Objective

The primary objective of this project is to create a seamless integration of facial recognition and mood detection technologies with a music playback system that personalizes the user experience based on emotional inference. Key goals include:

- To implement a reliable facial recognition system that accurately identifies users.
- To develop a mood detection algorithm that categorizes a user’s current emotional state.

- To build a responsive music player that selects and plays music tracks corresponding to the detected mood.
- To ensure secure and efficient user management through registration and login functionalities, with age-based access restrictions.

1.3 Problem Statement

While many music players and streaming services offer personalized playlists, few provide real-time music selection based on the user's current emotional state detected via facial expressions. This gap in service delivery can be filled by integrating mood-detection technology with facial recognition, offering a more tailored and responsive user experience. However, challenges such as ensuring the accuracy of facial and mood recognition algorithms, handling privacy and data security, and creating an intuitive user interface need to be addressed. Additionally, the system must be designed to be scalable and efficient, particularly in managing database interactions and user session data in real-time.

By addressing these issues, the project aims to enhance the interactivity of digital media platforms and set a benchmark for adaptive technology in personal entertainment.

1.4 Purpose & Intended Audience

The primary purpose of this project is to create an integrated system that uses facial recognition to identify users and their current mood, and then select and play music that corresponds to that mood. This integration aims to enhance the user experience by personalizing music playback based on real-time emotional feedback.

- **Music lovers:** Individuals who enjoy music and are interested in a personalized listening experience.
- **Tech enthusiasts:** Users who are keen on using cutting-edge technology to enhance their daily activities.

- **Mental health proponents:** People looking for therapeutic aids that use music to influence mood and emotional well-being.

1.5 Product Scope

The product will offer a web-based interface that allows users to register, log in, and interact with the music player through facial recognition. The scope includes:

- User authentication and age verification.
- Real-time facial and mood recognition.
- Music playback that is responsive to the user's detected mood.
- Basic controls for music playback, including play and stop.

1.6 Product Features

- **Facial Recognition:** Securely identifies the user from a webcam feed.
- **Mood Detection:** Analyzes the user's facial expressions to determine their current mood using a CNN.
- **Music Selection:** Automatically selects songs from a database that are tagged with mood labels corresponding to the user's detected mood.
- **User Interface:** Simple and intuitive web interface with play and stop controls for music playback.
- **Age Restriction:** Blocks registration for users under the age of 18.

1.7 User Characteristics

- Users are expected to have a basic understanding of web interfaces.
- Users should be comfortable with using webcams for the purpose of facial recognition.
- Users are seeking a highly personalized and responsive music experience.

1.8 System Requirements

1.8.1 Software Requirements

Requirement Type	Name/Technology	Purpose
Programming Languages	Python	Main programming language used for backend development and machine learning.
Web Framework	Flask	Used to create and manage the web server, routes, and web interfaces.
Machine Learning Library	TensorFlow	Utilized for building and running the CNN for facial recognition and mood detection.
Image Processing Library	OpenCV	Handles image processing tasks involved in facial recognition.
Database	MySQL	Stores user data, including registration details, and music library associated with mood tags.
Data Manipulation	Pandas	Used for handling and processing data operations, especially useful for managing datasets within the system.
Development Environment	Anaconda Server	Manages Python packages and environments, providing a robust platform for data science and machine learning tasks.
IDE	Visual Studio Code	Code editor for writing and managing the project’s codebase. Includes support for Python, Flask, and integration with databases.
Web Technologies	HTML, CSS, JavaScript	For building the frontend of the web application, including forms and buttons.

Table 1.1 Software Requirements

1.8.2 Hardware Requirements

Requirement Type	Specification	Purpose
Server	Multi-core CPU (e.g., 4 cores), 8GB RAM, 50GB Storage	To host and run the web server, manage the database, and perform machine learning operations efficiently.
User Device	Webcam	Essential for capturing live facial images for recognition and mood detection.
Network	High-speed Internet Connection	Required for seamless communication between the client (user) and server, particularly important for streaming music and handling web requests.

Table 1.2 Hardware Requirements

1.9 Constraints

- **Privacy Concerns:** Handling and storing facial recognition data securely.
- **Technology Limitations:** Dependence on the accuracy of the CNN model for mood detection.

1.10 Assumptions and Dependencies

1.10.1 Assumptions

- Users have access to devices with webcams and internet connectivity.
- Users are willing to provide real-time facial data for the purpose of mood detection.
- The system assumes that the mood accurately reflects the user's preference for music at that time.

1.10.2 Dependencies

- **Software Dependencies:** Relies on TensorFlow for machine learning, OpenCV for image processing, Flask for web framework, MySQL for database management, and Pandas for data handling.
- **Hardware Dependencies:** Requires a computer with a webcam and adequate processing power to handle real-time image processing and music streaming.

2. LITERATURE SURVEY

1. Facial Recognition and Machine Learning

Facial recognition technology has advanced significantly with the development of deep learning techniques, particularly Convolutional Neural Networks (CNNs). These methods have revolutionized image analysis, offering substantial improvements in accuracy and reliability. A cornerstone paper in this domain is by Krizhevsky et al. (2012), which discusses the advancements in CNNs that significantly improved image classification tasks.

Reference: Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Facial recognition algorithms, specifically applied within TensorFlow and OpenCV frameworks, are widely discussed in the literature. These tools provide robust environments for developing and deploying machine learning models that are capable of not only recognizing individual identities but also interpreting emotional expressions from facial cues.

2. Mood Detection Through Facial Expressions

The detection of mood from facial expressions intersects the fields of computer vision and psychology. Research by Ekman and Rosenberg (1997) provides foundational knowledge on linking facial expressions with emotional states. More recent studies have integrated these principles with machine learning technologies to automate mood detection, enhancing the personalization of digital interfaces.

Reference: Ekman, P., & Rosenberg, E. L. (Eds.). (1997). What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA.

3. Music Recommendation Systems Based on Emotional State

Music recommendation systems have traditionally used collaborative filtering or content-based filtering techniques. However, incorporating emotional states through facial recognition introduces a novel approach to personalization. A key study by Yang and Chen (2012) explores the impact of user emotion, detected through physiological signals, on music recommendation.

Reference: Yang, Y. H., & Chen, H. H. (2012). Machine recognition of music emotion: A review. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3), 40.

4. Integration of Technologies in Web Applications

The integration of TensorFlow, Flask, and MySQL for developing web-based applications is discussed in various sources. The use of Flask as a micro web framework facilitates rapid development and easy integration of Python-based analytics tools, which is critical for real-time applications like your project. Documentation and tutorials from the official Flask and TensorFlow websites provide practical guidance and best practices.

Web Resources:

Flask Documentation: Flask Pallets Project
TensorFlow Tutorials: TensorFlow Official Site

5. Ethical Considerations and User Privacy

The ethical implications of using facial recognition and mood detection technologies are significant, particularly in terms of privacy and consent. Papers discussing the ethics of AI and user data protection offer insights into designing systems that respect user privacy and adhere to regulatory standards.

Reference: Keyes, O., & Dignum, V. (2021). Misgendering Machines: Trans/HCI Implications of Automatic Gender Recognition. Proceedings of the ACM on Human-Computer Interaction, 4(CSCW2), 1-22.

3. SYSTEM ANALYSIS

3.1 Requirement Analysis

Requirement analysis for this study has been done by first identifying key factors that will affect the project and key people who will be affected by the project. Every system needs some requirements for the development process. Without requirements the system cannot be developed. It includes:

3.1.1 Functional Requirements

The functional requirement for a system should describe what the system do or perform. These requirements depend on the type of software being developed, the expected user of the software and the general approach taken by the organization when writing requirements. Functional requirements describe system function in detail with its input, output, and so on. [1][2]

1. **User Registration and Authentication:**

- Users must be able to register by providing necessary details.
- Age verification must be performed during registration to ensure no user under 18 can register.
- Users must be able to log in using credentials set during the registration process.

2. **Facial Recognition:**

- The system should capture a user’s facial image through a webcam.
- The system must identify registered users based on their facial features.

3. **Mood Detection:**

- The system should analyze the captured facial image to determine the user’s mood.
- Mood detection should classify emotions into predefined categories (e.g., happy, sad, neutral).

4. **Music Playback Based on Mood:**

- The system should select music tracks that match the detected mood from a database.
- Users should be able to play and stop music through the interface.

5. **Data Management:**

- The system should securely store user data, including login details and user preferences.
- The system should manage music data, associating tracks with specific moods.

3.1.2 Non-Functional Requirements

Non-functional requirements describe how the system performs certain tasks and under what constraints. Here are key non-functional requirements:

1. **Performance:**

- The system should process facial recognition and mood detection within a few seconds to ensure a responsive user experience.
- Music playback should start immediately after mood detection and selection.

2. **Usability:**

- The user interface should be intuitive and easy to navigate for users with basic digital literacy.
- Instructions and error messages should be clear and helpful.

3. **Reliability:**

- The system should be available for use at least 99% of the time.
- Facial recognition and mood detection should accurately function under varying lighting conditions.

4. **Scalability:**

- The system should support an increase in user numbers and data volume without significant performance degradation.

5. **Maintainability:**

- The system should be easy to update and maintain, with clear documentation provided for system configurations and updates.

6. **Privacy:**

- The system must include features to protect user privacy and comply with legal standards, including mechanisms for users to consent to the collection and use of their data.

3.2 Feasibility Study

A feasibility study is a preliminary assessment of a proposed project's practicality, including its technical, financial, and operational aspects. Here is a feasibility study for a System:

1. **Technical Feasibility:**

- **Development Stack:** The project will utilize Python for backend development, employing Flask framework for web development and TensorFlow for implementing the CNN algorithm for facial recognition. OpenCV will be used for capturing and processing facial images. MySQL will serve as the database management system. Anaconda server will be utilized for managing Python environments. Visual Studio Code will be the Integrated Development Environment (IDE) for coding.
- **Performance Optimization:** The system will be optimized for performance and scalability to handle processing of facial images, mood recognition, and music playback efficiently. Testing will be conducted to ensure compatibility across multiple devices and browsers.
- **Hosting:** The system will be hosted on a reliable and scalable cloud hosting platform to accommodate potential high traffic volumes and ensure uninterrupted service availability.

2. **Economic Feasibility:**

- **Development Cost:** The estimated development cost for the project, including design, development, and testing, is $10000.
- **Ongoing Costs:** The monthly ongoing costs for hosting, maintenance, and support are estimated to be $1000.

3. **Operational Feasibility:**

- **Personnel:** The project will be managed and maintained by a team of trained personnel responsible for updating content, managing user accounts, and ensuring the smooth operation of the system.

- **User-Friendly Interface:** The user interface will be designed with intuitive and user-friendly features to facilitate easy management and usage of the system by non-technical users.

4. **Schedule Feasibility:**

- **Project Timeline:** The estimated timeline for development from start to launch is 5 months. The development team will work closely with stakeholders to minimize delays and ensure adherence to the timeline.

5. **Legal Feasibility:**

- **Data Privacy Compliance:** The system will comply with all relevant data privacy regulations, including GDPR and CCPA. Measures will be implemented to protect user data privacy and security, including encryption of sensitive information and adherence to privacy policies.
- **Age Restriction:** The registration process will include validation to ensure that users under the age of 18 cannot register, complying with legal requirements regarding minors' online activity.

3.3 Deliverables

1. **Web Application:**

A fully functional web-based platform that users can access to register, log in, and use the facial recognition and music playback system.

2. **Registration and Login System:**

A secure user registration and login page that includes age verification to restrict access for users under 18.

3. **Facial Recognition Module:**

A robust system to capture and recognize user faces through a webcam interface.

4. **Mood Detection Algorithm:**

An implemented CNN model that analyzes facial expressions to determine the user’s mood.

5. **Music Playback System:**

A system that selects and plays music based on the mood detected, with basic controls (play and stop).

6. **Database System:**

A MySQL database setup to store user information, facial recognition data, mood data, and music library associations.

7. **Documentation:**

Comprehensive user and technical documentation, including system setup, operation guidelines, and troubleshooting.

8. **Training and Support Materials:**

Training materials for end-users and administrators, and ongoing technical support for system maintenance.

3.4 Ease of Use

1. **User Interface Design:**

The web application will have a simple, intuitive interface designed for ease of use, ensuring that users of all technical abilities can navigate and control music playback effortlessly.

2. **Automated Processes:**

Key features like facial recognition and mood detection are automated, requiring minimal user interaction, which enhances user experience by simplifying operations.

3. **Accessibility Features:**

Considerations for accessibility will be made to accommodate users with disabilities, ensuring the system is usable for a broader audience.

3.5 Limitations

1. **Dependence on Hardware:**

Users must have access to a device with a functional webcam and adequate processing capabilities, which might restrict accessibility for some.

2. Lighting and Environmental Conditions:

The accuracy of facial recognition and mood detection can be affected by poor lighting or other environmental factors, potentially limiting effectiveness in non-ideal conditions.

3. Privacy Concerns:

Handling sensitive data like facial images and mood information raises significant privacy issues, requiring strict data protection measures and potentially limiting user trust and adoption.

4. Mood-Music Match Precision:

The system's ability to accurately match music to complex emotional states may not always meet user expectations, as mood detection technology and emotional interpretation are still developing fields.

4. SYSTEM DESIGN

4.1 System Design

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.[3]

4.1.1 Site Map

- **Home Page**
    - Overview of the System
    - Quick Guide
- **Registration**
    - Sign-Up Form (with age verification)
- **Login**
    - User Login
- **Facial Recognition**
    - Capture Face
    - Mood Recognition
- **Music Player**
    - Song Library (categorized by mood)
    - Now Playing
    - Controls (Play, Stop)

4.1.2 System Design Overview

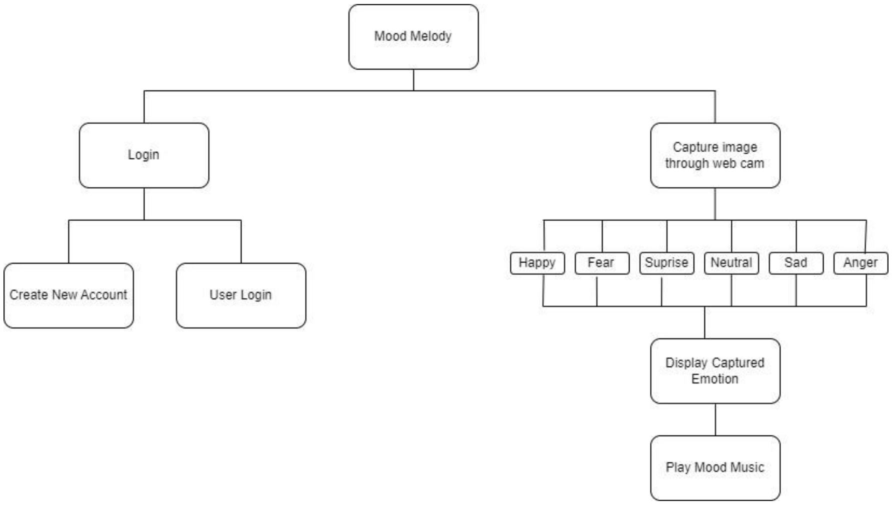


Fig. 4.1 System Design Overview

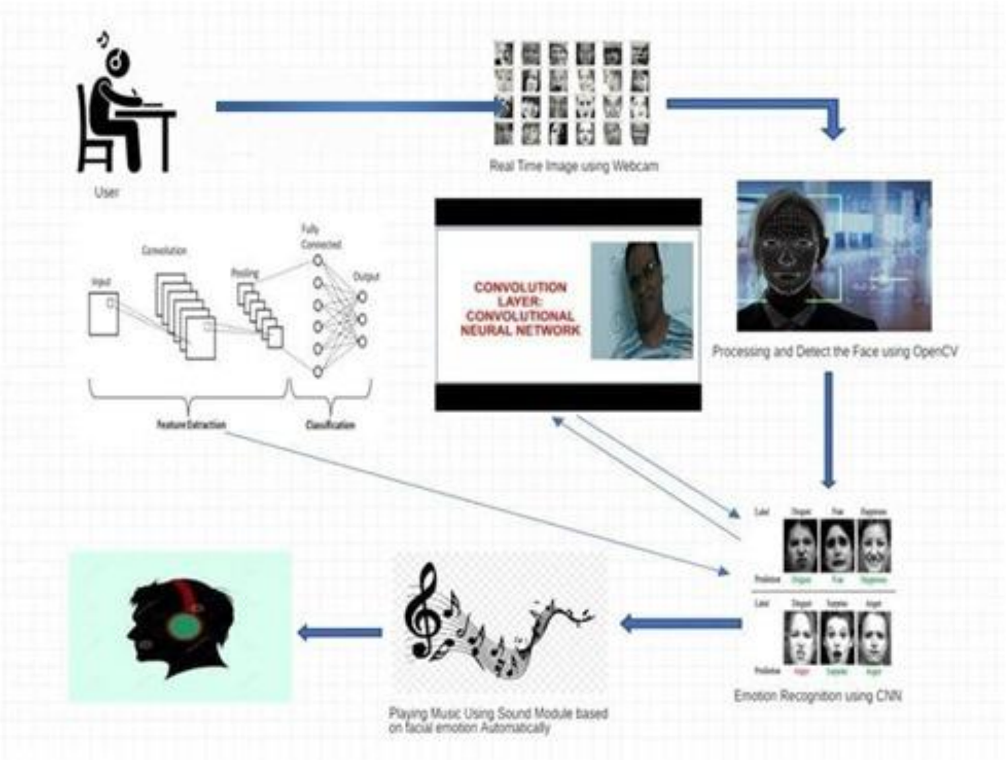

User
Real Time Image using Webcam
Processing and Detect the Face using OpenCV
Input
Convolution
Pooling
Fully Connected
Output
Feature Extraction
Classification
CONVOLUTION LAYER: CONVOLUTIONAL NEURAL NETWORK
Label
Predicted
Emotion Recognition using CNN
Playing Music Using Sound Module based on facial emotion Automatically

Fig. 4.2 System Architecture

Here's a detailed description of each component and how they interact within the system:

1. User: The starting point of the system where an individual interacts with the application through a user interface.

2. Real-Time Image Using Webcam: The user's facial expressions are captured in real-time using a webcam. This visual input serves as the data source for the emotion detection process.

3. Convolutional Neural Network (CNN):

- The system uses a Convolutional Neural Network to process and analyze the facial image data.
- The CNN likely consists of several layers, including:
- Input layer: Receives the raw pixel data from the images captured by the webcam.
- Convolution layers: Apply various filters to extract features from the images.

- Pooling layers: Reduce the spatial size of the representation, decreasing the number of parameters and computation in the network, which helps to control overfitting.
- Fully connected layers: These layers connect every neuron in one layer to every neuron in the next layer, culminating in an output that provides the final classification results.
- Feature extraction happens throughout the convolution and pooling layers, where CNN identifies distinguishing features from the facial images that are indicative of emotional states.
- Classification is the final step where CNN categorizes facial expressions into an emotion like happy, sad, angry, etc.[5]

4. Processing and Detecting the Face using OpenCV:

OpenCV (Open-Source Computer Vision Library) is used for real-time image processing, which likely includes detecting the face within the image frame, focusing on the relevant areas for emotion detection.

5. Emotion Recognition Using CNN:

The recognized facial features are then processed by the CNN to determine the emotion expressed by the user.

6. Playing Music Using Sound Module Based on Facial Emotion Automatically:

Once the emotion is recognized, the system communicates with a sound module (or a music library/player) to select and play music that corresponds to the detected emotion. The selection process might use predefined playlists or songs tagged with emotional labels in a database.

4.1.3 Flow chart

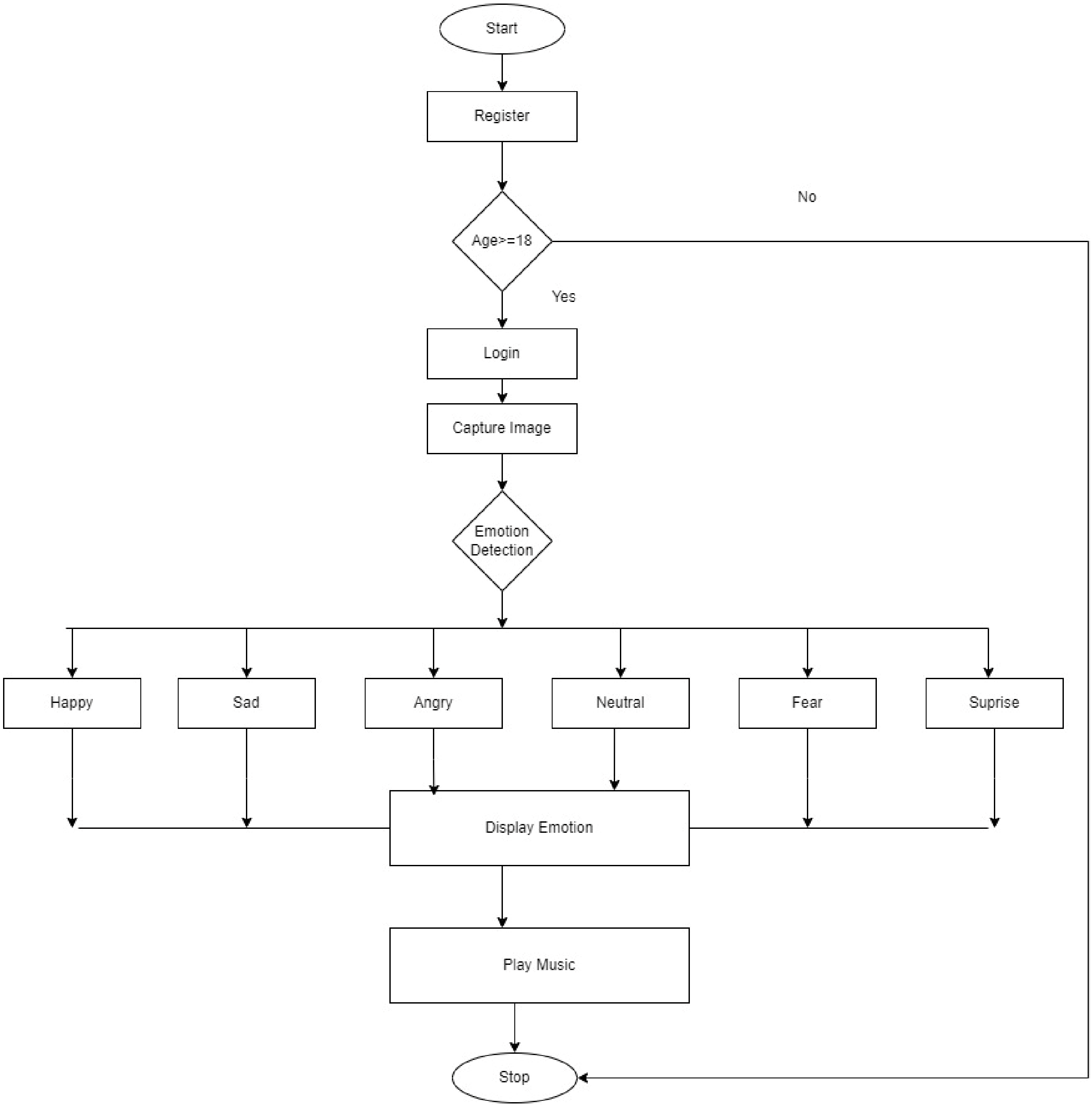


Fig. 4.3 Flowchart

The flowchart you provided depicts the operational sequence of an emotion detection and music play system. Here’s a detailed description:

1. Start: This is the initial step where the user begins interaction with the system.

2. Register: The user must first register. This likely involves entering personal details into the system.

3. Age>=18: A decision point where the system checks if the user's age is 18 or above. This is a control measure to prevent underage users from registering.

- If No, the registration process is halted, presumably denying the user access to the system.
- If Yes, the user proceeds to the next step.

4. Login: After registration and age verification, the user logs into the system to access its features.

5. Capture Image: The system captures a real-time image of the user's face, likely using a webcam or built-in camera.

6. Emotion Detection: The captured image is processed using emotion detection algorithms to identify the user's current emotional state.

7. Emotions (Happy, Sad, Angry, Neutral, Fear, Surprise): The possible outcomes of the emotion detection process. Each outcome leads to the next step, but the path suggests that the outcome might also influence the choice of music to be played.

8. Display Emotion: After detecting the emotion, the system displays the result to the user. This feedback allows the user to know which emotion was detected.

9. Play Music: The system then plays music that corresponds to the detected emotion. This implies that the system has a database or library of songs associated with different emotions.

10. Stop: The final step of the flowchart indicates that the music can be stopped, ending the process.

4.1.4 Usecase Diagram

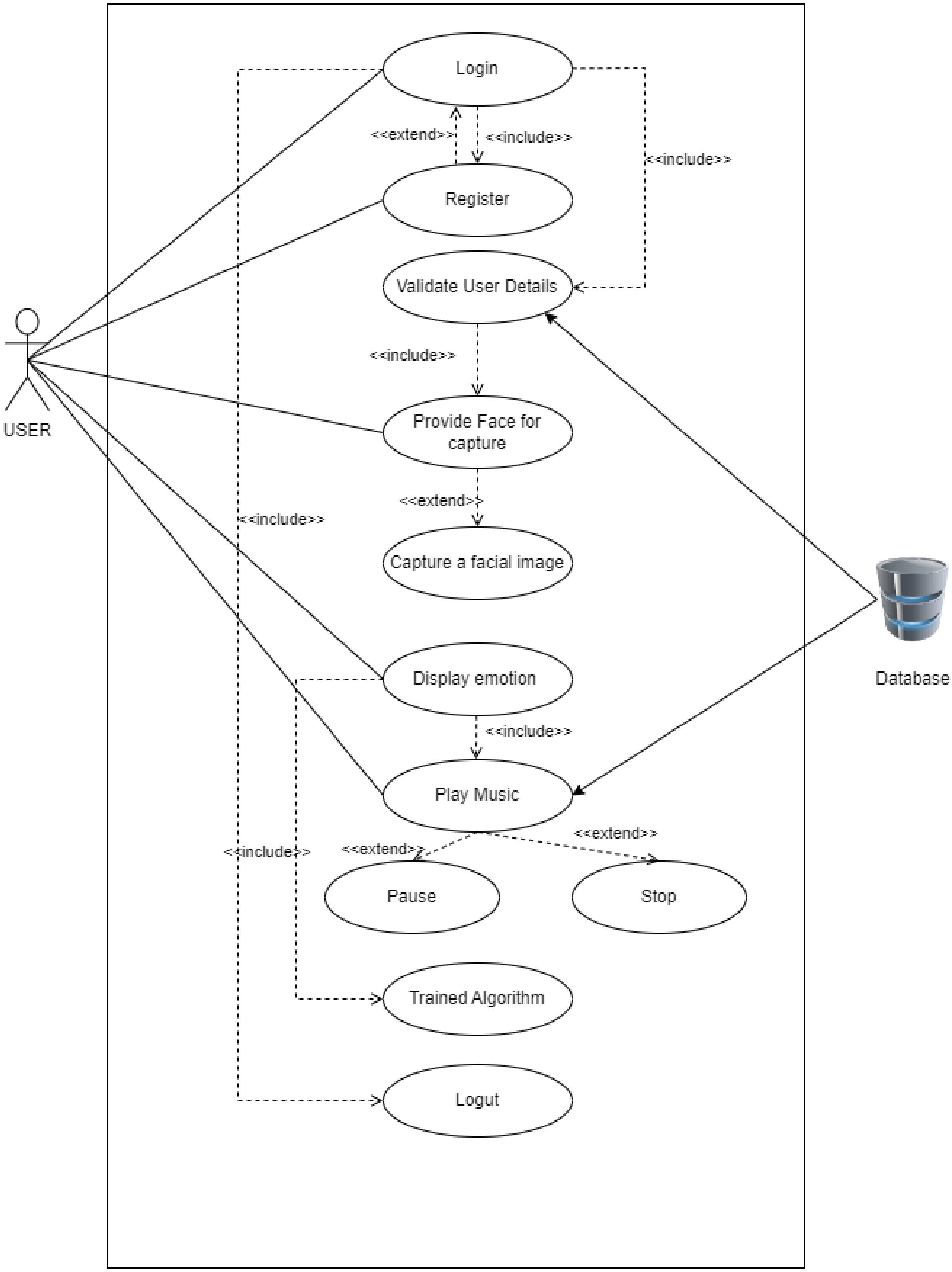


Fig. 4.4 Usecase Diagram

Actors:

1. User:

- Primary Actor in the system who interacts directly with the facial recognition and music player system.
- Performs actions like registering, logging in, having their face captured, and interacting with the music playback interface.
- The actor is subject to an age verification process to ensure compliance with the age restriction policy.

Database:

1. **Music Database:**

- A vital component that stores and manages data related to the users and the music files.
- Contains a collection of songs categorized by the mood they correspond to.
- Accessed by the system to retrieve the appropriate song based on the mood analysis results.
- Stores user registration details, including age, to manage login credentials and enforce age restrictions.

In the use case diagram, the User actor initiates the interactions that drive the system's processes, while the database supports these processes by storing and providing necessary data. The effective operation of the system depends on the harmonious interaction between the User actor and the database to provide a personalized and secure music listening experience based on facially recognized mood analysis.

USECASES:

1. Register:

- Users input their personal information to create an account.
- The system validates the age to ensure the user meets the minimum age requirement of 18.
- Upon successful validation, the system creates a new user account.

2. Login:

- Users enter their credentials to access their account.
- The system authenticates the provided information against the database.
- Access is granted if the credentials match, allowing users to proceed with other operations.

3. Provide Face for Capture:

- Users position themselves in front of the camera to have their facial image taken.
- The system readies the camera for capturing the image.

4. Capture a Facial Image:

- The system captures a real-time image of the user’s face using camera input.
- This image is used for mood analysis by the face recognition module.

5. Display Emotion:

- The captured image is processed by a CNN algorithm to identify facial cues associated with emotions.
- The system determines the user's current mood.
- The identified mood is displayed to the user on the interface.

6. Play Music:

- Based on the detected mood, the system queries the music database to select an appropriate song.
- The system begins music playback through the user interface.

7. Play/Pause/Stop:

- Users can control the music playback.
- They can start the music with "play", temporarily halt it with "pause", and completely stop it using the "stop" function.

8. Trained Algorithm:

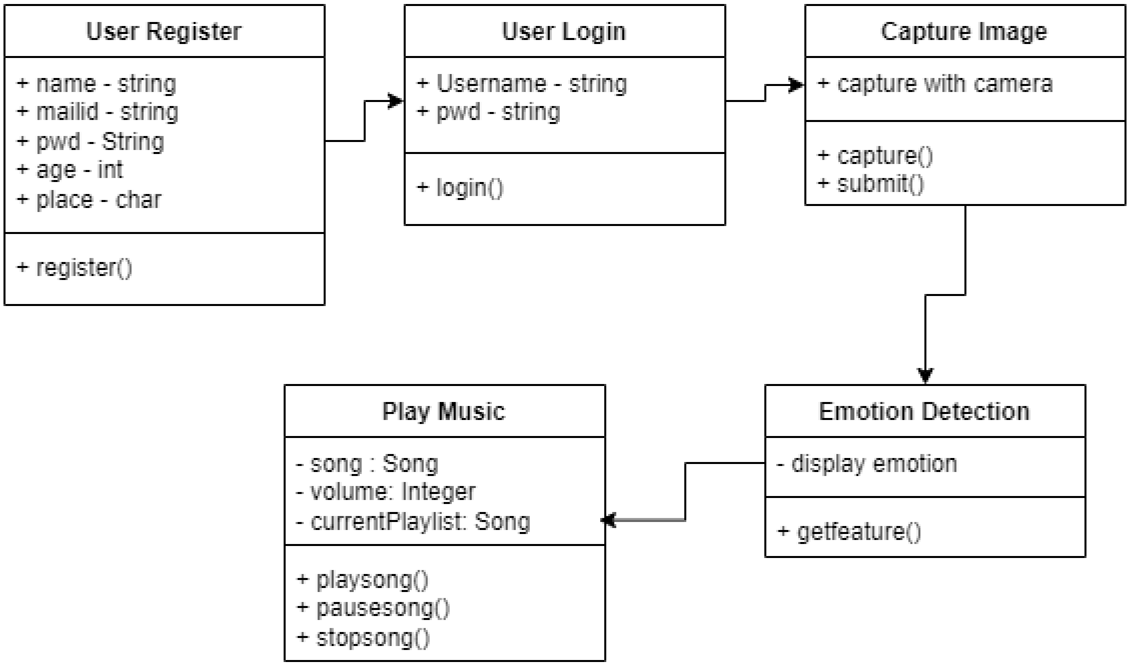
- The Trained Algorithm use case encapsulates the machine learning aspect of your system, specifically the Convolutional Neural Network (CNN) used for facial recognition and mood analysis. It represents the backend process where the captured facial images are analysed to identify the user's mood.

- When a facial image is captured, the system feeds this image into the trained algorithm, which processes it and outputs a mood classification based on learned patterns from training data. This mood classification then influences the subsequent Play Music use case, where the system selects and plays a song that aligns with the detected mood.
- These two use cases, while not directly interacting with each other, both play critical roles in the user experience. The Logout ensures the session is securely closed when user interaction is complete, and the Trained Algorithm provides the intelligent processing that underpins the system’s mood-based music recommendation functionality.

9. Logout:

- The Logout use case represents the functionality that allows users to securely exit their session within the system.
- Once the user decides to end their session, they can trigger the logout action, usually through a logout button in the UI.
- The system then terminates the session, ensuring that no unauthorized actions can be taken after the user leaves, and often redirects the user back to the login page or landing page.
- This process helps maintain the security of the user's account by preventing residual access after the user's intended interaction is complete.

Each use case describes the sequential interactions between the user and the system, from registration to enjoying music tailored to their current emotional state. The system is designed to seamlessly integrate facial recognition and mood analysis to enhance the user experience with personalized music playback.



4.1.5 Class Diagram

Fig 4.5 Class Diagram

The class diagram you've provided outlines the structure of your facial recognition and music player system. Let's go through each class with its attributes and operations:

Class: User Register

Attributes:

- name: string - Stores the full name of the user.
- mailid: string - Stores the email ID of the user.
- pwd: String - Holds the user's password.
- age: int` - Records the age of the user.
- place: char - Keeps a character data type value, which might be used to store abbreviated place information.

Operations:

- + register () - A method that handles the registration process for new users.

Class: User Login

Attributes:

- Username: string - The username for the user account.
- pwd: string - The password for the user account.

Operations:

- + login () - A method to authenticate users trying to log in to the system.

Class: Capture Image

Attributes:

- No attributes are listed for this class in the provided diagram.

Operations:

- + capture () - A method likely responsible for initiating the capture of a facial image via the camera.
- + submit () - This method may handle the submission of the captured image for further processing, such as mood analysis.

Class: Play Music

Attributes:

- song: Song - An object or structure representing a song.
- volume: Integer - An integer value representing the volume level.
- current Playlist: Song - Likely a collection or list of Song objects representing the current playlist.

Operations:

- + playsong() - A method that initiates the playback of a song.
- + pausesong() - A method that pauses the currently playing song.
- + stopsong()- A method that stops the playback of the music.

Class: Emotion Detection

Attributes:

- display emotion - This might be a text or graphical representation intended to display the detected emotion to the user. It is atypical for attributes to have space in their names; it may be an error in the diagram.

Operations:

• + getfeature() - A method that presumably analyzes the captured image to detect facial features that are used to determine the user’s current emotion.

In this diagram, each class represents a different component or functionality of the system. The "+" symbol before methods indicates that they are public and can be accessed by other classes or components. However, the diagram seems to mix different levels of abstraction and might not strictly follow convention (for example, attribute names usually do not contain spaces, and their types are typically more specific). It's also worth noting that relations between these classes are not explicitly detailed in the diagram, which would typically be represented by arrows indicating associations, dependencies, or inheritances.

4.1.6 Activity Diagram

1.Home (Start):

- This is the entry point where users begin their interaction with the system. It's a state of readiness for the user to choose the next action.

2. New User Decision:

- If the individual is accessing the system for the first time, they are considered a 'New User' and directed to the registration process.
- If the individual is already registered, they proceed to the 'Login' activity.

3. Register:

- New users provide necessary information to create an account.
- The system includes a conditional check for the age. It assesses whether the user's age is equal to or greater than 18 years.
- If the age condition is not met (i.e., the user is under 18), the registration process is halted, and the user cannot proceed further.
- If the user is 18 or older, the registration is accepted, and the user’s account is created.

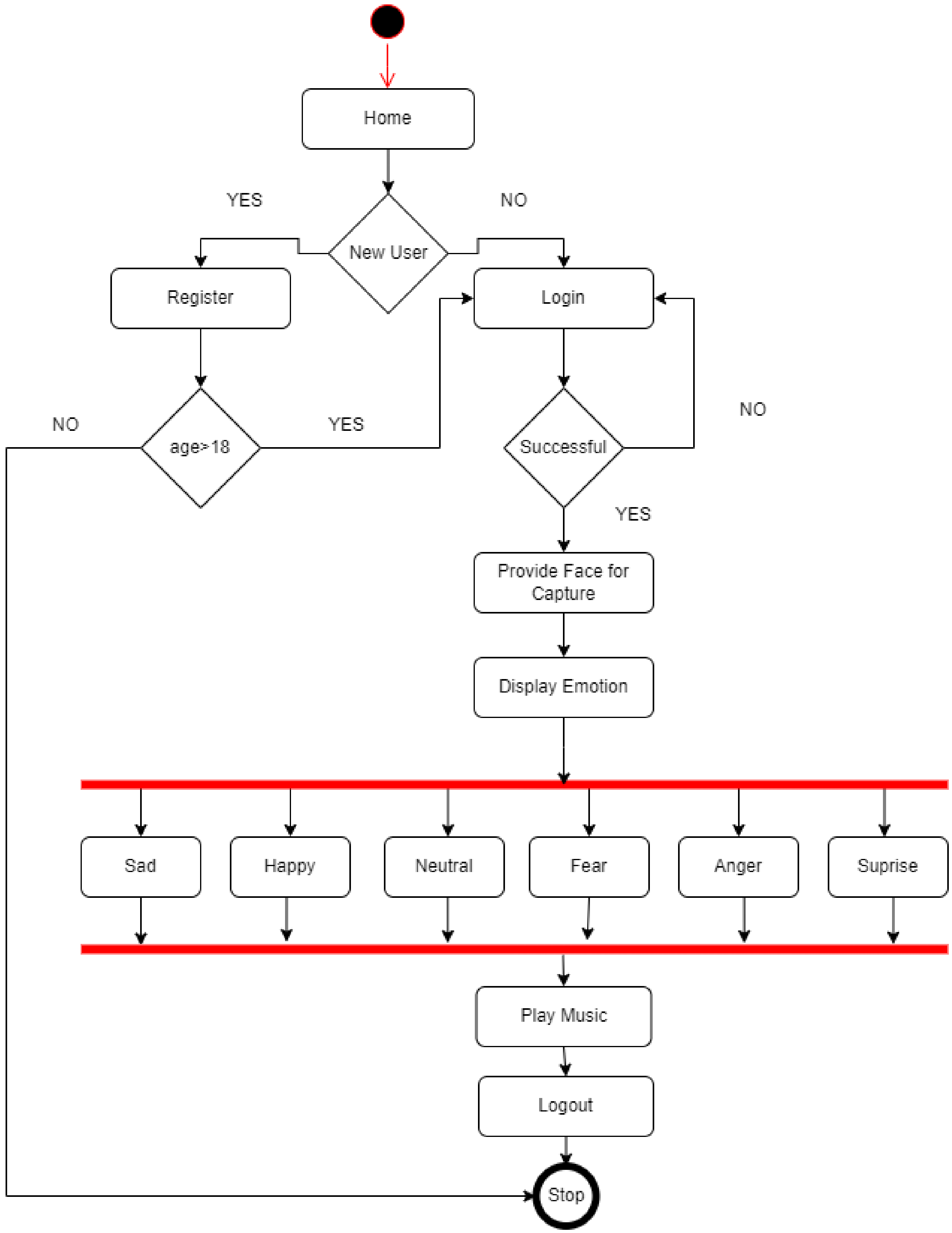


Fig. 4.6 Activity Diagram

4. Login:

- Returning users enter their credentials (username and password).
- A verification process checks the validity of the credentials.
- If the login is not successful (credentials do not match), the user is presumably prompted to try again.
- If the login is successful, access is granted to proceed to the next step in the system.

5. Provide Face for Capture:

- Post-login, the user is required to provide a facial image.
- This step involves the user positioning themselves in front of a camera device for the system to capture their facial expression.

6. Display Emotion:

- The system processes the captured facial image using a facial recognition algorithm, typically a machine learning model, to interpret the user's current emotional state.
- The emotion detected could be one of several predefined categories, such as Sad, Happy, Neutral, Fear, Anger, or Surprise.

7. Mood-based Music Playback:

- Depending on the identified emotion, the system then selects an appropriate music track from the database.
- Music corresponding to the emotion is played, creating a personalized user experience.
- The diagram suggests that the music will continue to play until a 'Stop' condition is reached.

8. Logout:

- After the music session, users can log out, which would be an action to securely close their active session with the system.
- This would ensure that their account is not left open or accessible by others post-interaction.

9. Stop (End Event):

- This signifies the user's active session within the system ending.

- This could be a result of the user choosing to stop the music manually and/or logging out of the system.
- It indicates the completion of the interaction cycle within the system, at which point the user can either start a new session from the 'Home' or exit the system entirely.

Throughout the diagram, the flow is dictated by decisions and actions taken by the user, with system processes responding to these inputs to provide a seamless and secure service from start to finish.

5. IMPLEMENTATION

5.1 Implementation

There are five major tasks in this phase: coding, testing, installation, documentation, and training.The purpose of this phase is to convert the physical system specifications into working and reliablesoftware and hardware, document the work that have been done and provide help for current and future users. The application is developed in keeping the user’s perspective. The application is targeted to end use so it must fulfill the requirement of the users. Various tools and coding languages are used, which are essential for the development of the application.[3][4]

The implementation steps include:

- Web-based application integrating facial recognition with music playback.
- Facial image capture for emotion recognition on the frontend.
- Real-time processing of the image using CNN via TensorFlow in the backend.
- Mood-based music streaming functionality within the user interface.

5.2 Development Methodology

5.2.1 Integration Plan

The integration plan involves connecting the frontend and backend services seamlessly. OpenCV integrates with the frontend to capture real-time images, TensorFlow processes these images for facial recognition, and Flask serves as the intermediary to handle requests and serve responses. MySQL stores user data, and pandas may be used for data manipulation for features like recommendations.

- Frontend and backend integration using Flask as the web server.
- Use of OpenCV for real-time image capture from the user's camera.
- TensorFlow for image processing and mood detection.
- MySQL for storing user data and session information.
- Pandas for data manipulation, possibly for music recommendation features.

5.2.2 Accessibility Guidelines

- Compliance with WCAG 2.1 standards for inclusivity.
- Keyboard navigability and screen reader compatibility.
- Sufficient color contrast and text size for readability.
- The website should be designed with a high contrast ratio to make it easier to read for users with visual impairments.

5.2.3 Responsive Design

- Use of a mobile-first design approach.
- Implementation with CSS media queries for responsiveness across devices.
- Testing on multiple devices for cross-platform compatibility.

5.3 Technology Used

5.3.1 Algorithm

Convolutional Neural Networks (CNNs) are a specialized type of neural network used predominantly for image processing and recognition tasks due to their architecture, which is particularly adept at capturing spatial hierarchies in visual data. The architecture of a CNN typically comprises several layers that each perform distinct transformations on the input data. The first is the convolutional layer, which applies various filters to detect low-level features such as edges and color gradients. These filters are designed to be sensitive to specific aspects of the input data, activating in response to particular patterns and features.[5]

Following the convolutional layer is the pooling or subsampling layer, which reduces the dimensionality of the data, thereby reducing the computational load for the network and helping to prevent overfitting. This is achieved by pooling the outputs of neuron clusters at one layer into a single neuron in the next layer. Commonly, a max pooling operation is used, which takes the maximum value from a set of inputs.

As we advance deeper into the network, the layers aggregate these low-level features to detect high-level features such as shapes and specific objects. These are then fed into fully connected layers, which interpret these high-level features to make a final prediction or classification. The network employs activation functions, like the ReLU or sigmoid functions, which introduce non-linear properties allowing the network to handle complex, non-linear hypotheses.

Training a CNN involves using a large dataset with labeled inputs, where the network adjusts its weights through a process known as backpropagation. This iterative optimization of weights allows CNN to improve its predictions over time. CNNs have revolutionized the field of computer vision, showcasing remarkable performance in tasks such as face recognition and emotion detection. In your project, the CNN algorithm plays a crucial role by interpreting facial expressions and determining the user's mood, which is then used to drive the music player system to select and play music that aligns with the detected emotional state.

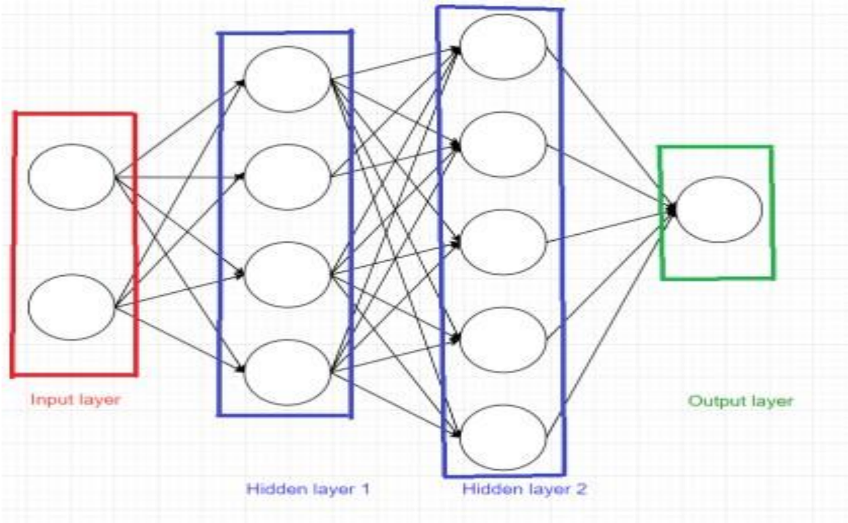


Fig 5.1 CNN Layers

5.3.2 Front-end Technology

- HTML5, CSS3, and JavaScript for building the user interface.
- Potential use of a JavaScript framework for dynamic content.

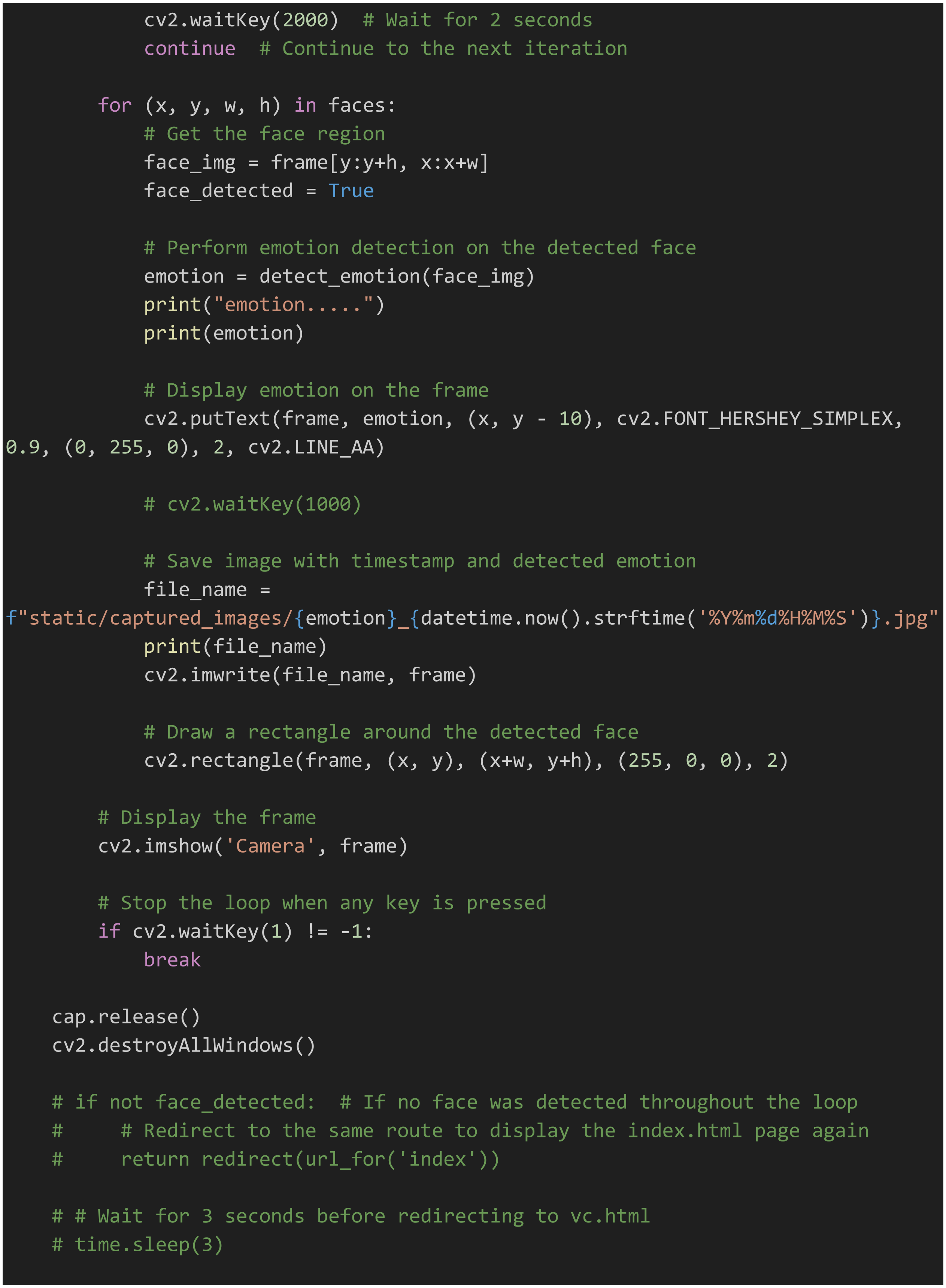
5.3.3 Backend Technology

- Flask as the server-side web framework written in Python.
- TensorFlow for running the CNN algorithm for emotion detection.
- MySQL database for persistent data storage.
- Anaconda for managing Python environments and package management.

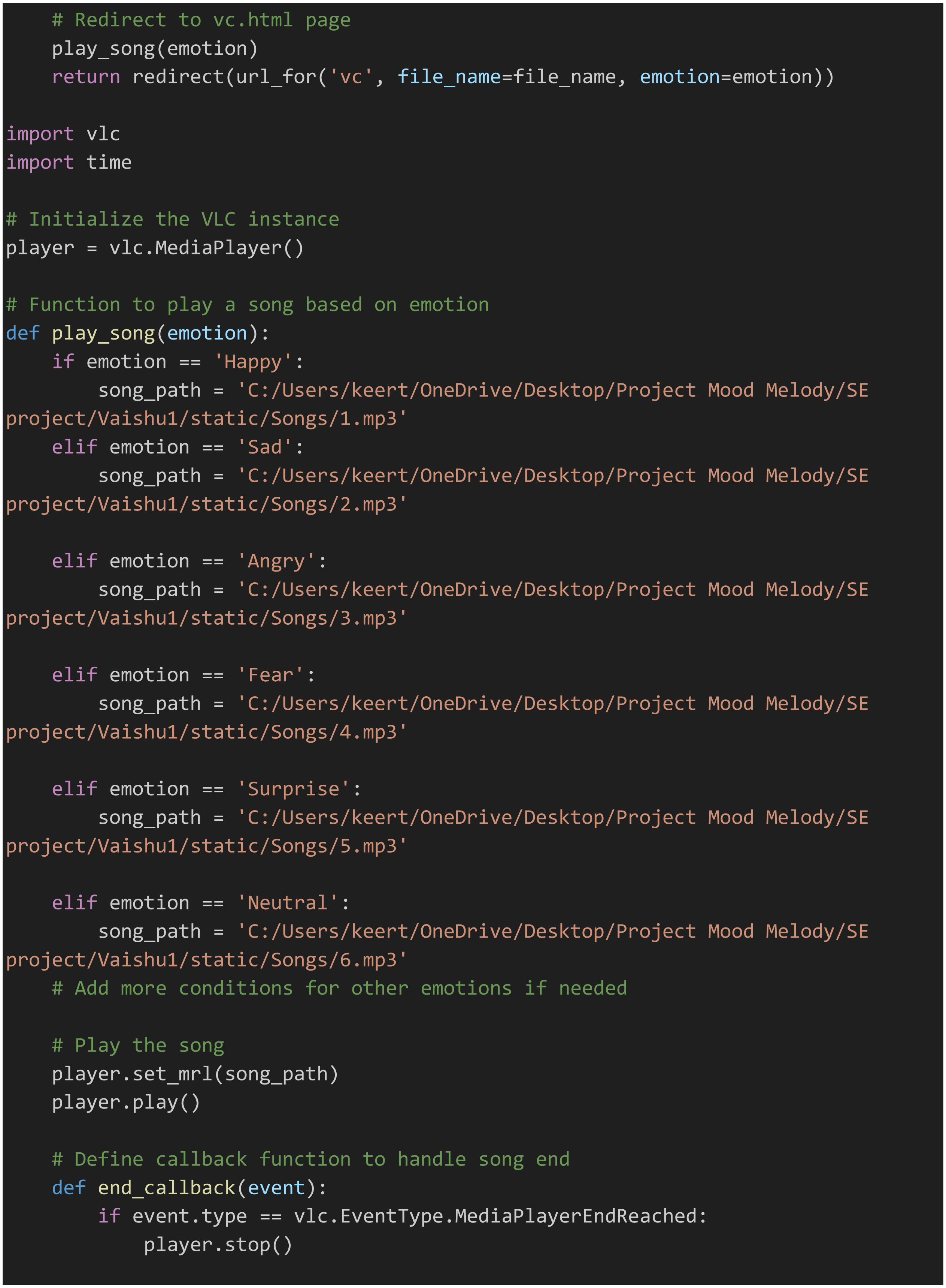
SOURCE CODE

App.py

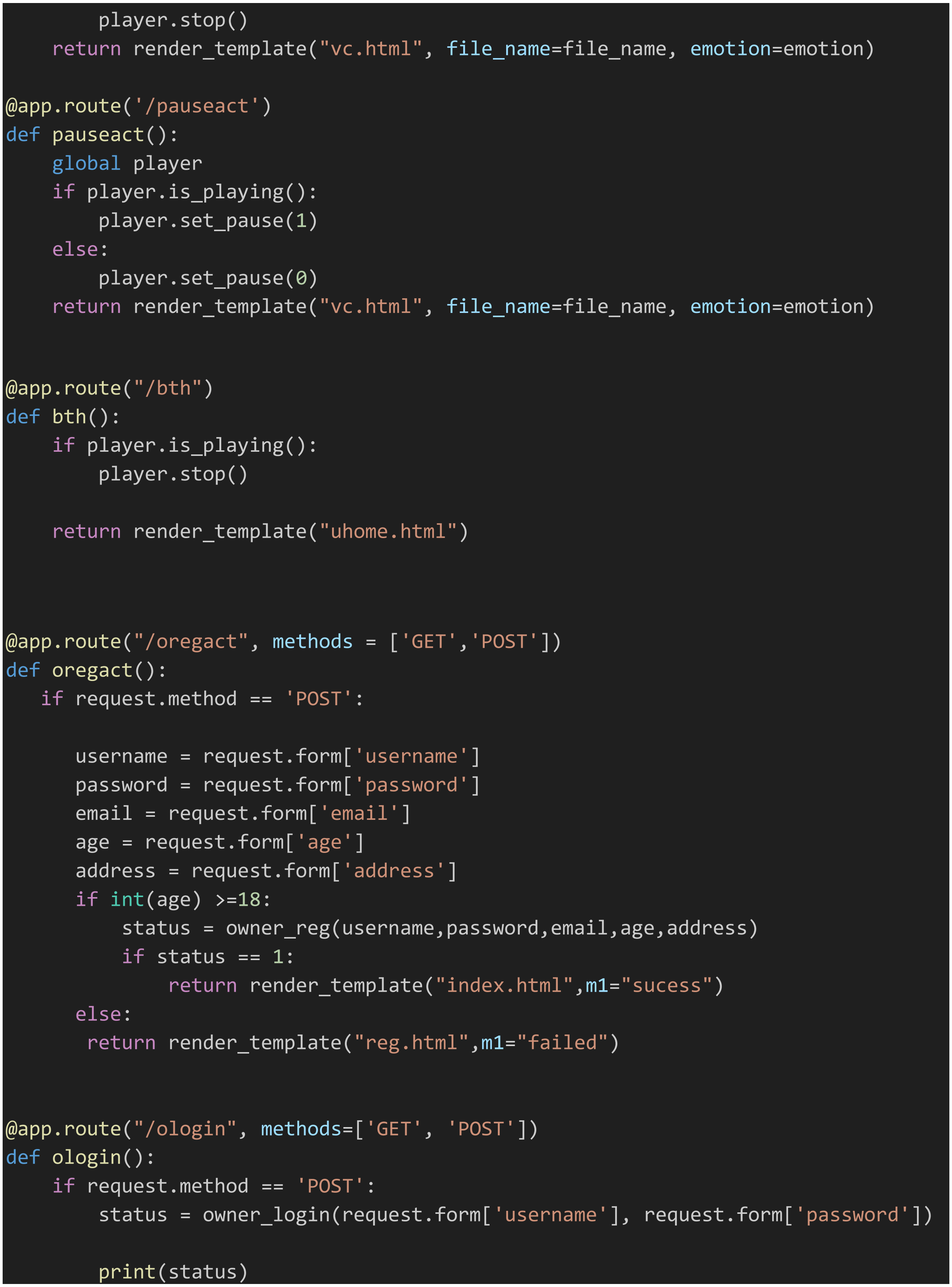
```
      from flask import Flask, render_template, redirect, url_for,
request,session
import cv2
from tensorflow.keras.models import load_model
import numpy as np
import os
from datetime import datetime
import time
from database import db_connect, upload_act,owner_reg,owner_login

app = Flask(__name__)
app.secret_key = os.urandom(24)

# Load your Kaggle emotion detection model
emotion_model = load_model("C:/Users/keert/OneDrive/Desktop/Project Mood
Melody/SE project/Vaishu1/train_model.h5")

# Emotions labels
emotion_labels = {0: 'Angry', 1: 'Disgust', 2: 'Fear', 3: 'Happy', 4: 'Sad', 5:
'Surprise', 6: 'Neutral'}

# Load the pre-trained Haar Cascade Classifier for face detection
face_cascade = cv2.CascadeClassifier("C:/Users/keert/OneDrive/Desktop/Project
Mood Melody/SE project/Vaishu1/haarcascade_frontalface_default.xml")


# Function to preprocess image for the model
def preprocess_image(img):
    # Convert image to RGB
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (48, 48))
    img = np.expand_dims(img, axis=0)
    img = img / 255.0
    return img

# Function to detect emotions from an image
def detect_emotion(img):
    img_processed = preprocess_image(img)
    prediction = emotion_model.predict(img_processed)
    emotion_label = emotion_labels[np.argmax(prediction)]
    return emotion_label

@app.route('/')
def index():
    return render_template('index.html')

@app.route("/reg.html")
def reg():
    return render_template("reg.html")

@app.route("/index")
def index1():
    return render_template("index.html")

@app.route('/detect_emotion', endpoint='detect_emotion')
def detect_emotion_route():
    cap = cv2.VideoCapture(0)

    # Create directory to save images
    os.makedirs('static/captured_images', exist_ok=True)

    face_detected = False  # Flag to indicate if a face is detected

    while True:
        ret, frame = cap.read()
        if not ret:
            break

        # Perform face detection
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3, 
minNeighbors=5)

        if len(faces) == 0:  # If no face is detected
            # Display popup message
            cv2.putText(frame, "Face Not Recognized", (10, 50), 
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2, cv2.LINE_AA)
            cv2.imshow('Camera', frame)
```



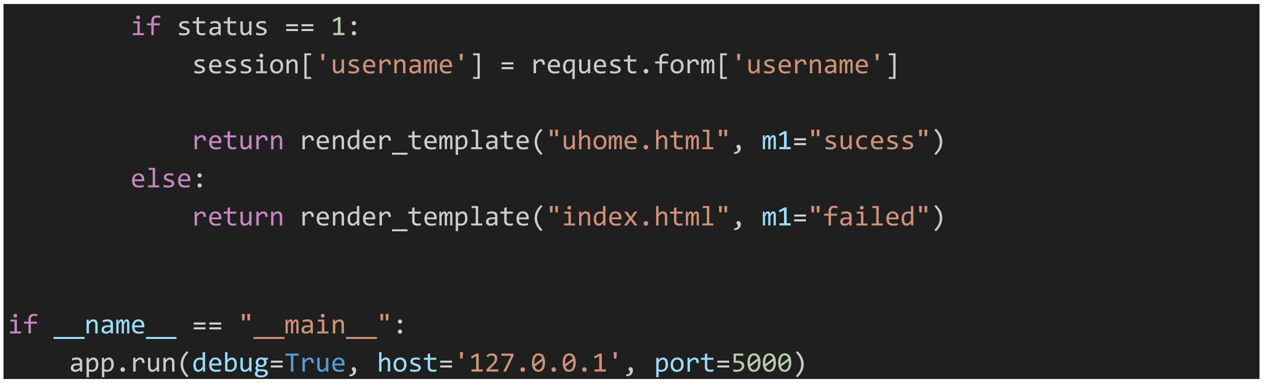
```
            cv2.waitKey(2000)  # Wait for 2 seconds
            continue  # Continue to the next iteration

        for (x, y, w, h) in faces:
            # Get the face region
            face_img = frame[y:y+h, x:x+w]
            face_detected = True

            # Perform emotion detection on the detected face
            emotion = detect_emotion(face_img)
            print("emotion.....")
            print(emotion)

            # Display emotion on the frame
            cv2.putText(frame, emotion, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.9, (0, 255, 0), 2, cv2.LINE_AA)

            # cv2.waitKey(1000)

            # Save image with timestamp and detected emotion
            file_name =
f"static/captured_images/{emotion}_{datetime.now().strftime('%Y%m%d%H%M%S')}.jpg"
            print(file_name)
            cv2.imwrite(file_name, frame)

            # Draw a rectangle around the detected face
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

        # Display the frame
        cv2.imshow('Camera', frame)

        # Stop the loop when any key is pressed
        if cv2.waitKey(1) != -1:
            break

    cap.release()
    cv2.destroyAllWindows()

    # if not face_detected:  # If no face was detected throughout the loop
    #     # Redirect to the same route to display the index.html page again
    #     return redirect(url_for('index'))

    # # Wait for 3 seconds before redirecting to vc.html
    # time.sleep(3)
```



```
    # Redirect to vc.html page
    play_song(emotion)
    return redirect(url_for('vc', file_name=file_name, emotion=emotion))

import vlc
import time

# Initialize the VLC instance
player = vlc.MediaPlayer()

# Function to play a song based on emotion
def play_song(emotion):
    if emotion == 'Happy':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/1.mp3'
    elif emotion == 'Sad':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/2.mp3'

    elif emotion == 'Angry':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/3.mp3'

    elif emotion == 'Fear':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/4.mp3'

    elif emotion == 'Surprise':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/5.mp3'

    elif emotion == 'Neutral':
        song_path = 'C:/Users/keert/OneDrive/Desktop/Project Mood Melody/SE
project/Vaishu1/static/Songs/6.mp3'
    # Add more conditions for other emotions if needed

    # Play the song
    player.set_mrl(song_path)
    player.play()

    # Define callback function to handle song end
    def end_callback(event):
        if event.type == vlc.EventType.MediaPlayerEndReached:
            player.stop()
    # Register the callback function
    player.event_manager().event_attach(vlc.EventType.MediaPlayerEndReached,
end_callback)

@app.route("/vc.html")
def vc():
    # Retrieve URL parameters
    global file_name
    file_name = request.args.get('file_name')
    global emotion
    emotion = request.args.get('emotion')
    return render_template("vc.html", file_name=file_name, emotion=emotion)

from flask import send_file
# import pygame

# @app.route('/playact' )
# def playact():
#     # Get the value of the 'emotion' parameter from the URL
#     emotion = request.args.get('emotion')
#     data = upload_act(emotion)
#     print(data)
#     data1 = data[0][0]
#     pygame.init()

#     # Load the song
#     file_path = "static/songs/" + data1
#     pygame.mixer.music.load(file_path)

#     # Play the song
#     pygame.mixer.music.play()

#     # Wait until the song finishes playing
#     while pygame.mixer.music.get_busy():
#         pygame.time.Clock().tick(10)  # Adjust the tick value as needed

#     # Clean up
#     pygame.mixer.music.stop()
#     pygame.quit()


@app.route('/stopact')
def stopact():
    global player
    if player.is_playing():
```



```
        player.stop()
    return render_template("vc.html", file_name=file_name, emotion=emotion)

@app.route('/pauseact')
def pauseact():
    global player
    if player.is_playing():
        player.set_pause(1)
    else:
        player.set_pause(0)
    return render_template("vc.html", file_name=file_name, emotion=emotion)


@app.route("/bth")
def bth():
    if player.is_playing():
        player.stop()

    return render_template("uhome.html")



@app.route("/oregact", methods = ['GET','POST'])
def oregact():
   if request.method == 'POST':

      username = request.form['username']
      password = request.form['password']
      email = request.form['email']
      age = request.form['age']
      address = request.form['address']
      if int(age) >=18:
          status = owner_reg(username,password,email,age,address)
          if status == 1:
              return render_template("index.html",m1="sucess")
      else:
       return render_template("reg.html",m1="failed")


@app.route("/ologin", methods=['GET', 'POST'])
def ologin():
    if request.method == 'POST':
        status = owner_login(request.form['username'], request.form['password'])

        print(status)
```



```
        if status == 1:
            session['username'] = request.form['username']

            return render_template("uhome.html", m1="sucess")
        else:
            return render_template("index.html", m1="failed")


if __name__ == "__main__":
    app.run(debug=True, host='127.0.0.1', port=5000)
```

6. TESTING

6.1 Software Description

The Facial Recognition and Music Player System is an interactive platform that employs a CNN algorithm to analyze and interpret users' facial expressions to detect their mood and plays music that reflects the detected emotion.[5]

6.1.1 Functions of the System

- **User Registration and Login:** Secure user registration with age verification and user login functionality.
- **Facial Image Capture:** Capture and process user facial expressions in real-time.
- **Emotion Detection:** Employ a CNN algorithm to detect the user's current mood based on the captured facial expression.
- **Music Selection and Playback:** Select and play a song from a pre-defined music library that corresponds to the detected mood. Include UI controls for playing and stopping the music.

6.1.2 Input

- User registration & User login
- User facial expressions will be captured through an integrated camera.
- User selections will be entered through the system’s interactive interface.

6.1.3 Output

- Audio playback of the recognized emotion
- Visual display of the detected emotion and playback controls on the user interface.

6.2 Requirements State Resources Needed

- **User Age Verification:** The system must verify that users are 18 years or older during registration.
- **Emotion Detection Accuracy:** The system should accurately detect and interpret a wide range of user emotions.

- **Music Playback Functionality:** Based on emotion detection, the system should play the appropriate music.

6.2.1 Equipment

- **Cameras:** High-quality cameras to capture facial expressions accurately.
- **Audio Output Devices:** For quality music playback during testing.

6.2.2 Software & Libraries

- **Flask -** A web framework, which is a type of library for building web applications in Python.
- **TensorFlow -** An open-source library for numerical computation and machine learning.
- **MySQL -** A relational database management system (RDBMS), which is software for creating, managing, and querying databases.
- **Pandas-** A library providing high-performance, easy-to-use data structures, and data analysis tools for Python.
- **OpenCV-** A library focused on real-time computer vision.
- **Anaconda-** A distribution of Python and R for scientific computing, which includes many data science packages and tools. It is software that manages libraries and environments.
- **Visual Studio Code -** An integrated development environment (IDE) software for building and debugging code.

6.3 Testing Materials Needed for the test

6.3.1 Documentation

- Test plan will outline strategies and approaches.
- Requirement specifications will be validated against test cases.

6.3.2 Test Inputs and Sample Outputs

- **Test Inputs:** Valid and invalid registration data, login credentials, various facial images representing different emotions, and user interactions with the playback controls.

- **Sample Outputs:** Confirmation of registration success or failure, login success or failure messages, the accuracy of emotion recognition, appropriateness of the selected music track, and responsiveness of the playback controls.

6.4 Test Objectives

Number	Test Objective	Test Priority	Completion Criteria
1	Validate accuracy of emotion detection.	High	Emotion detection is accurate in at least 95% of test cases under diverse conditions.
2	Ensure music selection aligns with detected emotion.	High	Music played corresponds with the detected emotion in 90% of test cases.
3	Assess UI responsiveness and usability.	Medium	UI responds within 2 seconds, and user satisfaction is rated 4 out of 5 in usability tests.
4	Test system performance under load.	Medium	System maintains functionality with a response time under 3 seconds during peak load tests.
5	Verify user override functionality for music selection.	Low	Users can successfully override the automated music selection in 100% of test cases.
6	Check cross-platform compatibility.	Medium	Application runs without issues on all targeted operating systems and devices.
7	Confirm data security and user privacy protection.	High	All personal data is encrypted, and user privacy settings are correctly applied in all scenarios.
8	Monitor system adaptability to user feedback.	Low	System shows a 10% improvement in music selection aligning with user preferences over time.
9	Validate user registration process.	High	User can register successfully with verification of age restriction, and data is stored securely.
10	Verify user login functionality.	High	User login is successful with correct credentials and fails with incorrect ones. Session management is secure.

Table 6.1 Test Objectives

6.5 Specifications and Evaluation

6.5.1 Test Conditions to be Evaluated

- **Registration and Login**
  - Users above 18 must be able to register successfully.
  - Users under 18 should not be able to register.
  - Registered users must be able to log in with the correct credentials.
  - Unregistered or invalid login attempts should be rejected.
- **Facial Recognition and Mood Detection**
  - The system should correctly identify and label the user’s mood from facial expressions.
  - The algorithm should handle variations in lighting and facial orientation.
- **Music Playback Based on Mood**
  - Music selected must correspond with the detected mood label.
  - Playback controls should function correctly (play, stop).

6.5.2 Requirements

- Secure user registration and login system with age verification.
- Accurate facial recognition to determine user mood.
- A well-structured music library with appropriately tagged mood categories.
- Responsive UI for an engaging user experience.

6.5.3 Software Functions

- **Registration and Authentication:** Handles new user registrations and login processes.
- **Facial Image Processing:** Captures and analyzes facial expressions.
- **Mood Detection:** Utilizes a CNN algorithm to detect emotions from facial images.
- **Music Database Management:** Stores and retrieves music tracks based on mood labels.
- **Playback Control:** Allows users to control music playback.

6.5.4 Test Matrix

Software Function	Unit Test	Integration Test	System Test	User Acceptance Test	Code Inspection
User Registration	✓	✓		✓	✓
User Login	✓	✓		✓	✓
Facial Image Capture	✓	✓	✓		✓
Emotion Detection Accuracy	✓	✓	✓		✓
Music Selection Appropriateness	✓	✓	✓	✓	✓
UI Responsiveness				✓	
System Performance under Load		✓	✓		
User Override Functionality	✓	✓		✓	✓
Cross-Platform Compatibility			✓	✓	
Data Security and Privacy	✓	✓	✓	✓	✓
Adaptability to User Feedback		✓		✓	

Table 6.2 Test Matrix

- **Unit Test:** Focuses on the smallest parts of the application independently, such as functions or methods.
- **Integration Test:** Checks the interaction between integrated units/modules to detect interface defects.
- **System Test:** Involves testing the complete and fully integrated software product, usually in an environment that mimics production.
- **User Acceptance Test:** Carried out with the actual intended users to ensure the system meets their requirements and is ready for use in real scenarios.

- **Code Inspection:** A thorough examination of the code for potential bugs, security vulnerabilities, and adherence to coding standards.

6.5.5 Test/Function Relationship

- **Registration and Authentication Tests:** Validate data input, verify age restrictions, and confirm security.
- **Facial Image Processing Tests:** Assess the accuracy of image capture under various conditions.
- **Mood Detection Tests:** Evaluate the CNN algorithm's effectiveness in different scenarios.
- **Music Database Management Tests:** Verify the integrity and query responsiveness of the music database.
- **Playback Control Tests:** Check the immediacy and responsiveness of UI controls.

6.5.6 Test Progression (Technical Architecture)

The progression of tests from one to another to complete the entire test cycle will be described as follows:

- Begin with Unit Testing of individual components (e.g., emotion detection accuracy, music selection logic).
- Proceed to Integration Testing where the interaction between components is tested (e.g., emotion detection triggering music playback).
- Follow with System Testing to evaluate the complete system's performance.
- Conclude with User Acceptance Testing to ensure the system meets user expectations and requirements.

A diagram of the system architecture will include:

- **Facial Recognition Module**: Will capture and process facial expressions to detect emotions.
- **Music Selection Engine**: Will query the music library and select suitable tracks.
- **User Interface**: Includes registration, login, facial capture, and music control interfaces

- **Data Storage**: Will securely store user preferences, historical data for learning algorithms, and system logs.
- **Data Transfer Connections**: Will include API calls between the system and music databases, as well as data exchange with user devices.

6.6 Test Strategy

The test strategy will describe how test objectives will be met across various test types:

- **Unit Testing**: Will focus on individual components to ensure they function correctly in isolation.
- **Functional Testing**: Will verify that each feature operates according to the requirement specifications.
- **Integration Testing**: Will assess the interactions between integrated components to ensure they work together as intended.
- **System Testing**: Will test the complete, integrated system to validate its compliance with the requirements.
- **Volume Testing**: Will check the system's ability to handle large volumes of data.
- **Stress Testing**: Will evaluate system behavior under peak loads.
- **Performance Testing**: Will measure system responsiveness and stability.
- **Configuration/Installation Testing**: Will ensure the system is properly configured and installed in various environments.

For each test subset, the objectives, techniques, and methods for developing test cases will include:

- **Objectives**: Each test subset will have specific goals, such as ensuring accuracy, reliability, and performance.
- **Technique**: Techniques will range from black-box testing for user experience to white-box testing for internal operations.
- **Test Case Development**: Test cases will be developed based on use cases, user stories, and system requirements.

6.6.1 A Complete Test Document

Software Project: Facial Recognition and Music Player System
Name of Test: User Registration Validity
Test No.: 1

Test Objective	Test Input	Test Procedures	Test Output	Test Controls	Software or Structure Attribute Tested
Validate the validity of user registration, ensuring accurate data handling and enforcement of age restrictions.	User details including full names, dates of birth, email addresses, usernames, and passwords.	1. Run the registration routine for each user detail set. 2. Verify age restriction enforcement. 3. Check for successful registration in the database.	Registration is successful for valid users and rejected for users under 18 or with incomplete data.	1. Run the test suite repeatedly for various age groups. 2. Make necessary adjustments until all registration criteria are met.	Registration process accuracy and data validation routines.

Table 6.3.1 User Registration Validity

Software Project: Emotion Detection and Music Player System
Name of Test: Emotion Detection Accuracy and Response
Test No.: 2

Test Objective	Test Input	Test Procedures	Test Output	Test Controls	Software or Structure Attribute Tested
Validate the accuracy of the emotion detection routine and ensure the music selected corresponds accurately to the detected emotion.	Images with facial expressions corresponding to a comprehensive range of emotions.	Run the emotion detection routine for each image and log the music track selected by the system.	The system accurately detects the emotion and plays a track that matches the detected emotion.	Run the test suite repeatedly, making necessary adjustments until the system meets the accuracy criteria.	Emotion detection algorithm accuracy and music selection relevance.

Table 6.3.2 Emotion Detection Accuracy and Response

Software Project: Emotion Detection and Music Player System
Name of Test: Music Selection Accuracy
Test No.: 3

Test Objective	Test Input	Test Procedures	Test Output	Test Controls	Software or Structure Attribute Tested
To ensure that the music selection algorithm accurately selects music tracks that correspond to the detected emotion.	Detected emotion labels (e.g., "happy", "sad", etc.) and corresponding user preferences.	For each detected emotion, verify that the music selection algorithm chooses a track that fits the emotion and user preferences.	The system plays a track that is deemed appropriate for the detected emotion and adheres to user preferences.	If an inappropriate track is selected, log the incident, adjust the selection criteria, and retest until the algorithm performs as expected.	Music selection algorithm's ability to match music tracks with detected emotions and user preferences.

Table 6.3.3 Music Selection Accuracy

Software Project: Emotion Detection and Music Player System
Name of Test: User Interface Responsiveness and Usability
Test No.: 4

Test Objective	Test Input	Test Procedures	Test Output	Test Controls	Software or Structure Attribute Tested
Ensure the user interface is responsive and user interactions are intuitive.	User commands through the UI such as navigation, music selection, settings adjustments.	Perform and record various user actions to assess responsiveness and functionality.	UI responds immediately and correctly to all user inputs.	Continuously monitor and record performance, iterating over tests after adjustments.	UI responsiveness and user interaction flow.

Table 6.3.4 User Interface Responsiveness and Usability

6.6.2 Test Script

Software Project: Facial Recognition and Music Player System
Software Module: User Registration Validity Module
Test No.: 2

Sequence	Source	Script Event	Evaluation Criteria	Comments
1	Test Engineer	Tester attempts to register a user under 18.	The system should reject the registration attempt.	Testing enforcement of age restriction policy.
2	Test Engineer	Tester attempts to register with a valid age but missing information.	The system should reject the registration due to incomplete data.	Verifying system requires complete data for registration.
3	Test Engineer	Tester attempts to register with all required fields properly filled for a user above 18.	The system should accept the registration and create a new account.	Confirming successful registration for eligible users.
4	Test Engineer	Tester attempts to re-register with an existing username.	The system should reject the registration due to a duplicate username.	Ensuring unique usernames for each account.
5	Test Engineer	Tester uses special characters in the username field not allowed by the system rules.	The system should validate input and reject invalid characters.	Testing input validation for special characters.

Table 6.4.1 User Registration Validity Module

Software Project: Emotion Detection and Music Player System
Software Module: Emotion Recognition Module
Test No: 1

Sequence	Source	Script Event	Evaluation Criteria	Comments
1	Test Engineer	Tester displays a "happy" image to the camera	The system should recognize the emotion as "happy"	Testing recognition accuracy for a single, clear emotion.
2	Test Engineer	Tester displays a "sad" image to the camera	The system should recognize the emotion as "sad"	Confirming system can differentiate between distinctly different emotions.
3	Test Engineer	Tester displays a "neutral" image to the camera	The system should recognize the emotion as "neutral"	Neutral is an important baseline for emotion detection.

Table 6.4.2 Emotion Recognition Module

Software Project: Emotion Detection and Music Player System
Software Module: Music Playback Module
Test No: 2

Sequence	Source	Script Event	Evaluation Criteria	Comments
1	Test Engineer	The system detects "happy" and triggers playback	The system should play music that corresponds to a "happy" mood	Assessing the correctness of mood-based music selection.
2	Test Engineer	The system detects "sad" and triggers playback	The system should play music that corresponds to a "sad" mood	Checking the system's response to varied emotional input.
3	User	User overrides automatic selection	The system should allow the user to select a different track and play it	Verifying user control over music selection.

Table 6.4.3 Music Playback Module

Software Project: Emotion Detection and Music Player System
Software Module: User Interface Module
Test No: 3

Sequence	Source	Script Event	Evaluation Criteria	Comments
1	User	User launches the system UI	The system should display the main dashboard promptly	Ensuring the UI loads efficiently.
2	User	User initiates emotion detection	The system should display the detected emotion on the UI	Checking real-time emotion display functionality.
3	User	User requests a music change through UI	The system should interrupt current playback and start the new selection	Testing user control over music playback.
4	User	User attempts to access help documentation	Help documentation should be easily accessible and provide useful information	Ensuring support features are adequate.
5	Test Engineer	Engineer tests UI responsiveness on different devices	The UI should be responsive and maintain layout across devices	Testing for responsive design compatibility.

6	Test Engineer	Engineer simulates a network disconnect	The system should gracefully handle the disconnect and inform the user	Assessing robustness of UI under network failure.

Table 6.4.4 User Interface Module

6.7 Module Testcases

6.7.1 Detailed Testcases with test data

User Registration - Age Verification

Test Case #	Description	Expected Result	Test Data	Actual Result	Test Case Status
1.1	Register user exactly 18 years old	Registration successful	DOB: "04/25/2006"	Registration successful	Pass
1.2	Register user under 18	Registration rejected	DOB: "04/25/2010"	Registration rejected	Pass
1.3	Register user over 18 without DOB	Registration rejected	No DOB provided	Registration rejected	Pass
1.4	Register user over 18 with valid data	Registration successful	DOB: "04/25/1990"	Registration successful	Pass
1.5	Register user over 18 with future DOB	Registration rejected	DOB: "04/25/2025"	Registration rejected	Pass

Table 6.5.1 User Registration – Age Verification

User Login - Authentication

Test Case #	Description	Expected Result	Test Data	Actual Result	Test Case Status
2.1	Login with valid credentials	Access granted	Username: "johndoe", Password: "SecurePass123"	Access granted	Pass
2.2	Login with invalid username	Access denied	Username: "janedoe", Password: "SecurePass123"	Access denied	Pass
2.3	Login with invalid password	Access denied	Username: "johndoe", Password: "Wrong123"	Access denied	Pass

2.4	Login with empty credentials	Access denied	Username: "", Password: ""	Access denied	Pass
2.5	Login with SQL injection attempt	Access denied	Username: "' OR '1'='1", Password: "' OR '1'='1"	Access denied	Pass

Table 6.5.2 Authentication

Facial Recognition - Mood Detection Accuracy

Test Case #	Description	Expected Result	Test Data	Actual Result	Test Case Status
3.1	Detect a happy mood	Mood "Happy" identified	User smiles at the camera	Mood "Happy" identified	Pass
3.2	Detect a sad mood	Mood "Sad" identified	User frowns at the camera	Mood "Sad" identified	Pass
3.3	Detect a surprised mood	Mood "Surprised" identified	User opens eyes wide at the camera	Mood "Surprised" identified	Pass
3.4	Detect a neutral mood	Mood "Neutral" identified	User has a blank expression	Mood "Neutral" identified	Pass
3.5	Detect an angry mood	Mood "Angry" identified	User scowls at the camera	Mood "Angry" identified	Pass

Table 6.5.3 Mood Detection Accuracy

Music Selection - Appropriateness to Mood Detected

Test Case #	Description	Expected Result	Test Data	Actual Result	Test Case Status
4.1	Play music for happy mood	Happy music plays	Detected mood: "Happy"	Correct happy music plays	Pass
4.2	Play music for sad mood	Sad music plays	Detected mood: "Sad"	Correct sad music plays	Pass
4.3	Play music for surprised mood	Surprised music plays	Detected mood: "Surprised"	Correct surprised music plays	Pass
4.4	Play music for neutral mood	Neutral music plays	Detected mood: "Neutral"	Correct neutral music plays	Pass

4.5	No mood detected - default music	Default music plays	No mood detected	Default music plays	Pass

Table 6.5.4 Appropriateness of Mood Detected

UI Responsiveness and Usability

Test Case #	Description	Expected Result	Test Data	Actual Result	Test Case Status
5.1	Play button responsiveness	Music plays immediately	Click play button	Music plays within 2 seconds	Pass
5.2	Stop button responsiveness	Music stops immediately	Click stop button	Music stops immediately	Pass
5.3	Responsiveness of mood detection	Mood displayed immediately	Present facial expression	Mood displayed within 2 seconds	Pass
5.4	Responsiveness of song selection after mood detection	Song plays soon after mood detection	Detected mood: "Happy"	Song plays within 3 seconds after mood display	Pass
5.5	Navigation responsiveness	UI navigates smoothly	Navigate between pages	Page loads within 2 seconds	Pas

Table 6.5.5 UI Responsiveness and Usability

6.8 Risks and Mitigation

1. Technology Risks:

- **Risk:** Inadequate performance of the CNN algorithm under varied lighting conditions.
- **Mitigation:** Research and implement more robust image preprocessing techniques to normalize lighting variations.

2. Project Management Risks:

- **Risk:** Project delays due to resource unavailability.

- **Mitigation:** Develop a flexible project schedule with buffer periods. Have a backup plan for resource allocation.

3. Data Privacy and Security Risks:

- **Risk:** Unauthorized access to personal data.
- **Mitigation:** Implement strong data encryption, secure API endpoints, and conduct regular security audits.

4. Integration Risks:

- **Risk:** Components (facial recognition, music database) do not integrate smoothly.
- **Mitigation:** Use well-defined interface contracts and conduct early integration testing.

6.9 Maintenance Plan

- **Regular Updates and Patches**

- **Objective:** To keep the system secure and functional, addressing any newly discovered bugs or security vulnerabilities.
- **Action Plan:** Schedule regular updates for all software components, including the libraries (TensorFlow, Flask, MySQL, Pandas, OpenCV) and development tools (Anaconda, Visual Studio Code). Patches should be applied as soon as they are available and tested.

- **Database Maintenance**

- **Objective:** Ensure the database performance remains optimal and data integrity is maintained.
- **Action Plan:** Perform routine database cleanups, optimizations, and backups. This includes archiving old data, re-indexing databases, and checking integrity constraints.

- **Performance Tuning**

- **Objective:** Maintain optimal system performance as user load and data volume grow.
- **Action Plan:** Regularly review system performance metrics and tune hardware or software configurations as needed. This might involve scaling resources up or down based on usage patterns.

7. CONCLUSION & FUTURE ENHANCEMENT

The development of facial recognition and music player system successfully demonstrates the integration of cutting-edge technologies such as convolutional neural networks, Flask, TensorFlow, and more to create a user-centric platform. This system effectively recognizes users' emotions through facial expressions and responds by playing mood-appropriate music, enhancing the user experience through personalization. The project overcame significant challenges, including accurately detecting varied facial expressions and implementing robust user management features like age verification. Looking forward, this system holds great potential for expansion and further refinement, promising to not only improve user engagement but also to explore new applications in personalized digital media and entertainment. This project exemplifies the transformative possibilities of combining AI with practical applications, setting a strong foundation for future innovations in interactive technology.

7.1 Future Enhancement

- **Mood Transition Playlists:** Implement an algorithm that doesn't just match the current mood but also curates a playlist designed to transition the user's mood from one state to another. For example, if a user is detected as sad, the system could start with soothing songs and gradually move to more upbeat tracks to gently lift their spirits.
- **User Feedback Learning:** Integrate machine learning algorithms that learn from user feedback. If a user skips or changes songs frequently, the system would learn their preferences and adjust future selections to better match their tastes.
- **Playlist Customization:** Give users the ability to customize playlists, such as adding or removing songs from the automated selections, saving favorites, and creating mood-based playlists that they can name and share.
- **Advanced Music Controls:** Introduce features such as song rewind, fast forward, shuffle, and repeat, giving users more control over their listening experience.

8. REFERENCES

[1] Quasim, M. T., et al. (2021). Emotion-based music recommendation and classification using machine learning with IoT framework. Soft Computing, 25(18), 12249-12260.

[2] Shalini, S. K., et al. (2021). Facial emotion based music recommendation system using computer vision and machine learning techniques. Turkish Journal of Computer and Mathematics Education, 12(2), 912-917.

[3] Abdullah, S. M. S. A., et al. (2021). Multimodal emotion recognition using deep learning. Journal of Applied Science and Technology Trends, 2(02), 52-58.

[4] Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. Sensors, 21(9), 3046.

[5] Abdulsalam, W. H., Alhamdani, R. S., & Abdullah, M. N. (2019). Facial emotion recognition from videos using deep convolutional neural networks. International Journal of Machine Learning and Computing, 9(1), 14-19.

INSTALLATION STEPS

Pre-requisite

Step 1: Install Anaconda

If you haven't already installed Anaconda, download it from Anaconda's website and follow the installation instructions.

Step 2: Create a New Conda Environment

Open the Anaconda Prompt and create a new virtual environment:

```
conda create --name myenv python=3.8
```

Step 3: Activate the Environment

Activate the newly created environment:

```
conda activate myenv
```

Step 4: Install Required Packages

Install Flask and any other necessary packages using conda or pip. If you know the specific packages and versions your project requires, install them directly.

You can also use pip to install packages from a requirements.txt file if your project includes one:

```
pip install -r requirements.txt
```

Step 5: Download and Extract Project

Download your project zip file to a known directory.

Extract the zip file in an appropriate working directory.

Step 6: Navigate to the Project Directory

In the Anaconda Prompt, change the directory to where your project files are located:

```
cd path\to\your\project
```

Step 7: Run the Flask Application

Run your Flask application:

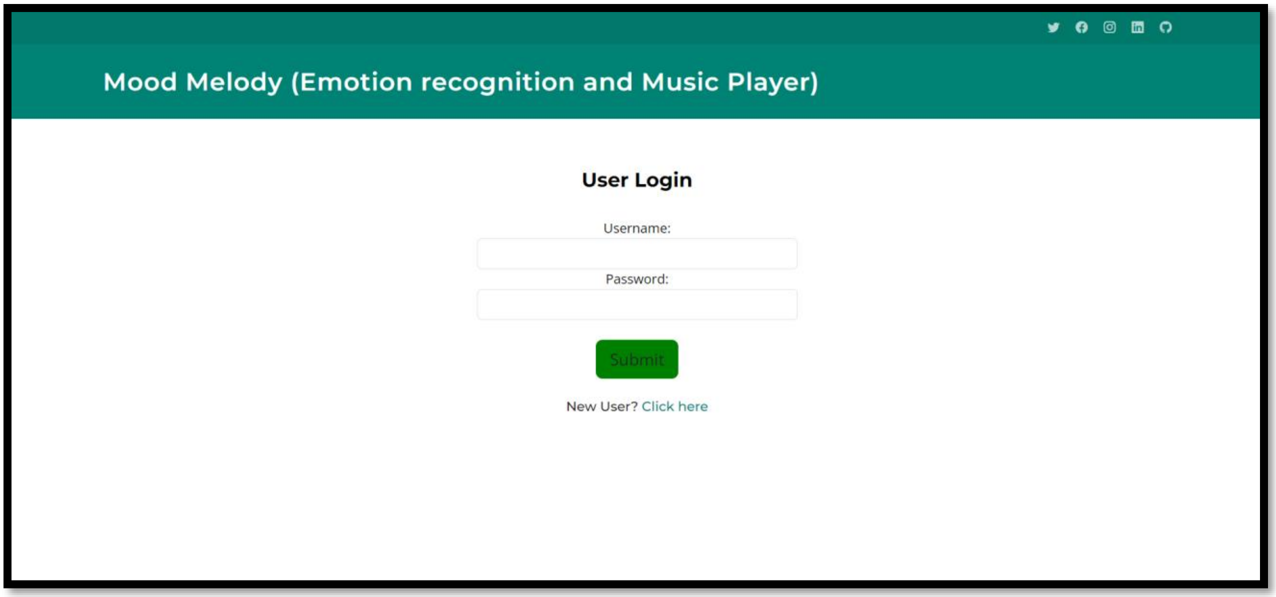
Step 8: Access the Application

Open a web browser and go to http://127.0.0.1:5000 to see your application in action.

USER MANUAL

1. System Home Page

When a user tries to access the website it navigates to the login page.

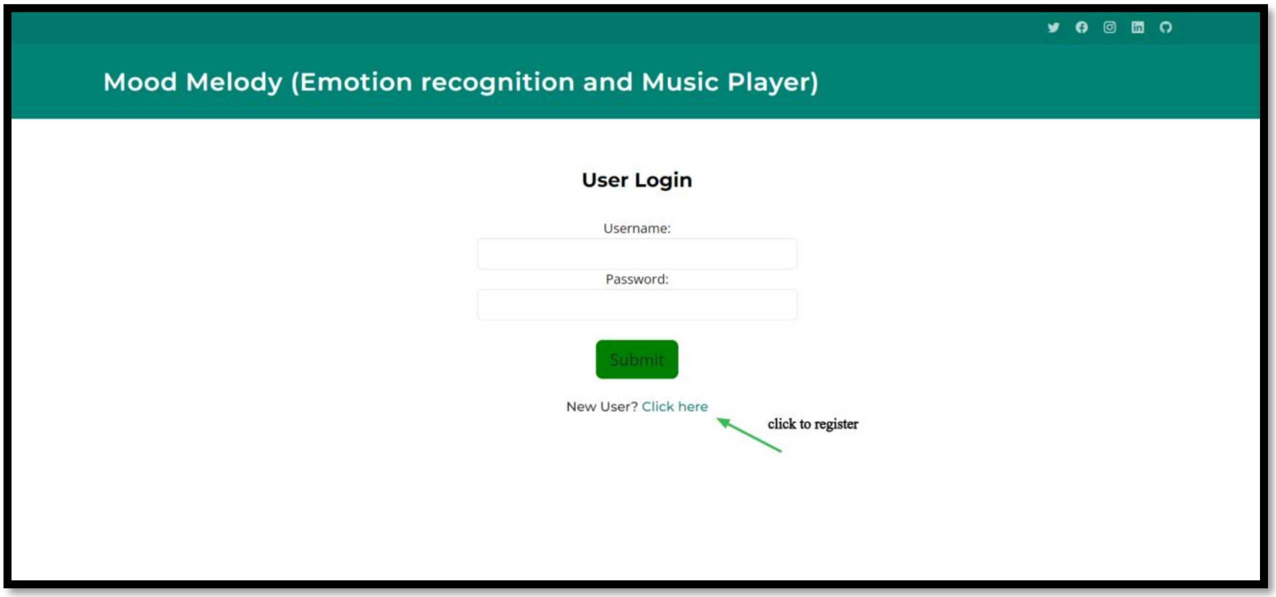


Mood Melody (Emotion recognition and Music Player)

User Login

2. User Login

User can login with credentials if existing or else can redirect to registration page.

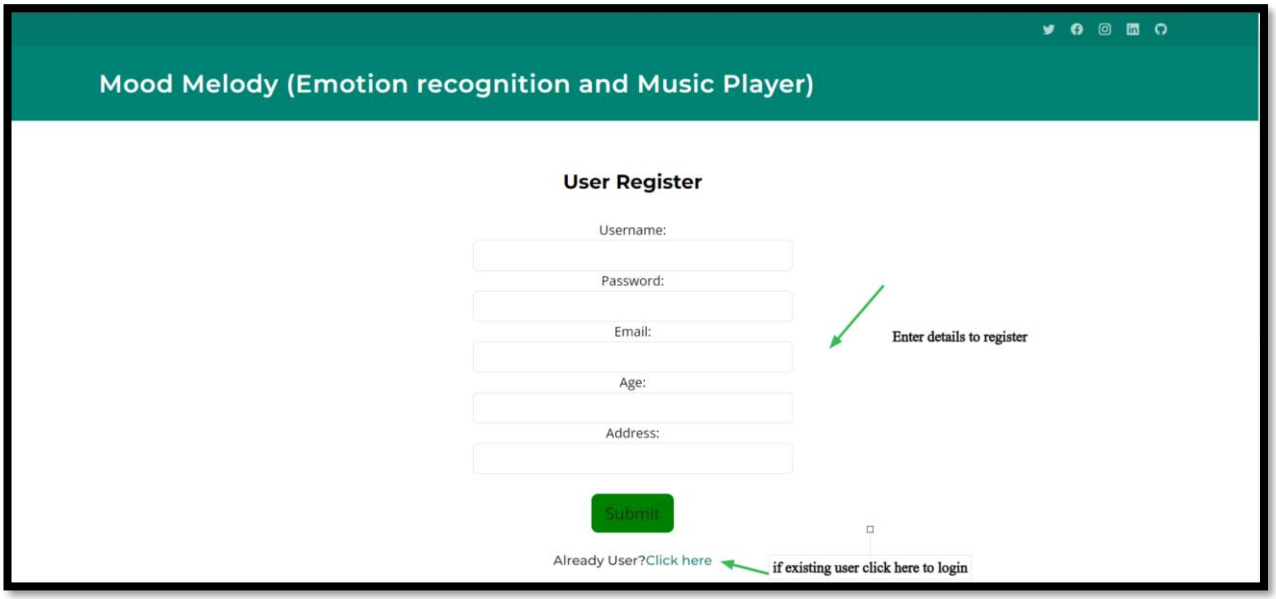


Mood Melody (Emotion recognition and Music Player)

User Login

3. User Registration

User can register with personal details or else if existing can redirect to login page.

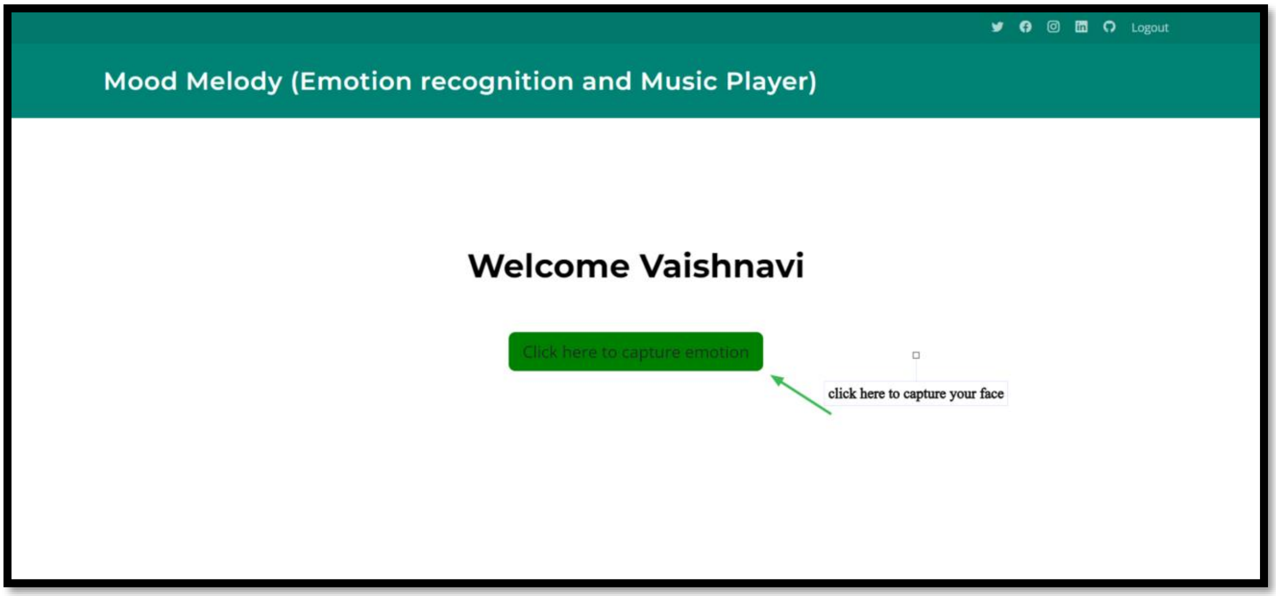


Enter details to register

if existing user click here to login

4. User Emotion Capture

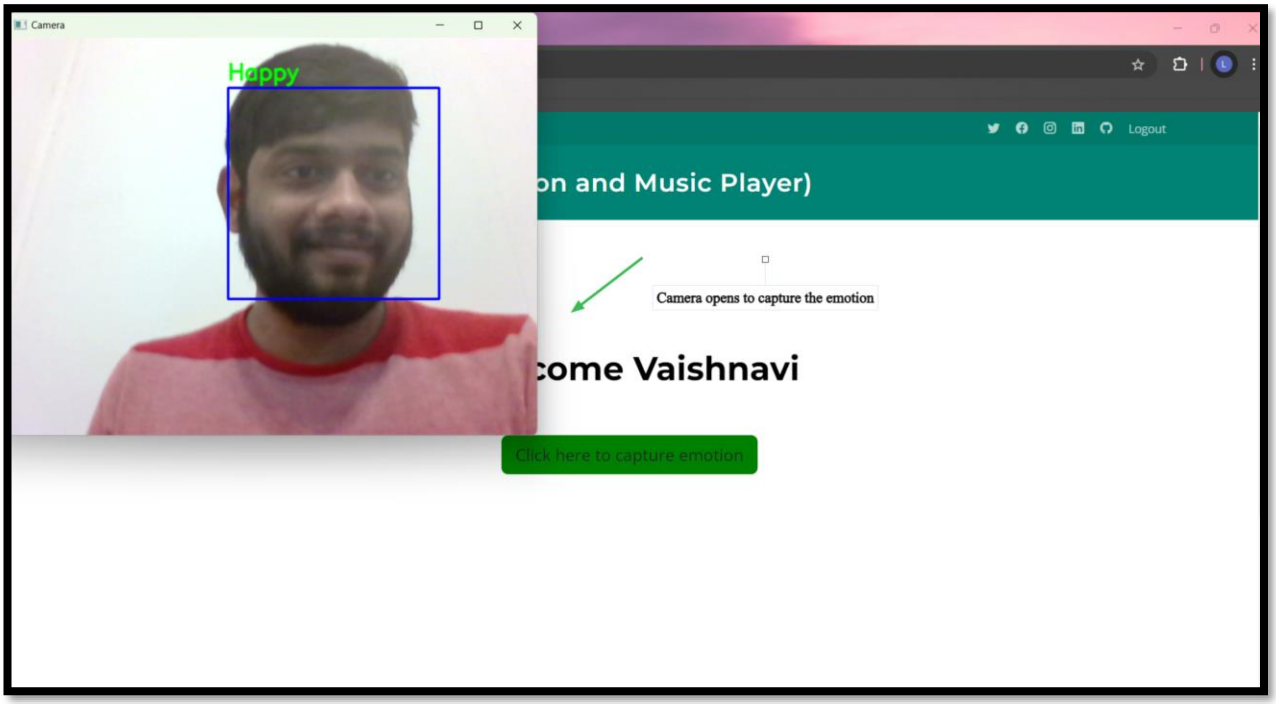
After login user is redirected to capture emotion page, where they can click the button to access camera.


Logout
Mood Melody (Emotion recognition and Music Player)
Welcome Vaishnavi
Click here to capture emotion

click here to capture your face

5. Camera

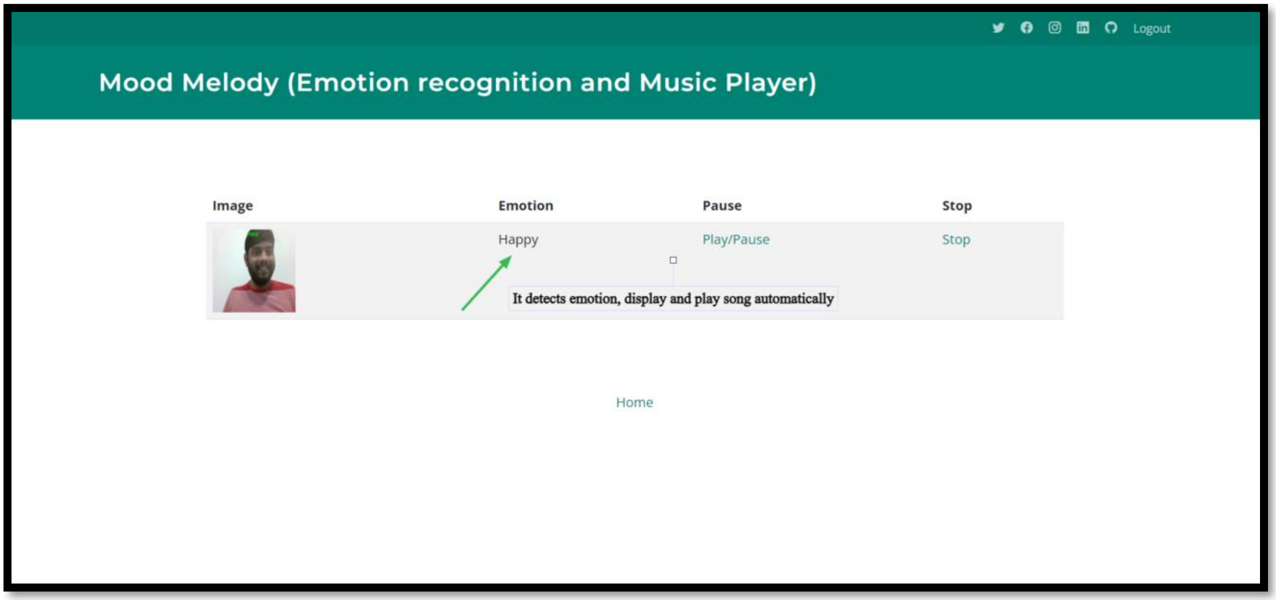
After clicking button, the camera pops up and captures the emotion to freeze press Q.



Camera opens to capture the emotion

6. Emotion Detection and Music Player

It displays the emotion and plays emotion related music automatically.



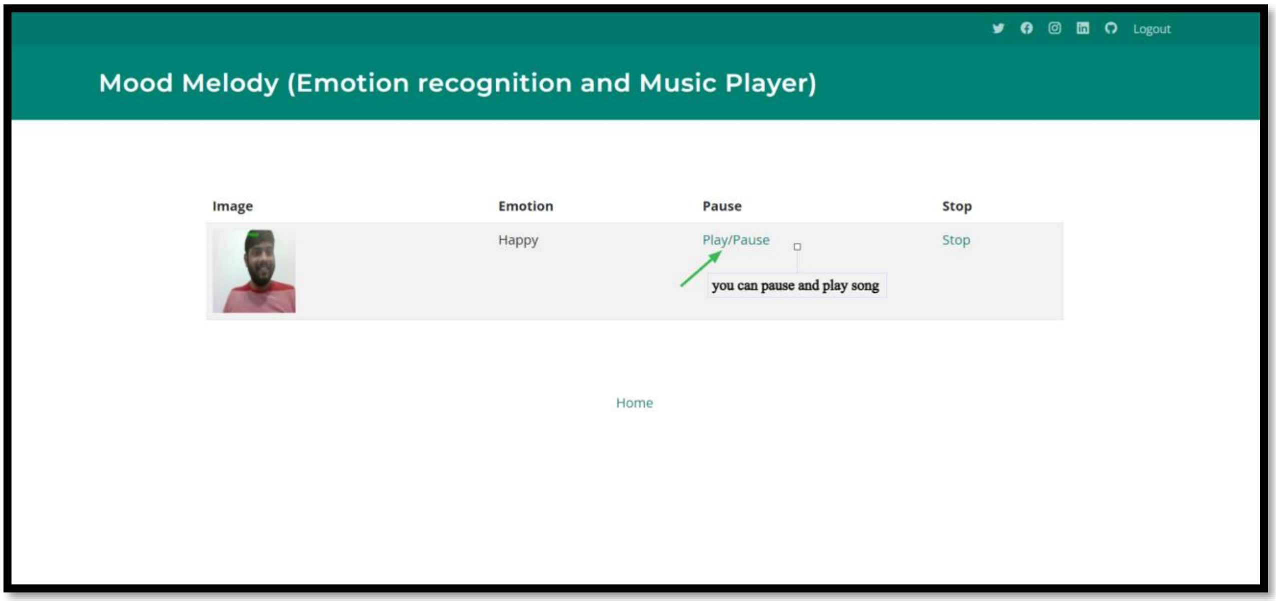
Mood Melody (Emotion recognition and Music Player)

Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop

It detects emotion, display and play song automatically

7. Pause/Play

Here the user can pause or play the song.



Mood Melody (Emotion recognition and Music Player)

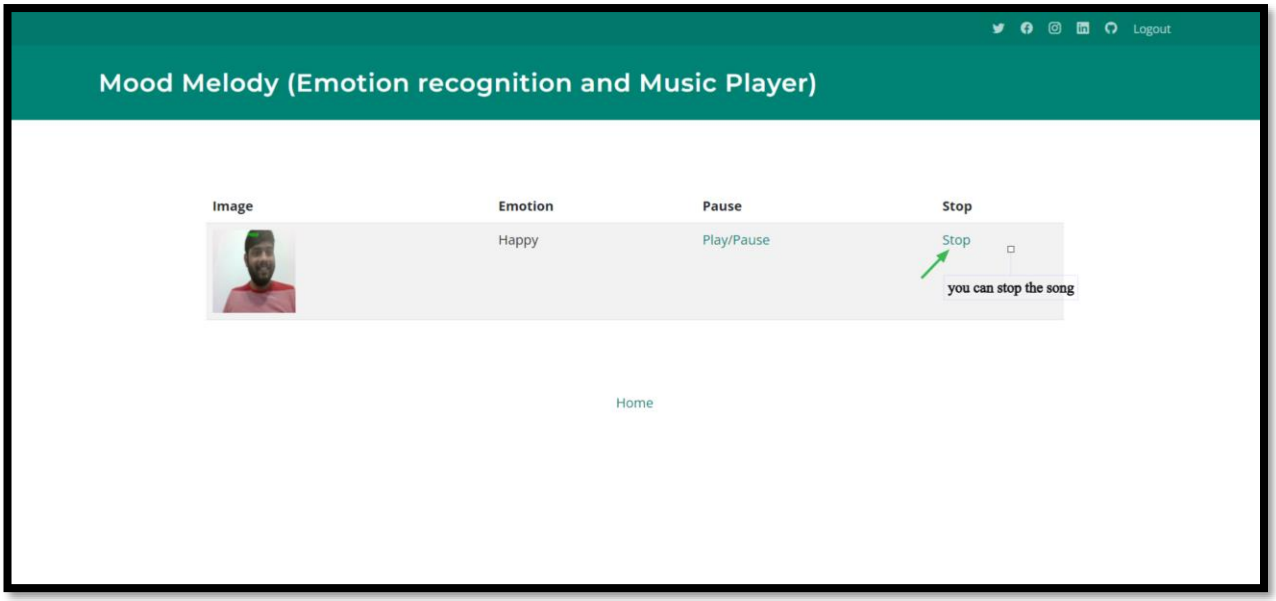
Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop

you can pause and play song

Home

8. Stop Music

User can Stop the song.



Mood Melody (Emotion recognition and Music Player)

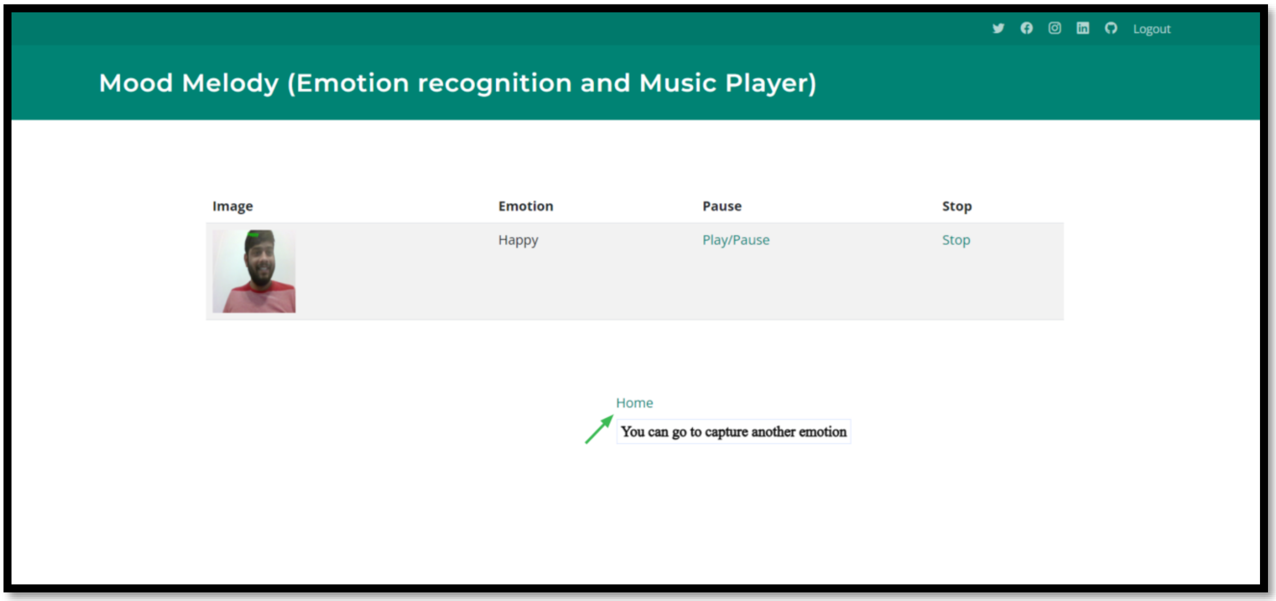
Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop

you can stop the song

Home

9. Home

User can click home to capture another emotion



Logout

Mood Melody (Emotion recognition and Music Player)

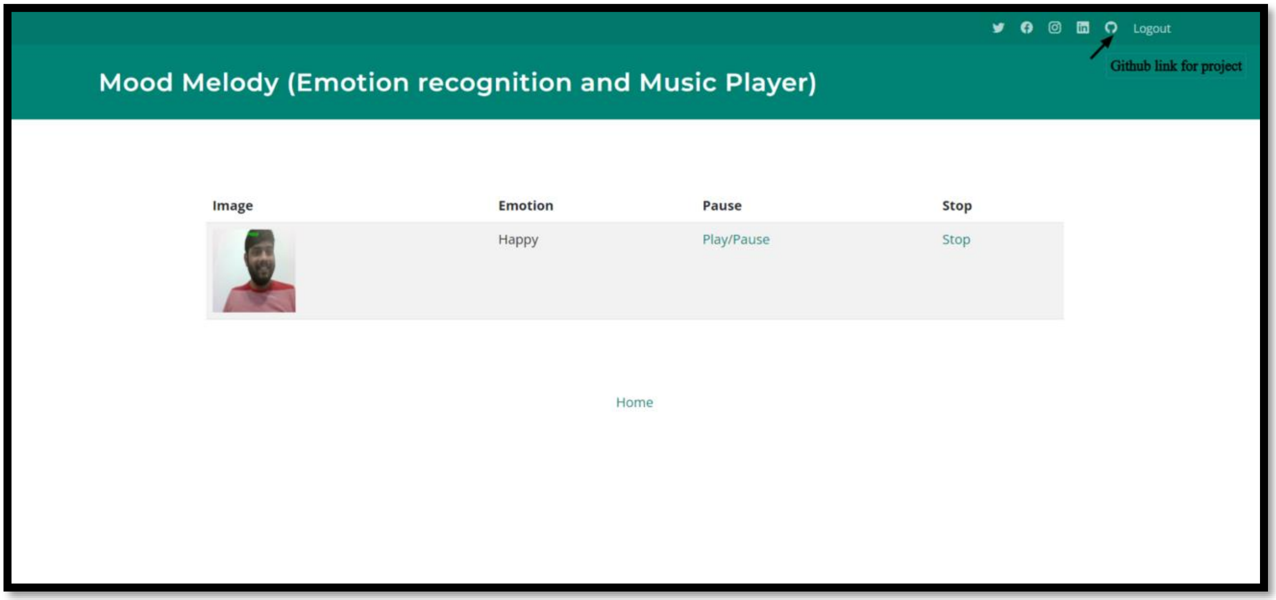
Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop

Home

You can go to capture another emotion

10. Github Link

If User wants to access the user manual or projects details he can redirect it through the github link.



Logout

Github link for project

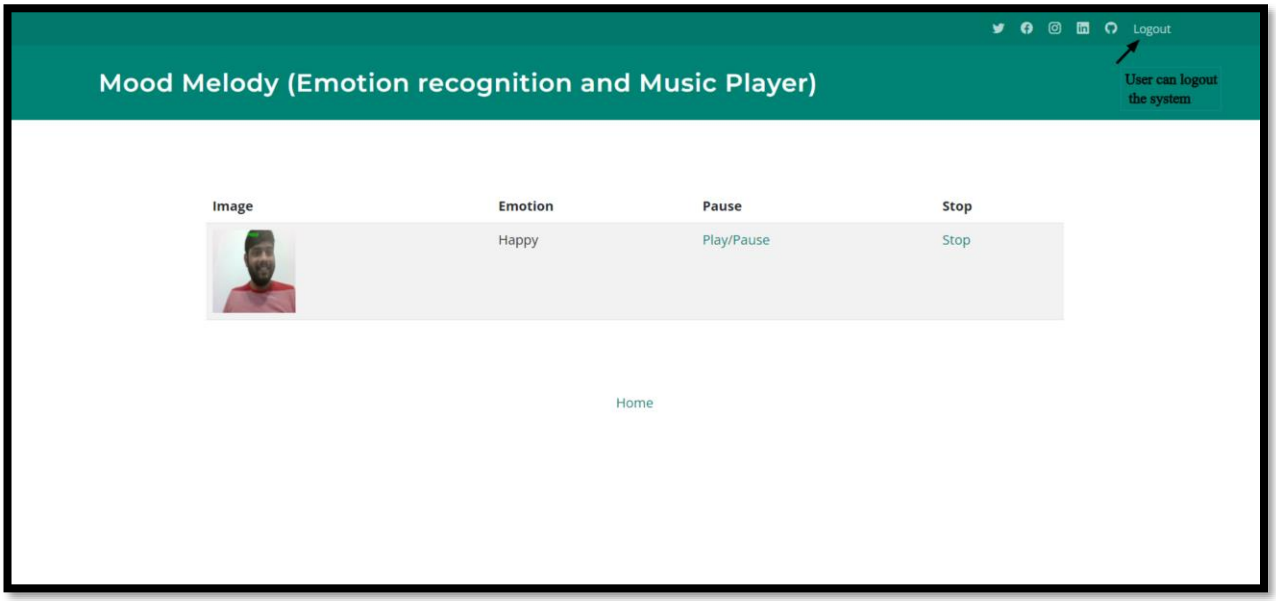
Mood Melody (Emotion recognition and Music Player)

Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop

Home

11. User Logout

The user can logout the system.



Mood Melody (Emotion recognition and Music Player)

User can logout the system

Image	Emotion	Pause	Stop
	Happy	Play/Pause	Stop