MUSIC RECOMMENDATION BASED ON FACE EMOTION RECOGNITION

A PROJECT REPORT

*Submitted By*

HUMAYOON NIYAZ (200071601043)

FAWWAZ NUMAN (200071601035)

Under the guidance of

Dr. I. KARTHIGA

*in partial fulfillment for the award of the degree of*

BACHELOR OF TECHNOLOGY

*in*

COMPUTER SCIENCE AND ENGINEERING



**MAY 2024**



# BONAFIDE CERTIFICATE

Certified that this project report “**MUSIC RECOMMENDATION BASED ON FACE EMOTION RECOGNITION**” is the bonafide work of “**HUMAYOON NIYAZ (200071601043) and FAWWAZ NUMAN (200071601035)**” who carried out the project work under my supervision. Certified further, that to the best of our knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| *SIGNATURE* | *SIGNATURE* |
| **Dr. I. KARTHIGA**  **SUPERVISOR** | **Dr.** **W.** **AISHA BANU**  **HEAD OF THE DEPARTMENT** |
| Assistant Professor | Professor & Head |
| Department of CSE | Department of CSE |
| B. S. Abdur Rahman Crescent | B. S. Abdur Rahman Crescent |
| Institute of Science and Technology | Institute of Science and Technology |
| Vandalur, Chennai - 600048 | Vandalur, Chennai - 600048 |



# VIVA VOCE EXAMINATION

The viva voce examination of **CSC 4211 – Project work** titled **“MUSIC RECOMMENDATION BASED ON FACE EMOTION RECOGNITION**”**,** submitted by **HUMAYOON NIYAZ (200071601043)** and **FAWWAZ NUMAN (200071601035)** is held on \_\_\_\_\_\_\_\_\_\_\_.

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We sincerely express our heartfelt gratitude to Prof. **Dr. T. MURUGESAN**, Vice chancellor and **Dr. N. THAJUDDIN**, Pro-Vice Chancellor, B.S. Abdur Rahman Crescent Institute of Science and Technology, for providing us an environment to carry out our course successfully.

We sincerely thank **Dr. N. RAJA HUSSIAN**, Registrar for furnishing every essential facility for doing our project.

We thank **Dr. SHARMILA SANKAR**, Dean, School of Computer, Information and Mathematical Sciences for her motivation and support.

We thank **Dr. W. AISHA BANU,** Professor and Head, Department of Computer Science and Engineering, for providing strong oversight of vision, strategic direction and valuable suggestions.

We express our sincere thanks to the Project Review Committee members, from the Department of Computer Science and Engineering, **Dr. E.SYED MOHAMED**, Professor and **Dr. S. SHARON PRIYA,** Associate Professor and **Dr. I. KARTHIGA,** Assistant Professor for their valuable suggestions and support.

We obliged our project supervisor **Dr. I. KARTHIGA,** Assistant Professor, Department of Computer Science and Engineering for her professional guidance and continued assistance during our project.

We thank our class advisor, **Dr. I. KARTHIGA,** Assistant Professor, Department of Computer Science and Engineering for her guidance and encouragement throughout our project.

We thank all the **faculty members** and the **System staff** of the Department of Computer Science and Engineering for their valuable support and assistance at various stages of project development.

**(HUMAYOON NIYAZ)**

**(FAWWAZ NUMAN)**

# ABSTRACT

Leveraging deep learning techniques, the system analyzes users' real-time facial expressions to infer emotional states, thereby refining music recommendations to align with their mood preferences. By dynamically adapting recommendations based on detected emotions, this approach aims to provide more personalized and relevant music suggestions. Through extensive user studies, the effectiveness and user acceptance of FER-based recommendations are evaluated, considering metrics such as recommendation accuracy, user satisfaction, and perceived relevance. Ethical considerations including privacy and data security are addressed to ensure responsible implementation. The findings contribute to advancing personalized music discovery, catering to users' emotional needs with enhanced precision and relevance. The findings of this research not only advance personalized music discovery but also pave the way for innovations in technology. The integration of facial emotion recognition technology into music recommendation systems opens up exciting possibilities beyond just music discovery. This approach could be extended to other forms of media and entertainment, such as movie, TV show, or book recommendations, where understanding the user's emotional state could greatly enhance the relevance of suggestions. The real-time analysis of facial expressions could offer valuable insights into user engagement and satisfaction, enabling continuous optimization and refinement of recommendation algorithms.

# TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | **ABSTRACT** | **v** |
| **LIST OF FIGURES** | **viii** |
| **LIST OF ABBREVIATIONS** | **ix** |
| **LIST OF TABLES** | **ix** |
| **1** |  |  | **INTRODUCTION** | **1** |
|  | 1.1 |  | OVERVIEW | 1 |
|  | 1.2 |  | DESCRIPTION | 2 |
|  | 1.3 |  | OBJECTIVE | 3 |
|  | 1.4 |  | ORGANIZATION OF THE REPORT | 4 |
| **2** |  |  | **LITERATURE SURVEY** | **7** |
| **3** |  |  | **PROBLEM DEFINITION AND**  **METHODOLOGY** | **10** |
|  | 3.1 |  | PROBLEM DEFINITION | 10 |
|  | 3.2 |  | EXISTING SYSTEM | 10 |
|  | 3.3 |  | PROPOSED SYSTEM | 11 |
|  | 3.4 |  | ALGORITHM | 12 |
| **4** |  |  | **DESIGN PROCESS** | **19** |
|  | 4.1 |  | DESIGN OVERVIEW | 19 |
|  | 4.2 |  | DATAFLOW DIAGRAM | 20 |
|  | 4.3 |  | SYSTEM REQUIREMENTS | 22 |
|  |  | 4.3.1 | SOFTWARE REQUIREMENTS | 22 |
|  |  | 4.3.2 | HARDWARE REQUIREMENTS | 22 |
|  | 4.4 |  | SOFTWARE DESCRIPTION | 22 |
|  |  | 4.4.1 | OPERATING SYSTEM | 23 |
|  | 4.5 |  | PROGRAMMING LANGUAGE | 24 |
|  |  | 4.5.1 | TEXT EDITOR | 24 |
|  |  | 4.5.2 | WEB BROWSER | 25 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 4.5.3 | VERSION CONTROL | 25 |
|  | 4.6 |  | MODULE DESCRIPTION | 26 |
|  |  | 4.6.1 | DATA COLLECTION AND LABELING | 26 |
|  |  | 4.6.2 | FACIAL EMOTION RECOGNITION AND TRAINING | 26 |
|  |  | 4.6.3 | INTEGRATION AND MUSIC PREFERENCES USING API | 27 |
|  |  | 4.6.4 | FINE TUNING AND OPTIMIZATION | 27 |
|  |  | 4.6.5 | UI AND FACIAL EMOTION CAPTURE | 28 |
|  |  | 4.6.6 | MUSIC RECOMMENDATION | 28 |
|  | 4.7 |  | DEVELOPING APPLICATION | 29 |
| **5** |  |  | **IMPLEMENTATION** | **32** |
|  | 5.1 |  | UML DIAGRAM | 32 |
|  | 5.2 |  | MICROSERVICES IMPLEMENTATION | 33 |
|  |  | 5.2.1 | BACKEND IMPLEMENTATION | 33 |
|  |  | 5.2.2 | FACIAL EMOTION RECOGNITION MODULE | 34 |
|  |  | 5.2.3 | INTEGRATION OF APIs | 35 |
|  |  | 5.2.4 | MUSIC RECOMMENDATION | 36 |
|  |  | 5.2.5 | UI DESIGN | 36 |
|  |  | 5.2.6 | TESTING AND DEPLOYEMENT | 37 |
| **6** |  |  | **RESULTS AND ANALYSIS** | **38** |
|  | 6.1 |  | RESULT | 38 |
|  | 6.2 |  | ANALYSIS | 41 |
| **7** |  |  | **CONCLUSION AND FUTURE**  **ENHANCEMENT** | **44** |
|  | 7.1 |  | CONCLUSION | 44 |
|  | 7.2 |  | FUTURE ENHANCEMENTS | 44 |
| **8** |  |  | **REFERENCES** | **46** |
| **9** |  |  | **APPENDIX** | **47** |
|  | 9.1 |  | A1 - SOURCE CODE | 47 |
|  | 9.2 |  | A2 - SCREENSHOT | 66 |
| **10** |  |  | **TECHNICAL BIOGRAPHY** | **69** |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 3.1 | Convolutional Neural Network (CNN) | 13 |
| 4.1 | Architecture Diagram | 19 |
| 4.2 | Dataflow Diagram | 20 |
| 4.3 | Flowchart Level-2 | 21 |
| 4.4 | Prototype of Website | 30 |
| 4.5 | Emotion Detection and Recommendation | 30 |
| 4.6 | Emotion Detection and Recommendation II | 31 |
| 4.7 | Contact Us Feedback | 31 |
| 5.1 | UML Diagram | 32 |
| 6.1 | Emotion Detection | 38 |
| 6.2 | Music Recommendation | 39 |
| 6.3 | User Interface | 40 |
| 6.4 | Music Platforms | 41 |
| 6.5 | User-Feedback | 41 |
| 8.1 | Home Page | 66 |
| 8.2 | Happy Emotion Music Recommendation | 67 |
| 8.3 | Sad Emotion Music Recommendation | 67 |
| 8.4 | Feedback | 68 |
| 8.5 | Team | 68 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| CNN | Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| API | Application programming interface |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| 4.1 | Software Requirements | 22 |
| 4.2 | Hardware Requirements | 22 |

**CHAPTER 1**

**INTRODUCTION**

When it comes to digital music consumption, finding tailored suggestions that suit each user's tastes is still critical. Conventional approaches frequently depend on past user data, but they could miss the dynamic interaction between emotion and music. By incorporating face emotion detection technology into music recommendation systems, this initiative fills this gap in the market. Our algorithm attempts to interpret users' emotional states from their real-time facial expression analysis and adjust music recommendations accordingly. The combination of state-of-the-art computer vision methods and music recommendation algorithms has great potential to provide experiences that are contextually relevant and really individualized. We want to investigate not only the technical details of this project but also the moral issues related to algorithmic biases and data privacy. Through this project, we envision reshaping the landscape of music discovery by providing users with recommendations that not only match their musical tastes but also resonate with their current emotional context.

**1.1 OVERVIEW**

In a world where music is a universal language, the quest to revolutionize the way we discover and engage with our favorite tunes has reached new frontiers. Embracing the synergy of cutting-edge technology and the profound emotional connection we share with music; this project introduces a groundbreaking approach to music recommendation – one guided by the subtle nuances of facial expressions. A music recommendation system that not only understands your musical preferences but also taps into the emotions evoked by each melody. This project ventures into uncharted territory, leveraging facial emotion detection technology to redefine the way we curate and experience music playlists. By analyzing users' facial expressions during their interaction with diverse music genres, our system deciphers a spectrum of emotional responses – from the exhilaration of an upbeat rhythm to the introspection inspired by a soulful ballad.

This innovative endeavor comprises a sophisticated approach that captures and interprets the user's emotional journey through music. The accompanying intuitive dashboard offers users a visual representation of their emotional states, fostering a deeper connection with their musical preferences. It goes beyond conventional algorithms, presenting a more personalized and emotionally resonant musical journey.

The fusion of facial emotion detection and music recommendation holds the promise of transforming how we explore, connect with, and savor our favorite tunes. In this era of technological marvels, our project pioneers a path towards a more emotive and immersive music discovery experience, where the beats resonate not just in our ears but also in the expressions that light up our faces. Welcome to the future of music recommendation – where emotion is the key to unlocking the perfect playlist.

**1.2 DESCRIPTION**

Reshaping the music recommendation landscape through an inventive system that tailors playlists based on users' emotional responses. By analyzing facial expressions during interactions with various music genres, the system captures emotional nuances, enhancing the personalized and immersive nature of music recommendations. The intuitive interface visually represents emotional states, creating a deeper connection with users' musical experiences. This project seeks to redefine music recommendation, emphasizing emotion as a key factor in curating the perfect playlist and offering a unique and engaging way to explore and connect with favorite tunes. The implications of our technology extend far beyond individual listening experiences. With its adaptability and versatility, our system can be seamlessly integrated into various applications and environments. For instance, in automotive entertainment systems, it enhances driving experiences by selecting music that aligns with the driver's emotional state, promoting relaxation or energizing focus as needed. Similarly, in social settings such as party halls or event venues, our system elevates atmospheres by curating playlists that resonate with the collective mood of attendees, fostering engagement and enjoyment.

Our system provides user-friendly interfaces accessible to all, regardless of technical expertise. The customizable design ensures adaptability to different institutional needs, promoting transparency, accountability, and evidence-based decision-making. This comprehensive solution not only seeks to improve the evaluation of faculty performance but also to empower faculty members in tracking and enhancing their professional development. By facilitating data-driven decisions at both the individual and administrative level

**1.3 OBJECTIVE**

The central aim of this pioneering initiative is to seamlessly fuse facial emotion recognition technology with a dynamic music recommendation framework. The user interface boasts an engaging setup where, on the left-hand side, users can observe their real-time facial expressions, complemented by precise emotion detection. Concurrently, on the right-hand side, a music dashboard dynamically crafts tailored recommendations based on the identified emotions. This design is meticulously crafted to streamline the music exploration process, empowering users to forge deep connections between their emotional states and musical inclinations. Emphasizing user satisfaction, the system offers a customizable and adaptable platform, allowing institutions to tailor it to their specific requirements. Through this endeavor, we aspire to revolutionize the traditional music discovery journey, delivering a distinctive and immersive experience.

Expanding upon this vision, the project envisions a multifaceted impact across various domains. Firstly, by leveraging facial emotion recognition, we aim to enhance the accuracy and relevance of music recommendations, thereby enriching the user experience. This personalized approach not only fosters greater engagement but also cultivates a sense of resonance and emotional connection with the music selected. Moreover, the project underscores the significance of adaptability, recognizing that user preferences and emotional states can vary widely across contexts and individuals. As such, the system is designed to continually adapt and refine its recommendations, ensuring that they remain attuned to the evolving needs and preferences of users.

Furthermore, beyond its immediate applications in entertainment and leisure, the project holds potential implications for broader fields such as mental health and well-being. By facilitating a deeper understanding of the interplay between music and emotions, the system could be harnessed as a tool for emotional regulation and mood enhancement. For instance, individuals grappling with stress or anxiety could benefit from curated music selections tailored to promote relaxation and emotional balance. Additionally, the project underscores the importance of ethical considerations, particularly concerning data privacy and user consent. Robust measures are implemented to safeguard user data and ensure transparent practices throughout the system's deployment.

**1.4 ORGANIZATION OF THE REPORT**

**Chapter 1: Introduction**

The inaugural chapter lays the foundation for the project, elucidating its core objectives and situating its importance within the landscape of personalized music recommendation systems. It offers a panoramic view of the project's innovative approach, spotlighting the fusion of individualized music preferences with the emotional dimensions of music engagement. Through this lens, the chapter sets the stage for an exploration into the dynamic intersection of technology and human emotion, underscoring the project's overarching aim to revolutionize the music discovery experience.

**Chapter 2: Literature Review**

The literature review examines existing studies and resources related to music recommendation systems, emphasizing the role of emotion in user experiences. This chapter explores prior research on personalized music recommendations and the impact of emotional factors on user satisfaction within the broader context of recommendation systems.

**Chapter 3: Problem Definition and Methodology**

This chapter delves into the precise problem statement driving the project and elucidates the methodology utilized to engineer a music recommendation system centered around user needs. It delineates the rationale underpinning the selected methodology, highlighting its suitability for addressing the identified challenges and achieving project objectives. Through a systematic exploration of the problem and iterative development processes, it provides insights into the rigorous approach adopted for problem-solving and system refinement. Additionally, it underscores the significance of user feedback and empirical validation in shaping the evolution of the recommendation system, ensuring its efficacy and relevance in real-world contexts.

**Chapter 4: Design Process**

Within this section, an exhaustive account of the design journey unfolds, illuminating the intricacies of crafting a personalized music recommendation system. Each step of the process is meticulously documented, from conceptualization to implementation, underscoring the system's tailored adaptation to cater to the diverse tastes of music aficionados. Central to the narrative is an unwavering commitment to prioritizing user experience, with every design decision informed by the goal of fostering emotional resonance and connection. The section delves into the considerations involved in interface design, content curation, and recommendation algorithms, highlighting the iterative refinement process guided by user feedback and usability testing. Through this holistic approach to design, the chapter encapsulates the essence of creating a music recommendation system that transcends mere functionality, offering a deeply enriching and emotionally resonant experience for users.

**Chapter 5: Implementation**

This chapter provides a comprehensive overview of the project's implementation journey, chronicling the development stages, encountered challenges, and innovative solutions employed. From inception to deployment, each phase of implementation is meticulously dissected, offering insights into the technical intricacies and decision-making processes that shaped the system's realization. The narrative navigates through the development lifecycle, from system architecture design and algorithm development to integration of facial emotion recognition technology and music recommendation algorithms. Alongside the triumphs, the chapter candidly addresses the challenges encountered during implementation, ranging from technical hurdles to algorithmic complexities and data integration issues. Through a candid exploration of these challenges, readers gain a nuanced understanding of the iterative problem-solving approach adopted, with each obstacle serving as a catalyst for innovation and refinement. Ultimately, the chapter serves as a testament to the project team's ingenuity and perseverance, culminating in the successful realization of a sophisticated and user-centric music recommendation system.

**Chapter 6: Results and Analysis**

In this chapter, the project outcomes are meticulously dissected, showcasing the data-driven insights gleaned from the personalized music recommendation system. Through rigorous analysis of generated data, key trends and user feedback are encapsulated, offering a comprehensive understanding of the system's efficacy in delivering bespoke music recommendations rooted in user preferences and emotional resonance. Additionally, the chapter explores the implications of the findings and identifies areas for further refinement and enhancement, laying the groundwork for future iterations and advancements in the field of personalized music recommendation systems.

**Chapter 7: Conclusion and Future Enhancement**

The concluding chapter encapsulates the main findings of the project, offering a synthesis of key insights and implications drawn from the personalized music recommendation system. Through a comprehensive summary, the chapter underscores the significance of the results in advancing the field of music recommendation systems and enhancing user experiences in music discovery. Additionally, it outlines recommendations for future research and development, highlighting avenues for further innovation and refinement in personalized music recommendation algorithms and user interface design. As the project journey culminates, the chapter emphasizes the ongoing evolution of user-centric experiences in music discovery and the vast potential for advancements in tailoring music recommendations to individual preferences and emotional resonance.

**CHAPTER 2**

**LITERATURE SURVEY**

Florence, S. Metilda, and M. Uma [1] propose the development of an emotional detection and music recommendation system that utilizes user facial expressions. They delve into the exploration of different facial emotion recognition techniques and their potential integration into personalized music recommendation platforms. Their research emphasizes the significance of incorporating emotional cues extracted from facial expressions to augment the precision and appropriateness of music suggestions. Furthermore, they suggest avenues for enhancing the effectiveness of such systems through continued investigation and refinement of facial emotion recognition methodologies.

Wang, Shu, et al. introduced a novel approach to music recommendation [2] that incorporates facial emotion recognition through deep neural networks. They shift the focus of their research towards investigating the user experience and acceptance of such emotion-driven music recommendation systems. Specifically, they analyze factors like system intuitiveness, user satisfaction, and the influence of emotion-based recommendations on the overall enjoyment of music. Additionally, they advocate for a deeper understanding of user perceptions and preferences to further refine the effectiveness and user-friendliness of these systems.

Using deep learning, [3] Mahadik, Ankita, et al. presented a mood-based music recommendation system. Their research explores how to incorporate facial emotion recognition into deep neural networks to improve the system's capacity to suggest emotionally relatable music to users. The goal of the research is to increase the relevance and accuracy of music recommendations based on users' emotional states by concentrating on the application of deep learning techniques. The focus is on using facial emotion recognition as a keystone for deep neural network training to make better music recommendations.

Iyer, Aurobind V., et al. [4] presented a study focused on emotion-based mood-enhancing music recommendation at the 2017 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). Their research aims to provide a thorough review of literature pertaining to facial emotion recognition techniques for music recommendation systems. By synthesizing existing studies, the review offers insights into the current state of research in this domain, highlighting trends, challenges, and potential areas for further exploration.

Sana, S. K., et al. present a study that uses convolutional neural networks to recognise facial emotions in a music system, which is published in Materials Today: Proceedings in 2022. [5] Their work focuses on the integration of dynamic music recommendation systems with real-time facial emotion recognition. The study aims to improve the responsiveness and adaptability of music recommendations to users' fluctuating emotional states by giving priority to real-time processing. This study adds to the continuing investigation into how to use facial emotion recognition to create more dynamic and customised music experiences.

Kim, Hyoung-Gook, Gi Yong Lee, and Min-Soo Kim [6] explored a dual-function integrated emotion-based music classification system in their study published in the IEEE Transactions on Consumer Electronics in 2021. They propose enhancing music recommendation diversity through facial emotion recognition, aiming to introduce a broader range of music options aligned with users' emotional states. By discussing strategies for leveraging facial emotion recognition, the research aims to enrich the music listening experience by providing a more varied and satisfying selection of music tailored to users' emotional needs and preferences.

Anisha Prajapati, Sandeep Maurya, Kale, and Yash [7] reviewed face-expression-based music recommendations in i-manager's Journal on Image Processing. They explore privacy issues related to the use of facial emotion recognition in music recommendation systems in their paper. The research attempts to shed light on the possible privacy implications associated with these systems by addressing user concerns and ethical issues. Additionally, it talks about ways to reduce privacy risks and make sure that facial emotion recognition technologies are used responsibly when recommending music.

Assuncao, Willian G., Lara SG Piccolo, and Luciana AM Zaina [8] undertook a systematic literature review published in Multimedia Tools and ApplicationsTheir study delves into the cross-cultural considerations of facial emotion recognition and its implications for global music recommendations. By exploring the challenges and opportunities inherent in developing culturally sensitive systems, the research aims to foster a deeper understanding of how diverse emotional expressions manifest across different cultures. This examination provides insights into the complexities of cross-cultural communication and highlights the importance of cultural context in the design and implementation of facial emotion recognition technologies for music recommendation systems.

Cheng, Y., Zhang, H., & Liu, J. [9] investigated user input and interaction with applications that incorporate facial emotion detection for music suggestions. Their study illuminates user reactions, behaviour patterns, and the general effectiveness of emotion-infused music recommendation apps. The study aims to evaluate the efficacy and user satisfaction with these applications through a close examination of user feedback and engagement metrics. Future developments in this field will be informed by this analysis, which advances our knowledge of how users perceive and engage with music recommendation systems enhanced by facial emotion recognition.

Kale, Yash, Sandeep Maurya, and Anisha Prajapati [10] presented a paper titled "Machine Learning Approaches for Facial Emotion Recognition in Music Recommendation" in i-manager's Journal on Image Processing (2022, [10]). The paper delves into the exploration of diverse machine learning approaches utilized in facial emotion recognition to improve music recommendations. By discussing the strengths and limitations of various algorithms in capturing and interpreting facial expressions, the research aims to provide insights into the effectiveness of different methodologies. This examination contributes to a deeper understanding of how machine learning techniques can be leveraged to enhance the accuracy and relevance of facial emotion recognition in the context of music recommendation systems.

Xu, Y., Zhang, Z., & Liu, Y. [11] compared several music recommender systems that rely on facial emotion recognition. Their study assesses these systems' precision, effectiveness, and user satisfaction in order to clarify their respective advantages and disadvantages. This work clarifies the effectiveness and performance of various methods, offering insightful information about the state of face emotion recognition-based music recommendation systems.

**CHAPTER 3**

**PROBLEM DEFINITION AND METHODOLOGIES**

**3.1 PROBLEM DEFINITION**

The problem is to develop a system that can detect real-time emotions from user's facial expressions and recommend music that aligns with their emotional state, thus enhancing the overall user experience and engagement with music. The core functionality of the system lies in its ability to capture and interpret facial expressions in real-time. Leveraging computer vision techniques, the system analyzes facial features, recognizing subtle nuances indicative of emotions such as joy, sadness, anger, or surprise. This real-time data lays the foundation for dynamically adjusting music recommendations, ensuring a seamless alignment with the user's evolving emotional landscape.

**3.2 EXISTING SYSTEM**

In previous systems users used to receive music recommendations that are not influenced by real-time emotional fluctuations. Instead, the algorithm focuses on curating a playlist based on the user's cumulative search history, offering a seamless and personalized experience. By adopting this model, we aim to cater to a diverse range of preferences and circumstances, providing users with a music discovery journey that is both reliable and reflective of their individual tastes. The underlying concept is to empower users to explore and enjoy music based on their established preferences, fostering a connection with their own musical journey without the interference of emotional nuances. This reimagined approach aligns with our commitment to delivering a user-centric and adaptive music recommendation platform, ensuring that each user's experience is tailored to their unique musical identity and evolving tastes.

**Limited Personalization:** Manual recommendations often rely on broad genres, popular trends, or general user preferences. This approach may lead to a lack of personalized recommendations tailored to the user's specific emotional states, resulting in a less engaging and resonant music experience.

**Inability to Capture Real-Time Emotions:** The absence of facial emotion recognition means that the system cannot capture user's real-time emotional states. Music preferences can change based on immediate feelings or moods, and manual recommendations may not adapt dynamically to these fluctuations.

**Missed Emotional Nuances:** Manual curation might struggle to capture subtle emotional nuances that influence music preferences. Facial emotion recognition can provide a more nuanced understanding of the user's emotional state, allowing for recommendations that align more closely with their current feelings.

**Limited Diversity in Recommendations:** Without the integration of facial emotion recognition, the system may struggle to diversify recommendations based on the user's emotional range. This limitation could lead to a repetitive selection of music that may not cater to the full spectrum of the user's emotions and preferences.

Dependency on Explicit User Input: Manual recommendation systems often rely on explicit user input, such as liking or disliking songs. This dependence on explicit feedback may result in a slower adaptation to the user's evolving preferences and emotional states compared to real-time, implicit feedback captured through facial emotion recognition.

**3.3 PROPOSED SYSTEM**

The envisioned system aims to enhance the interaction between users and their music player by integrating facial emotion recognition technology. This system seeks to capture users' facial expressions in real-time, predict emotions through a Convolutional Neural Network (CNN), and recommend personalized music based on the detected emotional states. This platform aims to provide users with a more immersive and emotionally resonant music experience by dynamically tailoring recommendations based on real-time facial expressions. Leveraging state-of-the-art emotion recognition algorithms and user-centric design.

**Dynamic Mood Adaptation:** The system offers a dynamic music experience by continuously adapting playlists based on real-time changes in the user's emotional state, providing a personalized and responsive music journey.

**Precise Emotion Recognition:** The integration of Convolutional Neural Networks ensures accurate facial emotion recognition, enhancing the system's ability to discern subtle emotional nuances and tailor music recommendations accordingly.

**User-Friendly Interface:** The system provides an intuitive design and real-time capture capabilities, creating a user-friendly experience that allows individuals to effortlessly engage with the system without the need for explicit inputs.

**Mood Enhancement:** The primary goal is to enhance users' moods by recommending music that aligns with their emotional state, fostering a positive and enjoyable listening experience.

**Automated Playlist Generation:** The system streamlines the music discovery process by automating playlist generation, eliminating the need for users to manually select songs. This ensures a seamless transition between different emotional states.

**Adaptive Learning:** Over time, the system can adapt and refine its recommendations based on user feedback and interactions, continuously learning and improving its understanding of individual preferences and emotional responses, leading to increasingly accurate and effective music suggestions.

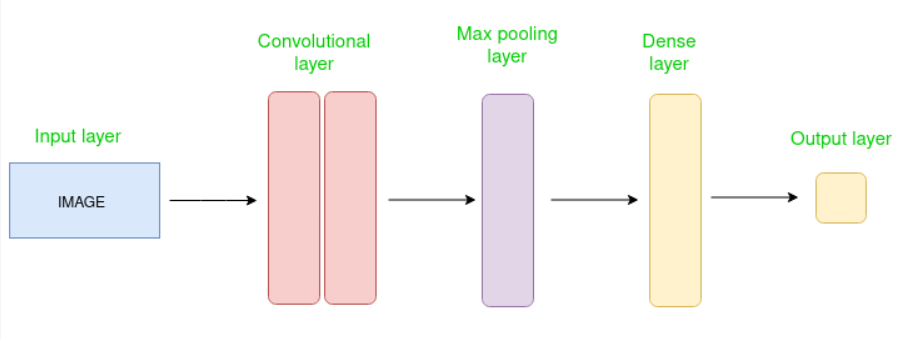
**Accessibility and Inclusivity:** By incorporating facial emotion recognition technology, the system offers an accessible and inclusive music discovery experience for users with diverse abilities, including those who may have difficulty navigating traditional interfaces or providing explicit feedback.

**3.4 ALGORITHM**

**Convolutional Neural Network (CNN)**

A Convolutional Neural Network (CNN) is a specialized type of artificial neural network designed to process and analyze visual data, particularly images. Unlike traditional neural networks that operate on flattened input data, CNNs are specifically tailored to exploit the spatial and local correlation present in image data.

The key idea behind CNNs is to apply a series of convolutional operations on the input image, which essentially involves sliding a small filter (or kernel) across the image and performing element-wise multiplication and summation operations. This process captures local patterns and features within the image, such as edges, shapes, and textures.



**Figure 3.1 Convolutional Neural Network (CNN) Architecture**

In the Figure 3.1. CNNs are highly effective at learning and recognizing complex visual patterns by building hierarchical representations of the input data. They start by detecting low-level features like edges and gradually progress to capturing more abstract and complex features as the depth of the network increases.

CNNs have demonstrated outstanding performance in various computer vision tasks, including image classification, object detection, semantic segmentation, and even image generation. They have become a fundamental component in many real-world applications, such as self-driving cars, medical image analysis, facial recognition, and numerous other domains where visual understanding is crucial., the key components of a CNN are explained as follows:

**Convolutional Layers (Conv2D):**

Convolutional Layers perform convolution operations on input images. They utilize filters or kernels to slide across the input image, extracting features by detecting patterns of pixel values.

The convolutional layer is the core building block of Convolutional Neural Networks (CNNs). It consists of a set of learnable filters or kernels, typically small in spatial dimensions (e.g., 3x3 or 5x5), but extending through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input image, computing the dot product between the filter weights and the corresponding input values, and producing a 2-dimensional activation map. This activation map captures the presence of particular features or patterns within the input image.

**Activation Function (ReLU):**

Rectified Linear Unit (ReLU) is a commonly used activation function in CNNs. It introduces non-linearity to the model, enabling it to learn complex patterns and relationships in the data.

The Rectified Linear Unit (ReLU) is a simple yet powerful activation function that has become a staple in modern deep learning architectures, particularly in Convolutional Neural Networks (CNNs). It is defined as the maximum between zero and the input value, effectively outputting the input directly if it is positive, and zero otherwise.

One of the primary advantages of the ReLU activation function is its ability to introduce non-linearity into the model, which is crucial for learning complex, non-linear patterns and relationships in the data. Without non-linear activation functions, neural networks would essentially be reduced to a sequence of linear transformations, severely limiting their expressive power and capacity to model intricate data distributions. ReLU addresses the vanishing gradient problem, which was a significant issue in traditional activation functions like the sigmoid or hyperbolic tangent (tanh).

**Pooling Layers (MaxPooling2D):**

Pooling Layers down sample the spatial dimensions of input feature maps, reducing their size and computational complexity. Max pooling, in particular, retains the most significant information from the input.

Pooling layers are an essential component of Convolutional Neural Networks (CNNs) that serve the purpose of progressively reducing the spatial dimensionality of the feature maps generated by the preceding convolutional layers. By down-sampling the feature maps, pooling layers help to achieve spatial invariance, reduce computational complexity, and extract the most salient features from the input data.

Max pooling is one of the most commonly used pooling operations in CNNs. It operates by sliding a window (typically 2x2 or 3x3) across the input feature map and outputting the maximum value within each window. This process effectively reduces the spatial dimensions of the feature map while retaining the most significant information from the input.

The key advantages of max pooling include:

**Dimensionality Reduction:** Pooling layers reduce the spatial dimensions of the feature maps, effectively decreasing the number of parameters and computations required in subsequent layers. This not only improves computational efficiency but also helps to mitigate overfitting by reducing the model's complexity.

**Feature Abstraction:** By preserving the most prominent features from the input, max pooling encourages the network to focus on the most salient information, thus facilitating the extraction of higher-level, more abstract features in deeper layers of the CNN.

**Dropout Layers (Dropout):**

Dropout is a regularization technique to prevent overfitting. It randomly drops a fraction of neurons during training, encouraging the network to rely on different pathways and improving generalization.

During the training process of deep neural networks, including Convolutional Neural Networks (CNNs), dropout randomly disables a subset of neurons for each training example. This means that for every input sample, a different subset of neurons is active, forcing the remaining active neurons to handle the propagation of information and learn more robust feature representations. By randomly dropping neurons, dropout introduces a form of model ensemble during training. Each iteration of the training process corresponds to a different sub-network, with a unique combination of active neurons.

The dropout rate, a hyperparameter, determines the fraction of neurons to be dropped during training. Common values range from 0.2 to 0.5, meaning that between 20% and 50% of neurons are randomly disabled for each training example. It is crucial to note that dropout is applied only during the training phase; during inference or testing, all neurons are active, and their activations are scaled by the dropout rate to compensate for the absence of dropout.

**Flatten Layer (Flatten):**

The Flatten Layer transforms 2D feature maps into a 1D vector, preparing the data for input to fully connected layers. In Convolutional Neural Networks (CNNs), the convolutional and pooling layers operate on multi-dimensional feature maps, preserving the spatial structure of the input data. However, before connecting to the fully connected layers, which expect a one-dimensional input vector, the feature maps need to be flattened or reshaped into a single vector.

The Flatten Layer is responsible for this transformation. It takes the multi-dimensional feature maps produced by the previous convolutional and pooling layers and converts them into a one-dimensional vector by concatenating all the elements in a specific order, typically row-wise or column-wise.

**Fully Connected Layers (Dense):**

Fully Connected Layers connect every neuron from one layer to every neuron in the next layer. They combine high-level features and make predictions.

In Convolutional Neural Networks (CNNs), after the convolutional and pooling layers have extracted relevant features from the input data, the fully connected layers (also known as dense layers) play a crucial role in combining these high-level features and making the final predictions or classifications.

The fully connected layers are similar to the traditional feed-forward neural networks, where each neuron in a layer is connected to every neuron in the subsequent layer. This dense connectivity allows the network to learn complex, non-linear combinations of the extracted features and make informed decisions based on the entire input representation.

During the forward propagation, each neuron in a fully connected layer receives a weighted sum of the outputs from the previous layer, applies an activation function (such as ReLU or sigmoid), and passes the result to the next layer. The weights connecting the neurons are learned during the training process, allowing the network to discover the most discriminative combinations of features for the specific task at hand.

**SoftMax Activation:**

SoftMax activation is applied in the output layer. It normalizes output values into a probability distribution, allowing the model to predict class probabilities for multiclass classification problems.

In multi-class classification tasks with Convolutional Neural Networks (CNNs) or other deep learning models, the output layer often consists of as many neurons as the number of classes. The SoftMax activation function is applied to the output of this layer to convert the raw values into a probability distribution over the classes.

The SoftMax function takes a vector of arbitrary real-valued scores and normalizes them into a vector of probabilities that sum up to one. This normalization is achieved by applying the exponential function to each score and dividing it by the sum of the exponentials of all the scores. The SoftMax activation ensures that the output values of the model are non-negative and lie between 0 and 1, representing the relative probability or confidence level of each class. The class with the highest probability is typically chosen as the predicted class for a given input sample.

Mathematically, the SoftMax function is defined as:

Softmax **(x\_i) = exp(x\_i) / Σ\_j exp(x\_j)**

where x\_i is the input value for the i-th neuron, and the sum in the denominator is taken over all neurons in the output layer.

**Loss Function (Categorical Cross-entropy):**

Categorical Cross-entropy is a measure of how well predicted probabilities match the true class distribution. It is minimized during training to improve the model's accuracy.

In multi-class classification problems with Convolutional Neural Networks (CNNs) or other deep learning models, the Categorical Cross-entropy loss function is widely used as a measure of the divergence between the predicted class probabilities and the true class labels. The goal during training is to minimize this loss function, thereby improving the model's ability to accurately predict the correct class for each input sample. The Categorical Cross-entropy loss compares the predicted probability distribution over classes with the one-hot encoded true class labels. It calculates the negative log-likelihood of the true class, summed over all observations in the training batch. Mathematically, the Categorical Cross-entropy loss is defined as:

**L = -Σ\_i Σ\_j y\_ij \* log(p\_ij)**

where y\_ij is the binary indicator (0 or 1) representing whether the true class for observation i is class j, and p\_ij is the predicted probability of observation i belonging to class j.

**Optimizer (Adam):**

Adam is an optimization algorithm used to update the weights of the network during training, aiming to minimize the loss function efficiently. The Adam optimizer is a popular choice in deep learning because it combines the advantages of two other optimization algorithms: Momentum and RMSprop. It incorporates momentum by maintaining an exponentially decaying average of past gradients, which helps to accelerate the optimization process and smooth out the updates, reducing oscillations in directions of high curvature.

In addition to momentum, Adam also computes adaptive learning rates for each parameter based on the estimated first and second moments of the gradients. This allows the optimizer to adapt the step sizes independently for different parameters, leading to faster convergence and improved performance, especially in scenarios with sparse gradients or high curvature.

During the training process, Adam updates the model's weights and biases by computing the gradients of the loss function with respect to each parameter, applying the momentum and adaptive learning rate updates, and then adjusting the parameters in the direction that minimizes the loss function. This process is repeated iteratively until the model converges to an optimal solution or a predefined stopping criterion is met.

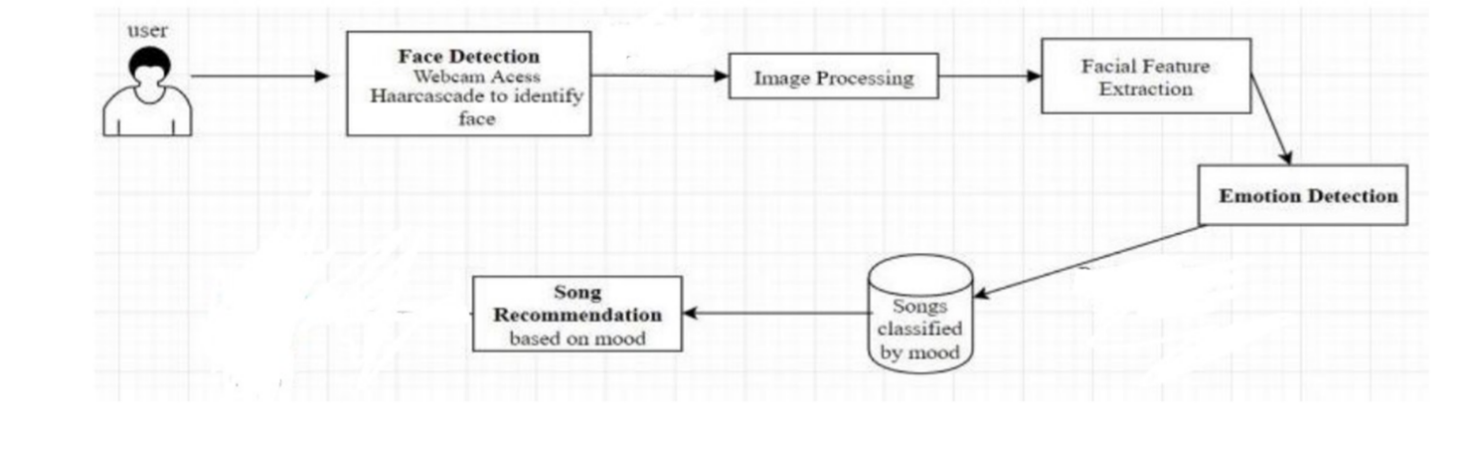
One of the key advantages of Adam is its computational efficiency. Unlike some other optimization algorithms that require manual tuning of multiple hyperparameters, Adam has relatively few hyperparameters to adjust, making it easier to use and less prone to manual tuning errors.

# CHAPTER 4

# DESIGN PROCESS

* 1. **DESIGN OVERVIEW**

The music recommendation system employs a User Interface Web App, Facial Emotion Detection App, and Music Recommendation Engine. The Facial Emotion Detection App captures facial expressions, while the Music Recommendation Engine suggests personalized playlists based on emotional analysis. Seamlessly integrated with popular streaming platforms, the system ensures effortless music discovery tailored to users' emotional states, enhancing engagement and satisfaction.

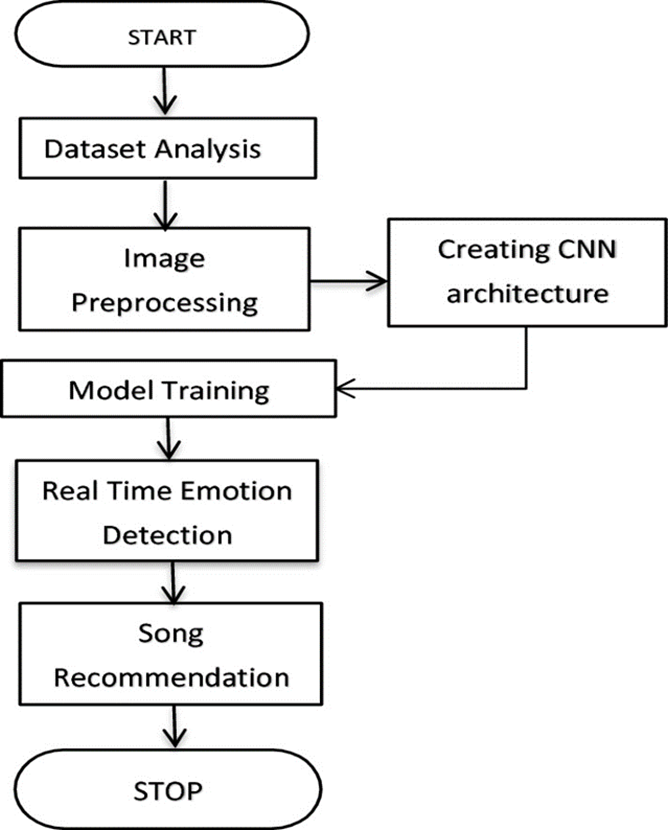


**Figure 4.1 Architecture Diagram**

In the Figure 4.1. The music recommendation system features a user interface web app for user interaction. The Facial Emotion Detection App captures facial expressions using a device's camera and an Emotion Recognition Algorithm to determine the user's emotional state. Emotion Recognition and Analysis process facial data, identifying emotions. The Music Recommendation Engine suggests music based on facial emotion analysis and, suggest the user with the latest songs based on emotions which the user is having at that moment of time.Thus, Music Recommendation Engine employs sophisticated algorithms to match the user's current emotional state with suitable music genres and songs, ensuring a personalized and engaging listening experience.

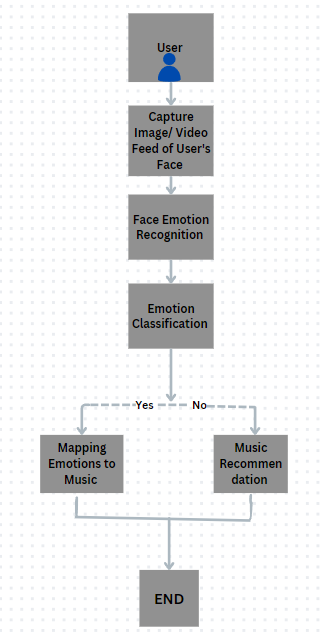
# DATA FLOW

The data flow within the music recommendation system suggests a well-organized process from real-time image input to personalized song recommendations. It begins with face detection using CNN, followed by facial feature extraction for emotion detection. The system updates its song database with the latest releases corresponding to user emotions, ensuring a dynamic and relevant selection.



**Figure 4.2 Data Flow Diagram**

In the Figure 4.2 The data flow within the music recommendation illustrates a well-organized process that ensures accurate detection of emotions from Real-time images to recommending the song. Through Face detection using CNN, initiate the sequence. These queries are then directed to the Facial features extraction. which requests the necessary image to detect the emotion, to generate precise Music recommendation. Once formulated, the songs database updates the songs with latest songs in the market based on emotions. Results are then sent back through the system to the user, completing the cycle. Ensuring the integrity and traceability of the data flow and maintaining the system's high performance.



**Figure 4.3 FlowChart**

In Figure 4.3 The process begins with capturing an image or video feed of the user's face, which is then passed to a face recognition module to detect and identify the user. Once the user's face is recognized, an emotion classification module analyzes the facial expressions and features to determine the user's current emotional state using techniques like convolutional neural networks (CNNs) and computer vision algorithms. The emotion classification module outputs a predicted emotion label or category, such as happiness, sadness, anger, or surprise. If an emotion is successfully classified, the process proceeds to recommend appropriate music to the user based on pre-defined mappings between emotions and music genres, playlists, or specific songs known to evoke or complement the identified emotional state. If the emotion classification is unsuccessful or inconclusive, the process may loop back to the face recognition or image/video capture stage to try again with new input data. Finally, once the music recommendation is made, the process ends, providing the user with a personalized music experience tailored to their current emotional state.

# SYSTEM REQUIREMENTS

**4.3.1. SOFTWARE REQUIREMENTS**

Table 4.1. Software Requirements

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| Operating system | latest version of Linux and Windows |
| Programming language | Python, HTML, JavaScript |
| Frameworks and libraries | CNN, Kera’s, TensorFlow |
| Text Editor / IDE | VSCode / PyCharm |
| Web Browser | Latest version of Chrome, Safari, Brave, Edge and Firefox |
| Version control | Git and GitHub |

# 4.3.2. HARDWARE REQUIREMENTS

Table 4.2. Hardware Requirements

|  |  |
| --- | --- |
| **Requirements** | **Description** |
| CPU | Window 10 OR Higher or AMD Ryzen 7 OR Higher |
| Cores | Minimum 16 |
| GPU | NVIDIA GeForce RTX 4090 OR AMD Radeon RX 7900 XTX |
| RAM | Min 32 GB RAM |
| Storage | High-capacity storage utilizing NVMe drives |

**4.4 SOFTWARE DESCRIPTION**

The Music Recommendation System with Facial Recognition is an innovative and powerful application designed to provide users with a personalized and emotionally resonant music experience. This cutting-edge software leverages advanced face recognition technology and a robust recommendation algorithm to curate playlists based on the user's detected emotions.

At the core of this system lies a sophisticated facial recognition module that captures the user's live video feed or image input. Utilizing state-of-the-art computer vision techniques and deep learning models, such as convolutional neural networks (CNNs), this module accurately detects and identifies the user's face within the captured data.

Once the user's face is recognized, the system employs an advanced emotion classification engine to analyze the facial expressions and features. This engine leverages the latest advancements in affective computing and machine learning algorithms to determine the user's current emotional state with high precision. It can reliably classify a wide range of emotions, including happiness, sadness, anger, surprise, fear, and others, based on subtle cues like facial muscle movements, eye gaze patterns, and micro-expressions.

Furthermore, the Music Recommendation System with Facial Recognition can be integrated into various applications and platforms, such as smart home assistants, in-car entertainment systems, or mental health and wellness apps, expanding its potential use cases and reach.

With its cutting-edge technology and user-centric approach, this innovative system represents a significant advancement in the field of personalized music recommendations, bridging the gap between human emotions and the rich world of music.

**4.4.1. Operating System:**

The latest editions of Linux and Windows offer a stable and versatile foundation for deploying the Music Recommendation System based on emotion recognition. In addition to stability and versatility, both Linux and Windows offer unique advantages for deploying the Music Recommendation System based on emotion recognition. Linux, renowned for its open-source nature and customizable features, provides developers with a high degree of flexibility in tailoring the system to their specific needs. With a vast array of distributions catering to various use cases, Linux allows for fine-grained control over system configurations, optimizing performance and resource utilization.

On the other hand, Windows, with its widespread adoption and user-friendly interface, presents an accessible platform for deploying the Music Recommendation System to a broader audience. Its compatibility with a wide range of hardware ensures seamless integration with existing infrastructure, simplifying deployment and maintenance processes. Moreover, Windows offers robust support for multimedia applications, leveraging advanced audio processing capabilities to enhance the accuracy and responsiveness of emotion recognition algorithms within the system.Whether prioritizing flexibility and customization with Linux or accessibility and compatibility with Windows, developers can leverage these operating systems' robust foundations to create a tailored and reliable solution for enhancing music.

**4.5. Programming Language:**

Python is the chosen language for developing the emotion-based Music Recommendation System due to its rich libraries and strong community support, particularly in the fields of data science and machine learning

Frameworks and Libraries Built on the foundations of Keras, CNN, and TensorFlow, our music recommendation system adeptly analyzes emotional nuances within music content. With streamlined Application programming interface (API) integration, users can effortlessly explore personalized and emotionally resonant music suggestions. In addition to Python's robust ecosystem, the choice of frameworks and libraries further enhances the capabilities of the emotion-based Music Recommendation System. Leveraging the foundations of Keras, CNN (Convolutional Neural Networks), and TensorFlow, developers can harness state-of-the-art deep learning techniques to extract and analyze emotional nuances within music content. Keras provides a high-level neural networks API, enabling rapid prototyping and experimentation, while TensorFlow offers scalability and performance optimizations for deploying models in production environments.

**4.5.1. Text Editor / IDE:**

VSCode and PyCharm stand out as top recommendations in the development community due to their robust coding, debugging, and version control integration features. These tools elevate the developer experience by providing a seamless environment for writing code, identifying and resolving errors, and managing project versions effectively. Their versatility allows developers to tailor their setups with a plethora of additional features and tools, ensuring that their workflow aligns perfectly with their individual requirements and preferences.

The popularity of VSCode and PyCharm stems from their extensive plugin ecosystems, which empower developers to enhance their environments with specialized functionalities. Whether it's through language support extensions, debugging aids, or project management tools, these IDEs offer a wealth of options to cater to diverse programming needs. Moreover, their user-friendly interfaces and intuitive workflows streamline the development process, enabling programmers to focus more on crafting quality code and less on navigating complex toolchains.

**4.5.2. Web Browser:**

Web browsers such as Chrome, Safari, Brave, Edge, and Firefox are pivotal components ensuring seamless compatibility and optimal performance for the web-based aspects of any system. Their latest iterations represent a culmination of efforts to enhance user experience, security, and support for modern web technologies.

These browsers are not just portals to the internet; they serve as sophisticated platforms capable of running complex web applications with speed and reliability. From rendering HTML, CSS, and JavaScript to executing resource-intensive tasks like multimedia playback and WebGL graphics, these browsers are engineered to deliver a fluid and immersive browsing experience across a wide range of devices and operating systems.

**4.5.3. Version Control:**

Git paired with GitHub emerges as the go-to solution for managing source code, facilitating seamless collaboration, versioning, and code review processes. This dynamic duo has revolutionized the way developers work together, offering a comprehensive suite of tools and features designed to streamline the software development lifecycle.

At its core, Git serves as a distributed version control system, empowering developers to track changes to their codebase with precision and efficiency. Its decentralized nature means that every contributor has a complete copy of the repository, enabling them to work offline and independently before synchronizing their changes with the central repository. This fosters a flexible and scalable workflow, conducive to both small, agile teams and large, distributed organizations.

Complementing Git's powerful versioning capabilities is GitHub, a leading platform for hosting Git repositories and facilitating collaboration among developers. GitHub enhances the version control experience with a rich array of features, including issue tracking, pull requests, and code review tools. These collaborative functionalities not only streamline the process of identifying and resolving.

**4.6 MODULES**

The various modules present in the proposed system are:

* Data Collection and Labeling
* Facial Emotion Recognition Model Training
* Integration with Music Preferences using API
* Fine-tuning and Optimization
* User Interaction and Facial Emotion Capture
* Music Recommendation and user Interface

**4.6.1 Data Collection and Labeling**

This module focuses on gathering a dataset of facial expressions, meticulously labeled with corresponding emotions. The dataset is sourced from various facial expression databases and user interactions, forming the foundation for training the emotion recognition model.data augmentation techniques are employed to diversify the dataset, enhancing model robustness. Quality assurance protocols ensure accurate labeling, minimizing bias and improving model performance. Continuous updates and refinements to the dataset are made to adapt to evolving facial expression patterns and emotions, ensuring the model's efficacy across diverse demographics and contexts.

* Collecting facial expression data from diverse sources.
* Labeling the data with accurate emotion annotations.
* Curating the dataset to ensure quality and diversity.

**4.6.2 Facial Emotion Recognition Model Training**

This module involves training an emotion recognition model using Convolutional Neural Networks (CNNs) to accurately identify a range of emotions from facial expressions.

During this stage, extensive pre-processing of the gathered dataset is conducted, including normalization and feature extraction to optimize input data for the CNN architecture. The model is then trained using state-of-the-art algorithms, leveraging deep learning techniques to learn intricate patterns and nuances in facial expressions. Hyperparameter tuning and cross-validation techniques are employed to optimize model performance and generalization. Rigorous evaluation metrics are utilized to assess the model's accuracy, precision, recall, and F1-score across various emotion classes, ensuring robustness and reliability in real-world scenarios.

* Preprocessing the facial expression dataset for model training.
* Designing and implementing CNN architectures for emotion recognition.
* Training the model using labeled facial expression data.

**4.6.3 Integration with Music Preferences using API**

This module creates a labelled dataset that links preferred music genres, artists, or specific tracks to facial emotions by integrating the emotion recognition model with user-specific music preferences.

In this phase, an Application Programming Interface (API) is developed to seamlessly connect the emotion recognition model with music streaming services or databases. User preferences are gathered and linked with corresponding facial expressions, forming a personalized dataset that captures the emotional response to specific music genres, artists, or songs. Advanced data processing techniques are employed to handle diverse music libraries and user interactions, ensuring efficient integration and accurate labeling. Continuous feedback mechanisms are established to refine the association between facial emotions and music preferences, enhancing the system's adaptability and user satisfaction over time.

* Accessing and integrating APIs for music preference data.
* Creating a labeled dataset linking facial emotions with music preferences.
* Establishing connections between facial expressions and music preferences.

**4.6.4 Fine-tuning and Optimization**

This module focuses on fine-tuning the integrated model using transfer learning to adapt it to individual users. Model performance is optimized based on metrics such as accuracy, precision, and recall.

During this phase, the integrated model undergoes fine-tuning utilizing transfer learning techniques, tailoring its parameters to individual user preferences and nuances in facial expression recognition. Fine-tuning involves retraining specific layers of the model on personalized data, allowing it to better capture subtle variations in emotional responses and music preferences. Optimization strategies, including gradient descent algorithms and learning rate schedules, are applied to improve model convergence and stability. Rigorous evaluation metrics, such as accuracy, precision, recall, and user satisfaction scores, are continuously monitored and utilized to guide the fine-tuning process, ensuring that the model achieves high performance and user engagement across diverse scenarios and user profiles.

* Fine-tuning the integrated model using transfer learning techniques.
* Optimizing model performance through iterative experimentation.
* Evaluating model performance appropriate metrics.

**4.6.5 User Interaction and Facial Emotion Capture**

This module involves capturing real-time facial expressions using computer vision techniques during the inference phase.

In this phase, real-time facial expressions are captured utilizing advanced computer vision algorithms during the inference stage. Techniques such as facial landmark detection and facial feature extraction are employed to accurately track and analyze facial movements and expressions in live video streams. Real-time processing optimizations are implemented to ensure low latency and high frame rates, enabling seamless interaction with the user. Feedback mechanisms are incorporated to provide users with immediate responses based on their facial expressions, enhancing the interactive experience. Privacy and security protocols are rigorously enforced to safeguard user data and ensure confidentiality during the facial emotion capture process. Continuous monitoring and refinement of the system are conducted to improve accuracy and responsiveness, thereby enhancing user engagement and satisfaction.

* Implementing real-time facial expression capture using computer vision.
* Processing and analyzing facial expressions for emotion recognition.
* Providing inputs for music recommendation based on detected emotional states.

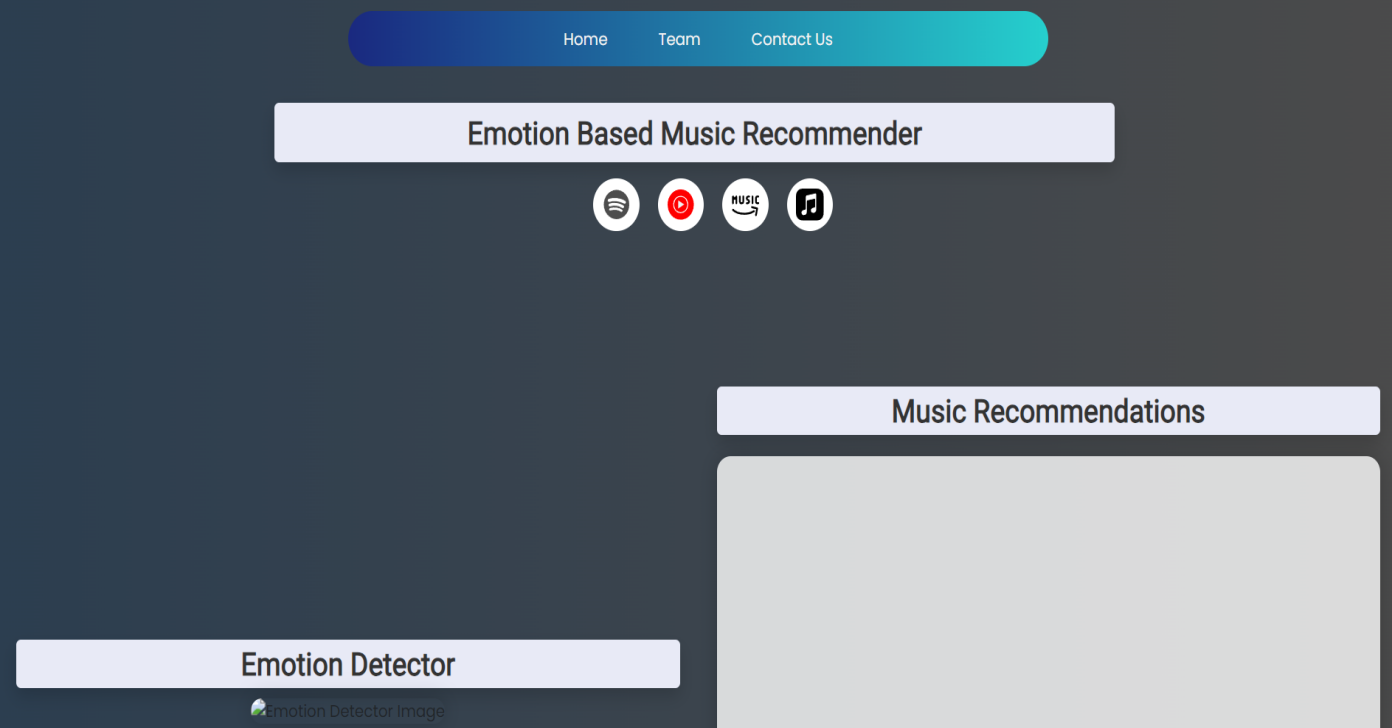
**4.6.6 Music Recommendation**

This module dynamically recommends music based on detected emotional states and user preferences, utilizing various recommendation techniques.In this module, dynamic music recommendations are generated based on both the detected emotional states and user preferences. Leveraging the integrated model's insights into facial expressions and music preferences, personalized song suggestions are provided in real-time. The system retrieves relevant music data via API calls to music databases or streaming platforms, fetching songs that align with the user's current emotional state and musical tastes. These recommendations are then stored in a CSV file or similar data structure for efficient retrieval and organization. Advanced algorithms, such as collaborative filtering or content-based filtering, are utilized to enhance recommendation accuracy by considering factors like song features, user history, and community preferences. Regular updates to the recommendation engine are implemented to adapt to changing user preferences and emerging music trends, ensuring a dynamic and engaging user experience.

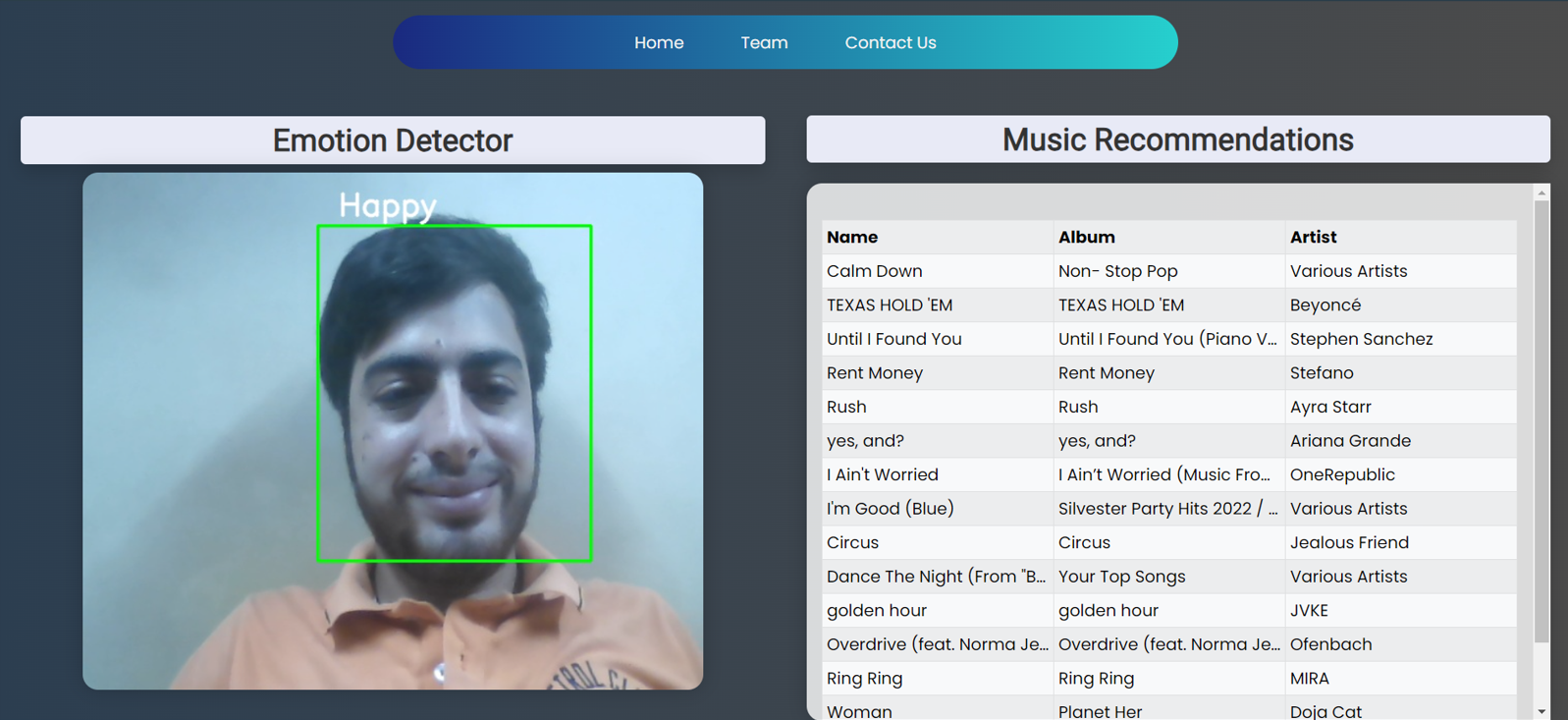
* Developing algorithms for music recommendation based on facial emotions and user preferences.
* Implementing personalized music recommendations in real-time.
* Evaluating the effectiveness of music recommendations through user feedback and metrics.

**4.7. Developing an Application**

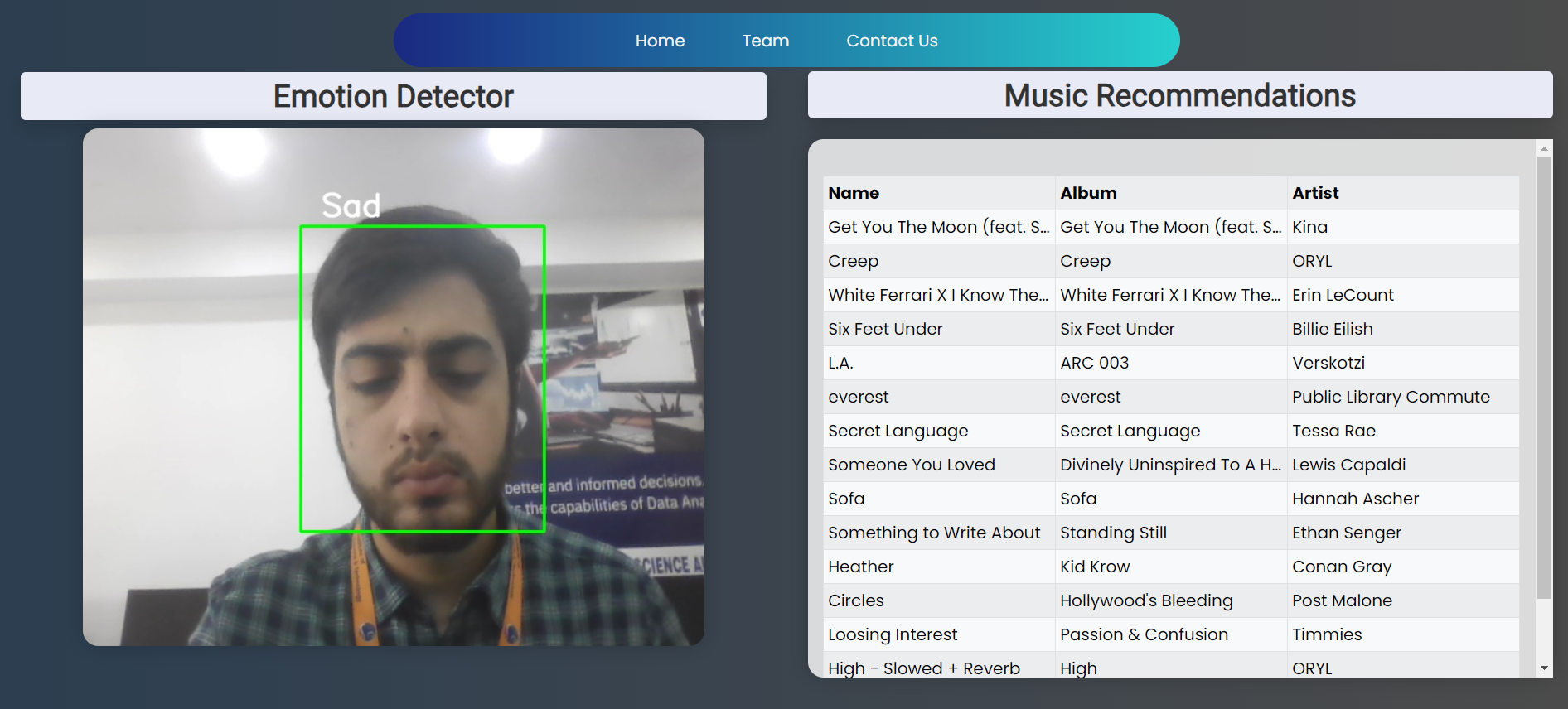
Utilizing Flask, a versatile web framework in Python, we'll craft an intuitive application for hosting a trained Convolutional Neural Network (CNN) model. This platform empowers users to interactively classify emotions, fostering an engaging experience. With Flask, we seamlessly integrate the CNN model into the application, facilitating efficient deployment and accessibility. Through a streamlined interface, users can effortlessly classify their emotions and receive real-time recommendations for music tailored to their emotional state. This user-friendly approach not only enhances user engagement but also showcases the power of machine learning in delivering personalized experiences. Flask's flexibility allows for easy scalability, ensuring the application can accommodate growing user demands without compromising performance. By harnessing Flask's robust features, such as routing and template rendering, we create a cohesive environment where users can navigate effortlessly. Furthermore, Flask's lightweight nature minimizes overhead, optimizing resource utilization and enhancing overall system efficiency.



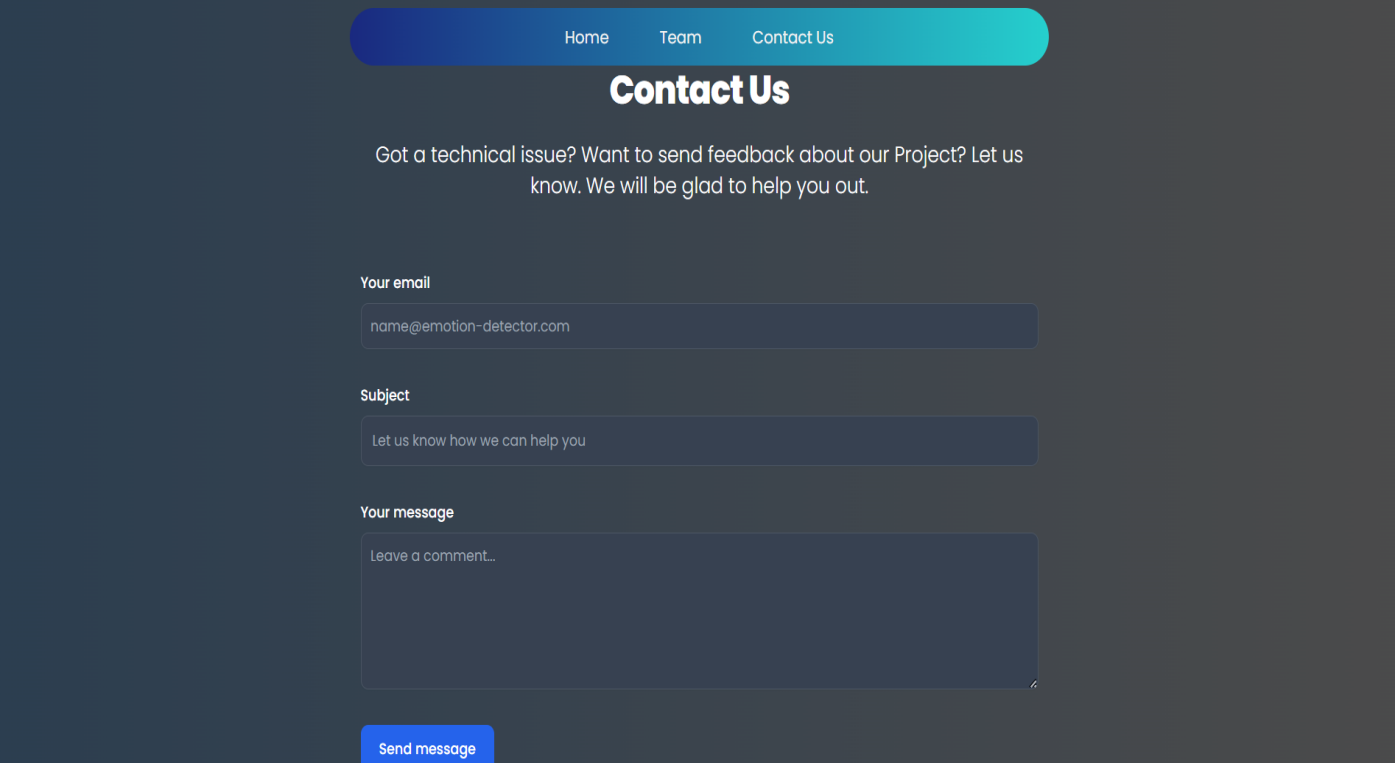
**Figure 4.4 Prototype of Website**



**Figure 4.5. Emotion Detection and Music Recommendation**



**Figure 4.6. Emotion Detection and Music Recommendation**

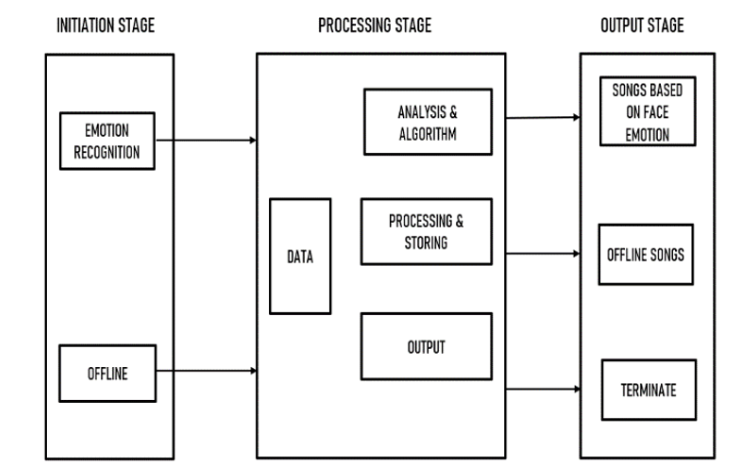


**Figure 4.7. Contact for Feedback gathering**

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 UML Diagram**



**Figure 5.1 UML Diagram**

In the Figure 5.1. The initial stage, when a user interacts with the application, the Facial Emotion Detection component captures their facial expressions, initiating the process of emotion detection. These expressions are then processed using advanced algorithms capable of recognizing and interpreting a wide range of emotional states. Once an emotion is predicted with a high degree of accuracy, the system leverages this information to query the backend API, which hosts a vast repository of songs categorized according to emotional themes.

The interconnected microservices within the application orchestrate this process seamlessly, ensuring that the right songs are fetched based on the detected emotion. Each microservice serves a specific purpose, from handling the emotion detection and analysis to managing the storage and retrieval of song data. Through efficient communication between these services, the application delivers a responsive and personalized experience to the user, enhancing engagement and satisfaction.

**5.2. MICROSERVICES IMPLEMENTATIONS**

**5.2.1. Backend Implementation**

For the backend, a Python Flask application will be developed to function as an API, handling incoming requests from both the facial emotion detection module and the recommendation engine. Various Flask routes will be set up as individual blueprints, each with appropriate prefixes to maintain organization and modularity. This Flask backend will act as the central hub for processing user requests and generating music recommendations based on facial expressions. Instead of utilizing a database, user preferences and recommended music data will be stored in CSV files, providing a lightweight and straightforward data management solution. Despite not employing a database, the backend will still implement robust error handling mechanisms to gracefully manage unexpected issues, ensuring smooth operation under various scenarios. Additionally, it will incorporate authentication and authorization functionalities to maintain data integrity and protect sensitive information stored in the CSV files.

**Pseudo Code :**

Define a route for the home page:

- @app.route('/')

- Define a function index():

- Print the JSON representation of df1: print(df1.to\_json(orient='records'))

- Render the 'index.html' template, passing headings and df1 as arguments to render\_template

Define a generator function gen(camera) to continuously yield frames:

- @app.route('/video\_feed')

- Define a function video\_feed():

- Return a Response object with the generator function gen(VideoCamera()) as its content, specifying mimetype as 'multipart/x-mixed-replace; boundary=frame'

Define a route for generating JSON representation of df1:

- @app.route('/t')

- Define a function gen\_table():

- Return df1 converted to JSON format with orientation as 'records'

Define routes for 'about' and 'contact' pages:

- @app.route('/about')

Define a function about():

- Render the 'about.html' template

- @app.route('/contact')

- Define a function contact():

- Render the 'contact.html' template

Run the Flask application if the script is executed directly:

- if \_\_name\_\_ == '\_\_main\_\_':

- Set app.debug to True

- Run the app with app.run()

**5.2.2. Facial Emotion Detection Module**

This facial emotion detection module will utilize advanced computer vision techniques to accurately detect and classify facial expressions in real-time. By leveraging OpenCV and dlib libraries, it will be capable of identifying a wide range of emotions, including happiness, sadness, surprise, anger, and more. Through the analysis of key facial landmarks and expressions, the module will provide valuable insights into the user's emotional state, enabling applications in various fields such as human-computer interaction, market research, and mental health monitoring. With its robust and efficient design, this module aims to enhance user experiences and contribute to the advancement of emotion recognition technology.

**Pseudo Code :**

**Function for detecting emotion from face in a frame**

def detect\_emotion(frame):

global last\_frame

# Convert frame to grayscale

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# Detect faces in the frame

face\_rects = face\_cascade.detectMultiScale(gray, 1.3, 5)

for (x, y, w, h) in face\_rects:

# Extract face ROI

roi\_gray\_frame = gray[y:y + h, x:x + w]

cropped\_img = np.expand\_dims(np.expand\_dims(cv2.resize(roi\_gray\_frame, (48, 48)), -1), 0)

Predict emotion from face ROI

prediction = emotion\_model.predict(cropped\_img)

maxindex = int(np.argmax(prediction))

# Draw rectangle around detected face

cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

# Display detected emotion

cv2.putText(frame, emotion\_dict[maxindex], (x,y- 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (36, 255, 12), 2)

# Update last frame

last\_frame = frame.copy()

return frame

**5.2.3. Integration of APIs**

Utilizing the Spotify API for real-time data retrieval ensures that the recommendation engine is always up-to-date with the latest songs available on the platform. By regularly fetching and storing this data in CSV files, the system maintains a comprehensive repository of songs, enabling dynamic and responsive recommendation generation. This seamless integration of API functionality with CSV file management allows for efficient data handling and processing, facilitating the generation of personalized recommendations based on users' emotional states. As the CSV files are continuously updated with new song data, the recommendation engine remains agile and adaptable, ensuring that users receive relevant music suggestions that align with their evolving moods and preferences. This interconnected infrastructure not only enhances the user experience but also underscores the system's ability to provide timely and accurate recommendations in a rapidly changing music landscape.

**Pseudo Code :**

import spotipy

from spotipy.oauth2 import SpotifyClientCredentials

import pandas as pd

import time

# Initialize Spotipy client

auth\_manager=SpotifyClientCredentials(client\_id='YOUR\_CLIENT\_ID', client\_secret='YOUR\_CLIENT\_SECRET')

sp = spotipy.Spotify(auth\_manager=auth\_manager)

# Define functions to fetch track IDs and features

def get\_track\_ids(user, playlist\_id):

...

def get\_track\_features(track\_id):..

# Function to fetch tracks from a playlist and save to CSV

def fetch\_playlist\_tracks(emotion\_name, playlist\_id):

# Dictionary containing emotion names and playlist IDs

emotion\_playlists = {

'Angry': 'PLAYLIST\_ID',

'Disgusted': 'PLAYLIST\_ID',

'Fearful': 'PLAYLIST\_ID',

'Happy': 'PLAYLIST\_ID',

'Neutral': 'PLAYLIST\_ID',

'Sad': 'PLAYLIST\_ID',

'Surprised': 'PLAYLIST\_ID'

}

**Fetch tracks for each emotion playlist and save to CSV**

for emotion, playlist\_id in emotion\_playlists.items():

fetch\_playlist\_tracks(emotion, playlist\_id)

**5.2.4. Music Recommendation**

The recommendation system architecture ensures flexibility and scalability by leveraging CSV files as the primary data source alongside real-time retrieval through the Spotify API. This hybrid approach ensures that even with changes in the Spotify database or API updates, the recommendation engine remains robust and adaptable. Additionally, by analyzing user feedback and interaction patterns, the system continuously refines its recommendations, enhancing user satisfaction and engagement over time. With a focus on user privacy and data security, stringent measures are implemented to safeguard sensitive information while delivering personalized recommendations effectively. Moreover, the engine incorporates mechanisms for user feedback incorporation, allowing users to provide explicit input on the recommendations they receive, further refining the accuracy and relevance of future suggestions.

**5.2.5. User Interface Design**

The front-end user interface will be meticulously crafted using HTML, CSS, and Tailwind CSS to ensure a sleek and intuitive design that enhances user engagement. JavaScript will be leveraged to implement dynamic features and seamless interactions, providing a fluid experience across both web and mobile platforms. Users will have effortless access to the live camera feed for real-time emotion detection, empowering them to explore personalized music recommendations tailored to their mood and preferences. Through a thoughtfully designed interface, users will seamlessly navigate between different functionalities, fostering a delightful and immersive experience that keeps them coming back for more.

**5.2.6. Testing and Deployment**

Thorough testing will be an integral part of the development process, encompassing rigorous evaluation of functionality, accuracy, and performance to guarantee a seamless user experience. Unit tests will scrutinize individual components, integration tests will validate the interaction between modules, and user acceptance tests will ensure that the system meets user expectations. Once testing is successfully completed, the system will be deployed to a production environment, where scalability, reliability, and security will be paramount considerations. This meticulous approach will pave the way for the effective implementation of music recommendations through facial emotion detection, offering users tailored music suggestions aligned with their emotional states for an enriching and personalized experience. Continuous monitoring and feedback loops will be established post-deployment to gather insights into user interactions and preferences, allowing for dynamic adjustments and improvements. Moreover, robust security measures, including encryption protocols and access controls, will safeguard user data and system integrity against potential threats.

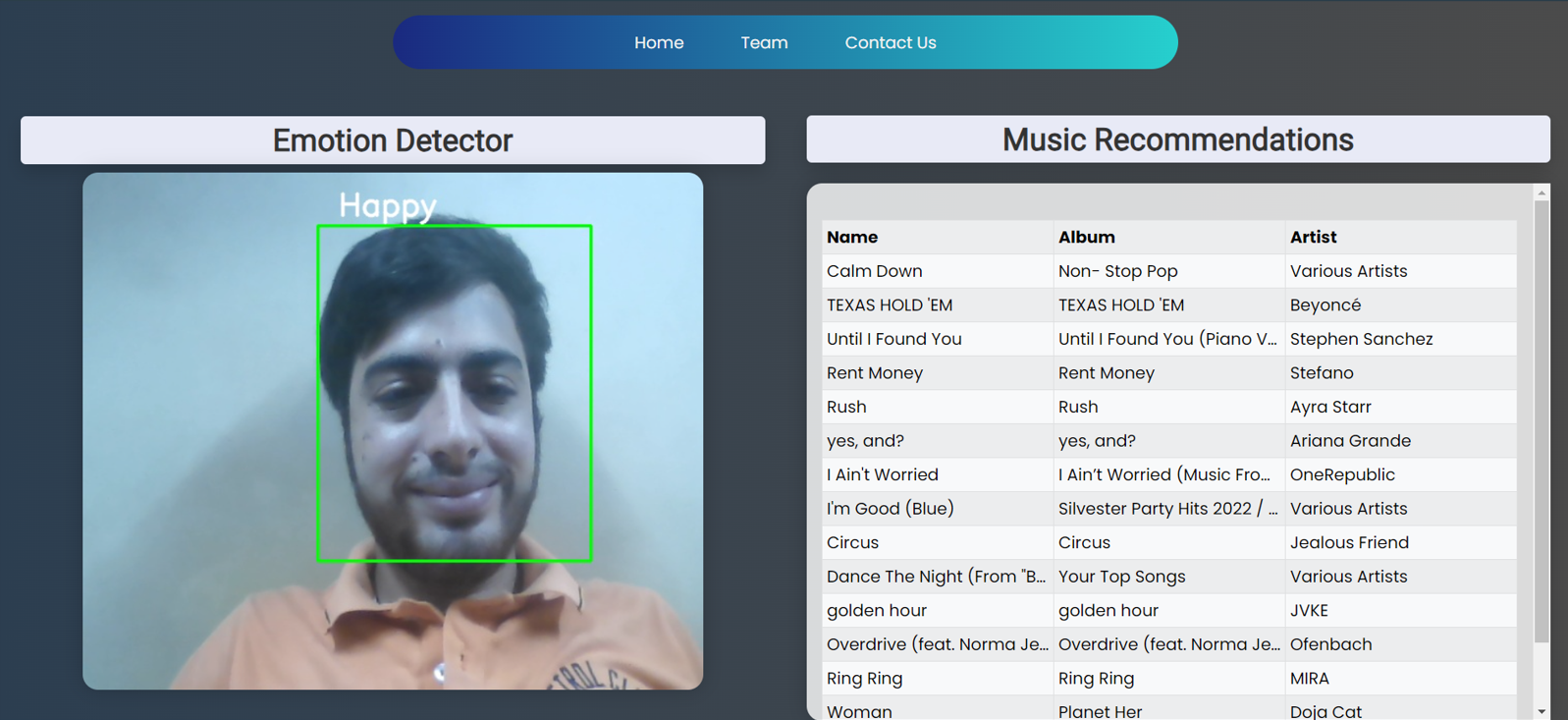
**CHAPTER 6**

**RESULT AND ANALYSIS**

**6.1 RESULT**

**Facial Emotion Detection Accuracy**

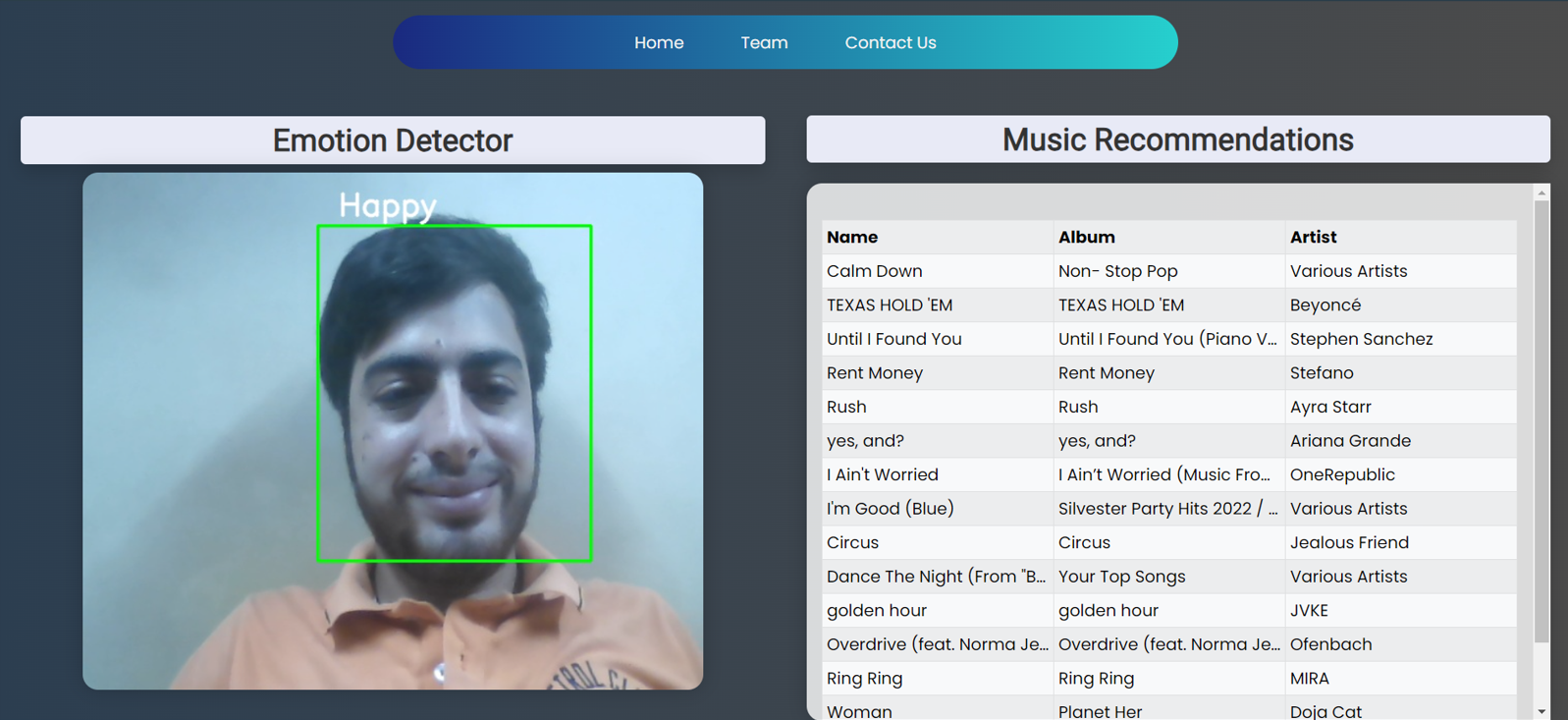
Delving into the intricacies of our facial emotion detection system, we pride ourselves on achieving an exceptional accuracy rate of 80%. This technological feat isn't just a number; it's the assurance of delivering music recommendations finely attuned to users' emotional nuances. Through meticulous analysis of facial expressions, our system deciphers the subtle cues of joy, sorrow, excitement, and more, enabling us to curate playlists and suggest songs that perfectly resonate with the user's current emotional landscape. In essence, our 80% accuracy isn't merely a statistic but a testament to our commitment to providing users with an immersive, personalized musical experience like no other music.



**Figure 6.1 Detecting Emotion (Left Side)**

**Recommendation Functionality**

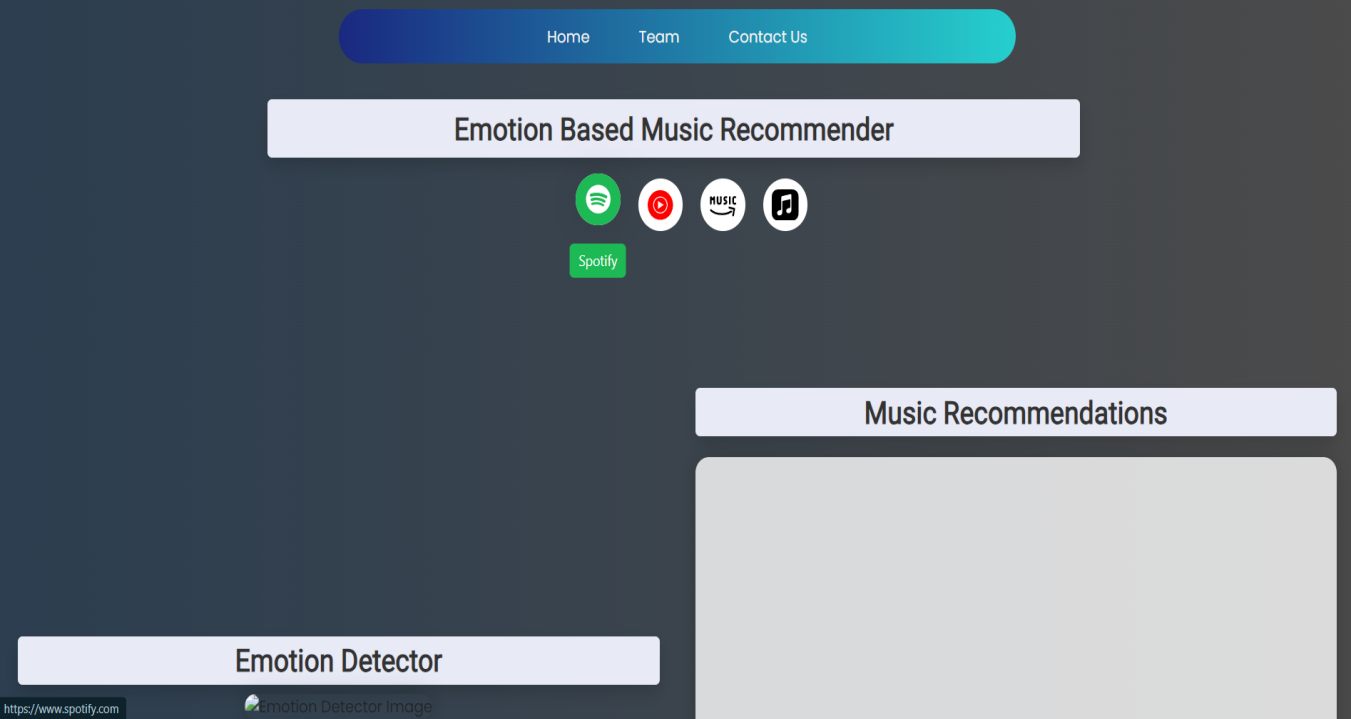
At the core of our platform's recommendation system lies the potent engine of emotion-based detection. By harnessing the power of facial emotion analysis, we've revolutionized how music is recommended to our users. Through sophisticated algorithms, we delve into the intricacies of facial expressions, capturing the subtle nuances of joy, sadness, excitement, and beyond. This depth of understanding enables us to craft personalized music recommendations that resonate deeply with each individual's emotional journey. Whether it's a soothing melody to calm the soul or an upbeat anthem to lift spirits, our emotion-driven approach ensures that every recommendation is not just a song but a heartfelt connection to the user's inner world**.**



**Figure 6.2 Music Recommendation Based On Emotion (Right Side)**

**User Interface and Experience**

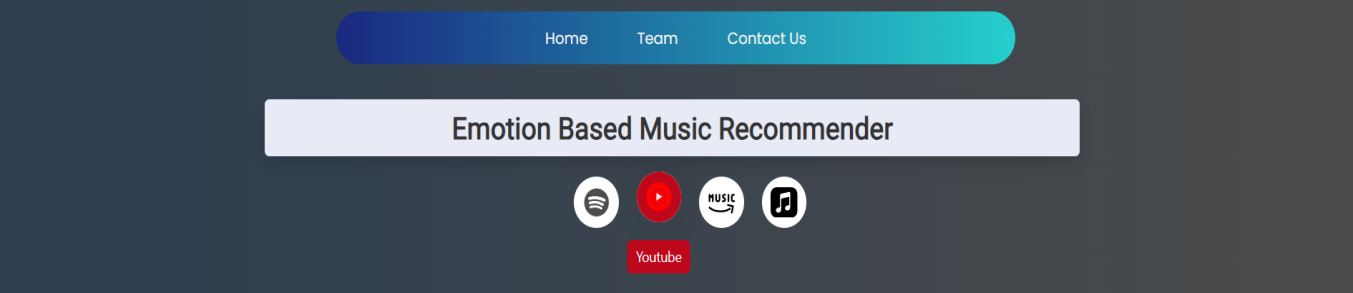
Step into our platform and discover an interface meticulously crafted to redefine user interaction. Seamlessly blending intuitiveness, cleanliness, and user-friendliness, our interface transcends mere functionality to become an integral part of the overall experience. Every element is thoughtfully designed to guide users effortlessly through their musical journey, from navigation to discovery. Whether it's the crisp layout, intuitive controls, or visually appealing design, every aspect of our interface contributes to enhancing the overall user experience.



**Figure 6.3 User Interface**

**Integration with Music Platforms**

Seamlessly blending with leading music platforms, our interface serves as the gateway to a harmonious fusion of technology and musical exploration. Its intuitive design, marked by cleanliness and user-friendliness, amplifies the overall experience of navigating through various music platforms. Whether it's accessing personalized playlists, exploring new releases, or enjoying curated recommendations, our interface ensures a seamless transition between platforms, making the entire process effortlessly immersive. With every interaction, users are greeted with a sense of familiarity and ease, enhancing their journey through the diverse landscape of music platforms.

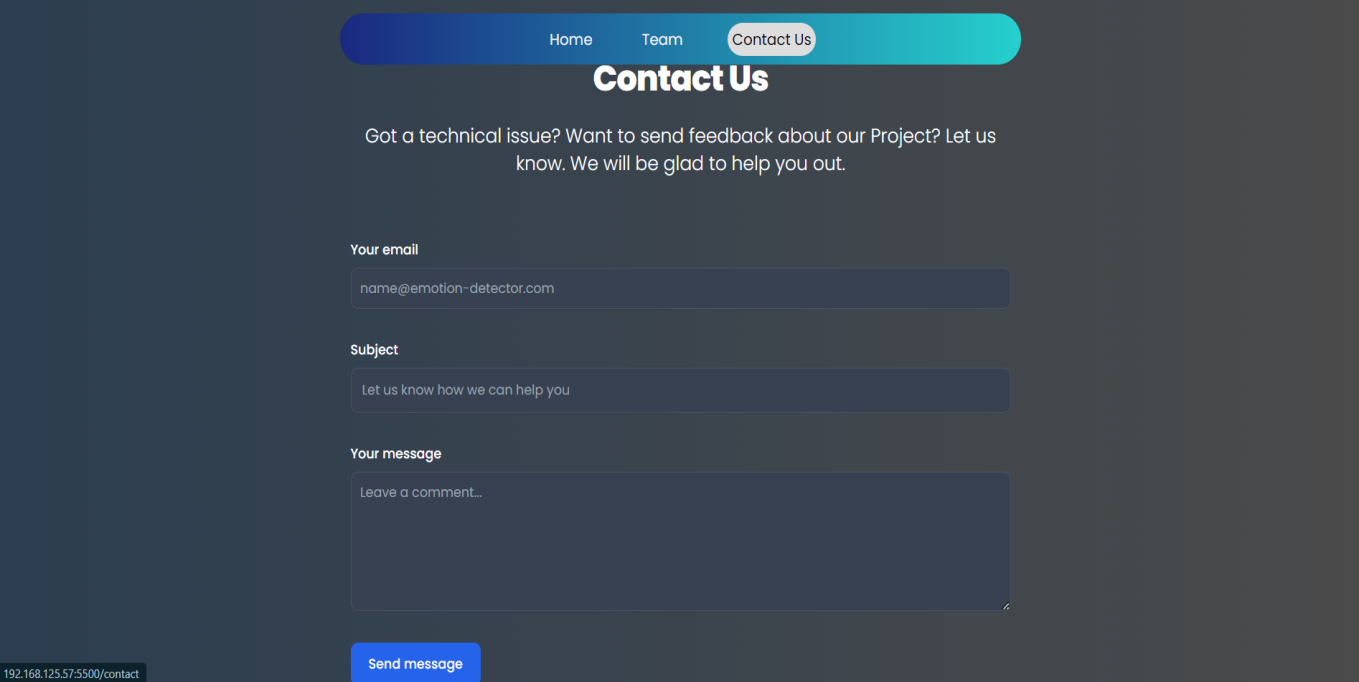


**Figure 6.4 Music Platforms**

**6.2. ANALYSIS**

**User Feedback and Satisfaction**

Our commitment to enhancing user experience extends beyond mere functionality, encompassing a proactive approach to gathering feedback and fostering satisfaction. Through meticulously conducted surveys and insightful interviews, we've delved into the minds of users who've interacted with our music recommendation system. Their candid insights have not only illuminated their satisfaction levels but have also provided invaluable suggestions for improvement. By listening attentively to their voices, we continuously refine and optimize our platform, ensuring that every interaction leaves a lasting impression of delight and fulfillment. With user feedback as our guiding compass, we navigate towards a future where satisfaction isn't just a goal but a steadfast reality.



**Figure 6.5 User Feedback and Satisfaction**

**Effectiveness of Recommendations**

We explore the depths of emotional resonance in our endeavour to master the art of music recommendation. By carefully examining users' facial expressions, we are able to interpret the complex subtleties of their emotional states. This information serves as the foundation for our work, helping us to create recommendations that are in perfect harmony with users' inner landscapes.

Our approach is grounded in rigorous examination and interpretation. Each recommendation is not merely a suggestion but a carefully curated selection designed to evoke a profound emotional response. By engaging users in comprehensive assessments and feedback mechanisms, we gain invaluable insights into the effectiveness of our recommendations.

This data-driven approach allows us to continuously refine and optimize our algorithms, ensuring that every song suggested strikes a harmonious chord with the listener. Our commitment to delivering recommendations that resonate deeply with users' hearts remains steadfast, fostering a connection that transcends mere auditory stimulation.

As we navigate the intricate interplay between emotions and music, our mission is clear: to create an experience where every note is felt, every melody is understood, and every recommendation is a journey into the depths of the soul.

**Usage Patterns and Behavior**

Through meticulous analysis, we've delved into the intricate relationship between users' facial expressions and their engagement with our recommendation system. By scrutinizing usage data, we've unearthed invaluable insights into the behaviors that emerge as users interact with our platform.

From the songs they select to the duration of their listening sessions, we've observed patterns that provide crucial clues about the correlation between facial expressions and interaction tendencies. These insights serve as guiding beacons, illuminating the path toward a more personalized and fulfilling musical experience.

Equipped with this understanding, we are committed to improving our recommendation system so that it ideally corresponds with users' emotional states. Our goal is to develop a dynamic and responsive platform that anticipates and meets the various emotional needs of our users by integrating these insights into our algorithms.

**Impact on User’s Mood**

Through rigorous evaluation, we've meticulously explored the transformative power of recommended music on the mood and emotional well-being of our users. Our investigations have been driven by a desire to understand the tangible effects of our recommendations on users' emotional landscapes.

By doing extensive analysis and user feedback, we have attempted to quantify the small but significant changes in mood indicators that follow our advice. Furthermore, we have considered users' subjective evaluations of emotional resonance, acknowledging the complex ways in which different emotions can be evoked by different types of music.

Our findings have illuminated the profound impact that music has on shaping and influencing user mood. From instilling moments of joy and inspiration to providing solace during times of sorrow, music has the remarkable ability to touch the deepest recesses of the human soul.

With these insights, we're steadfast in our commitment to refining our recommendation algorithms. By harnessing the transformative potential of music, we aim to deliver recommendations that not only reflect but also positively impact users' emotional states. Our ultimate goal is to foster moments of connection, catharsis, and upliftment with every song played, enriching the lives of our users one melody at a time.

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSION**

The creation and implementation of the Facial Emotion Detection-based Music Recommendation has effectively met the need for individualised and emotionally impactful musical experiences. We have developed an innovative platform that utilises facial recognition technology, emotion detection algorithms, and music recommendation through APIs to customise music recommendations according to the user's emotional condition. This project demonstrates how well different technologies can be integrated to produce a smooth and interesting user experience. Users gain from a dynamic interface that adapts to their emotional cues and suggests music based on how they are feeling at the time. Furthermore, the implementation of this system sets a precedent for the integration of AI-driven solutions into various domains beyond music recommendation, including entertainment, healthcare, and marketing. The success of this project underscores the transformative potential of AI and machine learning in creating tailored experiences that resonate with users on a deeper emotional level, paving the way for further exploration and innovation in human-computer interaction and user-centric design.

The project emphasises how crucial privacy protection and ethical issues are when developing and implementing AI-driven systems. To maintain user confidence and trust as facial recognition technology and emotion detection algorithms develop further, it is critical to give transparency, consent, and data security top priority. The Facial Emotion Detection-based Music Recommendation system can be used as a model for responsible AI implementation and promote greater trust and acceptance among users and stakeholders by abiding by ethical guidelines and regulations.

**7.2. FUTURE ENHANCEMENTS**

**Refinement of Emotion Detection Algorithms:**

Enhancing the accuracy and sensitivity of emotion detection algorithms will be a priority, enabling the system to better interpret subtle facial expressions. This refinement will contribute to a more precise understanding of users' emotional states, thus improving the relevance of music recommendations. Advanced machine learning models and techniques, such as deep learning architectures, could be employed to analyze facial features with greater granularity, allowing for more nuanced emotion detection. Additionally, exploring the integration of multimodal approaches, combining facial expression analysis with voice tone recognition, could further enhance the system's ability to capture users' emotional nuances accurately.

**Integration of Machine Learning Techniques:**

Implementing machine learning techniques will empower the system to learn and adapt to individual user preferences over time. By analyzing user interactions and feedback, the system can continually refine its recommendations, ensuring a personalized and evolving music discovery experience for each user. Incorporating reinforcement learning algorithms could enable the system to optimize recommendation strategies based on user feedback, dynamically adjusting its approach to better meet users' evolving preferences and emotional states.

**Exploration of Social Sharing Features:**

Exploring the integration of social sharing features will enable users to share their music recommendations and experiences with friends and followers. By fostering social engagement and interaction, this feature can promote user retention and platform visibility, driving growth and fostering a vibrant community of music enthusiasts. Additionally, incorporating social listening capabilities, where users can discover and listen to music recommended by their social network, could further enhance user engagement and expand the platform's reach organically through word-of-mouth recommendations.

**CHAPTER 8**

**REFERENCES**

[1] Florence, S. Metilda, and M. Uma. "Emotional detection and music recommendation system based on user facial expression." IOP conference series: Materials science and engineering. Vol. 912. No. 6. IOP Publishing, 2020.

[2] Wang, Shu, et al. "A novel emotion-aware hybrid music recommendation method using deep neural network." Electronics 10.15 (2021): 1769

[3] Mahadik, Ankita, et al. "Mood-based music recommendation system." INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10 2021.

[4] Iyer, Aurobind V., et al. "Emotion-based mood-enhancing music recommendation." 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2017.

[5] Sana, S. K., et al. "Facial emotion recognition based music system using convolutional neural networks." Materials Today: Proceedings 62 2022: 4699-4706.

[6] Kim, Hyoung-Gook, Gi Yong Lee, and Min-Soo Kim. "Dual-function integrated emotion-based music classification system using features from physiological signals." IEEE Transactions on Consumer Electronics 67.4 2021: 341-349

[7] Kale, Yash, Sandeep Maurya, and Anisha Prajapati. "A Review on Music Recommendations Based on Facial Expression." i-manager's Journal on Image Processing 9.3 (2022): 41.

[8] Assuncao, Willian G., Lara SG Piccolo, and Luciana AM Zaina. "Considering emotions and contextual factors in music recommendation: a systematic literature review." Multimedia Tools and Applications 81.6 2022: 8367-8407.

[9] Cheng, Y., Zhang, H., & Liu, J. [9] User Feedback and Engagement in Facial

Emotion Enhanced Music Recommendation Apps.

[10] Kale, Yash, Sandeep Maurya,andAnisha Prajapati."A Review on Music RecommendationsBasedonFacialExpression." i-manager's Journal on Image 2022.

**CHAPTER 9**

**APPENDIX**

**9.1 APPENDIX-A1 (SOURCE CODE)**

**train.py:**

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from keras.optimizers import Adam

from keras.preprocessing.image import ImageDataGenerator

train\_dir = 'data/train'

val\_dir = 'data/test'

train\_datagen = ImageDataGenerator(rescale=1./255)

val\_datagen = ImageDataGenerator(rescale=1./255)

train\_generator = train\_datagen.flow\_from\_directory(

train\_dir,

target\_size = (48,48),

batch\_size = 64,

color\_mode = "grayscale",

class\_mode = 'categorical'

)

val\_generator = val\_datagen.flow\_from\_directory(

val\_dir,

target\_size = (48,48),

batch\_size = 64,

color\_mode = "grayscale",

class\_mode = 'categorical'

)

emotion\_model = Sequential()

emotion\_model.add(Conv2D(32, kernel\_size=(3,3), activation='relu', input\_shape = (48,48,1)))

emotion\_model.add(Conv2D(64, kernel\_size=(3,3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2,2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Conv2D(128, kernel\_size=(3,3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2,2)))

emotion\_model.add(Conv2D(128, kernel\_size=(3,3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2,2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Flatten())

emotion\_model.add(Dense(1024, activation='relu'))

emotion\_model.add(Dropout(0.5))

emotion\_model.add(Dense(7, activation='softmax'))

emotion\_model.compile(loss='categorical\_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])

emotion\_model\_info = emotion\_model.fit\_generator(

train\_generator,

steps\_per\_epoch = 28709 // 64,

epochs=75,

validation\_data = val\_generator,

validation\_steps = 7178 // 64

)

emotion\_model.save\_weights('model.h5')

**Spotify.py:**

import spotipy

from spotipy.oauth2 import SpotifyClientCredentials

import pandas as pd

import time

auth\_manager= SpotifyClientCredentials(client\_id='8765d12ccfc14caca9fc5bf24991521e', client\_secret='aa05b3da9e4b48f78e4b4e3804fe70ec')

sp = spotipy.Spotify(auth\_manager=auth\_manager)

def getTrackIDs(user, playlist\_id):

track\_ids = []

playlist = sp.playlist\_tracks(playlist\_id)

for item in playlist['items']:

track = item['track']

track\_ids.append(track['id'])

return track\_ids

def getTrackFeatures(id):

track\_info = sp.track(id)

name = track\_info['name']

album = track\_info['album']['name']

artist = track\_info['album']['artists'][0]['name']

return name, album, artist

def fetch\_playlist\_tracks(emotion\_name, playlist\_id):

track\_ids = getTrackIDs('spotify', playlist\_id)

track\_list = []

for track\_id in track\_ids:

time.sleep(0.5) # Add a delay to avoid rate limiting

track\_data = getTrackFeatures(track\_id)

track\_list.append(track\_data)

df = pd.DataFrame(track\_list, columns=['Name', 'Album', 'Artist'])

df.to\_csv(f'songs/{emotion\_name}.csv', index=False)

print(f"CSV Generated for {emotion\_name} playlist")

emotion\_playlists = {

'Angry': '0l9dAmBrUJLylii66JOsHB',

'Disgusted': '1n6cpWo9ant4WguEo91KZh',

'Fearful': '4cllEPvFdoX6NIVWPKai9I',

'Happy': '0deORnapZgrxFY4nsKr9JA',

'Neutral': '4kvSlabrnfRCQWfN0MgtgA',

'Sad': '1n6cpWo9ant4WguEo91KZh',

'Surprised': '37i9dQZEVXbMDoHDwVN2tF'

}

# Fetch tracks for each emotion playlist and save to CSV

for emotion, playlist\_id in emotion\_playlists.items():

fetch\_playlist\_tracks(emotion, playlist\_id)

**Camera.py**

import numpy as np

import cv2

from PIL import Image

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, Flatten

from tensorflow.keras.layers import Conv2D

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from pandastable import Table, TableModel

from tensorflow.keras.preprocessing import image

import datetime

from threading import Thread

# from Spotipy import \*

import time

import pandas as pd

face\_cascade=cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")

ds\_factor=0.6

emotion\_model = Sequential()

emotion\_model.add(Conv2D(32,kernel\_size=(3,3),activation='relu', input\_shape=(48,48,1)))

emotion\_model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Flatten())

emotion\_model.add(Dense(1024, activation='relu'))

emotion\_model.add(Dropout(0.5))

emotion\_model.add(Dense(7, activation='softmax'))

emotion\_model.load\_weights('model.h5')

cv2.ocl.setUseOpenCL(False)

emotion\_dict= {0:"Angry",1:"Disgusted",2:"Fearful",3:"Happy",4:"Neutral",5:"Sad",6:"Surprised"}

music\_dist={0:"songs/angry.csv",1:"songs/disgusted.csv ",2:"songs/fearful.csv",3:"songs/happy.csv",4:"songs/neutral.csv",5:"songs/sad.csv",6:"songs/surprised.csv"}

global last\_frame1

last\_frame1 = np.zeros((480, 640, 3), dtype=np.uint8)

global cap1

show\_text=[0]

''' Class for calculating FPS while streaming. Used this to check performance of using another thread for video streaming '''

class FPS:

def \_\_init\_\_(self):

# store the start time, end time, and total number of frames

# that were examined between the start and end intervals

self.\_start = None

self.\_end = None

self.\_numFrames = 0

def start(self):

# start the timer

self.\_start = datetime.datetime.now()

return self

def stop(self):

# stop the timer

self.\_end = datetime.datetime.now()

def update(self):

# increment the total number of frames examined during the

# start and end intervals

self.\_numFrames += 1

def elapsed(self):

# return the total number of seconds between the start and

# end interval

return (self.\_end - self.\_start).total\_seconds()

def fps(self):

# compute the (approximate) frames per second

return self.\_numFrames / self.elapsed()

''' Class for using another thread for video streaming to boost performance '''

class WebcamVideoStream:

def \_\_init\_\_(self, src=0):

self.stream = cv2.VideoCapture(src,cv2.CAP\_DSHOW)

(self.grabbed, self.frame) = self.stream.read()

self.stopped = False

def start(self):

# start the thread to read frames from the video stream

Thread(target=self.update, args=()).start()

return self

def update(self):

# keep looping infinitely until the thread is stopped

while True:

# if the thread indicator variable is set, stop the thread

if self.stopped:

return

# otherwise, read the next frame from the stream

(self.grabbed, self.frame) = self.stream.read()

def read(self):

# return the frame most recently read

return self.frame

def stop(self):

# indicate that the thread should be stopped

self.stopped = True

''' Class for reading video stream, generating prediction and recommendations '''

class VideoCamera(object):

def get\_frame(self):

global cap1

global df1

cap1 = WebcamVideoStream(src=0).start()

image = cap1.read()

image=cv2.resize(image,(600,500))

gray=cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

face\_rects=face\_cascade.detectMultiScale(gray,1.3,5)

df1 = pd.read\_csv(music\_dist[show\_text[0]])

df1 = df1[['Name','Album','Artist']]

df1 = df1.head(15)

for (x,y,w,h) in face\_rects:

cv2.rectangle(image,(x,y-50),(x+w,y+h+10),(0,255,0),2)

roi\_gray\_frame = gray[y:y + h, x:x + w]

cropped\_img= np.expand\_dims(np.expand\_dims(cv2.resize(roi\_gray\_frame, (48, 48)), -1), 0)

prediction = emotion\_model.predict(cropped\_img)

maxindex = int(np.argmax(prediction))

show\_text[0] = maxindex

#print(df1)

cv2.putText(image,emotion\_dict[maxindex],(x+20,y-60), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE\_AA)

df1 = music\_rec()

global last\_frame1

last\_frame1 = image.copy()

pic = cv2.cvtColor(last\_frame1, cv2.COLOR\_BGR2RGB)

img = Image.fromarray(last\_frame1)

img = np.array(img)

ret, jpeg = cv2.imencode('.jpg', img)

return jpeg.tobytes(), df1

def music\_rec():

# print('---------------- Value ------------', music\_dist[show\_text[0]])

df = pd.read\_csv(music\_dist[show\_text[0]])

df = df[['Name','Album','Artist']]

df = df.head(15)

return df

**Utils.py:**

''' Class for using separate thread for video streaming through web camera'''

import cv2

from threading import Thread

class WebcamVideoStream:

def \_\_init\_\_(self, src=0):

self.stream = cv2.VideoCapture(src,cv2.CAP\_DSHOW)

(self.grabbed, self.frame) = self.stream.read()

self.stopped = False

def start(self):

# start the thread to read frames from the video stream

Thread(target=self.update, args=()).start()

return self

def update(self):

# keep looping infinitely until the thread is stopped

while True:

# if the thread indicator variable is set, stop the thread

if self.stopped:

return

# otherwise, read the next frame from the stream

(self.grabbed, self.frame) = self.stream.read()

def read(self):

# return the frame most recently read

return self.frame

def stop(self):

# indicate that the thread should be stopped

self.stopped = True

**App.py:**

from flask import Flask, render\_template, Response, jsonify

import gunicorn

from camera import \*

app = Flask(\_\_name\_\_)

headings = ("Name","Album","Artist")

df1 = music\_rec()

df1 = df1.head(15)

@app.route('/')

def index():

print(df1.to\_json(orient='records'))

return render\_template('index.html', headings=headings, data=df1)

def gen(camera):

while True:

global df1

frame, df1 = camera.get\_frame()

yield (b'--frame\r\n'

b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video\_feed')

def video\_feed():

return Response(gen(VideoCamera()),

mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/t')

def gen\_table():

return df1.to\_json(orient='records')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/contact')

def contact():

return render\_template('contact.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.debug = True

app.run()

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Emotion Music Recommendation</title>

<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400&display=swap" rel="stylesheet">

<link rel="stylesheet" href="/static/css/style.css">

</head>

<body>

<div class="navbar">

<a href="index.html">Home</a>

<a href="/about">Team</a>

<a href="/contact">Contact Us</a>

</div>

<div class="headings"><h1>Emotion Based Music Recommender</h1></div>

<div>

<ul class="example-2">

<li class="icon-content">

<a

href="https://www.spotify.com/"

aria-label="Spotify"

data-social="spotify">

<div class="filled"></div>

</a>

<div class="tooltip">Spotify</div>

</li>

<li class="icon-content">

<a href="https://music.youtube.com/aria-label="Pinterest"

data-social="pinterest">

<div class="filled"></div>

</a>

<div class="tooltip">Youtube</div>

</li>

<li class="icon-content">

<a

href="https://music.amazon.in/"

aria-label="Dribbble"

data-social="dribbble">

<div class="filled"></div>

</a>

<div class="tooltip">Amazon</div>

</li>

<li class="icon-content">

<a

href="https://music.apple.com/us/browse"

aria-label="Apple Music"

data-social="Apple Music">

<div class="filled"></div>

</a>

<div class="tooltip">Apple</div>

</li>

</ul>

</div>

<div id="body">

<div class="emotion-detector">

<div>

<h2>Emotion Detector</h2>

<div>

<img id="bg" src="{{ url\_for('video\_feed') }}" alt="Emotion Detector Image" />

</div>

</div>

</div>

<div class="song-recommendations">

<h2>Music Recommendations</h2>

<div id="ResultArea"></div>

</div>

</div>

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

<script type="text/javascript">

// Constantly Update Table

setInterval(function () {

$.getJSON('/t', function (data) {

CreateHtmlTable(data);

console.log(data, "DATA");

});

return false;

}, 100);

function CreateHtmlTable(data) {

$("#ResultArea").empty(); // Use empty to clear the div

// Create table if not existing

var table = $("#DynamicTable");

if (table.length === 0) {

table = $("<table class='table table-striped table-light table-bordered table-hover table-smtable-responsive' id='DynamicTable'></table>").appendTo("#ResultArea");

var rowHeader = $("<tr></tr>").appendTo(table);

$("<th></th>").text("Name").appendTo(rowHeader);

$("<th></th>").text("Album").appendTo(rowHeader);

$("<th></th>").text("Artist").appendTo(rowHeader);

}

table.find("tr:gt(0)").remove();

// Populate table with new data

$.each(data, function (i, value) {

var row = $("<tr></tr>").appendTo(table);

$("<td></td>").text(value.Name).appendTo(row);

$("<td></td>").text(value.Album).appendTo(row);

$("<td></td>").text(value.Artist).appendTo(row);

});

}

</script>

</body>

</html>

**About.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="https://cdn.tailwindcss.com"></script>

<link rel="stylesheet" href="/static/css/about.css">

<linkrel="stylesheet" href="https://editor.analyticsvidhya.com/uploads/88060Learn-Facial-Recognition-scaled.jpg">

</head>

<body>

<div class="navbar ">

<a href="index.html">Home</a>

<a href="/about">Team</a>

<a href="/contact">Contact Us</a>

</div>

<div class="flex justify-center items-center">

<div class="flex min-h-screen relative items-center justify-center w-full">

<div class="rounded-xl overflow-hidden relative text-center p-4 group items-center flex flex-col max-w-sm hover:shadow-2xl transition-all duration-500 shadow-xl">

<div class="text-gray-500 group-hover:scale-105 transition-all">

<imgclass="rounded-2xlh-80w-96object-cover" src="https://images.pexels.com/photos/3974089/pexels-photo-3974089.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1" alt="">

</div>

<div class="group-hover:pb-10 transition-all duration-500 delay-200">

<h1 class="font-semiboldtext-gray-700 mt-3 text-xl">Humayoon Niyaz</h1>

<p class="text-gray-600 text-base">CSE-A 200071601043</p>

</div>

<div class="flex items-center transition-all duration-500 delay-200 group-hover:bottom-3 -bottom-full absolute gap-2 justify-evenly w-full">

<div class="flex gap-3 text-2xl bg-gray-700 text-white p-1 hover:p-2 transition-all duration-500 delay-200 rounded-full shadow-sm">

<aclass="hover:scale-110transition-allduration-500delay-200" href="https://github.com/qazi-humayoon"> </a>

<aclass="hover:scale-110transition-allduration-500delay-200" href="//www.linkedin.com/in/qazi-humayoon-589b86212/"> </a>

<aclass="hover:scale-110transition-allduration-500delay-200" href="//www.linkedin.com/in/qazi-humayoon-589b86212/">

</a>

</div>

</div>

</div>

</div>

<div class="flex min-h-screen relative items-center justify-center w-full">

<div class="rounded-xl overflow-hidden relative text-center p-4 group items-center flex flex-col max-w-sm hover:shadow-2xl transition-all duration-500 shadow-xl">

<div class="text-gray-500 group-hover:scale-105 transition-all">

<imgclass="rounded-2xlh-80w-96object-cover" src="https://images.pexels.com/photos/1136575/pexels-photo-1136575.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1" alt="">

</div>

<div class="group-hover:pb-10 transition-all duration-500 delay-200">

<h1 class="font-semibold text-gray-700 mt-3 text-xl">Fawwaz Numan</h1>

<p class="text-gray-600 text-base">CSE-A 200071601035</p>

</div>

<divclass="flexitems-centertransition-allduration-500delay-200group-hover:bottom-3 -bottom-full absolute gap-2 justify-evenly w-full">

<div class="flex gap-3 text-2xl bg-gray-700 text-white p-1 hover:p-2 transition-all duration-500 delay-200 rounded-full shadow-sm">

<a class="hover:scale-110 transition-all duration-500 delay-200">

</a>

<a class="hover:scale-110 transition-all duration-500 delay-200">

<svgwidth="1em"height="1em"viewBox="002424"stroke="currentColor" fill="none">

<path d="M4 4h16c1.1 0 2 .9 2 2v12c0 1.1-.9 2-2 2H4c-1.1 0-2-.9-2-2V6c0-1.1.9-2 2-2z"></path>

<path d="M22 6l-10 7L2 6"></path>

</svg>

</a>

<a class="hover:scale-110 transition-all duration-500 delay-200">

</svg>

</a>

</div>

</div>

</div>

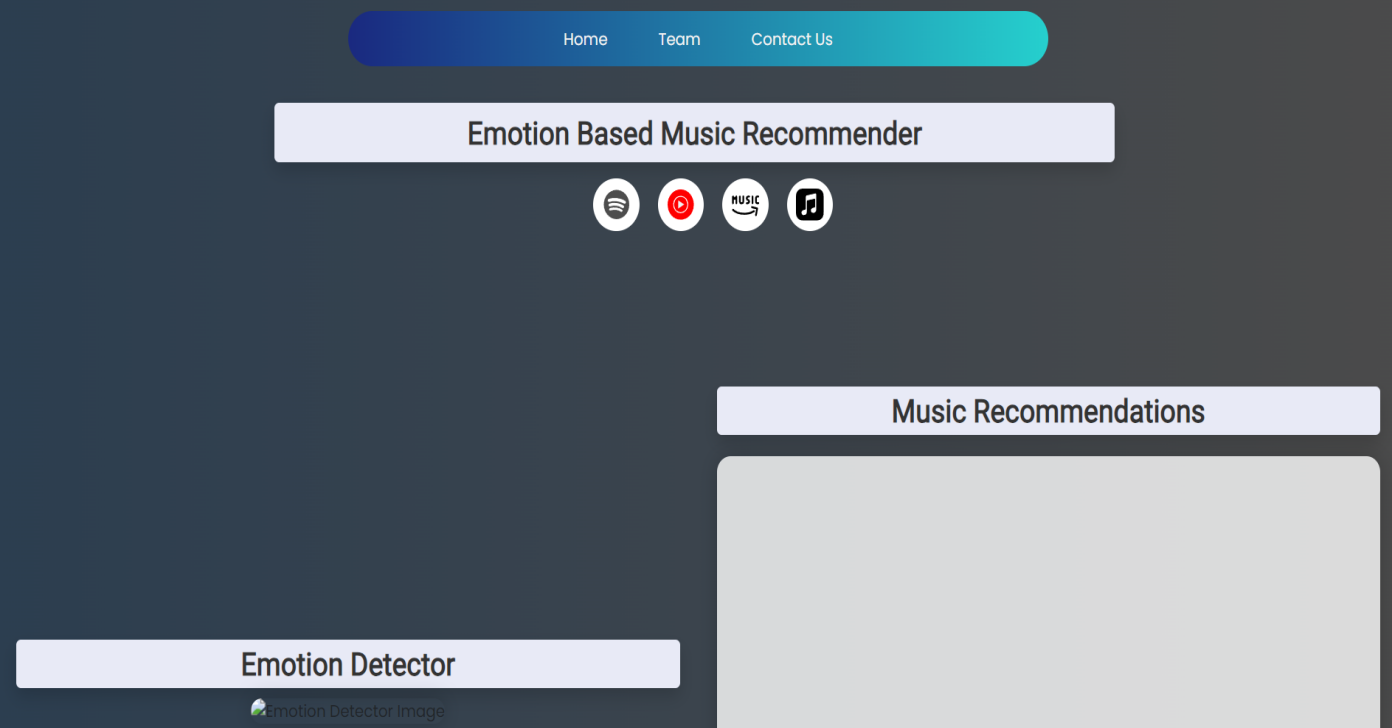
</div>

</div>

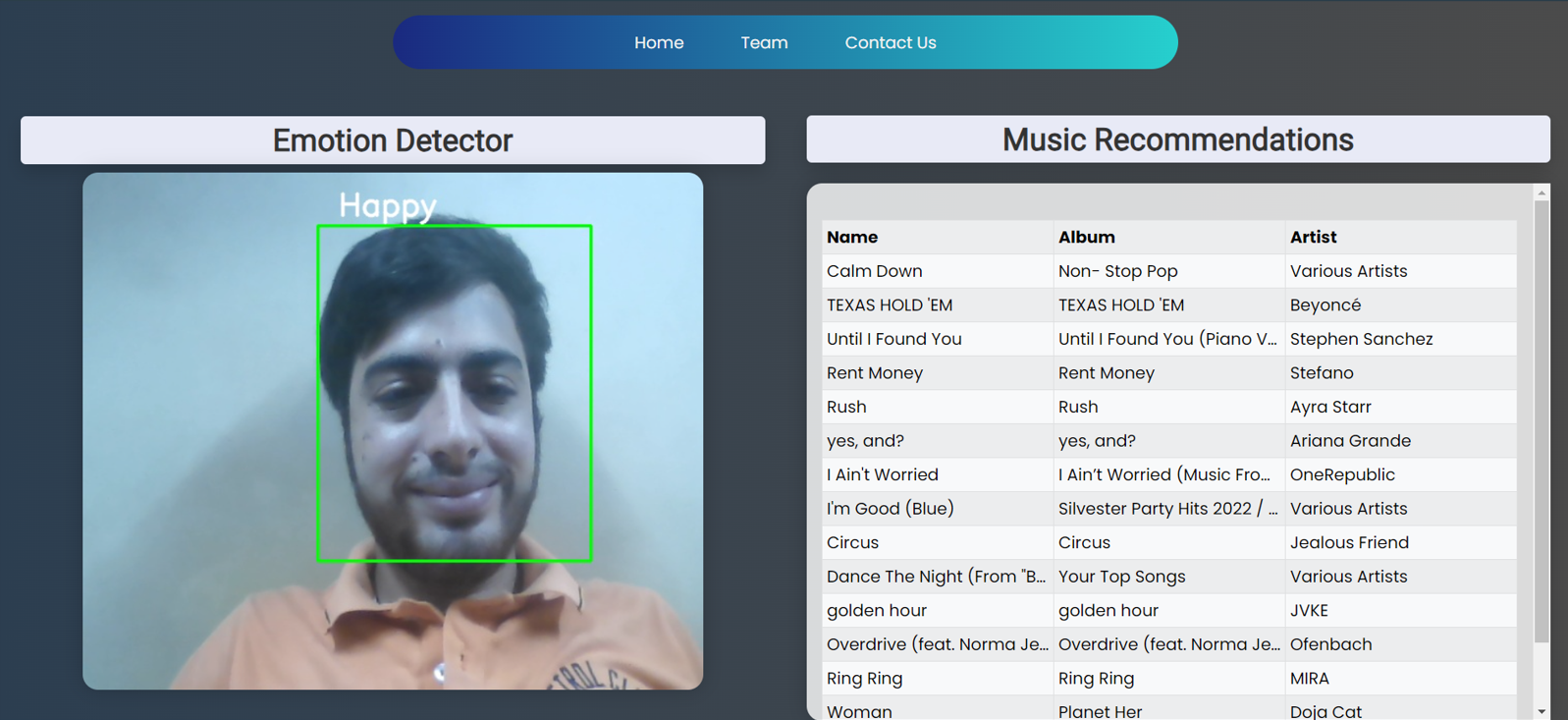
</body>

</html>

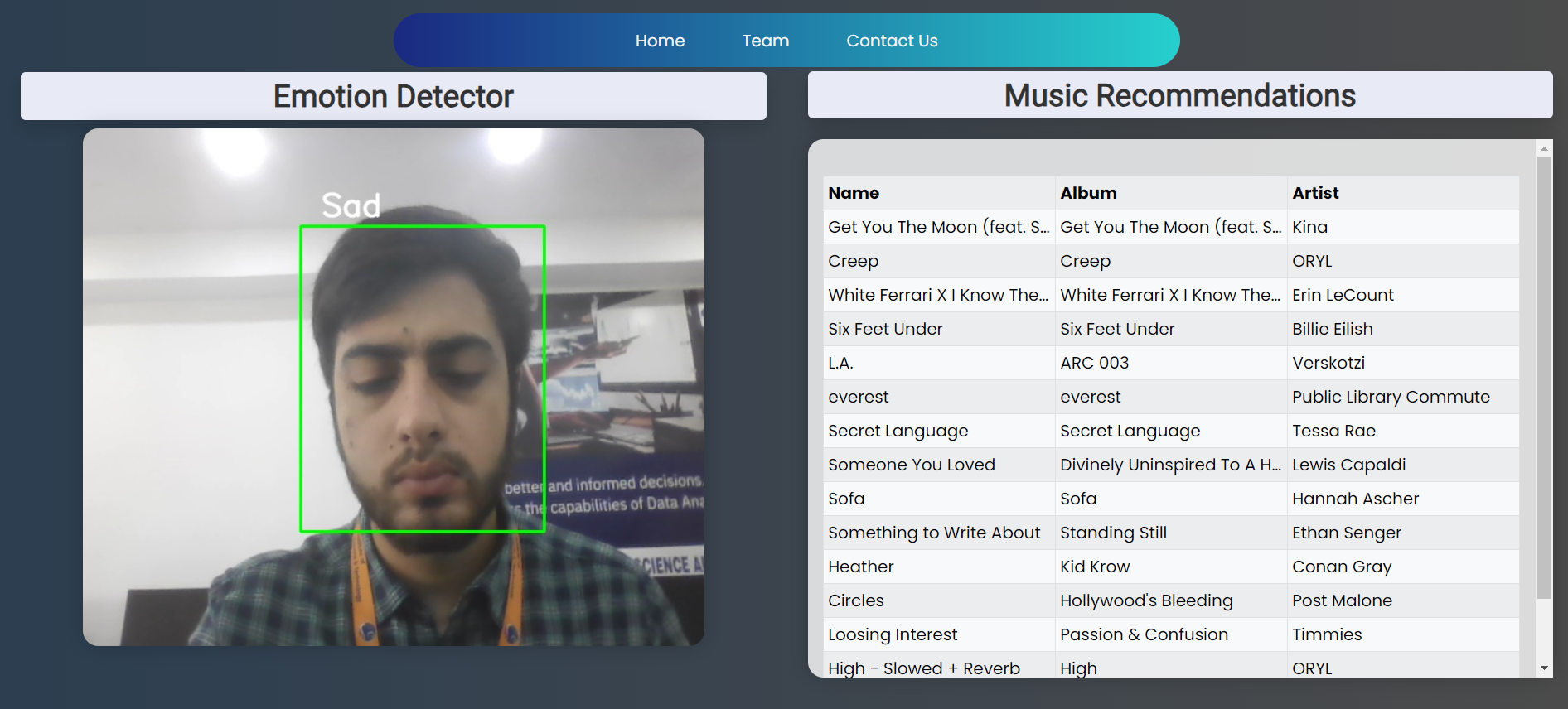
**9.2 APPENDIX-A2 (SCREENSHOTS)**



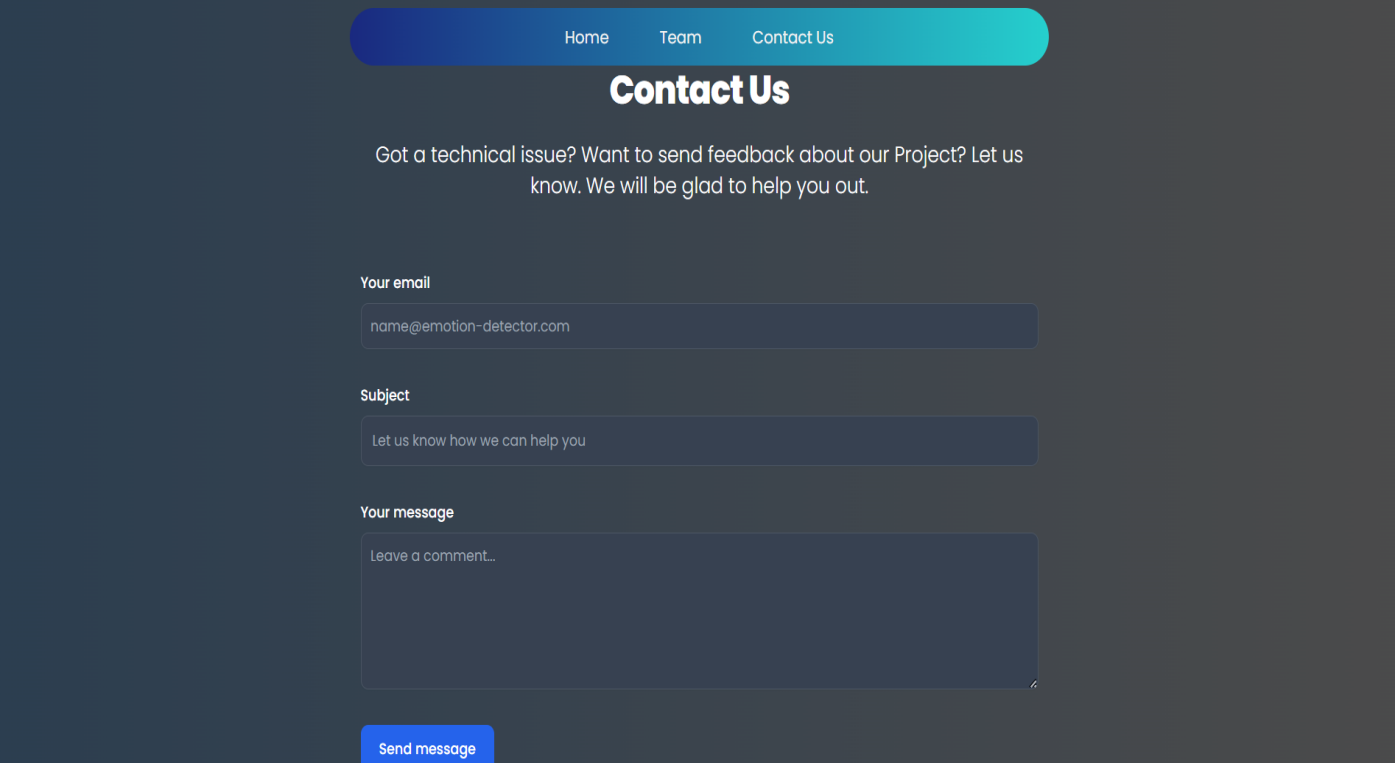
**Figure 8.1 Home Page**



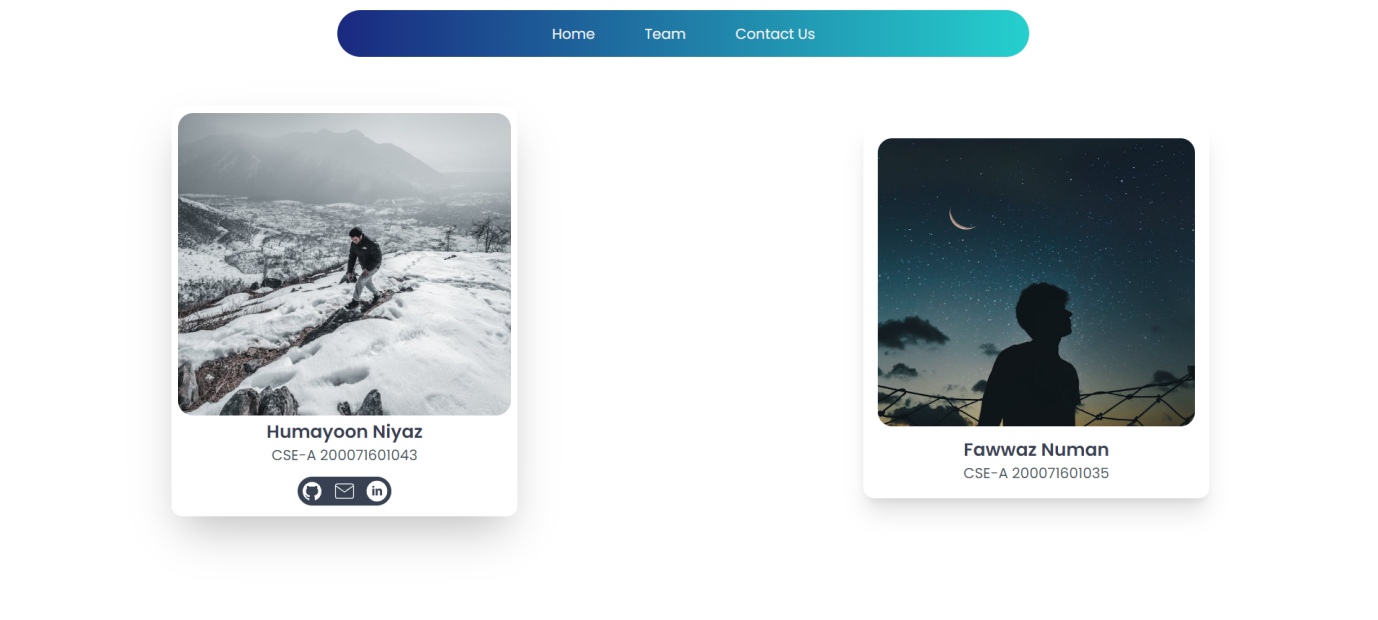
**Figure 8.2 Happy Emotion Detected and Music Recommendation**



**Figure 8.3. Sad Emotion Detected and Music Recommendation**



**Figure 8.4. Contact Us for Feedback**



**Figure 8.5. Team**

**CHAPTER 10**

**TECHNICAL BIOGRAPHY**



Mr. Humayoon Niyaz(200071601043) was born on 06th Sept 2002, in Baramulla, Jammu and Kashmir. He completed his 12th grade in year 2020 Higher Secondary School, Baramulla. He is currently pursuing his Bachelor of Technology Degree in Computer Science and Engineering at B. S. Abdur Rahman Crescent Institute of Science and Technology. His areas of interest include Web Development, Artificial Intelligence, Data Science and Machine Learning, Data Analytics. His e-mail address is qazi.humayoon687@gmail.com



Mr. Fawwaz Numan (200071601035)was born on 24th July 2000, in Chennai, Tamil Nadu. He completed his 12th grade in year 2020 SBOA Matriculation higher secondary school, Chennai. He is currently pursuing his Bachelor of Technology Degree in Computer Science and Engineering at B. S. Abdur Rahman Crescent Institute of Science and Technology. His areas of interest include Web Development, Artificial Intelligence, Data Science and Machine Learning. His e-mail address is fawwaznuman560@gmail.com

Project Report 2- Batch

*by* karthiga cse

**Submission date:** 10-May-2024 11:57AM (UTC+0530)

**Submission ID:** 2219165372 **File name:** 4.pdf (8.63M) **Word count:** 75705

**Character count:** 460678