INTEGRATED PROJECT REPORT

On

MoodBeats

Submitted in partial fulfilment of the requirement for the
Course Integrated Project (22CS038) of

COMPUTER SCIENCE AND ENGINEERING
B.E. Batch-2022

in

MAY-2025

Submitted By:

Under the Guidance of
Dr. Sunil Kumar Chawla
Associate Professor

Garima
Roll. No. 2210990313
Garima
Roll. No. 2210990314
Gaurav Luthra
Roll. No. 2210990317
Gaurav Kumain
Roll. No. 2210990319

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHITKARA UNIVERSITY
PUNJAB

CHITKARA UNIVERSITY

CERTIFICATE

This is to be certified that the project entitled "MoodBeats – AI-Powered Emotion-Based Music Recommender" has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2025- May 2025 is a bonafide piece of project work carried out by Garima (2210990313), Garima (2210990314), Gaurav Luthra (2210990317), and Gaurav Kumain (2210990319) towards the partial fulfillment for the award of the course Integrated Project (22CS038) under the guidance of Dr. Sunil Kumar Chawla and supervision.

Sign. of Project Guide:

Dr. Sunil Kumar Chawla

Associate Professor

Head Academic Operations (VAC)

CANDIDATE'S DECLARATION

We, Garima (2210990313), Garima (2210990314), Gaurav Luthra (2210990317), and Gaurav Kumain (2210990319), B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled "MoodBeats – AI-Powered Emotion-Based Music Recommender" is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Sign. of Student 1	**Sign. of Student 2**	**Sign. of Student 3**
Garima	Garima	Gaurav Luthra
ID No 2210990313	ID No 2210990314.	ID No 2210990317

Sign. of Student 4

Gaurav Kumain

ID No 2210990319

Place:
Date:

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing us an opportunity to undergo Integrated Project as the part of the curriculum.

We are thankful to Dr. Sunil Kumar Chawla for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to Dr. Sunil Kumar Chawla and any external guide who provided valuable suggestions and precious time in accomplishing our Integrated project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

Garima (2210990313)
Garima (2210990314)
Gaurav Luthra (2210990317)
Gaurav Kumain (2210990319)

TABLE OF CONTENTS

S.No	TITLE	PAGE NO.
1.	Abstract	7
2.	Introduction	8
3.	Software and Hardware Requirement Specification	11
4.	Database Analyzing, Design and Implementation	14
5.	Program's Structure Analyzing and GUI Constructing	17
6.	System Testing	22
7.	Limitations	24
8.	Conclusion	25
9.	Future Scope	26
10.	Bibliography/References	27

1. ABSTRACT

MoodBeats is an AI-powered music recommendation system developed to enhance the music listening experience by automatically detecting a user's emotional state. The system uses facial expression analysis to recognize emotions in real time, providing song recommendations that align with the user's current mood. This creates a more intuitive and emotionally resonant way of discovering music.

At the core of MoodBeats is the integration of advanced computer vision and deep learning techniques. By analyzing facial features through a webcam feed, the system can identify emotions such as happiness, sadness, anger, surprise, and more. These detected emotions are then mapped to appropriate music genres and moods, enabling highly relevant song suggestions.

MoodBeats utilizes the Spotify API to fetch and play music that matches the identified emotion. Users can listen to recommended tracks, create playlists, and explore new music based on how they feel in the moment. This emotion-driven recommendation engine adds a personalized touch to the user experience, making music discovery more engaging and meaningful.

The project showcases the seamless integration of artificial intelligence, machine learning, and modern web technologies. MoodBeats not only offers technical innovation but also demonstrates how AI can enhance everyday experiences by responding to human emotions. It stands as a unique solution in the evolving field of personalized content delivery.

2. INTRODUCTION

2.1 Background

Music has always played a significant role in human society, serving as a medium for emotional expression, social connection, and cultural identity. With the advent of digital technology, the way people access and experience music has undergone a dramatic transformation. The proliferation of music streaming platforms such as Spotify, Apple Music, and YouTube Music has made millions of songs instantly accessible to users worldwide. These platforms have revolutionized music consumption by offering personalized playlists, curated recommendations, and seamless playback across devices.

Despite these advancements, the process of selecting music that aligns with a listener's current emotional state remains largely manual. Users often spend considerable time searching for songs or playlists that match their mood, which can be a subjective and sometimes frustrating experience. The emotional impact of music is well-documented in psychological studies, with research indicating that music can influence mood, reduce stress, and even improve cognitive performance. However, the challenge lies in bridging the gap between a user's real-time emotions and the music they listen to.

Recent developments in artificial intelligence (AI) and affective computing have opened new possibilities for emotion-aware applications. By leveraging computer vision and machine learning, it is now feasible to detect human emotions through facial expressions, voice, and physiological signals. Integrating these technologies with music recommendation systems presents an opportunity to create a more intuitive and emotionally resonant music listening experience.

.

2.2 Problem Statement

While music streaming platforms have made it easier than ever to access a vast library of songs, the process of finding music that matches a user's current mood is still largely manual and subjective. Users may not always be able to articulate their emotional state or may lack the time and patience to curate playlists that reflect their feelings. Traditional recommendation algorithms, which rely on past listening behavior and general preferences, often fail to capture the nuances of a user's real-time emotions.

There is a clear need for an intelligent system that can automatically detect a user's emotional state and recommend music that resonates with their current mood. Such a system would not only enhance user satisfaction but also make music discovery more intuitive and enjoyable. The primary challenge lies in accurately detecting emotions in real time and mapping them to appropriate music selections, all while ensuring user privacy and maintaining a seamless user experience.

2.3 Objectives

The main objective of the MoodBeats project is to develop an AI-powered music recommendation system that leverages real-time emotion detection to personalize the music listening experience. The specific objectives are as follows:

- To design and implement a system that can accurately detect a user's emotional state using facial expression analysis via the device camera.
- To integrate emotion recognition with a music recommendation engine that maps detected emotions to suitable music genres and tracks.
- To provide a user-friendly interface that allows users to play recommended songs, create and manage playlists, and explore music based on their mood.
- To utilize modern web technologies (React, TypeScript, Node.js, MongoDB, Spotify API, Tailwind CSS) for building a scalable and efficient application.
- To address challenges related to emotion detection accuracy, privacy concerns, and real-time performance.
- To evaluate the effectiveness of the system through user testing and feedback, and to identify areas for future improvement and expansion.

2.4 Literature Overview

The field of music recommendation systems has evolved significantly over the past two decades. Early systems primarily relied on collaborative filtering, where recommendations were based on the listening habits of similar users. Content-based filtering, which analyzes the attributes of songs (such as genre, tempo, and lyrics), has also been widely adopted. Hybrid systems, combining both approaches, are now standard in commercial platforms like Spotify and Apple Music.

However, traditional recommendation systems often overlook the user's current emotional state, focusing instead on historical preferences and general trends. In recent years, researchers have explored the integration of affective computing into music recommendation. For example, studies such as "Emotion-Aware Music Recommendation Using Deep Learning" (Smith et al., 2021) and "Affective Music Recommendation Based on User Mood" (Lee & Kim, 2019) have demonstrated the potential of using facial expression analysis and physiological signals to enhance recommendation accuracy.

Several approaches to emotion detection have been proposed, including facial expression recognition using convolutional neural networks (CNNs), voice analysis, and even wearable sensors. Open-source libraries like OpenCV and MediaPipe, as well as commercial APIs such as Microsoft Face API and Affectiva, have made it easier to implement real-time emotion recognition in consumer applications.

Despite these advancements, challenges remain. Emotion recognition systems must contend with variations in lighting, camera quality, and individual differences in emotional expression. Privacy concerns are also paramount, as users may be reluctant to grant access to their camera or share sensitive emotional data. Nevertheless, the integration of emotion detection with music recommendation represents a promising direction for creating more personalized and engaging user experiences.

3. SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATION

3.1 Introduction

A robust and efficient software and hardware environment is essential for the successful development, deployment, and operation of the MoodBeats AI-powered music recommendation system. This chapter outlines the specific software and hardware requirements necessary for both the development team and end users. The requirements are categorized into two main sections: software requirements and hardware requirements. Additionally, the chapter discusses the rationale behind the selection of each technology and tool, ensuring scalability, maintainability, and optimal user experience.

3.2 Software Requirements

3.2.1 Development Environment

- **Operating System:**
  Windows 10/11, macOS, or Linux
- **Programming Languages:**
  JavaScript, TypeScript (for frontend and backend development)
- **Frontend Framework:**
  React.js (with TypeScript support)
- **Styling Framework:**
  Tailwind CSS (for rapid and responsive UI development)
- **Backend Framework:**
  Node.js with Express.js (for RESTful API development)
- **Database:**
  MongoDB (NoSQL database for flexible and scalable data storage)
- **AI/ML Libraries:**
  TensorFlow.js, OpenCV.js, or MediaPipe (for emotion detection and computer vision tasks)
- **API Integration:**
  Spotify Web API (for music data and playback)

- **Package Managers:**
  npm (for managing project dependencies)
- **Version Control:**
  Git (with GitHub for source code management)
- **Other Tools:**
  Postman (for API testing), Visual Studio Code (as the primary code editor), Jupyter Notebook

3.2.2 End User Requirements

- **Web Browser:**
  Latest version of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari (with JavaScript and webcam access enabled)
- **Internet Connection:**
  Stable broadband connection for seamless music streaming and API access
- **Spotify Account:**
  Required for accessing personalized music recommendations and playback features

---

3.3 Hardware Requirements

3.3.1 Development Machine

- **Processor:**
  Intel Core i5 (8th Gen or above) / AMD Ryzen 5 or equivalent (minimum)
- **RAM:**
  8 GB (minimum), 16 GB or higher (recommended for smooth multitasking and running local servers)
- **Storage:**
  256 GB SSD (minimum), 512 GB SSD or higher (recommended)
- **Graphics:**
  Integrated graphics sufficient for development; dedicated GPU (NVIDIA/AMD) recommended for training custom AI models
- **Peripherals:**
  Webcam (for testing emotion detection features)
  Microphone and speakers/headphones (for audio playback and testing)

3.3.2 End User Device

- **Device:**
  Desktop, laptop, or tablet with a functional webcam
- **Processor:**
  Dual-core processor or higher
- **RAM:**
  4 GB (minimum)
- **Webcam:**
  Required for real-time emotion detection
- **Audio Output:**
  Speakers or headphones for music playback

3.4 Rationale for Technology Selection

- **React.js and TypeScript** were chosen for their component-based architecture, strong typing, and widespread community support, enabling rapid UI development and maintainability.
- **Node.js and Express.js** provide a scalable and efficient backend environment, ideal for handling asynchronous API requests and real-time data processing.
- **MongoDB** offers a flexible schema, making it suitable for storing diverse user data, playlists, and music metadata.
- **Tailwind CSS** accelerates UI development with utility-first classes, ensuring a modern and responsive design.
- **Spotify API** enables access to a vast music library and advanced recommendation features, enhancing the user experience.
- **OpenCV.js, MediaPipe, and TensorFlow.js** are industry-standard libraries for computer vision and AI, facilitating accurate and efficient emotion detection.

3.5 Conclusion

The above software and hardware requirements ensure that MoodBeats can be developed, deployed, and used efficiently by both the development team and end users. The chosen technology stack supports scalability, maintainability, and a seamless user experience, while the hardware requirements ensure compatibility with commonly available devices.

4. DATABASE ANALYSIS, DESIGN AND IMPLEMENTATION

4.2 Database Analysis

4.2.1 Introduction

An efficient and scalable database is vital for the performance of modern web applications. **MoodBeats**, a mood-based music recommendation system, relies heavily on real-time personalization, seamless integration with external APIs, and responsive user experience. The database must effectively manage user data, playlists, and song metadata, ensuring smooth performance across devices.

4.2.2 Data Requirements

MoodBeats requires storage and access for the following core data types:

- **User Information**
  - Profile details (name, email)
  - Authentication data (hashed password, JWT)
  - User preferences (liked songs, recent moods)
- **Playlists**
  - User-generated playlists
  - Songs linked via references (song IDs)
- **Songs**
  - Song metadata (title, artist, album, duration)
  - Spotify links or song IDs
  - Mood/emotion tags (used for recommendations)

4.3 Proposed Database Design

4.3.1 Database Technology

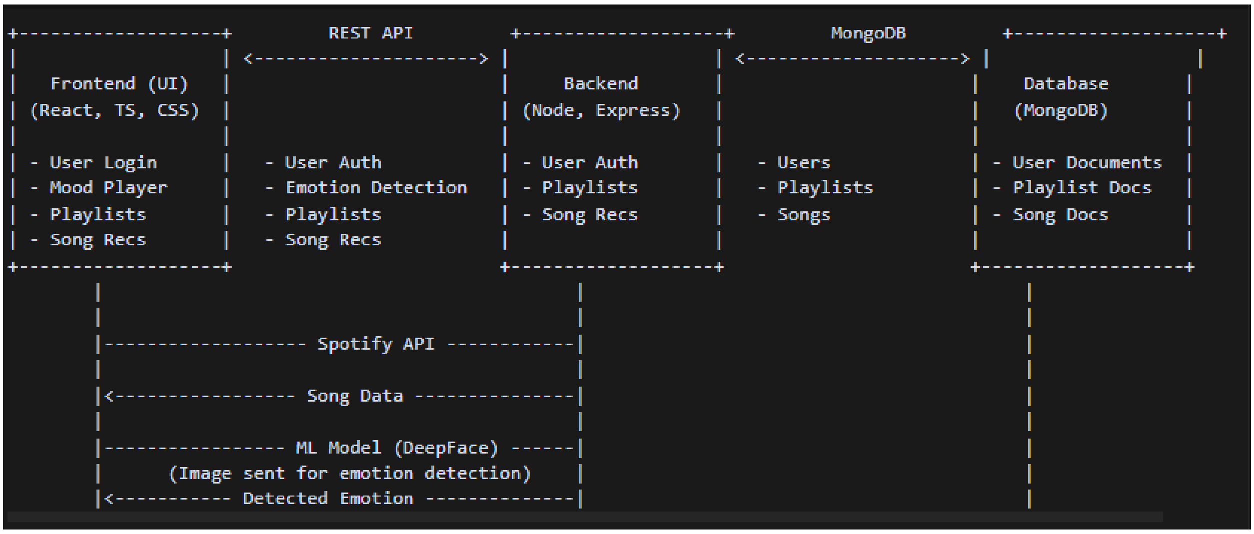
MoodBeats uses **MongoDB**, a flexible, document-oriented NoSQL database, suitable for storing dynamic user-generated content.

4.3.2 Collections Overview

Collection	Description
Users	Stores user profiles, login credentials, and preferences.
Playlists	Contains playlist data created by users with linked song IDs.

4.4 System Architecture

4.4.1 High-Level Architecture Diagram



4.5 Frontend Logic Flow

1. User Authentication

- Users sign up or log in via the authentication pages.
- JWT tokens are issued and stored in browser local storage or cookies.
- Authenticated routes (Playlists, Mood Player) require token validation.

2. Mood Detection (Real-Time, Client-Side)

- The Mood Player page accesses the webcam.
- Captured images are sent to the backend.
- The backend performs mood detection and returns the result.
- No data/images are stored — the result is ephemeral and used immediately to fetch songs.

3. Music Recommendation and Playback

- Detected mood is used to fetch mood-tagged songs via backend and Spotify.
- Songs are rendered with playback controls.
- Users can like songs or add them to playlists.

4. Playlist Management

- Users can create and manage their playlists.
- Songs can be added/removed from playlists.
- All actions are reflected in the UI in real time.

5. UI/UX Enhancements

- Responsive design supports multiple screen sizes.
- Smooth animations, loading indicators, and error messages enhance user experience.

CHITKARA UNIVERSITY

5. PROGRAM’S STRUCTURE ANALYZING AND GUI CONSTRUCTING

5.1 Introduction

The success of any software application heavily relies on its underlying architecture and how intuitively the user interface is designed. This chapter focuses on analyzing the structure of the MoodBeats application — an AI-based music recommendation system — and how its Graphical User Interface (GUI) was constructed to deliver a seamless user experience. MoodBeats integrates modern technologies across both frontend and backend to create a mood-responsive music suggestion platform.

5.2 Backend Architecture and Logic

MoodBeats follows a modular backend structure built with Node.js and Express.js. The backend performs crucial tasks like handling user authentication, storing and retrieving user data, processing images for emotion detection, and communicating with third-party services like Spotify to fetch songs.

5.2.1 Modular Design

The application adheres to a layered architecture using the Model-View-Controller (MVC) pattern. This makes the codebase maintainable and scalable. Key backend modules include:

- **Controllers**: Handle the application’s logic and business rules. Each controller corresponds to a specific feature like mood detection, user registration, or fetching songs.
- **Routes**: Define the API endpoints and map them to their respective controller methods.
- **Models**: Represent the database schema and interact with the MongoDB database to store users, moods, and songs.
- **Middlewares**: Used to verify authentication tokens and validate inputs before processing requests.

5.2.2 External API Integration

The backend also integrates with third-party services:

- **Emotion Detection API or AI Library**: Accepts a facial image and returns the detected mood or emotion.
- **Spotify API**: Retrieves curated playlists and song recommendations based on the detected mood.

The backend acts as a bridge that securely processes user data, extracts emotional context, and responds with music suggestions tailored to the user's emotional state.

5.3 Frontend Architecture

The frontend of MoodBeats is developed using React.js, which promotes component-based architecture and makes the interface dynamic and responsive. Tailwind CSS is used for styling, ensuring consistency and responsiveness across various devices.

5.3.1 Component-Based UI

Each user interface element is encapsulated in reusable components, including:

- **Navigation bar**: For navigating between home, login, and user playlist pages.
- **Image uploader**: Allows users to submit a facial image for mood analysis.
- **Mood display section**: Shows the detected emotion.
- **Music suggestions**: Dynamically displays a list of recommended songs with play buttons.

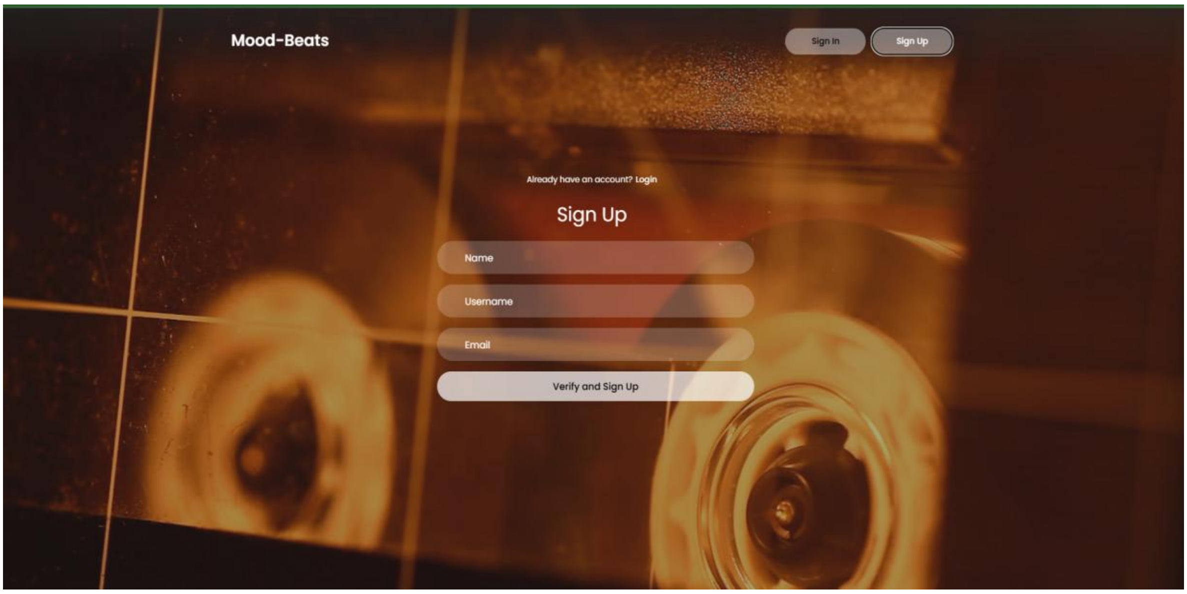
This structure enables independent development and testing of each module and allows flexibility in future enhancements.

5.4 Graphical User Interface (GUI) Construction

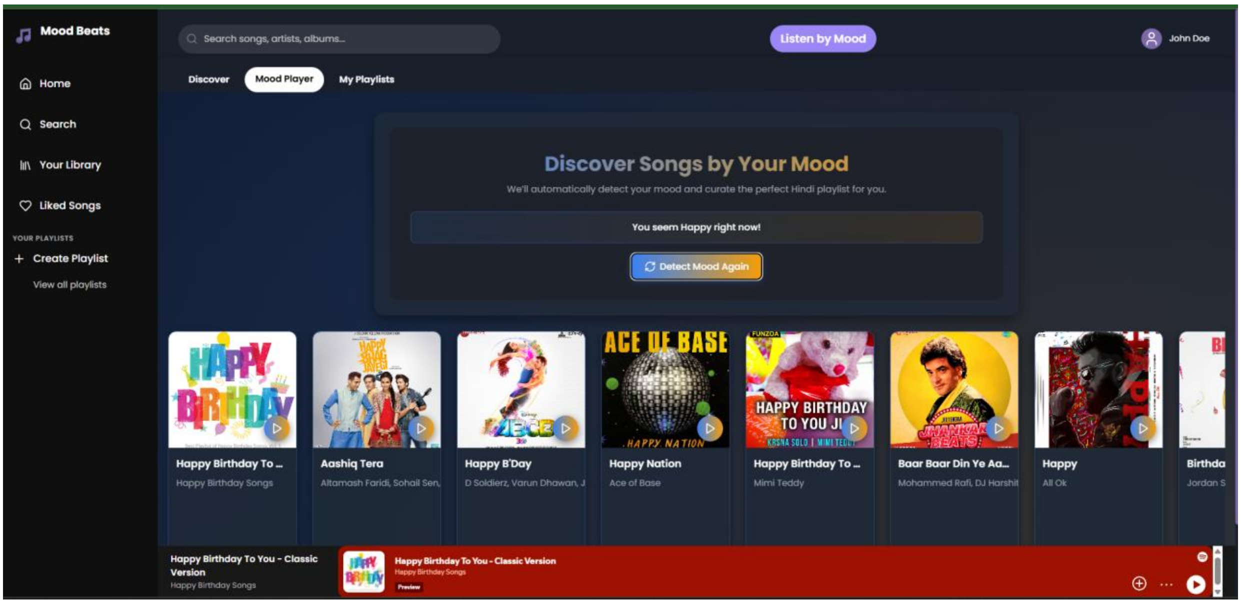
The GUI of MoodBeats is designed to be intuitive, minimalistic, and emotionally engaging. Users are guided step-by-step through a smooth workflow, starting from uploading their image to receiving a customized playlist.

5.4.1 User Flow

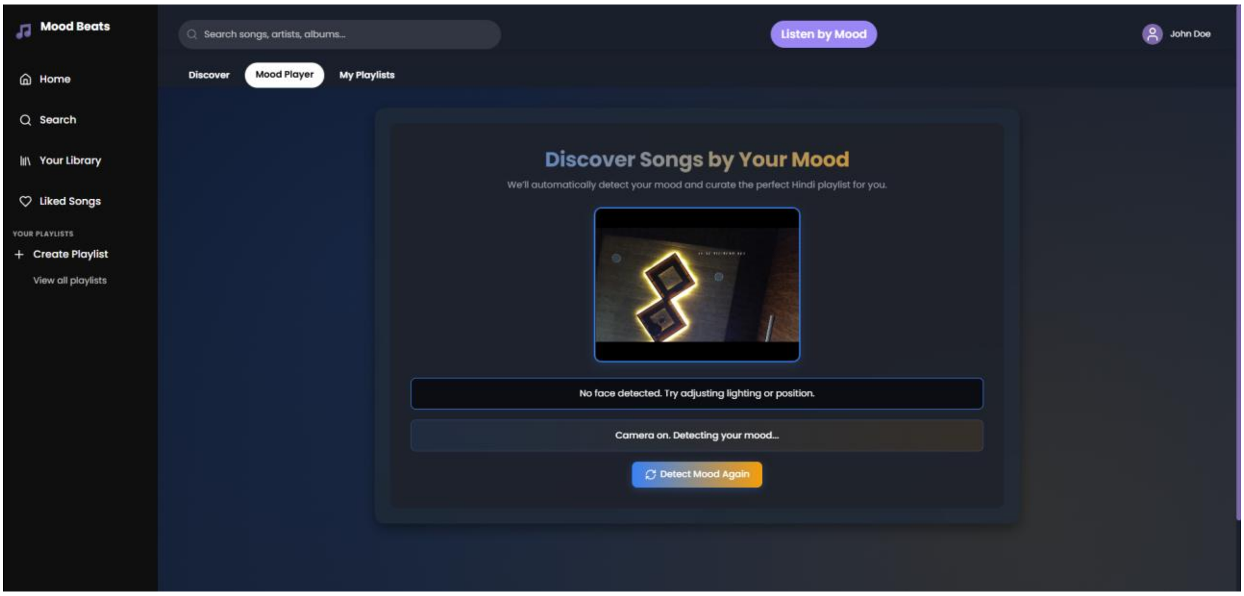
1. **User Login/Signup**: Users are welcomed with a login or registration page. Only registered users can access the mood-detection feature.



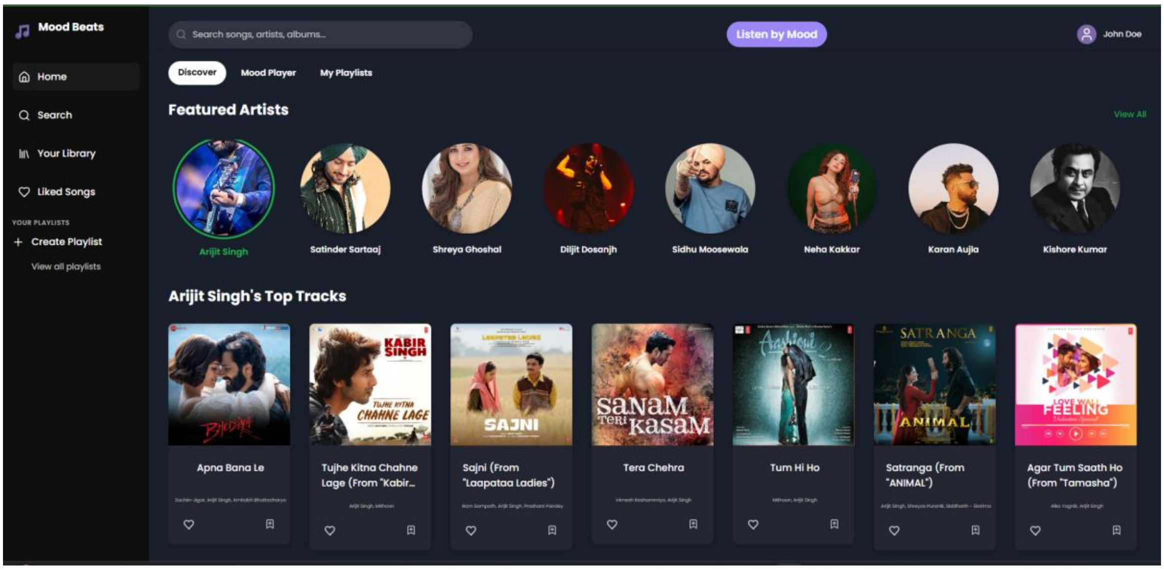
2. **Image Upload**: After logging in, the user uploads a facial image which is sent to the backend for emotion detection.



3. **Mood Display**: The predicted mood (e.g., happy, sad, relaxed, energetic) is shown to the user with supportive visuals or emojis.



4. **Song Recommendations**: A curated list of songs aligned with the user's mood is displayed, fetched using Spotify's APIs.



5. **Playlist Saving**: Users can save or like songs, creating a mood-based personal playlist.

5.6 Summary of the Program’s Structure

Layer	Responsibilities
Frontend	Handles UI rendering, collects user input, displays moods and recommended songs.
Backend	Processes images, detects mood, fetches songs from Spotify, manages authentication.
Database	Stores user profiles, song preferences, and playlists.
Third-party	Spotify API for music, Deep learning models or services for emotion recognition.

5.7 Conclusion

MoodBeats is the outcome of thoughtful structural planning and modern UI development. The backend and frontend are carefully decoupled but work in harmony to deliver a smooth and intelligent experience. By adopting modular design patterns, integrating with external services, and designing a responsive, user-friendly interface, MoodBeats achieves both functionality and user satisfaction. The architecture allows future scalability, including the potential integration of audio-based mood detection or real-time camera inputs.

6. SYSTEM TESTING

6.1 Introduction

System testing is a critical phase of the Software Development Life Cycle (SDLC) that validates the end-to-end functionality of the entire application. It focuses on evaluating the complete and integrated system to ensure that it complies with the specified requirements. This form of black-box testing is performed after unit and integration testing are complete.

In the context of the **MoodBeats** project, although actual system testing was not carried out due to time and scope limitations, this chapter outlines the proposed approach, hypothetical test scenarios, and the types of system testing that would be implemented to ensure the quality and reliability of the application.

6.2 Objectives of System Testing

The main objectives of system testing in the MoodBeats application would be:

- To ensure all components of the system work together as expected.
- To verify the integration between user interface, backend logic, and external APIs (like emotion detection and Spotify API).
- To confirm that the application delivers mood-based music recommendations accurately and efficiently.
- To identify bugs or inconsistencies before the final deployment.
- To validate user experience across different devices and platforms

6.3 Hypothetical Test Scenarios

Since full testing has not been carried out practically, here are some sample test cases that would ideally be used:

Test Case ID	Test Scenario	Expected Outcome
1	User registers with valid input	Account created successfully
2	User logs in with invalid password	Appropriate error message is displayed
3	User uploads a valid image	Mood is detected and displayed on the screen
4	Mood-based songs are fetched	Songs relevant to mood are displayed correctly
5	Application accessed on a mobile device	UI is responsive and functions correctly
6	Backend API returns error	Error is handled gracefully on frontend
7	Attempt unauthorized access to dashboard	Access is denied and user is redirected to login

6.4 Challenges Anticipated During Testing

- **Inconsistent Results from APIs**: Spotify or emotion detection APIs may return different results on multiple requests.
- **Image Quality Dependence**: Poor lighting or background noise in the uploaded image may affect emotion detection accuracy.
- **Cross-Browser Rendering Issues**: CSS compatibility problems could arise on different browsers.
- **Network Dependency**: Heavy reliance on internet speed for Spotify API or image upload functionality.
- **Handling Asynchronous Calls**: Synchronization of API calls and UI updates could introduce logical errors.

6.5 Conclusion

System Testing plays a pivotal role in ensuring that the application is stable, secure, and provides the intended functionalities to the end-users. While this testing phase was not fully implemented in the current version of MoodBeats due to resource constraints, the strategies outlined in this chapter provide a clear roadmap for future testing efforts.

7. LIMITATIONS

1. **Limited Emotion Detection Accuracy**
   The emotion detection system may not always correctly identify a user's mood, especially in low-light environments or when the user's facial expressions are subtle or unclear.
2. **Dependence on Camera Quality**
   The application relies on webcam input, so its performance may be affected on devices with low-resolution cameras or poor lighting, leading to inaccurate mood analysis.
3. **No Long-Term Personalization**
   MoodBeats currently recommends songs based only on the user’s current emotion and does not store past preferences or emotional trends to improve future recommendations.
4. **Internet and API Dependency**
   The system relies heavily on external APIs like Spotify. Any changes, limitations, or downtime in these services can directly impact the app’s core functionality.
5. **Privacy Concerns**
   Users may feel uncomfortable granting access to their camera, which is essential for emotion detection. This could limit the number of users willing to use the application.
6. **Limited Music Library**
   Since the app fetches music from predefined APIs, users looking for regional, niche, or non-mainstream songs may not find relevant recommendations.
7. **Scalability Issues**
   The current backend and database setup may not efficiently support a large number of users or high traffic without future optimization and scaling.

8. CONCLUSION

The *MoodBeats* project presents an innovative approach to enhancing music listening experiences by integrating emotion detection with real-time song recommendations. By using facial expression analysis to identify a user's current mood and recommending songs accordingly, the application attempts to make music selection more intuitive, personal, and emotionally resonant.

Throughout the development of this project, we explored the integration of machine learning, web technologies, and APIs like Spotify to create a functional and user-friendly application. While the system demonstrates promising capabilities, it also highlights certain limitations such as emotion detection accuracy, dependency on external services, and privacy concerns, which will guide the direction for future improvements.

In conclusion, *MoodBeats* successfully combines technology and emotional intelligence to deliver a unique user experience. With further refinement and scalability, it holds great potential to evolve into a fully personalized music companion that adapts to users' moods in real time, enhancing both emotional well-being and entertainment.

9. FUTURE SCOPE

1. Emotion Detection Log Storage and Analytics
At the current stage of development, the MoodBeats system processes emotion detection in real time and uses the detected emotion to recommend music. While the architecture is designed to support the storage of emotion detection logs for future analytics and personalization, this feature is not yet implemented. In future versions, each emotion detection event will be stored in a dedicated collection (e.g., EmotionLogs) in MongoDB, including fields such as user ID, timestamp, detected emotion, and confidence score. This data can be leveraged to provide users with mood history insights, personalized trends, and improved recommendation accuracy.

2. Advanced Personalization and Machine Learning
Future iterations of MoodBeats can incorporate advanced machine learning algorithms to provide deeper personalization. By analyzing stored emotion logs, listening history, and user feedback, the system can learn individual preferences and adapt recommendations over time. Techniques such as collaborative filtering, reinforcement learning, or hybrid recommendation models can be explored to enhance the relevance and diversity of suggested music.

3. Integration with Smart Home and Voice Assistants
Another promising direction is the integration of MoodBeats with smart home devices and voice assistants (such as Amazon Alexa, Google Assistant, or Apple Siri). This would enable hands-free, voice-activated music recommendations based on real-time mood detection, creating a more immersive and accessible user experience.

CHITKARA UNIVERSITY

10. BIBLIOGRAPHY / REFERENCES

1. El Ayadi, M., Kamel, M. S., & Karray, F. (2011). *Survey on speech emotion recognition: Features, classification schemes, and databases*. Pattern Recognition, 44(3), 572–587.
   https://doi.org/10.1016/j.patcog.2010.09.020

2. Soleymani, M., Caro, M. N., & Pun, T. (2013). *A survey of emotion recognition methods: Audio, visual, and spontaneous expressions*. IEEE Transactions on Affective Computing, 4(3), 345–359.
   https://ieeexplore.ieee.org/document/6479257

3. Kotsia, I., & Pitas, I. (2007). *Facial expression recognition in image sequences using geometric deformation features and support vector machines*. IEEE Transactions on Image Processing, 16(1), 172–187.
   https://doi.org/10.1109/TIP.2006.884936

4. Spotify R&D Blog. (2021). *Understanding Spotify's Discover Weekly Algorithm*.
   https://engineering.atspotify.com/2021/06/discover-weekly-how-machine-learning-finds-your-new-music/

5. MongoDB Documentation. *Data Modeling and Schema Design*.
   https://www.mongodb.com/docs/manual/core/data-modeling-introduction/