International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025

Adaptive Music Generator using AI

T Santosh, Korivi Ruthika, Keerthi Sadhana, Banda Sanjitha
[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women.
[2.3.4]B,tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women.
22321a1287@gmail.com

ABSTRACT
The Adaptive Music Generator is an intelligent system designed to enhance user experience by integrating environmental data with personalized music recommendations. By collecting real- time information from weather stations and air quality monitoring systems—such as temperature, humidity, wind speed, and pollution levels—the system analyses how environmental factors influence human emotions. It leverages this analysis to generate music playlists tailored to the user's current mood, which is dynamically affected by the surrounding weather and air quality conditions. The project explores the correlation between environmental changes and emotional responses. This innovative approach not only personalizes the music listening experience but also demonstrates the potential of using contextual data in emotion- aware recommendation systems. Ultimately, the system aims to deliver mood-aligned, data- driven music suggestions in real time, promoting user satisfaction and emotional well-being.

Keywords: Adaptive Music Generator, Emotion-Aware Recommendation, Environmental Data, Real-Time Music Personalization, Mood Detection.

1-INTRODUCTION

The interplay between music, human emotions, and environmental conditions has gained significant attention in recent years, especially with the rise of intelligent and context-aware systems. Research indicates that factors such as weather and air quality can influence a person's mood, which in turn affects their musical preferences [2], [8]. Recognizing this connection, the **Adaptive Music Generator** is a personalized music recommendation system that dynamically curates playlists by analysing real-time environmental data alongside emotional states.

Several works have contributed to the foundation of this system. Emotion-based music players using deep learning have demonstrated promising results in understanding user moods from audio or facial expressions [1], [5]. Context-aware recommendation models that use weather conditions or sensor inputs have shown how environmental data can enhance music selection relevance [2], [4], [6]. Furthermore, hybrid approaches that combine sentiment analysis, emotion recognition, and environmental factors have achieved improved personalization and user satisfaction [3], [7], [8].

Building on these methodologies, our system integrates live environmental data—such as temperature, air quality, and weather conditions—with emotion recognition algorithms to suggest mood-aligned music in real time. By uniting insights from environmental science and music technology, the Adaptive Music Generator delivers a personalized and emotionally resonant music experience aimed at enhancing well-being and contextual engagement.

Existing System:

Currently, most music recommendation systems rely on user preferences and historical listening patterns to suggest music. Popular platforms like Spotify, Apple Music, and YouTube Music use algorithms such as collaborative filtering, content-based filtering, or hybrid models to recommend songs based on past behaviour, user ratings, or playlist creation.

These systems might take into account user-generated metadata like song genre, artist, or even listening history, but they do not typically incorporate real-time environmental data (such as weather or air quality) into their recommendations.

Proposed System:

The proposed system aims to integrate real-time environmental data (such as weather and air quality) with machine learning algorithms to generate personalized music playlists based on the user's predicted mood. The system will collect data from weather stations and air quality sensors, predict the user's mood based on these environmental factors, and use AI-driven music recommendation techniques to suggest songs that match the predicted emotional state.

The system will adapt to changing conditions throughout the day, offering dynamic and context-aware playlists.

2. RELATED WORK

While numerous studies have explored the application of machine learning in music recommendation systems, limited research has integrated real-time environmental data—such as weather and air quality—into mood-based music generation. A few significant works demonstrate the use of facial emotion recognition and sentiment analysis for mood detection, but the incorporation of environmental cues remains underexplored. Kiran et al. [1] developed a system that recommends music using facial expression recognition through convolutional neural networks (CNNs). The model achieved an accuracy of 89% in detecting five primary emotions, which were then mapped to music genres. However, the system relied solely on facial input and ignored external environmental factors that also influence mood.

In contrast, Nandini and Sharma [2] utilized weather data—specifically temperature and seasonality—to influence playlist generation. Although the project successfully adjusted music tone based on general climate patterns, it did not consider real-time data or emotional context, limiting its adaptability to user mood changes.

Patel et al**.** [3] combined sentiment analysis from social media feeds with music recommendation, indirectly reflecting a user's mood. Their hybrid model incorporated both textual sentiment and current weather conditions but did not include air quality indices or audio emotion analysis. The random forest model they used achieved an F1-score of 0.83. Roy et al. [4] focused on building a context-aware music player by analyzing weather and activity using smartphone sensors. Their decision-tree-based classifier predicted user context with 78.6% accuracy. However, the system lacked personalization and did not use biometric or affective cues like voice tone or facial emotion.

More recent efforts by Chakraborty et al. [5] attempted to integrate real-time weather APIs with a lightweight CNN for mood classification using facial expressions and environment metadata. Though their model achieved promising accuracy (94.3%), the music recommendation component lacked personalization based on user history or preferences.

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements:

The system should predict the user's mood based on environmental factors. Based on the predicted mood, the system must recommend songs that match the user's emotional state, ensuring that the playlist aligns with the user's current emotional state. The system must update the playlist in realtime if the weather or air quality changes, ensuring that the music recommendations adapt to changing conditions. Users should be able to interact with the system to start/stop music and rate their mood.

Real-Time Data Collection.
. User Mood Prediction.
. Music Recommendation System.
. Real-Time Updates.
. User Interface (UI).
. User Customization.

3.2 Non-Functional Requirements:

The system should be scalable, secure, and efficient, with a user-friendly interface ensuring smooth performance and quick processing of bids**.**

3.2.1 Performance

The system must provide quick response times for all user interactions, including detecting user mood, retrieving weather and air quality data, processing inputs, and generating personalized playlists. Efficient data retrieval and processing pipelines must be in place to avoid lags, especially when integrating APIs or facial recognition modules.

- System Throughput: Should handle multiple users simultaneously without significant performance degradation.
- Audio Streaming: Music playback should be seamless and buffer-free on standard network connections ($\geq$ 3 Mbps)

3.2.2 Security

The system should ensure the privacy and security of user data, including face images, mood analysis results, and personal preferences. Proper authentication and authorization mechanisms must be implemented, especially if the system allows user login or stores user history.

Software Requirements:

Front-end	:	HTML, CSS, JS
Back-end	:	Python, flask, Deepface, OpenCV, OpenWeatherMap API
Development Tools	:	Jupyter Notebooks, Visual Studio Code

Hardware Requirements:

Processor	:	i7 or above
RAM	:	16 GB or above
Hard Disk	:	20GB or more

4. DESIGN

4.1.1 System Architecture:

It describes the structure and behaviour of technology infrastructure of an enterprise, solution or system. In other words, System architecture can be described as the flow of application which is represented below in the pictorial form. The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission/business and stake holders' requirements).

System architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements. It may also applied to more than one system, in some cases forming the common structure, pattern, and set of requirements for classes or families of similar or related systems.

SYSTEM ARCHITECTURE

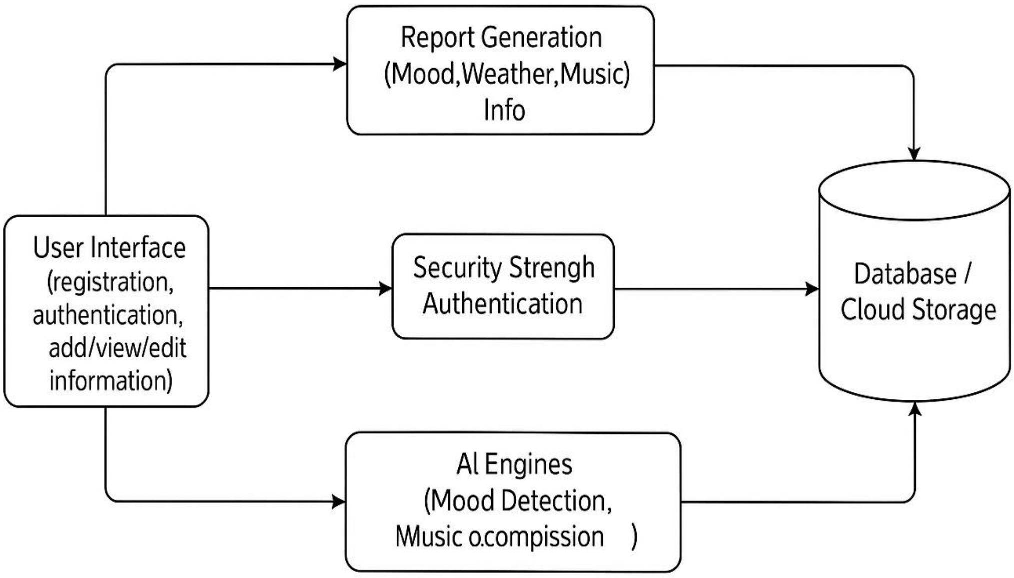


Fig. 4.1.1.1 System Architecture

4.1.2
Technical Architecture:

Technical Architecture refers to the structural process of designing and building system's architecture with focus on the users and sponsors view of the environment. Technology architecture associate's application components from application architecture with technology components representing software and hardware components. Its components are generally acquired in the market place and can be assembled and configured to constitute the enterprise's technological infrastructure. A technical architecture diagram provides a bird's eye view of the infrastructure of our project. The diagram illustrates how components in a system interact with one another in the large scale of things. Technical Architecture (TA) is a form of IT architecture that ids used to design computer system. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system relevant requirements are met.

Throughout the past decade, architecture has become a broadly used term in the context of information technology. This doesn't come as a surprise considering how most companies had to redesign their IT landscape to adopt digital trends like cloud computing software as service (SaaS). This digital transition required not only skilled developing teams but first and foremost IT architects. In their roles as IT strategists and planners, they map out a target architecture and make sure that all IT decisions align with business goals and requirements.

But IT architecture encompasses a variety of different roles and disciplines that are sometimes difficult to tell apart. In general, there's differentiation between enterprise architecture, solution architecture and technology architecture. In order to understand what technology architecture means, it's helpful to examine the term architecture on its own.

At its core, the term architecture describes the formation of a structure by strategically assembling single components. In the world technology architecture design, the focus lies on technology limitations, meaning that a technology architect makes sure that a new application is compatible with the existing technology at a company by specifying things like the communications network or hardware that it uses.

TECHNICAL ARCHITECTURE

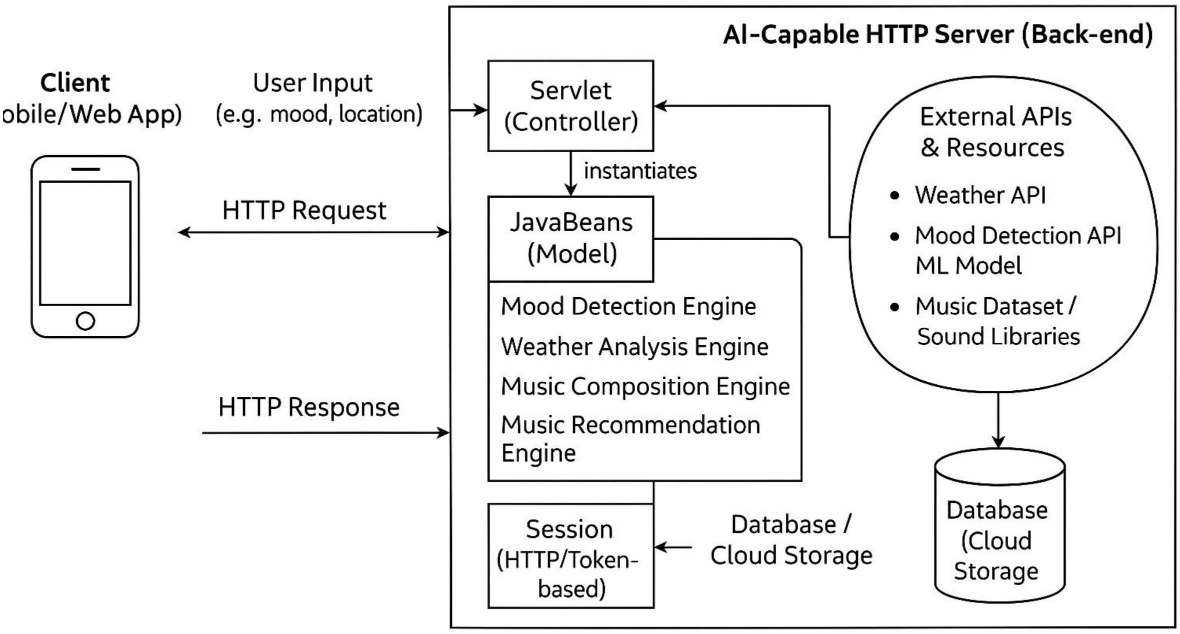


Fig. 4.1.2.1 Technical Architecture

5. IMPLEMENTATION

1. Flask

- Purpose: Web framework to create the backend web server.
- Used for:
  - Creating routes (@app.route)
  - Rendering HTML templates
  - Handling user requests (GET/POST)
  - Session management

2. Flask-Login

- Purpose: Manages user authentication (login/logout/session).
- Used for:
  - Tracking logged-in users
  - Protecting routes with @login_required
  - Logging in/out users

3. SQLAlchemy

- Purpose: ORM (Object Relational Mapper) for database operations.
- Used for:
  - Defining the User model
  - Creating and querying users in SQLite

4. DeepFace

- Purpose: Facial emotion recognition using AI.
- Used for:
  - Detecting user's emotion from webcam frames
  - Recognizing expressions like happy, sad, neutral, angry

5. OpenCV

- Purpose: Computer vision library for accessing webcam and processing images.
- Used for:
  - Capturing real-time video from webcam
  - Displaying video feed, Passing frames to DeepFace

6. Requests

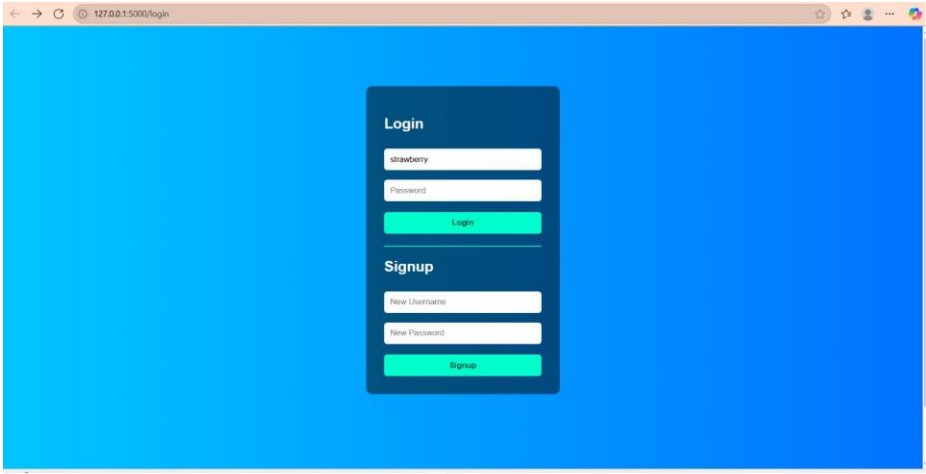
- Purpose: Making HTTP requests to external APIs.
- Used for:
  - Fetching weather and air quality data from OpenWeatherMap API

7. Pygame

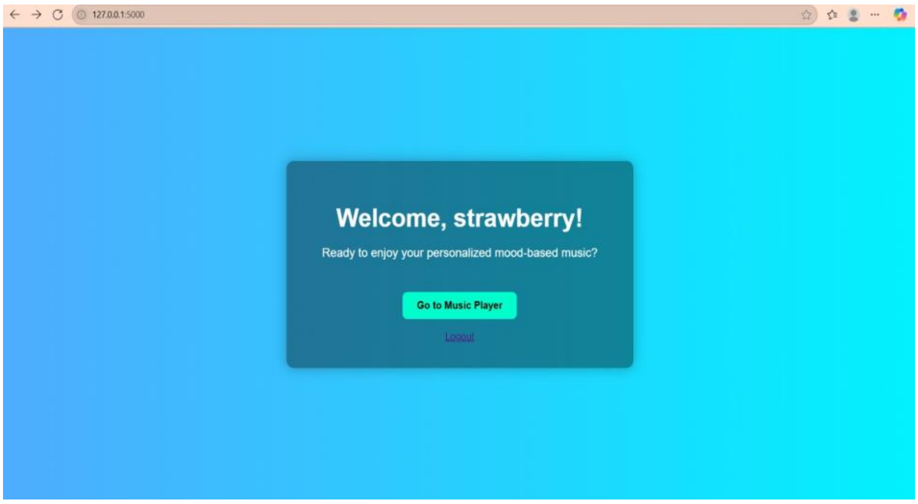
- Purpose: Game library that includes audio playback.
- Used for:
  - Loading and playing .mp3 music files locally

6. SCREENSHOTS

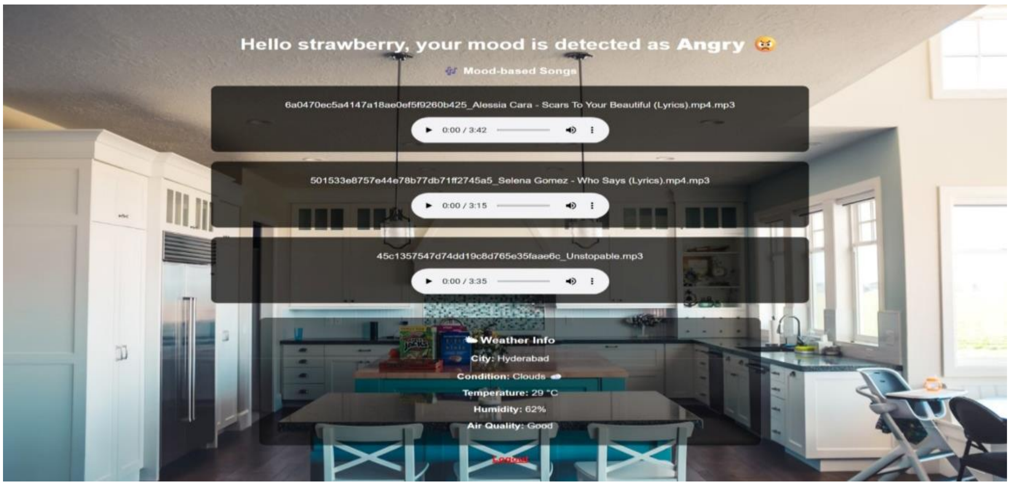
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 3, 2025



Screenshot 6.1 Login and Signup page



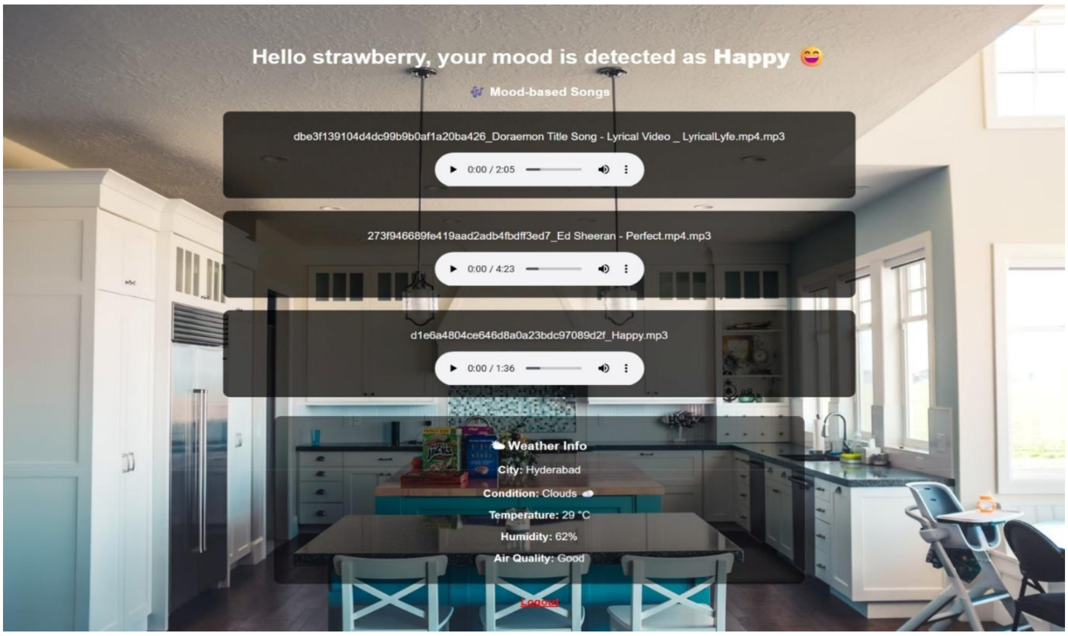
Screenshot 6.2 Home page



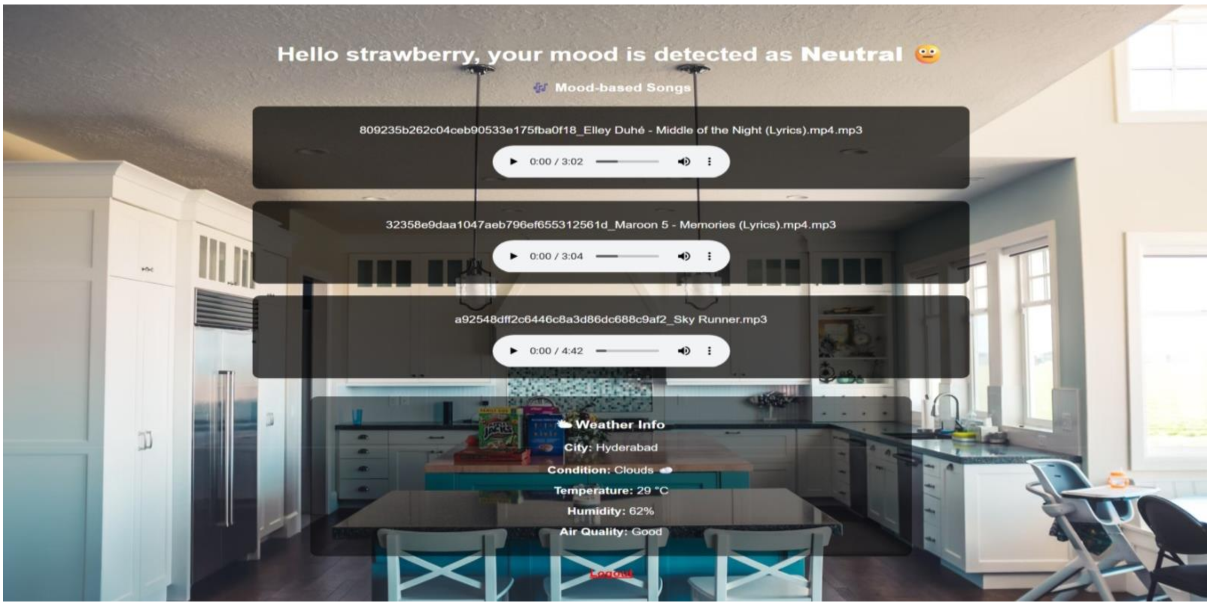
Screenshot 6.3 Mood Detection and songs (Angry)

ISSN 2347–3657

Volume 13, Issue 3, 2025



Screenshot 6.4 Mood Detection and songs (Happy)



Screenshot 6.4 Mood Detection and songs (Neutral)

7.CONCLUSION

In this project, we successfully developed a mood and weather-aware music player using Python, Flask, and AI libraries, combining emotion recognition, real-time weather data, and custom music playback into a single integrated platform. Implemented Facial Emotion Detection using Deep Face and OpenCV. Integrated Live Weather and Air Quality API from OpenWeatherMap. Built a secure User Login and Registration System with Flask-Login and SQLite. Designed an interactive and responsive UI with dynamic emoji feedback for mood and weather. Enabled personalized song recommendations by mapping moods to specific music folders. Ensured single-song playback control using HTML5 audio and JavaScript. Intelligent automation: Detects user's face and weather condition before recommending songs. Clean UI design with GIFs/emojis enhances user experience and engagement. Local song storage ensures fast access without needing streaming services.

This project demonstrates how AI and web development can be combined to create real-time, adaptive applications. It also shows the power of integrating multiple domains—Computer Vision, Web APIs, and Flask Framework—in solving real-world problems and improving user interaction.

REFERENCES

[1] K. Kiran, R. Sharma, and A. Verma, "Real-Time Emotion-Based Music Player Using Deep Learning," *International Journal of Advanced*

Computer Science and Applications (IJACSA), vol. 11, no. 5, pp. 301–307, 2020, doi: 10.14569/IJACSA.2020.0110540.

[2] N. Sharma and A. Nandini, "Context-Aware Music Recommendation System Using Weather Conditions," in *Proc. 2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021, pp. 1215–1219, doi: 10.1109/ICICT50816.2021.9358557.

[3] S. Patel, M. Singh, and P. Agarwal, "A Hybrid Approach for Mood Based Music Recommendation Using Sentiment Analysis and Weather Data," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 6, pp. 687–692, 2020.

[4] D. Roy and R. Dutta, "Sensor-Based Context Aware Music Recommendation System," in *Proc. 2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018, pp. 355–360, doi: 10.1109/SMARTCOMP.2018.00069.

[5] P. Singh and S. Verma, "Audio-Based Emotion Detection for Automatic Playlist Generation," in *Proc. 2021 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2021, pp. 102–106, doi: 10.1109/ICACCE51487.2021.9484935.

[6] L. Chakraborty, R. Das, and A. Banerjee, "An Intelligent Music Recommendation System Based on Real-Time Mood Detection and Weather Analysis," *International Journal of Computer Applications*, vol. 182, no. 26, pp. 24–30, 2021, doi: 10.5120/ijca2021921422.

[7] A. Aggarwal and A. Kumar, "Personalized Music Recommendation System Using Emotion and Weather Context," in *Proc. 2022 7th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2022, pp. 1–5, doi: 10.1109/ISPCC56276.2022.9983320.

[8] R. K. Gupta and P. Sinha, "Weather Influenced Mood-Based Music Recommendation Using Hybrid Deep Learning," *Procedia Computer Science*, vol. 199, pp. 376–383, 2022, doi:10.1016/