RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Mini Project Report On

SerenadeBot

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Adithyan Darshan Kidav (U2103016)

Aedna Mary Reji (U2103017)

Alan Anu Sam (U2103020)

Allwyn Antony Rodrigues (U2103028)

Under the guidance of

Ms. Meenu Mathew

Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
May 2024

CERTIFICATE

This is to certify that the mini project report entitled ***"SerenadeBot"*** *is a bonafide record of the work done by* ***Adithyan Darshan Kidav (U2103016)***, ***Aedna Mary Reji (U2103017)***, ***Alan Anu Sam (U2103020)***, ***Allwyn Anthony Rodrigues (U2103028)***, *submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Project Guide:
Ms. Meenu Mathew
Asst. Professor, DCS
Dept. of CSE
RSET

Project Coordinator:
Mr. Harikrishnan M
Asst. Professor, DCS
Dept. of CSE
RSET

Head of the Department:
Dr. Preetha K G
Professor & Head of the Department
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Head of the Department of Computer Science and Engineering for providing me with the opportunity to undertake my mini project, "SerenadeBot".

I am highly indebted to my project coordinator, Mr.Harikrishnan M, Assistant Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to our project guide Ms. Meenu Mathew for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Adithyan Darshan Kidav

Aedna Mary Reji

Alan Anu Sam

Allwyn Antony Rodrigues

Abstract

In our project, we aim to enhance music discovery and stress relief by leveraging advanced technology. So, we wish to create a chatbot-driven music recommendation system by integrating the last.fm API with a pretrained chatbot, we create a system that recommends music based on the user's mood expressed in their conversation with the chatbot. We achieve this by analyzing the tone and detecting emotions using NLTK (Natural Language Toolkit) and other APIs if required for NLP (Natural Language Processing) with Python. We will also try to incorporate the use of html,Css and javascript to build Web user interfaces in Python. This innovative approach allows users to discover soothing music tailored to their emotions, ultimately providing a personalized and immersive listening experience. By harnessing the power of technology and music, we strive to improve well-being and enhance the way people interact with and enjoy music in their daily lives.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In today's fast-paced world, individuals frequently experience a range of emotions due to various stressors such as work, relationships, and societal pressures. Despite the plethora of music available online, many struggle to find songs that match their current emotional state. This difficulty navigating music libraries amidst emotional turmoil highlights a need for personalized support in accessing mood-appropriate music.

Importance of the Project:

The development of a mood-based music recommendation chatbot addresses this need by leveraging natural language processing (NLP) and machine learning. By analyzing user speech or text, the chatbot accurately assesses the user's mood and suggests music accordingly. This personalized approach enhances mental well-being by providing tailored support in moments of loneliness, sadness, stress, or even joy.

Furthermore, the chatbot offers accessibility and convenience, available anytime and anywhere. Interacting with the chatbot encourages emotional awareness and promotes healthier coping mechanisms through music therapy. Continuous learning from user feedback ensures the chatbot's recommendations remain relevant and effective.

In summary, the project amalgamates technology and mental health support, offering a dynamic solution for emotional well-being in an increasingly digital world. By harnessing the therapeutic power of music and artificial intelligence, the chatbot aims to positively impact individuals' lives, providing comfort, inspiration, and guidance when needed most.

1.2 Problem Definition

The problem to be addressed is the difficulty individuals face in finding music that aligns with their current emotional state amidst a vast array of options, highlighting the need for a personalized and accessible solution for mood-based music recommendations.

1.3 Scope and Motivation

Scope: The scope of this project encompasses the development of a mood-based music recommendation chatbot, utilizing natural language processing (NLP) and machine learning algorithms. The chatbot will analyze user input, including speech and text, to accurately gauge their mood and suggest appropriate music selections. It will offer personalized recommendations tailored to specific emotional states, such as loneliness, sadness, stress, or joy, ensuring relevance and effectiveness. Additionally, the chatbot will provide a user-friendly interface for seamless interaction, accessible across various digital platforms and devices. The project's scope includes continuous refinement and optimization based on user feedback to enhance the chatbot's accuracy and usability over time.

Motivation: The motivation behind this project stems from the growing recognition of the therapeutic benefits of music in managing emotions and promoting mental well-being. By developing a chatbot capable of understanding and responding to users' emotional needs through music recommendations, we aim to provide accessible and personalized support to individuals navigating their daily challenges. Moreover, the project seeks to harness the potential of artificial intelligence to address a pressing need in the digital age: providing meaningful connections and resources for emotional support in an increasingly fast-paced and technology-driven world. Through this initiative, we aspire to empower individuals to better understand and regulate their emotions while fostering a sense of connection and resilience in their journey towards improved mental health.

1.4 Objectives

- Develop a natural language processing (NLP) model capable of accurately analyzing user speech or text to identify and categorize emotional states.

- Implement machine learning algorithms to generate personalized music recommendations based on the user's emotional state, considering factors such as mood, tempo, and genre preferences.
- Design and develop an intuitive user interface for seamless interaction with the chatbot across various digital platforms and devices.
- Incorporate continuous feedback mechanisms to iteratively refine and optimize the chatbot's recommendation algorithms and user experience.
- Ensure the chatbot's scalability and robustness to handle a growing user base while maintaining high levels of accuracy and responsiveness.
- Conduct user testing and evaluation to assess the effectiveness, usability, and user satisfaction of the chatbot in providing mood-based music recommendations and supporting emotional well-being.

1.5 Challenges

The project faces challenges in accurately interpreting and categorizing the complexities of human emotions from speech or text inputs, which can vary greatly depending on context and individual nuances. Implementing existing natural language processing models and integrating them effectively with the chatbot's functionalities while ensuring their adaptability to diverse linguistic styles and expressions poses a significant technical hurdle.

1.6 Assumptions

Assumptions associated with this project are:

- Users' expressed emotions accurately reflect their internal emotional states.
- Users will engage with the chatbot in a genuine manner and provide accurate feedback on the music recommendations provided.
- Users will perceive music as a meaningful tool for managing and influencing their emotions.

1.7 Societal / Industrial Relevance

The project of developing a mood-based music recommendation chatbot has broad applications across both societal and industrial contexts:

- Mental Health Support Services: In the realm of societal applications, the chatbot can be integrated into mental health support services, offering individuals a convenient and accessible tool for managing their emotional well-being. By providing personalized music recommendations tailored to users' moods, the chatbot enhances the effectiveness of therapy and counseling sessions, promoting self-expression and emotional regulation.

- Wellness and Self-Care Apps: Within the industry, the chatbot can be incorporated into wellness and self-care apps, catering to a growing demand for holistic approaches to health and well-being. By integrating mood-based music recommendations, these apps can offer users a comprehensive suite of tools for managing stress, improving mood, and enhancing overall mental resilience.

- Employee Assistance Programs (EAPs): In corporate settings, the chatbot can be utilized as part of employee assistance programs (EAPs) to support the mental health and well-being of staff members. By providing personalized music recommendations to alleviate stress, boost morale, and foster a positive work environment, the chatbot contributes to employee satisfaction, productivity, and retention.

- Education and Research: Additionally, the project holds relevance for educational institutions and research organizations interested in exploring the intersection of technology and mental health. By studying the effectiveness of mood-based music recommendations in managing emotions and promoting well-being, researchers can gain valuable insights into the therapeutic potential of music and inform future interventions and treatments.

Overall, the project's applicability lies in its ability to leverage technology to address fundamental human needs for emotional support and self-care. Whether deployed in societal contexts to enhance mental health services or integrated into industry solutions

for employee well-being, the chatbot offers a versatile and impactful tool for promoting emotional resilience and enhancing overall quality of life.

1.8 Organization of the Report

1. Introduction

1.1 Purpose

- Provide users with personalized music recommendations tailored to their current emotional state.
- Enhance users' emotional well-being by offering music selections that can uplift, comfort, or resonate with their mood.
- Foster a sense of connection and support by leveraging technology to offer accessible tools for managing emotions through music.
- Continuously refine recommendations based on user feedback to ensure relevance and effectiveness in promoting mental health.

1.2 Product Scope

- Functionality 1: Mood Analysis: The system interprets user input, whether text or speech, to discern their emotional state accurately.
- Functionality 2: Music Recommendation: Based on the detected mood, the system suggests a curated list of songs from its database that match the user's emotional context.
- Boundary 1: Emotion Detection Limitations: The system may encounter challenges in accurately deciphering nuanced emotions, leading to potential inaccuracies in mood analysis.
- Boundary 2: Music Selection Constraints: The availability of recommended songs is contingent upon the system's licensed music library, potentially limiting the range of available options.

1.3 System Overview

- User Interface: The system's front-end component facilitates user interaction, allowing users to input text or speech and receive music recommendations.
- Emotion Analysis Module: This component processes user input to detect emotional cues using natural language processing (NLP) techniques, determining the user's mood.
- Recommendation Engine: Based on the identified mood, the recommendation engine generates a list of music tracks from the system's database that align with the user's emotional state.
- Feedback Mechanism: Users can provide feedback on the recommended songs, which is used to refine the recommendation algorithm over time, enhancing the system's accuracy and relevance.

2. System Description

2.1 Product Perspective

- Integration with Music Platforms: The system can be integrated as a component within existing music platforms, augmenting their functionality by offering mood-based music recommendations to users.
- Complementing Content Discovery: The system complements existing content discovery features within music platforms, offering an additional layer of recommendation based on emotional context, thereby diversifying content exploration options for users.
- Contributing to Platform Differentiation: Incorporating the system's mood-based recommendation capability can differentiate music platforms in a competitive market landscape, attracting and retaining users seeking personalized and emotionally resonant music experiences.

2.2 Product Functions

- Emotion detection from user input (text or speech).
- Music recommendation based on the detected emotion.

- User feedback collection and integration to refine future recommendations.

2.3 Operating Environment

- Hardware Requirements:

    1. Personal computer, laptop, or mobile device capable of running software applications.
    2. Microphone (for speech input) or keyboard (for text input), depending on user interaction preferences.

- Software Requirements:

    1. Chatbot application or software framework capable of processing natural language input and generating responses.
    2. Natural language processing (NLP) libraries or APIs for sentiment analysis and emotion detection.
    3. User interface components for interaction and displaying recommendations.

- Network Requirements:

    1. Stable internet connection for accessing online music databases and services (if applicable) and sending/receiving data.
    2. Secure communication protocols to protect user data and maintain privacy during interactions with the chatbot.

2.4 Assumptions and Dependencies

- Third-Party Music Databases: Access to external music databases or streaming platforms for retrieving song recommendations based on user preferences and emotional context. (here; Last.fm)

- Natural Language Processing APIs: Integration with third-party NLP APIs or libraries for sentiment analysis and emotion detection from user input.

3. System Architecture & Design (Deeper Dive)

3.1 System Components

- User Interface: Facilitates interaction between the user and the system, allowing input of text or speech and display of music recommendations.
- Emotion Detection Module: Analyzes user input to detect emotional cues and determine the user's current emotional state.
- Recommendation Engine: Generates personalized music recommendations based on the user's detected emotion and preferences.

3.2 Architectural Design

Sequence diagram or other visual representation illustrating the interaction flow between components included.

3.3 Dataset

Data source used by the system (Last.fm API). Some of its properties are:

- Music Metadata: Access to extensive metadata about music tracks, albums, artists, and genres.
- User Data: Retrieval of user-specific information, such as listening history, favorite tracks, and recommended artists.
- Scrobbling: Support for scrobbling, which involves submitting track playback information to a user's Last.fm profile.
- Tagging and Similarity: Ability to retrieve tags associated with tracks or artists, as well as recommendations based on similarity.
- Charts and Trends: Access to charts and trend data, showcasing popular tracks, artists, and genres based on user activity.

4. Functional and Non-Functional Requirements (SRS)

4.1 Functional Requirements

Description and Priority

- This feature enables users to interact with the system by providing input, either through text or speech. It is a fundamental aspect of the system's functionality and is of high priority.

Stimulus/Response Sequences

- Stimulus: User accesses the system interface and provides input via keyboard (text) or microphone (speech).
- Response: The system processes the input and prepares it for analysis or further action.

Functional Requirements

- The system must provide an intuitive interface for users to input text or speech.
- It should support multiple input modalities, including keyboard and microphone.
- It should include error handling mechanisms to manage invalid or ambiguous user input gracefully.

4.2 Non-Functional Requirements

4.2.1 Performance Requirements

- Response Time: Aim for response times of under 500 milliseconds for user interactions, ensuring a smooth and seamless user experience.
- Scalability: Design the system to scale horizontally, allowing it to handle a growing user base and increased workload without compromising performance.
- Accuracy: Strive for emotion detection and music recommendation algorithms with accuracy rates exceeding 90%, providing users with relevant and personalized recommendations consistently.

- Reliability: Ensure system uptime of at least 99.9%, minimizing downtime and service interruptions through robust error handling and fault tolerance mechanisms.

4.2.2 Software Quality Attributes

- Reliability: The software should operate consistently and predictably under varying conditions, minimizing downtime and errors to ensure uninterrupted service availability.

- Usability: The software should be intuitive and user-friendly, with clear navigation and easily understandable interfaces, catering to users of all technical levels.

- Maintainability: The software should be well-organized and modular, with clear documentation and coding standards, facilitating ease of maintenance, updates, and future enhancements.

- Performance: The software should perform efficiently, with fast response times and scalable architecture to accommodate growing user demand while maintaining high levels of responsiveness.

5.Implementation Details

5.1 User Interface Design

Design of the user interface, including wireframes or mockups detailed.

5.2 Implementation Strategies

- Python: Utilized as the primary programming language for backend development due to its versatility, extensive libraries, and robust ecosystem for natural language processing (NLP) and machine learning (ML).

- NLTK (Natural Language Toolkit): Employed for sentiment analysis and emotion detection, providing a comprehensive suite of tools and resources for processing and analyzing text data.

- HTML, CSS: HTML, CSS, and JavaScript are the essential tools for crafting the appearance and functionality of web applications. By seamlessly connecting with

Python on the backend, they enable developers to rapidly build and deploy dynamic and interactive websites with ease and efficiency.

5.3 Module Division

Development tasks to specific modules or team members assigned.

5.4 Work Schedule (Gantt Chart)

Visual timeline outlining project milestones and tasks represented using Gantt Chart.

Chapter 2

Software Requirements Specification

2.1 Introduction

2.1.1 Purpose

This Software Requirements Specification (SRS) outlines the specifications for a Chatbot Song Recommendation System. The system aims to provide personalized song recommendations based on user preferences and interactions. This SRS covers the functional and non-functional requirements of the chatbot system, including its interaction capabilities, recommendation algorithms, user interface, and performance metrics. It details the features and behaviors expected from the chatbot system to fulfill its purpose effectively.

2.1.2 Product Scope

The Chatbot Song Recommendation System is a software application designed to provide personalized song recommendations to users based on their preferences and interactions. The primary purpose of this system is to enhance user experience by offering tailored music suggestions, thereby saving users time and effort in discovering new music that aligns with their tastes.

Key objectives and benefits of the Chatbot Song Recommendation System include:

- **Improved user engagement:** By offering personalized recommendations, the system aims to increase user satisfaction and engagement with the music platform.
- **Time-saving:** Users can quickly discover new songs without manually searching through vast catalogs, enhancing convenience.

- **Enhanced user retention:** Providing relevant recommendations can lead to increased user loyalty and retention, contributing to the long-term success of the music platform.
- **Data-driven insights:** The system gathers user preferences and interactions, enabling the platform to gain valuable insights into user behavior and preferences for future enhancements.

This project aligns with corporate goals or business strategies by focusing on enhancing user experience, leveraging data-driven insights to improve service offerings, and fostering user retention and loyalty within the music platform.

2.2 Overall Description

2.2.1 Product Perspective

The Chatbot Song Recommendation System is a standalone product developed as part of a larger music platform ecosystem. While it operates independently to provide personalized song recommendations to users, it interfaces with other components of the music platform, such as the last.fm API. The product specified in this SRS is a new, self-contained software system designed to address specific requirements. It is not a part of an existing product family or a replacement for any current systems. The system is independent and aims to fulfill unique functionalities outlined in the SRS document. A visual representation like a system architecture diagram can illustrate major components, subsystem connections, and external interfaces, enhancing understanding of the overall system structure.

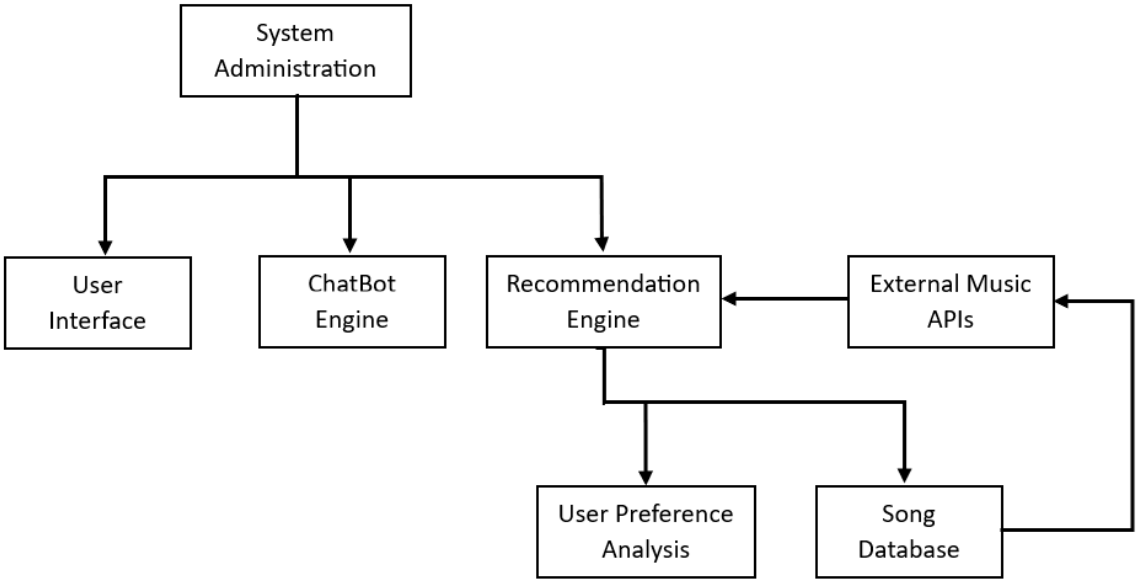


Figure 2.1: Product Perspective

2.2.2 Product Functions

• **Gather user preferences**: Collect user input and data related to music preferences, genres, artists, and mood.

• **Analyze user data:** Process user input and interaction history to generate personalized song recommendations.

• **Provide recommendations:** Present tailored song suggestions to users based on their preferences and interactions.

• **Web interface with Html,Css integration:**Incorporate a web-based user interface using Html,Css for seamless interaction with the Chatbot Song Recommendation System.Top of Form

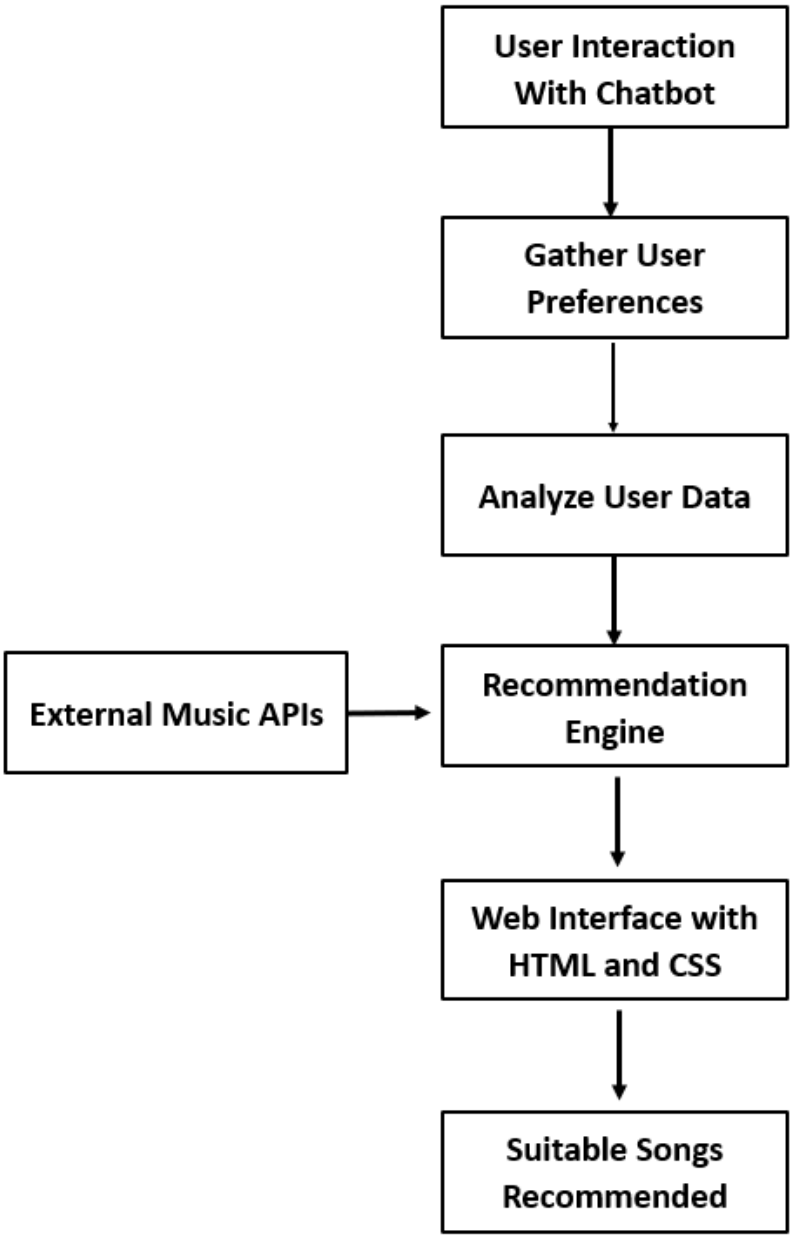


Figure 2.2: Architecture Diagram

2.2.3 Operating Environment

• **Hardware Platform:** The software will operate on standard hardware commonly used for web applications, mainly including desktops, laptops.

• **Operating System:** The system will be compatible with major operating systems such as Windows, macOS.

• **Web Browser:** The software will be accessed through web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

• **Html Integration:** The system will utilize HTML, CSS, and JavaScript to construct and host the web interface, ensuring seamless integration and functionality.
• **Internet Connection:** A stable internet connection will be required for accessing the web-based interface and retrieving song recommendations.

2.2.4 Design and Implementation Constraints

Design and Implementation Constraints:
• **Regulatory Policies:** Compliance with copyright laws and licensing agreements governing the use of music content for recommendation purposes.
• **Hardware Limitations:** Ensure the software is optimized to operate within the memory and processing constraints of the target hardware platforms, including mobile devices.
• **Html Integration:** Employing HTML, CSS, and JavaScript, the web interface is constructed, limiting the range of technologies and tools available for frontend development.
• **Language Requirements:** Develop the chatbot system using programming languages and frameworks suitable for web development, such as Python for backend development and frontend development.
• **Security Considerations:** Implement robust security measures to protect user data, including secure authentication mechanisms, and protection against common web security vulnerabilities.
• **Design Conventions:** Adhere to established design conventions and programming standards to ensure maintainability and scalability of the delivered software.

2.2.5 Assumptions and Dependencies

• **Third-party APIs:** The project assumes access to third-party APIs for retrieving music metadata, such as song titles, artists, genres, and recommendations.
• **Availability of Html Services:** The project depends on the availability and capabilities of HTML, CSS, and JavaScript for constructing and hosting the web interface.
• **User Engagement:** It is assumed that users will actively engage with the chatbot system and provide sufficient feedback for refining the recommendation algorithm.
• **Stable Internet Connection:** The project depends on users having a stable internet connection for accessing the web interface and receiving real-time recommendations.

• **Compliance with Copyright Laws**: Assumption that all music content used for recommendation purposes complies with copyright laws and licensing agreements.

• **Development Resources:** Availability of development resources including manpower, expertise, and time to implement and maintain the software according to the specified requirements.

• **Compatibility with Web Browsers:** The project assumes compatibility with major web browsers for delivering the web-based interface and ensuring a consistent user experience across different platforms.

2.3 External Interface Requirements

2.3.1 User Interfaces:

The software product will feature a user-friendly interface with logical characteristics designed to enhance user experience. Key elements include:

• **Sample Screen Images:** A set of sample screen images will be provided to represent data like the API we are using, pictorially representation of user's mood etc.

• **GUI Standards and Style Guides**: The user interface will adhere to established GUI standards and follow the organization's product family style guide for consistency and branding.

• **Screen Layout Constraints:** Screens will be logically organized to ensure a coherent and intuitive user flow.

• **Standard Buttons and Functions:** Screen will include standard buttons for common functions like to start the search for song recommendations, picking the song from the given list etc.

• **Error Message Display Standards:** Error messages will follow a standardized format, providing clear and informative feedback to users.

2.3.2 Hardware Interfaces:

The software will establish logical and physical connections with hardware components. Characteristics include:

• **Supported Device Types:** The software will support standard device types, including desktop computers, laptops, and tablets.

• **Data and Control Interactions**: Interactions between the software and hardware will be managed through standard communication protocols, ensuring seamless data and control exchanges.

2.3.3 Software Interfaces:

The software product will interface with various software components, including:

• **Last.fm API**: This interface facilitates communication with the Last.fm database to retrieve music recommendations based on user preferences and mood expressed in the conversation with the chatbot.

• **NLTK (Natural Language Toolkit):** Utilized for tone analysis and emotion detection in user conversations. It acts as an interface to process natural language input and extract relevant emotional cues.

• **Html:** HTML, CSS, and JavaScript are employed to create web user interfaces for the system, bridging the gap between the recommendation system's backend logic and the frontend interface. This enables smooth integration and interaction between the chatbot and the user interface.

• **Python Libraries and Tools**: Various Python libraries and tools are utilized for implementing NLP algorithms, integrating APIs, and handling data processing tasks. These libraries serve as interfaces for specific functionalities required in the system.

• **Operating System:** The system is designed to be compatible with different operating systems, ensuring flexibility and accessibility for users across various platforms.

• **Data Sharing:** Data shared across software components include user input (conversation with the chatbot), emotion analysis results, music recommendations, and user preferences. These data items are communicated between different modules of the system to provide personalized music recommendations for the user's mood.

2.3.4 Communications Interfaces:

Communication functions will be required for various purposes:

• **Chatbot-User Interaction**: This interface facilitates communication between the chatbot and the user. It includes text-based exchanges where the user expresses their mood or preferences, and the chatbot responds with music recommendations. Message formatting may involve structured data formats for conveying system responses.

• **API Communication**: The system interacts with external APIs such as Last.fm for retrieving music recommendations. This communication involves sending requests in a predefined format like HTTP requests and receiving responses containing music data. Data transfer rates and security protocols like HTTPS may be relevant considerations.

• **Web Interface Interaction:** HTML, CSS, and JavaScript are employed to construct web user interfaces, allowing users to engage with the recommendation system via a web browser. Communication protocols like HTTP/HTTPS are utilized to transfer data between the interface on the user's device and the system's server-side logic.

• **Data Transfer and Security**: Data transfer rates may vary depending on the size and complexity of exchanged data. Security measures such as encryption (e.g., HTTPS) may be implemented to ensure the confidentiality and integrity of communication channels, particularly when handling sensitive user information.

• **Synchronization Mechanisms:** Synchronization mechanisms may be employed to ensure consistency and coherence in data exchange between different components of the system, especially in scenarios involving concurrent user interactions or real-time updates to music recommendations.

Overall, communication interfaces facilitate seamless interaction between the various components of the system, as well as between the system and external entities such as users and APIs, while addressing considerations such as data security and synchronization.

2.4 System Features

2.4.1 Casual Conversation of User with Chatbot

Description and Priority

The Casual Conversation feature allows users to engage in informal dialogues with the chatbot. This feature is of High priority as it forms the core foundation of the system for understanding user emotions and preferences.

Stimulus/Response Sequences

• **Stimulus:** User initiates a casual conversation.

• **Response:** The chatbot responds with contextually appropriate and friendly replies, fostering a conversational atmosphere.

• **Stimulus:** User shares personal experiences or feelings.

• **Response:** The chatbot tailors its responses to empathize or engage further, maintaining a natural conversational flow.

Functional Requirements

• **REQ-1:** The chatbot must recognize and respond to common casual conversation cues.

• **REQ-2:** The chatbot should employ natural language processing to understand colloquial language.

• **REQ-3:** Responses should be varied and contextually relevant to create an engaging and realistic conversation.

2.4.2 Analysis of Conversation for Emotion Detection

Description and Priority

The Emotion Detection feature analyzes user conversations to discern emotional cues. This is of High priority as it enhances the system's ability to provide personalized recommendations based on user mood.

Stimulus/Response Sequences

• **Stimulus:** User expresses joy, sadness, excitement, etc.

• **Response:** The system employs emotion detection algorithms to identify and categorize the user's emotional state.

Functional Requirements

• **REQ-4:** The system must incorporate emotion detection algorithms, such as NLTK, to identify user emotions in real-time.
• **REQ-5:** Emotion analysis should analyze textual expressions to enhance accuracy.
• **REQ-6:** Detected emotions should be categorized into a range of moods, such as happy, sad, energetic, etc.

2.4.3 Song Recommendation Based on User's Emotion

Description and Priority

The Song Recommendation feature leverages detected user emotions to provide tailored music suggestions. This is of High priority as it directly contributes to the core functionality of personalized music discovery.

Stimulus/Response Sequences

• **Stimulus:** User expresses interest in music recommendations based on their current mood.
• **Response:** The system, having detected the user's emotion, communicates with the last.fm API to fetch and display music suggestions aligned with the user's emotional state.

Functional Requirements

• **REQ-7:** The system must integrate with music databases, such as last.fm, for a diverse collection of song recommendations.
• **REQ-8:** Music suggestions should be dynamically updated based on real-time user emotions.

2.4.4 Web Interface with HTML Integration

Description and Priority

The inclusion of a Web Interface feature, integrated with HTML, CSS, and JavaScript, aims to improve user engagement by offering an attractive and smooth interface for exploring music. This feature holds moderate importance as it supplements the system's main functions, offering users an additional way to convey their mood and receive music suggestions.

Stimulus/Response Sequences

• **Stimulus:** User accesses the system through a web browser.

• **Response:** The system showcases an appealing web interface designed with HTML, CSS, and JavaScript enabling users to engage with the chatbot and discover music recommendations interactively.

• **Stimulus:** User interacts with the web interface to input mood preferences.

• **Response:** System captures user inputs and triggers the chatbot-driven music recommendation process.

Functional Requirements

• **REQ-9:** The system must integrate with the Html,Css platform to build a responsive and user-friendly web interface.

• **REQ-10:** The web interface should provide similar functionality to the chatbot interface, allowing users to express their mood and receive music recommendations.

• **REQ-11:** The design of the web interface should adhere to modern GUI standards and provide a cohesive visual experience.

• **REQ-12:** Error handling mechanisms should be in place to manage issues related to web interface interactions.

2.5 Other Nonfunctional Requirements

2.5.1 Performance Requirements:

• **Response Time:** The system should respond to user queries within a reasonable time frame, ensuring minimal latency.

• **Scalability:** The system should be able to handle varying loads of user requests without a significant degradation in performance.

• **Throughput:** It should be capable of processing multiple requests simultaneously to accommodate concurrent users efficiently.

2.5.2 Security Requirements:

• **Authentication and Authorization:** Users should be properly authenticated before accessing personalized recommendations, and access should be restricted based on roles and permissions.

• **Data Privacy:** Ensure that user data is stored securely and handled in compliance with relevant privacy regulations.

• **Secure Communication:** All communication between the user and the system should be encrypted to prevent unauthorized access or data breaches.

2.5.3 Safety Requirements:

Data Protection:

• Privacy Preservation: Implement measures to protect user privacy and ensure that personally identifiable information (PII) is securely handled and stored.
• Data Encryption: Encrypt sensitive user data during transmission and storage to prevent unauthorized access or interception.
• Compliance with Regulations: Ensure compliance with relevant data protection regulations (e.g., GDPR, CCPA) to safeguard user data and privacy rights.

Prevention of Harm:

• User Controls: Provide users with options to customize content preferences and filter out undesirable recommendations based on their preferences and sensitivities.
• Content Moderation: Implement moderation mechanisms to detect and remove harmful or inappropriate content, ensuring a safe and positive user experience.

System Integrity:

• Data Validation: Perform thorough validation of user inputs and recommendations to prevent injection attacks, data corruption, or manipulation.
• Regular Audits: Conduct security audits and vulnerability assessments regularly to identify and address potential security weaknesses or vulnerabilities in the system.

Emergency Response:

• Incident Response Plan: Develop an incident response plan outlining procedures for handling security incidents, data breaches, or emergencies promptly and effectively.
• User Support: Provide users with clear instructions and support channels to report security concerns, suspicious activities, or incidents promptly.

2.5.4 Software Quality Attributes:

Reliability:

• The system should consistently perform its intended functions accurately and reliably over time, without unexpected failures or errors.
• Users should trust the system's recommendations and rely on its accuracy and consistency.

Usability:

• The system should be easy to use and navigate, with an intuitive user interface that guides users through the recommendation process seamlessly.
• Users should find it effortless to interact with the chatbot and understand its recommendations.

Scalability:

• The system should be capable of scaling to accommodate increasing numbers of users and growing data volumes without sacrificing performance or reliability.
• It should support horizontal and vertical scaling to distribute the workload effectively and optimize resource utilization.

Maintainability:

• The system should be modular and well-structured, allowing for easy maintenance, updates, and enhancements without disrupting its overall functionality.
• Comprehensive documentation and code comments should be provided to facilitate understanding and modification by developers.

Availability:

• The system should be available and accessible to users whenever they need music recommendations, with minimal downtime or service interruptions.
• Redundancy measures and failover mechanisms should be in place to ensure continuous service availability and resilience to hardware failures or network disruptions.

Chapter 3

System Architecture and Design

3.1 System Overview

The SerenadeBot system architecture is designed to seamlessly integrate various components to deliver personalized music recommendations based on user interactions and emotional cues. This section provides a comprehensive overview of the project architecture

Architecture Diagram

The following diagram illustrates the high-level architecture of the SerenadeBot system:

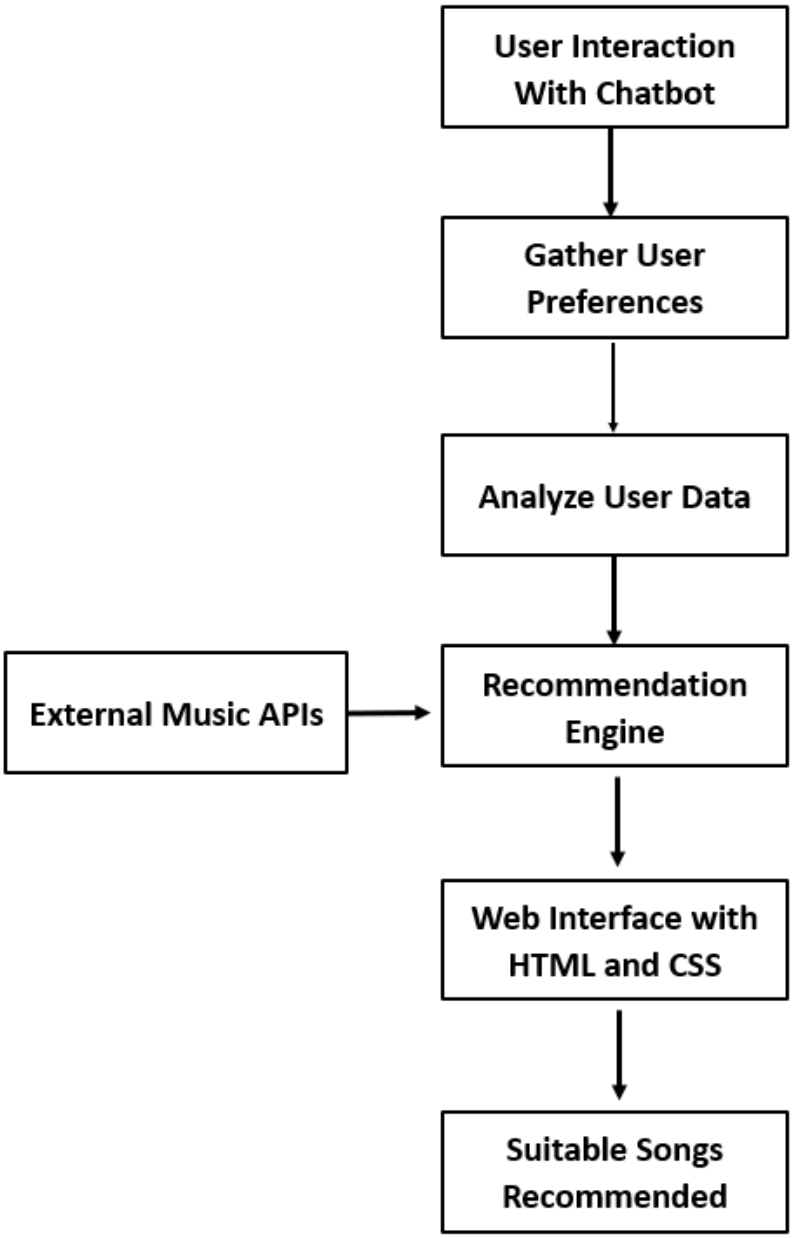


Figure 3.1: Architecture Diagram

System Components

- User Interface (Html,Css): The web-based user interface allows users to interact with the SerenadeBot chatbot, input their mood preferences, and receive personalized music recommendations.

- Chatbot (NLP Integration): The pretrained chatbot component utilizes natural language processing (NLP) techniques to understand user inputs, engage in conver-

sational interactions, and extract relevant emotional cues.

- Emotion Detection Algorithms: Emotion detection algorithms, such as NLTK (Natural Language Toolkit), analyze user conversations to identify emotional states expressed by users during interactions.

- Last.fm API (Music Database): The system interfaces with the Last.fm API to retrieve music metadata, including song titles, artists, genres, and recommendations based on user preferences and emotional context.

- Recommendation Engine: The recommendation engine processes user preferences, interaction data, and emotion analysis results to generate tailored music recommendations aligned with the user's mood.

- User Preferences and Interaction Data: Data collected from user interactions, including mood preferences, chat history, and feedback, are used to refine the recommendation engine and improve the overall user experience.

Detailed Process Outline

- User Interaction: Users initiate interactions through the web-based user interface, starting a conversation with the SerenadeBot chatbot.

- Natural Language Processing (NLP): The pretrained chatbot employs NLP techniques to parse user inputs, extract meaningful information even from colloquial terms, and engage in contextually relevant conversations.

- Emotion Analysis: Emotion detection algorithms analyze textual expressions within user conversations to categorize emotional states (e.g., happy, sad, energetic).

- Music Recommendation Retrieval: Based on detected emotions and user preferences, the system interacts with the Last.fm API to fetch relevant music recommendations.

- Personalized Music Recommendations: The recommendation engine processes retrieved music data and user interaction history to generate personalized song suggestions aligned with the user's current mood.

- User Feedback and Iteration: User feedback, including likes, dislikes, and interactions, is incorporated into the recommendation engine to continuously refine and improve music recommendations for future interactions.

3.2 Architectural Design

Sequence Diagram

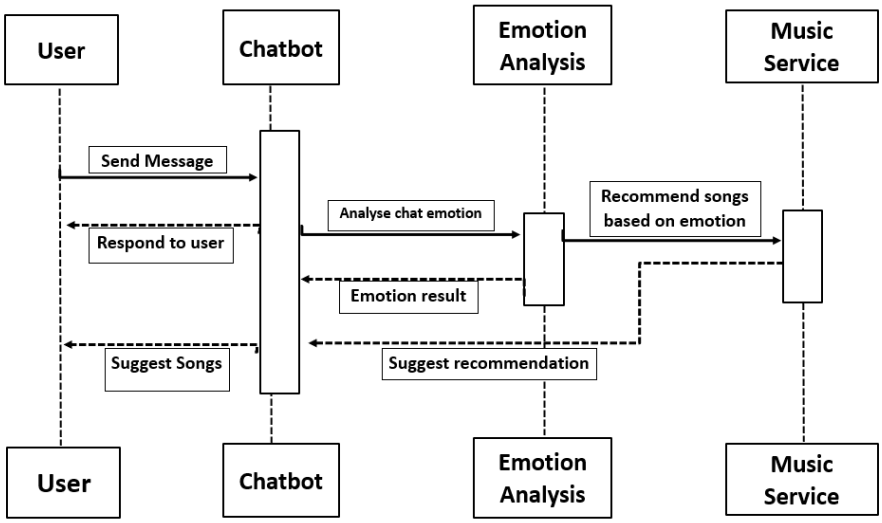


Figure 3.2: Sequence Diagram

Use Case Diagram

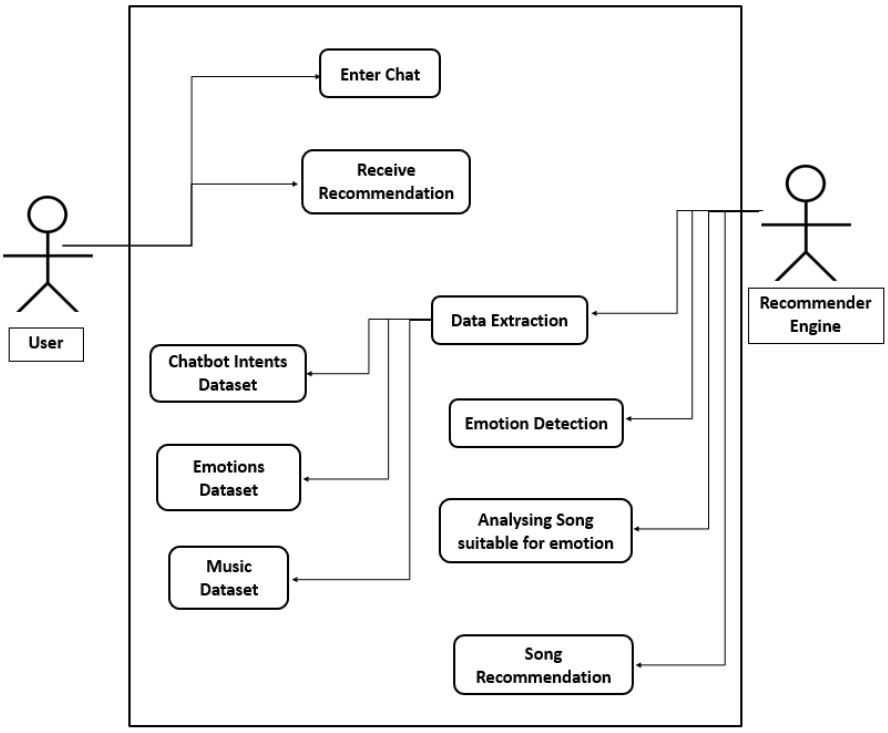


Figure 3.3: Use Case Diagram

State Diagram

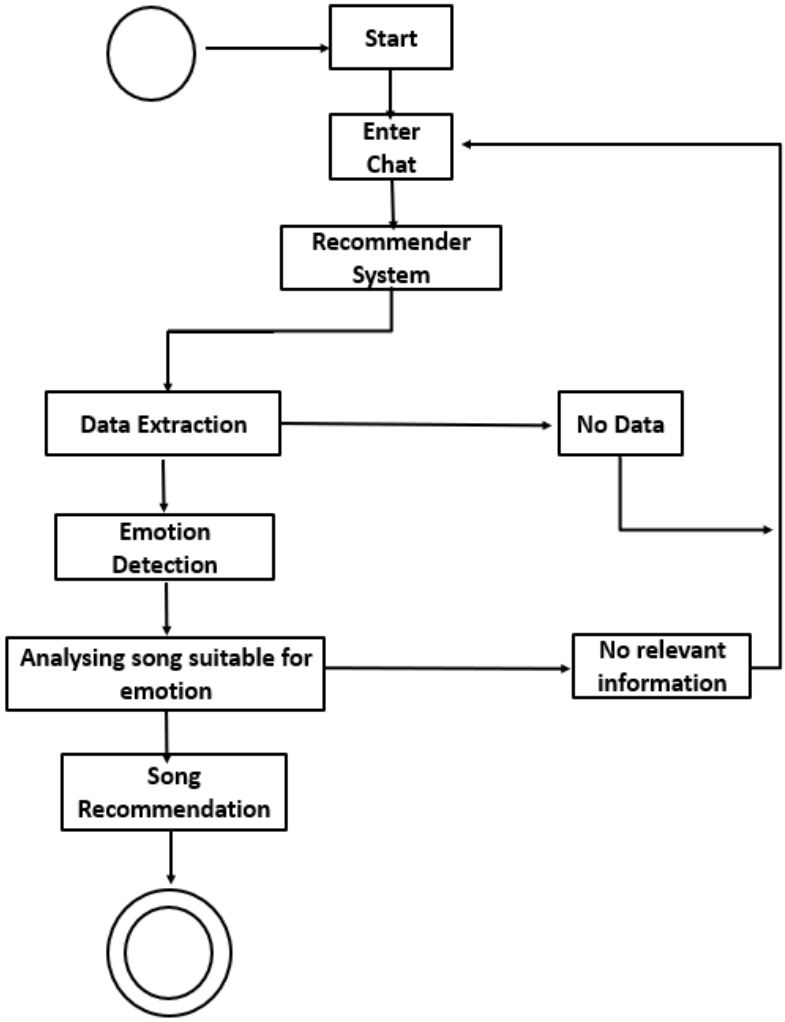


Figure 3.4: State Diagram

3.3 Dataset identified

Chatbot Intent Dataset:

Chatbot intents refer to the underlying goals or purposes users have when interacting with a chatbot. By accurately recognizing these intents, chatbots can provide more relevant and efficient responses, leading to a smoother user experience.

When a user chats with a chatbot, their message is analyzed to understand their goal (intent) using Natural Language Understanding. This lets the chatbot respond directly to their needs, whether it's providing information, completing a task, or asking for clar-

ification. This focus on intent improves the user experience, increases chatbot efficiency, and ultimately leads to higher customer satisfaction.

Emotion Analysis Dataset:

Emotion analysis datasets are collections of data specifically designed to train and test machines in understanding and classifying emotions.

- Understanding User Emotions:
  These datasets can help train algorithms to recognize emotions expressed through text .
- Matching Music to Mood:
  By linking emotions to music attributes , the recommender system can suggest songs that align with the user's current mood .
- Building a Richer Experience:
  Emotion-based recommendations can create a more personalized and engaging music listening experience.

Last.fm API

Last.fm offers a free API that allows developers to access their vast music database and integrate it with their projects.

- Leverage user interactions:
  Analyze the user's conversation to understand their mood or preferred genres (e.g., "feeling down today").
- Query Last.fm API:
  Based on the identified preferences, use the Last.fm API to:
  - Find similar artists or tracks.
  - Explore top tracks within a specific genre.
  - Discover artists related to user-mentioned artists.

- Personalized Suggestions:
  Tailor recommendations by incorporating the user's conversation and understanding their mood.

- Respond with Options:
  Present the user with several music suggestions from the Last.fm results.

3.4 Proposed Methodology/Algorithms

The Natural Language Processing (NLP) emotion analysis algorithm is a methodology used to identify and extract emotions expressed within text data. Steps involved in an NLP emotion analysis algorithm:

Text Preprocessing:

1. Tokenization: The text is divided into individual words or tokens.
2. Lowercasing: All words are converted to lowercase to ensure consistency.
3. Removing Punctuation: Punctuation marks such as commas, periods, and exclamation points are removed.
4. Stopword Removal: Commonly occurring words (stopwords) such as "is", "the", "and" are removed as they often do not carry significant meaning in emotion analysis.

Emotion Lexicon:

An emotion lexicon or dictionary is used, containing words associated with specific emotions. Each word in the lexicon is labeled with one or more emotions it represents.

Emotion Extraction:

Each word in the preprocessed text is compared against the words in the emotion lexicon. If a word from the text matches a word in the emotion lexicon, the associated emotion(s) are recorded.

Emotion Aggregation:

The frequencies of different emotions detected in the text are aggregated or counted. This step involves creating a summary of the emotions expressed in the text and their relative intensities.

Visualization and Analysis:

The results of the emotion analysis may be visualized using graphs or charts to illustrate the distribution of emotions in the text. Further analysis can be conducted to explore patterns, trends, or correlations between emotions and other variables.

3.5 User Interface Design

The user interface (UI) design of the SerenadeBot system plays a crucial role in providing a seamless and intuitive experience for users interacting with the chatbot and exploring music recommendations.

The web interface provides a platform for users to interact with SerenadeBot, accessible through standard web browsers. It allows users to engage in natural language conversations and receive music recommendations based on their mood and preferences. Key features of the web interface include:

- Emotion Indicator: A visual indicator (e.g., emoji) displays the detected emotion of the user's messages.
- Submit Button: Initiates the mood-based music recommendation process upon user input.
- Music Recommendation Display: Presents recommended songs with metadata (title, artist, album) in a visually appealing format.
- Input Box for User Messages: Positioned at the bottom of the interface, users can type messages to interact with the chatbot with a send button.
- Chat Message Display: Conversation history is displayed in a chat-style format, showing both user messages and chatbot responses.

- Recommendation Display: Music recommendations are presented as clickable options within the chat interface.

Figure 1: Web Interface Wireframe The wireframe depicted below illustrate the layout and functionality of the user interfaces for SerenadeBot, emphasizing usability and clarity in user interactions.

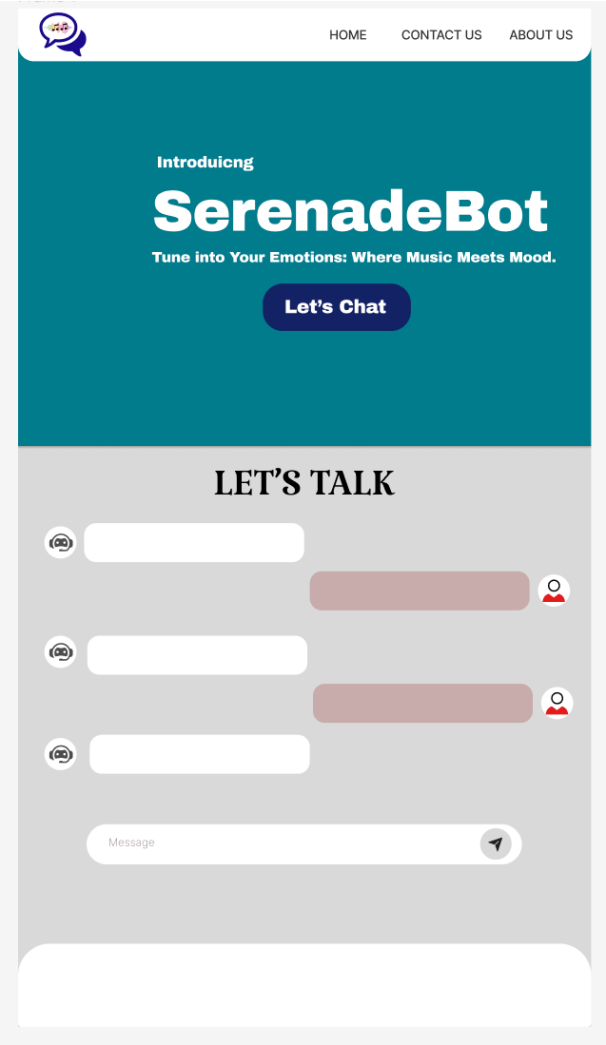


Figure 3.5: Wireframe

3.6 Database Design

In the development of the Chatbot Song Recommendation System, we will utilize the Last.fm API as a primary source of music-related data. While Last.fm itself manages a comprehensive database of music metadata, user preferences, and community-generated

content, our system will interact with this database through API calls rather than directly managing its schema.

Database Used: Last.fm API

Schema Explanation: The Last.fm API serves as a centralized repository for music information and user data. It encompasses the following key components:

- Music Metadata: Details about artists, albums, and tracks, including titles, genres, release dates, and popularity metrics.
- User Preferences and Profiles: User-specific data capturing preferences, favorite genres, artists, and mood associations.
- Community-Generated Tags: Tags assigned by users to categorize music based on genres, moods, and themes.

Reason for Choosing the Last.fm API:

- Rich Music Data: Last.fm provides extensive music metadata and user-related information, which is crucial for building an effective song recommendation system.
- Community-Driven Recommendations: Leveraging Last.fm's community-generated data (e.g., user scrobbles, tags) allows us to tap into collective user preferences and trends.
- API Accessibility: The Last.fm API offers straightforward access to its database through well-documented endpoints, making it suitable for integration into our chatbot system.
- Scalability and Reliability: Last.fm's infrastructure ensures scalability and reliability, accommodating a large volume of user interactions and data queries.

3.7 Description of Implementation Strategies

The implementation of the chatbot song recommendation system involves several key steps and strategies:

• Libraries Used: The implementation relies on various Python libraries for different tasks. NLTK is used for text processing, including tokenization, lemmatization, and stemming. Keras is employed for building and training deep learning models, specifically neural networks. Other libraries like json, pickle, numpy, and random are utilized for tasks such as data manipulation, serialization, numerical computing, and random number generation.

```
from google.colab import drive
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
# from keras.optimizers import SGD
from tensorflow.keras.optimizers import SGD
import random
import nltk
```

• **Preprocessing**: Before building the model, the text data needs to be preprocessed. This involves tasks such as tokenizing sentences into words, lemmatizing words to their base forms, and preparing the corpus of text data. The corpus consists of patterns from the intents.json file, with each pattern associated with a specific intent. Unique words and intents are extracted from the corpus and stored in separate lists for further processing. Snippet:

```
import nltk
nltk.download('punkt')
words=[]
classes = []
documents = []
ignore_letters = ['!', '?', ',', '.']
for intent in intents['intents']:
    for pattern in intent['patterns']:
        #tokenize each word
        word = nltk.word_tokenize(pattern)
        words.extend(word)
        #add documents in the corpus
        documents.append((word, intent['tag']))
        # add to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

• **Model Loading and Inference:** Using the pre-trained model it can be loaded for inference. The pretrained model, along with other necessary data such as words, classes, and intents, is loaded into memory using appropriate functions from libraries like Keras, pickle, and json. User input is preprocessed in a similar manner to the

training data, generating a bag-of-words representation. This representation is then fed into the loaded model to make predictions about the user's intent. Based on the predicted intent, a response is selected from the list of responses associated with that intent.

```
def clean_up_sentence(sentence):
    # tokenize the pattern - splitting words into array
    sentence_words = nltk.word_tokenize(sentence)
    # stemming every word - reducing to base form
    sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
    return sentence_words
# return bag of words array: 0 or 1 for words that exist in sentence
def bag_of_words(sentence, words, show_details=True):
    # tokenizing patterns
    sentence_words = clean_up_sentence(sentence)
    # bag of words - vocabulary matrix
    bag = [0]*len(words)
    for s in sentence_words:
        for i,word in enumerate(words):
            if word == s:
                # assign 1 if current word is in the vocabulary position
                bag[i] = 1
                if show_details:
                    print ("found in bag: %s" % word)
    return(np.array(bag))
```

- **Integration with last.fm API**: The chatbot song recommendation system integrates with the last.fm API to provide music recommendations based on the user's mood expressed in their conversation with the chatbot. The API allows access to a vast database of music-related information, enabling the system to recommend relevant songs based on the user's current emotional state.
- **Enhancing User Experience**: To enhance user interaction and experience, the implementation could integrate tools like HTML, CSS, and JavaScript for constructing web user interfaces. These technologies allow users to access the music recommendation system through a user-friendly web interface, improving accessibility and usability for discovering personalized music based on their emotions.

Overall, the implementation aims to leverage advanced technology, including natural language processing (NLP) and deep learning, to create a chatbot-driven music recommendation system. By analyzing the user's mood and preferences expressed in their conversation with the chatbot, the system provides personalized music recommendations, ultimately enhancing music discovery and stress relief for users.

3.8 Module Division

1. **Text Preprocessing Module**

- Responsible for preprocessing user input text, including tokenization and lemmatization.
- Handles the loading of the pretrained model and necessary data (words, classes, intents) for inference.
- Preprocesses user input and makes predictions using the loaded model and selects responses based on predicted intents.
- Assigned to: Aedna Mary Reji

2. **Integration with last.fm API Module**

- Integrates the chatbot with the last.fm API for music recommendations based on user mood.
- Fetches music recommendations from the last.fm database based on user interactions with the chatbot.
- Assigned to: Adithyan Darshan Kidav

3. **Emotional Analysis Module (NLP with NLTK)**

- Utilizes NLTK for emotional analysis of user chats to determine the user's mood.
- Performs sentiment analysis, emotion detection, or tone analysis on user input text.
- Extracts emotional features that influence music recommendations.
- Assigned to: Alan Anu Sam

4. **User Interface Module**

- Implements the web-based user interface using Html,Css, allowing users to interact with the chatbot and receive music recommendations.
- Enhances user experience and accessibility by providing a user-friendly interface.
- Assigned to: Allwyn Antony Rodrigues

3.9 Work Schedule - Gantt Chart

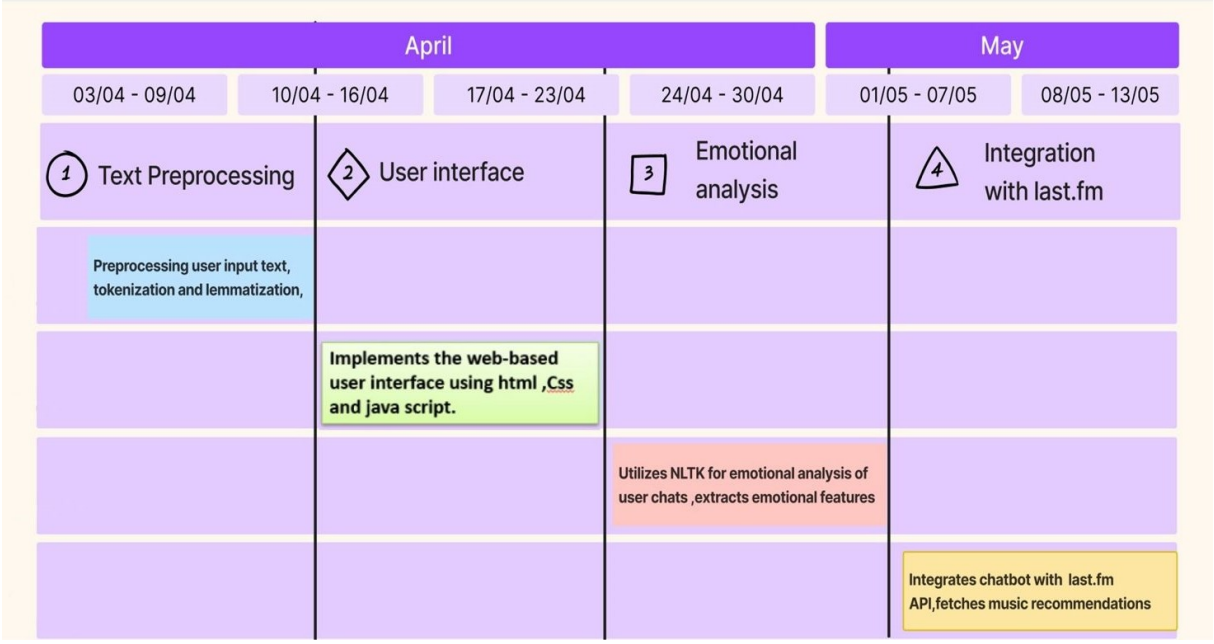


Figure 3.6: Gantt Chart

Chapter 4

Results and Discussion

4.1 Overview

The chatbot utilizes a trained neural network model to predict intents and provide appropriate responses from a predefined set of intents stored in a JSON file. Additionally, it employs sentiment analysis using NLTK's VADER module to detect emotions in user messages, modifying the sentiment if negation terms are present. The application also interacts with the Last.fm API to search for tracks associated with detected emotions. It includes routes for various functionalities like retrieving chatbot responses, getting emotion counts, and searching for tracks based on a specified emotion. The application's success could be assessed based on user engagement, the accuracy of chatbot responses, and the relevance of recommended tracks to users' expressed emotions. Further analysis could involve tracking user interactions, sentiment trends over time, and refining the emotion detection and recommendation algorithms for improved user experience.

4.2 Testing

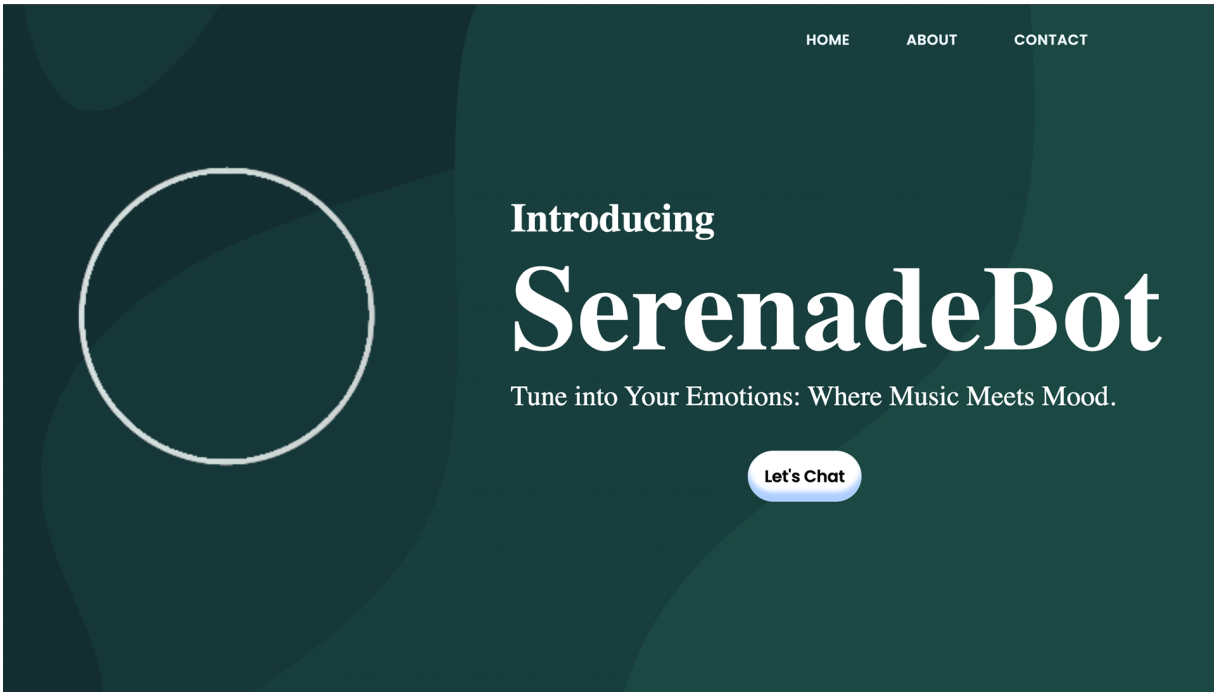


Figure 4.1: Home Page

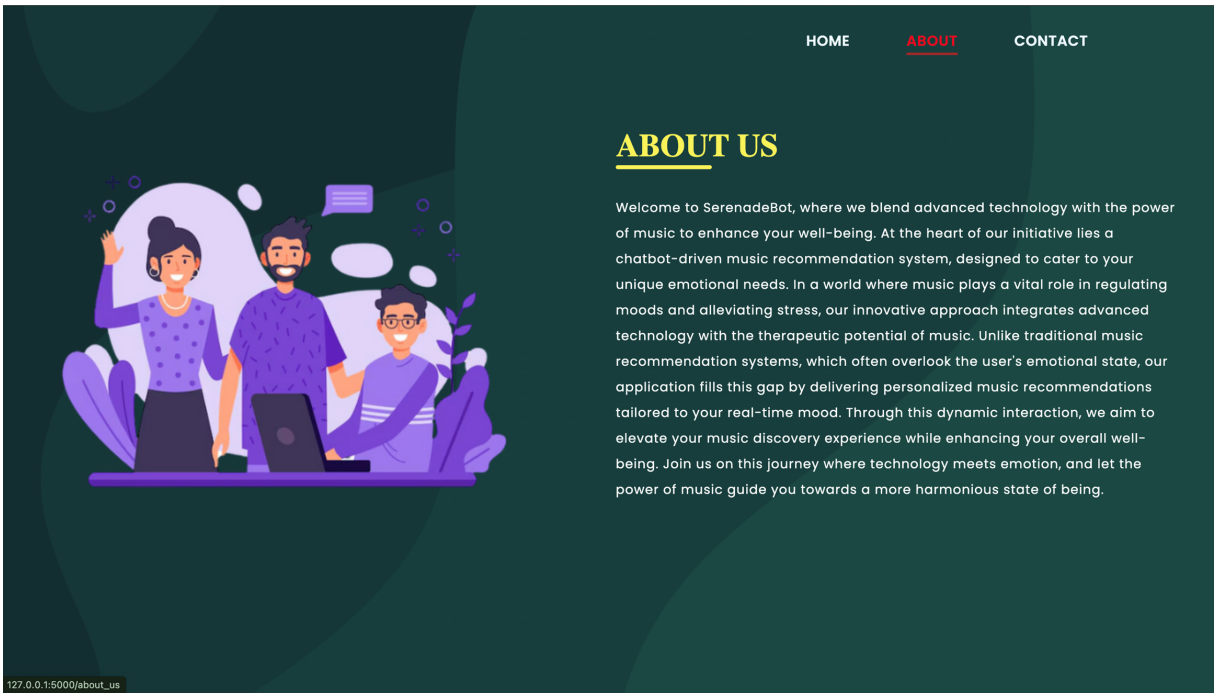


Figure 4.2: About-Us Page

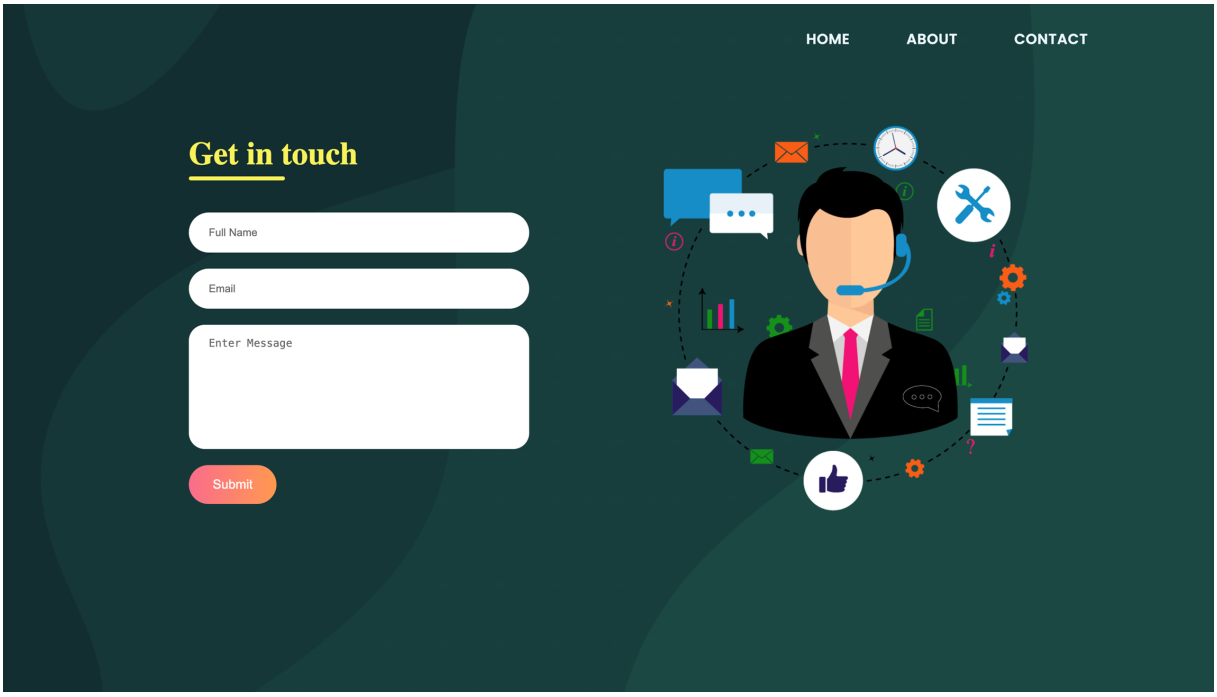


Figure 4.3: Contact-Us Page

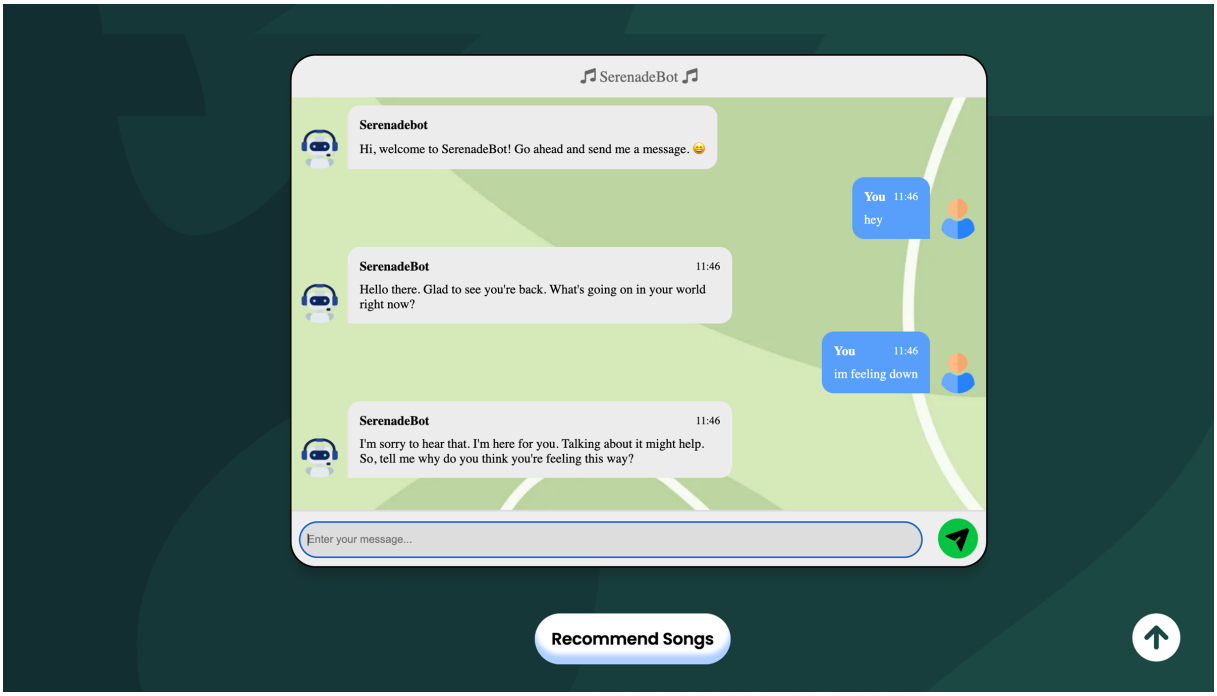


Figure 4.4: Chatbot Interface

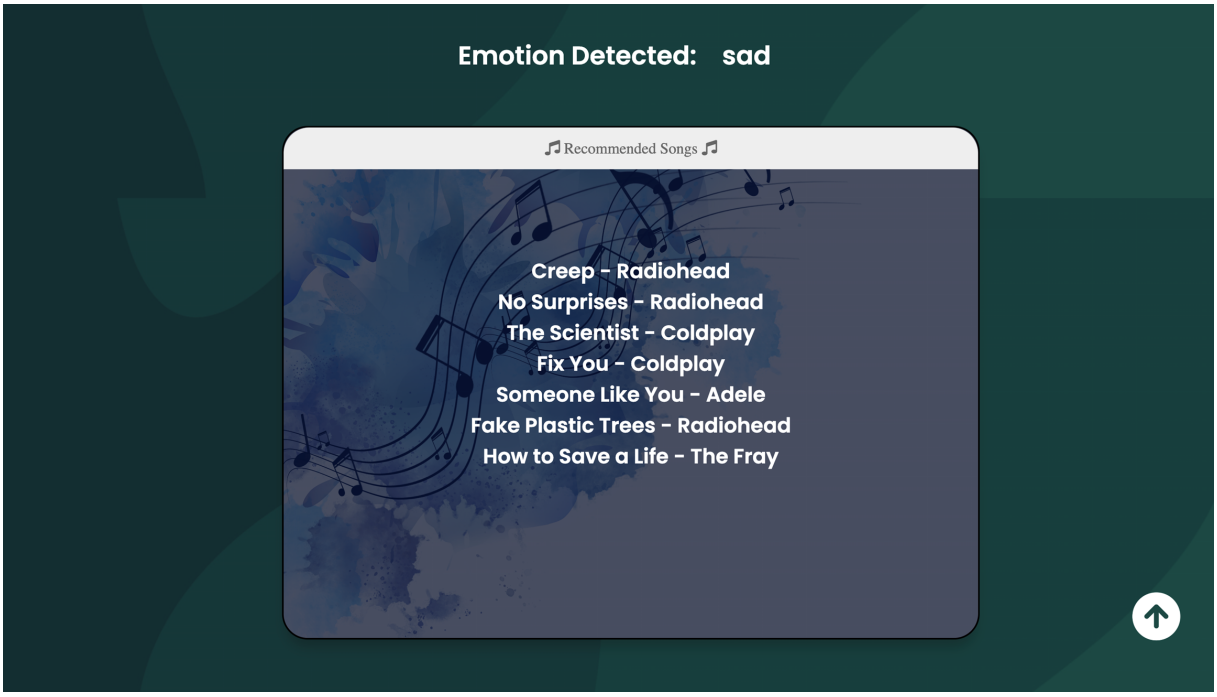


Figure 4.5: Emotion Detection And Song Recommendation

4.3 Discussion

The chatbot is equipped with emotion analysis capabilities, offering users a personalized interaction experience. Utilizing a pre-trained Keras model, the chatbot accurately classifies user intents based on input messages, responding with relevant information or prompts retrieved from a predefined set of intents. Additionally, the application integrates emotion analysis, employing natural language processing techniques to detect sentiments in user messages. By considering negations and mapping words to predefined emotions, it tailors responses to users' emotional states, enhancing the conversational interface's effectiveness and user engagement.

Moreover, the application seamlessly integrates with the Last.fm API to enhance user experience by providing personalized music recommendations based on detected emotions. Leveraging Last.fm's vast database, the application retrieves top tracks corresponding to the predominant emotion identified in user interactions. Through this integration, users not only receive tailored responses but also discover new music that resonates with their emotional context, enriching their overall interaction with the chatbot and fostering a more immersive experience.

Chapter 5

Conclusion

5.1 Conclusion

The Chatbot Song Recommendation System is a sophisticated platform designed to revolutionize music discovery and stress relief through personalized song suggestions. By leveraging advanced algorithms, it delivers tailored music recommendations based on users' preferences and mood, offering a unique and enriching listening experience. Whether users are seeking upbeat tracks to lift their spirits or calming melodies to unwind, the system intuitively understands their emotional state and curates playlists that resonate with their mood, fostering a deeper connection with music and enhancing overall well-being.

This innovative system boasts a Seamless Interaction mechanism, featuring a web-based interface crafted with HTML and CSS for an intuitive and real-time interaction with the chatbot. Through this interface, users can effortlessly engage with the chatbot, exploring music suggestions and refining their preferences with ease. The seamless integration of user-friendly design elements ensures a smooth and enjoyable user experience, encouraging active participation and facilitating effortless music discovery. Whether users are accessing the platform from their desktop or mobile devices, the web-based interface offers a consistent and immersive experience, making music exploration both accessible and enjoyable.

Furthermore, the Chatbot Song Recommendation System goes beyond conventional music platforms by prioritizing user engagement and satisfaction. By continually analyzing user interactions and feedback, it adapts its recommendations to meet evolving preferences and mood states, ensuring that every song suggestion resonates with users on a personal level. This commitment to personalized service not only enhances music discovery but also promotes stress relief and emotional well-being, establishing the system as

a trusted companion in users' daily lives. Through its innovative features and user-centric approach, the Chatbot Song Recommendation System redefines the way users engage with music, offering a dynamic and fulfilling music discovery experience.

5.2 Future Scope

• Implementation of machine learning algorithms to improve music recommendation accuracy by learning from user feedback and behavior patterns.
• Integration of additional APIs for accessing diverse music sources and expanding the scope of recommendations beyond last.fm.
• Enhancement of the chatbot's NLP capabilities to better understand user context and provide more sophisticated responses tailored to individual preferences and emotions.

Bibliography

[1] Rumiantcev, Mikhail, and Oleksiy Khriyenko. "Emotion based music recommendation system." In Proceedings of Conference of Open Innovations Association FRUCT. Fruct Oy, 2020.

[2] Parikh, Dhruv Piyush, Shatakshi Shree, and Abhay Arora. "Emotion based music recommendation system." Int J Res Appl Sci Eng Technol 9 (2021): 1674-1680.

[3] Vidyasagar, Bagde Sarika, and V. Karwande. "Emotion based music recommendation system by using different ml approach." International journal 6, no. 6 (2021).

[4] James, H. Immanuel, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, and R. Saranya. "Emotion based music recommendation system." Emotion 6, no. 3 (2019).

[5] Mahadik, Ankita, Shambhavi Milgir, Janvi Patel, Vijaya Bharathi Jagan, and Vaishali Kavathekar. "Mood based music recommendation system." International Journal of Engineering Research & Technology (IJERT) Volume 10 (2021).

[6] Phaneendra, A., Madhusmitha Muduli, Siri Lakshmi Reddy, and R. Veenasree. "EMUSE–An emotion based music recommendation system." International Research Journal of Modernization in Engineering Technology and Science 4, no. 5 (2022): 4159-4163.

Appendix A: Presentation

SerenadeBot

Team Members:
•Adithyan Darshan Kidav (U2103016)
•Aedna Mary Reji (U2103017)
•Alan Anu Sam (U2103020)
•Allwyn Antony Rodrigues (U2103028)

Guided By: Ms. Meenu Mathew
Assistant Professor, DCS

1

Contents

2

Introduction

- Music significantly impacts well-being, aiding mood regulation and stress relief. Technological advancements allow personalized music discovery.
- Our project merges technology and music discovery through a chatbot-driven system.
- By analyzing user mood in chat conversations, we customize recommendations using last.fm API and NLP.
- Traditional systems overlook emotional states, resulting in less effective suggestions.
- Our project fills this gap by offering tailored recommendations based on real-time mood analysis, improving music discovery and stress management.

3

Problem Definition

A chatbot-driven music recommendation system that analyzes user mood through conversation and recommends soothing music tailored to users' emotions, enhancing music discovery and stress relief through a chatbot interface integrated with last.fm API and NLP techniques.

4

Objectives

1.Emotion Analysis:
- Implement NLP and sentiment analysis to understand user emotions.

2.Integration with Last.fm API:
- Connect to Last.fm API for music recommendations based on user emotions.

3.User Interface:
- Design an intuitive, visually appealing interface using HTML and CSS.

4.Chatbot Interface Development:
- Develop user-friendly chatbot interface using a pre-trained Keras sequential model.

5

Scope and Relevance

•Chatbot-driven music recommendation system integrates last.fm API

•Utilizes NLTK and NLP for emotion analysis

•Developed with a web UI using Html, Css, and JavaScript

•Offers personalized music recommendations based on user mood

•Relevant for personalized recommendations, stress relief through curated music, showcasing NLP applications

•Useful for music enthusiasts, stress relief seekers, and developers interested in NLP-driven systems

6

System Design

System Overview :

1.User Interface: Facilitates text input and music recommendations.

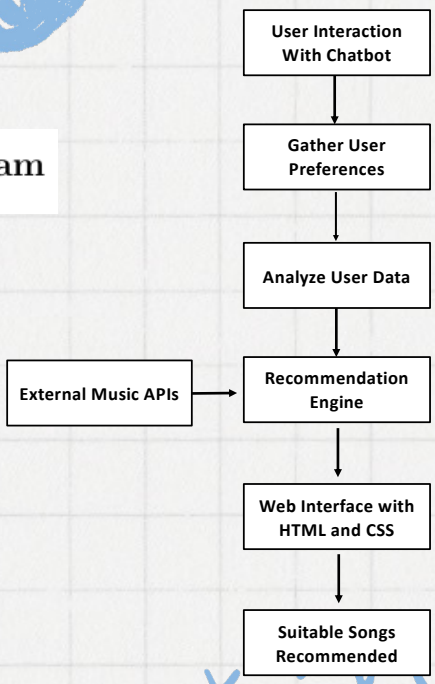
2.Emotion Analysis: Detects user mood through NLP.

3.Recommendation Engine: Suggests music based on detected emotions.

4.Chatbot: Integrates pretrained chatbot for smooth communication and advanced conversation.

7

Architecture Diagram



8

System Design

Module Division

1.Text Preprocessing Module:

- •Prepares user input for analysis.
- •Breaks text into smaller parts (tokenization).
- •Simplifies words to their base form (lemmatization).
- •Loads pre-trained model and necessary data for understanding user queries.
- •Uses model to predict user intent and select responses.

2.Integration with Last.fm API Module:

- •Connects chatbot to Last.fm API, a vast music library.
- •Fetches music recommendations based on user's emotions.
- •Retrieves recommendations from Last.fm database.
- •Shares recommendations with user through chatbot.

9

System Design

3.Emotional Analysis Module (NLP with NLTK):

- •Utilizes NLTK for natural language processing.
- •Performs sentiment analysis, emotion detection, and tone analysis to understand user mood.
- •Helps suggest music matching user's emotions.

4.User Interface Module:

- •Creates chatbot's visible and interactive part using HTML, CSS, and Flask.
- •Aims for easy and enjoyable chatting experience for users.
- •Ensures accessibility and enjoyment of music suggestions through user-friendly interface.

10

System Design

NLP Emotion Analysis Algorithm Steps:

1.Text Preprocessing:

- Tokenization: Divide text into words or tokens.
- Lowercasing: Convert all words to lowercase.
- Removing Punctuation: Eliminate punctuation marks.
- Stopword Removal: Exclude common words lacking significant meaning.

2.Emotion Lexicon:

- Use lexicon/dictionary with words linked to specific emotions.

11

System Design

3. Emotion Extraction:

- Match words from text with lexicon entries to identify associated emotions.

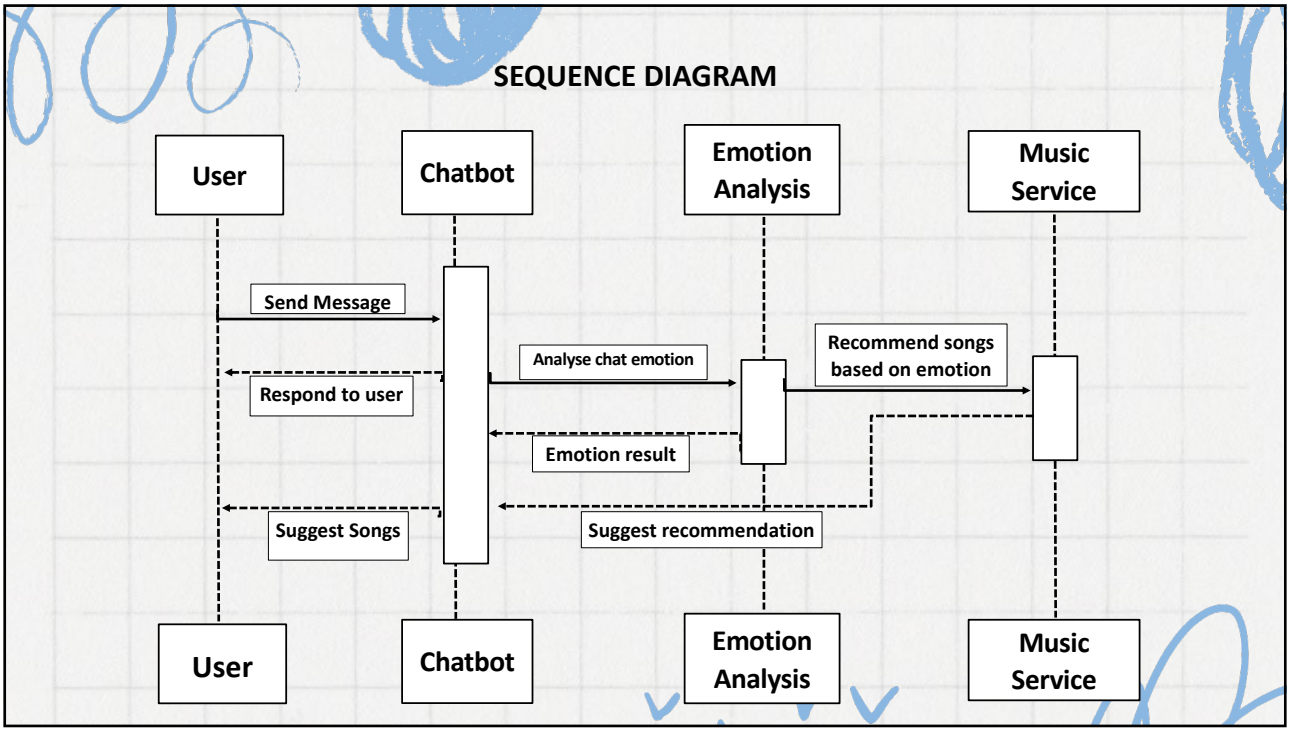
4. Emotion Aggregation:

- Count frequencies of detected emotions to summarize their distribution.

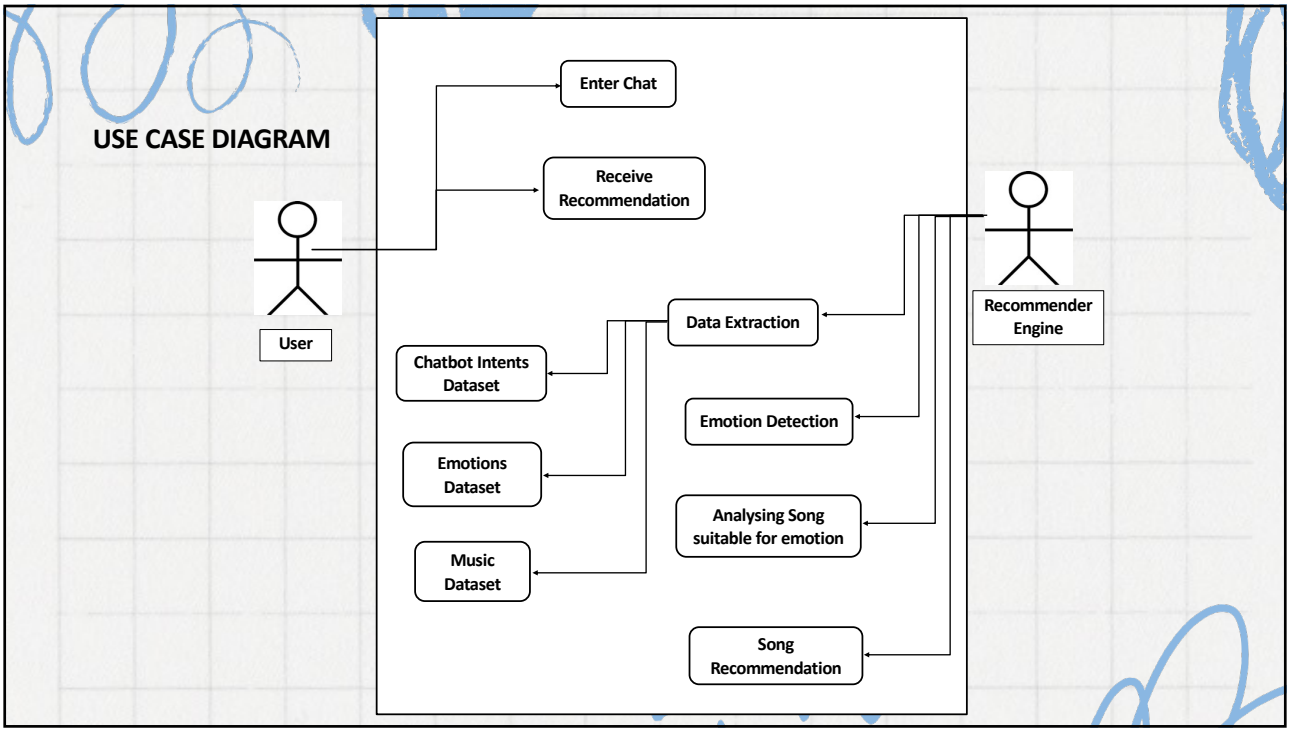
5. Visualization and Analysis:

- Evaluate results by comparing dominant emotions.
- Explore patterns or correlations between emotions and other factors.

12

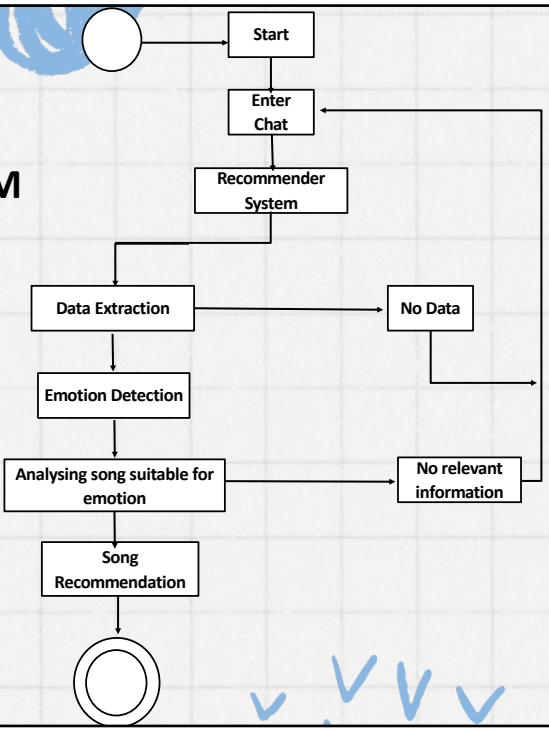

SEQUENCE DIAGRAM
User
Chatbot
Emotion Analysis
Music Service
Send Message
Respond to user
Analyse chat emotion
Emotion result
Recommend songs based on emotion
Suggest Songs
Suggest recommendation
User
Chatbot
Emotion Analysis
Music Service

13


USE CASE DIAGRAM
User
Enter Chat
Receive Recommendation
Recommender Engine
Data Extraction
Chatbot Intents Dataset
Emotion Detection
Emotions Dataset
Analysing Song suitable for emotion
Music Dataset
Song Recommendation

14

STATE DIAGRAM



15

Datasets

The intents.json dataset was sourced from Github and further modified by us. The dataset can be accessed through the following link: *BotDataset*


```
{"intents": [
    {"tag": "greeting",
     "patterns": ["Hi", "Hey", "Is anyone there?","Hi there", "Hello", "Hey there", "Howdy", "Hola", "Bonjour","Hay", "Sasa", "Good Evening", "Good afternoon"],
     "responses": ["Hello there. Tell me how are you feeling today?", "Hi there. What brings you here today?", "Hi there. How are you feeling today?", "Great to see
    },
    {"tag": "morning",
        "patterns": ["Good morning"],
        "responses": ["Good morning. I hope you had a good night's sleep. How are you feeling today? "]
    },
    {"tag": "afternoon",
        "patterns": ["Good afternoon"],
        "responses": ["Good afternoon. How is your day going?"]
    },
    {"tag": "evening",
        "patterns": ["Good evening"],
        "responses": ["Good evening. How has your day been?"]
    },
    {"tag": "night",
        "patterns": ["Good night"],
        "responses": ["Good night. Get some proper sleep", "Good night. Sweet dreams."]
    },
```


16

UI Design

SerenadeBot UI is essential for smooth user experience. . Key features:

- **Emotion Indicator**: A visual indicator displays the detected emotion of the user's messages.
- **Submit Button:** Initiates the emotion-based music recommendation process upon user input.
- **Music Recommendation Display:** Presents recommended songs with metadata (title, artist, album) in a visually appealing format.

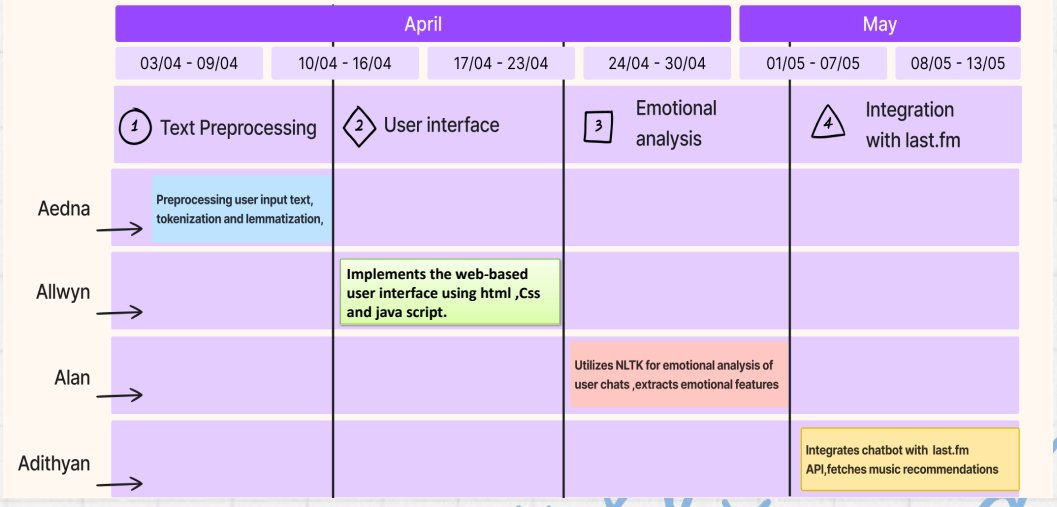
17

UI Design

- **Input Box for User Messages:** Positioned at the bottom of the interface, users can type messages to interact with the chatbot with a send button.
- **ChatMessage Display:** Conversation history is displayed in a chat-style format, showing both user messages and chatbot responses.
- **Recommendation Display:** Music recommendations are presented as clickable options within the chat interface.

18

Work Division - Gantt Chart



19

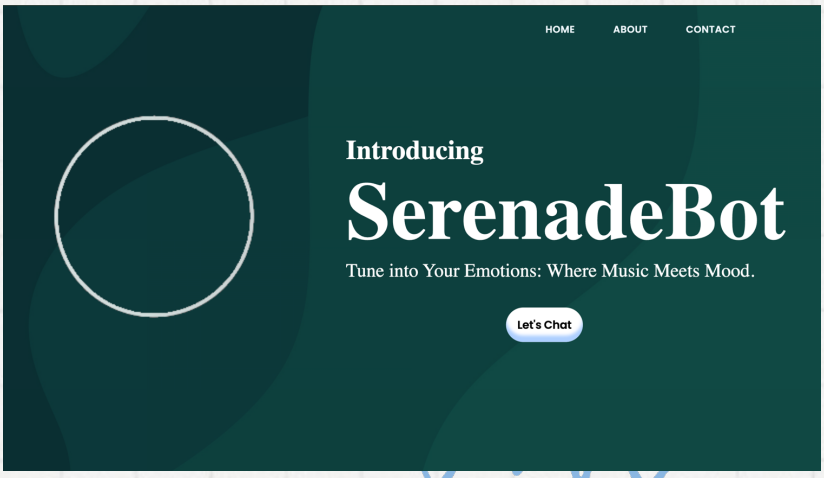
Software/Hardware Requirements

- **Hardware Platform:** Standard hardware for web applications.
- **Operating System:** Compatible with Windows, macOS.
- **Web Browser:** Google Chrome, Firefox, Safari, Edge.
- **UI Integration:** Develop web interfaces utilizing HTML, CSS, and Flask framework.
- **Internet Connection:** Stable connection for web interface and recommendations.

20

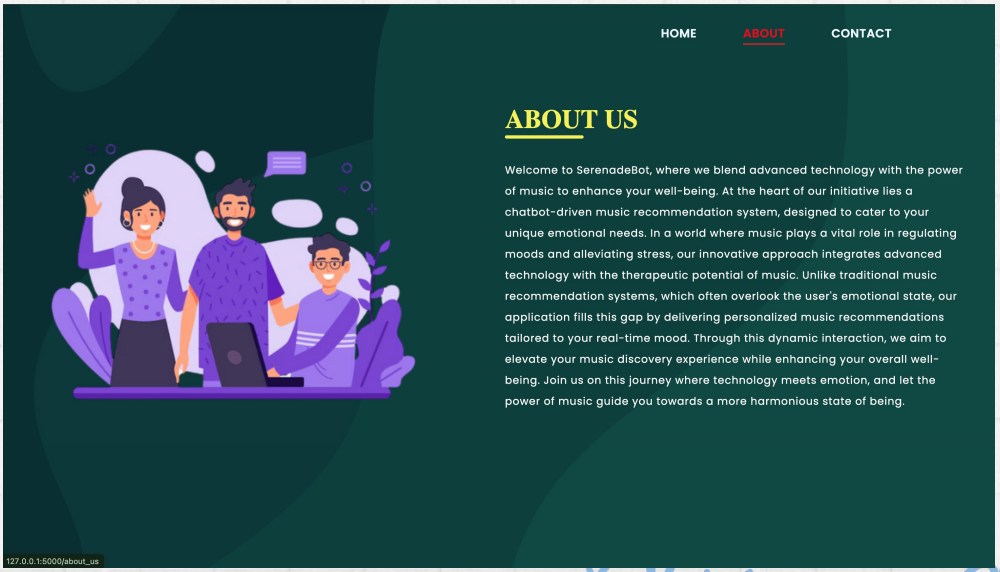
RESULTS

Home Page:



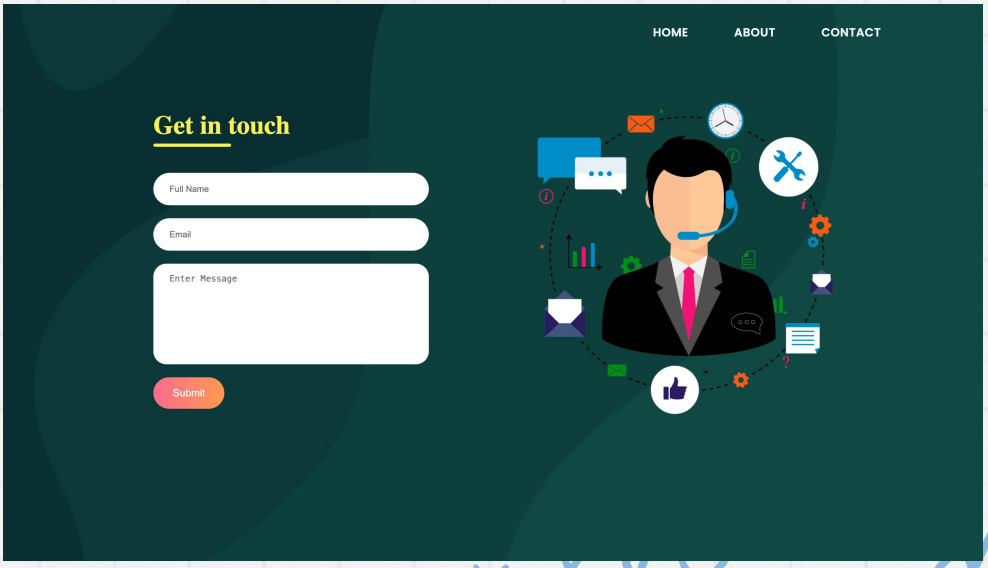
21

ABOUT US PAGE:



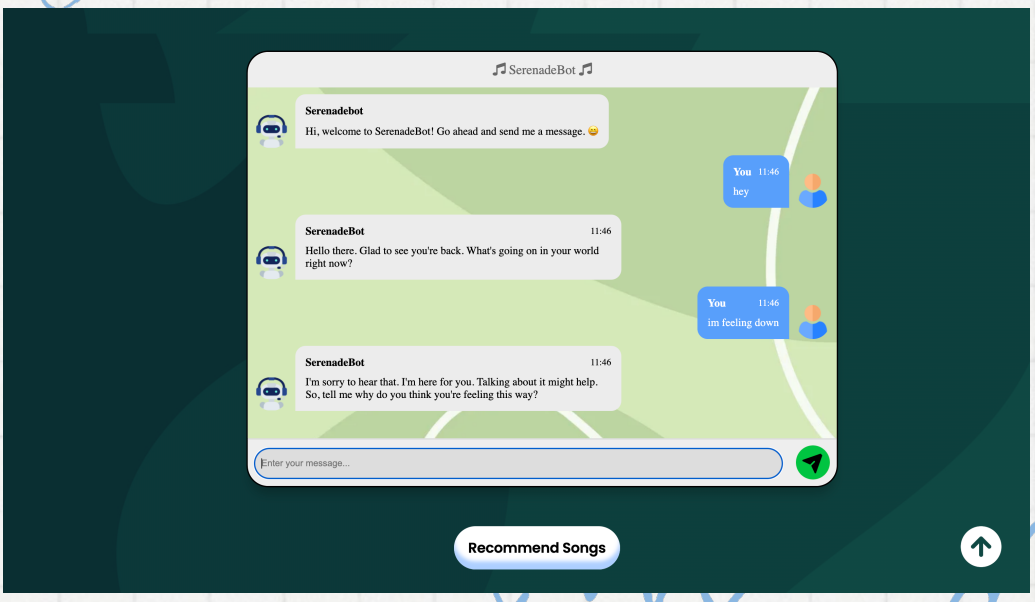
22

CONTACT PAGE:



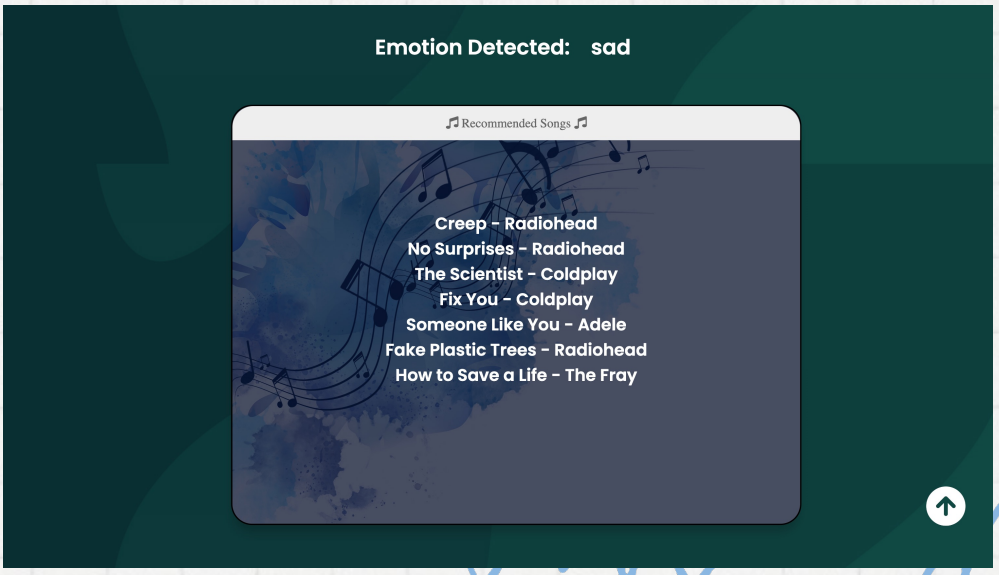
23

CHATBOT INTERFACE:



24

EMOTION DETECTION AND SONG RECOMMENDATION:



25

Conclusion

- A Chatbot Song Recommendation System that delivers personalized music suggestions based on user preferences and mood, enhancing music discovery and stress relief.
- It provides personalized song suggestions tailored to users' mood. And has Seamless Interaction , web-based interface with HTML and CSS for intuitive and real-time interaction with the chatbot, ensuring a smooth user experience.

26

Future Enhancements

- Implementation of better machine learning algorithms to improve music recommendation accuracy by learning from user feedback and behavior patterns.
- Integration of additional APIs for accessing diverse music sources and expanding the scope of recommendations beyond last.fm.
- Enhancement of the chatbot's NLP capabilities to better understand user context and provide more sophisticated responses tailored to individual preferences and emotions.

27

References

- Rumiantcev, Mikhail, and Oleksiy Khriyenko. "Emotion based music recommendation system." Proceedings of 26th Conference of Open Innovations Association FRUCT, Yaroslavl, Russia, 23-25 April 2020 (pp. 639-645).
- Parikh, Dhruv Piyush, Shatakshi Shree, and Abhay Arora. "Emotion based music recommendation system.", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 9 Issue X Oct 2021.
- Vidyasagar, bagde sarika, and v. Karwande. "Emotion based music recommendation system by using different ML approach." International Journal Of Advance Scientific Research And Engineering Trends, Volume 6 (2021).

28

References

- James, H. Immanuel, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, and R. Saranya. "Emotion based music recommendation system." International Research Journal of Engineering and Technology (IRJET), Volume 6 (2019).

- Mahadik, Ankita, Shambhavi Milgir, Janvi Patel, Vijaya Bharathi Jagan, and Vaishali Kavathekar. "Mood based music recommendation system.", International Journal of Engineering Research & Technology (Ijert), Volume 10 (2021).

- Phaneendra, A., Madhusmitha Muduli, Siri Lakshmi Reddy, and R. Veenasree. "EMUSE–An Emotion based music recommendation system.", International Research Journal of Modernization in Engineering Technology and Science 4, no. 5 (2022): 4159-4163.

29

Thank you very much!

30

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039
(Affiliated to APJ Abdul Kalam Technological University)

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

After the completion of the course the student will be able to:

CO1:

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

CO2:
Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

CO3:
Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

CO4:
Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

CO5:
Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	DESCRIPTION	Blooms' Taxonomy Level
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
101003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
101003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
101003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
101003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
101003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
101003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
101003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
101003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
101003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
101003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
101003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
101003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
101003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
101003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

101003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
101003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
101003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
101003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
101003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
101003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
101003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
101003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
101003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
101003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
101003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
101003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
101003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
101003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
101003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

101003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
101003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
101003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
101003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
101003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
101003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
101003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
101003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

101003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
101003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
101003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
101003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
101003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
101003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
101003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
101003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
101003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
101003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

101003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
101003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

101003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
101003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.