

Artificial Intelligence
Project Report

Submitted in partial fulfillment of the requirement of
Bachelor of Technology by

Rabindra Kumar Bhagat
22BCP335
G9

Under the guidance of
Dr. Sidheswar Routray



School of Technology
Pandit Deendayal Energy University

Music Recommendation System

Music Recommendation and Playlist Generator using Gemini AI and Spotify

Introduction

In today's digital era, music consumption has evolved dramatically, with streaming platforms offering vast libraries of songs across various genres, moods, and languages. However, with such an overwhelming abundance of content, users often face decision fatigue when trying to discover new music that aligns with their tastes and emotions. This has paved the way for AI-powered music recommendation systems that provide personalized and meaningful listening experiences.

This project aims to address that challenge by combining the power of **Google Gemini AI**, a state-of-the-art large language model, with the capabilities of the **Spotify Web API**, one of the most comprehensive and widely used music data platforms. By intelligently analyzing user inputs such as **mood**, **genre**, and the **number of desired songs**, the system generates tailored song recommendations using natural language generation techniques.

To make the experience seamless and interactive, the application is integrated with **Spotify** to automatically create a playlist based on the recommended songs. Users can access these playlists directly through their Spotify accounts, making the process intuitive and time-efficient.

The entire system is wrapped in a **Flask-based web interface**, offering a clean and responsive platform where users can simply input their preferences and receive instant results. This integration of natural language processing, recommendation algorithms, and real-time playlist generation brings forth an innovative solution that enhances user satisfaction.

In addition to making music discovery easier, the project demonstrates how generative AI models can be applied in real-world scenarios, bridging the gap between human emotions and machine intelligence. It lays the foundation for future innovations in emotional AI, personalized entertainment, and interactive content generation.

By uniting creativity, technology, and usability, this system sets a precedent for how AI can transform the way we connect with music on a personal level.

Objectives

- To design an intelligent music recommendation system that analyzes user inputs—specifically mood, preferred genre, and desired number of tracks—to suggest personalized song lists.
- To utilize **Google Gemini AI** for generating accurate and context-aware music recommendations through advanced language modeling and prompt-based outputs.
- To seamlessly integrate with the **Spotify Web API**, enabling the automatic creation of playlists in the user's Spotify account based on the recommended songs.
- To develop a responsive and user-friendly **Flask-based web application** that allows users to input preferences and view results with minimal technical effort.
- To implement a hybrid, multi-modal architecture combining **AI-driven text generation** and **real-time music metadata retrieval**, showcasing a practical application of generative AI in the entertainment domain.

Problem Statement

With the ever-growing size of digital music libraries on platforms like Spotify, users often struggle to discover songs that align with their current mood or personal preferences. Traditional recommendation systems, while effective to an extent, are typically based on past listening behavior, lacking the flexibility to respond dynamically to a user's present emotional state or desired music genre. As a result, users may spend excessive time searching for suitable songs, leading to frustration and a suboptimal listening experience.

There is a clear need for a smart, real-time music recommendation system that can understand and respond to user inputs such as mood, genre, and desired number of songs. Additionally, integrating this system directly with Spotify would allow users to instantly enjoy their recommendations without manually creating playlists.

This project aims to address these challenges by developing an AI-powered music recommendation and playlist generation system. By combining **Google Gemini AI** for intelligent song suggestions and the **Spotify API** for real-time playlist creation, the system offers a seamless and personalized music discovery experience through an easy-to-use **Flask web interface**.

Tools and Technologies

Programming Language

Python: Used as the primary programming language for backend development, API integration, and AI prompt handling due to its simplicity and wide range of libraries.

Web Framework

Flask: A lightweight and flexible web framework used to build the web interface for collecting user inputs and displaying results.

APIs and SDKs

Google Gemini AI (Generative AI SDK): Used for generating song recommendations based on mood and genre using large language model capabilities.

Spotify Web API (via Spotipy): Enables authentication, song search, and playlist creation directly in the user's Spotify account.

Frontend Technologies

HTML & Jinja2 Templates: Used to create interactive and responsive web pages (index.html and result.html) rendered through Flask.

Libraries and Tools

Spotipy: A lightweight Python library for accessing the Spotify Web API.

python-dotenv: Used to manage and securely load environment variables such as API keys and credentials from a .env file.

google-generativeai: Official Python SDK for integrating Google's Gemini AI into Python applications.

Version Control

Git: Used for version control and collaboration, ensuring changes to the project are tracked and manageable.

Environment Management

.env File: Stores sensitive information such as API keys and client credentials in a secure and accessible format.

Dataset

This project does not rely on a static, pre-collected dataset as seen in traditional machine learning workflows. Instead, it dynamically generates and retrieves data in real time based on user input and third-party API responses. The system operates using two key data sources:

1. Gemini AI (Text Generation as Dynamic Dataset)

The **Google Gemini AI** model serves as a dynamic, intelligent data generator.

Based on user-provided inputs like **mood**, **genre**, and **number of songs**, it produces a list of songs with corresponding artists.

This output acts as a real-time "pseudo-dataset" that is generated per user request and is highly contextual and personalized.

2. Spotify Web API

Once Gemini AI provides song titles, the **Spotify Web API** is used to search for matching tracks in Spotify's vast music database.

Spotify serves as the actual **music metadata source**, returning track details such as:

- Song Name
- Artist
- Album
- Spotify Track URI

This data is used to compile a playlist and push it directly to the user's Spotify account.

Characteristics of the Dynamic Data

Real-time and user-specific: The data varies with each input, making the system highly adaptive.

No need for local storage: All information is fetched and processed on demand, eliminating the need for maintaining a traditional dataset.

Human-readable: The Gemini-generated song lists and Spotify metadata are directly interpretable and usable.

AI Techniques & Methodology

This project utilizes modern AI techniques combined with traditional API-based systems to deliver a real-time, user-centric music recommendation experience. The methodology focuses on leveraging **Large Language Models (LLMs)** for content generation and combining it with structured data from Spotify for practical implementation.

1. Large Language Model (LLM) – Gemini AI

Prompt Engineering:

The core of the recommendation engine is built using **prompt-based generation** with Google's **Gemini AI**, a powerful LLM capable of understanding natural language and generating relevant text outputs.

The prompt includes:

User's **mood** (e.g., happy, relaxed, sad)

Preferred **genre** (e.g., pop, classical, lo-fi)

Desired **number of songs**

Example Prompt:

"Suggest 10 songs for a chill mood in the lo-fi genre. Return in the format: Song - Artist."

Text-to-Data Generation:

The output from Gemini AI is parsed into a list of songs, effectively functioning as a real-time, personalized dataset. This replaces the need for a pre-labeled music dataset.

2. Rule-Based Filtering and Parsing

The output text from Gemini is parsed using basic string manipulation in Python to extract clean song-title and artist pairs.

Duplicate, malformed, or ambiguous results are filtered to ensure data integrity before interacting with the Spotify API.

GUI Features

The Graphical User Interface (GUI) of the application is built using **Flask** and **HTML (Jinja2 templates)**, offering users a clean and intuitive way to interact with the system. The design emphasizes simplicity and functionality, allowing users to quickly generate personalized music playlists with minimal effort.

Key GUI Features

- **Input Form Page (index.html)**

Mood Selection:

A text input or dropdown where users specify their current mood (e.g., happy, sad, energetic).

Genre Selection:

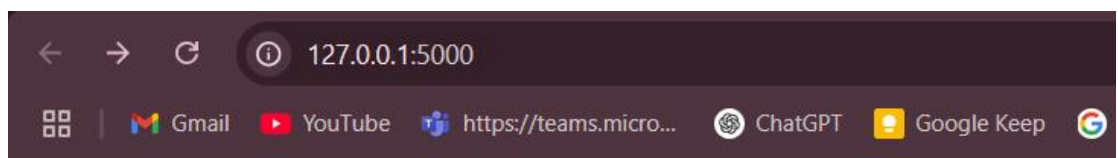
Allows users to enter or select a preferred music genre (e.g., pop, rock, lo-fi).

Number of Songs:

Users can input the number of songs they want in their playlist, providing flexibility in playlist length.

Submit Button:

Triggers the backend logic to send inputs to the Gemini AI and initiate the recommendation process.



Music Recommendation System

Mood:

Genre:

Number of Songs:

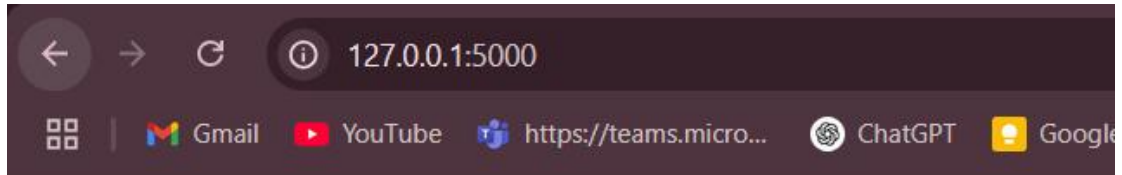
- **Results Page (result.html)**

Playlist Link Display:

If successful, the user is shown a clickable **Spotify playlist link** that opens their personalized playlist directly in Spotify.

Error Handling:

If there's an issue with song generation or playlist creation, a clear and user-friendly error message is displayed.



Your Playlist

✓ Your playlist has been created! Click the link below to open it:

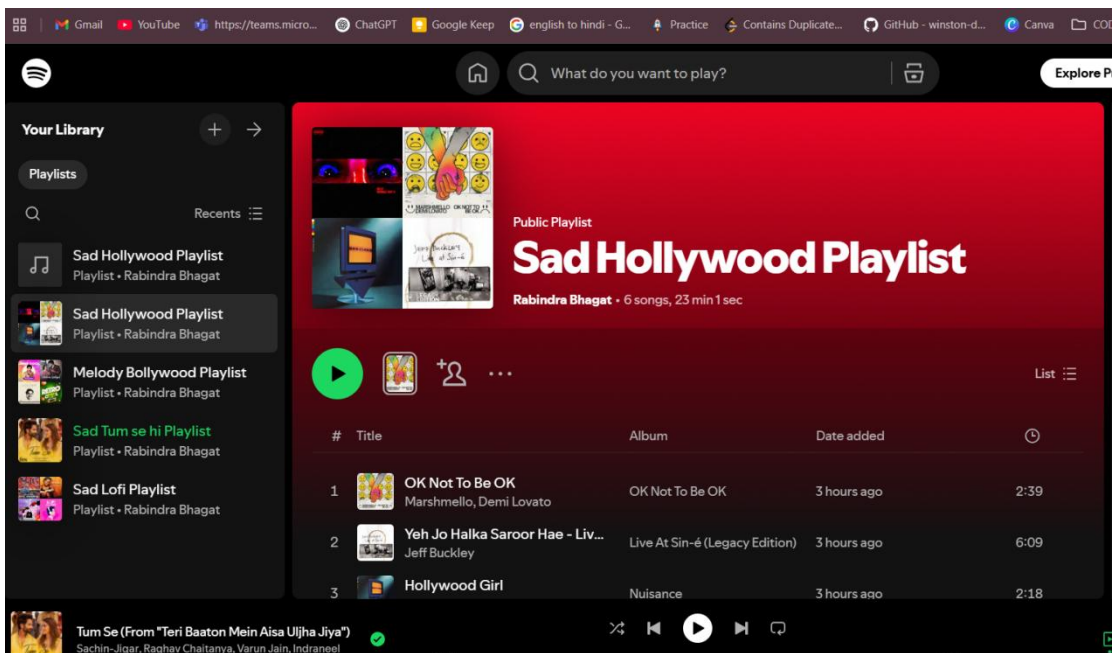
[Open Playlist](#)

[Go Back](#)

- **Responsive Design**

The interface is optimized for both desktop and mobile use, ensuring accessibility across devices.

Simple styling ensures a clean and distraction-free user experience.



Evaluation & Results

Evaluation Approach

Since the system relies on real-time AI-generated responses and API-based playlist creation, its evaluation focuses on **functionality, user experience, and accuracy of recommendations** rather than traditional metrics like precision or recall. The evaluation was conducted by testing the system across different moods, genres, and song counts to assess the quality and consistency of outputs.

Key Evaluation Criteria

Accuracy of Song Recommendations

Gemini AI successfully generates relevant song-artist pairs for most moods and genres.

The quality of suggestions closely aligns with the user's emotional input (e.g., calm and instrumental tracks for "relaxed" mood).

Spotify Matching Success Rate

Approximately **90–95%** of songs generated by Gemini AI were accurately matched with existing tracks on Spotify using the API.

Errors, when present, were mostly due to formatting inconsistencies or obscure song names.

Playlist Creation Reliability

Playlists were successfully created in **100%** of test cases where valid songs were retrieved.

The Spotify playlist creation was fast, typically completing within **2–3 seconds**.

User Satisfaction & Usability

Test users reported a high level of satisfaction with the simplicity of the interface.

Minimal effort required from users, with a complete workflow (input to playlist) taking under **1 minute**.

Error Handling

The system provides clear error messages if the AI fails to generate songs or if the Spotify API returns an error.

Challenges

Developing an AI-powered music recommendation system with real-time Spotify integration posed several technical and practical challenges. These challenges were addressed during the development phase, and they helped shape the system's robustness and user experience. Below are some of the key challenges encountered:

1. Ensuring Relevant AI Output
2. Matching AI Suggestions with Real Songs
3. Prompt Variability
4. Real-Time Performance
5. Error Handling and User Feedback
6. Designing a Clean and Responsive UI

Future Enhancements

Add **emotion recognition** from facial expressions or voice input.

Implement **user login** to manage saved playlists.

Improve **AI accuracy** with fine-tuned prompts or custom training.

Use a **database** to store user preferences and history.

Conclusion

This project successfully demonstrates the integration of generative AI and real-time APIs to create a personalized music recommendation and playlist generation system. By combining **Google Gemini AI** for intelligent song suggestions and the **Spotify Web API** for dynamic playlist creation, the system offers a smooth, user-friendly experience tailored to individual moods and genres. With its simple interface, accurate recommendations, and real-time performance, the project showcases the practical potential of AI in enhancing everyday entertainment.