BIDM PROJECT

PRESENTED BY: GROUP 3

Akriti Sharma
EMBADTA24014

Akanksha Singh
EMBADTA24006

Gunjan Kapoor
EMBADTA24003

Introduction

GITHub link: https://github.com/SAkriti03/MoodifyHome/tree/main

Streamlit link: https://moodifyhome-kt8bkcpkdm6boemeeyuymz.streamlit.app/

Moodify is a web-based music discovery and streaming platform designed to provide a personalized user experience by leveraging mood-based recommendations, smart playlist generation, and curated multimedia content. Built using Streamlit, a Python-based framework for building interactive web applications, Moodify integrates data-driven algorithms with a user-friendly interface to deliver a seamless music exploration experience. The platform is structured around three core sections: Home, Music Hub, and Podcast, each offering distinct functionalities tailored to enhance user engagement and satisfaction.

This report provides a comprehensive analysis of Moodify's functional capabilities, technical architecture, and potential future enhancements. The discussion is structured to highlight the platform's design philosophy, technical implementation, and performance optimization strategies, while also exploring avenues for future development.

1. Functional Overview

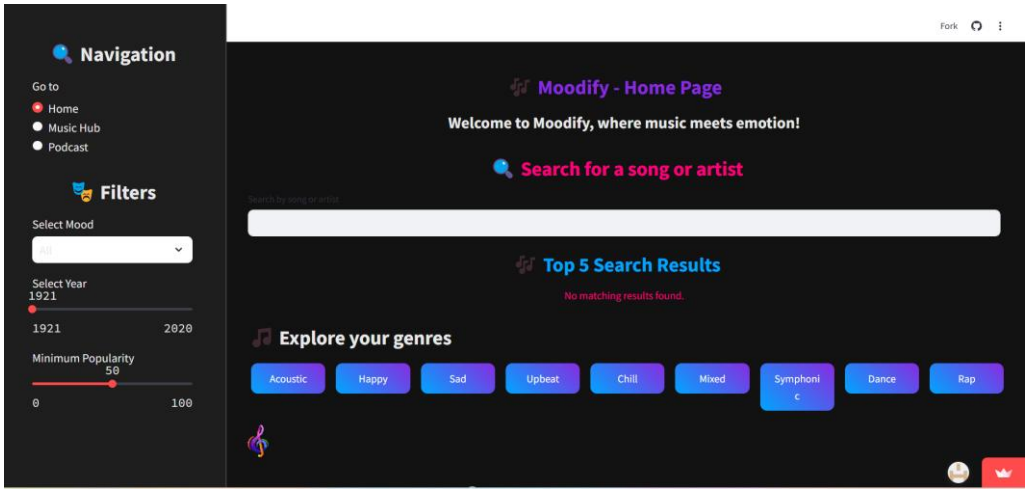
Moodify's design is centered around three core sections: Home, Music Hub, and Podcast. Each section is designed to serve a specific purpose in delivering a tailored music exploration experience. The platform's functionality is driven by user interaction, data analysis, and dynamic content rendering, ensuring a responsive and engaging user experience.

1.1 Navigation and User Interface

The navigation system in Moodify is facilitated through a sidebar interface, which allows users to seamlessly switch between the Home Page, Music Hub, and Podcast Section. The interface employs a dark theme with neon accents, ensuring readability and a visually engaging experience. The use of custom CSS for styling ensures that the platform maintains a consistent aesthetic across all sections, while also providing intuitive user interactions.

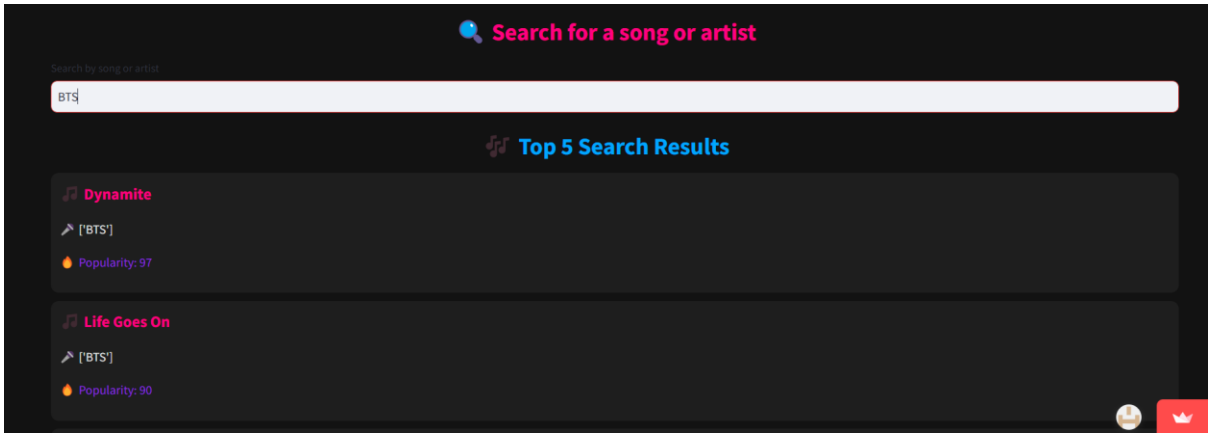
The sidebar navigation is implemented using Streamlit's radio button component, which dynamically updates the session state to reflect the user's current page. This ensures that the platform remains responsive and user-friendly, even as users navigate between different sections.

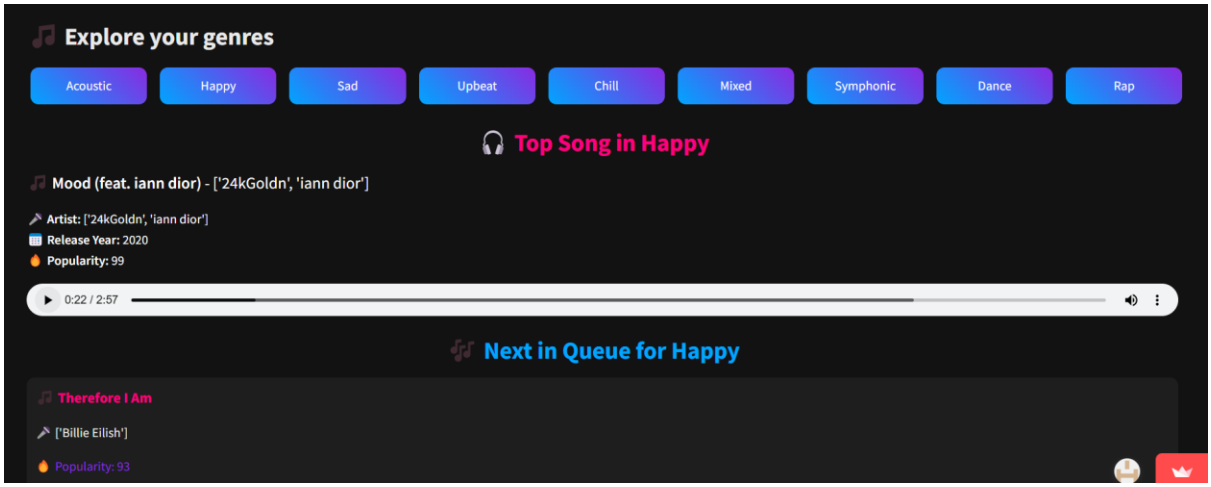
1.2 Home Page: Search and Discovery

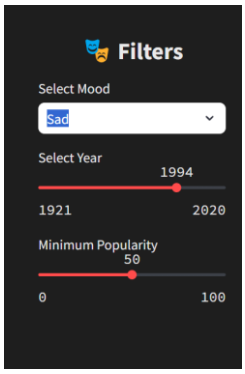


The Home Page serves as the central hub for discovering new music. It includes a search bar that allows users to look up songs or artists by name. The search functionality is powered by Pandas, which filters the dataset (data1.csv) based on the user's query and retrieves the top five most relevant and popular results. This ensures that users are presented with high-quality recommendations that align with their search intent.

In addition to the search functionality, the Home Page incorporates several dynamic features designed to enhance user engagement:

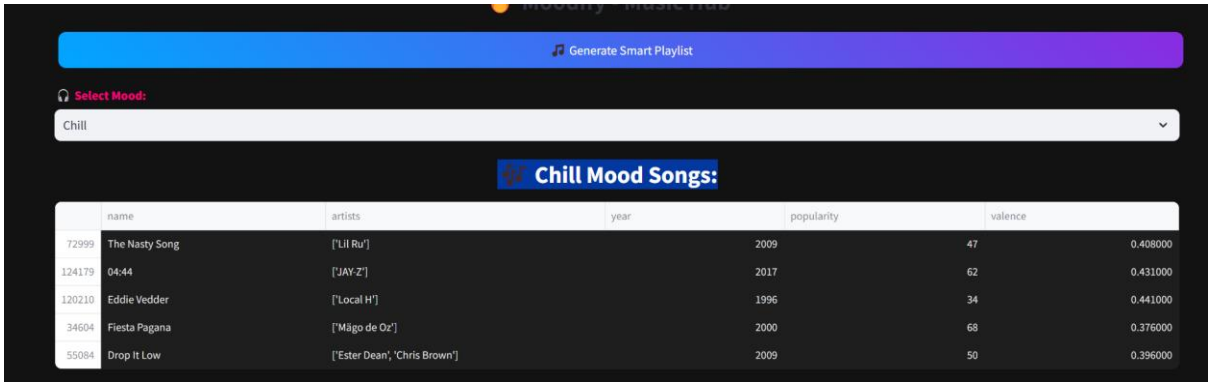






- **Mood-based filters**: Users can filter songs based on their current mood, with options such as **Happy, Sad, Energetic, and Calm**. These filters are implemented using **Pandas' data filtering capabilities**, which analyze song attributes such as **valence** and **energy** to provide mood-aligned recommendations.
- **Genre-based selection**: The platform allows users to explore different music genres, with **buttons** for each genre dynamically generated based on the dataset. When a genre is selected, the platform displays the **top song** in that genre, along with a **queue of recommended tracks**.
- **Dynamic recommendation engine**: The Home Page also features a **trending songs section**, which displays the **most popular tracks** based on the dataset's popularity metric. This ensures that users are always aware of the **latest and most popular music**.

1.3 Music Hub: Smart Playlists and Mood-Based Recommendations

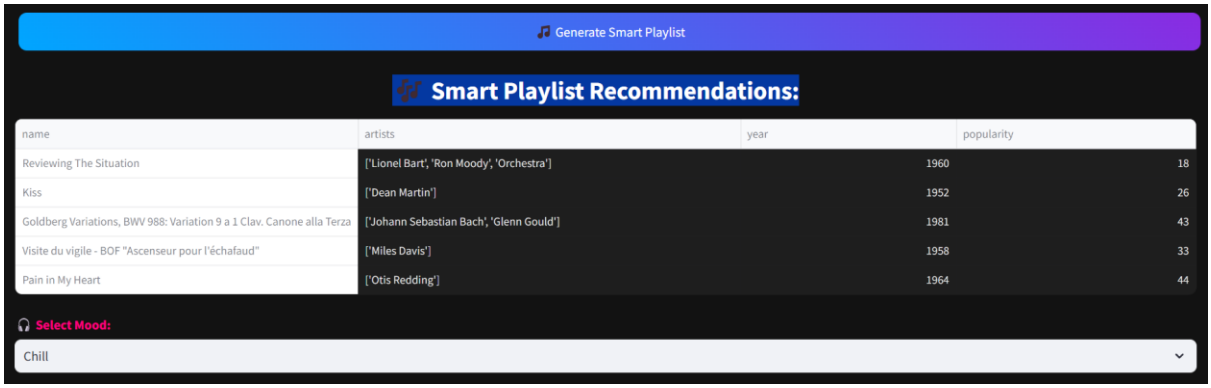


The Music Hub is the core functional component of Moodify, offering a smart playlist generation system that curates a selection of five songs based on historical preferences and listening patterns. This feature is designed to provide users with a personalized listening experience, ensuring that the recommendations align with their musical tastes.

A key feature of the Music Hub is the mood-based recommendation system, which suggests songs by analyzing attributes such as valence (happiness) and energy levels. The platform uses a mood mapping dictionary to associate specific mood ranges with song attributes, ensuring that the recommendations are emotionally aligned with the user's current state. Over time, the platform builds a mood trend graph, which visually represents the user's changing preferences and emotional engagement with music. This graph is generated using Matplotlib, providing users with a clear and intuitive visualization of their listening habits.

Additional functionalities in the Music Hub include:

- **Trending songs**: A section that displays the **most popular songs** based on the dataset's popularity metric.
- **Featured artists**: A section that showcases **prominent musicians** based on their frequency in the dataset.
- **User-created playlists**: Users can create and manage their own playlists by adding songs manually. This feature is implemented using **Streamlit's session state**, which allows the platform to persist user data across sessions.

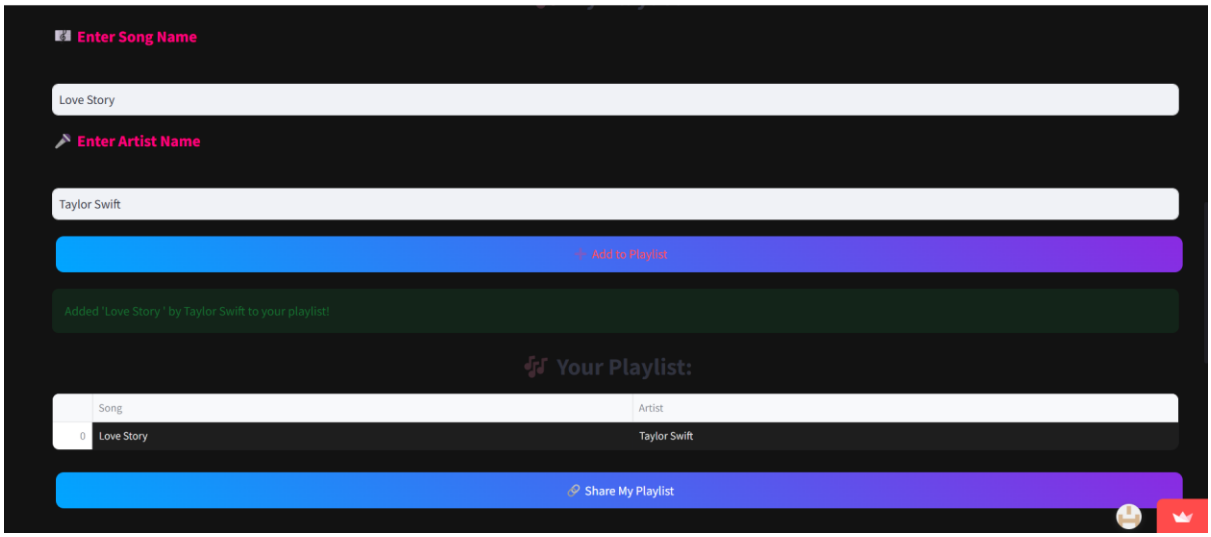


Trending Songs

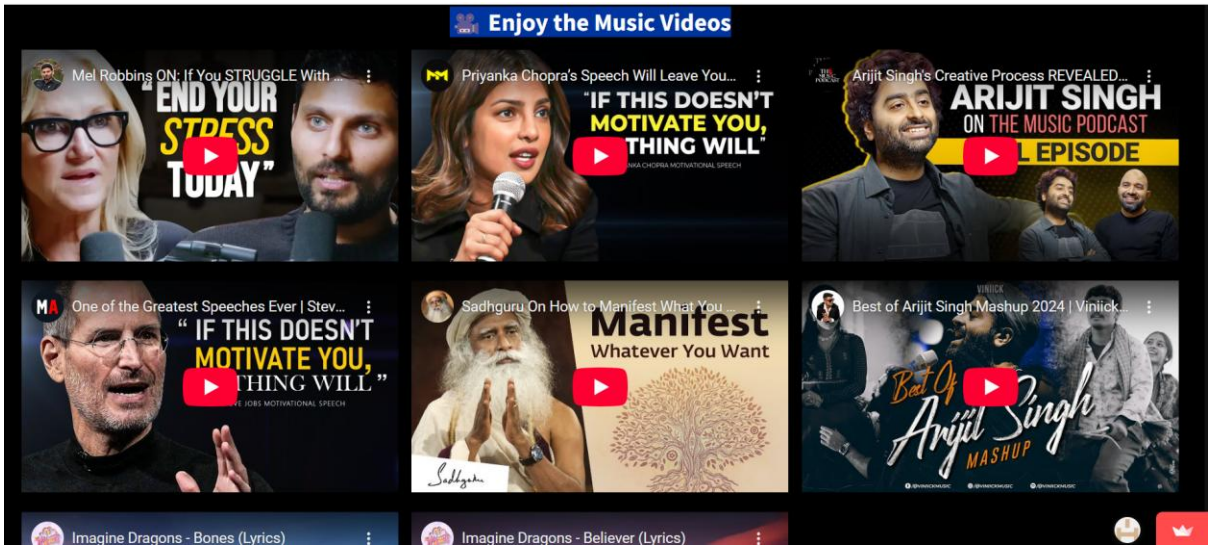
	name	artists	year	popularity
19611	Dakiti	['Bad Bunny', 'Jhay Cortez']	2020	100
19606	Mood (feat. iann dior)	['24kGoldn', 'iann dior']	2020	99
19618	Dynamite	['BTS']	2020	97
19608	WAP (feat. Megan Thee Stallion)	['Cardi B', 'Megan Thee Stallion']	2020	96
19612	What You Know Bout Love	['Pop Smoke']	2020	96
19610	positions	['Ariana Grande']	2020	96
19616	Blinding Lights	['The Weeknd']	2020	96
19615	Holy (feat. Chance The Rapper)	['Justin Bieber', 'Chance the Rapper']	2020	95
19620	Lonely (with benny blanco)	['Justin Bieber', 'benny blanco']	2020	95
19607	For The Night (feat. Lil Baby & DaBaby)	['Pop Smoke', 'Lil Baby', 'DaBaby']	2020	95

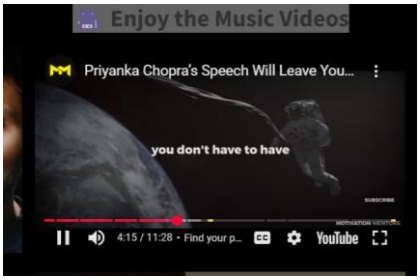
Featured Artists

artists	count
['Эрнест Хемингуэй']	1089
['Эрих Мария Ремарк']	953
['Francisco Canaro']	927
['Ignacio Corsini']	613
['Frank Sinatra']	572



1.4 Podcast Section: Visual and Audio Content





Beyond music recommendations, Moodify offers a podcast-style experience through curated YouTube music videos. This section integrates embedded video streaming using Streamlit's video component, allowing users to enjoy a combination of audio and visual content. The videos are displayed in a grid layout, with three videos per row, ensuring that the interface remains organized and visually appealing.

The Podcast Section is designed to expand Moodify's entertainment offerings, providing users with a diverse range of content that goes beyond traditional music streaming. This section is particularly appealing to users who enjoy music videos and live performances, offering a multimedia experience that complements the platform's core music discovery features.

2. Technical Overview

Moodify's architecture is structured around a Python-based backend using Streamlit for UI development, Pandas for data handling, and Matplotlib for visualization. The system is designed to be efficient, responsive, and scalable, with caching mechanisms in place to improve performance.

2.1 Tech Stack and Frameworks

- **Frontend**: Streamlit (UI framework).
- **Backend**: Python (handling data logic and dynamic updates).
- **Data Management**: Pandas (for filtering and recommendations).
- **Styling**: Custom CSS (for theme customization).
- **Caching**: @st.cache_data (to optimize dataset loading).
- **Visualization**: Matplotlib (for mood-tracking trends).

2.2 Core Modules and Their Roles

1. **Main Script (mainV1.0.py)**
    - The main script serves as the entry point for the application, handling navigation logic and dynamic content rendering. It is responsible for loading the dataset (data1.csv) and ensuring real-time updates based on user interactions.
    - The script implements search functionality using Pandas' string operations, allowing users to search for songs or artists by name. It also includes filtering logic based on mood, year, and popularity, ensuring that users can refine their search results to align with their preferences.
    - The session state is used to persist user data across sessions, ensuring that the platform remains responsive and user-friendly.
2. **Music Hub (musichub.py)**
    - The Music Hub module is responsible for playlist generation and mood-based filtering. It uses Pandas' data manipulation capabilities to analyze song attributes and generate recommendations based on the user's mood.
    - The module also implements trending song retrieval and featured artist selection, ensuring that users are always aware of the latest and most popular music.
    - The mood trend graph is generated using Matplotlib, providing users with a visual representation of their listening habits over time.
3. **Podcast Section (podcast.py)**
    - The Podcast Section provides embedded video playback for music-related content. It uses Streamlit's video component to display YouTube videos in a grid layout, ensuring that the interface remains organized and visually appealing.

- The section is styled using custom CSS, ensuring that it maintains a consistent aesthetic with the rest of the platform.

3. Design and User Experience

Moodify's design philosophy is focused on usability, responsiveness, and aesthetic consistency. The dark mode interface is chosen to reduce eye strain, while neon highlights create a modern and visually appealing experience. The UI elements are structured for clarity, ensuring that users can interact with the platform effortlessly.

Key design elements include:

- Dark Background (#121212) for reduced visual fatigue.
- Text Highlights in Blue (#00A6FF), Pink (#FF007F), and Purple (#8A2BE2) for improved readability.
- Dynamic hover effects for interactive buttons, enhancing user engagement.

4. Performance Optimization

Efficiency is a critical aspect of Moodify's development. To ensure fast response times and a smooth user experience, several optimization techniques have been implemented:

1. **Session Management**
    - st.session_state is used to persist user data, avoiding redundant computations.
2. **Dataset Caching**
    - @st.cache_data reduces the need for repetitive dataset loading, significantly improving performance.
3. **Efficient Query Execution**
    - All song searches and recommendations are handled using optimized Pandas operations, minimizing latency.
4. **Lazy Data Loading**
    - Large datasets are processed on-demand, ensuring that only necessary data is loaded at any given time.

5. Future Enhancements

Moodify's foundation is robust, but there are several enhancements that could significantly improve its capabilities:

1. **Machine Learning-Based Recommendations**
    - Integrating an AI-powered recommendation engine could enhance song suggestions by learning from user behavior over time.

2. **User Authentication and Cloud Storage**
    - Adding Google or social media login options would allow users to store playlists persistently in a cloud database, instead of session-based storage.
3. **Spotify or Apple Music API Integration**
    - Direct integration with streaming services would enable real-time music playback, rather than relying on pre-loaded datasets.
4. **Expanded Podcast Section**
    - A dedicated podcast streaming feature with support for Spotify Podcast API would diversify content offerings.

6. Conclusion

Moodify presents a well-structured, user-centric platform for music discovery, playlist management, and multimedia content streaming. Its data-driven recommendations, mood-based filtering, and dynamic UI design make it a compelling alternative to conventional music platforms.

The technical foundation is scalable and efficient, leveraging Streamlit's capabilities, caching techniques, and session-based storage for a seamless experience. While Moodify already provides a diverse feature set, its future iterations have the potential to further personalize the listening experience through AI, cloud storage, and streaming service integration.

As it continues to evolve, Moodify has the potential to redefine how users interact with music and digital content in a data-driven, emotionally intelligent way.

Hope you like our project! 😊