

Mood-Based Music Player

Project Report

Submitted By:

Shrenika Rajpoot

Reg No: 25BAI11384

B.Tech CSE (AI & ML) – 1st Year

VIT Bhopal University

Submitted To:

Shristika Raikwar

Course: Introduction to Programming & Problem Solving

Abstract

This project presents a Python-based Mood-Based Music Player capable of selecting and playing music according to the user's emotional state. The program uses simple text-based input, a curated playlist, ASCII animation, and smart random selection to

provide an engaging user experience. The project demonstrates Python fundamentals such as functions, loops, modules, dictionaries, randomization, and user interaction.

Introduction

Music has a strong emotional connection with humans. People choose songs based on their mood—sad, happy, romantic, party, study, or motivational. This project automates this selection process by allowing users to input their mood, after which the program plays a song best suited for that state. The project also includes engaging animations and a music visualizer for a creative touch.

Objectives

- Create an interactive Python program.
- Implement mood-based music selection.
- Use animations for better user experience.
- Demonstrate understanding of Python modules and functions.
- Build a simple but unique project suitable for first-year coursework.

Requirements

- Python 3.8 or above
- Web browser module (built-in)
- Random module (built-in)
- Time module (built-in)
- Sys module (built-in)
- Iter tools module (built-in)
- Internet connection to play YouTube links

Methodology

1. User selects a mood from predefined categories.
2. The program validates the mood.
3. A loading animation runs to simulate processing.
4. A random song is selected from a playlist mapped to the mood.
5. ASCII visualizer runs to create engagement.
6. The selected YouTube link opens automatically in the browser.
7. User can continue or exit the program.

Key Features

- Smart mood recognition
- Playlist-based song selection
- Rotating loading animation
- ASCII music visualizer
- YouTube integration
- Simple and clean UI
- Beginner-friendly Python implementation

Source Code

The following Python code implements the complete project:

```
import time
```

```
import random
```

```
import web browser
```

```
import sys
```

```
import iter tools
```

```
# -----
```

```

# TITLE SCREEN

# -----

def title_screen():
    print("\n" + "="*60)
    print("      MOOD-BASED MUSIC PLAYER ")
    print("      by Shrenika")
    print("="*60)
    time.sleep(1)

# -----

# LOADING ANIMATION

# -----

def loading(msg="Analyzing your mood"):
    for c in itertools.cycle(['|', '/', '-', '\\']):
        sys.stdout.write(f'\r{msg} ' + c)
        sys.stdout.flush()
        time.sleep(0.2)
        if random.random() > 0.93:
            break
    print("\n")

# -----

# MUSIC DATABASE

```

music_library = {

 "happy": [

 "https://youtu.be/OPf0YbXqDm0?si=UwbrKIMUvan2Sz-b",
 "https://youtu.be/QGJuMBdaqlw?si=XVpUbgwKPaV1lqb3",
 "https://youtu.be/dCmp56tSSmA?si=1MRCzzW4wCfyysTY",
 "https://youtu.be/YR12Z8f1Dh8?si=k09jKnzUPI66__wt",
 "https://youtu.be/9a4izd3Rvdw?si=bXSS_2UE9HZOFQfA",
 "https://youtu.be/wp43OdtAAkM?si=W-_tlSS3Z4CuzrF2",
 "https://youtu.be/q_b8tcbRhXE?si=usr-lv4o2BaX5whM",
 "https://youtu.be/QrOe2h9RtWI?si=AdYonVYJzFFZNgRW",
 "https://youtu.be/ApXoWvfEYVU?si=WyTcze_ehwocFsZm",
 "https://youtu.be/wnJ6LuUFpMo?si=JefLIQAtixLzyi3o",
 "https://youtu.be/tA8h_exda3E?si=Miul929zxYRVF-tS",
 "https://youtu.be/WKbwopSXLWU?si=wpIO5IjMnji_AkDO",
 "https://youtu.be/w5tWYmIOWGk?si=bG8lUsYE0g6X8bcf",
 "https://youtu.be/OBgOwAf-oVI?si=Kixc6E9prwmzh427",
 "https://youtu.be/1nWQs6lxTrY?si=arJ-OLghbiVnzJzj"

],

 "sad": [

 "https://youtu.be/pGuc4hPhiKw?si=hPxrURfdNzrV66Hb",
 "https://youtu.be/e5LSHAE03A?si=Qsl4WN5bAQiEfLTj",
 "https://youtu.be/9T-Zbxg9X_4?si=koGxjpbFmuFj0Asf",

"https://youtu.be/LMnJp_dSdnw?si=WvmZ5HXOwEhiqdlB",
"https://youtu.be/2AO2Utn2eoU?si=ddhNRCtT7Tc1LAVS",
"https://youtu.be/NRHIBKNNzAk?si=e0o91taXfSYCcLMd",
"https://youtu.be/YLkdm2pe8hA?si=8UnMb6cBnF_z9yE6",
"https://youtu.be/__bHEfJ29j0?si=0alUvWghEUnJ3iVI",
"https://youtu.be/bzSTpdcs-El?si=ZD1-NfD9Rx5AwhRO",
"https://youtu.be/6d5SS0gS5bU?si=dzO9hdsrfyREUFSa",
"https://youtu.be/xdZiUuwZeOl?si=pcVKZ3RnqfmjH_2f"

],

"romantic": [

"https://youtu.be/1wEtmB3z_yc?si=s6UYFbD9gascdHsk",
"https://youtu.be/pkzOBl1p7y4?si=CXF9MoiFJAZ8joNM",
"https://youtu.be/qgDxKAqLVgU?si=QucURNW7yKEjWG1T",
"https://youtu.be/LSUR0075KLI?si=23OGJwMKil6-yFlp",
"https://youtu.be/mZQH8CPQ-wo?si=sSXbR5BSMNly_agL",
"https://youtu.be/SxTYjptEzZs?si=gthFv0B9VtD_JCH0",
"https://youtu.be/0KozfDYK1EU?si=lxAUVa1zL_dIU-kf",
"https://youtu.be/WnU0lH6C0EA?si=gXWM8TCRo995k_Go",
"https://youtu.be/CWfCp96-yck?si=l8NrGiWgQBm70LcS",
"https://youtu.be/vJQCAtzSfu0?si=tKpQCR60gj7jJg9E",
"https://youtu.be/GxldQ9eX2wo?si=zUwr_mlk8Twsu4k2",
"https://youtu.be/u7Jl-WDeUo4?si=346fLKMlBOsNk8DI"

],

"study":[

"https://youtu.be/lkkGlVWvkLk?si=B-jampLCU_0RgfTM",
"https://youtu.be/lTRiuFIWV54?si=z3Cnhpx1gmqTULUr",
"https://youtu.be/jXZAbnn1kTU?si=X2w39U0Ptvb8J5l1",
"https://youtu.be/WPni755-Krg?si=WNC5S7VfEGeFJdEF"
],

"motivational":[

"https://youtu.be/7wtfhZwyrcc?si=eFKGH-sw_3HypEJb",
"https://youtu.be/Ax0G_P2dSBw?si=3Tm5xcXqepRvhMPF",
"https://youtu.be/RCgbE6eS-DU?si=6XAuLWhupXxcRbhl",
"https://youtu.be/T94PHkuydcw?si=0gfnnnczUQK1j0w7",
"https://youtu.be/mk48xRzuNvA?si=0WSO2ToZllli7EmV",
"https://youtu.be/CevxZvSJLk8?si=3Utbifvseit7U6dr",
"https://youtu.be/btPJPFnesV4?si=gINA-VVIL0VEkVvG"

],

"party": [

"https://youtu.be/sONw3dihCRs?si=8HAGssdE7_5qtugT",
"https://youtu.be/lu8210k9WQc?si=ckk8Nvp4d8bFzoZp",
"https://youtu.be/4dsFQFCvVGU?si=xBqMHS9_xxlC_5dV",
"https://youtu.be/jXlNjZnnCgc?si=aRgb5wvac7PUacs_",
"https://youtu.be/T4tedh_11hg?si=UBsouJKWlGvVrajw",
"https://youtu.be/jCEdTq3j-0U?si=UcteEswx_7NTxztP",


```

    "https://youtu.be/LEYXdZ_rVbo?si=2EduuqTIFSYg-eII",
    "https://youtu.be/pElk1ShPrcE?si=7OzOGwEwudzP2s33",
    "https://youtu.be/t4H_Zoh7G5A?si=FR4r60kAmY1ffHBe",
    "https://youtu.be/DUT5rEU6pqM?si=ypYwUl9QRaryb7LN",
    "https://youtu.be/-5HZ8mgpsik?si=6vcHRubpSV3XtEaY"

]
}

```

```

# -----
# VISUALIZER
# -----

def music_visualizer():
    print("\n Initializing Music Visualizer...")

    bars = ["_", "▒", "▓", "█", "░", "▒", "▓", "█"]

    for _ in range(30):

        line = "".join(random.choice(bars) for i in
range(random.randint(10,20)))

        print(" " + line)

        time.sleep(0.1)

# -----
# MAIN MOOD FUNCTION

```

```
# -----  
  
def play_music_by_mood():  
    print("\nYour mood options:")  
    print(" → happy\n → sad\n → romantic\n → party\n → study\n → motivational\n")  
  
    mood = input("Tell me your mood: ").lower()  
  
    if mood not in music_library:  
        print("\n⚠ Oops! Mood not recognized. Try again.")  
        return  
  
    loading("Finding the best song for you")  
  
    selected_song = random.choice(music_library[mood])  
  
    print(f"\n Playing a {mood} vibe song for you... Enjoy!\n")  
    music_visualizer()  
    webbrowser.open(selected_song)  
  
# -----  
  
# PROGRAM LOOP  
  
# -----
```

```
title_screen()
```

```
while True:
```

```
    play_music_by_mood()
```

```
    again = input("\nWant to play another mood? (yes/no):  
").strip().lower()
```

```
    if again != "yes":
```

```
        print("\n Thanks for using Mood-Based Music Player. Stay  
Happy! \n")
```

```
        break
```

RESULTS

The Mood-Based Music Player successfully:

- Takes user input for mood selection
- Shows a realistic loading animation
- Displays a dynamic ASCII music visualize
- Opens a randomly selected YouTube song based on mood
- Allows repeated use through a program loop This demonstrates use of Python decision-making, loops, functions, animations, and external linking.

✓ The project executed successfully and met all requirements.

CONCLUSION

This project successfully demonstrates the implementation of a mood-based music system using Python. It showcases logical thinking, modular programming, data handling, animations, and user interaction – making it a strong first-year project that is unique, creative, and academically valuable.