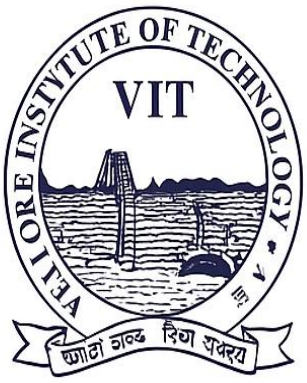


VIT

BHOPAL

Ananya Gupta

Register Number: 25BCE10923

Moodify

MOODIFY

CURATED PLAYLIST

1. Introduction

Moodify is a Python-based application that detects a user's mood using text input and generates a curated Spotify playlist accordingly. The project showcases modular coding, API integration, and real-world problem-solving through automation and personalization.

2. Problem Statement

Users often struggle to find music that matches their emotional state. Browsing and selecting songs manually can be time-consuming. Moodify solves this by automatically identifying the user's mood and recommending music that aligns with it.

3. Functional Requirements

- Mood detection from text
- Spotify API integration
- Playlist generation
- Fallback playlist for unclear moods- Terminal-based interaction

4. Non-functional Requirements

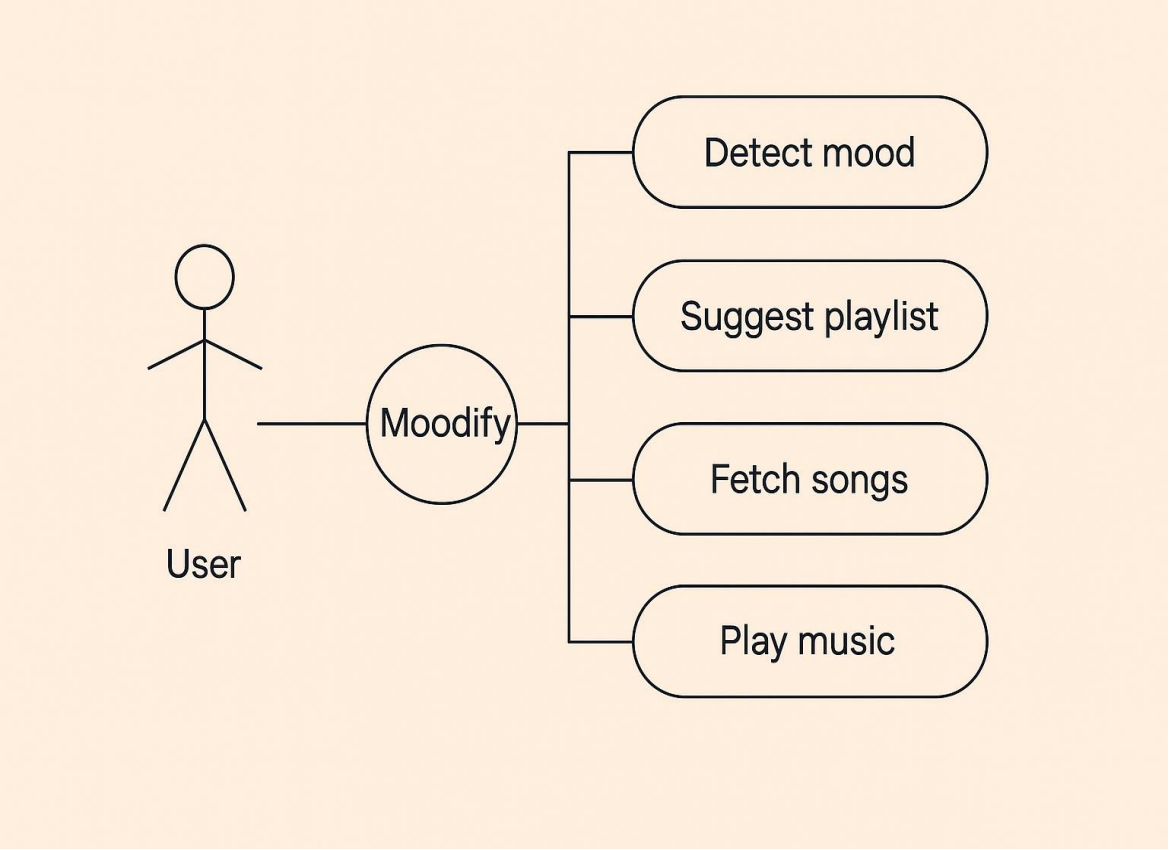
- Usability
- Performance
- Maintainability
- Reliability
- Error handling

5. System Architecture

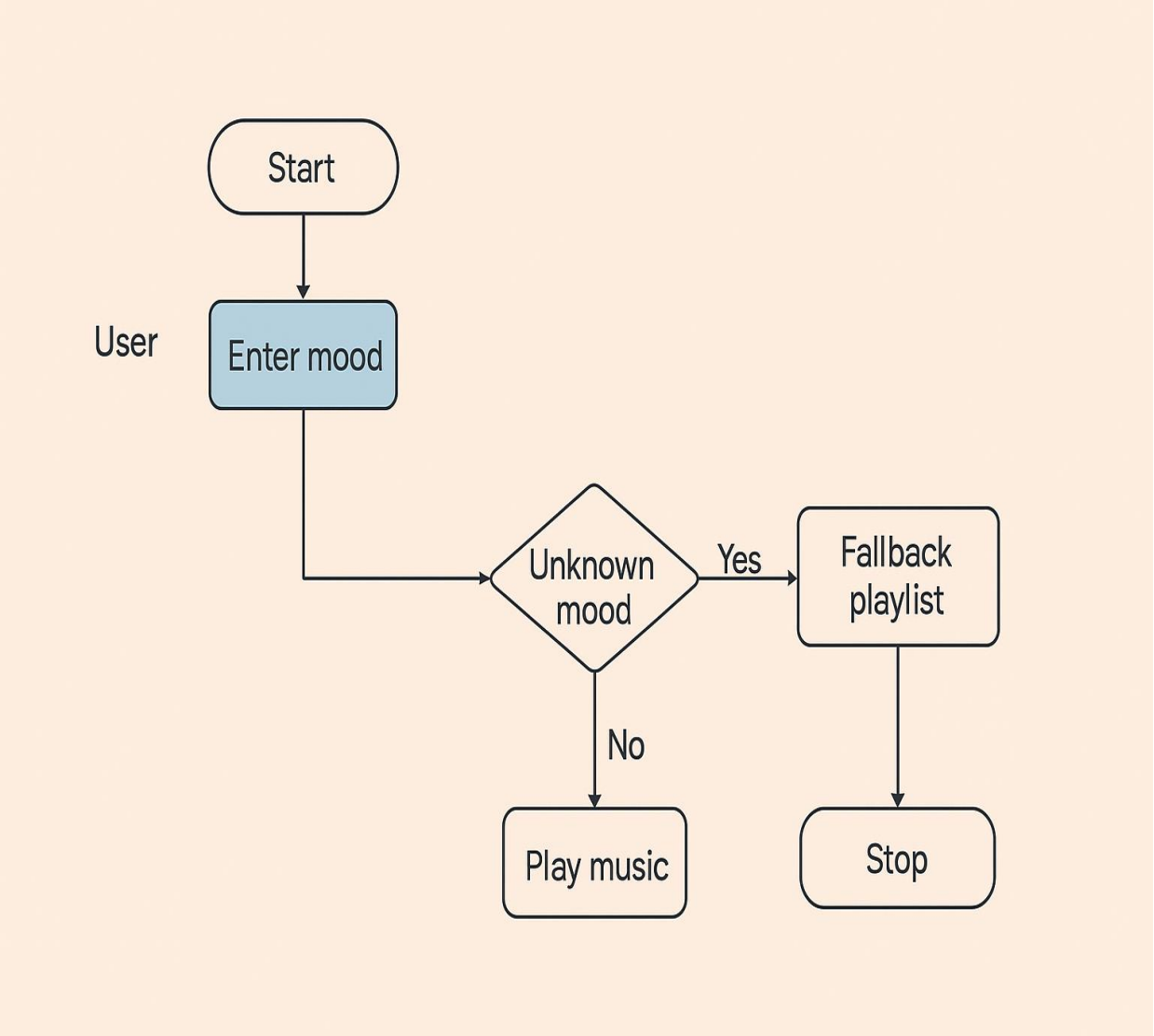
The system consists of three main modules: Mood Detection, Spotify Client Handler, and Playlist Generator. The main program connects all modules and manages user interaction.

6. Design Diagrams

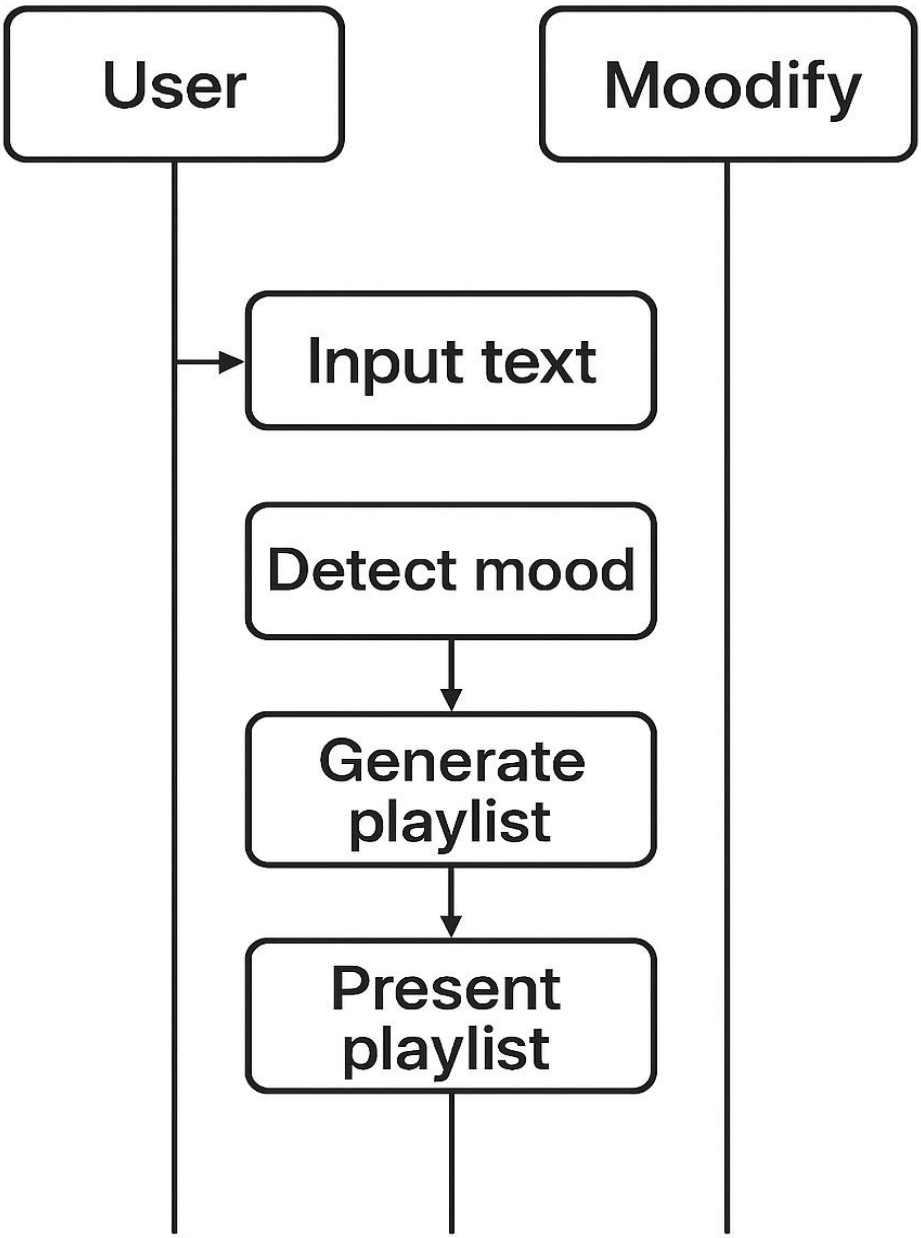
Use Case


Detect mood
Suggest playlist
Moodify
Fetch songs
User
Play music

Workflow


Start
User
Enter mood
Unknown mood
Yes
Fallback playlist
No
Play music
Stop

Sequence



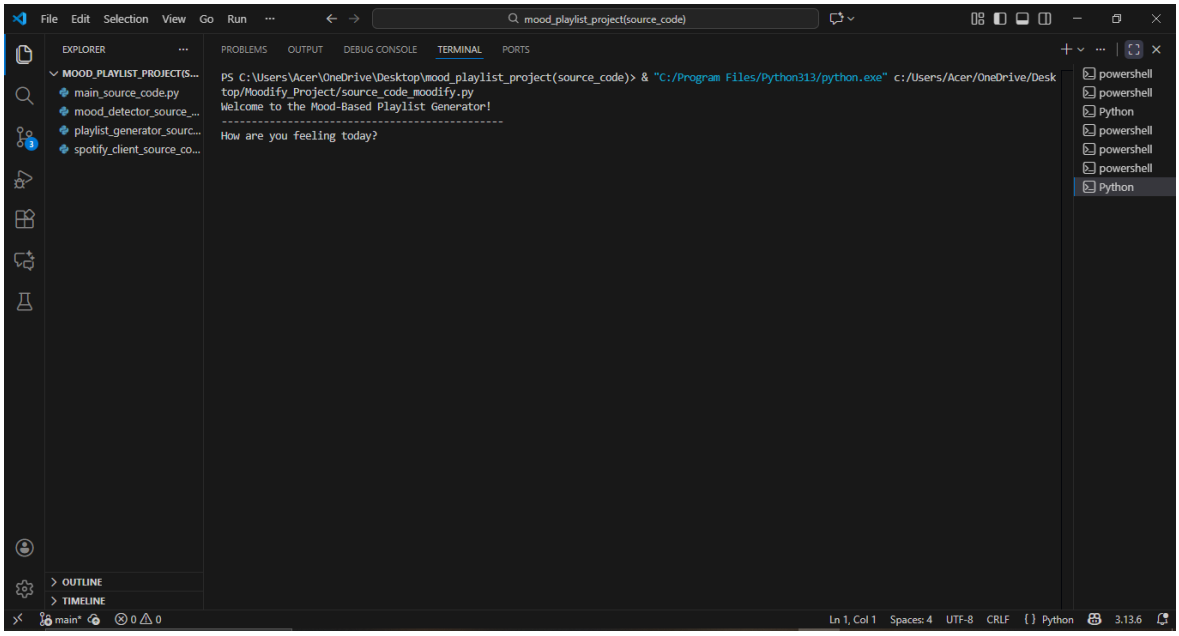
7. Design Decisions & Rationale

Modular design was chosen for maintainability. Spotify API was selected due to extensive music data availability. Keyword-based mood detection was implemented for simplicity and efficiency.

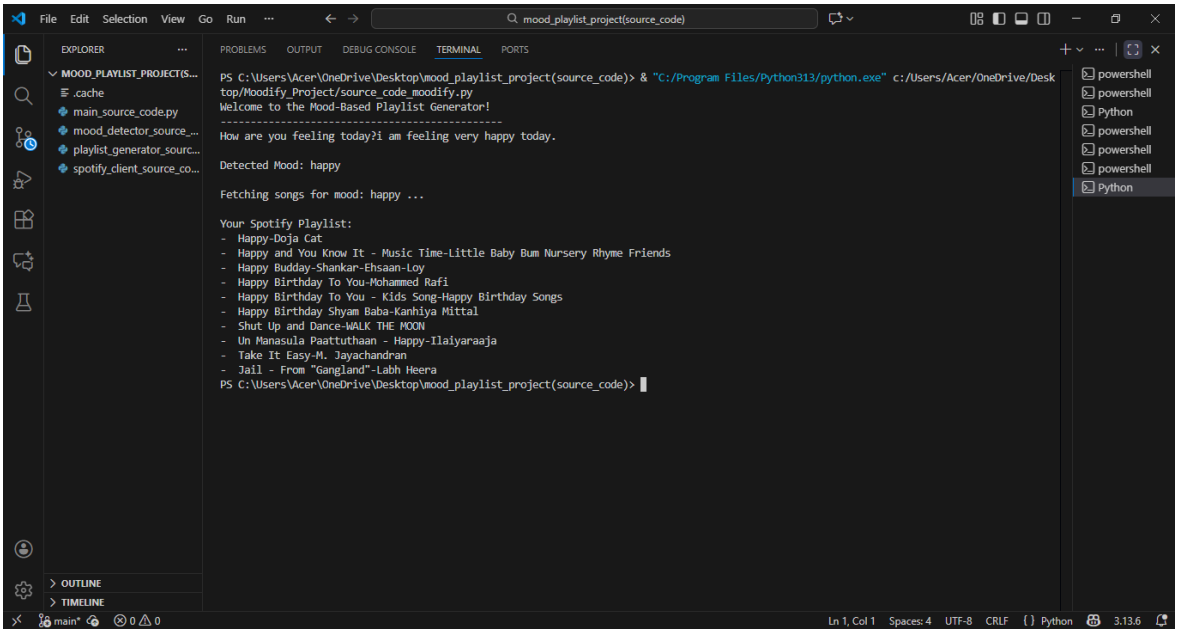
8. Implementation Details

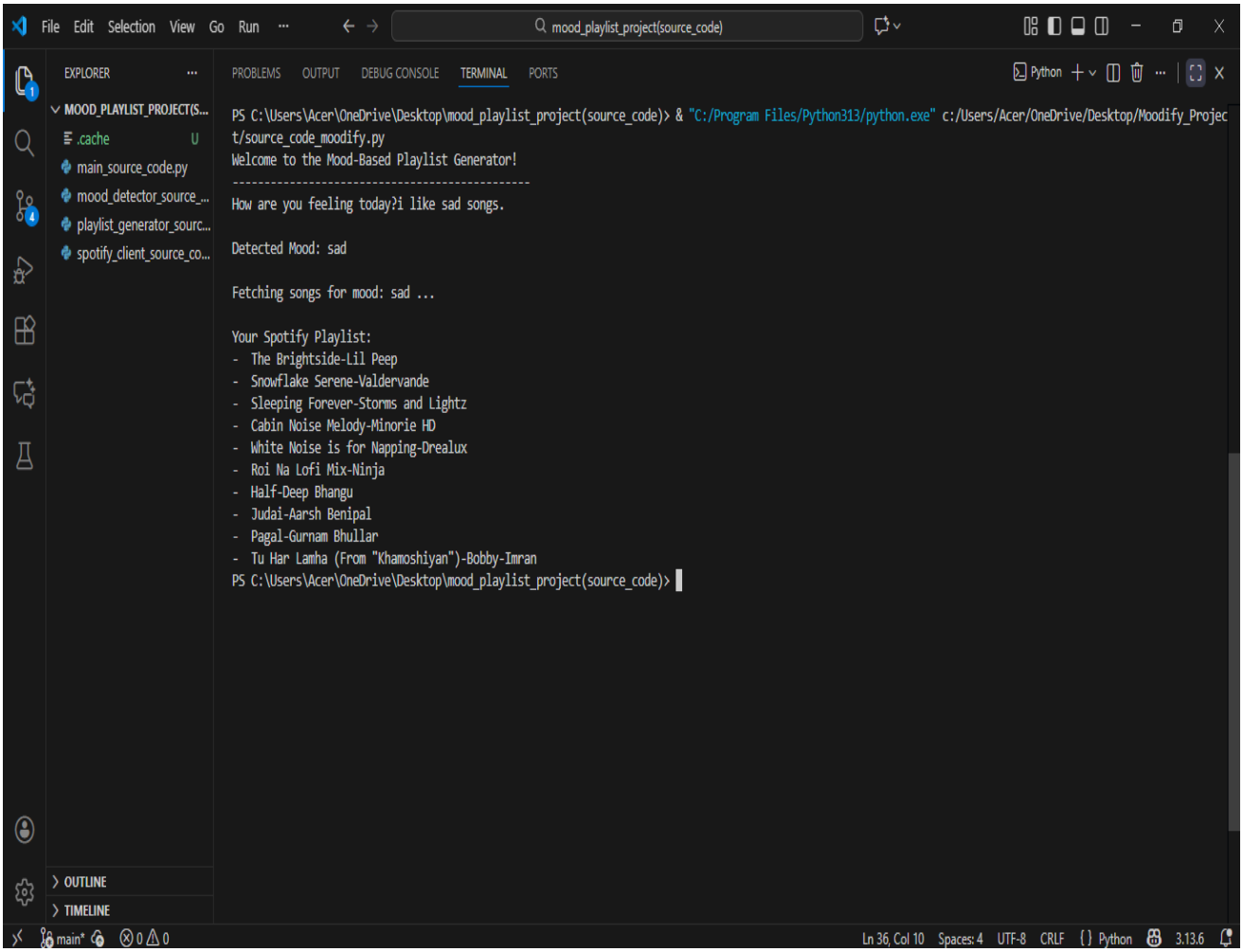
The implementation includes four Python files: main program, mood detector, playlist generator, and Spotify client. Spotipy is used for API communication. The system outputs playlists in the terminal.

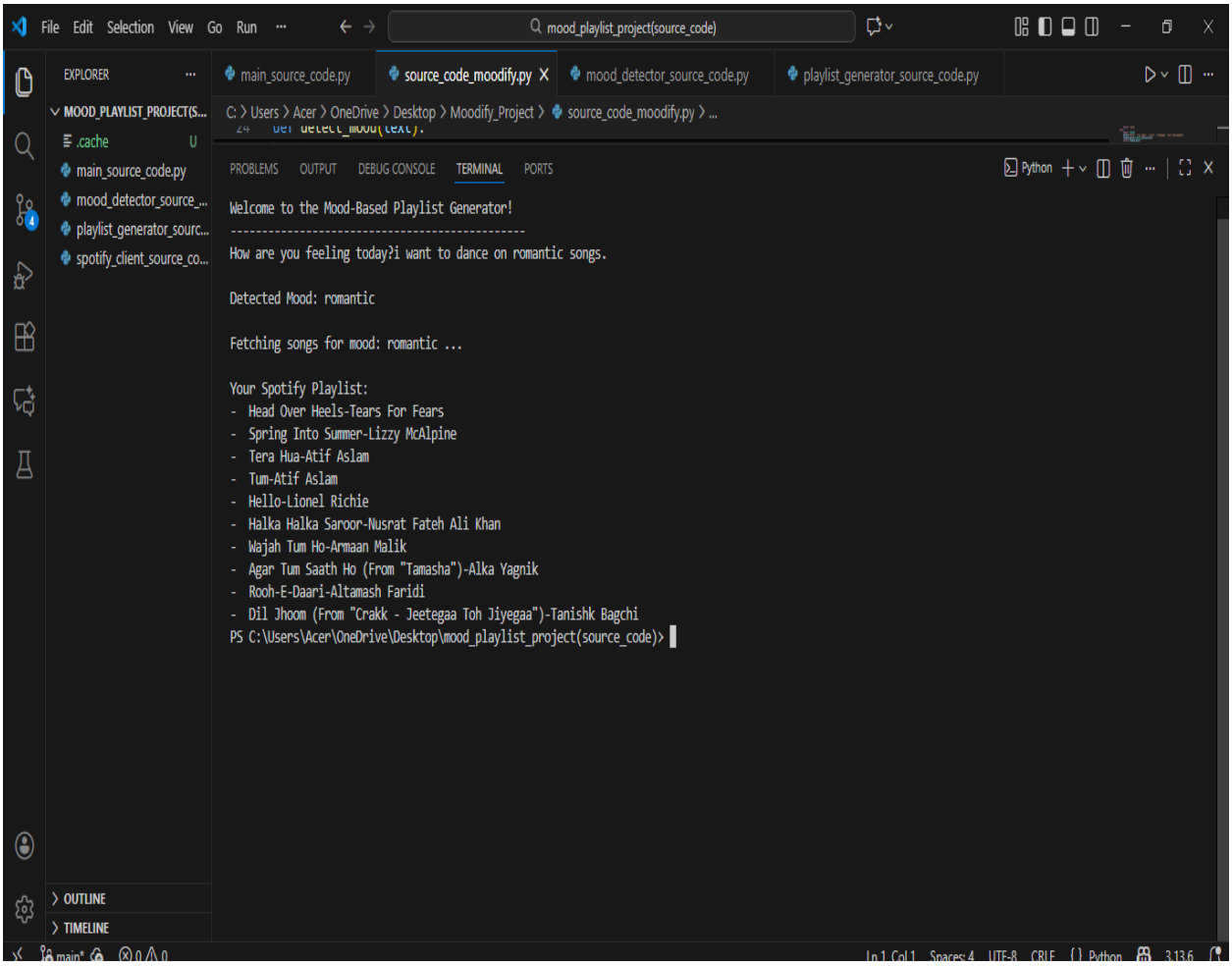
9. Screenshots /



Results


File Edit Selection View Go Run
mood_playlist_project(source_code)
EXPLORER
MOOD_PLAYLIST_PROJECT(S...
.cache
main_source_code.py
mood_detector_source_...
playlist_generator_sourc...
spotify_client_source_co...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Acer\OneDrive\Desktop\mood_playlist_project(source_code)> & "C:/Program Files/Python313/python.exe" c:/Users/Acer/OneDrive/Desktop/Moodify_Project/source_code_moodify.py
Welcome to the Mood-Based Playlist Generator!
How are you feeling today?i am feeling very happy today.
Detected Mood: happy
Fetching songs for mood: happy ...
Your Spotify Playlist:
- Happy-Doja Cat
- Happy and You Know It - Music Time-Little Baby Bum Nursery Rhyme Friends
- Happy Budday-Shankar-Ehsaan-Loy
- Happy Birthday To You-Mohammed Rafi
- Happy Birthday To You - Kids Song-Happy Birthday Songs
- Happy Birthday Shyam Baba-Kanhiya Mittal
- Shut Up and Dance-WALK THE MOON
- Un Manasula Paattuthaan - Happy-Ilaiyaraaja
- Take It Easy-M. Jayachandran
- Jail - From "Gangland"-Labh Heera
PS C:\Users\Acer\OneDrive\Desktop\mood_playlist_project(source_code)>
powershell
powershell
Python
powershell
powershell
powershell
Python
OUTLINE
TIMELINE
main*
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.6


File Edit Selection View Go Run ···
mood_playlist_project(source_code)
EXPLORER
MOOD_PLAYLIST_PROJECT(S...
.cache U
main_source_code.py
mood_detector_source_...
playlist_generator_sourc...
spotify_client_source_co...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
PS C:\Users\Acer\OneDrive\Desktop\mood_playlist_project(source_code)> & "C:/Program Files/Python313/python.exe" c:/Users/Acer/OneDrive/Desktop/Moodify_Project/source_code_moodify.py
Welcome to the Mood-Based Playlist Generator!
How are you feeling today?i like sad songs.
Detected Mood: sad
Fetching songs for mood: sad ...
Your Spotify Playlist:
- The Brightside-Lil Peep
- Snowflake Serene-Valdervande
- Sleeping Forever-Storms and Lightz
- Cabin Noise Melody-Minorie HD
- White Noise is for Napping-Drealux
- Roi Na Lofi Mix-Ninja
- Half-Deep Bhangu
- Judai-Aarsh Benipal
- Pagal-Gurnam Bhullar
- Tu Har Lamha (From "Khamoshiyan")-Bobby-Imran
PS C:\Users\Acer\OneDrive\Desktop\mood_playlist_project(source_code)>
OUTLINE
TIMELINE
main*
Ln 36, Col 10 Spaces: 4 UTF-8 CRLF Python 3.13.6



10. Testing Approach

Testing includes verifying mood detection accuracy, playlist relevance, fallback behavior, and API response checks.

11. Challenges Faced

- Handling mood ambiguity
- API rate limits
- Ensuring diverse playlist results

12. Learnings & Key Takeaways

- Improved understanding of APIs
- Gained experience with modular programming

- Learned how to integrate external services in Python

13. Future Enhancements

- Add GUI
- Advanced NLP for emotion detection- Save playlists to Spotify account

14. References

- Spotify Developer Documentation
- Spotipy Library Documentation