

# **INFOSYS SPRINGBOARD VIRTUAL INTERNSHIP 6.0 COMPLETION REPORT**

## **Team Details**

Batch Number : **6.0**

Start date : **13 August 2025**

Internship Duration: **8 Weeks**

## **1. Project Title**

**ManoRaag** - Emotion Based Song Recommendation System

## **2. Project Objective**

The primary objective of the project “**ManoRaag - Emotion-Based Song Recommendation System using Spotify API**” was to develop an intelligent application that bridges the gap between human emotions and personalized music experiences. The project aimed to detect users’ emotions through facial expressions and provide curated music recommendations that align with their current mood.

This solution not only enhances user satisfaction by offering emotionally relevant playlists but also demonstrates the potential of AI-driven personalization in digital entertainment. For the organization, the project highlights innovation in combining artificial intelligence, machine learning, and frontend technologies to address real-world user engagement challenges, ultimately contributing to enhanced user interaction, mental well-being support, and technological advancement in emotion-aware applications.

## **3. Project description in detail**

The project “**ManoRaag - Emotion-Based Song Recommendation System using Spotify API**” focuses on enhancing music consumption by integrating artificial intelligence with emotional intelligence. The core idea was to create an application capable of analyzing a user’s emotions in real time and recommending songs that resonate with their current state of mind.

The approach involved three key phases:

### ❖ **Emotion\_Detection:**

Using advanced computer vision techniques and deep learning models (TensorFlow, OpenCV), the system analyzes facial expressions captured through image uploads or webcam. These expressions are mapped to specific emotions such as happiness, sadness, anger, surprise, or calmness.

### ❖ **Music\_Recommendation:**

Based on the detected emotion, the system communicates with the **Spotify API** to fetch relevant songs and playlists. This ensures that the recommendations are dynamic, diverse, and emotionally aligned with the user's current state.

### ❖ **UserExperience\_And\_Interaction:**

A front-end interface was developed using **React**, integrated with a **Flask backend**. The application also includes features such as emotion-based playlists, chatbot (mood companion), and webcam uploads, making it interactive and user-friendly.

### ❖ **Technologies Used**

- **Frontend & Backend:** React, Flask
- **Authentication:** Google Firebase
- **AI/ML Frameworks:** TensorFlow, OpenCV
- **Music Integration:** Spotify API

### **Real\_World\_Impact:**

This project demonstrates how technology can create more personalized and meaningful digital experiences. By tailoring music recommendations to emotional states, the system improves user satisfaction, supports mental health and well-being, and paves the way for broader applications in industries such as entertainment, healthcare, and education. In real-world use, this type of system could act as a digital companion, offering comfort through music and potentially reducing stress, anxiety, and mood-related challenges.

## 4. Timeline Overview

Week	Activities Planned	Activities Completed
<b>Week 1</b>	Set up repository, development environments, and dependencies. Prepare dataset for training.	Repository created, environments set up, dataset collected and preprocessed.
<b>Week 2</b>	Define baseline model, set up Spotify developer account, and build frontend skeleton.	Baseline CNN model designed, Spotify app configured, frontend with webcam preview and placeholders created.
<b>Week 3</b>	Train CNN model and test real-time emotion detection using webcam and image upload.	Model trained and evaluated, real-time and upload-based detection tested successfully.
<b>Week 4</b>	Develop backend APIs and integrate Spotify for recommendations.	Backend APIs implemented, Spotify integration completed, emotion-to-genre mapping finalized.
<b>Week 5</b>	Connect frontend with backend, display detected emotions, and fetch music tracks.	Frontend integrated with backend, UI shows emotions and Spotify tracks, preview playback enabled.
<b>Week 6</b>	Improve UI/UX, optimize performance, and add chatbot feature.	UI polished, prediction smoothing applied, caching implemented, chatbot integrated.
<b>Week 7</b>	Conduct bug fixing, security checks, and prepare final documentation.	Bugs resolved, security checks completed, documentation drafted.
<b>Week 8</b>	Final testing, review, and presentation.	End-to-end testing completed, documentation finalized, project presented.

## 5a. Key Milestones

Milestone	Description	Date Achieved
<b>Project Kickoff</b>	Introduction to the project team, mentor, and coordinator. Overview of project objectives, tools, and expected deliverables.	13 August 2025
<b>Prototype/First Draft</b>	Initial implementation of the system. Completed dataset preparation, baseline CNN model design, and a frontend skeleton with webcam preview.	26 August 2025
<b>Mid-Term Review</b>	Review of progress including trained emotion detection model, real-time webcam/image upload detection, and Spotify integration for recommendations.	09 September 2025
<b>Final Submission</b>	Full system completion with frontend-backend integration, optimized performance, chatbot feature, and finalized documentation and report submission.	03 October 2025
<b>Presentation</b>	Final presentation and live demo of the project. Included rehearsals, mock runs, and feedback incorporation for smooth delivery.	03 October 2025

## 5b. Project execution details

The execution of the project “***ManoRaag - Emotion-Based Song Recommendation System using Spotify API***” followed a structured, milestone-driven approach to ensure smooth progress from planning to deployment. The workflow was divided into four key phases, each contributing towards building a fully functional and optimized application.

### ❖ Initial Setup and Foundations (Weeks 1–2):

- A clean repository structure was created with separate folders for frontend, backend, machine learning, and documentation.
- Development environments for Python and Node.js were set up, and required dependencies such as TensorFlow, OpenCV, Flask, and React were installed.
- The facial emotion dataset was collected, pre processed (resized, normalized, converted to grayscale), and organized into training, validation, and testing splits.
- A baseline CNN model was designed, while the frontend skeleton with a webcam preview and placeholders for emotion display and music player was implemented.
- A Spotify developer account was created, and secure storage of API credentials was configured.

### ❖ Model Training and Emotion-to-Music Mapping (Weeks 3–4):

- The CNN model was trained on the preprocessed dataset, with hyperparameter tuning and validation to improve accuracy.
- Real-time emotion detection was tested using webcam feed and uploaded images.
- Backend APIs were developed to handle image input, process emotions, and return predictions.
- Spotify integration was added to map detected emotions to appropriate genres and fetch track recommendations dynamically.

**❖ Integration and Functional Application (Weeks 5–6):**

- The frontend and backend were integrated, enabling smooth data flow between emotion detection and music recommendations.
- The UI was enhanced to display detected emotions, confidence levels, and recommended Spotify tracks with preview playback.
- Additional features such as a chatbot companion and playlist variations were developed.
- Performance optimization was applied, including prediction smoothing, caching of Spotify results, and responsive design adjustments.

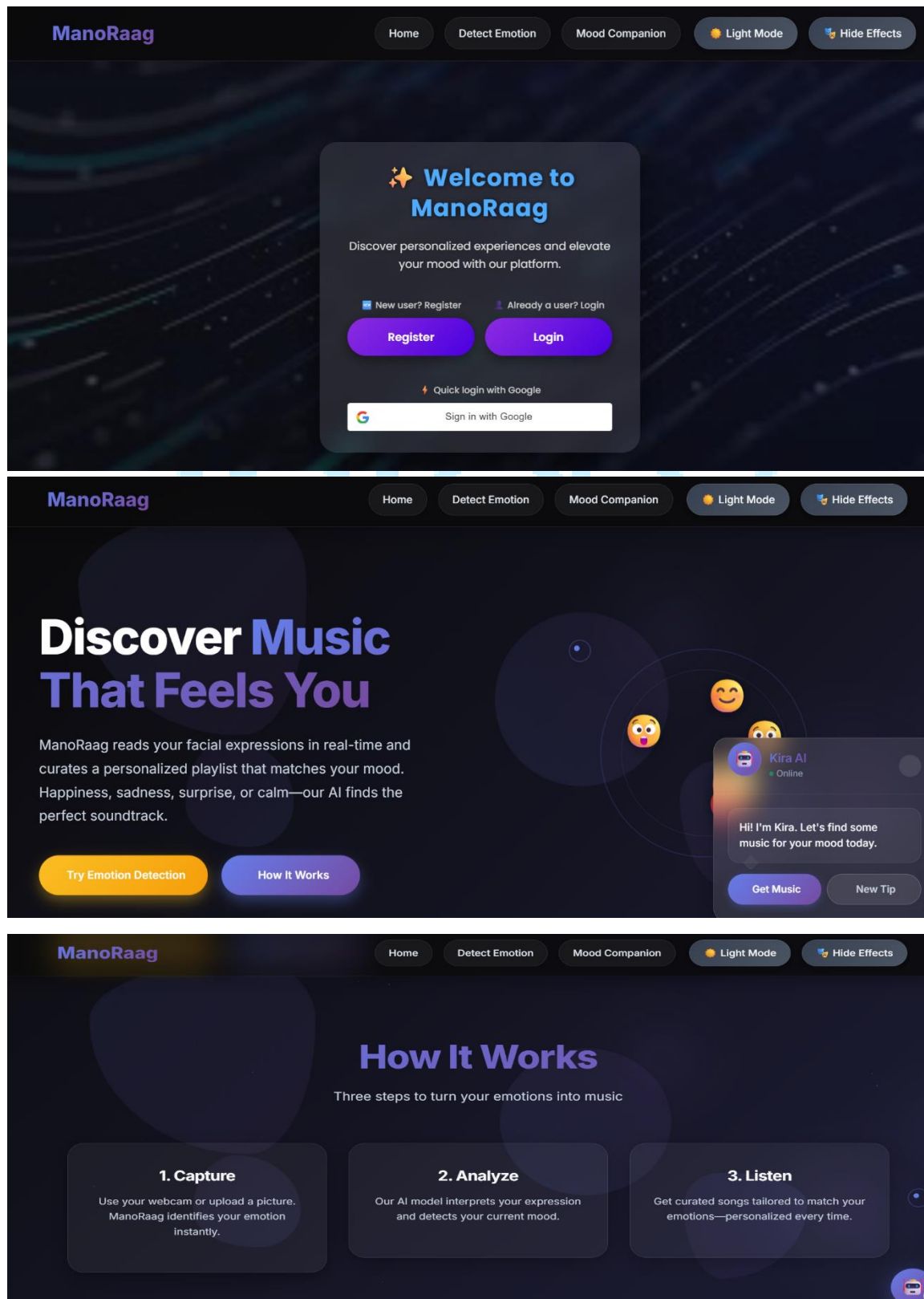
**❖ Testing, Optimization, and Finalization (Weeks 7–8):**

- Comprehensive testing was carried out for both webcam and upload-based pipelines, covering edge cases such as multiple faces, no faces detected, and invalid file inputs.
- Bugs were fixed, performance was optimized, and security checks were performed.
- The documentation and final project report were prepared, followed by mock presentations and feedback sessions.
- The project concluded with the final submission of deliverables and a live presentation to the team.

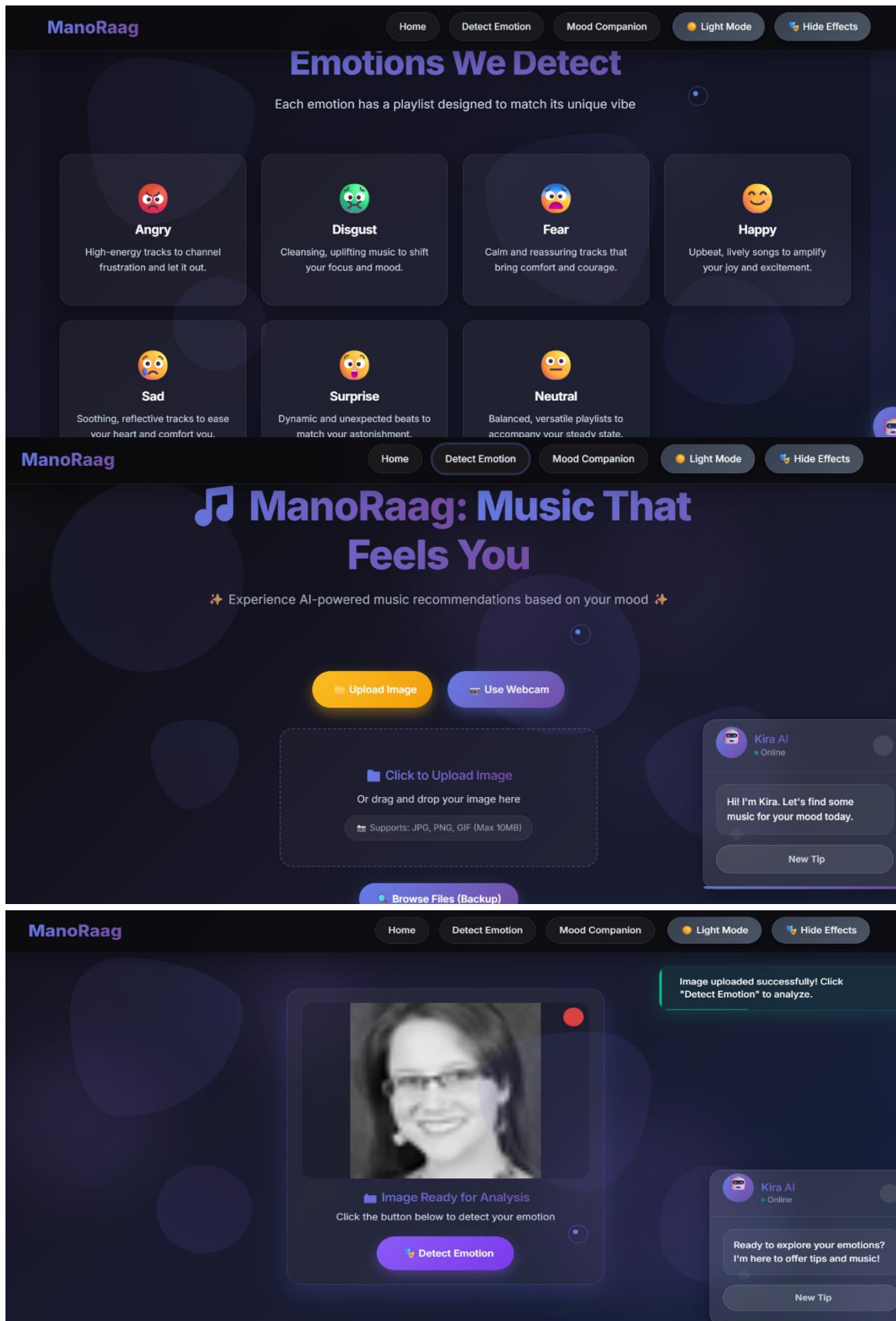
**Overall Execution Outcome:**

The project was executed systematically with continuous testing, integration, and feedback. By the end of the internship, a fully functional emotion-aware music recommendation system was developed, successfully demonstrating how artificial intelligence and music APIs can enhance user experiences through personalized recommendations.

## 6. Snapshots / Screenshots







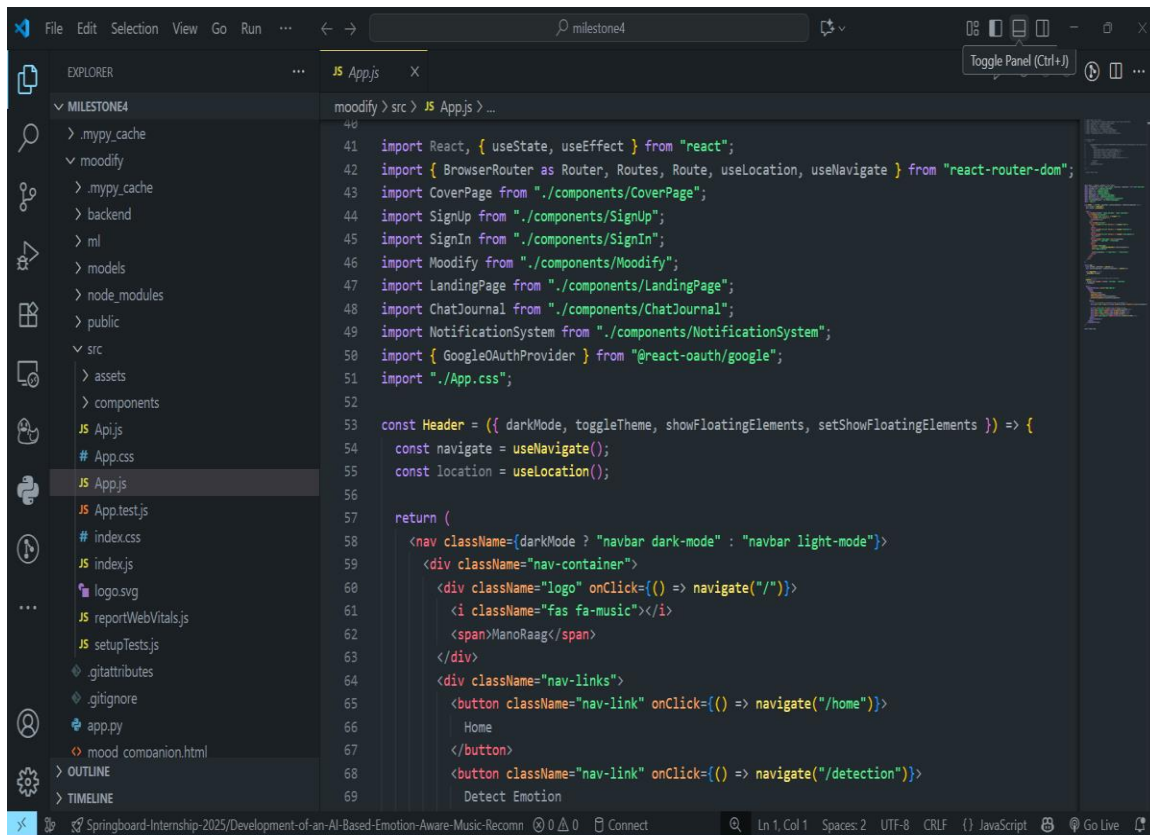


The image displays three screenshots of the ManoRaag application interface, which is designed for mood-based music discovery and journaling.

**Top Screenshot:** The main interface features a navigation bar with options: Home, Detect Emotion, Mood Companion, Light Mode, and Hide Effects. Below the navigation bar, there are three filter sections: Song Limit (a slider from 1 to 50, currently set at 10), Languages (10/10, with a checkbox for "Select All Languages"), and Artists (234/234, with a checkbox for "Select All Artists"). Below these filters, it says "Showing 10 of 2340 filtered songs" and "Total: 2340 available". The main content area shows three song cards: "Viva Forever" by P!nk, "Mood" by Alex Lenard, and "Mood" by Pop Box.x. Each card includes a play button and a "Mood" label.

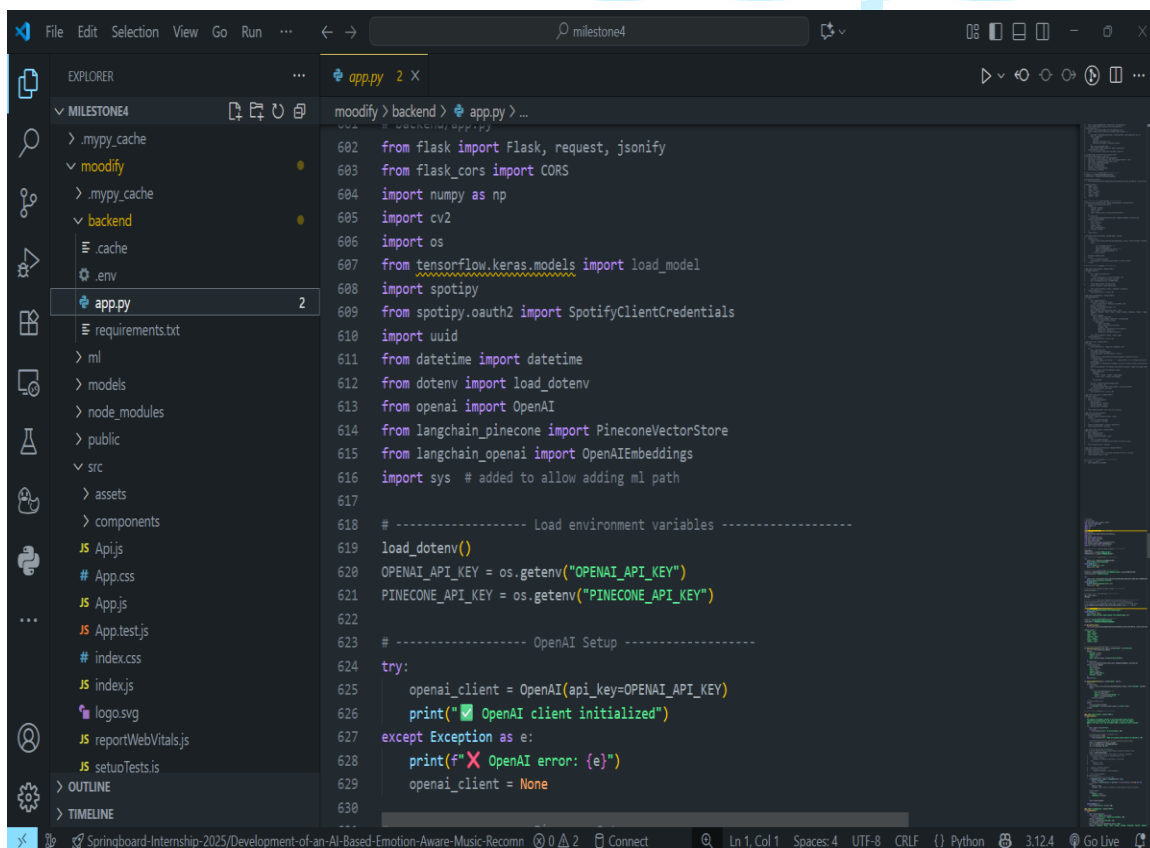
**Middle Screenshot:** This screenshot shows the Chatbot interface. The chatbot has responded to the user's message "I am feeling Happy suggest me songs" with a list of 7 songs: 1. "Happy" by Pharrell Williams, 2. "Walking on Sunshine" by Katrina and the Waves, 3. "Good Vibrations" by The Beach Boys, 4. "Shake It Off" by Taylor Swift, 5. "Uptown Funk" by Mark Ronson ft. Bruno Mars, 6. "Can't Stop the Feeling!" by Justin Timberlake, and 7. "I'm a Believer" by The Monkees. The chatbot also includes a "Send" button and a "Save Journal" button.

**Bottom Screenshot:** This screenshot shows the "Saved Journals" section. It includes a search bar for "Search journal by title...". Below the search bar, there is a list of saved journals. The first journal is titled "sad" (2025-09-29 14:21:44). The chatbot has responded to the user's message "i am feeling sad, suggest me song" with a list of 5 songs: 1. "Fix You" by Coldplay, 2. "The Night We Met" by Lord Huron, 3. "Someone Like You" by Adele, 4. "Hallelujah" by Jeff Buckley, and 5. "Slipping Through My Fingers" by ABBA. The chatbot also includes a "Delete" button and a "Save Journal" button.



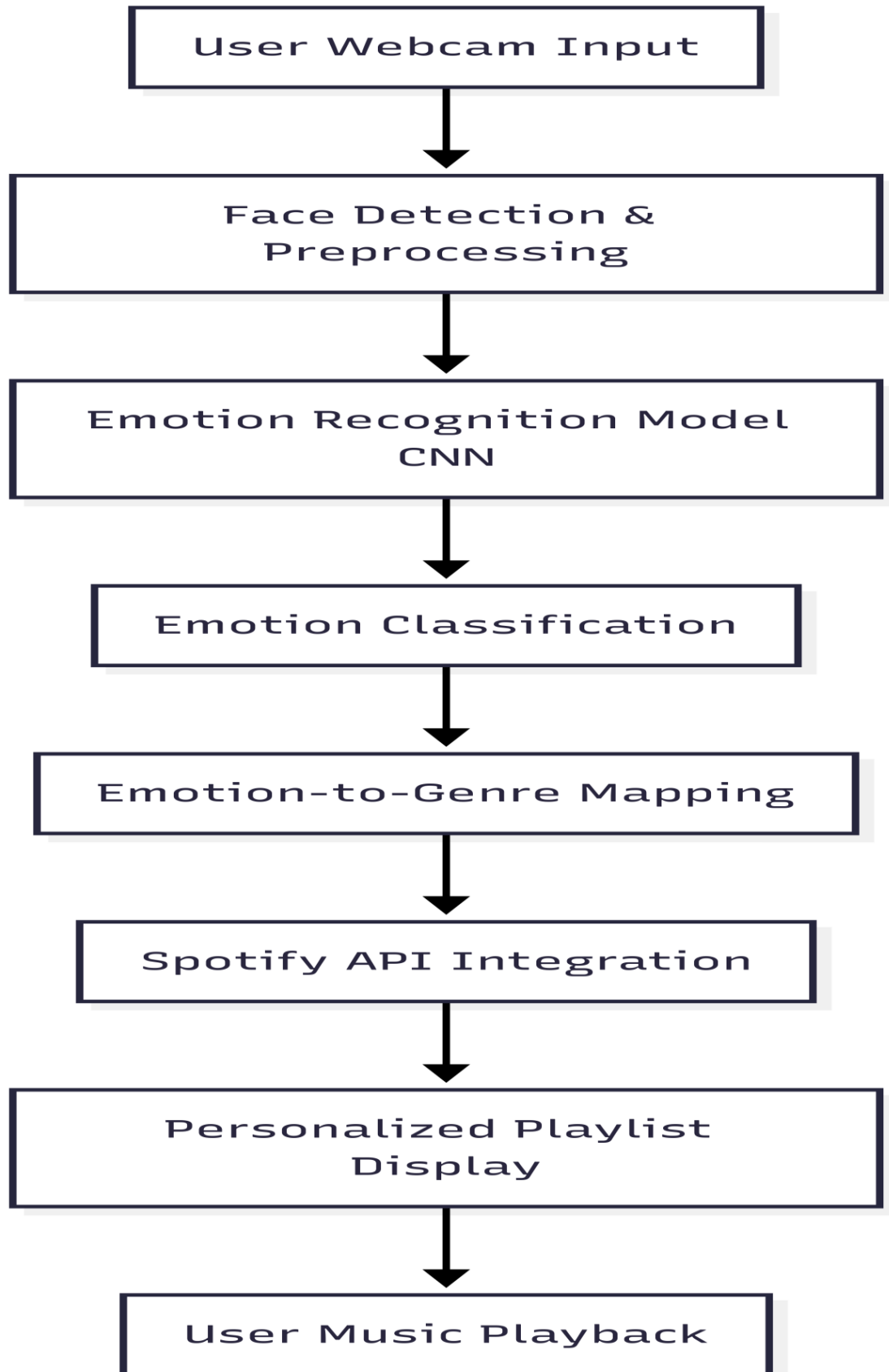
The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like .myppy\_cache, modify, .myppy\_cache, backend, ml, models, node\_modules, public, src, assets, components, JS App.js, JS App.test.js, # index.css, JS index.js, logo.svg, reportWebVitals.js, JS setupTests.js, .gitattributes, .gitignore, app.py, mood\_companion.html, OUTLINE, and TIMELINE. The main editor area shows the App.js file with the following code:

```
41 import React, { useState, useEffect } from "react";
42 import { BrowserRouter as Router, Routes, Route, useLocation, useNavigate } from "react-router-dom";
43 import CoverPage from "./components/CoverPage";
44 import SignUp from "./components/SignUp";
45 import SignIn from "./components/SignIn";
46 import Moodify from "./components/Moodify";
47 import LandingPage from "./components/LandingPage";
48 import ChatJournal from "./components/ChatJournal";
49 import NotificationSystem from "./components/NotificationSystem";
50 import { GoogleAuthProvider } from "@react-oauth/google";
51 import "./App.css";
52
53 const Header = ({ darkMode, toggleTheme, showFloatingElements, setShowFloatingElements }) => {
54   const navigate = useNavigate();
55   const location = useLocation();
56
57   return (
58     <nav className={darkMode ? "navbar dark-mode" : "navbar light-mode"}>
59       <div className="nav-container">
60         <div className="logo" onClick={() => navigate("/")}>
61           <i className="fas fa-music"></i>
62           <span>ManoRaag</span>
63         </div>
64         <div className="nav-links">
65           <button className="nav-link" onClick={() => navigate("/home")}>
66             Home
67           </button>
68           <button className="nav-link" onClick={() => navigate("/detection")}>
69             Detect Emotion
```



The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like .myppy\_cache, modify, .myppy\_cache, backend, .cache, .env, app.py, requirements.txt, ml, models, node\_modules, public, src, assets, components, JS App.js, JS App.test.js, # index.css, JS index.js, logo.svg, reportWebVitals.js, JS setupTests.js, OUTLINE, and TIMELINE. The main editor area shows the app.py file with the following code:

```
602 from flask import Flask, request, jsonify
603 from flask_cors import CORS
604 import numpy as np
605 import cv2
606 import os
607 from tensorflow.keras.models import load_model
608 import spotipy
609 from spotipy.oauth2 import SpotifyClientCredentials
610 import uuid
611 from datetime import datetime
612 from dotenv import load_dotenv
613 from openai import OpenAI
614 from langchain_pinecone import PineconeVectorStore
615 from langchain_openai import OpenAIEmbeddings
616 import sys # added to allow adding ml path
617
618 # ----- Load environment variables -----
619 load_dotenv()
620 OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")
621 PINECONE_API_KEY = os.getenv("PINECONE_API_KEY")
622
623 # ----- OpenAI Setup -----
624 try:
625     openai_client = OpenAI(api_key=OPENAI_API_KEY)
626     print("✅ OpenAI client initialized")
627 except Exception as e:
628     print(f"❌ OpenAI error: {e}")
629     openai_client = None
630
```



## 7. Challenges Faced

During the course of the internship project, several challenges were encountered across technical, operational, and communication aspects.

From a **technical perspective**, one of the key challenges was handling inconsistencies in the facial emotion dataset. The dataset contained imbalanced classes, leading to biased predictions during early model training. This issue was mitigated by applying data augmentation techniques and carefully splitting the dataset to maintain proportional representation across all classes. Another technical challenge was optimizing the real-time webcam detection pipeline. Initial implementations resulted in slow frame processing and occasional lag in predictions. This was resolved by resizing frames efficiently, applying prediction smoothing, and caching results to improve performance. Additionally, Spotify API integration posed difficulties in managing authentication tokens and handling missing preview URLs. These were resolved by securely storing API keys in environment variables, refreshing tokens as needed, and implementing fallback mechanisms when certain tracks were unavailable.

On the **operational side**, coordinating the workflow between machine learning, backend, and frontend components initially caused delays due to interdependencies. This was addressed by following a milestone-based approach, where tasks were divided into smaller deliverables that could be tested independently before integration. Proper documentation etc.

From a **communication standpoint**, aligning the team's understanding of requirements and expectations with the mentor and coordinator was crucial. Occasional misinterpretations regarding technical scope were resolved through regular review meetings, continuous feedback, and progress demonstrations. This helped in refining the project direction and ensured that all stakeholders were aligned throughout the development process.

Overall, the challenges faced during the internship provided valuable learning experiences. By applying problem-solving strategies such as data augmentation, optimization techniques, modular development, and consistent communication, the team was able to successfully mitigate issues and deliver a functional and reliable emotion-aware music recommendation system.

## 8. Learnings & Skills Acquired

### Technical Learnings:

- Practical application of **Artificial Intelligence and Machine Learning** in facial emotion recognition.
- Hands-on experience with **TensorFlow** and **OpenCV** for dataset preparation, preprocessing, model training, and evaluation.
- Addressing challenges with **imbalanced datasets** and applying optimization techniques.
- Integration of **Spotify API** for music recommendations, gaining knowledge of external APIs, authentication flows, and dynamic data handling.
- Backend development using **Flask** and frontend integration with **React**, enhancing full-stack development skills.
- Implementation of **Firebase** for authentication and data management in cloud applications.
- Proficiency in **Python, NumPy, Pandas, Matplotlib** for data handling and visualization.
- Effective use of **Git and GitHub** for version control and collaborative development.
- Improved skills in **debugging, testing, and performance optimization**.

### Soft Skills & Professional Learnings:

- Enhanced **team collaboration** through coordinated milestone-based execution.
- Improved **communication and presentation** skills via regular discussions and knowledge-sharing sessions.
- Stronger **time management and planning abilities**, ensuring tasks were completed within deadlines.

## 9. Testimonials from team

Working on the “**ManoRaag - Emotion-Based Song Recommendation System using Spotify API**” was an enriching experience that enhanced both our technical and teamwork skills. Breaking the project into milestones helped us manage tasks efficiently and achieve steady progress.

We gained valuable knowledge in machine learning, emotion detection, and API integration, which gave us confidence in building real-world applications. The successful combination of AI-based emotion recognition with music recommendations was a major success point for the team.

Beyond technical growth, this project strengthened our collaboration, communication, and problem-solving abilities. The guidance and support from our mentor and coordinator were truly invaluable in keeping us aligned and motivated.

Overall, the project was a memorable journey that improved our skills and teamwork, while delivering a meaningful and innovative solution.

## 10. Conclusion

The internship project “**ManoRaag - EmotionBased Song Recommendation System using Spotify API**” has been a comprehensive and transformative learning experience. It provided us with the opportunity to apply theoretical knowledge into practice by developing an end-to-end system that combines **artificial intelligence, machine learning, and modern web technologies to solve a real-world problem**. Through this project, we not only enhanced our technical expertise but also strengthened our ability to work collaboratively in a structured and goal-oriented manner.

The project demonstrated the potential of AI-driven applications in bridging human emotions with digital experiences. By **detecting emotions through facial expressions and mapping them to music recommendations, the system successfully showcased how technology can be harnessed to create personalized and meaningful user interactions**. This also highlighted the role of innovation in addressing challenges related to mental well-being, engagement, and personalization in the digital era.

On the technical front, we gained hands-on experience with tools such as **TensorFlow, OpenCV, Flask, React, Firebase, and the Spotify API**, along with essential practices like **dataset preparation, model evaluation, and**

**performance optimization.** On the professional front, the project taught us **valuable lessons in communication, teamwork, problem-solving, and milestone-based execution.** The guidance and support from our mentor and coordinator played a crucial role in refining our approach and keeping us motivated throughout the internship.

From an academic and career perspective, this project aligns strongly with our future aspirations in the fields of artificial intelligence, data science, and software development. It gave us the confidence to build practical solutions, adapt quickly to challenges, and contribute meaningfully to projects that combine innovation with impact.

In conclusion, the successful completion of this project stands as a testament to our learning, dedication, and collaboration. It has not only helped us grow as professionals but also inspired us to continue exploring AI-based applications that can improve everyday experiences and contribute to society in meaningful ways.

## 11. Acknowledgements

We would like to express our heartfelt gratitude to our mentor, **Ms. Revathi Venugari**, for her invaluable guidance, constant encouragement, and insightful feedback throughout the course of this internship. Her support played a crucial role in shaping our understanding and helping us overcome challenges at every stage of the project.

We also sincerely thank our coordinator, **Mr. Pradyumna V**, for his continuous support, coordination, and motivation, which ensured smooth progress and timely completion of the project milestones.

Our appreciation extends to the organization for providing us with the opportunity to work on this innovative project. The resources, environment, and learning opportunities offered during the internship have been instrumental in enhancing our technical knowledge and professional skills.

Finally, we would like to acknowledge the collaborative effort of our team members, whose dedication, cooperation, and teamwork were key factors in the successful completion of the project.