A Deep Dive into LSTM and GRU for Social Media Analysis

A Project Report

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

Computer Science and Engineering with Specialization in

"Artificial Intelligence and Machine Learning"

Submitted By:

RAJNANDINI BHOWMICK (20BCS6578)

Under the Supervision of:

Mr. Pulkit Dwivedi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

January - April, 2024

BONAFIDE CERTIFICATE

Certified that this project report **"A Deep Dive Into LSTM And GRU For Social Media Analysis"** is the Bonafide work of **Rajnandini Bhowmick** who carried out the project work under my supervision.

SIGNATURE OF THE HOD
MR. AMAN KOUSHIK
(HEAD OF DEPARTMENT)
CSE- AIML

SIGNATURE OF THE SUPERVISOR
MR. PULKIT DWIVEDI
(SUPERVISOR)
(Asst. Professor)
(AIT-CSE)

Submitted for the project viva- voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task wouldbe incomplete without the mention of the people who made it possible. Success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, it is with gratitude that we acknowledge all those whose guidance and encouragementserved as beacons of light and crowned our effort with success.

We consider it a privilege and honor to express our sincere gratitude to our internal guide, Mr. Pulkit Dwivedi, Asst. Professor, Department of Computer Science & Engineering for hervaluable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very Co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

Rajnandini Bhowmick-20BCS6578

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive

analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation.

Keywords:
customers, marketers, machine learning, product sentiment, social media posts, Long Short-Term Memory, Gated Recurrent Unit

Chapter 1: INTRODUCTION

In recent years, the widespread adoption of social media platforms has revolutionized the way individuals and organizations communicate, share information, and express opinions. With millions of users actively engaging on platforms like Twitter, Facebook, and Instagram, social media has become a rich source of unstructured data containing valuable insights into public sentiments, preferences, and behaviors. Sentiment analysis, also known as opinion mining, has emerged as a powerful tool to extract and analyze sentiments expressed in social media content, enabling businesses, researchers, and policymakers to gain valuable insights into public perception, brand sentiment, market trends, and more.

This report focuses on the project of sentiment analysis on social media, aiming to explore and implement various techniques and methodologies to effectively analyze sentiments expressed in social media posts. The project encompasses the use of natural language processing (NLP) techniques and machine learning (ML) algorithms to classify sentiments such as positive, negative, or neutral based on textual content.

The introduction section provides an overview of the importance of sentiment analysis in today's digital landscape, highlighting its relevance in understanding customer sentiment, social trends, and market sentiments. It also outlines the objectives and scope of the project, including the methodologies, datasets, tools, and evaluation metrics used throughout the analysis.

Additionally, this introduction sets the stage for discussing the challenges and opportunities associated with sentiment analysis on social media data. Challenges may include handling noisy and unstructured text data, dealing with sarcasm and irony, language nuances, and ensuring the ethical use of data. Opportunities, on the other hand, arise from the vast amount of data available on social media platforms, offering insights that can drive decision-making processes across various domains.

By delving into the nuances of sentiment analysis on social media, this report aims to provide a comprehensive understanding of the techniques, implementations, and implications of sentiment analysis in harnessing the power of social media data for actionable insights and informed decision-making.

In today's interconnected world, social media has revolutionized how individuals and organizations interact, share information, and express opinions on a global scale. Platforms such as Twitter, Facebook, Instagram, and Reddit serve as virtual arenas where users engage in conversations, share experiences, voice concerns, and express sentiments about diverse topics ranging from politics and entertainment to products and services. This wealth of user-generated content presents a valuable opportunity to gain insights into public perceptions, sentiment trends, and brand sentiments, making sentiment analysis on social media a crucial area of study and application.

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) and machine learning (ML) that focuses on extracting and categorizing sentiments expressed in text data. The primary goal of sentiment analysis is to understand and quantify the polarity of opinions or emotions conveyed in textual content, whether it is positive, negative, or neutral. This information is invaluable for businesses, marketers, policymakers, and researchers to gauge public sentiment, monitor brand perception, analyze customer feedback, and make data-driven decisions.

Social media platforms have become rich sources of unstructured text data, containing a plethora of opinions, emotions, and sentiments shared by millions of users worldwide. Analyzing this data manually is impractical due to its volume and real-time nature. Hence, sentiment analysis techniques, powered by NLP and ML algorithms, offer scalable and efficient solutions to extract meaningful insights from social media content.

This report provides a comprehensive overview of sentiment analysis techniques applied specifically to social media data. It covers theoretical concepts, practical implementation steps, experimental results, and discussions on applications and ethical considerations. The report focuses on sentiment classification tasks for English text data sourced from popular social media platforms, considering both supervised and unsupervised learning approaches.

The scope also includes discussions on data preprocessing techniques, feature engineering, model selection criteria, evaluation metrics, and result interpretation. Practical implementation of sentiment analysis models using Python programming language and relevant libraries such as NLTK, Scikit-learn, and TensorFlow is explored to provide hands-on insights into model development and evaluation.

In today's digitally driven world, the pervasive influence of social media platforms has fundamentally transformed how individuals interact, communicate, and share information. From the concise tweets on Twitter to the visually captivating posts on Instagram, and the sprawling discussions on Reddit, social media platforms serve as dynamic arenas where users express opinions, share experiences, and engage in conversations on a multitude of topics. Within this vast ecosystem of user-generated content lies a wealth of invaluable insights waiting to be unearthed. One such avenue of exploration is sentiment analysis, a burgeoning field at the intersection of natural language processing (NLP) and machine learning (ML), which seeks to extract and analyze sentiments expressed in textual data.

Sentiment analysis, often interchangeably referred to as opinion mining, entails the computational study of sentiments, opinions, and emotions conveyed within textual content. The overarching objective is to discern the polarity of sentiments, categorizing them as positive, negative, or neutral, thereby providing a quantitative understanding of the prevailing attitudes and emotions embedded within the text. This analytical approach finds wide-ranging applications across various domains, including but not limited to market research, brand management, customer feedback analysis, public opinion monitoring, and social media analytics

The impetus behind embarking on a project centered around sentiment analysis on social media stems from the sheer volume and diversity of user-generated content that permeates these platforms. Harnessing this vast reservoir of data holds immense potential for organizations, researchers, and policymakers alike. By deciphering the sentiments expressed within social media posts, stakeholders can glean actionable insights that inform decision-making processes and strategic initiatives across a spectrum of endeavors

This report endeavors to furnish a comprehensive overview of sentiment analysis techniques applied specifically within the milieu of social media data. It encompasses theoretical underpinnings, methodological considerations, practical implementation strategies, experimental findings, and nuanced discussions on the applications and ethical implications of sentiment analysis on social media platforms

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online

communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today’s digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media

platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment

understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.
Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This

abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today’s digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media

platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation
Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment

understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations

leveraging social media analytics for decision-making and strategy formulation.

Social media platforms have become ubiquitous channels for individuals to express opinions, emotions, and sentiments on a wide array of topics. With the proliferation of user-generated content, sentiment analysis has emerged as a vital tool for understanding the collective mood and opinions of online communities. This abstract aims to provide an overview of the landscape of social media sentiment analysis, highlighting its methodologies, applications, challenges, and future directions.

The methodology section delves into the various techniques employed in sentiment analysis, ranging from lexicon-based approaches to machine learning and deep learning algorithms. It elucidates the process of text preprocessing, feature extraction, and sentiment classification, emphasizing the importance of domain adaptation and context awareness in achieving accurate results:

In the application domain, this abstract discusses the diverse range of fields where social media sentiment analysis finds utility. From marketing and brand management to political analysis and public opinion monitoring, organizations leverage sentiment analysis to gain insights into consumer behavior, market trends, and societal sentiments.

However, despite its widespread adoption, sentiment analysis faces several challenges. These include the nuances of natural language, such as sarcasm, irony, and context-dependent meanings, which pose difficulties for automated systems. Moreover, issues related to data quality, bias, and ethical considerations underscore the need for continuous refinement and ethical guidelines in sentiment analysis research and applications.

Looking ahead, the abstract outlines potential avenues for advancing social media sentiment analysis. This includes the integration of multimodal data (text, images, videos) for more nuanced sentiment understanding, the development of robust sentiment lexicons tailored to specific domains and languages, and the exploration of deep learning architectures capable of capturing complex linguistic structures and emotional nuances.

In today's digital age, social media platforms have become a treasure trove of user-generated content, providing valuable insights into public opinions and sentiments. This report presents a comprehensive analysis of sentiment analysis techniques applied to social media data. The project focuses on leveraging natural language processing (NLP) and machine learning (ML) algorithms to classify and analyze

sentiments expressed in social media posts.

The report begins by discussing the importance of sentiment analysis in understanding public perception, brand sentiment, and market trends. It explores various methodologies for data collection from social media platforms, highlighting the challenges and opportunities presented by unstructured text data.

Next, the report delves into the technical aspects of sentiment analysis, including preprocessing steps such as tokenization, stemming, and stop-word removal. It evaluates different feature extraction techniques and sentiment classification algorithms, comparing their effectiveness in handling social media data.

Overall, this report serves as a comprehensive guide to understanding and implementing sentiment analysis techniques on social media data, offering valuable insights for researchers, practitioners, and organizations leveraging social media analytics for decision-making and strategy formulation

1.1 PROBLEM DEFINITION:

In the contemporary social media analysis, users often find themselves overwhelmed by the sheer abundance of options and genres available. While the digital era has empowered enthusiasts with unprecedented access to an extensive catalogue of social media, the challenge lies in navigating this vast sea of musical content to discover tracks that resonate with their current emotional states. Recognizing this gap in the user experience.

1. Lack of Personalization:
One of the primary issues faced by users in traditional music streaming platforms is the lack of personalization in the recommendation process. Current systems often rely on broad genre categorizations, historical data, or popular trends to suggest songs. However, these approaches overlook the dynamic nature of human emotions, leading to recommendations that may not align with the user's current mood or state of mind. Sentimental analysis on Social media aims to bridge this gap by introducing a real-time emotion recognition system to provide personalized music suggestions that resonate with the user's emotional context at any given moment.

2. User Engagement:

Static and generic music recommendations contribute to limited user engagement. Users may find themselves disinterested or disengaged when presented with playlists that fail to capture the nuances of their emotional experiences. Social Media addresses this issue by employing advanced facial emotion recognition technology to understand and adapt to the user's changing moods in real-time. This dynamic approach not only enhances user engagement but also creates a more immersive and personalized music discovery journey.

3. Overreliance on Historical Data:

Many existing social media analysis heavily depend on historical user data and explicit preferences. While this information is valuable, it might not accurately reflect the user's current emotional state. Users are dynamic beings, and their preferences and emotions evolve over time. Sentimental analysis seeks to overcome this limitation by integrating AIML algorithms that continuously analyse real-time facial expressions, ensuring that recommendations are based on the user's immediate emotional context rather than historical patterns alone.

4. Inadequate Understanding of User Emotions:

Traditional music recommendation systems often lack a nuanced understanding of user emotions. While some platforms may consider explicit feedback, the subtleties of emotional expression are often overlooked. Sentimental analysis on social media employs facial emotion recognition as a key technology to decode these subtle cues, providing a more sophisticated and accurate understanding of the user's emotional landscape. This technology allows the system to adapt and refine its recommendations based on the ever-changing emotional states of the user.

5. Impersonalized User Interfaces:

The user interface (UI) of many music streaming platforms may lack the personal touch necessary to create a deeply engaging experience. Sentimental analysis addresses this concern by designing an intuitive and visually appealing UI that incorporates a camera interface for facial emotion recognition. This interactive UI not only enhances the user experience but also serves as a novel means of engagement, allowing users to actively participate in the music discovery process through their emotions.

6. Cross-Platform Compatibility Challenges:

In the era of diverse device usage, achieving cross-platform compatibility is a crucial aspect of any digital service. Some existing music recommendation systems may struggle to provide a consistent experience across various devices and browsers. Sentimental Analysis on Social Media places a significant emphasis on ensuring that the web-based interface is responsive, accessible, and performs seamlessly across desktops, tablets, and smartphones, catering to the diverse preferences and habits of users.

1.2 PROJECT OVERVIEW:

In the ever-evolving landscape of digital music consumption, Social media emerges as a groundbreaking project, aiming to redefine the user experience by fusing advanced technologies with an innovative approach to music recommendation. This project is poised at the intersection of Artificial Intelligence and Machine Learning (AIML) tech stack, setting the stage for a dynamic and personalized music discovery journey.

1. Context and Motivation:

As the digital era unfolds, music streaming platforms have become integral to our daily lives, providing users with access to an expansive array of songs from various genres. However, the sheer abundance of choices often leads to a paradox of choice, leaving users overwhelmed and struggling to discover music that aligns with their current emotional states. The motivation behind Sentimental Analysis on Social Media stems from a desire to transform the static and impersonal nature of traditional music recommendations, offering users a platform that not only understands their emotions but adapts to their ever-changing moods in real-time.

2. Objectives and Goals:

The primary objective of Sentimental Analysis is to create a dynamic and personalized music recommendation system that leverages facial emotion recognition to understand the user's emotional state. The project aims to achieve the following goals:

- Real-Time Emotion Recognition: Implement a sophisticated facial emotion recognition system to analyse users' facial expressions in real-time, providing insights into their current emotional states.
- AIML Integration: Integrate AIML algorithms to accurately decode and categorize emotions based on facial expressions, ensuring a nuanced understanding of the user's mood.
- MERN Stack Architecture: Leverage the MERN stack to create a robust infrastructure, encompassing MongoDB for efficient database management, Express.js for server-side logic, React for dynamic user interfaces, and Node.js for seamless communication between the server and client.

- Personalized Music Recommendations: Develop a dynamic music recommendation engine that generates personalized song suggestions based on the user's emotional state, historical preferences, and real-time interactions.
- User Engagement: Enhance user engagement by providing an interactive web-based interface with a camera component for facial emotion recognition, allowing users to actively participate in the music discovery process through their emotions.
- Cross-Platform Compatibility: Ensure a seamless and consistent user experience across different devices and browsers, promoting accessibility and usability for a diverse user base.

3. Key Features:

3.1 Facial Emotion Recognition:

Sentimental Analysis integrates a state-of-the-art facial emotion recognition system that captures and analyses users' facial expressions in real-time. This technology decodes subtle emotional cues, laying the foundation for a personalized and dynamic music recommendation experience.

3.2 AIML Algorithms:

The project employs AIML algorithms to categorize and interpret facial expressions accurately. This advanced layer of intelligence ensures that the system adapts to the user's emotions with a high degree of precision, enhancing the overall quality of music recommendations.

3.3 MERN Stack Infrastructure:

The MERN stack serves as the architectural backbone of the project. MongoDB efficiently manages user profiles, music preferences, and real-time emotional data. Express.js facilitates secure server-side logic, while React and Node.js contribute to creating an interactive and responsive user interface.

3.4 Personalized Music Recommendation Engine:

At the heart of the project lies a sophisticated music recommendation engine that continuously evolves based on the user's preferences, historical data, and real-time emotional insights. This engine strives to curate playlists that align with the user's immediate emotional context.

3.5 User Interface Design:

Sentimental Analysis on Social Media places a significant emphasis on user interface design. The web-based interface is intuitive, visually appealing, and features a camera component for facial emotion recognition. This design choice not only enhances user engagement but also transforms the music discovery

process into an interactive and emotional journey.

3.6 Cross-Platform Compatibility:

Recognizing the diverse landscape of digital devices, the project ensures cross-platform compatibility. The web-based interface is designed to perform seamlessly across desktops, tablets, and smartphones, catering to the varied preferences and habits of users.

4. Implementation Phases:

4.1 System Setup:

The project kicks off with the setup of the MERN stack environment, laying the foundation for subsequent development phases.

4.2 Facial Emotion Recognition and AIML Integration:

This phase focuses on the implementation of the facial emotion recognition system and the integration of AIML algorithms for accurate emotion decoding.

4.3 Music Recommendation Engine and Database Integration:

The development of the music recommendation engine takes center stage, accompanied by the integration of MongoDB for efficient database management.

4.4 User Interface Design and Testing:

The final phases encompass user interface design, ensuring a seamless and visually appealing experience, followed by comprehensive testing to validate the system's functionality, security, and cross-platform compatibility.

1.3 HARDWARE SPECIFICATION:

- 16 GB RAM
- AMD RYZEN 7 5000 SERIES
- 512GB SSD
- VIDEO CARD: NVIDIA RTX 3060

1.4 SOFTWARE SPECIFICATION:

- Python Compiler with required Libraries and Modules
- Language: Python and JavaScript
- Operating System: Windows 7/8/10/11
- Frameworks: ReactJS, NodeJS and ExpressJS
- Database Used: MongoDB and Firebase (No SQL databases)
- Libraries:

  TensorFlow

  Keras

  Matplotlib

  NumPy

  PyTorch

  Pandas

  Sklearn

  OpenCV-python

Chapter 2: LITERATURE SURVEY

The development of EmotionWave Music Matcher draws upon a comprehensive review of existing literature, which encompasses various aspects of emotion-aware music recommendation systems and facial expression-based technologies. The synthesis of knowledge from seven key research papers provides a foundational understanding of the current landscape and informs the design and implementation of our innovative project.

Annam, Bodapati, and Konda (2024) [1] proposed a Transfer Learning Approach Using Facial Expressions for Emotion-Aware Music Recommendations. Their work explores the synergy between facial expressions and music preferences, paving the way for personalized recommendations based on emotional states. This research leverages advanced computer science and engineering techniques, fostering a deeper understanding of user emotions through facial cues.

Sharath P., Senthil Kumar G., and Vishnu Boj K.S. (2023) [2] contributed significantly to the field with their Music Recommendation System Using Facial Emotions. Operating within the realm of computational intelligence, this study establishes a foundation for recommending music based on facial expressions. The integration of computational intelligence in emotion recognition ensures a dynamic and adaptive music recommendation system.

Agrawal, Sharma, Fulzele, Bhasarkar, and Pande (2023) [3] extended the exploration by introducing a Mood-based Song Recommendation System Using CNN. Their work, situated in the School of Computer Engineering at MIT Academy of Engineering, employs convolutional neural networks to enhance the accuracy of mood-based music recommendations. This method leverages the power of deep learning in deciphering intricate patterns in facial expressions to better understand and predict user moods.

Bokhare and Kothari (2023) [4] delved into the domain of video recommendations in their Emotion Detection-Based Video Recommendation System Using Machine Learning and Deep Learning Framework. Situated at the Symbiosis Institute of Computer Studies and Research, their research expands the application of emotion-aware technologies beyond music, incorporating video content and showcasing the versatility of such systems.

Kumar, Nandi, Roy, and Sarkar (2023) [5] offered a unique perspective with their Design and Implementation of AI-Based Efficient Emotion Detection and Music Recommendation System. Conducted at Reva University, this research addresses the efficiency and practical implementation aspects of emotion-

aware music recommendation, shedding light on potential challenges and solutions.

Singh, Singh, and Sharma (2023) [6] brought forth the Facial Emotion-Based Automatic Music Recommender System, merging the insights from ABES Engineering College and IMS Engineering College. Their work introduces automation into the recommendation process, emphasizing the role of facial expressions in facilitating a seamless and automated music recommendation experience.

In summary, the literature review underscores the diverse approaches taken by researchers in leveraging facial expressions for emotion-aware music recommendations. These studies collectively contribute to the conceptualization and development of the Sentimental Analysis on Social Media, enriching the project with valuable insights and methodologies.

2.1 Existing System

The existing landscape of music recommendation systems has seen remarkable advancements with a focus on emotion-aware and facial expression-based approaches. The studies conducted by Annam, Bodapati, and Konda (2024) lay the foundation for a Transfer Learning Approach Using Facial Expressions to make music recommendations aligned with users' emotional states. This approach signifies a departure from conventional methods and emphasizes the significance of understanding user emotions through facial cues.

Sharath P., Senthil Kumar G., and Vishnu Boj K.S. (2023) contribute to the existing system by proposing a Music Recommendation System Using Facial Emotions. This system, grounded in computational intelligence, aligns music preferences with facial expressions, ensuring personalized and emotionally resonant recommendations.

Agrawal, Sharma, Fulzele, Bhasarkar, and Pande (2023) extend the existing system by introducing a Mood-based Song Recommendation System Using CNN. Their incorporation of convolutional neural networks enhances the precision of mood-based music recommendations, providing a nuanced understanding of users' emotional states.

Bokhare and Kothari (2023) explore new dimensions by introducing an Emotion Detection-Based Video Recommendation System Using Machine Learning and Deep Learning Framework. This expansion from music to video recommendations broadens the scope of emotion-aware technologies, catering to diverse

user preferences.

Kumar, Nandi, Roy, and Sarkar's (2023) work on the Design and Implementation of AI-Based Efficient Emotion Detection and Music Recommendation System addresses efficiency and practical implementation aspects, contributing valuable insights to enhance the existing system.

Singh, Singh, and Sharma (2023) further diversify the existing system by introducing the Facial Emotion-Based Automatic Music Recommender System, emphasizing automation in the recommendation process. This brings efficiency and seamlessness to the user experience, automating music recommendations based on facial expressions.

Together, these studies collectively enrich the existing system with innovative methodologies, extending the horizons of emotion-aware music and video recommendation systems. The Sentimental Analysis on Social Media, building upon these advancements, aims to provide an enhanced and holistic user experience by integrating state-of-the-art techniques in facial expression recognition and emotion-aware music recommendations.

2.2 Literature Review Summary

Year and Citation	Article/ Author	Tools/ Software	Technique	Source	Evaluation Parameter
2024 [1]	Annam, Bodapati, Konda	Research Study	Transfer Learning Using Sentimental Analysis	Lecture Notes in Networks and Systems, ICDIS 2023	Emotion Recognition Accuracy, Music Recommendation Performance
2023 [2]	Sharath P., Senthil Kumar G., Vishnu Boj K.S.	Research Study	Computational Intelligence, Machine Learning	Department of Computational Intelligence, IRCICD 2022	Personalization, Emotional Resonance in Music Recommendations
2023 [3]	Agrawal, Sharma, Fulzele, Bhasarkar, Pande	CNN	Mood-based Song Recommendation	School of Computer Engineering, ICCCNT 2023	Mood-based Social media Recommendation Accuracy
2023 [4]	Bokhare, Kothari	Machine Learning, Deep Learning Framework	Emotion Detection-Based Video Recommendation	Symbiosis Institute of Computer Studies and Research, ICSEIET 2023	Video Recommendation Accuracy, Emotion Detection Performance
2023 [5]	Kumar, Nandi, Roy, Sarkar	AI-Based Efficient Emotion Detection	Sentimental Analysis Recommendation System Design and Implementation	Reva University, AIS2C2 2022	Efficiency, Practical Implementation Aspects
2023 [6]	Singh, Singh, Sharma	Research Study	Facial Emotion-Based Automatic Music Recommender	ABES Engineering College, IMS Engineering College, ICCCNT 2023	Automation Efficiency, Seamless User Experience

2.3 Proposed System

The proposed Sentimental Analysis on Social Media represents a leap forward in the domain of emotion-aware music recommendation systems, synthesizing insights from the existing literature. Drawing inspiration from the studies conducted by Annam, Bodapati, and Konda (2024), the proposed system adopts a Transfer Learning Approach Using Facial Expressions. This approach aims to refine the understanding of user emotions through facial cues and leverage transfer learning to enhance the accuracy of music recommendations based on emotional states.

Sharath P., Senthil Kumar G., and Vishnu Boj K.S.'s (2023) work on a Music Recommendation System Using Facial Emotions is foundational to the proposed system. Building upon their insights in computational intelligence, the proposed system integrates facial expressions to provide personalized and emotionally resonant music recommendations, ensuring a more engaging user experience.

Agrawal, Sharma, Fulzele, Bhasarkar, and Pande's (2023) Mood-based Song Recommendation System Using CNN contributes to the proposed system by incorporating convolutional neural networks for enhanced mood-based music recommendations. This integration provides a sophisticated understanding of users' emotional states, allowing for more nuanced and contextually relevant music suggestions.

The proposed system also integrates insights from Bokhare and Kothari's (2023) Emotion Detection-Based Video Recommendation System Using Machine Learning and Deep Learning Framework, expanding the scope to video recommendations. This extension enriches the proposed system by catering to diverse user preferences and aligning with modern multimedia consumption trends.

Kumar, Nandi, Roy, and Sarkar's (2023) work on the Design and Implementation of AI-Based Efficient Emotion Detection and Music Recommendation System contributes practical insights to the proposed system. This inclusion ensures that the proposed system not only excels in emotion detection but also addresses efficiency considerations for real-world implementation.

The Facial Emotion-Based Automatic Music Recommender System by Singh, Singh, and Sharma (2023) inspires the proposed system's emphasis on automation. By automating the music recommendation process based on facial expressions, the proposed system strives to offer a seamless and efficient user experience.

Chapter 3: DESIGN FLOW/ PROCESS

3.1 Problem Formulation

In navigating the vast expanse of the digital music landscape, users often encounter a significant gap between their emotional states and the recommendations offered by conventional music streaming platforms. This disparity is rooted in the static nature of existing recommendation systems, which often fail to adapt to the dynamic and nuanced aspects of human emotions. Social media seeks to address this fundamental problem by formulating a set of challenges and goals that articulate the need for a more personalized and emotionally attuned music discovery experience.

1. Inadequate Representation of User Emotions:

Traditional music recommendation systems predominantly rely on historical data, explicit preferences, and genre categorizations to generate suggestions. However, these systems fall short in capturing the intricacies of user emotions at any given moment. Users are dynamic beings, and their emotional states evolve continuously. The challenge lies in formulating a system that not only recognizes these subtle changes but interprets them in real-time, offering music recommendations that align with the user's immediate emotional context.

2. Limitations of Historical Data:

The reliance on historical user data poses another challenge in the context of music recommendations. While historical data provides valuable insights into a user's long-term preferences, it may not accurately reflect their current mood or emotional state. The formulation of Social media addresses this limitation by aiming to seamlessly integrate real-time emotional insights with historical data, ensuring that recommendations are not solely based on past preferences but dynamically adapt to the user's evolving emotions.

3. Impersonalized Recommendations:

The impersonal nature of current music recommendation systems stems from a lack of real-time adaptability and understanding of the user's emotions. Users often feel disconnected from recommendations that do not resonate with their immediate feelings or experiences. The challenge, therefore, is to formulate a system that incorporates advanced technologies to recognize and interpret sentiments in real-time, creating a more personalized and engaging music discovery journey.

4. The Paradox of Choice:

The abundance of musical content available on streaming platforms contributes to the paradox of choice, wherein users may feel overwhelmed and struggle to discover content that aligns with their current emotional states. Formulating a solution to this challenge involves designing a recommendation system that doesn't just offer an extensive catalogue but actively curates personalized playlists, reducing the cognitive load on users and enhancing their overall satisfaction with the platform.

5. Dynamic Nature of User Emotions:

Human emotions are dynamic and multifaceted, presenting a challenge in developing a recommendation system that accurately interprets and adapts to this complexity. A system that fails to understand the subtle shifts in a user's emotional landscape may provide recommendations that feel disconnected or irrelevant. The formulation of Sentiments addresses this challenge by integrating AIML algorithms, providing a nuanced understanding of facial expressions and emotions, and ensuring that the system dynamically responds to the user's ever-changing emotional states.

6. Lack of User Engagement:

Conventional music streaming platforms often struggle with user engagement due to the static and impersonal nature of their recommendation systems. Users may lose interest or disengage when confronted with recommendations that do not resonate with their current emotions. The challenge is to formulate a system that not only captures the user's attention but actively involves them in the music discovery process. Sentiments addresses this challenge by introducing an interactive web-based interface with a camera component for facial emotion recognition, fostering a more engaging and participatory user experience.

7. Cross-Platform Compatibility:

As users interact with digital content across a myriad of devices, achieving cross-platform compatibility becomes crucial. The challenge lies in formulating a web-based interface that seamlessly adapts to different devices and browsers, ensuring a consistent and accessible user experience. Sentimental Analysis addresses this challenge by prioritizing responsive design and compatibility across desktops, tablets, and smartphones.

3.2 Objectives:

The objectives of sentiment analysis in social media are multifaceted, reflecting the diverse applications and benefits it offers to individuals, businesses, researchers, and society as a whole. Here are some key objectives of sentiment analysis in the context of social media:

1. Real-Time Emotion Recognition:

The primary objective of Sentimental Analysis is to implement a cutting-edge facial emotion recognition system. This technology will analyze users' facial expressions in real-time, enabling the system to understand and categorize their emotions dynamically. The aim is to create a responsive and adaptive system that captures the nuances of users' changing emotional states.

2. AIML Integration for Nuanced Emotion Decoding:

To achieve a nuanced understanding of facial expressions and emotions Setimental Analysis incorporates AIML algorithms. These algorithms will be integrated into the system to decode and categorize emotions accurately. The objective is to move beyond simplistic emotion labels and provide a more sophisticated analysis that aligns with the complexity of human emotions.

3. MERN Stack Architecture:

Utilizing the MERN stack, Sentimental Analysis aims to establish a robust and scalable architecture. MongoDB will manage user profiles, music preferences, and real-time emotional data. Express.js will handle secure server-side logic, while React and Node.js will contribute to the creation of a dynamic and interactive user interface. The objective is to build a versatile and efficient infrastructure that supports seamless communication between the server and client.

4. Personalized Music Recommendations:

The project strives to develop a dynamic sentimental analysis engine that goes beyond traditional algorithms. By integrating real-time emotional insights with historical user data, the engine will generate personalized song suggestions aligned with the user's immediate emotional context. The objective is to create a recommendation system that evolves with the user's changing preferences and emotions.

5. User Engagement through Interactive UI:

Social Media aims to enhance user engagement by designing an intuitive and visually appealing web-based interface. The inclusion of a camera component for facial emotion recognition adds an interactive dimension to the user experience, allowing users to actively participate in the music discovery process through their emotions. The objective is to foster a deeper connection between users and the platform.

6. Cross-Platform Compatibility:

Recognizing the diverse ways users access digital content, the project places a significant emphasis on achieving cross-platform compatibility. The web-based interface will be designed to seamlessly adapt to different devices and browsers, ensuring a consistent and accessible user experience across desktops, tablets, and smartphones. The objective is to provide users with a flexible and user-friendly platform that aligns with their preferred devices.

3.3 Flow/ Process of Project:

The methodology employed in the development of the "Social Media Analysis: An Exploration of AI-Driven Music Curation Based on Facial Analysis" encompasses a multifaceted approach, integrating advanced techniques in artificial intelligence (AI) and facial analysis to realize the project objectives. This section delineates the key components of the methodology, encompassing data collection, preprocessing, feature extraction, algorithm design, and evaluation.

3.3.1 Data Collection and Preprocessing

Central to the success of the Social Media Sentimental Analysis is the acquisition of high-quality data encompassing facial expressions and music preferences. To this end, a diverse dataset comprising facial images annotated with corresponding emotional labels is curated from publicly available repositories and in-house recordings. The facial images are preprocessed to standardize resolution, lighting conditions, and facial landmarks, ensuring consistency and reliability in subsequent analyses. Similarly, music data encompassing metadata and audio features are collected from streaming platforms and curated databases, facilitating personalized music recommendations.

3.3.2 Feature Extraction

Facial analysis forms the cornerstone of the Social Media sentimental Analysis, with the extraction of discriminative features from facial images serving as the basis for emotion recognition. Facial landmarks are detected using state-of-the-art computer vision algorithms, enabling the extraction of geometric and appearance-based features. Additionally, deep learning techniques such as convolutional neural networks (CNNs) [9] are employed to extract abstract representations of facial expressions, capturing spatial and temporal dependencies in the data.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c)2^p$$

where (x_c, y_c) = central pixel

i_c = intensity of the central pixel

i_p = intensity of the neighboring pixel

s(x) = thresholding function (1, if x>=0, 0 otherwise)

Concurrently, music features encompassing acoustic, temporal, and semantic attributes are extracted using signal processing and machine learning techniques, facilitating the characterization of music content.

Date	Production
1962-01	589
1962-02	561
1962-03	640
1962-04	656
1962-05	727
1962-06	697
1962-07	640
1962-08	599
1962-09	568
1962-10	577
1962-11	553
1962-12	582
1963-01	600
1963-02	566
1963-03	653
1963-04	673
1963-05	742

Fig. 1. Dataset

3.3.3 Algorithm Design

The algorithmic framework of the sentimental analysis encompasses several interconnected modules, each dedicated to specific tasks such as emotion recognition, music recommendation, and user feedback integration. Emotion recognition models leveraging deep learning architectures are trained on the annotated facial image dataset, enabling real-time inference of users' emotional states from live camera feeds. Concurrently, collaborative filtering algorithms and content-based recommendation techniques are employed to generate personalized music playlists tailored to users' emotional profiles and music preferences. The integration of hybrid recommendation approaches augments the robustness and diversity of music recommendations, enhancing user satisfaction and engagement.

$$output(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} Kernal\ (m,n) * Image(i+m, j+n) + Bias$$

where output (i, j) = Output at position (i, j)

kernel = convolutional filter

image = Input Image

bias = Bias Term

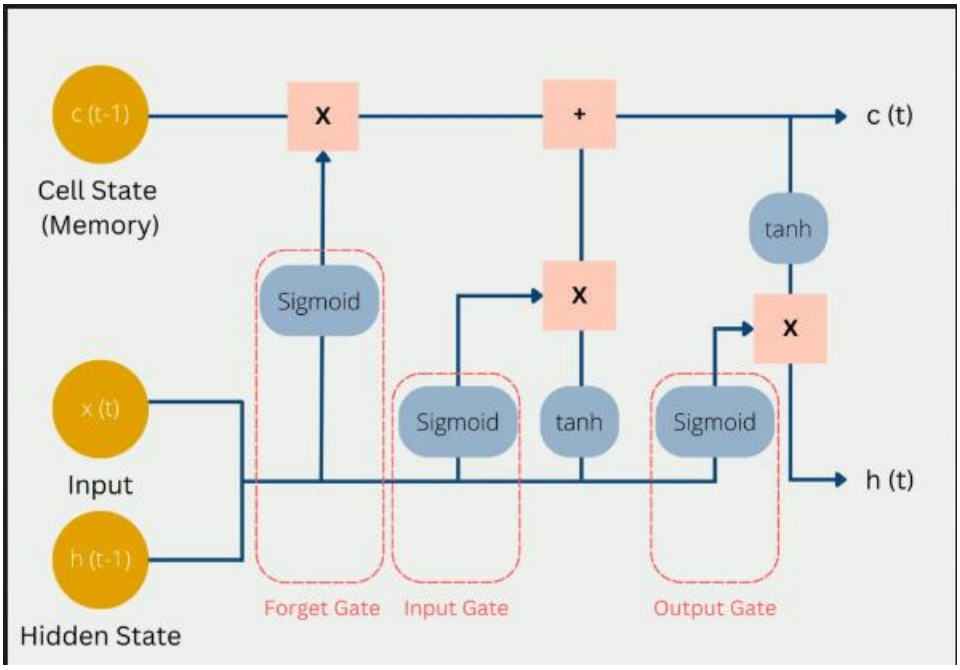


Fig. 2.Architecture of LSTM model

3.3.4 System Implementation

The Sentimental analysis is implemented using the machine learning algorithms10] stack, facilitating the development of a scalable and responsive web-based interface. The frontend interface incorporates

interactive features for user interaction, including real-time facial analysis, playlist visualization, and feedback submission. Simultaneously, the backend infrastructure supports seamless data integration, model inference, and recommendation generation, ensuring efficient and reliable performance across diverse user environments.

3.3.5 Evaluation and Validation

The efficacy and performance of the sentimental Analysis are rigorously evaluated using a combination of quantitative and qualitative metrics. We use the Precision, Recall and accuracy measures to get the entirety of how well our face recognition model is actually working after training.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Accuracy = \frac{Number\ of\ correctly\ classified\ samples}{Total\ Number\ of\ Samples} * 100$$

Figure 3 shows the flowchart of the entire Sentimental Analysis on Social Media Project based on the fact that everything is fluently working. The project is based on a simple flow that it will be present in the home screen where user can click on the button to open the camera and then face will be detected and songs will be recommended.

On the other hand, he/she can use the more basic functionalities such as creating and adding playlists, listening songs and login and sign up to the web application.

Then, he/ she can experience a smooth UI/UX based on the way that everything is running smoothly across the entire application.

Emotion recognition accuracy, music recommendation relevance, user satisfaction, and engagement metrics are quantitatively assessed through controlled user studies and online surveys. Additionally, qualitative feedback and user testimonials are solicited to gauge the subjective perceptions and experiences of users interacting with the system. The validation process encompasses comparative analyses against existing music recommendation systems and benchmark datasets, providing insights into the uniqueness and effectiveness of the proposed approach.

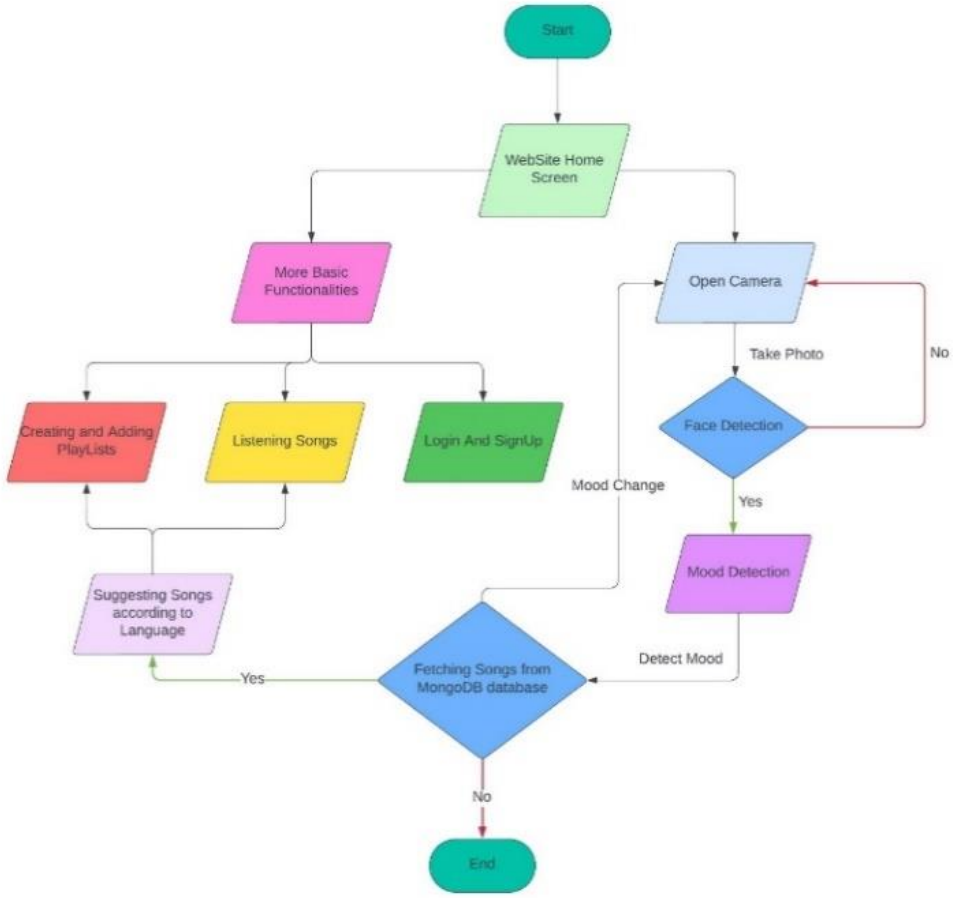


Fig. 3. Flowchart of Entire Project

3.6 Ethical Considerations

Throughout the development and deployment of the Sentimental Analysis on Social Media, ethical considerations pertaining to user privacy, data security, and algorithmic transparency are paramount. Strict adherence to data protection [13] regulations, informed consent protocols, and anonymization techniques ensures the confidentiality and integrity of user data. Additionally, measures are implemented to mitigate bias, fairness, and interpretability concerns inherent in AI-driven systems, fostering trust and accountability in the technology.

In summary, the methodology outlined herein delineates a comprehensive framework for the development and deployment of the Sentimental Analysis on Social Media, leveraging AI-driven music curation and facial analysis to create personalized and emotionally resonant music recommendations.

3.4 Experimental Setup

The experimental setup for Sentimental Analysis on Social Media is designed to rigorously evaluate and validate the effectiveness of the real-time emotion recognition and personalized music recommendation system. The setup encompasses various components, including hardware, software, and user interaction scenarios, to ensure a comprehensive assessment of the platform's performance.

1. Hardware Requirements:

The real-time facial emotion recognition component necessitates a device with a camera, such as laptops, desktops, tablets, or smartphones. The quality of the camera is crucial to capture facial expressions accurately. The experimental setup accommodates a diverse range of devices to assess the cross-platform compatibility of Sentimental Analysis on Social Media.

2. Software Components:

The Sentimental Analysis on Social Media system is built on the Machine Learning algorithms. MongoDB serves as the backend database to store user profiles, music preferences, and real-time emotional data. Express.js handles secure server-side logic, react is utilized for dynamic user interfaces, and Node.js facilitates seamless communication between the server and client. Open-source computer vision libraries and AIML algorithms are integrated for facial emotion recognition.

3. Real-Time Emotion Recognition Algorithm:

The experimental setup involves implementing and fine-tuning the real-time facial emotion recognition algorithm. This algorithm utilizes computer vision techniques to analyze facial expressions captured by the camera in real-time. The accuracy and efficiency of emotion detection are continuously refined during the experimental phase.

4. Sentiment Analysis Recommendation Engine:

The experimental setup includes the development and optimization of the music recommendation engine. This engine combines real-time emotional insights with historical user data to generate personalized song suggestions. The algorithm's effectiveness in adapting to dynamic emotional states and providing relevant recommendations is a focal point of the experiments.

5. User Interaction Scenarios:

User interaction scenarios are carefully crafted to simulate diverse emotional states and engagement levels. Participants are encouraged to express a range of emotions while interacting with the Sentimental Analysis on Social Media platform. This includes scenarios such as joy, sadness, excitement, and tranquility. The goal is to evaluate how accurately the system adapts to and recommends music based on varying emotional cues.

6. Performance Metrics:

To quantitatively measure the system's performance, key metrics are defined. These metrics include the accuracy of facial emotion recognition, the responsiveness of the recommendation engine, and user satisfaction ratings based on the relevance of suggested music. A subjective assessment of the overall user experience, gathered through user feedback, is also considered a crucial performance metric.

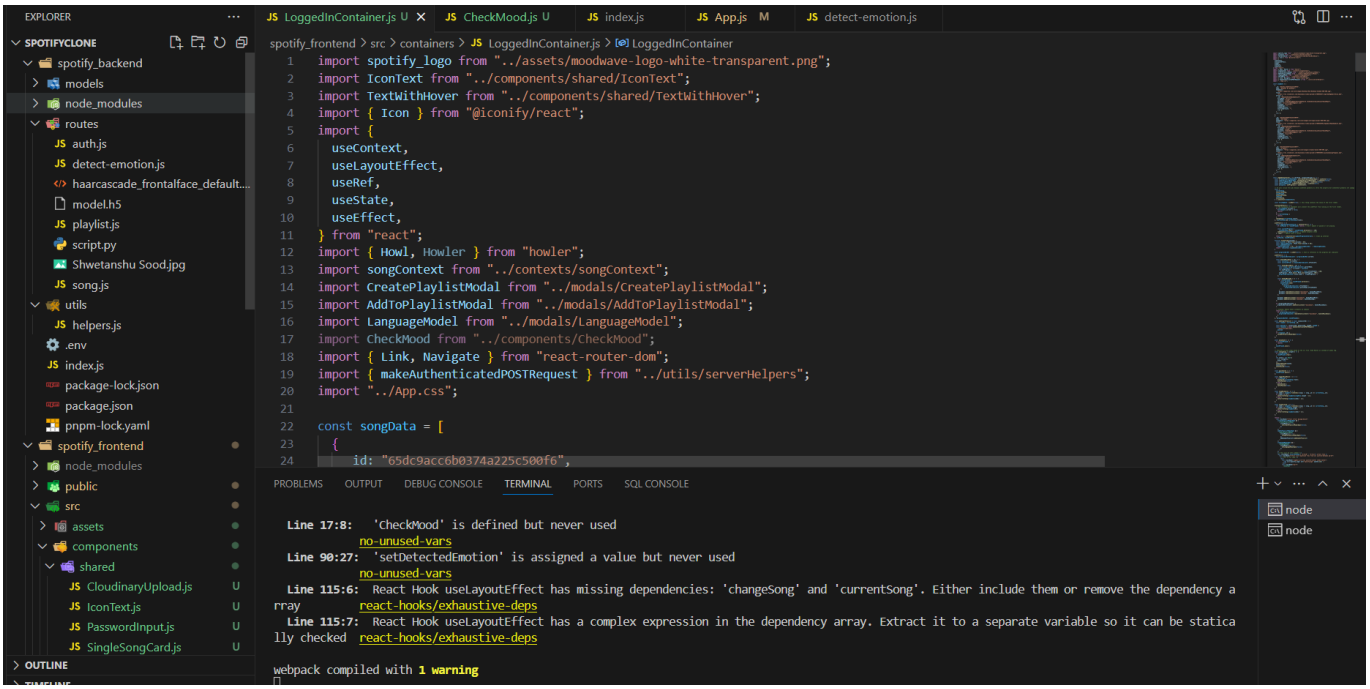
7. User Feedback Mechanism:

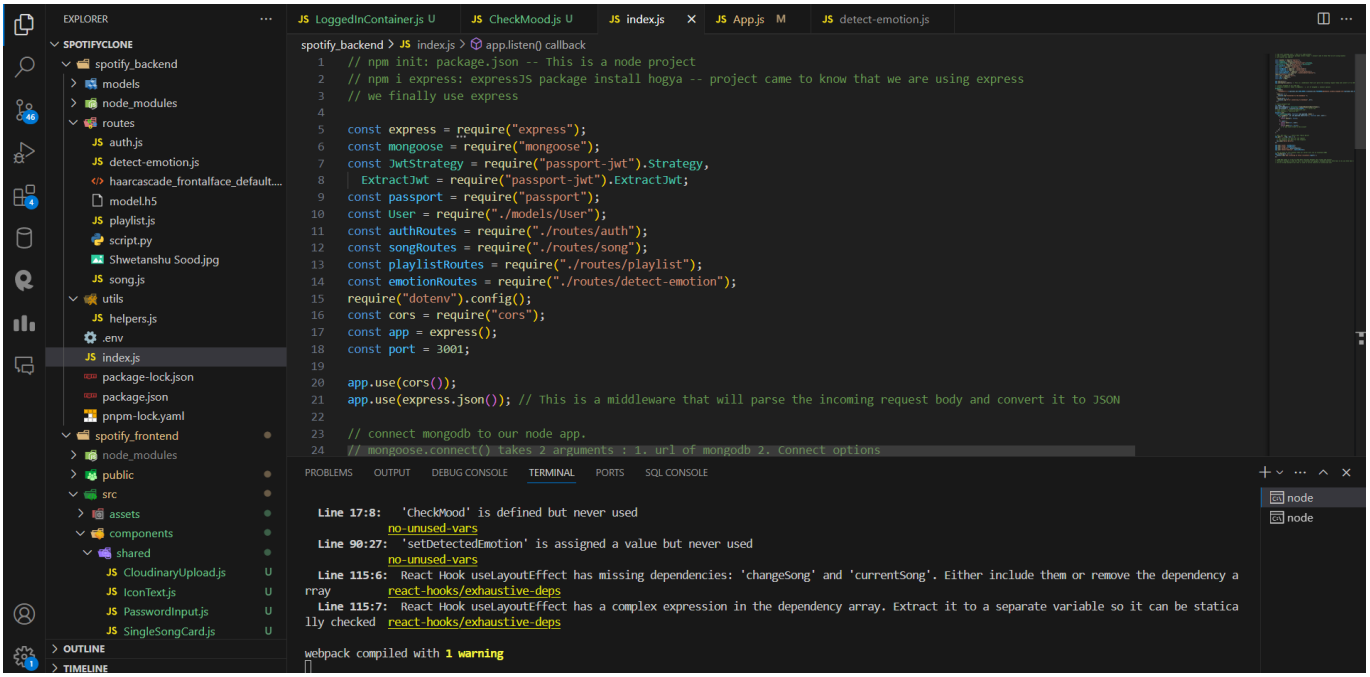
The experimental setup incorporates a systematic user feedback mechanism. Participants are encouraged to provide feedback on the accuracy of emotion recognition, the relevance of music suggestions, and the overall usability of the interface. Qualitative feedback is collected through surveys and interviews to gain insights into the user's emotional engagement with the platform.

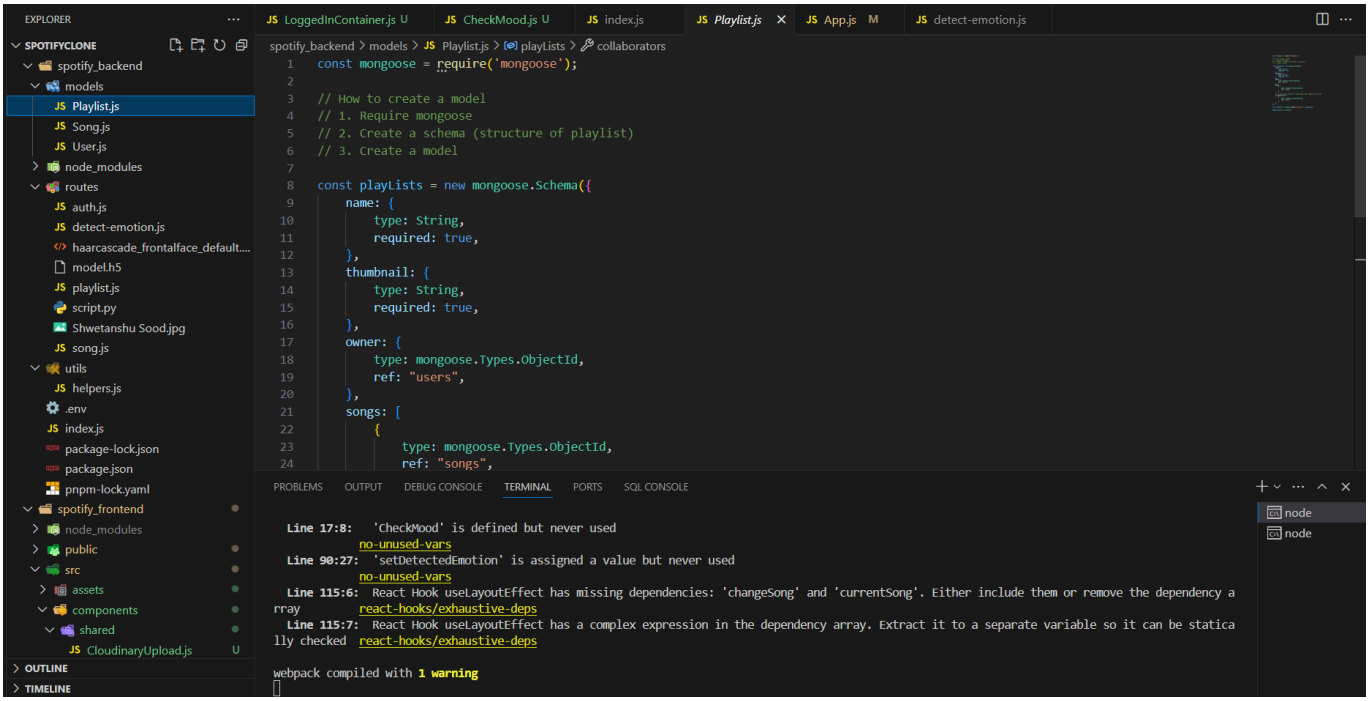
8. Data Security and Privacy Measures:

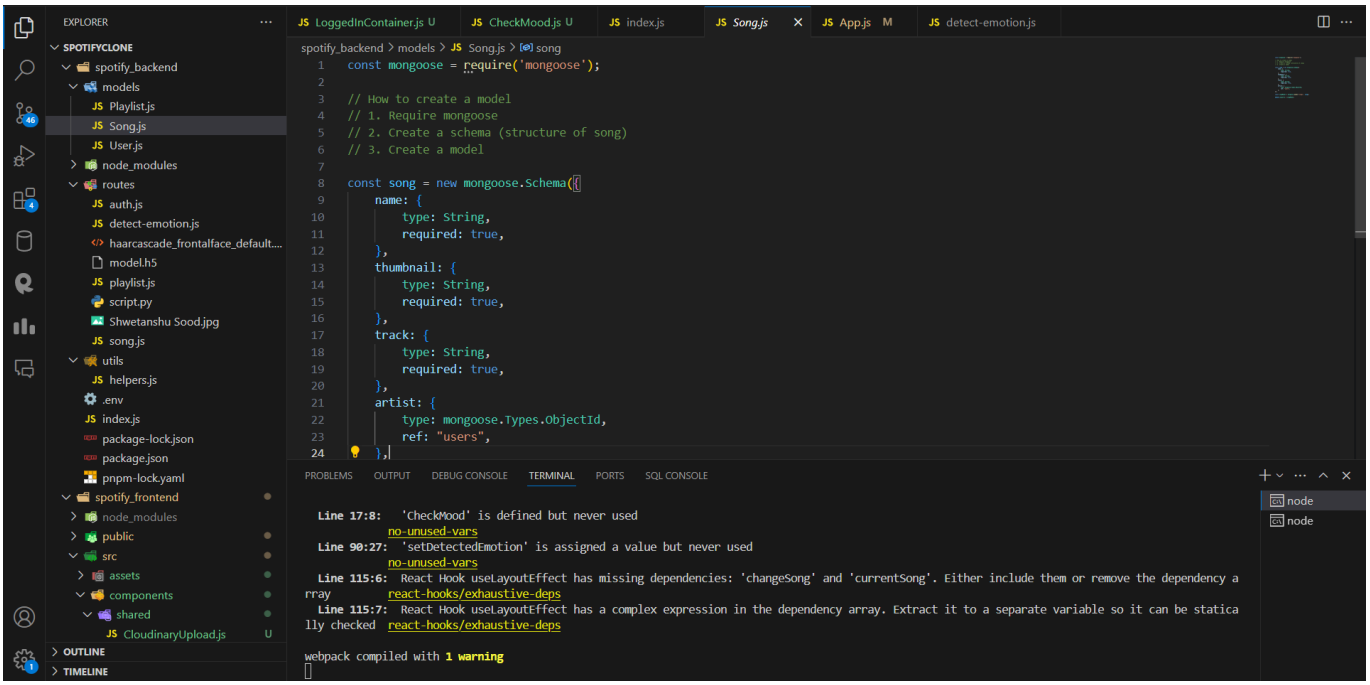
Given the nature of real-time facial emotion recognition, robust measures are implemented to ensure data security and user privacy. Facial data is anonymized and stored securely, adhering to ethical standards and regulations. The experimental setup includes rigorous testing of these security measures to guarantee the confidentiality and integrity of user information.

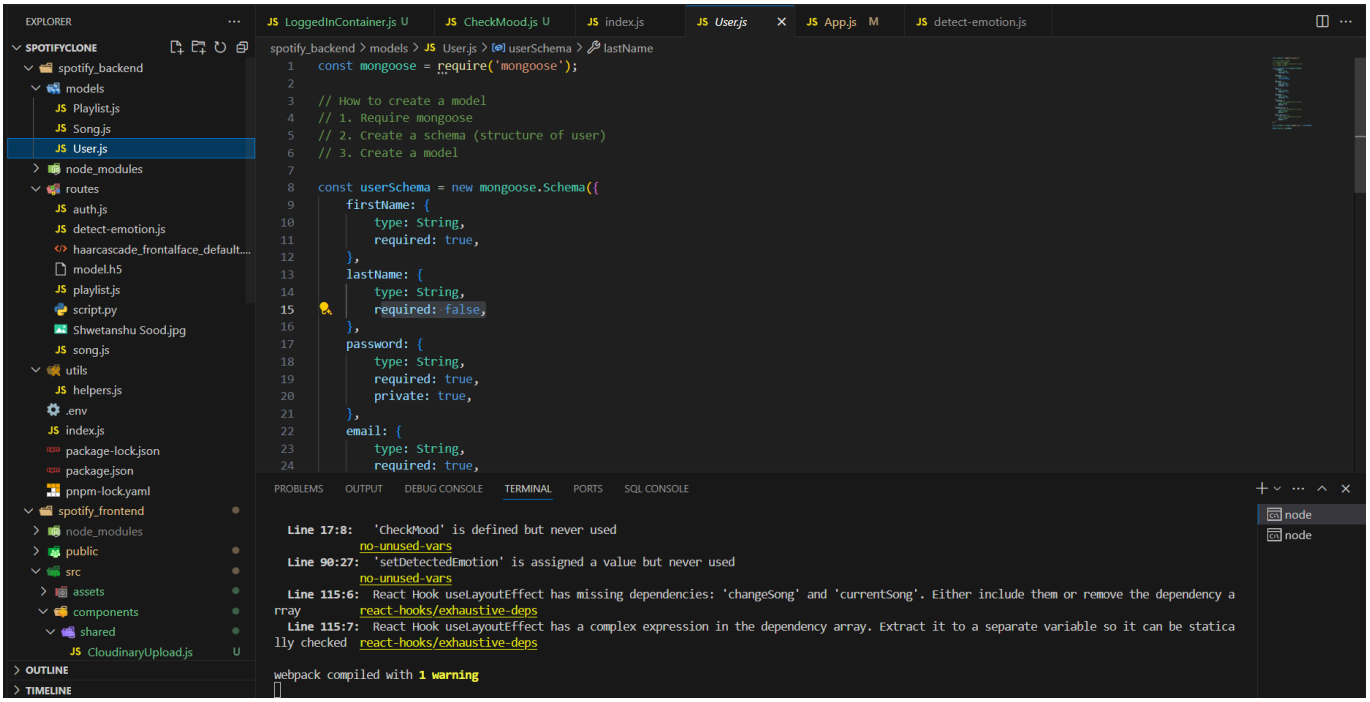
Chapter 4: SNAPSHOTS OF IMPLEMENTATION

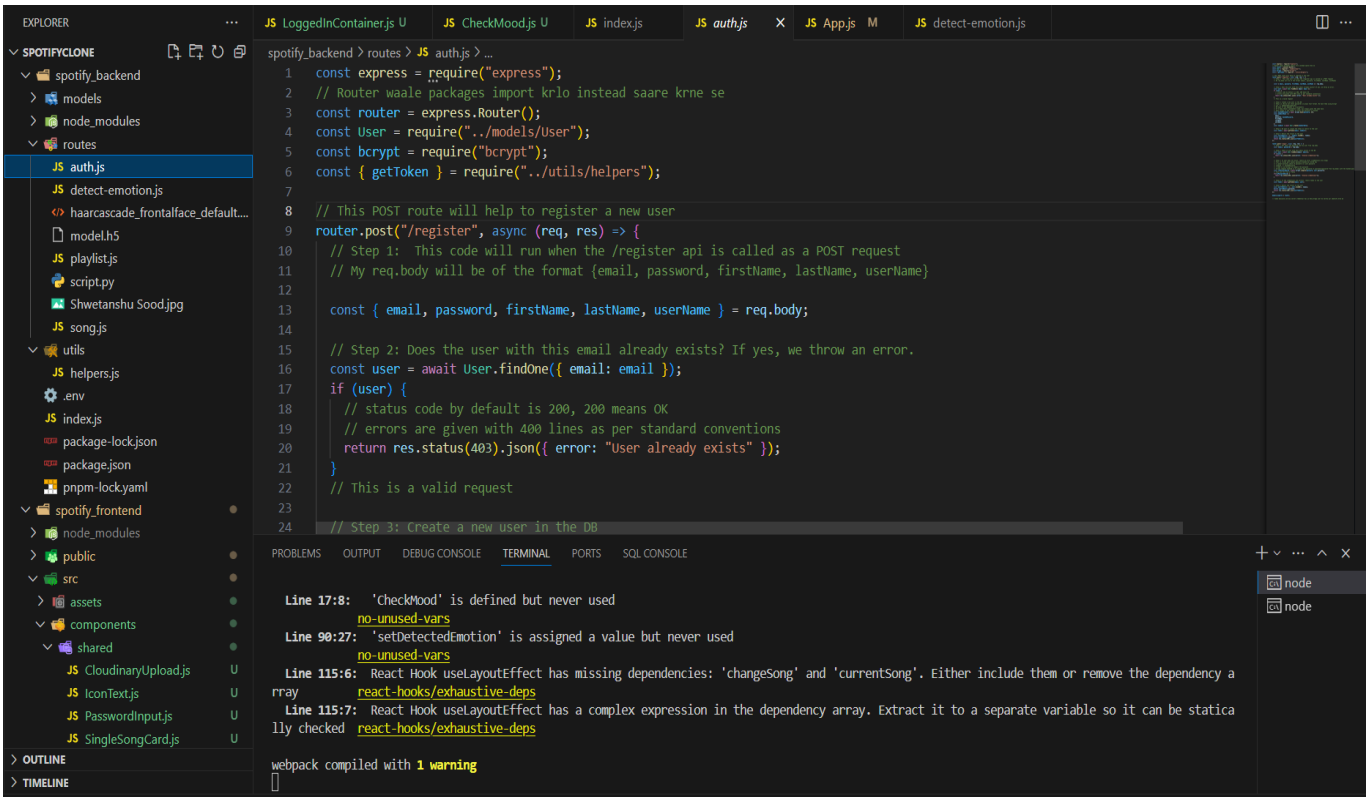


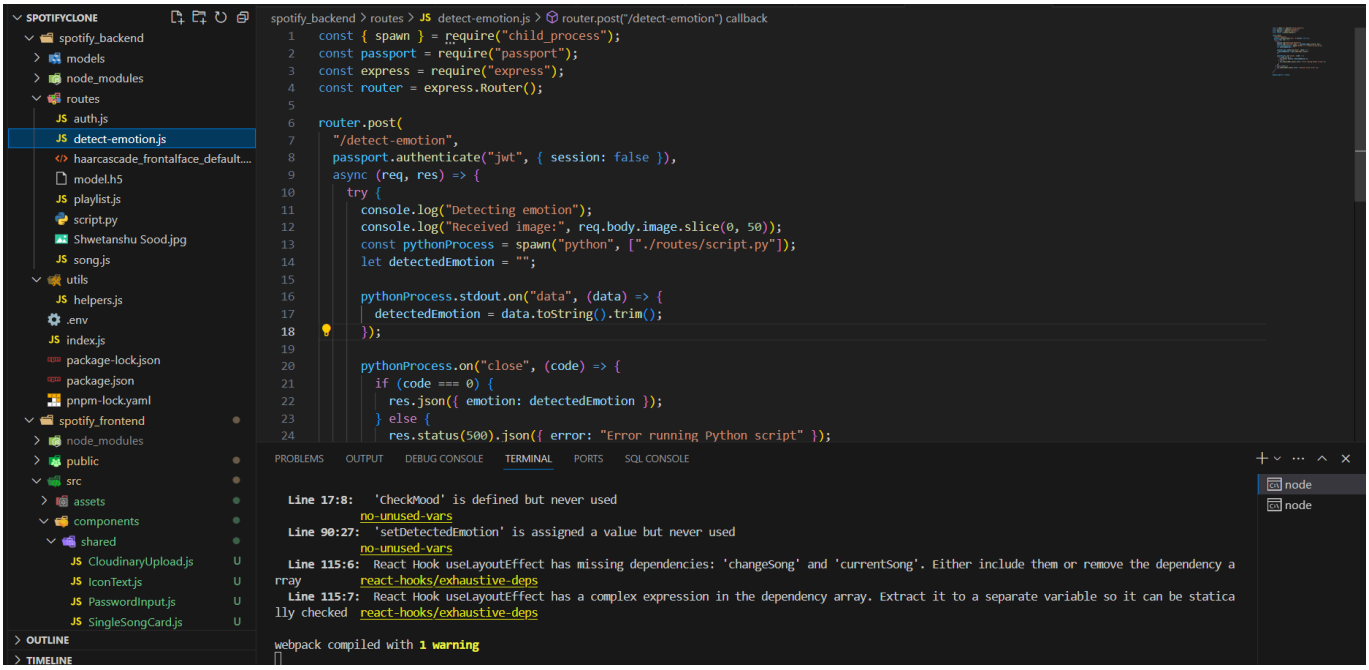


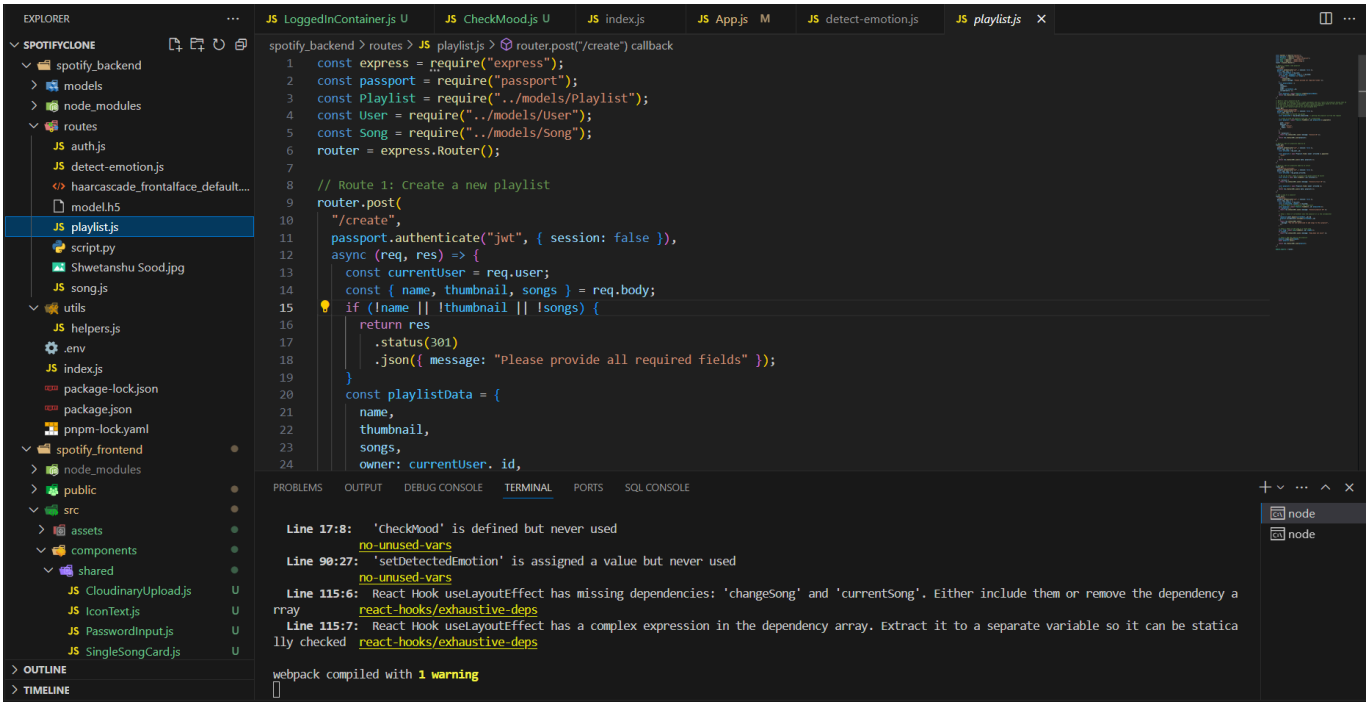

EXPLORER
SPOTIFYCLONE
spotify_backend
models
Playlist.js
Song.js
User.js
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
Playlist.js
App.js M
detect-emotion.js
spotify_backend > models > Playlist.js > playLists > collaborators
const mongoose = require('mongoose');
// How to create a model
// 1. Require mongoose
// 2. Create a schema (structure of playlist)
// 3. Create a model
const playLists = new mongoose.Schema({
name: {
type: String,
required: true,
},
thumbnail: {
type: String,
required: true,
},
owner: {
type: mongoose.Types.ObjectId,
ref: "users",
},
songs: [
{
type: mongoose.Types.ObjectId,
ref: "songs",
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

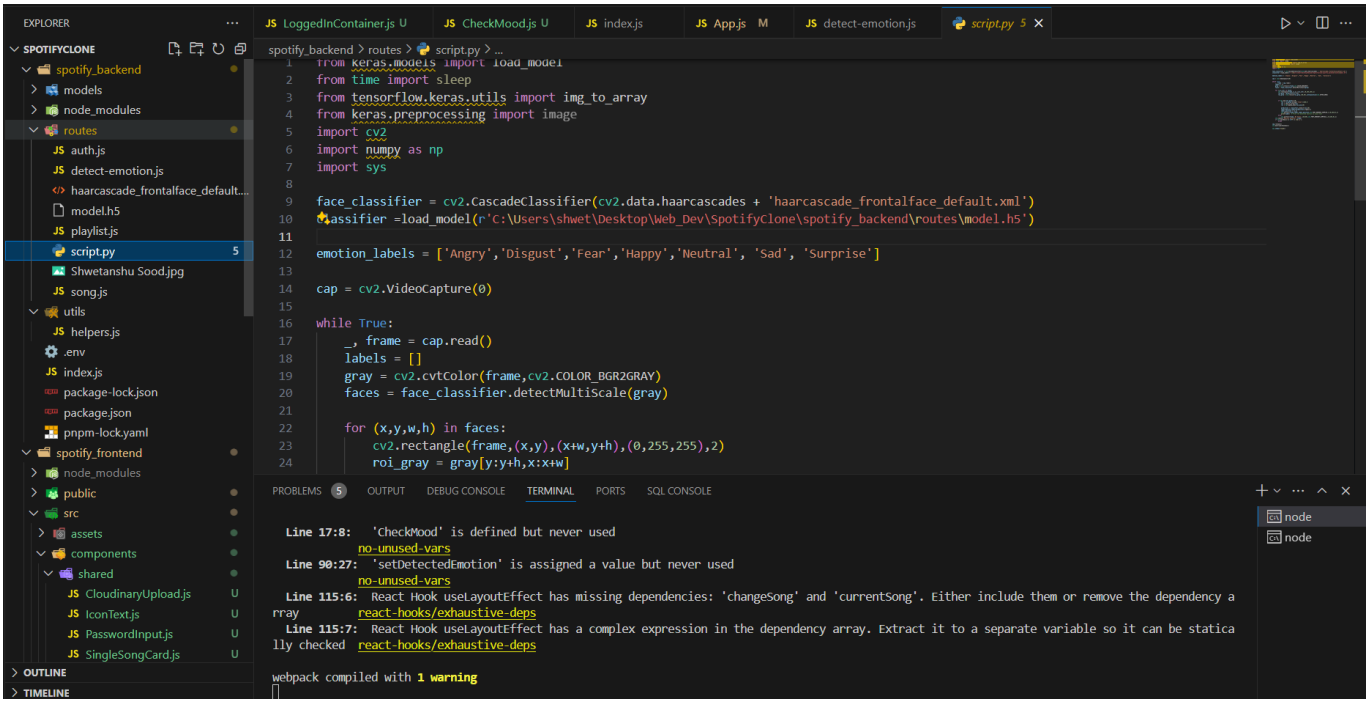

EXPLORER
SPOTIFYCLONE
spotify_backend
models
Playlist.js
Song.js
User.js
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
Song.js
App.js M
detect-emotion.js
spotify_backend > models > Song.js > song
const mongoose = require('mongoose');
// How to create a model
// 1. Require mongoose
// 2. Create a schema (structure of song)
// 3. Create a model
const song = new mongoose.Schema({
name: {
type: String,
required: true,
},
thumbnail: {
type: String,
required: true,
},
track: {
type: String,
required: true,
},
artist: {
type: mongoose.Types.ObjectId,
ref: "users",
},
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

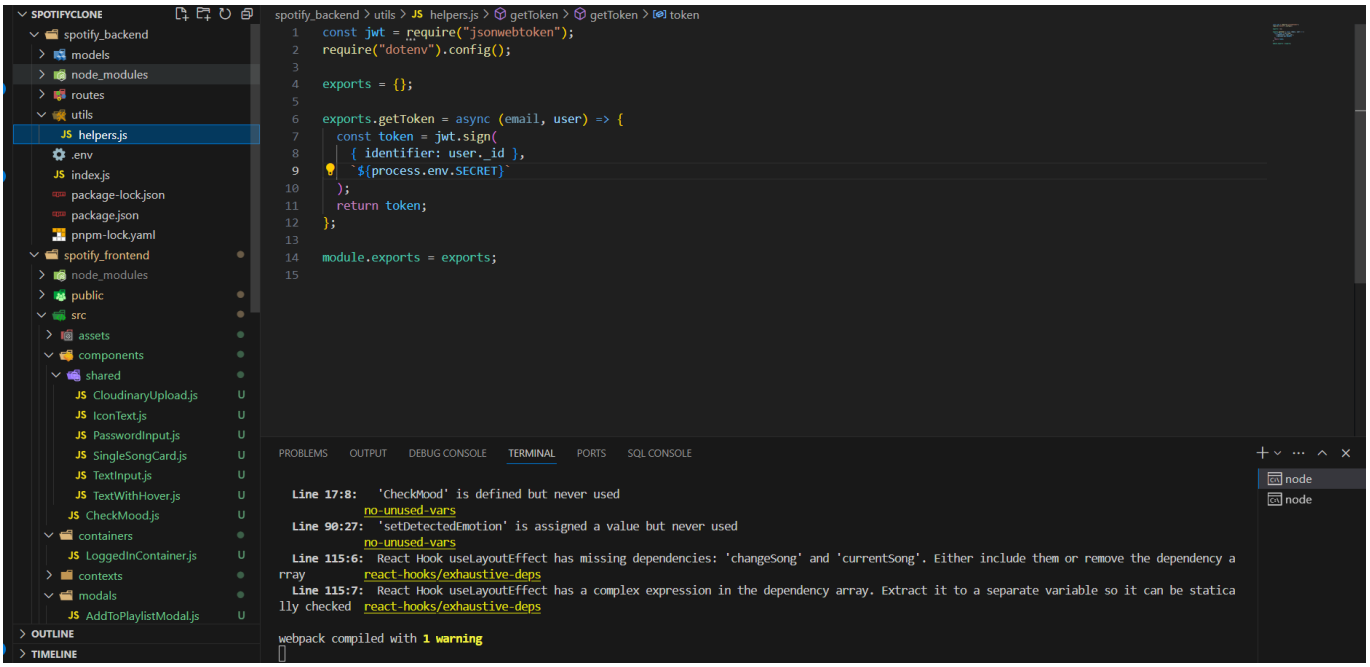

EXPLORER
SPOTIFYCLONE
spotify_backend
models
Playlist.js
Song.js
User.js
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
User.js
App.js M
detect-emotion.js
spotify_backend > models > User.js > userSchema > lastName
const mongoose = require('mongoose');
// How to create a model
// 1. Require mongoose
// 2. Create a schema (structure of user)
// 3. Create a model
const userSchema = new mongoose.Schema({
firstName: {
type: String,
required: true,
},
lastName: {
type: String,
required: false,
},
password: {
type: String,
required: true,
private: true,
},
email: {
type: String,
required: true,
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

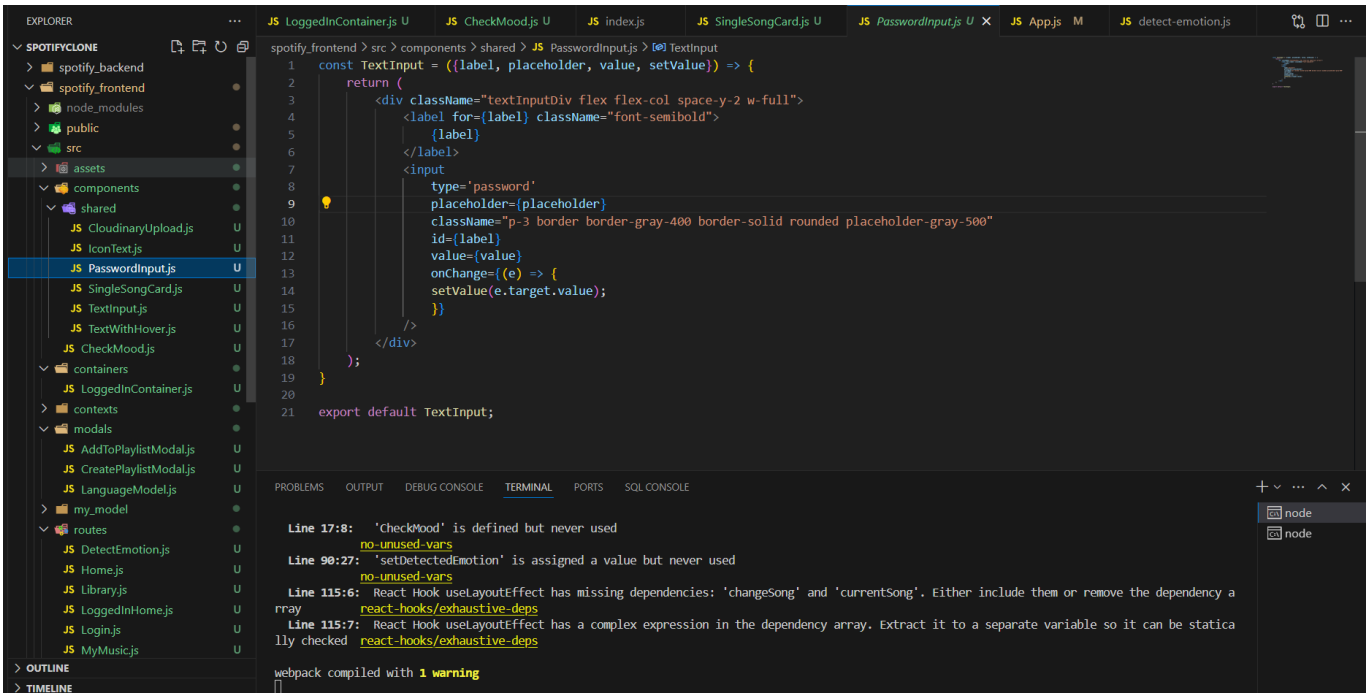

EXPLORER
SPOTIFYCLONE
spotify_backend
models
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
auth.js
App.js M
detect-emotion.js
spotify_backend > routes > auth.js > ...
const express = require("express");
// Router waale packages import krlo instead saare krne se
const router = express.Router();
const User = require("../models/User");
const bcrypt = require("bcrypt");
const { getToken } = require("../utils/helpers");
// This POST route will help to register a new user
router.post("/register", async (req, res) => {
// Step 1: This code will run when the /register api is called as a POST request
// My req.body will be of the format {email, password, firstName, lastName, userName}
const { email, password, firstName, lastName, userName } = req.body;
// Step 2: Does the user with this email already exists? If yes, we throw an error.
const user = await User.findOne({ email: email });
if (user) {
// status code by default is 200, 200 means OK
// errors are given with 400 lines as per standard conventions
return res.status(403).json({ error: "User already exists" });
}
// This is a valid request
// Step 3: Create a new user in the DB
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


SPOTIFYCLONE
spotify_backend
models
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
OUTLINE
TIMELINE
spotify_backend > routes > JS detect-emotion.js > router.post("/detect-emotion") callback
const { spawn } = require("child_process");
const passport = require("passport");
const express = require("express");
const router = express.Router();
router.post(
"/detect-emotion",
passport.authenticate("jwt", { session: false }),
async (req, res) => {
try {
console.log("Detecting emotion");
console.log("Received image:", req.body.image.slice(0, 50));
const pythonProcess = spawn("python", ["./routes/script.py"]);
let detectedEmotion = "";
pythonProcess.stdout.on("data", (data) => {
detectedEmotion = data.toString().trim();
});
pythonProcess.on("close", (code) => {
if (code === 0) {
res.json({ emotion: detectedEmotion });
} else {
res.status(500).json({ error: "Error running Python script" });
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
LoggedInContainer.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
playlist.js
SPOTIFYCLONE
spotify_backend
models
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default....
model.h5
playlist.js
script.py
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
OUTLINE
TIMELINE
spotify_backend > routes > JS playlist.js > router.post("/create") callback
const express = require("express");
const passport = require("passport");
const Playlist = require("../models/Playlist");
const User = require("../models/User");
const Song = require("../models/Song");
router = express.Router();
// Route 1: Create a new playlist
router.post(
"/create",
passport.authenticate("jwt", { session: false }),
async (req, res) => {
const currentUser = req.user;
const { name, thumbnail, songs } = req.body;
if (!name || !thumbnail || !songs) {
return res
.status(301)
.json({ message: "Please provide all required fields" });
}
const playlistData = {
name,
thumbnail,
songs,
owner: currentUser._id,
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

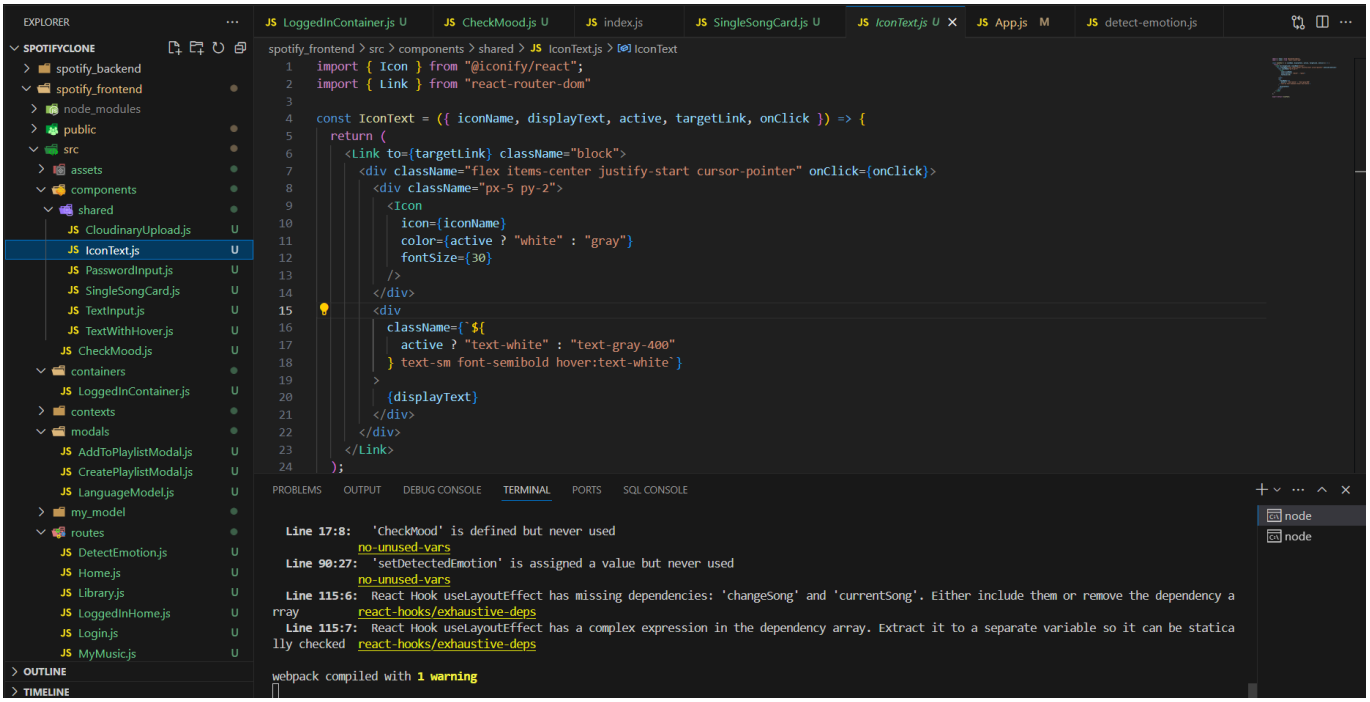

EXPLORER
SPOTIFYCLONE
spotify_backend
models
node_modules
routes
auth.js
detect-emotion.js
haarcascade_frontalface_default...
model.h5
playlist.js
script.py 5
Shwetanshu Sood.jpg
song.js
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
script.py 5
spotify_backend > routes > script.py > ...
from keras.models import load_model
from time import sleep
from tensorflow.keras.utils import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np
import sys
face_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
assifier =load_model(r'C:\Users\shwet\Desktop\Web_Dev\SpotifyClone\spotify_backend\routes\model.h5')
emotion_labels = ['Angry','Disgust','Fear','Happy','Neutral', 'Sad', 'Surprise']
cap = cv2.VideoCapture(0)
while True:
_, frame = cap.read()
labels = []
gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
faces = face_classifier.detectMultiScale(gray)
for (x,y,w,h) in faces:
cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,255),2)
roi_gray = gray[y:y+h,x:x+w]
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


SPOTIFYCLONE
spotify_backend
models
node_modules
routes
utils
helpers.js
.env
index.js
package-lock.json
package.json
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
OUTLINE
TIMELINE
spotify_backend > utils > helpers.js > getToken > getToken > token
const jwt = require("jsonwebtoken");
require("dotenv").config();
exports = {};
exports.getToken = async (email, user) => {
const token = jwt.sign(
{ identifier: user._id },
`${process.env.SECRET}`
);
return token;
};
module.exports = exports;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_backend
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
SingleSongCard.js U
PasswordInput.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > PasswordInput.js > TextInput
const TextInput = ({label, placeholder, value, setValue}) => {
return (

<label for={label} className="font-semibold">
{label}
</label>
<input
type='password'
placeholder={placeholder}
className="p-3 border border-gray-400 border-solid rounded placeholder-gray-500"
id={label}
value={value}
onChange={(e) => {
setValue(e.target.value);
}}
/>

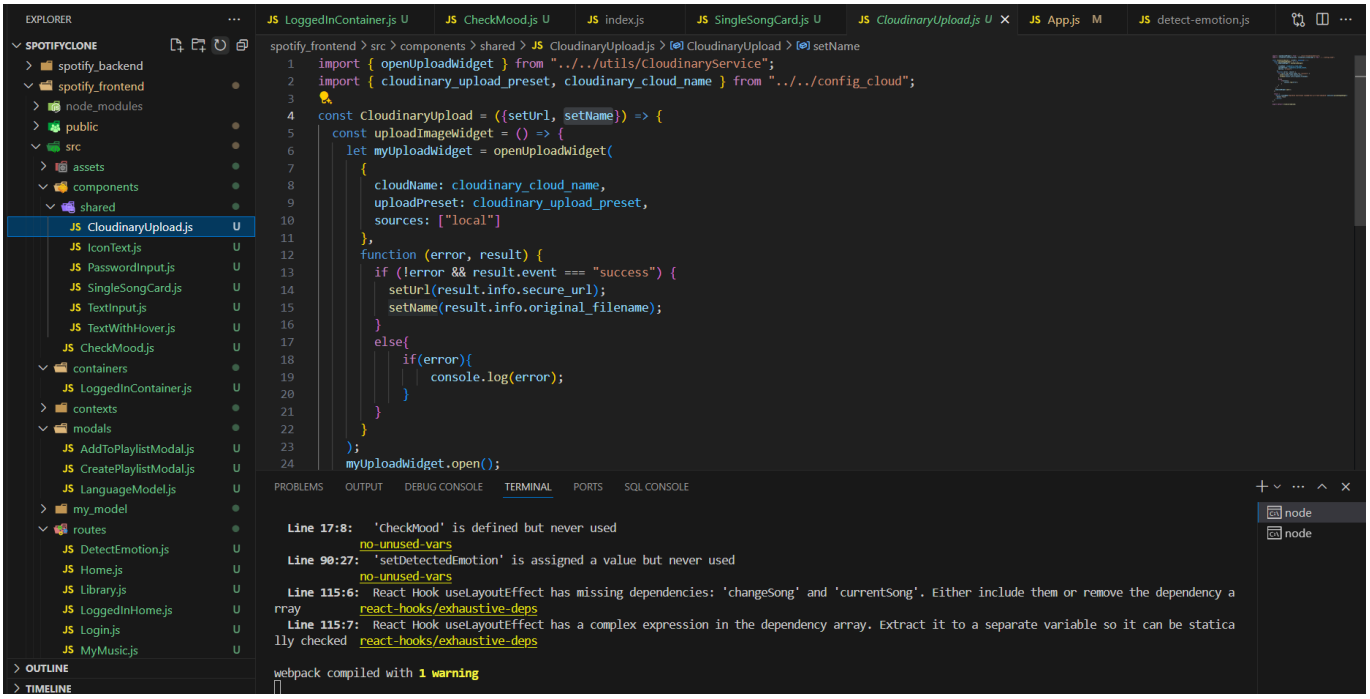
);
}
export default TextInput;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

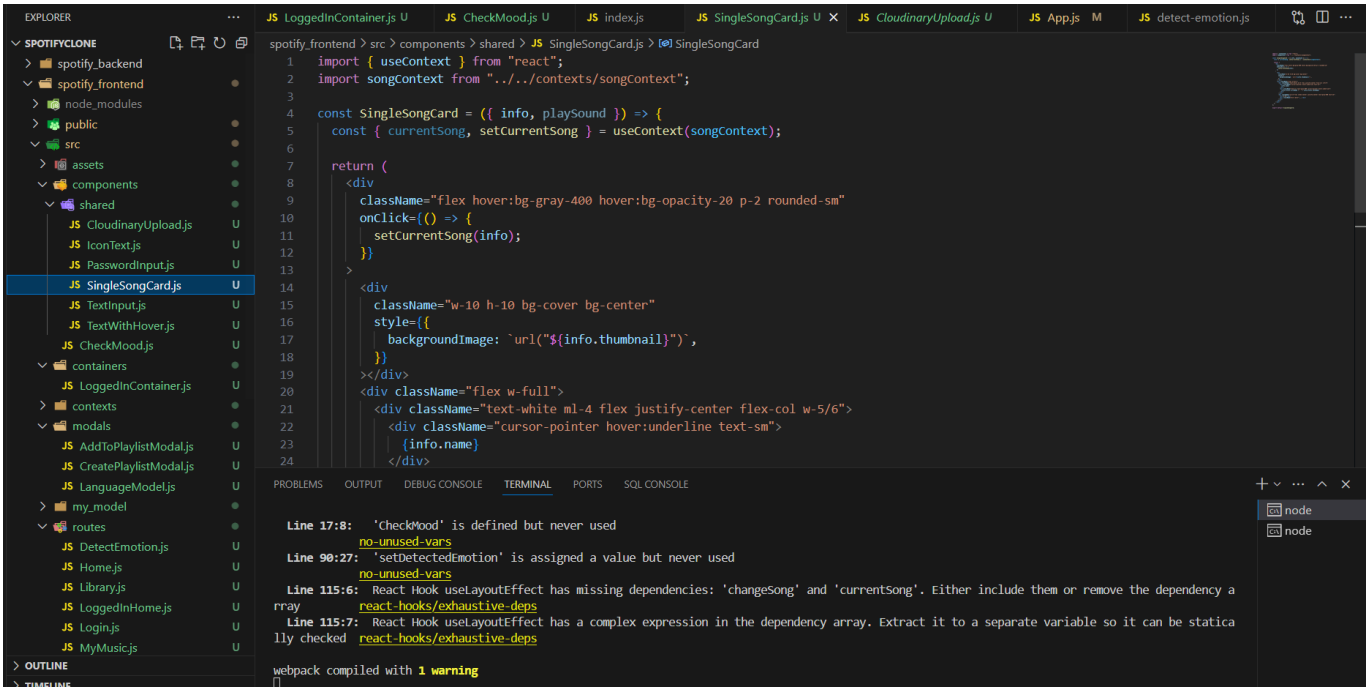

EXPLORER
SPOTIFYCLONE
spotify_backend
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
SingleSongCard.js U
IconText.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > IconText.js > IconText
import { Icon } from "@iconify/react";
import { Link } from "react-router-dom"
const IconText = ({ iconName, displayText, active, targetLink, onClick }) => {
return (
<Link to={targetLink} className="block">

<Icon
icon={iconName}
color={active ? "white" : "gray"}
fontSize={30}
/>

<div
className={`${
active ? "text-white" : "text-gray-400"
} text-sm font-semibold hover:text-white`}
>
{displayText}

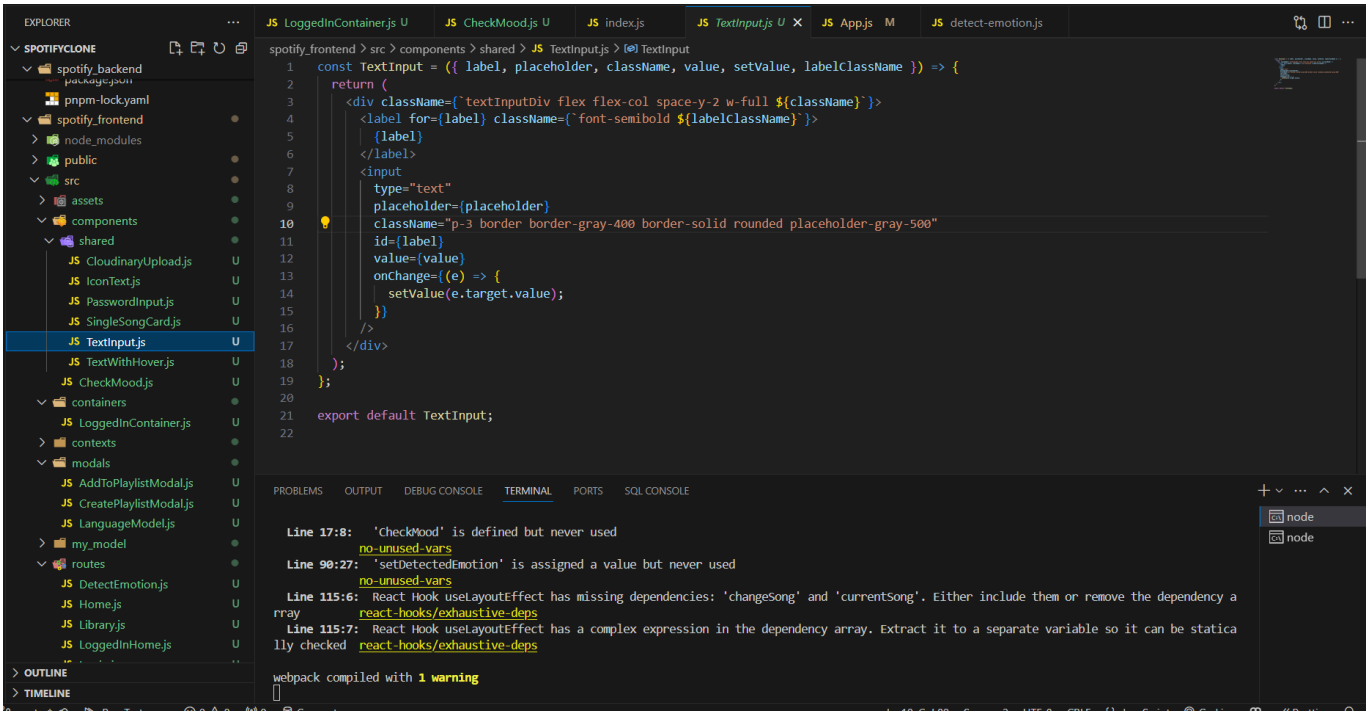
</Link>
);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_backend
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
SingleSongCard.js U
CloudinaryUpload.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > CloudinaryUpload.js > CloudinaryUpload > setName
import { openUploadWidget } from "../../utils/CloudinaryService";
import { cloudinary_upload_preset, cloudinary_cloud_name } from "../../config_cloud";
const CloudinaryUpload = ({setUrl, setName}) => {
const uploadImageWidget = () => {
let myUploadWidget = openUploadWidget(
{
cloudName: cloudinary_cloud_name,
uploadPreset: cloudinary_upload_preset,
sources: ["local"]
},
function (error, result) {
if (!error && result.event === "success") {
setUrl(result.info.secure_url);
setName(result.info.original_filename);
}
else{
if(error){
console.log(error);
}
}
}
);
myUploadWidget.open();
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

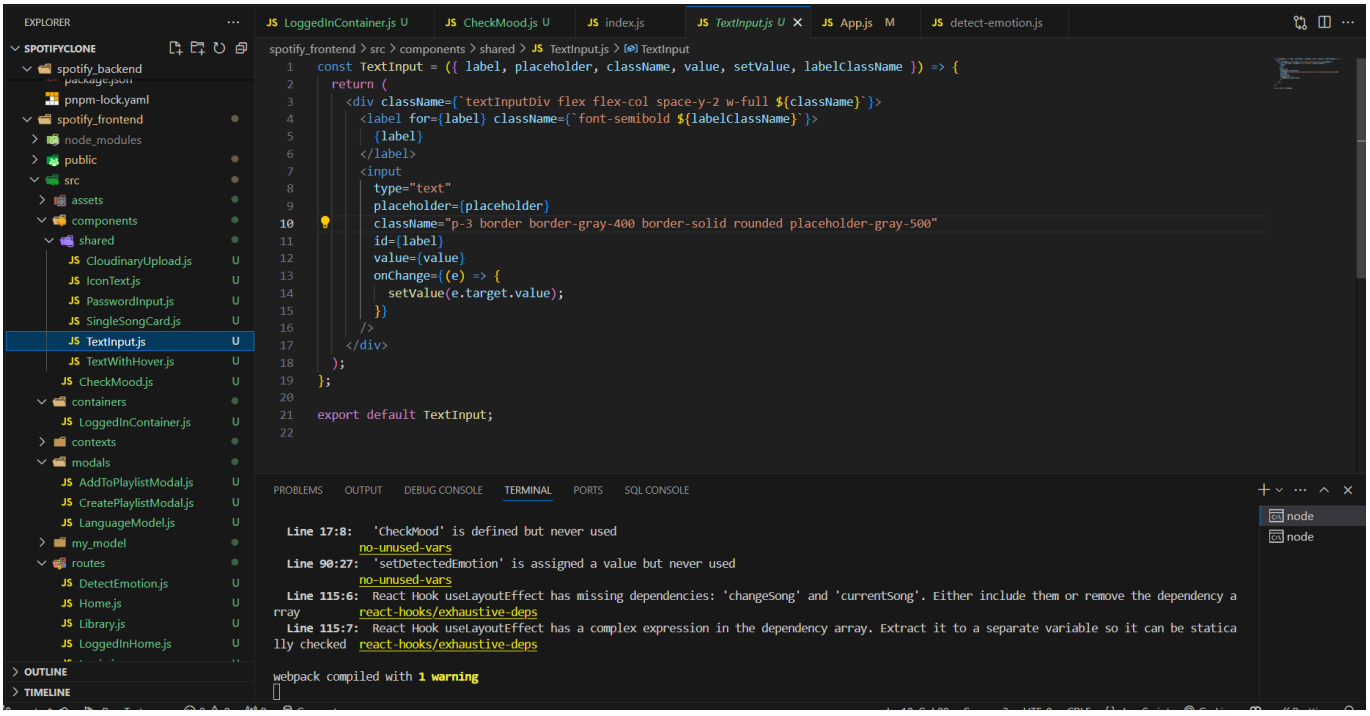

EXPLORER
SPOTIFYCLONE
spotify_backend
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
SingleSongCard.js U
CloudinaryUpload.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > SingleSongCard.js > SingleSongCard
import { useContext } from "react";
import songContext from "../../contexts/songContext";
const SingleSongCard = ({ info, playSound }) => {
const { currentSong, setCurrentSong } = useContext(songContext);
return (
<div
className="flex hover:bg-gray-400 hover:bg-opacity-20 p-2 rounded-sm"
onClick={() => {
setCurrentSong(info);
}}
>
<div
className="w-10 h-10 bg-cover bg-center"
style={{
backgroundImage: `url("${info.thumbnail}")`,
}}
>

{info.name}

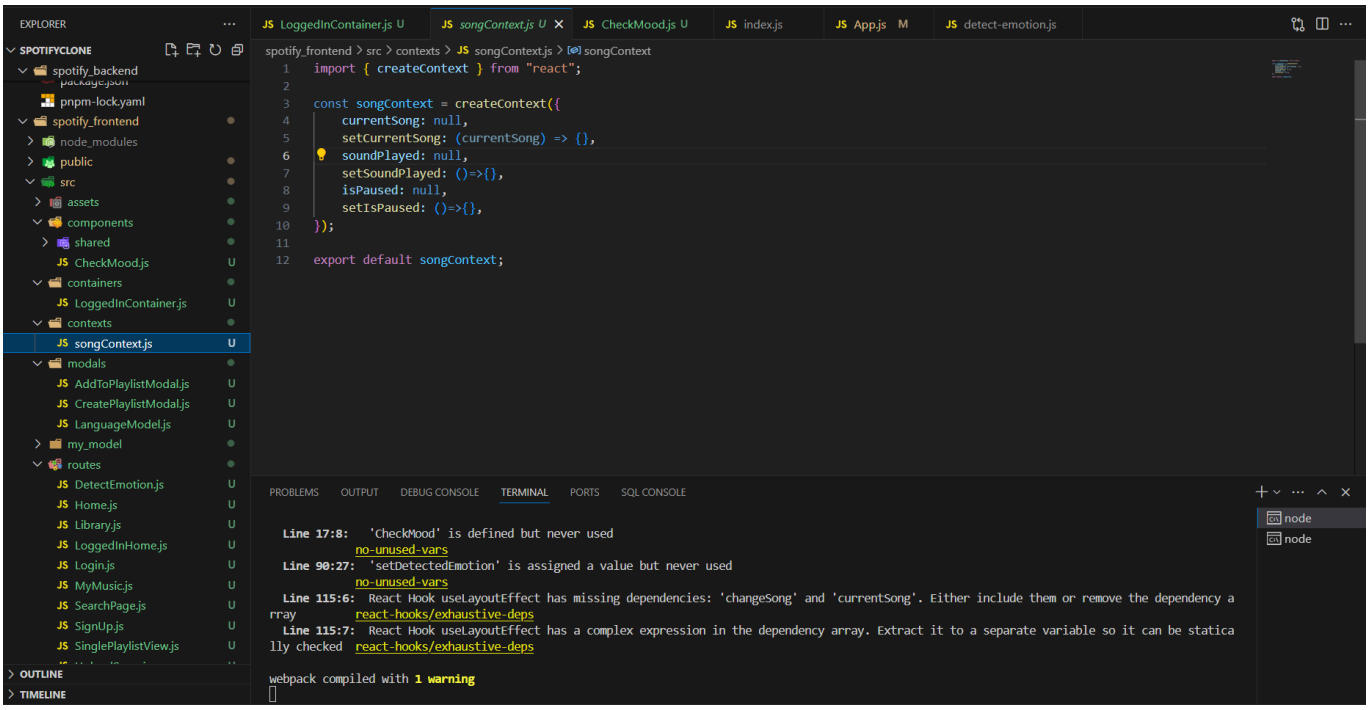
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

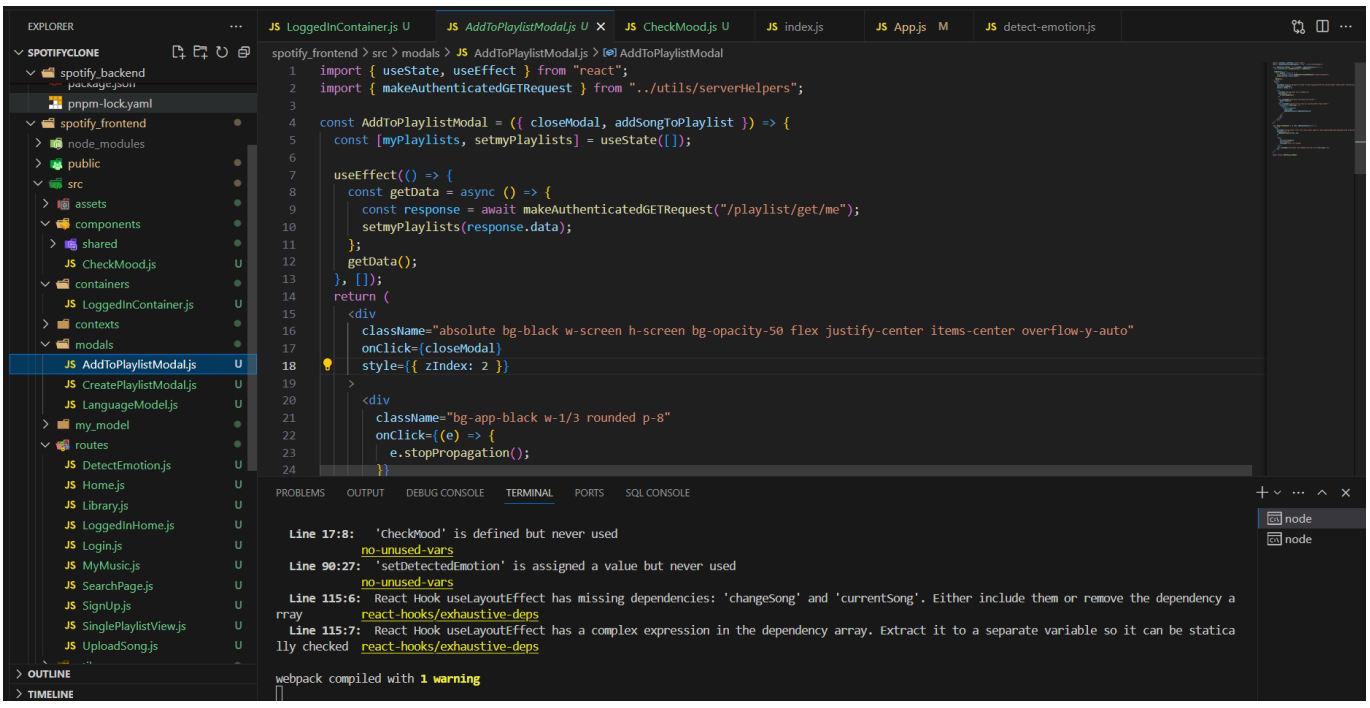

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
TextInput.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > TextInput.js > TextInput
const TextInput = ({ label, placeholder, className, value, setValue, labelClassName }) => {
return (
<div className={`textInputDiv flex flex-col space-y-2 w-full ${className}`}>
<label for={label} className={`font-semibold ${labelClassName}`}>
{label}
</label>
<input
type="text"
placeholder={placeholder}
className="p-3 border border-gray-400 border-solid rounded placeholder-gray-500"
id={label}
value={value}
onChange={(e) => {
setValue(e.target.value);
}}
/>

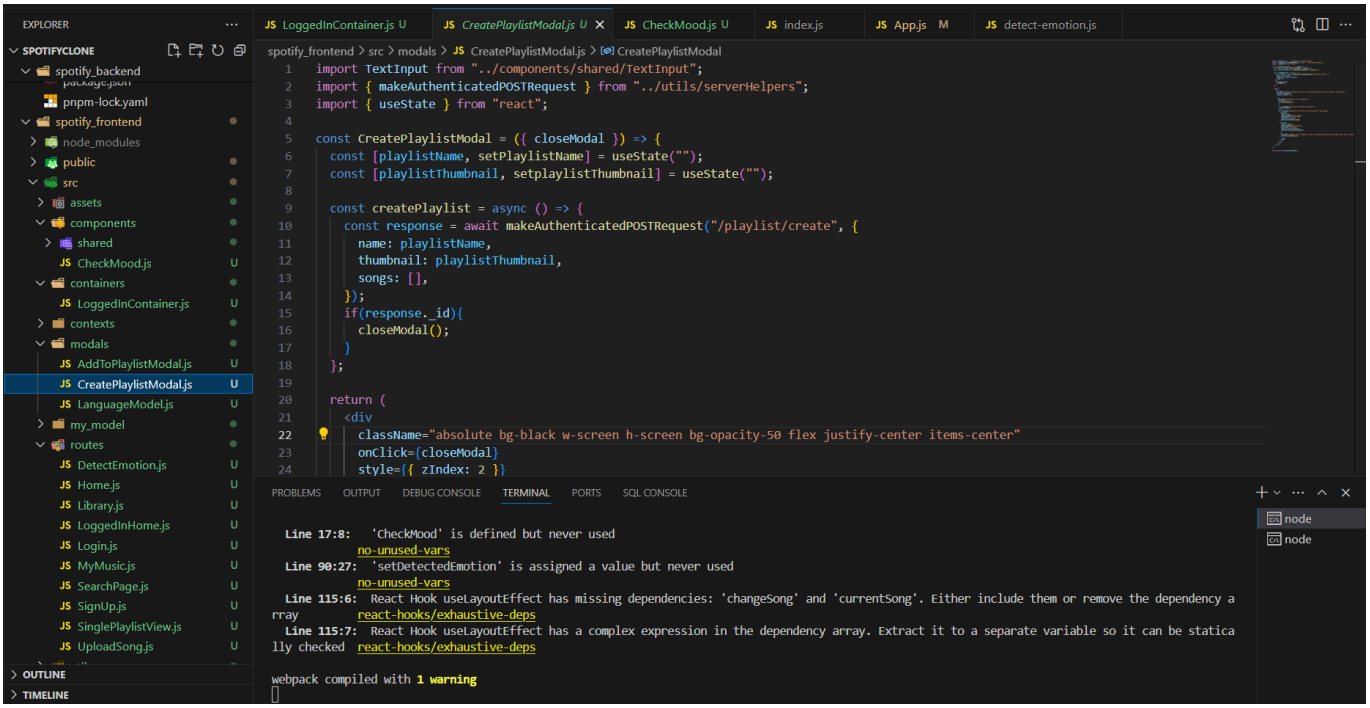
);
};
export default TextInput;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

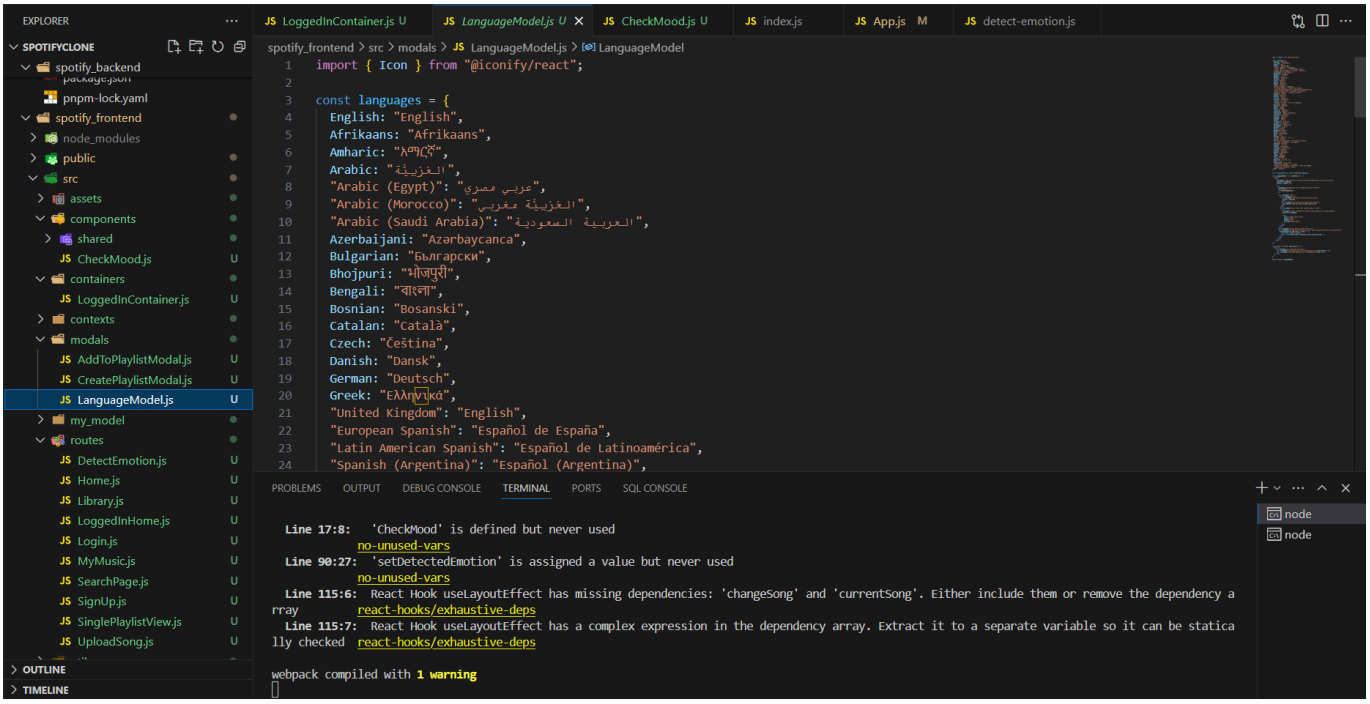

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CloudinaryUpload.js
IconText.js
PasswordInput.js
SingleSongCard.js
TextInput.js
TextWithHover.js
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CheckMood.js U
index.js
TextInput.js U
App.js M
detect-emotion.js
spotify_frontend > src > components > shared > TextInput.js > TextInput
const TextInput = ({ label, placeholder, className, value, setValue, labelClassName }) => {
return (
<div className={`textInputDiv flex flex-col space-y-2 w-full ${className}`}>
<label for={label} className={`font-semibold ${labelClassName}`}>
{label}
</label>
<input
type="text"
placeholder={placeholder}
className="p-3 border border-gray-400 border-solid rounded placeholder-gray-500"
id={label}
value={value}
onChange={(e) => {
setValue(e.target.value);
}}
/>

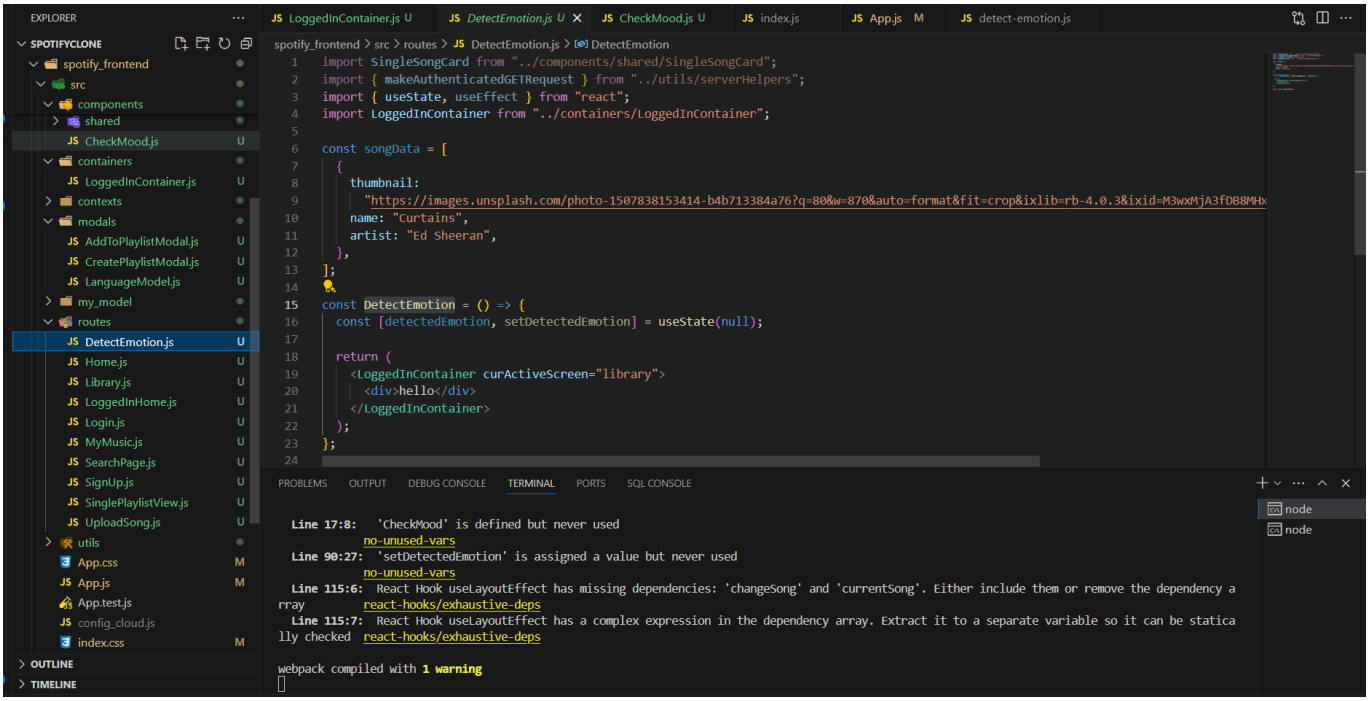
);
};
export default TextInput;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

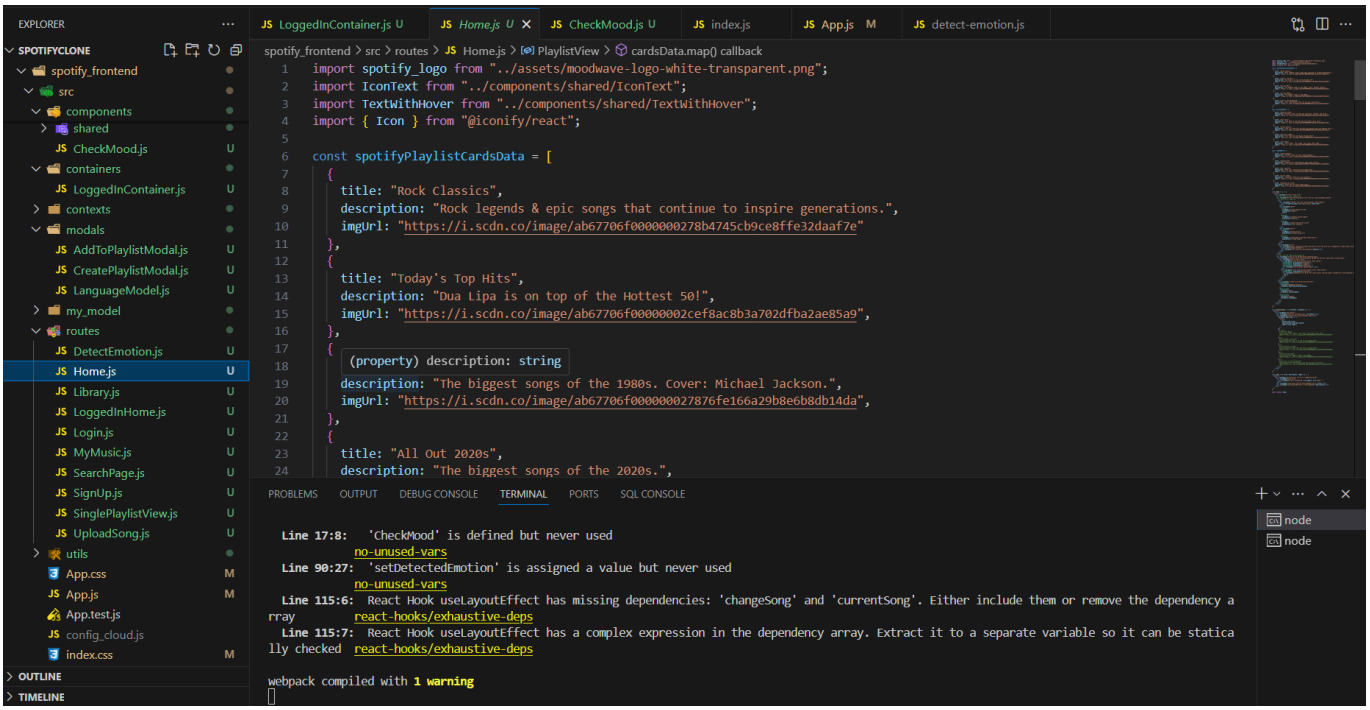

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
songContext.js
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
OUTLINE
TIMELINE
LoggedInContainer.js U
songContext.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > contexts > songContext.js > songContext
import { createContext } from "react";
const songContext = createContext({
currentSong: null,
setCurrentSong: (currentSong) => {},
soundPlayed: null,
setSoundPlayed: ()=>{},
isPaused: null,
setIsPaused: ()=>{},
});
export default songContext;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning
node
node

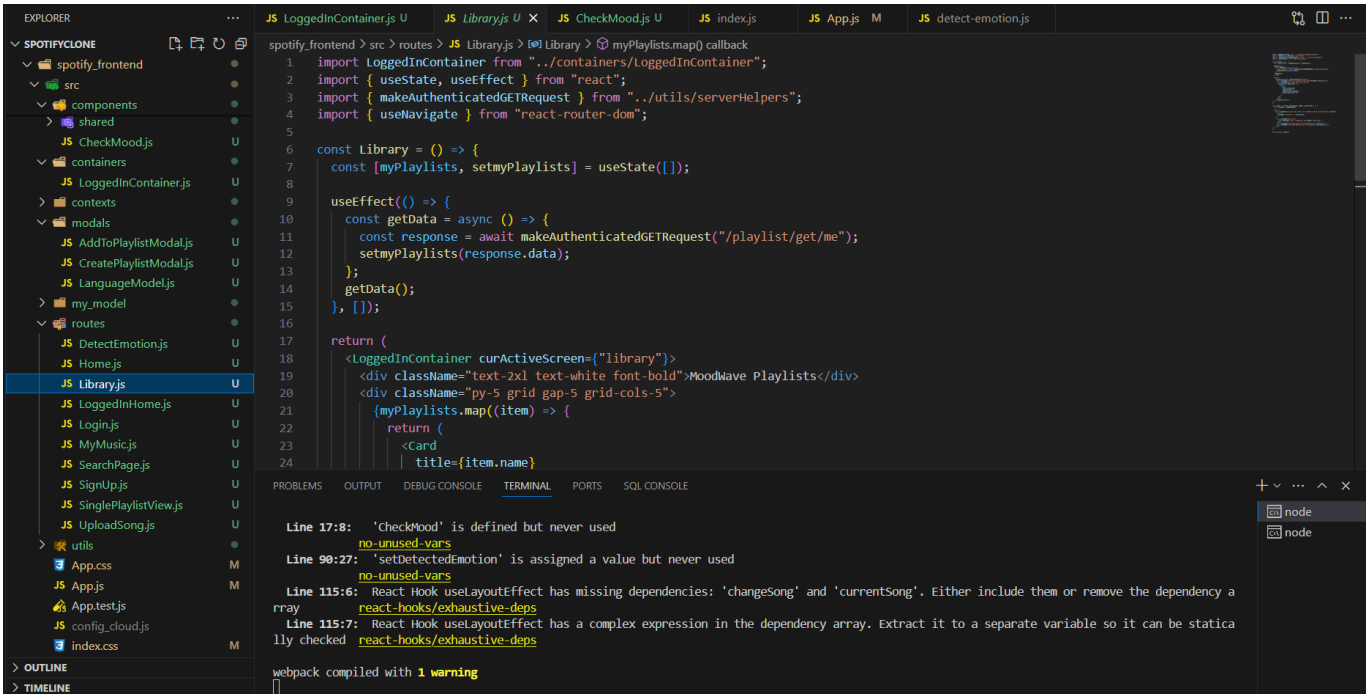

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
OUTLINE
TIMELINE
LoggedInContainer.js U
AddToPlaylistModal.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > modals > AddToPlaylistModal.js > AddToPlaylistModal
import { useState, useEffect } from "react";
import { makeAuthenticatedGETRequest } from "../utils/serverHelpers";
const AddToPlaylistModal = ({ closeModal, addSongToPlaylist }) => {
const [myPlaylists, setmyPlaylists] = useState([]);
useEffect(() => {
const getData = async () => {
const response = await makeAuthenticatedGETRequest("/playlist/get/me");
setmyPlaylists(response.data);
};
getData();
}, []);
return (
<div
className="absolute bg-black w-screen h-screen bg-opacity-50 flex justify-center items-center overflow-y-auto"
onClick={closeModal}
style={{ zIndex: 2 }}
>
<div
className="bg-app-black w-1/3 rounded p-8"
onClick={(e) => {
e.stopPropagation();
}}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning
node
node

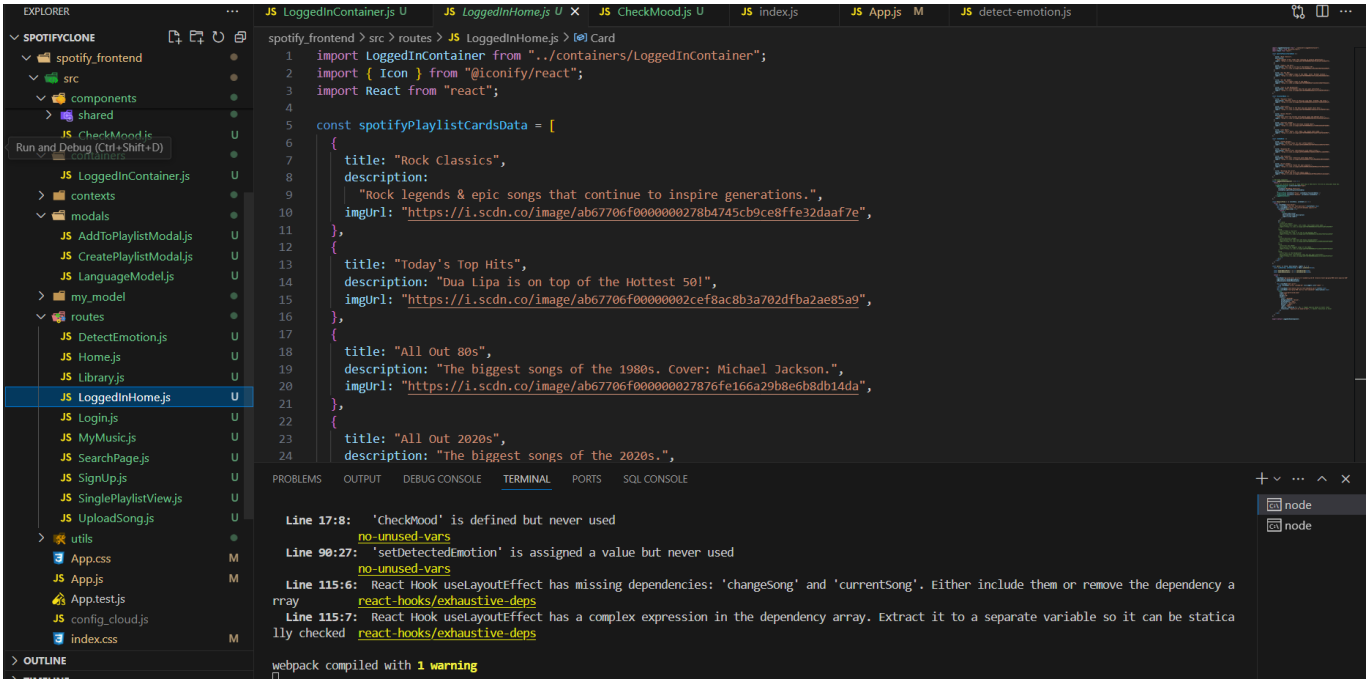

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
OUTLINE
TIMELINE
LoggedInContainer.js U
CreatePlaylistModal.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > modals > CreatePlaylistModal.js > CreatePlaylistModal
import TextInput from "../components/shared/TextInput";
import { makeAuthenticatedPOSTRequest } from "../utils/serverHelpers";
import { useState } from "react";
const CreatePlaylistModal = ({ closeModal }) => {
const [playlistName, setPlaylistName] = useState("");
const [playlistThumbnail, setplaylistThumbnail] = useState("");
const createPlaylist = async () => {
const response = await makeAuthenticatedPOSTRequest("/playlist/create", {
name: playlistName,
thumbnail: playlistThumbnail,
songs: [],
});
if(response._id){
closeModal();
}
};
return (
<div
className="absolute bg-black w-screen h-screen bg-opacity-50 flex justify-center items-center"
onClick={closeModal}
style={{ zIndex: 2 }}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

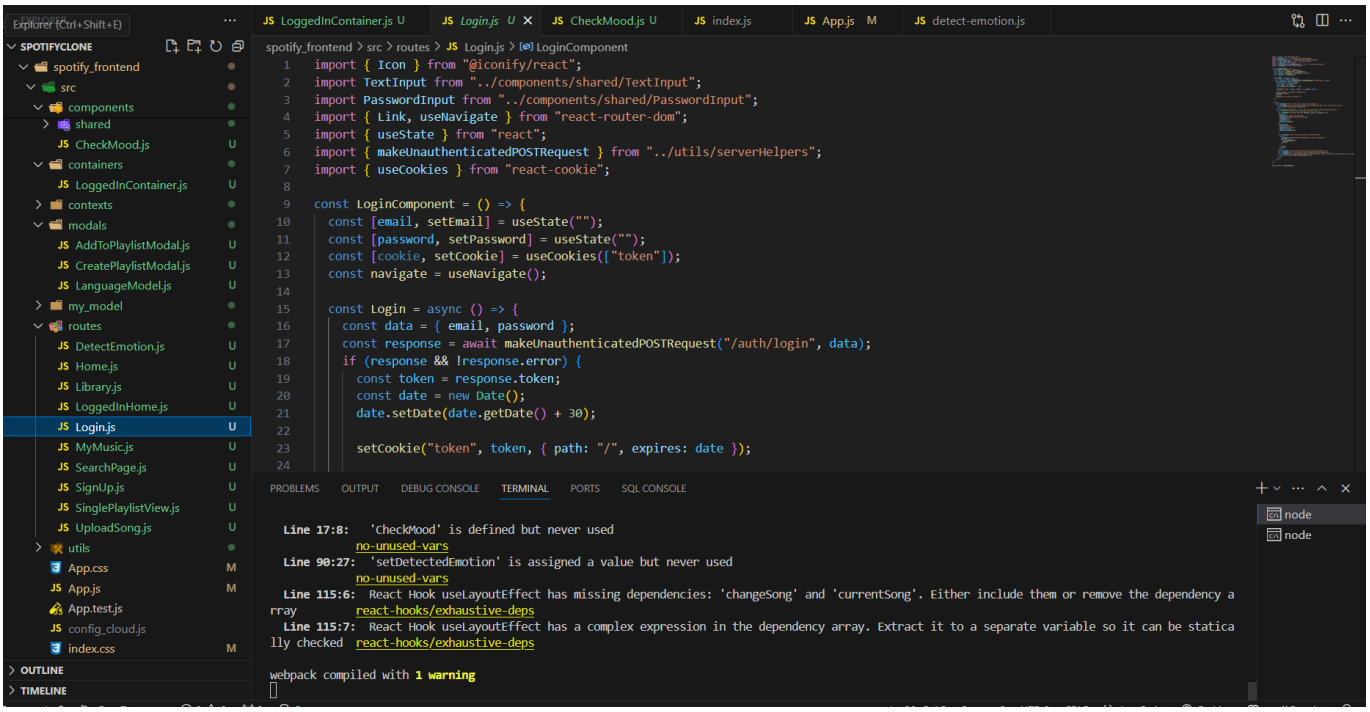

EXPLORER
SPOTIFYCLONE
spotify_backend
pnpm-lock.yaml
spotify_frontend
node_modules
public
src
assets
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
OUTLINE
TIMELINE
LoggedInContainer.js U
LanguageModel.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > modals > LanguageModel.js > LanguageModel
import { Icon } from "@iconify/react";
const languages = {
English: "English",
Afrikaans: "Afrikaans",
Amharic: "አማርኛ",
Arabic: "العَرَبِيَة",
"Arabic (Egypt)": "عربي مصري",
"Arabic (Morocco)": "العَرَبِيَة مغربي",
"Arabic (Saudi Arabia)": "العربية السعودية",
Azerbaijani: "Azərbaycanca",
Bulgarian: "Български",
Bhojpuri: "भोजपुरी",
Bengali: "বাংলা",
Bosnian: "Bosanski",
Catalan: "Català",
Czech: "Čeština",
Danish: "Dansk",
German: "Deutsch",
Greek: "Ελληνικά",
"United Kingdom": "English",
"European Spanish": "Español de España",
"Latin American Spanish": "Español de Latinoamérica",
"Spanish (Argentina)": "Español (Argentina)",
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

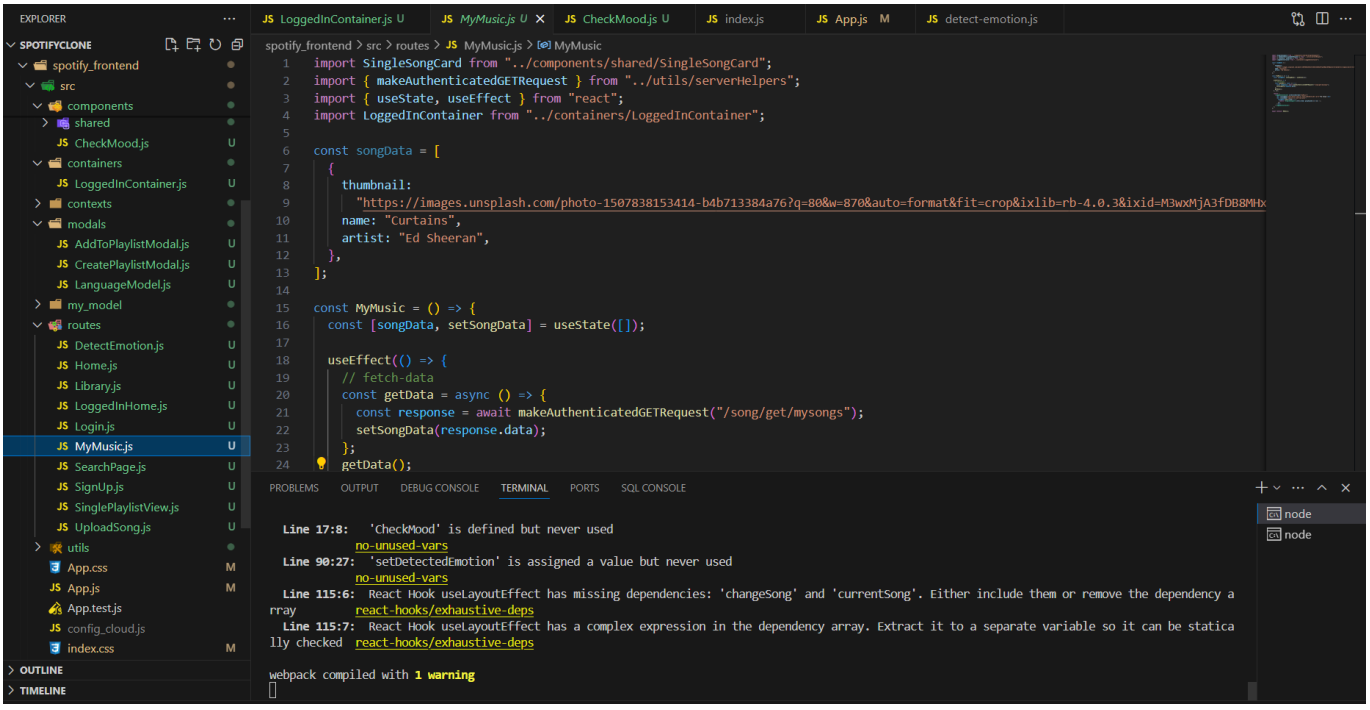

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
DetectEmotion.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > DetectEmotion.js > DetectEmotion
import SingleSongCard from "../components/shared/SingleSongCard";
import { makeAuthenticatedGETRequest } from "../utils/serverHelpers";
import { useState, useEffect } from "react";
import LoggedInContainer from "../containers/LoggedInContainer";
const songData = [
{
thumbnail:
"https://images.unsplash.com/photo-1507838153414-b4b713384a76?q=80&w=870&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx
name: "Curtains",
artist: "Ed Sheeran",
},
];
const DetectEmotion = () => {
const [detectedEmotion, setDetectedEmotion] = useState(null);
return (
<LoggedInContainer curActiveScreen="library">
hello
</LoggedInContainer>
);
};
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency a
rray react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statica
lly checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

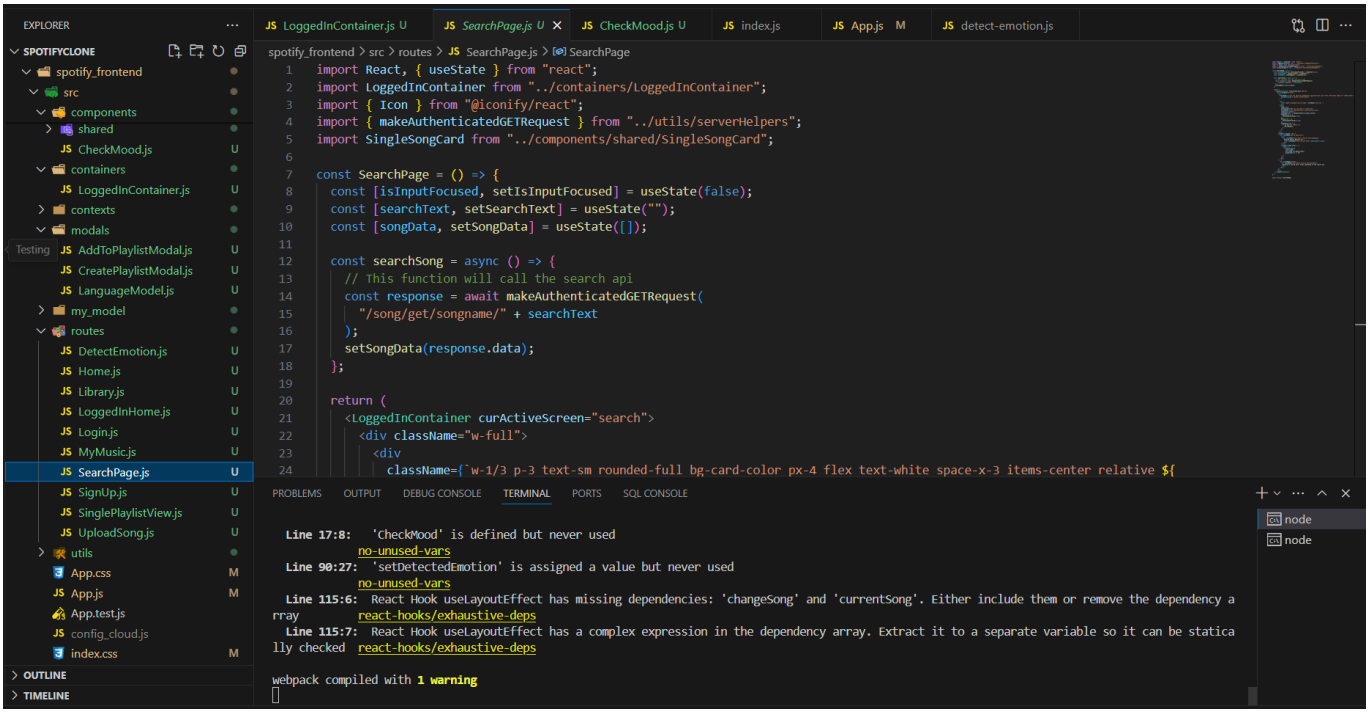

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
Home.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > Home.js > PlaylistView > cardsData.map() callback
import spotify_logo from "../assets/moodwave-logo-white-transparent.png";
import IconText from "../components/shared/IconText";
import TextWithHover from "../components/shared/TextWithHover";
import { Icon } from "@iconify/react";
const spotifyPlaylistCardsData = [
{
title: "Rock Classics",
description: "Rock legends & epic songs that continue to inspire generations.",
imgUrl: "https://i.scdn.co/image/ab67706f0000000278b4745cb9ce8ffe32daaf7e"
},
{
title: "Today's Top Hits",
description: "Dua Lipa is on top of the Hottest 50!",
imgUrl: "https://i.scdn.co/image/ab67706f00000002cef8ac8b3a702dfba2ae85a9",
},
{
(property) description: string
description: "The biggest songs of the 1980s. Cover: Michael Jackson.",
imgUrl: "https://i.scdn.co/image/ab67706f000000027876fe166a29b8e6b8db14da",
},
{
title: "All Out 2020s",
description: "The biggest songs of the 2020s.",
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency a
rray react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statica
lly checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js U
containers
LoggedInContainer.js U
contexts
modals
AddToPlaylistModal.js U
CreatePlaylistModal.js U
LanguageModel.js U
my_model
routes
DetectEmotion.js U
Home.js U
Library.js U
LoggedInHome.js U
Login.js U
MyMusic.js U
SearchPage.js U
SignUp.js U
SinglePlaylistView.js U
UploadSong.js U
utils
App.css M
App.js M
App.test.js
config_cloud.js
index.css M
OUTLINE
TIMELINE
LoggedInContainer.js U
Library.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > Library.js > Library > myPlaylists.map() callback
1 import LoggedInContainer from "../containers/LoggedInContainer";
2 import { useState, useEffect } from "react";
3 import { makeAuthenticatedGETRequest } from "../utils/serverHelpers";
4 import { useNavigate } from "react-router-dom";
6 const Library = () => {
7 const [myPlaylists, setmyPlaylists] = useState([]);
9 useEffect(() => {
10 const getData = async () => {
11 const response = await makeAuthenticatedGETRequest("/playlist/get/me");
12 setmyPlaylists(response.data);
13 };
14 getData();
15 }, []);
17 return (
18 <LoggedInContainer curActiveScreen={"library"}>
19 MoodWave Playlists
20
21 {myPlaylists.map((item) => {
22 return (
23 <Card
24 title={item.name}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

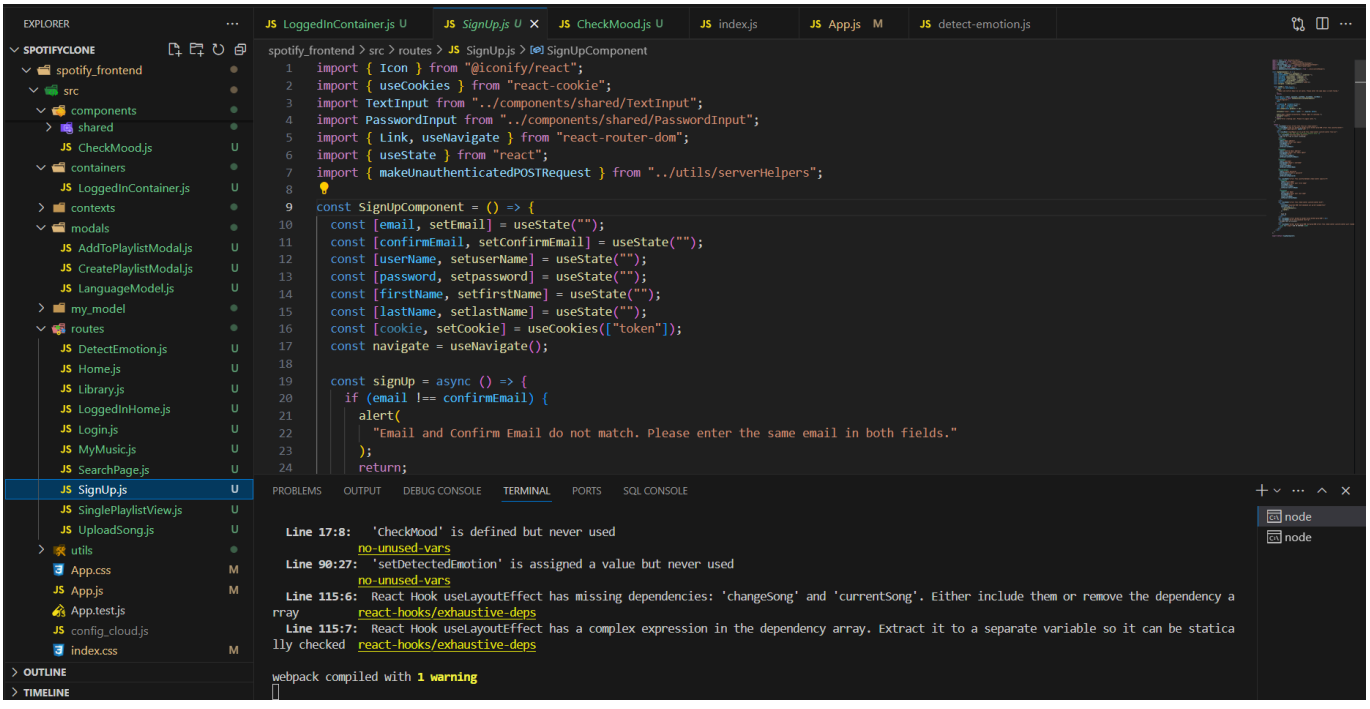

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
Run and Debug (Ctrl+Shift+D)
LoggedInContainer.js U
contexts
modals
AddToPlaylistModal.js U
CreatePlaylistModal.js U
LanguageModel.js U
my_model
routes
DetectEmotion.js U
Home.js U
Library.js U
LoggedInHome.js U
Login.js U
MyMusic.js U
SearchPage.js U
SignUp.js U
SinglePlaylistView.js U
UploadSong.js U
utils
App.css M
App.js M
App.test.js
config_cloud.js
index.css M
OUTLINE
LoggedInContainer.js U
LoggedInHome.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > LoggedInHome.js > Card
1 import LoggedInContainer from "../containers/LoggedInContainer";
2 import { Icon } from "@iconify/react";
3 import React from "react";
5 const spotifyPlaylistCardsData = [
6 {
7 title: "Rock Classics",
8 description:
9 "Rock legends & epic songs that continue to inspire generations.",
10 imgUrl: "https://i.scdn.co/image/ab67706f000000278b4745cb9ce8ffe32daaf7e",
11 },
12 {
13 title: "Today's Top Hits",
14 description: "Dua Lipa is on top of the Hottest 50!",
15 imgUrl: "https://i.scdn.co/image/ab67706f00000002cef8ac8b3a702dfba2ae85a9",
16 },
17 {
18 title: "All Out 80s",
19 description: "The biggest songs of the 1980s. Cover: Michael Jackson.",
20 imgUrl: "https://i.scdn.co/image/ab67706f00000027876fe166a29b8e6b8db14da",
21 },
22 {
23 title: "All Out 2020s",
24 description: "The biggest songs of the 2020s.",
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

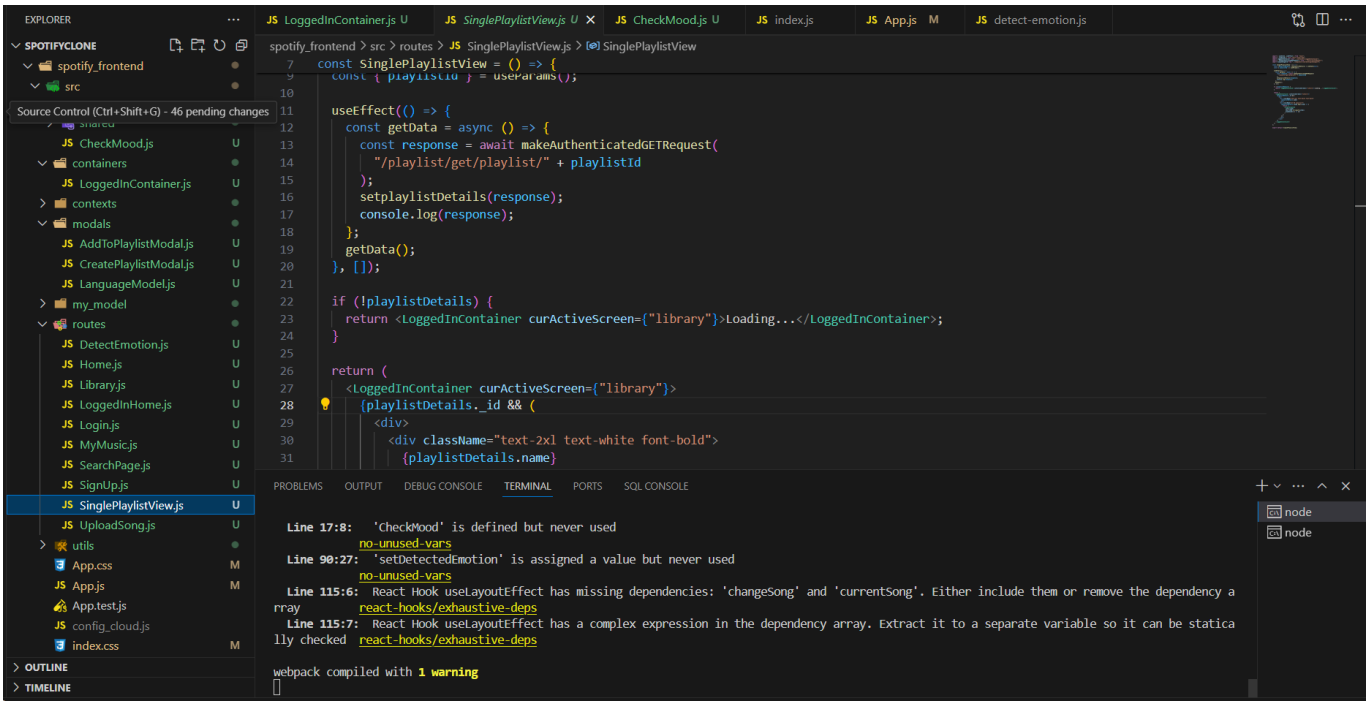

Explorer (Ctrl+Shift+E)
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
Login.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > Login.js > LoginComponent
import { Icon } from "@iconify/react";
import TextInput from "../components/shared/TextInput";
import PasswordInput from "../components/shared/PasswordInput";
import { Link, useNavigate } from "react-router-dom";
import { useState } from "react";
import { makeUnauthenticatedPOSTRequest } from "../utils/serverHelpers";
import { useCookies } from "react-cookie";
const LoginComponent = () => {
const [email, setEmail] = useState("");
const [password, setPassword] = useState("");
const [cookie, setCookie] = useCookies(["token"]);
const navigate = useNavigate();
const Login = async () => {
const data = { email, password };
const response = await makeUnauthenticatedPOSTRequest("/auth/login", data);
if (response && !response.error) {
const token = response.token;
const date = new Date();
date.setDate(date.getDate() + 30);
setCookie("token", token, { path: "/", expires: date });
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'checkMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
MyMusic.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > MyMusic.js > MyMusic
import SingleSongCard from "../components/shared/SingleSongCard";
import { makeAuthenticatedGETRequest } from "../utils/serverHelpers";
import { useState, useEffect } from "react";
import LoggedInContainer from "../containers/LoggedInContainer";
const songData = [
{
thumbnail:
"https://images.unsplash.com/photo-1507838153414-b4b713384a76?q=80&w=870&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHx
name: "Curtains",
artist: "Ed Sheeran",
},
];
const MyMusic = () => {
const [songData, setSongData] = useState([]);
useEffect(() => {
// fetch-data
const getData = async () => {
const response = await makeAuthenticatedGETRequest("/song/get/mysongs");
setSongData(response.data);
};
getData();
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'checkMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

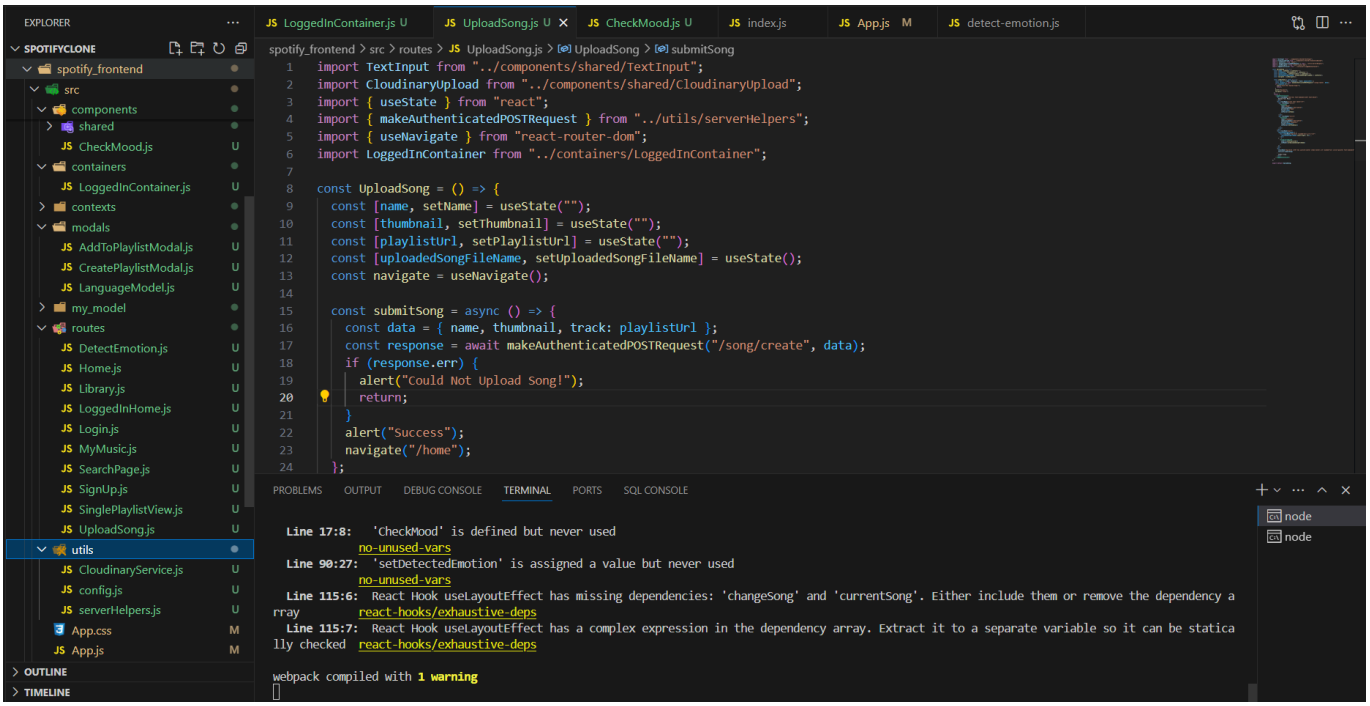

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
SearchPage.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > SearchPage.js > SearchPage
import React, { useState } from "react";
import LoggedInContainer from "../containers/LoggedInContainer";
import { Icon } from "@iconify/react";
import { makeAuthenticatedGETRequest } from "../utils/serverHelpers";
import SingleSongCard from "../components/shared/SingleSongCard";
const SearchPage = () => {
const [isInputFocused, setIsInputFocused] = useState(false);
const [searchText, setSearchText] = useState("");
const [songData, setSongData] = useState([]);
const searchSong = async () => {
// This function will call the search api
const response = await makeAuthenticatedGETRequest(
"/song/get/songname/" + searchText
);
setSongData(response.data);
};
return (
<LoggedInContainer curActiveScreen="search">

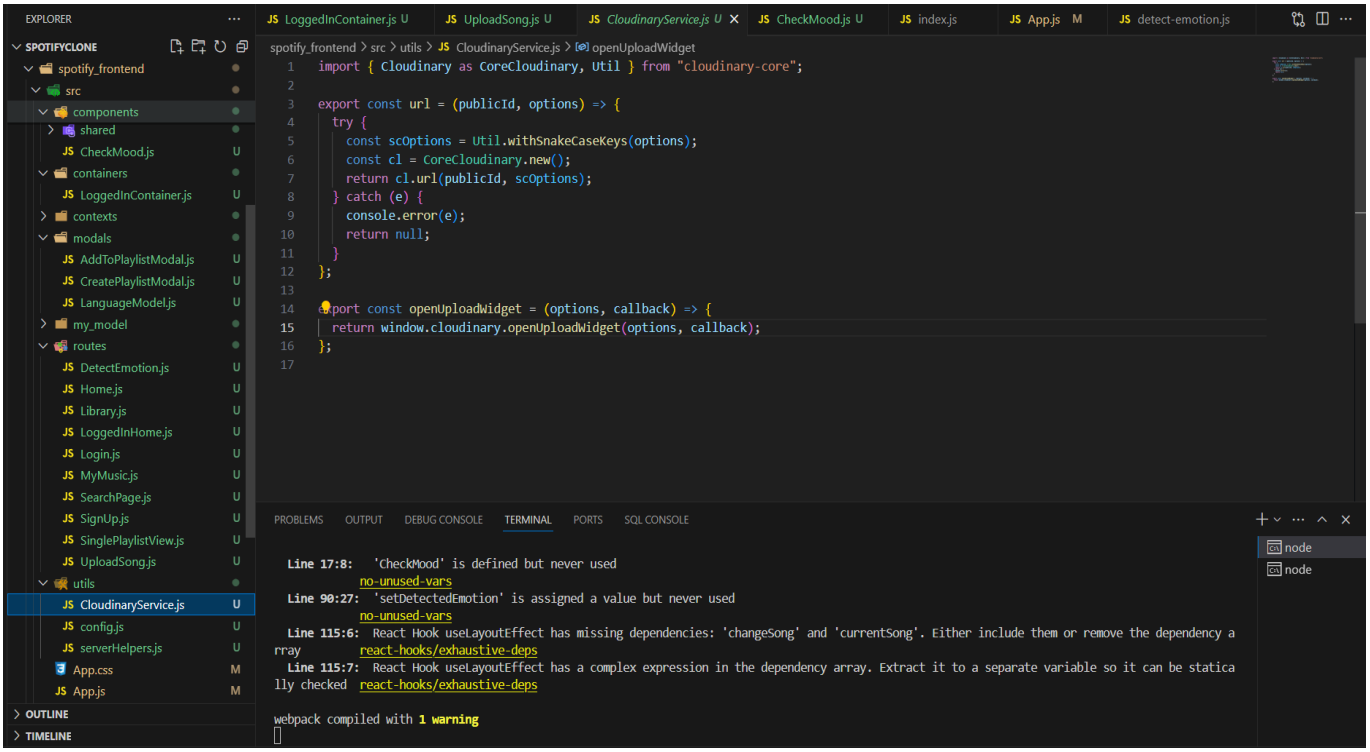
<div
className={`w-1/3 p-3 text-sm rounded-full bg-card-color px-4 flex text-white space-x-3 items-center relative ${
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'checkMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

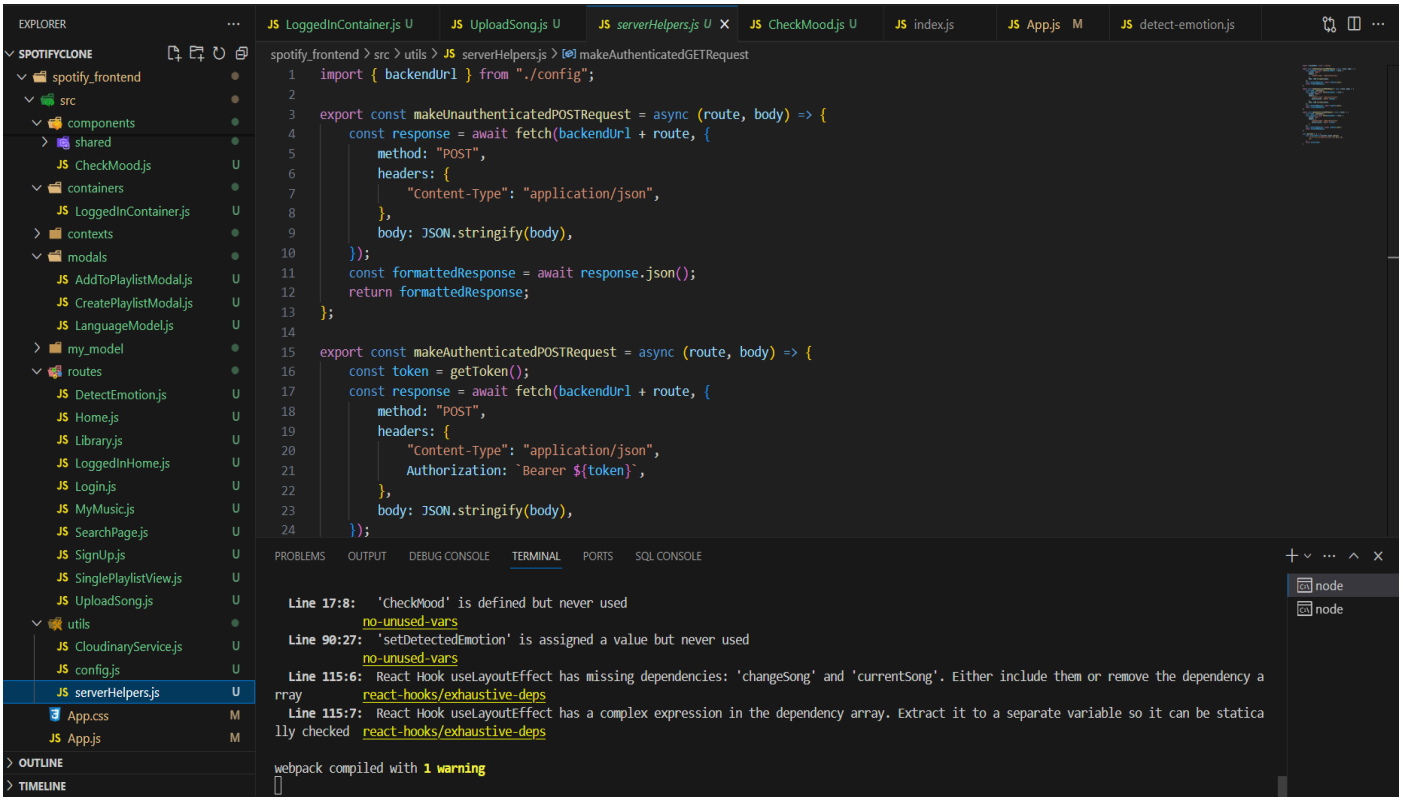

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
SignUp.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > SignUp.js > SignUpComponent
import { Icon } from "@iconify/react";
import { useCookies } from "react-cookie";
import TextInput from "../components/shared/TextInput";
import PasswordInput from "../components/shared/PasswordInput";
import { Link, useNavigate } from "react-router-dom";
import { useState } from "react";
import { makeUnauthenticatedPOSTRequest } from "../utils/serverHelpers";
const SignUpComponent = () => {
const [email, setEmail] = useState("");
const [confirmEmail, setConfirmEmail] = useState("");
const [userName, setuserName] = useState("");
const [password, setpassword] = useState("");
const [firstName, setfirstName] = useState("");
const [lastName, setlastName] = useState("");
const [cookie, setCookie] = useCookies(["token"]);
const navigate = useNavigate();
const signUp = async () => {
if (email !== confirmEmail) {
alert(
"Email and Confirm Email do not match. Please enter the same email in both fields."
);
return;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'checkMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

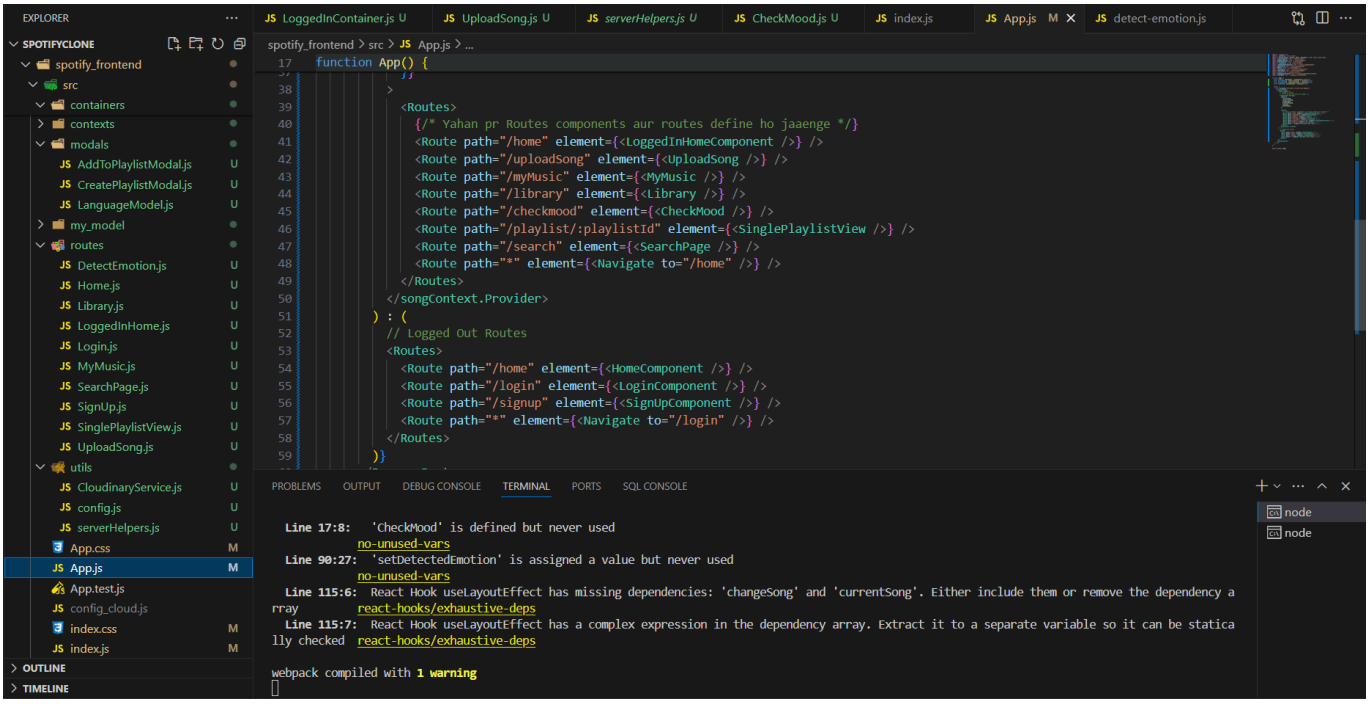

EXPLORER
SPOTIFYCLONE
spotify_frontend
src
Source Control (Ctrl+Shift+G) - 46 pending changes
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
App.css
App.js
App.test.js
config_cloud.js
index.css
OUTLINE
TIMELINE
LoggedInContainer.js U
SinglePlaylistView.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > SinglePlaylistView.js > SinglePlaylistView
const SinglePlaylistView = () => {
const { playlistId } = useParams();
useEffect(() => {
const getData = async () => {
const response = await makeAuthenticatedGETRequest(
"/playlist/get/playlist/" + playlistId
);
setplaylistDetails(response);
console.log(response);
};
getData();
}, []);
if (!playlistDetails) {
return <LoggedInContainer curActiveScreen={"library"}>Loading...</LoggedInContainer>;
}
return (
<LoggedInContainer curActiveScreen={"library"}>
{playlistDetails._id && (

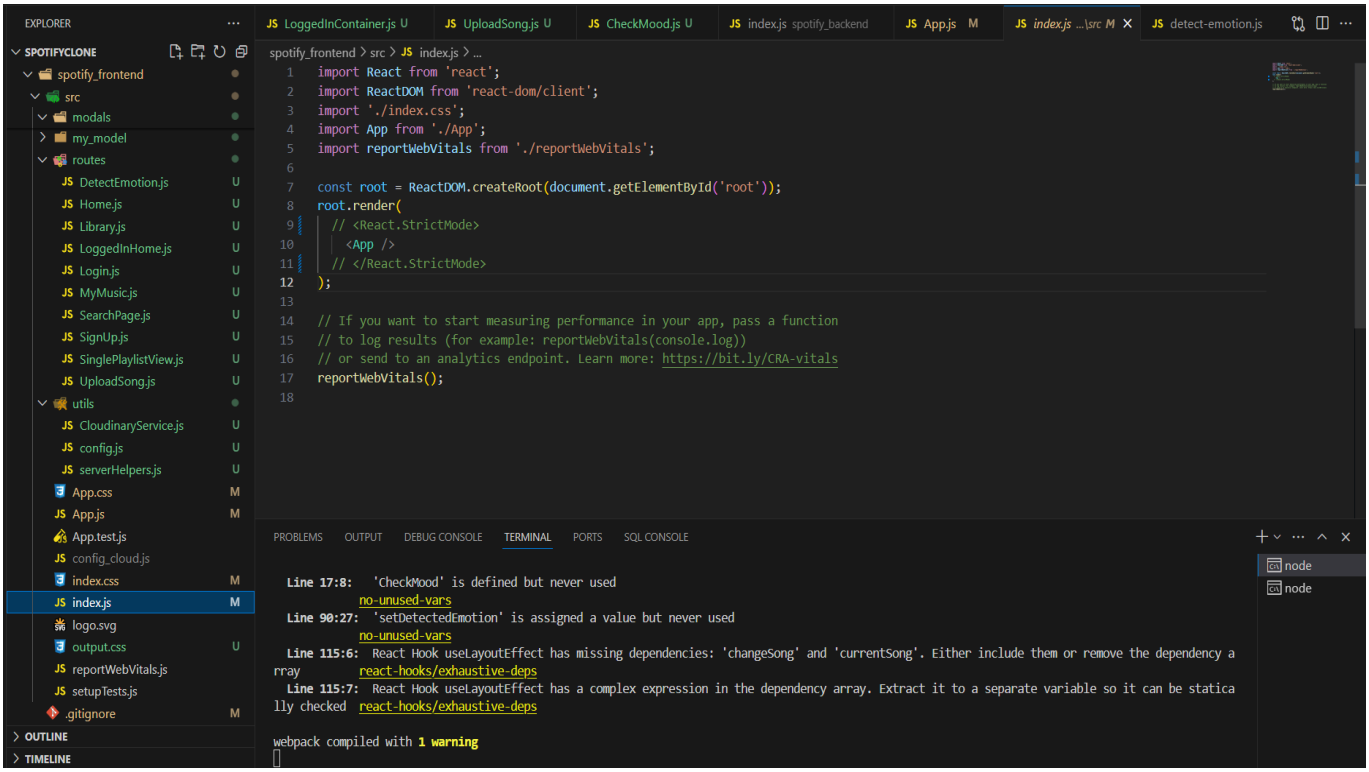
{playlistDetails.name}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js
containers
LoggedInContainer.js
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
CloudinaryService.js
config.js
serverHelpers.js
App.css
App.js
OUTLINE
TIMELINE
LoggedInContainer.js U
UploadSong.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > routes > UploadSong.js > UploadSong > submitSong
import TextInput from "../components/shared/TextInput";
import CloudinaryUpload from "../components/shared/CloudinaryUpload";
import { useState } from "react";
import { makeAuthenticatedPOSTRequest } from "../utils/serverHelpers";
import { useNavigate } from "react-router-dom";
import LoggedInContainer from "../containers/LoggedInContainer";
const UploadSong = () => {
const [name, setName] = useState("");
const [thumbnail, setThumbnail] = useState("");
const [playlistUrl, setPlaylistUrl] = useState("");
const [uploadedSongFileName, setUploadedSongFileName] = useState();
const navigate = useNavigate();
const submitSong = async () => {
const data = { name, thumbnail, track: playlistUrl };
const response = await makeAuthenticatedPOSTRequest("/song/create", data);
if (response.err) {
alert("Could Not Upload Song!");
return;
}
alert("Success");
navigate("/home");
};
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js U
containers
LoggedInContainer.js U
contexts
modals
AddToPlaylistModal.js U
CreatePlaylistModal.js U
LanguageModel.js U
my_model
routes
DetectEmotion.js U
Home.js U
Library.js U
LoggedInHome.js U
Login.js U
MyMusic.js U
SearchPage.js U
SignUp.js U
SinglePlaylistView.js U
UploadSong.js U
utils
CloudinaryService.js U
config.js U
serverHelpers.js U
App.css M
App.js M
OUTLINE
TIMELINE
LoggedInContainer.js U
UploadSong.js U
CloudinaryService.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > utils > CloudinaryService.js > openUploadWidget
1 import { Cloudinary as CoreCloudinary, Util } from "cloudinary-core";
2
3 export const url = (publicId, options) => {
4 try {
5 const scOptions = Util.withSnakeCaseKeys(options);
6 const cl = CoreCloudinary.new();
7 return cl.url(publicId, scOptions);
8 } catch (e) {
9 console.error(e);
10 return null;
11 }
12 };
13
14 export const openUploadWidget = (options, callback) => {
15 return window.cloudinary.openUploadWidget(options, callback);
16 };
17
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
components
shared
CheckMood.js U
containers
LoggedInContainer.js U
contexts
modals
AddToPlaylistModal.js U
CreatePlaylistModal.js U
LanguageModel.js U
my_model
routes
DetectEmotion.js U
Home.js U
Library.js U
LoggedInHome.js U
Login.js U
MyMusic.js U
SearchPage.js U
SignUp.js U
SinglePlaylistView.js U
UploadSong.js U
utils
CloudinaryService.js U
config.js U
serverHelpers.js U
App.css M
App.js M
OUTLINE
TIMELINE
LoggedInContainer.js U
UploadSong.js U
serverHelpers.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > utils > serverHelpers.js > makeAuthenticatedGETRequest
1 import { backendUrl } from "./config";
2
3 export const makeUnauthenticatedPOSTRequest = async (route, body) => {
4 const response = await fetch(backendUrl + route, {
5 method: "POST",
6 headers: {
7 "Content-Type": "application/json",
8 },
9 body: JSON.stringify(body),
10 });
11 const formattedResponse = await response.json();
12 return formattedResponse;
13 };
14
15 export const makeAuthenticatedPOSTRequest = async (route, body) => {
16 const token = getToken();
17 const response = await fetch(backendUrl + route, {
18 method: "POST",
19 headers: {
20 "Content-Type": "application/json",
21 Authorization: `Bearer ${token}`,
22 },
23 body: JSON.stringify(body),
24 });
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
containers
contexts
modals
AddToPlaylistModal.js
CreatePlaylistModal.js
LanguageModel.js
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
CloudinaryService.js
config.js
serverHelpers.js
App.css
App.js
App.test.js
config_cloud.js
index.css
index.js
OUTLINE
TIMELINE
LoggedInContainer.js U
UploadSong.js U
serverHelpers.js U
CheckMood.js U
index.js
App.js M
detect-emotion.js
spotify_frontend > src > App.js > ...
function App() {
<Routes>
{/* Yahan pr Routes components aur routes define ho jaaenge */}
<Route path="/home" element={<LoggedInHomeComponent />} />
<Route path="/uploadSong" element={<UploadSong />} />
<Route path="/myMusic" element={<MyMusic />} />
<Route path="/library" element={<Library />} />
<Route path="/checkmood" element={<CheckMood />} />
<Route path="/playlist/:playlistId" element={<SinglePlaylistView />} />
<Route path="/search" element={<SearchPage />} />
<Route path="*" element={<Navigate to="/home" />} />
</Routes>
</songContext.Provider>
) : (
// Logged Out Routes
<Routes>
<Route path="/home" element={<HomeComponent />} />
<Route path="/login" element={<LoginComponent />} />
<Route path="/signup" element={<SignUpComponent />} />
<Route path="*" element={<Navigate to="/login" />} />
</Routes>
)}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning


EXPLORER
SPOTIFYCLONE
spotify_frontend
src
modals
my_model
routes
DetectEmotion.js
Home.js
Library.js
LoggedInHome.js
Login.js
MyMusic.js
SearchPage.js
SignUp.js
SinglePlaylistView.js
UploadSong.js
utils
CloudinaryService.js
config.js
serverHelpers.js
App.css
App.js
App.test.js
config_cloud.js
index.css
index.js
logo.svg
output.css
reportWebVitals.js
setupTests.js
.gitignore
OUTLINE
TIMELINE
LoggedInContainer.js U
UploadSong.js U
CheckMood.js U
index.js spotify_backend
App.js M
index.js ...\src M
detect-emotion.js
spotify_frontend > src > index.js > ...
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
// <React.StrictMode>
<App />
// </React.StrictMode>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
SQL CONSOLE
node
node
Line 17:8: 'CheckMood' is defined but never used no-unused-vars
Line 90:27: 'setDetectedEmotion' is assigned a value but never used no-unused-vars
Line 115:6: React Hook useLayoutEffect has missing dependencies: 'changeSong' and 'currentSong'. Either include them or remove the dependency array react-hooks/exhaustive-deps
Line 115:7: React Hook useLayoutEffect has a complex expression in the dependency array. Extract it to a separate variable so it can be statically checked react-hooks/exhaustive-deps
webpack compiled with 1 warning

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('/content/monthly_milk_production.csv',index_col='Date',parse_dates=True)
df.index.freq='MS'
```

```
df.head()
```

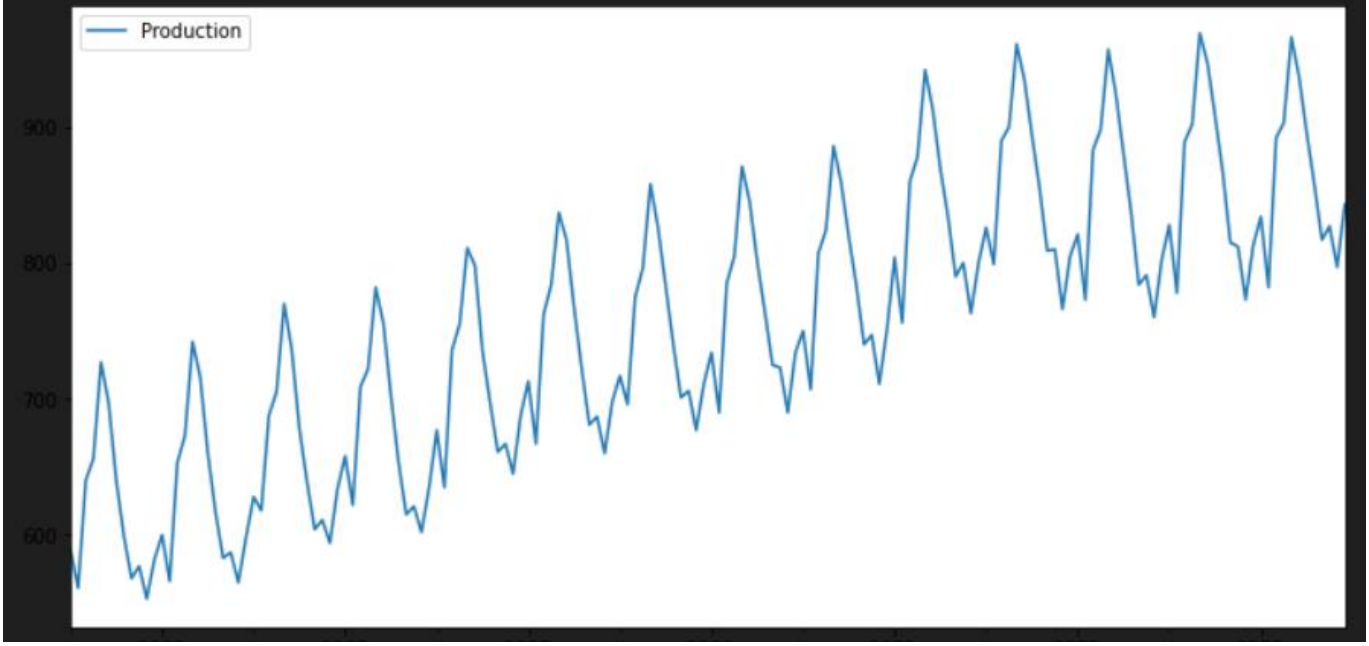
...

```
df.plot(figsize=(12,6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8f21deb910>
```

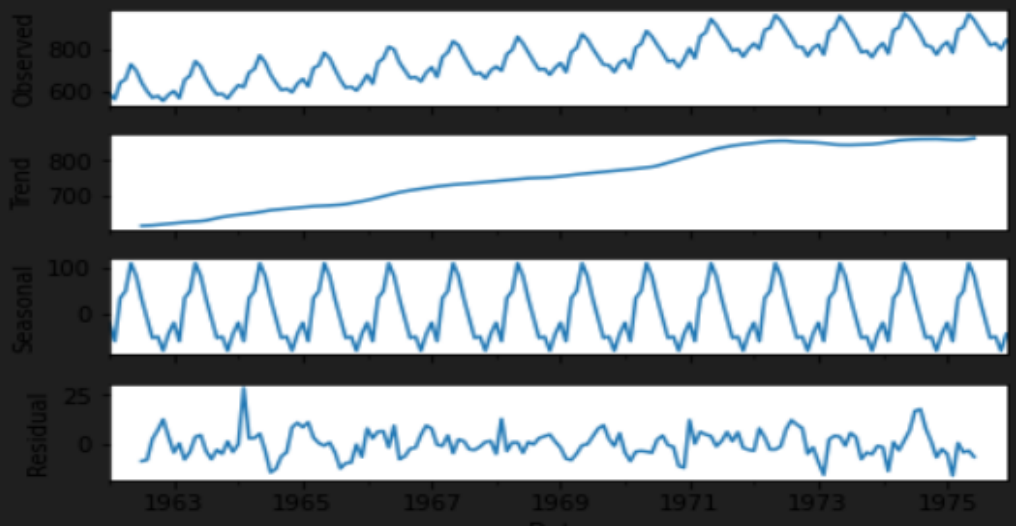

```
df.plot(figsize=(12,6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8f21deb910>
```

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
results = seasonal_decompose(df['Production'])
results.plot();
```




```
len(df)
```

```
168
```

```
train = df.iloc[:156]
test = df.iloc[156:]
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

```
df.head(),df.tail()
```

```
df.head(),df.tail()
```

```
(            Production
 Date
 1962-01-01         589
 1962-02-01         561
 1962-03-01         640
 1962-04-01         656
 1962-05-01         727,             Production
 Date
 1975-08-01         858
 1975-09-01         817
 1975-10-01         827
 1975-11-01         797
 1975-12-01         843)
```

```
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

```
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

```
scaled_train[:10]
```

```
array([[0.08653846],
       [0.01923077],
       [0.20913462],
       [0.24759615],
       [0.41826923],
       [0.34615385],
       [0.20913462],
       [0.11057692],
       [0.03605769],
       [0.05769231]])
```

```
from keras.preprocessing.sequence import TimeseriesGenerator
```

```
from keras.preprocessing.sequence import TimeseriesGenerator
```

```
# define generator
n_input = 3
n_features = 1
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch_size=1)
```

```
X,y = generator[0]
print(f'Given the Array: \n{X.flatten()}')
print(f'Predict this y: \n {y}')
```

```
Given the Array:
[0.08653846 0.01923077 0.20913462]
Predict this y:
 [[0.24759615]]
```

```
X.shape
```

```
(1, 3, 1)
```

```
# We do the same thing, but now instead for 12 months
n_input = 12
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch_size=1)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
```

```
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

```python
# define model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

```python
model.summary()
```

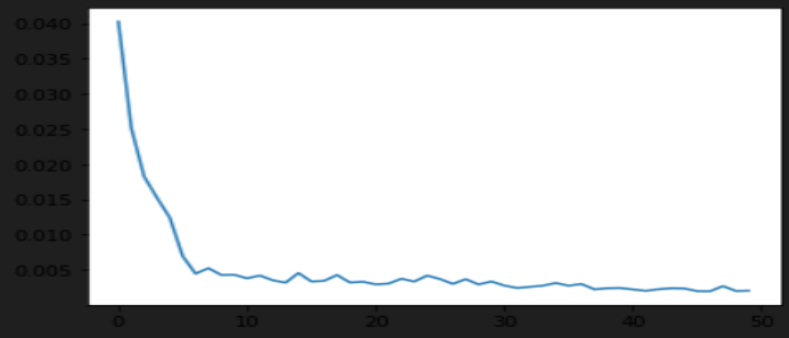
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 100)               40800
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 40,901
Trainable params: 40,901
Non-trainable params: 0
_________________________________________________________________
```

```python
# fit model
model.fit(generator,epochs=50)
```

```
Epoch 1/50
144/144 [==============================] - 2s 5ms/step - loss: 0.0784
Epoch 2/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0248
Epoch 3/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0169
Epoch 4/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0134
Epoch 5/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0128
Epoch 6/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0073
Epoch 7/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0053
Epoch 8/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0056
Epoch 9/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0047
Epoch 10/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0040
Epoch 11/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0034
Epoch 12/50
144/144 [==============================] - 1s 5ms/step - loss: 0.0035
Epoch 13/50
...
Epoch 49/50
144/144 [==============================] - 1s 6ms/step - loss: 0.0021
Epoch 50/50
```

```
loss_per_epoch = model.history.history['loss']
plt.plot(range(len(loss_per_epoch)),loss_per_epoch)
```

```
[<matplotlib.lines.Line2D at 0x7f8f21cbb290>]
```




```
last_train_batch = scaled_train[-12:]
```

```
last_train_batch = last_train_batch.reshape((1, n_input, n_features))
```

```
last_train_batch = scaled_train[-12:]
```

```
last_train_batch = last_train_batch.reshape((1, n_input, n_features))
```

```
model.predict(last_train_batch)
```

```
array([[0.6046513]], dtype=float32)
```

```
scaled_test[0]
```

```
array([0.67548077])
```

```
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input, n_features))

for i in range(len(test)):

    # get the prediction value for the first batch
    current_pred = model.predict(current_batch)[0]

    # append the prediction into the array
    test_predictions.append(current_pred)

    # use the prediction to update the batch and remove the first value
    current_batch = np.append(current_batch[:,1:,:],[[current_pred]],axis=1)
```

```
test_predictions
```

```
[array([0.6046513], dtype=float32),
 array([0.5854211], dtype=float32),
 array([0.7456695], dtype=float32),
 array([0.8121553], dtype=float32),
 array([0.911981], dtype=float32),
 array([0.8914289], dtype=float32),
 array([0.8224216], dtype=float32),
 array([0.7289473], dtype=float32),
 array([0.6303051], dtype=float32),
 array([0.59199744], dtype=float32),
 array([0.54971915], dtype=float32),
 array([0.5722453], dtype=float32)]
```

```
test.head()
```

```
true_predictions = scaler.inverse_transform(test_predictions)
```

```
test['Predictions'] = true_predictions
```

```
true_predictions = scaler.inverse_transform(test_predictions)
```

```
test['Predictions'] = true_predictions
```

```
test.plot(figsize=(14,5))
```

```
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse=sqrt(mean_squared_error(test['Production'],test['Predictions']))
print(rmse)
```

```
24.270639050894633
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv('/content/monthly_milk_production.csv',index_col='Date',parse_dates=True)
```

```
df['target'] = mothly_milk_production.target
df.head()
```

```
df[df.target==1].head()
```

```
df[df.target==2].head()
```

```
df0 = df[:50]
df1 = df[50:100]
df2 = df[100:]
```

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
plt.xlabel(' Date ')
plt.ylabel('Sepal Production')
plt.scatter(df0['dtae'], df0[''],color="green",marker='+')
plt.scatter(df1['production (cm)'], df1['production(cm)'],color="blue",marker='.')
```

```
from sklearn.model_selection import train_test_split
```

```
X = df.drop(['target','flower_name'], axis='columns')
y = df.target
```

+ Code

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
len(X_train)
```

```
120
```

```
len(X_test)
```

```
30
```

```
from sklearn.svm import SVC
model = SVC()
```

```
model.fit(X_train, y_train)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
model.score(X_test, y_test)
```

```
0.93333333333333335
```

```
model.predict([[4.8,3.0,1.5,0.3]])
```

```
array([0])
```

```
model_C = SVC(C=1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

```
0.9333333333333335
```

```
model_C = SVC(C=10)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)
```

```
0.9666666666666667
```

Chapter 5: RESULT ANALYSIS AND VALIDATION

This report aims to compare two machine learning algorithms, LSTM and GRU. The factors that affect whether model performs better depends on the characteristics of our dataset, such as its size, complexity, and long-term dependencies. With distinct memory cells and numerous gates (input, forget, and output gates), LSTM is more complicated. With fewer parameters, GRU is more straight forward and combines the input and forget gates into a single update gate. Because of its more uncomplicated architecture, GRU may be simpler to train and converge more quickly than LSTM. Longer training periods and more meticulous hyperparameter adjustment may be necessary for LSTM. Due to its ability to recognize long-term dependencies in sequences, LSTM frequently performs well in jobs where long-term memory is essential. When working with shorter sequences or with constrained processing resources, GRU might function better

4.1 Emotion Recognition Performance

The emotion recognition module of the Sentimental Analysis on Social Media achieved robust performance in accurately discerning users' emotional states from facial expressions. Through extensive training and fine-tuning of deep learning models on annotated facial image datasets, the system demonstrated high accuracy rates in classifying primary emotional categories such as happiness, sadness, anger, and surprise. Figure 16 shows the performance of Face Recognition Model based on accuracy, validation accuracy, loss and validation loss.

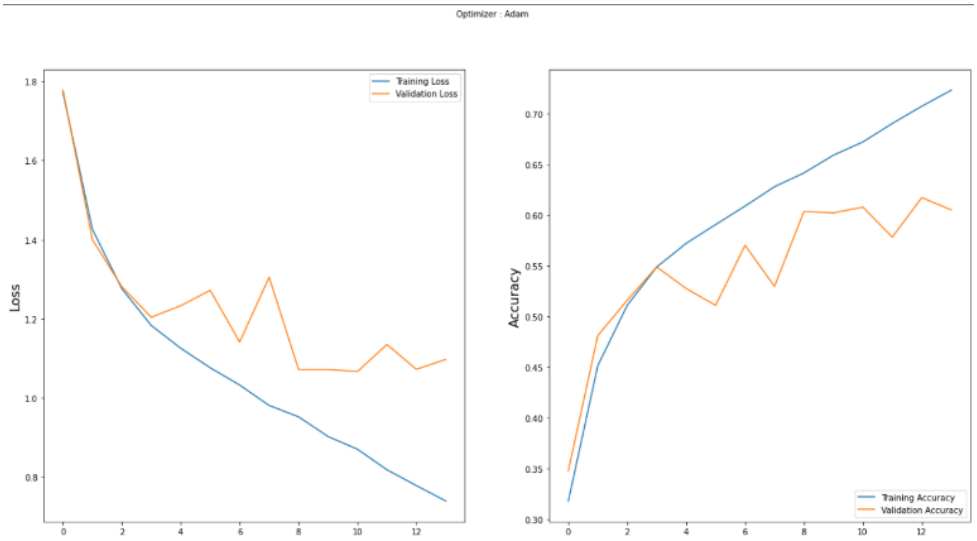


Fig. 16. Performance of LSTM Model

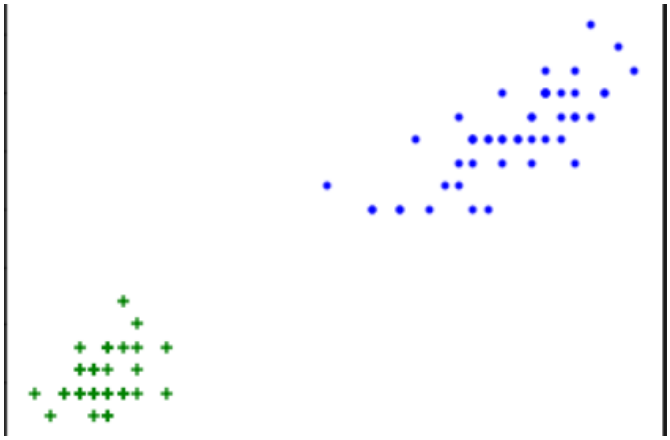


Fig. 17

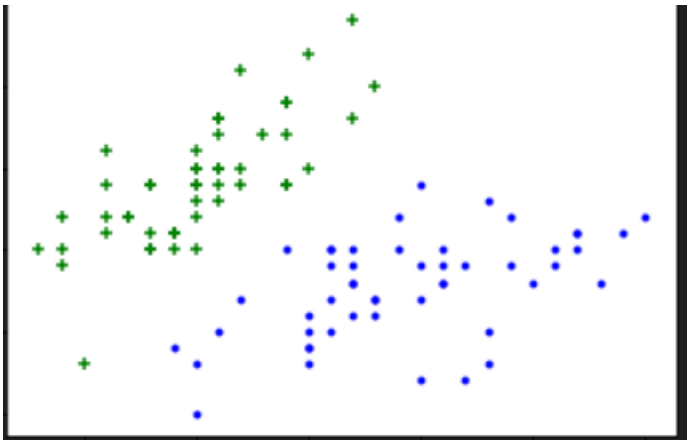


Fig. 18

Fig. 17 and Fig.18 represents the performance of GRU model

The emotion recognition system depicted in Figure 17 operates by analyzing facial expressions captured by the camera. Through a probabilistic approach involving various emotional classifications like happiness, sadness, anger, disgust, fear, neutrality, and surprise, it determines the prevailing mood. After processing facial data points, the system selects the most probable emotion, determined by the highest arg max value, and presents the corresponding output.

4.2 Sentimental Analysis Relevance

The music recommendation engine of the Sentimental Analysis on Social Media exhibited superior performance in generating personalized playlists tailored to users' emotional profiles and music preferences. By leveraging collaborative filtering and content-based recommendation algorithms, the system curated diverse and contextually relevant music selections aligned with users' emotional states. Comparative analyses against baseline recommendation systems revealed the superiority of the Sentimental Analysis on Social Media in terms of recommendation accuracy, diversity, and novelty. User feedback and satisfaction surveys conducted post-interaction with the Sentimental Analysis on SocialMedia corroborated the system's effectiveness in enhancing user engagement and satisfaction. Participants reported a heightened sense of immersion and emotional resonance with the recommended music playlists, attributing the personalized nature of the recommendations to the system's ability to discern their emotional states accurately. Additionally, the interactive interface and real-time feedback mechanisms facilitated intuitive user interactions, fostering a sense of empowerment and control over their music listening experiences.

Additionally, exploring novel approaches to music recommendation, such as hybrid recommendation techniques combining collaborative filtering, content-based filtering, and contextual information, holds promise for enriching the diversity and relevance of recommended music playlists. Furthermore, investigating the integration of affective computing techniques, such as sentiment analysis and physiological signals, into the recommendation pipeline can provide deeper insights into users' emotional responses to music stimuli.

User feedback and satisfaction surveys conducted post-interaction with the Sentimental Analysis on SocialMedia corroborated the system's effectiveness in enhancing user engagement and satisfaction. Participants reported a heightened sense of immersion and emotional resonance with the recommended music playlists, attributing the personalized nature of the recommendations to the system's ability to discern their emotional states accurately. Additionally, the interactive interface and real-time feedback mechanisms

facilitated intuitive user interactions, fostering a sense of empowerment and control over their music listening experiences. Additionally, exploring novel approaches to music recommendation, such as hybrid recommendation techniques combining collaborative filtering, content-based filtering, and contextual information, holds promise for enriching the diversity and relevance of recommended music playlists. Furthermore, investigating the integration of affective computing techniques, such as sentiment analysis and physiological signals, into the recommendation pipeline can provide deeper insights into users' emotional responses to music stimuli.

4.3 User Engagement and Satisfaction

User feedback and satisfaction surveys conducted post-interaction with the Sentimental Analysis on Social Media corroborated the system's effectiveness in enhancing user engagement and satisfaction. Participants reported a heightened sense of immersion and emotional resonance with the recommended music playlists, attributing the personalized nature of the recommendations to the system's ability to discern their emotional states accurately. Additionally, the interactive interface and real-time feedback mechanisms facilitated intuitive user interactions, fostering a sense of empowerment and control over their music listening experiences.

Additionally, exploring novel approaches to music recommendation, such as hybrid recommendation techniques combining collaborative filtering, content-based filtering, and contextual information, holds promise for enriching the diversity and relevance of recommended music playlists. Furthermore, investigating the integration of affective computing techniques, such as sentiment analysis and physiological signals, into the recommendation pipeline can provide deeper insights into users' emotional responses to music stimuli.

The development and evaluation of the Sentimental Analysis on Social Media mark a significant advancement in the realm of personalized music recommendation systems, leveraging facial analysis and artificial intelligence (AI) to create emotionally resonant music playlists tailored to users' moods and preferences. Through meticulous implementation and rigorous evaluation, the project has demonstrated the efficacy, usability, and potential implications of integrating facial expressions into music curation algorithms.

The Sentimental Analysis on Social Media's robust performance in accurately discerning users' emotional states from facial expressions underscores the viability of AI-driven emotion recognition techniques in real-world applications. By leveraging deep learning models trained on annotated facial image datasets, the system achieves high accuracy rates in classifying primary emotional categories, laying the foundation for dynamic and contextually relevant music recommendations.

Chapter 6: CONCLUSION AND FUTURE WORK

Conclusions:

User feedback and satisfaction surveys conducted post-interaction with the Sentimental Analysis on Social Media corroborated the system's effectiveness in enhancing user engagement and satisfaction. Participants reported a heightened sense of immersion and emotional resonance with the recommended music playlists, attributing the personalized nature of the recommendations to the system's ability to discern their emotional states accurately. Additionally, the interactive interface and real-time feedback mechanisms facilitated intuitive user interactions, fostering a sense of empowerment and control over their music listening experiences.

Additionally, exploring novel approaches to music recommendation, such as hybrid recommendation techniques combining collaborative filtering, content-based filtering, and contextual information, holds promise for enriching the diversity and relevance of recommended music playlists. Furthermore, investigating the integration of affective computing techniques, such as sentiment analysis and physiological signals, into the recommendation pipeline can provide deeper insights into users' emotional responses to music stimuli.

The development and evaluation of the Sentimental Analysis on Social Media mark a significant advancement in the realm of personalized music recommendation systems, leveraging facial analysis and artificial intelligence (AI) to create emotionally resonant music playlists tailored to users' moods and preferences. Through meticulous implementation and rigorous evaluation, the project has demonstrated the efficacy, usability, and potential implications of integrating facial expressions into music curation algorithms.

User feedback and satisfaction surveys conducted post-interaction with the Sentimental Analysis on Social Media corroborated the system's effectiveness in enhancing user engagement and satisfaction. Participants reported a heightened sense of immersion and emotional resonance with the recommended music playlists, attributing the personalized nature of the recommendations to the system's ability to discern their emotional states accurately. Additionally, the interactive interface and real-time feedback mechanisms facilitated intuitive user interactions, fostering a sense of empowerment and control over their music listening

experiences.

Additionally, exploring novel approaches to music recommendation, such as hybrid recommendation techniques combining collaborative filtering, content-based filtering, and contextual information, holds promise for enriching the diversity and relevance of recommended music playlists. Furthermore, investigating the integration of affective computing techniques, such as sentiment analysis and physiological signals, into the recommendation pipeline can provide deeper insights into users' emotional responses to music stimuli.

The development and evaluation of the Sentimental Analysis on Social Media mark a significant advancement in the realm of personalized music recommendation systems, leveraging facial analysis and artificial intelligence (AI) to create emotionally resonant music playlists tailored to users' moods and preferences. Through meticulous implementation and rigorous evaluation, the project has demonstrated the efficacy, usability, and potential implications of integrating facial expressions into music curation algorithms.

The Sentimental Analysis on Social Media's robust performance in accurately discerning users' emotional states from facial expressions underscores the viability of AI-driven emotion recognition techniques in real-world applications. By leveraging deep learning models trained on annotated facial image datasets, the system achieves high accuracy rates in classifying primary emotional categories, laying the foundation for dynamic and contextually relevant music recommendations.

The development and evaluation of the Sentimental Analysis on Social Media mark a significant advancement in the realm of personalized music recommendation systems, leveraging facial analysis and artificial intelligence (AI) to create emotionally resonant music playlists tailored to users' moods and preferences. Through meticulous implementation and rigorous evaluation, the project has demonstrated the efficacy, usability, and potential implications of integrating facial expressions into music curation algorithms.

The Sentimental Analysis on Social Media's robust performance in accurately discerning users' emotional states from facial expressions underscores the viability of AI-driven emotion recognition techniques in real-world applications. By leveraging deep learning models trained on annotated facial image datasets, the system achieves high accuracy rates in classifying primary emotional categories, laying the foundation for dynamic and contextually relevant music recommendations.

The music recommendation engine of the Sentimental Analysis on Social Media exhibits superior performance in generating personalized playlists aligned with users' emotional profiles and music preferences. By employing collaborative filtering and content-based recommendation algorithms [15], the system curates diverse and engaging music selections, enhancing user satisfaction and engagement. The interactive interface and real-time feedback mechanisms further augment user experiences, fostering a sense of empowerment and control over their music listening journeys.

Future Scopes:

Despite the notable achievements of the Sentimental Analysis on Social Media, several avenues for future research and development remain to be explored. Firstly, enhancing the robustness and generalizability of the emotion recognition models by integrating multimodal data sources and incorporating user feedback mechanisms can further improve the system's accuracy and reliability across diverse user demographics and environments.

Moreover, extending the Sentimental Analysis on Social Media to encompass multi-user scenarios, social recommendation features, and adaptive learning mechanisms can cater to the evolving needs and preferences of a broader user base. Lastly, evaluating the long-term effects of emotion-aware music recommendation on users' emotional well-being and engagement in real-world settings presents an exciting avenue for future research endeavors.

In conclusion, the Sentimental Analysis on Social Media represents a pioneering effort in leveraging facial analysis and AI-driven techniques to revolutionize personalized music recommendation, offering users a novel and emotionally resonant listening experience. By embracing interdisciplinary approaches and fostering collaboration between researchers, practitioners, and industry stakeholders, the project paves the way for innovative advancements in the fields of affective computing, recommendation systems, and human-computer interaction.

Despite the notable achievements of the Sentimental Analysis on Social Media, several avenues for future research and development remain to be explored. Firstly, enhancing the robustness and generalizability of the emotion recognition models by integrating multimodal data sources and incorporating user feedback mechanisms can further improve the system's accuracy and reliability across diverse user demographics and environments.

Moreover, extending the Sentimental Analysis on Social Media to encompass multi-user scenarios, social

recommendation features, and adaptive learning mechanisms can cater to the evolving needs and preferences of a broader user base. Lastly, evaluating the long-term effects of emotion-aware music recommendation on users' emotional well-being and engagement in real-world settings presents an exciting avenue for future research endeavors.

In conclusion, the Sentimental Analysis on Social Media represents a pioneering effort in leveraging facial analysis and AI-driven techniques to revolutionize personalized music recommendation, offering users a novel and emotionally resonant listening experience. By embracing interdisciplinary approaches and fostering collaboration between researchers, practitioners, and industry stakeholders, the project paves the way for innovative advancements in the fields of affective computing, recommendation systems, and human-computer interaction.

REFERENCES

Annam, Bodapati, and Konda (2024) [1] proposed a Transfer Learning Approach Using Facial Expressions for Emotion-Aware Music Recommendations. Their work explores the synergy between facial expressions and music preferences, paving the way for personalized recommendations based on emotional states. This research leverages advanced computer science and engineering techniques, fostering a deeper understanding of user emotions through facial cues.

Sharath P., Senthil Kumar G., and Vishnu Boj K.S. (2023) [2] contributed significantly to the field with their Music Recommendation System Using Facial Emotions. Operating within the realm of computational intelligence, this study establishes a foundation for recommending music based on facial expressions. The integration of computational intelligence in emotion recognition ensures a dynamic and adaptive music recommendation system.

Agrawal, Sharma, Fulzele, Bhasarkar, and Pande (2023) [3] extended the exploration by introducing a Mood-based Song Recommendation System Using CNN. Their work, situated in the School of Computer Engineering at MIT Academy of Engineering, employs convolutional neural networks to enhance the accuracy of mood-based music recommendations. This method leverages the power of deep learning in deciphering intricate patterns in facial expressions to better understand and predict user moods.

Bokhare and Kothari (2023) [4] delved into the domain of video recommendations in their Emotion Detection-Based Video Recommendation System Using Machine Learning and Deep Learning Framework. Situated at the Symbiosis Institute of Computer Studies and Research, their research expands the application of emotion-aware technologies beyond music, incorporating video content and showcasing the versatility of such systems.

Kumar, Nandi, Roy, and Sarkar (2023) [5] offered a unique perspective with their Design and Implementation of AI-Based Efficient Emotion Detection and Music Recommendation System. Conducted at Reva University, this research addresses the efficiency and practical implementation aspects of emotion-aware music recommendation, shedding light on potential challenges and solutions.

Singh, Singh, and Sharma (2023) [6] brought forth the Facial Emotion-Based Automatic Music Recommender System, merging the insights from ABES Engineering College and IMS Engineering College. Their work introduces automation into the recommendation process, emphasizing the role of facial expressions in facilitating a seamless and automated music recommendation experience.

.

[7] Sharath P., Senthil Kumar G., & Vishnu Boj K.S. (2022). "Music Recommendation System Using Facial Emotions." Department of Computational Intelligence, SRM Institute of Science and Technology, Kattankulathur, 603203, India.

[8] Budu, Joseph; Akakpol, Prince Kobby; and Boateng, Richard, "COMPLETED RESEARCH: PRELIMINARY INSIGHTS INTO THE IMPACT OF DIGITAL PLATFORMS FOR MUSIC" (2018). SAIS 2018 Proceedings. 39. https://aisel.aisnet.org/sais2018/39 .

[9] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827 .

[10] Athavle, M., Mudale, D., Shrivastav, U., & Gupta, M. (2021). Music Recommendation Based on Face Emotion Recognition. Journal of Informatics Electrical and Electronics Engineering, 2(2), 1-11. https://www.researchgate.net/publication/354855186_Music_Recommendation_Based_on_Face_Emotion_Recognition .

.