

# MOODSPHERE

A N E M O T I O N B A S E D M U S I C R E C O M M E N D E R



# AGENDA

- Team Member Roles and Responsibilities
- Improvements made from Professor Feedback
- Project Description
- Team-Work Agreement
- Personas
- MVP
- Technologies
- Algorithms
- Diagrams
- Sprint 2 Recap
- Product Backlog
- Sprint 3 Backlog
- Metrics
- Retrospective
- Sprint 4
- Project Demo - Sprint 3 (current sprint)
- GitHub link
- Live Application Demo



# TEAM



**Bhavik Chopra**

Data Scientist



**Dhyey Dave**

Lead Developer



**Krushil Sheladiya**

UI/UX Developer



**Mahesh Nakka**

Backend developer/QA

# TEAM



**Nisarg Bhuva**

Scrum Master/QA



**Shane Parmar**

Backend Developer/Product  
Owner



**Urmil Trivedi**

Architect/Developer



**Vijay Devkate**

ML Engineer/ UI Designer



# IMPROVEMENTS FROM FEEDBACK + CHANGES

- **Project Description Format**(Slide 5)
- **Modified more Personas**(Slide 10-15)
- **Modified MVP Explanation**(Slide 16-19)
- **Changed and Added Diagrams**(Slide 25-31)
- **Added Sprint 2 Recap**(Slide 32-34)
- **Updated Product Backlog**(Slide 35-41)
- **Changed User Stories**(Slide 43-48)
- **Added Stories Completed and Not Completed**(Slide 49-53)
- **Added Metrics**(Slide 54-71)



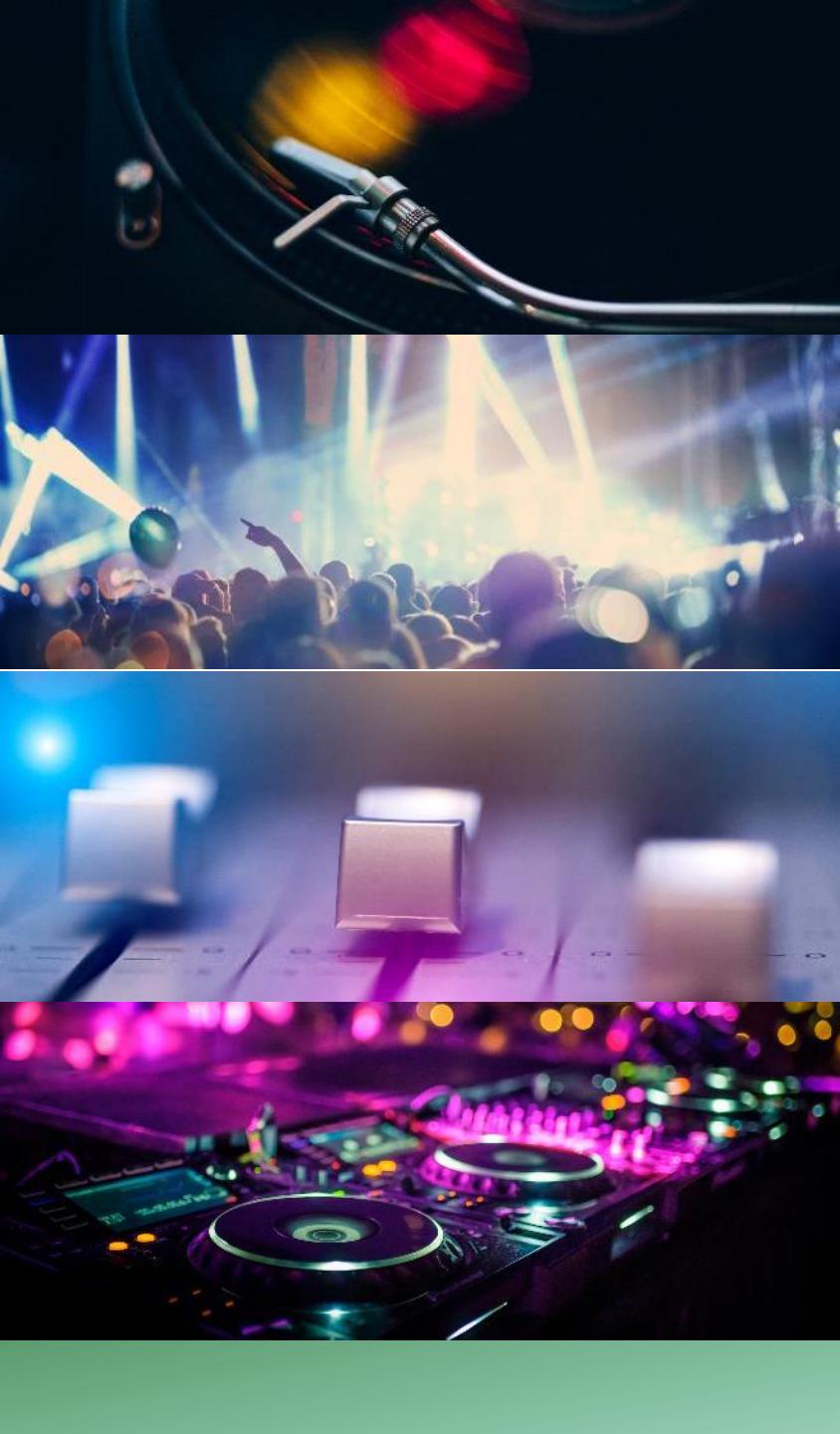
This symbol represents the **Improvements** made.



# PROJECT DESCRIPTION

Project Name:	MoodSphere
Team:	DevDynasty
Project Description:	<p>Using facial expression detection techniques and machine learning algorithms in real time to understand the user's mood.</p> <p><b>for</b> the music enthusiasts seeking a personalized and emotionally resonant listening experience,</p> <p><b>who</b> wants to listen to their music based on their current emotional state <b>the MoodSphere</b></p> <p><b>is an</b> application that takes out the hassle of going through all the options in traditional music applications to hear the right music for their mood, that employs facial recognition and CNNs to analyze your real-time emotional expressions captured through the camera.</p> <p><b>unlike</b> traditional music recommendation systems solely relying on user preferences, MoodSphere distinguishes itself by incorporating facial emotion analysis.</p> <p><b>our</b> application dynamically adjusts music recommendations as your emotional expressions evolve allowing users to experience enhanced recommendations within the application.</p>





# PROBLEM STATEMENT

Conventional music streaming services frequently rely on general recommendation algorithms that ignore emotional relevance, which leaves users' emotional demands unmet. The Emotion-Based Music Recommender using Facial Recognition and Convolutional Neural Networks (CNN) project aims to combine state-of-the-art computer vision techniques in order to address this. The system uses CNNs and facial recognition technologies to accurately assess users' facial expressions for emotional indicators. The goal of this integration of sophisticated machine learning algorithms with visual data is to provide consumers with individualized music recommendations that are in line with their current emotional states, improving their listening experience as a whole.

# TEAM-WORK AGREEMENT



# CS691 - Teamwork Agreement ( Team Dev Dynasty )

We at Dev Dynasty follow the seamless and triumphant completion in the project, where we are committed to the following principles and expectations. Team Collaboration, Proactiveness are classified by Involvement, Awareness, Task Allocation, and Time management. As, we members of Team Dev Dynasty take initiative based on individual skills, prioritise the capstone project success to ensure the completion.

## INVOLVEMENT

- As the involvement is necessary we came to a point to meet In-person twice in 10 days which will make the team more stronger and collaborative.
- We shall agree that during the meet time or discussion about any crucial part all are requested to put their opinions and comments on what will be the best for the team to succeed in the outcome.
- The three moto's i.e. Trustworthiness, Truthfulness and Openness based on this value every individual shall or can have diverse perspectives, provide equal opportunity and a new ideology which can be developed towards great success of the project, instead of blaming people when issues occur.
- If the task assigned to the teammate gets undone or its tough to get complete then he must report and communicate to teammates via whatsapp.

## AWARENESS

- As performing tasks if a teammate posts something which he's unaware the other if are available can react to it as quickly as possible.
- All should be aware that the discussion took place during the meeting hours and it's no one's responsibility to take care of each other; some exceptions might be taken in a state of medical emergency or sickness.
- We will communicate on every second day to keep updated about the task distributed and its individual responsibilities to openly ask for help if needed rather wasting time which he can't persist off. This can make a smooth process for the project to be on the right track.
- All the deliverables or the task based on theories will only be performed via Google doc which can be given access before submitting the final copy.
- Following the task distributions, planning for the upcoming sprint, next meeting times all will be followed on a single platform i.e. Jira.

## Task allocation:

- Meeting via online or In-person for a daily standup which will eventually help teammates to get updated on every task.
- The project work should be distributed according to the individual knowledge and skill which can provide actual results and can help in problem solving.
- Based on the roles assigned if the teammates failed to perform correctly and failed to meet the deadline the scrum master has the right of decision making and make sure the task gets fulfilled or the teammate completes it in the next sprint.
- As respecting the privacy of every teammate, timing contact unless it's necessary regarding the project work.

## Time Management

- The scrum master will make sure that the meeting links have been reached to everyone and everyone is readily available on same time for better coordination.
- We shall coordinate on each other's schedules to maintain consistency on working projects followed by a track on it after every discussion.

Team Member's	Email IDs
Urmil Trivedi	ut24256n@pace.edu
Bhavik Chopra	bc04992n@pace.edu
Dhyey Dave	dd28633n@pace.edu
Krushil Sheladiya	ks84830n@pace.edu
Shane Parmar	sp91003n@pace.edu
Vijay Devkate	vd19129n@pace.edu
Mahesh Nakka	mn01776n@pace.edu
Nisarg Bhuva	nb95325n@pace.edu

# PERSONAS



# PERSONA 1



Elena

Technical Project Manager

MOODSPHERE

## Background:

Elena is a 32-year-old project manager in a tech company, often under high stress and pressure of completing project to deliver best outcomes.

## Emotional Profile:

Given her high-stress role, Elena values an app that can detect subtle shifts in her mood through her interactions — be it the pace at which she scrolls or the time of day — and adjust the music without her needing to manually search for tracks. This anticipates her needs and saves time, allowing her to remain focused on her work or relaxation.

## Preferred Genres:

Classical for relaxation, soft rock for mood uplift, ambient for focus.

## Interests:

Yoga, meditation, and wellness podcasts.



# PERSONA 2



Max:

The Energetic Fitness Enthusiast

MOODSPHERE

## Background:

Max is a 27-year-old fitness trainer reliant on energetic music during workouts to stay energized and motivated.

## Emotional Profile:

Seeks dynamic, high-tempo music for workouts and calming tunes for cooldowns. The app would suggest serene and tranquil music, aiding in the transition from high activity to relaxation. The app could potentially measure the duration of the workout to time the cooldown music perfectly.

## Preferred Genres:

Upbeat EDM for workouts, soft instrumentals for cooldowns.

## Interests:

High-intensity interval training, marathons, and sports nutrition.



# PERSONA 3



Lia:

Remote Worker

MOODSPHERE

## Background:

Lia is a 20-year-old freelance writer who works from home on content on client's demand

## Emotional Profile:

Her choice of music transcends mere background ambience, acting as a vital catalyst for her focus and creativity. It's this profound relationship with music that not only entertains but also significantly nurtures her artistic expression and emotional health. Each melody plays a specific role in her day, meticulously chosen to align with her feelings, thereby transforming her workspace into a sanctuary for inspiration and emotional balance.

## Preferred Genres:

Lo-fi hip hop, smooth jazz, and classical.

## Interests:

Creative writing, reading, and interior design.



# PERSONA 4



Sam:

The Commuter

MOODSPHERE

## Background:

Sam is a 35-year-old commuter who has long commutes and uses music to mentally prepare or unwind

## Emotional Profile:

Sam needs energizing music in the morning and relaxing music in the evening. Sam also seeks music that can serve as a personal soundtrack to the rhythm of the city during his commutes, providing a sense of connection and vitality. In moments of stress or fatigue, he looks for tracks that can offer solace and a mental escape, helping to rejuvenate his spirit.

## Preferred Genres:

Pop and indie for energy, acoustic for relaxation.

## Interests:

Travel podcasts, audiobooks, and city walking tours.



# PERSONA 5



MOODSPHERE

Zara:

An Aspiring Chef

## Background:

Zara is a 22-year-old culinary student who enjoys cooking with music which makes her mood calm and focused.

## Emotional Profile:

Zara seeks music that matches the energy of her cooking sessions. Zara enjoys music that keeps her kitchen lively and her mind sharp while she cooks. She also likes songs that go well with the different kinds of food she makes, making her cooking feel more special.

## Preferred Genres:

Upbeat Latin for cooking, classical for baking.

## Interests:

World cuisines, food blogging, and cooking shows.



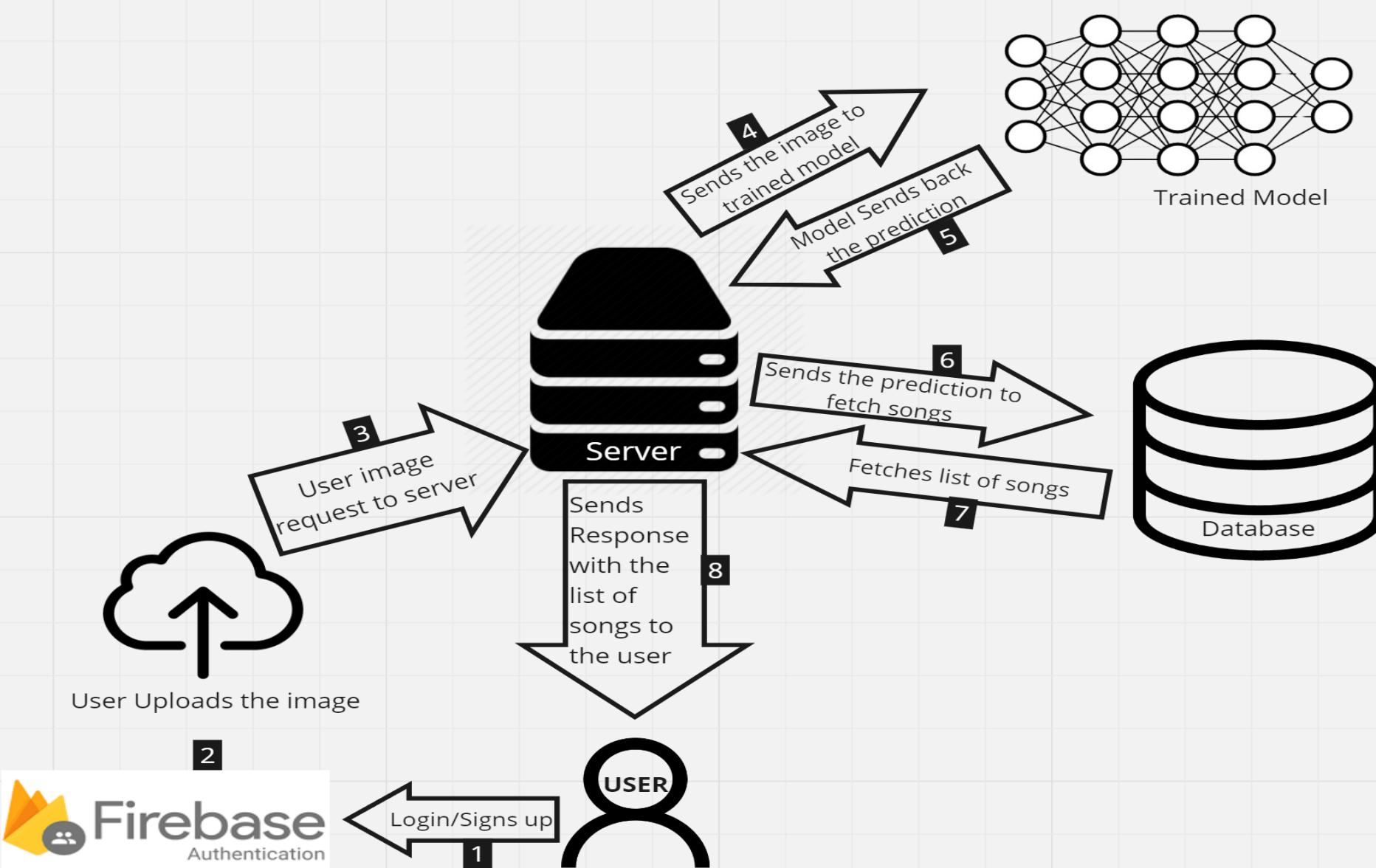
# MINIMAL VIABLE PRODUCT(MVP)



# MINIMUM VIABLE PRODUCT

- Web application with user emotion prediction and song recommendation functionality.
- Allowing user to create an account in the application.
- Interactive UI to display user uploaded image.
- CNN model to predict user's emotion on image input.
- API to accept user image in request and send the model's prediction and the list of songs from static database in response.
- Server to handle the above API along with two different endpoints to send songs based on genre and artist.
- Model and backend connection to predict emotions.

# MVP DIAGRAM



# MVP Experiment Canvas

Project name:

MoodSphere

Made by:

Dev  
Dynasty

Start date/time: 01/23/24

End date/time: 05/07/24

## 1. Problem

- Users often struggle to find music that aligns with their current emotional state.
- Difficulty in understanding users' emotions through digital platforms.

## 2. Existing Alternatives

- Manual selection of songs based on mood or genre.
- Basic recommendation algorithms that suggest music based on listening history rather than current emotional state.

## 3. Solution

- A web application that predicts the user's emotion based on an uploaded image and recommends songs accordingly.
- Features include account creation, an interactive UI for displaying the uploaded image, and a CNN model for emotion prediction.
- The application will also offer an API to process the image, predict the emotion, and send back a list of song recommendations from a static database.

## 4. Unique Value Proposition

- Real-time emotion recognition from user-uploaded images to tailor music recommendations, enhancing the user experience by aligning music with their current emotional state.

## 5. Key Metrics

- User engagement rates (time spent on the application, number of songs listened to, etc.).
- Accuracy of emotion prediction and user satisfaction with song recommendations.

## 6. Channels

- Social media marketing targeting music enthusiasts and tech-savvy individuals.
- Collaborations with music forums and communities.

## 7. User Perspective Workflow

- Account Setup: Users sign up or log in, entering basic information.
- Image Upload: Users upload an image reflective of their current mood.
- Waiting for Prediction: The system processes the image, and users wait briefly.
- Receiving Recommendations: Based on the emotion detected, users are presented with a list of song recommendations.
- Engaging with Music: User will be allowed to click on song's external music links.

## 8. System Perspective Workflow

- Manage Accounts: Securely handle user registrations and logins.
- Process Image: Store and validate the uploaded image before processing.
- Predict Emotion: Use a CNN model to identify the user's emotion from the image.
- Retrieve Songs: Select songs matching the predicted emotion from a categorized database.
- Serve Content: Offer additional endpoints for music exploration and ensure a responsive UI for user interaction.

## 9. Learnings and Insights

### 9a. Technical Learning

- AI and Machine Learning: Gain expertise in CNN models for emotion recognition, including data handling and model optimization.
- Web Development: Learn full-stack development, focusing on responsive UI design, API integration, and user data management.
- Deployment: Experience in deploying AI models in a web environment, emphasizing scalability and performance.

### 9b. Personal and Professional Growth

- Project Management: Navigating the complexities of bringing an AI-driven product from concept to market enhances skills in project management, team collaboration, and agile development methodologies.
- Adaptability and Problem-Solving: The challenges encountered during development and post-launch adjustments will foster adaptability and innovative problem-solving skills.
- Networking and Collaboration: Engaging with users, investors, and other stakeholders offers opportunities for networking and may lead to collaborations or partnerships.

## 9c. Design and UX

UX/UI Design for AI Applications: Understanding how users interact with AI features and what design elements enhance their experience. This includes simplifying complex processes (like emotion prediction) into user-friendly interfaces.



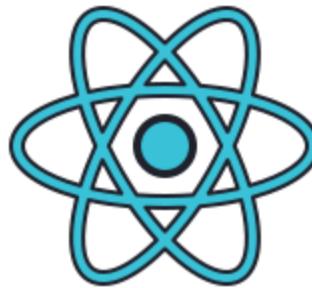
# TECHNOLOGIES



# TECHNOLOGIES



TensorFlow



React



Firebase



Python



HTML5



CSS3



Flask  
Flask



Jira

# TECHNOLOGIES DESCRIPTION



TensorFlow will be the foundation for building the emotion detection engine of our music recommender. Main applications of using TensorFlow will be Model Development and Real-Time Inference



React serves as the cornerstone for constructing the user interface (UI) of your music recommender app. Main applications of using React will be UI Design and Data Visualization



It is a suite of cloud-based services from Google that simplifies building mobile and web applications by providing features like: Authentication, Database Cloud Functions Hosting Analytics



Python supports OOP principles, allowing you to structure your code using classes and objects. This promotes code reusability, modularity, and maintainability, especially when dealing with complex data structures like user profiles and music recommendations.

# TECHNOLOGIES DESCRIPTION



HTML5 Defines the structure and content of web pages. Creates the basic building blocks of web pages using elements like headings, paragraphs, lists, images, and forms.



CSS3 controls the presentation of web pages, applying visual styles like fonts, colors, layout, and animations. It defines styles that are applied to HTML elements, controlling their appearance.



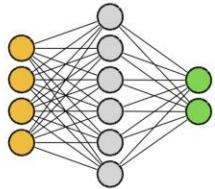
Flask

High-level web framework built in Python, simplifying the development process for building web applications. Can be employed to create the backend of your application, handling data processing, routing, and interacting with Jira



Project management software specifically designed for agile teams, facilitating task tracking, issue reporting, and collaboration.

# ALGORITHMS AND EXPLANATION



Convolutional neural networks (CNNs) are a form of deep neural network that are frequently used in computer vision tasks like object detection, picture segmentation, and image recognition. When the term "CNN algorithm" is used, it usually refers to CNNs.

## ResNet-50

PRETRAINED MODEL

ResNet-50 is a deep convolutional neural network with 50 layers, renowned for its use of residual connections that aid in training very deep networks. Because of its outstanding performance in tasks like object identification, picture segmentation, and image classification, it has been frequently used in computer vision applications.

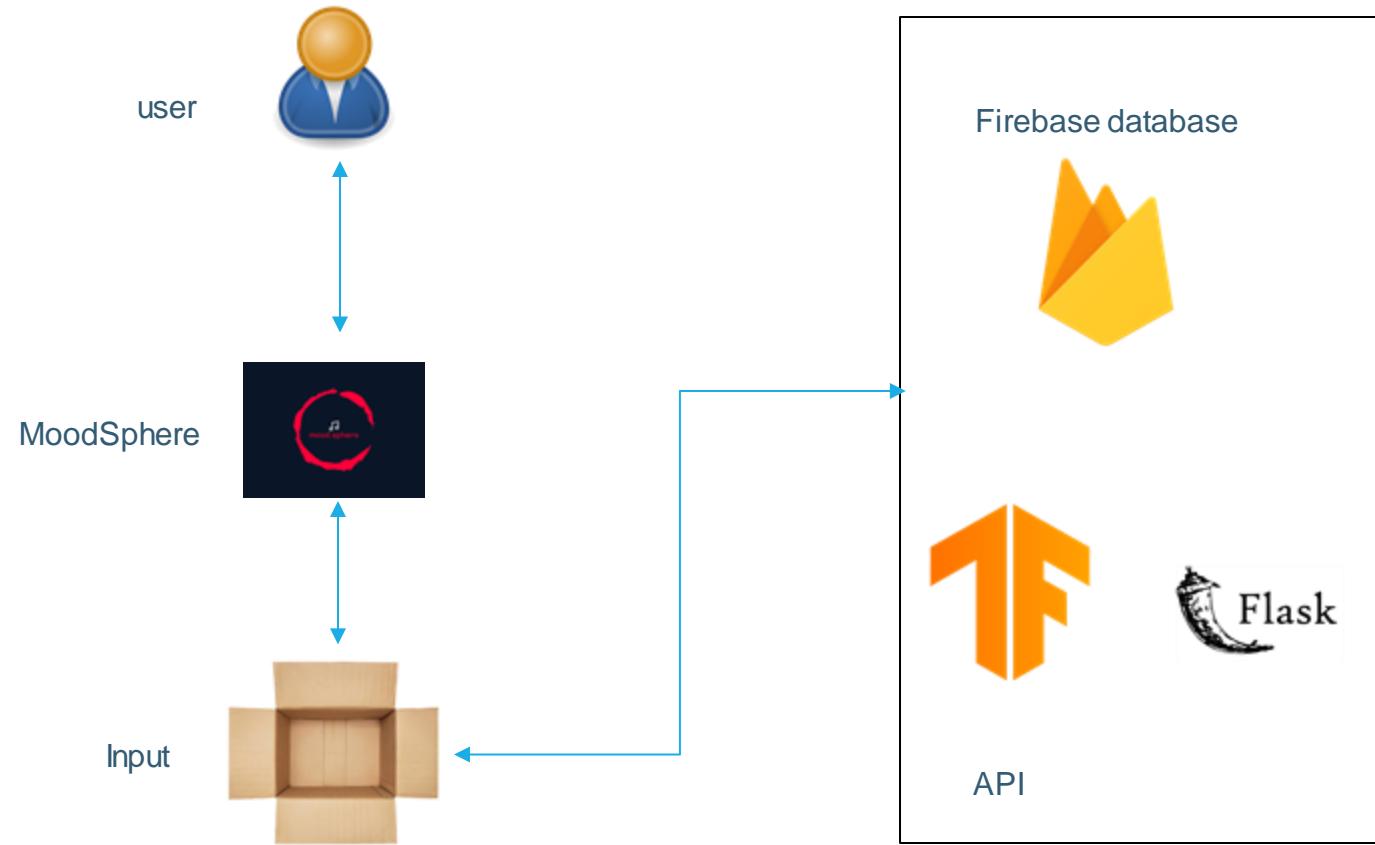


Keras is a popular Python framework that makes neural network construction and training (simple and straightforward). It provides an elevated interface to deep learning frameworks like as TensorFlow and Theano, facilitating swift model deployment and development.

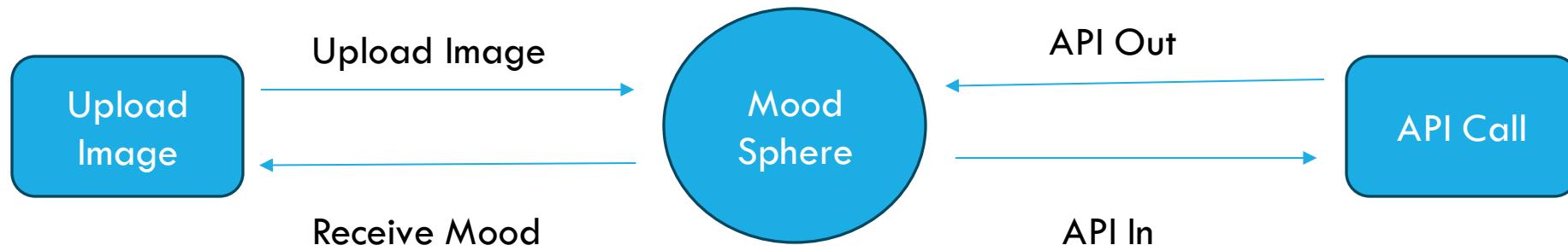
# DIAGRAMS



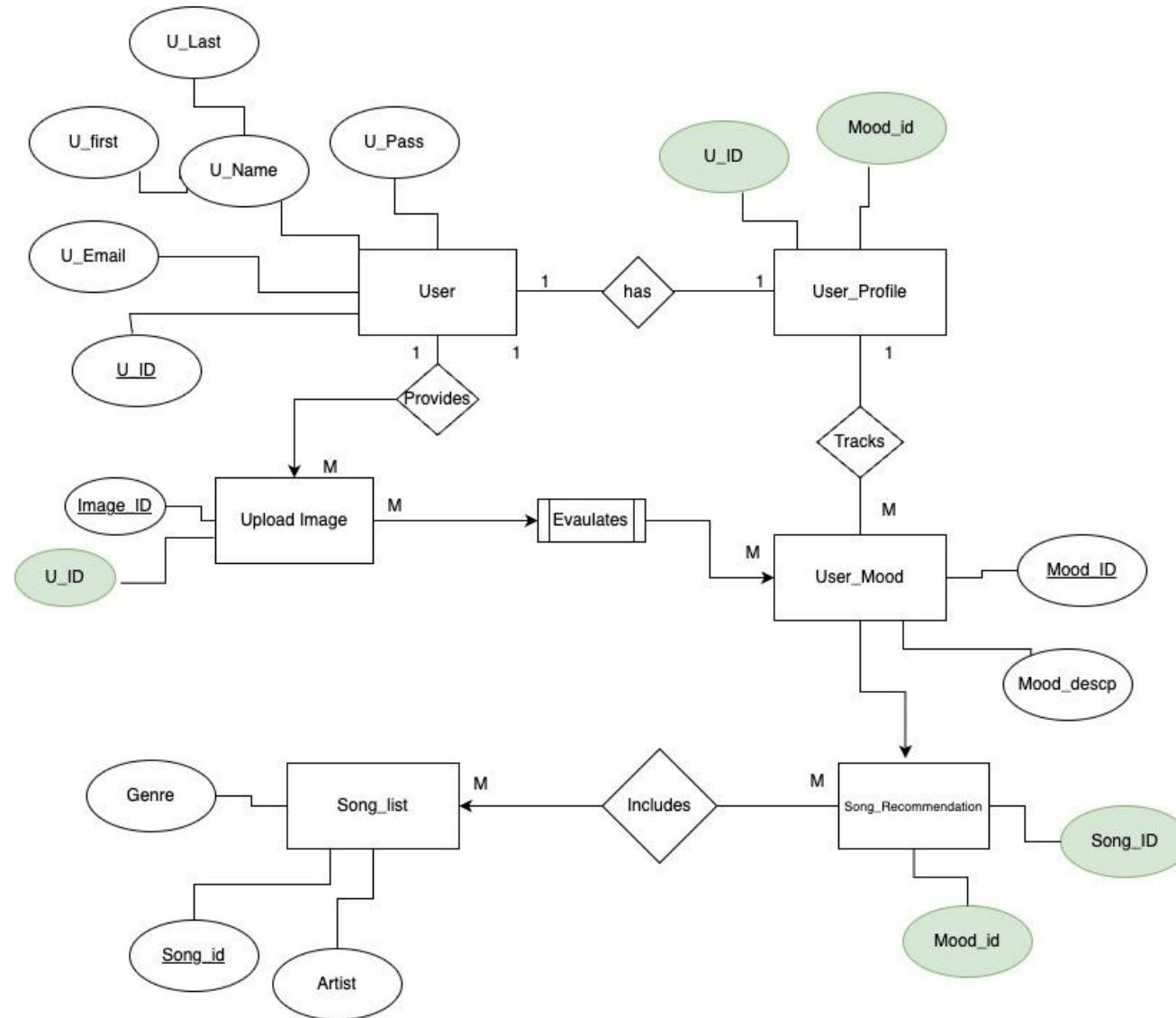
# Architecture Diagram



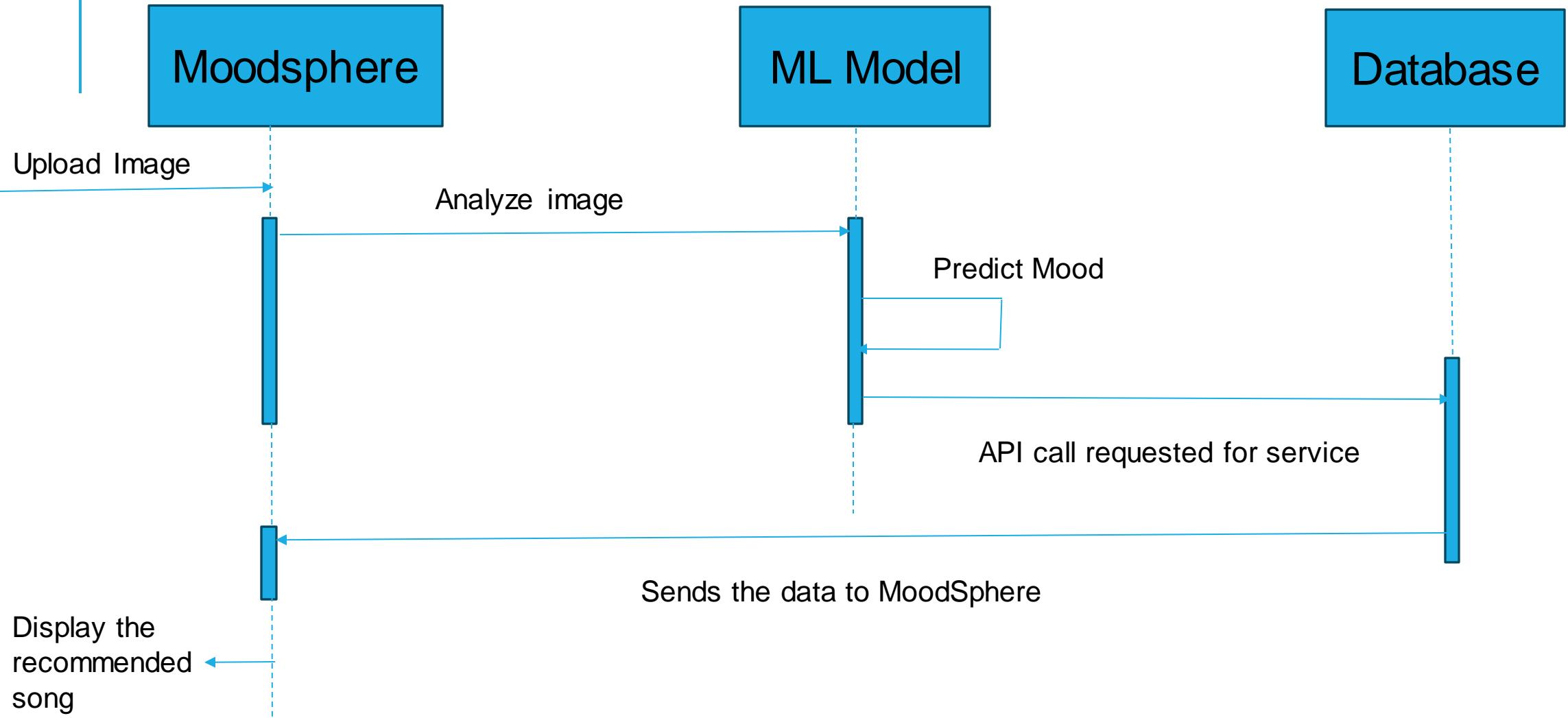
# Context Diagram



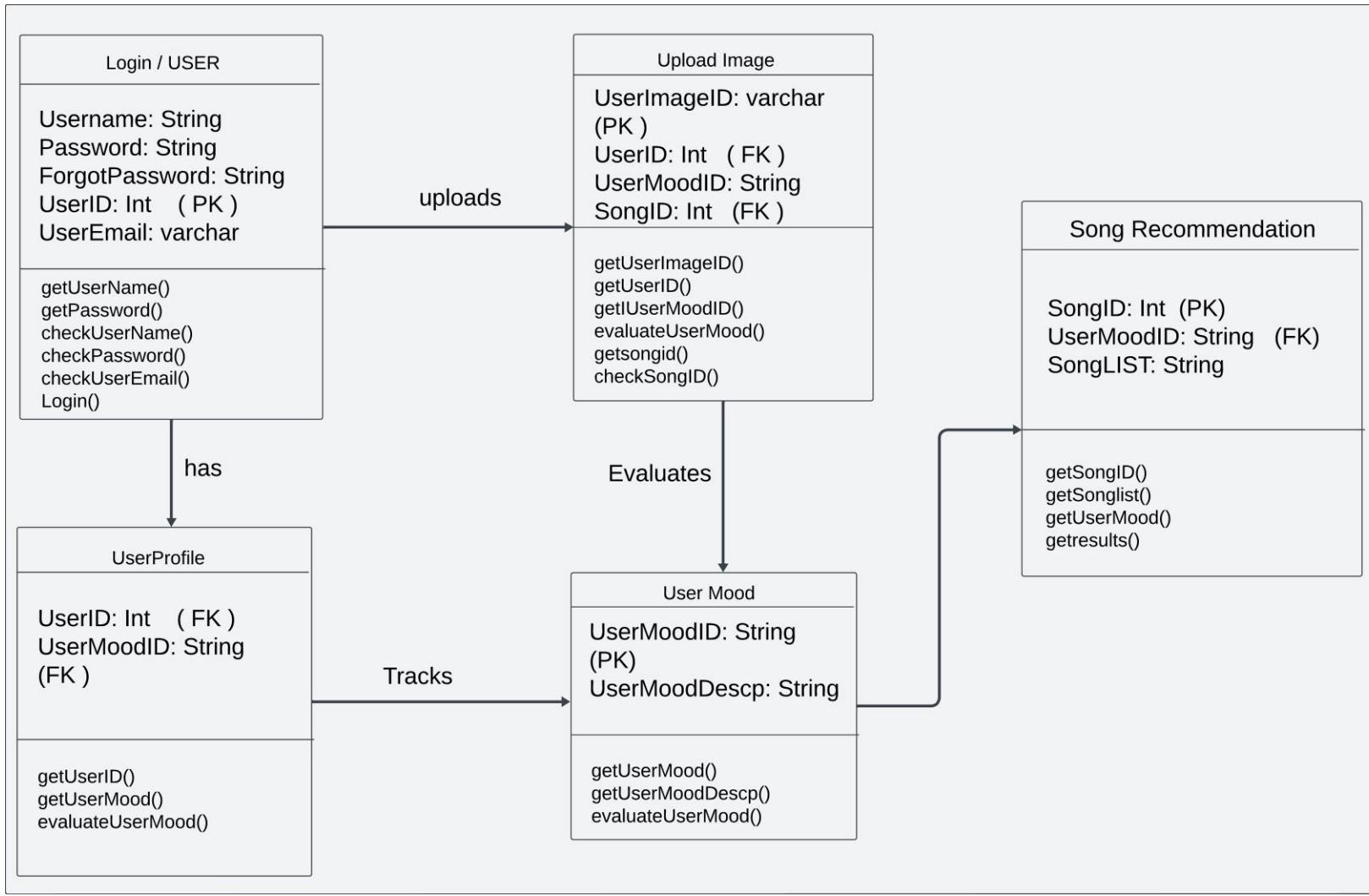
# ENTITY RELATIONSHIP DIAGRAM



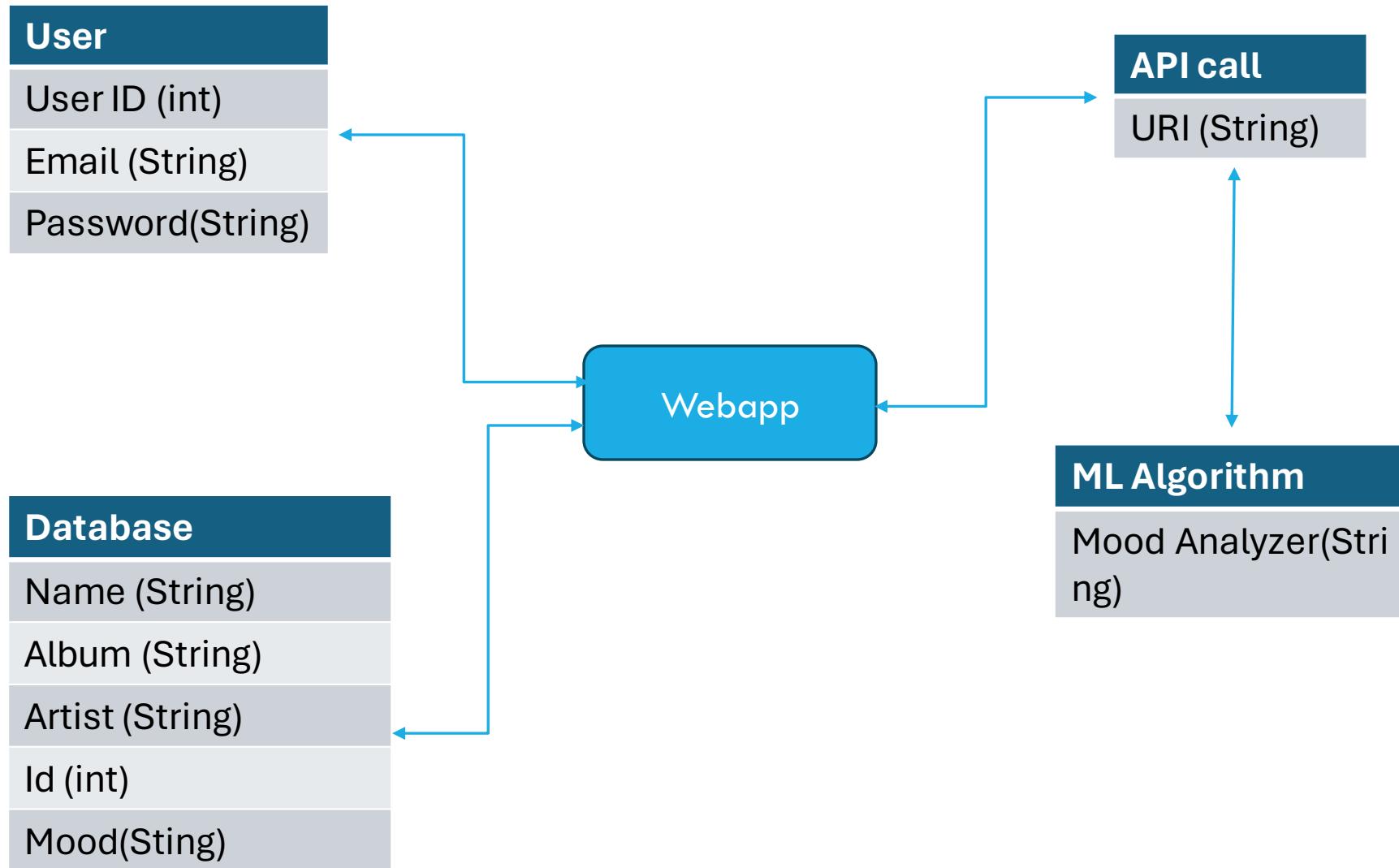
# Sequence Diagram



# STATE DIAGRAM



# Class Diagram



# SPRINT 2 RECAP



<b>MS_4</b>	8	As a user, I want to receive personalized music recommendations based on my current mood.  Criteria, The system should retrieve a list of songs associated with the predicted mood from a CSV file. Songs should be relevant and appropriate for the predicted mood.	Done	<b>Sprint-2</b>
<b>MS_9</b>	5	As a user, I want to be able to explore different moods and receive recommendations tailored to each mood.  Criteria, Users should have the option to select different moods and receive corresponding song recommendations. Each mood selection should provide a unique set of recommended songs.	Done	<b>Sprint-2</b>
<b>MS_13</b>	5	As a user, I want to be able to view and edit my profile information.  Criteria, Users can view and update their profile information (e.g., name, email, profile picture). Profile editing should be straightforward and secure.	Done	<b>Sprint-2</b>
<b>MS_14</b>	5	As a user, I want to be able to sign up for an account to access personalized features and settings.  Criteria, Users can create a new account by providing necessary information (e.g., email, password). Account creation should include email verification for security purposes.	Done	<b>Sprint-2</b>
<b>MS_18</b>	5	As a user, I want the application to provide a diverse range of song recommendations for each mood, including different genres and styles based on the popularity of the song.  Criteria, Recommended songs should cover a variety of genres and musical preferences. Users should have options to explore different music styles within each mood category with the recommendation based on popularity.	Done	<b>Sprint-2</b>
<b>MS_19</b>	5	As a user, I want the application to be secure and protect my privacy when uploading pictures or interacting with the site.  Criteria, The application should use HTTPS to encrypt data transmission. User data should be stored securely and not shared with third parties without consent.	Done	<b>Sprint-2</b>
<b>MS_26</b>	3	As a user, I want to view my current mood status without any song recommendations.  Criteria, Users can navigate to a separate page or section to view their current mood status. The mood status should be displayed visually (e.g., emoticons) along with a text description	Done	<b>Sprint-2</b>



- **Login/Signup Page**  
Created login and signup page users to easily login into the app
- **Firebase Connection**  
Used fireauth to validate the credential for users and for database
- **Figma Prototype**  
Created wireframe designs to finalize the theme
- **Model Training**  
Trained and Tuned the model to increase the accuracy of prediction



## Home Page

Created Home Page with four main navigation buttons

## MVP Completion

Were able to complete the main functionality of uploading image and recommending songs

## Mood Status Page

User can upload an image, predicts a mood, returns a song list with links

## Get Songs By Artist Page

Providing sorted list of songs categorized by artist



# PRODUCT BACKLOG



Task ID	Story Points	User Story	Status	Sprint
MS_4	3	As a user, I want to receive personalized music recommendations based on my current mood.  Criteria, The system should retrieve a list of songs associated with the predicted mood from a CSV file. Songs should be relevant and appropriate for the predicted mood.	Done	Sprint - 2
MS_9	5	As a user, I want to be able to explore different moods and receive recommendations tailored to each mood.  Criteria, Users should have the option to select different moods and receive corresponding song recommendations. Each mood selection should provide a unique set of recommended songs.	Done	Sprint - 2
MS_13	5	As a user, I want to be able to view and edit my profile information.  Criteria, Users can view and update their profile information (e.g., name, email, profile picture). Profile editing should be straightforward and secure.	Done	Sprint - 2
MS_14	5	As a user, I want to be able to sign up for an account to access personalized features and settings.  Criteria, Users can create a new account by providing necessary information (e.g., email, password). Account creation should include email verification for security purposes.	Done	Sprint - 2
MS_18	5	As a user, I want the application to provide a diverse range of song recommendations for each mood, including different genres and styles based on the popularity of the song.  Criteria, Recommended songs should cover a variety of genres and musical preferences. Users should have options to explore different music styles within each mood category with the recommendation based on popularity.	Done	Sprint - 2



Task ID	Story Points	User Story	Status	Sprint
MS_19	5	As a user, I want the application to be secure and protect my privacy when uploading pictures or interacting with the site.  Criteria, The application should use HTTPS to encrypt data transmission. User data should be stored	Done	Sprint - 2
MS_26	3	As a user, I want to view my current mood status without any song recommendations.  Criteria, Users can navigate to a separate page or section to view their current mood status. The mood status should be displayed visually (e.g., emoticons) along with a text description	Done	Sprint - 2
MS_1	3	As a user, I want to be able to upload a picture representing my current mood From the Already Saved Images in the gallery  Criteria, The user can upload an image file. Supported image formats include JPG, PNG, and GIF.	In Progress	Sprint - 3
MS_2	5	As a user, I want the system to analyze the uploaded picture to determine my current mood More Precisely.  Criteria, The system should use a CNN model to extract features from the image. The model should classify the mood into predefined categories (e.g., happy, sad, angry) More Precisely	Done	Sprint - 3
MS_3	2	As a user, I want to see the predicted mood based on the uploaded picture.  Criteria, The predicted mood should be displayed to the user. The mood prediction should be visually represented (e.g., emoticons).	Done	Sprint - 3



Task ID	Story Points	User Story	Status	Sprint
MS_6	3	As a user, I want to be able to navigate the application easily and intuitively.  Criteria, The application should have clear navigation menus and buttons. Users should be able to easily switch between different sections of the application.	Done	Sprint - 3
MS_7	5	As a user, I want the application to be responsive and work well on different devices, including desktops, tablets, and smartphones.  Criteria, The application should adapt to different screen sizes and resolutions. All features should be accessible and functional on various devices.	Done	Sprint - 3
MS_8	5	As a user, I want to be able to search for specific songs or artists.  Criteria, Users can search for songs or artists using a search bar. Search results should be relevant and displayed in a clear manner.	In Progress	Sprint - 3
MS_12	5	As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.  As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.	Done	Sprint - 3
MS_16	3	As a user, I want to be able to access external links associated with each recommended song.  Criteria, The list of songs should include external links in one column, allowing users to access additional information or listen to the song.	Done	Sprint - 3



Task ID	Story Points	User Story	Status	Sprint
MS_22	3	As a user, I want the application to display the mood category along with each recommended song.  Criteria, Each recommended song entry should include the corresponding mood category. Mood categories should be clearly labeled and easy to distinguish.	Done	Sprint - 3
MS_23	5	As a user, I want the application to provide recommendations based on my current mood, even if I'm not logged in or don't have an account.  Criteria, Users should be able to access mood-based recommendations without requiring authentication or account creation.	In Progress	Sprint - 3
MS_5	5	Criteria, Users should be able to access mood-based recommendations without requiring authentication or account creation.  Criteria, Users should have the option to add songs to a personal playlist. The playlist should persist across sessions for registered users.	To Do	Sprint - 4
MS_10	3	As a user, I want to be able to create and customize multiple playlists for different moods or occasions.  Criteria, Users can create new playlists and give them custom names. Users can add or remove songs from playlists.	To Do	Sprint - 4
MS_11	5	As a user, I want the application to provide additional information about recommended songs  Criteria, Users can access additional information about recommended songs by clicking on them. Information should be presented in a clear and concise manner.	To Do	Sprint - 4



Task ID	Story Points	User Story	Status	Sprint
MS_15	5	As a user, I want to be able to provide feedback on recommended songs or report any issues encountered while using the application.	To Do	Sprint - 4
		Criteria, Users should have a feedback mechanism to submit comments or report problems		
MS_17	3	As a user, I want to be able to clear my mood selection and start over if I'm not satisfied with the recommended songs.	To Do	Sprint - 4
		Criteria, Users should have the option to clear their mood selection and return to the initial state.Clearing the mood selection should reset the recommended songs.		
MS_20	3	As a user, I want the application to provide clear instructions and guidance on how to use its features effectively.	To Do	Sprint - 4
		Criteria, The application should include help documentation or tooltips to explain its functionality.		
MS_21	3	As a user, I want the application to display loading indicators or progress bars to indicate when mood analysis and song retrieval are in progress.	To Do	Sprint - 4
		Criteria, Users should see visual cues indicating that the application is processing their request>Loading indicators should be displayed prominently and disappear once processing is complete.		
MS_24	3	As a user, I want the application to store uploaded pictures for future reference.	To Do	Sprint - 4
		Criteria, Uploaded pictures should be saved securely in the user's account.Users should be able to access their stored pictures from their profile or settings.		



Task ID	Story Points	User Story	Status	Sprint
MS_25	3	As a user, I want the ability to delete uploaded pictures if needed.	Done	Sprint - 4
		Criteria, Users can delete uploaded pictures from their account. Deletion should be confirmed with a confirmation prompt to prevent accidental deletion.		



# SPRINT-3 BACKLOG



# SPRINT-3 USER STORIES & ACCEPTANCE CRITERIA



Task ID	Story Points	User Story	Status	Sprint
MS_1	3	As a user, I want to be able to upload a picture representing my current mood From the Already Saved Images in the gallery	In Progress	Sprint - 3
		Criteria, The user can upload an image file. Supported image formats include JPG, PNG, and GIF.		
MS_2	5	As a user, I want the system to analyze the uploaded picture to determine my current mood More Precisely.	Done	Sprint - 3
		Criteria, The system should use a CNN model to extract features from the image. The model should classify the mood into predefined categories (e.g., happy, sad, angry) More Precisely		
MS_3	2	As a user, I want to see the predicted mood based on the uploaded picture.	Done	Sprint - 3
		Criteria, The predicted mood should be displayed to the user. The mood prediction should be visually represented (e.g., emoticons).		
MS_6	3	As a user, I want to be able to navigate the application easily and intuitively.	Done	Sprint - 3
		Criteria, The application should have clear navigation menus and buttons. Users should be able to easily switch between different sections of the application.		
MS_22	3	As a user, I want the application to display the mood category along with each recommended song.	Done	Sprint - 3
		Criteria, Each recommended song entry should include the corresponding mood category. Mood categories should be clearly labeled and easy to distinguish.		
MS_23	5	As a user, I want the application to provide recommendations based on my current mood, even if I'm not logged in or don't have an account.	In Progress	Sprint - 3
		Criteria, Users should be able to access mood-based recommendations without requiring authentication or account creation.		

Task ID	Story Points	User Story	Status	Sprint
MS_7	5	As a user, I want the application to be responsive and work well on different devices, including desktops, tablets, and smartphones.	Done	Sprint - 3
		Criteria, The application should adapt to different screen sizes and resolutions. All features should be accessible and functional on various devices.		
MS_8	5	As a user, I want to be able to search for specific songs or artists.	In Progress	Sprint - 3
		Criteria, Users can search for songs or artists using a search bar. Search results should be relevant and displayed in a clear manner.		
MS_12	5	As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.	Done	Sprint - 3
		As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.		
MS_16	3	As a user, I want to be able to access external links associated with each recommended song.	Done	Sprint - 3
		Criteria, The list of songs should include external links in one column, allowing users to access additional information or listen to the song.		

# TESTCASES FOR SPRINT-3 BACKLOG



Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
MS_1	only Upload Picture	Verify Successful Upload of JPG Image	1. Navigate to the picture upload feature. 2. Select a JPG image file to upload. 3. Click on the upload button.	JPG File	The JPG image is uploaded successfully.	Image is sucessfully passed to the CNN model	pass
MS_1	only Upload Picture	Verify Successful Upload of PNG Image	1. Navigate to the picture upload feature. 2. Select a PNG image file to upload. 3. Click on the upload button.	PNG File	The PNG image is uploaded successfully.	Image is sucessfully passed to the CNN model	pass
MS_2	Analyze Uploaded Picture	Verify Mood Analysis for Uploaded Picture	1. Upload a picture representing a known mood (e.g., happy, sad). 2.Allow the system to analyze the uploaded picture.	JPG/PNG File	The system accurately determines the mood of the uploaded picture.	Model Predicts accurately	pass
MS_3	View Predicted Mood	Verify Display of Predicted Mood	1.Upload a picture representing a known mood (e.g., happy, sad). 2. Check the predicted mood displayed on the screen.	JPG/PNG File	The predicted mood is displayed accurately and visually represented (e.g., emoticon).	Model Predicts accurately and displays	pass
MS_6	Navigate Application Easily	Verify Clear Navigation Menus	1. Navigate through different sections of the application.	Navigation with Mobile/Desktop	Navigation menus and buttons are clear and intuitive.	Navigation buttons are working	pass

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
MS_7	Ensure Responsive Design	Verify Application Responsiveness on Desktop	1. Access the application from a desktop device.	Desktop Browser Testing	The application adapts smoothly to different screen sizes on desktop devices.	Can clearly able to navigate	pass
			2. Resize the browser window.				
MS_7	Ensure Responsive Design	Verify Application Responsiveness on Mobile	Access the application from a mobile device.	Mobile Device Testing	The application adapts smoothly to different screen sizes on mobile devices.	Can clearly able to navigate and see the results	pass
MS_12	Sort Recommended Songs Based on Criteria	Verify Sorting by Popularity	1. Check the order of the recommended songs.	Sort by Popularity	The recommended songs are sorted based on their popularity ratings.	Can see the List of songs based on popularity	pass
			2. Check the sorting option for popularity from the available criteria.				
MS_16	Access External Links for Recommended Songs	Verify Access to External Links	1. Click on a recommended song. 2. Check for the presence of external links associated with the song.	External Link Functionality	External links for additional information about the song are accessible.	Can be taken to the external application	pass
MS_22	Display Mood Category with Recommended Songs	Verify Mood Category Displayed with Each Recommended Song	1. Navigate to the section displaying recommended songs. 2. Check if each song entry includes the corresponding mood category.	Verify Mood Category Displayed	Each recommended song entry displays the mood category it belongs to.	Can see the Mood of the displayed song	pass

# COMPLETED STORIES IN SPRINT-3



Task ID	Story Points	User Story	Status	Sprint
<b>MS_2</b>	5	As a user, I want the system to analyze the uploaded picture to determine my current mood More Precisely.  Criteria, The system should use a CNN model to extract features from the image.The model should classify the mood into predefined categories (e.g., happy, sad, angry) More Precisely	Done	Sprint - 3
<b>MS_3</b>	2	As a user, I want to see the predicted mood based on the uploaded picture.  Criteria, The predicted mood should be displayed to the user.The mood prediction should be visually represented (e.g., emoticons).	Done	Sprint - 3
<b>MS_6</b>	3	As a user, I want to be able to navigate the application easily and intuitively.  Criteria, The application should have clear navigation menus and buttons.Users should be able to easily switch between different sections of the application.	Done	Sprint - 3
<b>MS_7</b>	5	As a user, I want the application to be responsive and work well on different devices, including desktops, tablets, and smartphones.  Criteria, The application should adapt to different screen sizes and resolutions.All features should be accessible and functional on various devices.	Done	Sprint - 3

Task ID	Story Points	User Story	Status	Sprint
MS_12	5	As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.	Done	Sprint - 3
		As a user, I want the application to provide an option to sort recommended songs based on popularity or other criteria.		
MS_16	3	As a user, I want to be able to access external links associated with each recommended song.	Done	Sprint - 3
		Criteria, The list of songs should include external links in one column, allowing users to access additional information or listen to the song.		
MS_22	3	As a user, I want the application to display the mood category along with each recommended song.	Done	Sprint - 3
		Criteria, Each recommended song entry should include the corresponding mood category. Mood categories should be clearly labeled and easy to distinguish.		

# INCOMPLETED STORIES IN SPRINT-3



Task ID	Story Points	User Story	Status	Sprint
MS_1	3	As a user, I want to be able to upload a picture representing my current mood From the Already Saved Images in the gallery	In Progress	Sprint - 3
		Criteria, The user can upload an image file. Supported image formats include JPG, PNG, and GIF.		
MS_8	5	As a user, I want to be able to search for specific songs or artists.	In Progress	Sprint - 3
		Criteria, Users can search for songs or artists using a search bar. Search results should be relevant and displayed in a clear manner.		
MS_23	5	As a user, I want the application to provide recommendations based on my current mood, even if I'm not logged in or don't have an account.	In Progress	Sprint - 3
		Criteria, Users should be able to access mood-based recommendations without requiring authentication or account creation.		

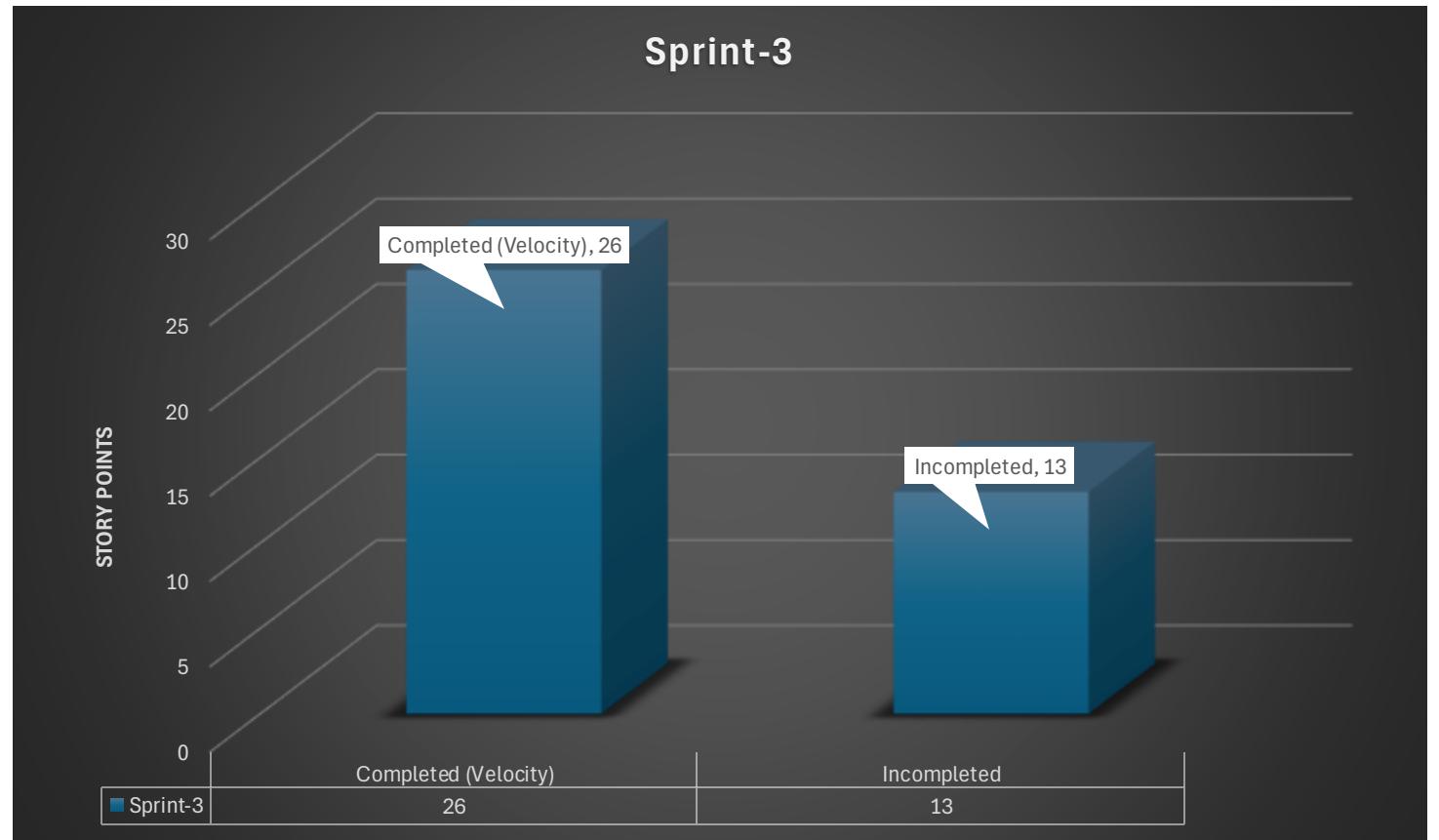
# METRICS



# SPRINT -3 TEAM VELOCITY



TEAM VELOCITY FOR SPRINT- 3 IS  
(NUMBER OF COMPLETED STORIES) = 26



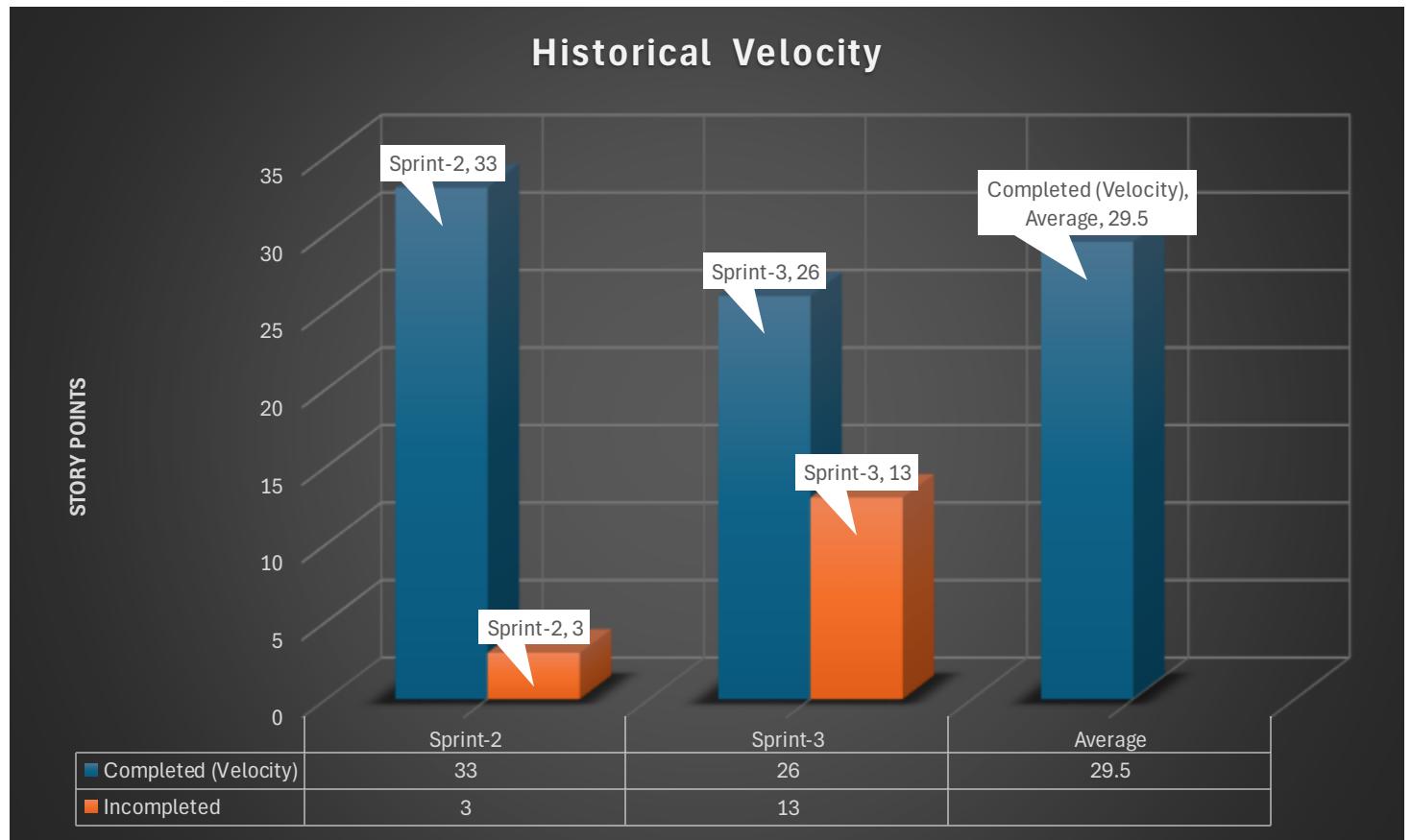
# TEAM'S HISTORICAL VELOCITY



AVERAGE VELOCITY =

$$\frac{\text{SUM OF VELOCITIES FOR COMPLETED SPRINTS}}{\text{NUMBER OF COMPLETED SPRINTS}}$$

HISTORICAL VELOCITY OF MOODSPHERE IS 29.5



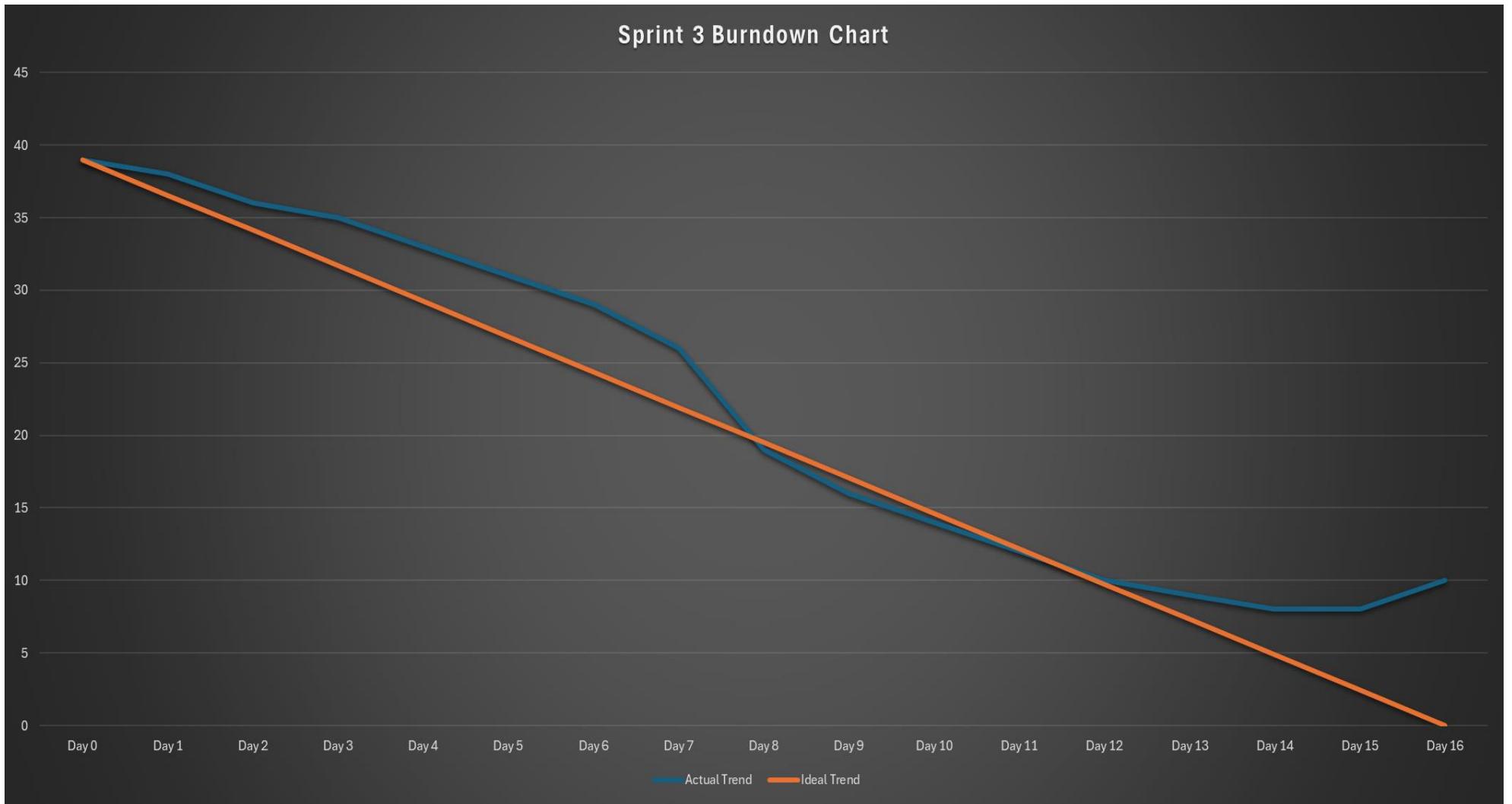
# BURNDOWN CHARTS



# SPRINT-3 BURNDOWN CHART



## Sprint 3 Burndown Chart

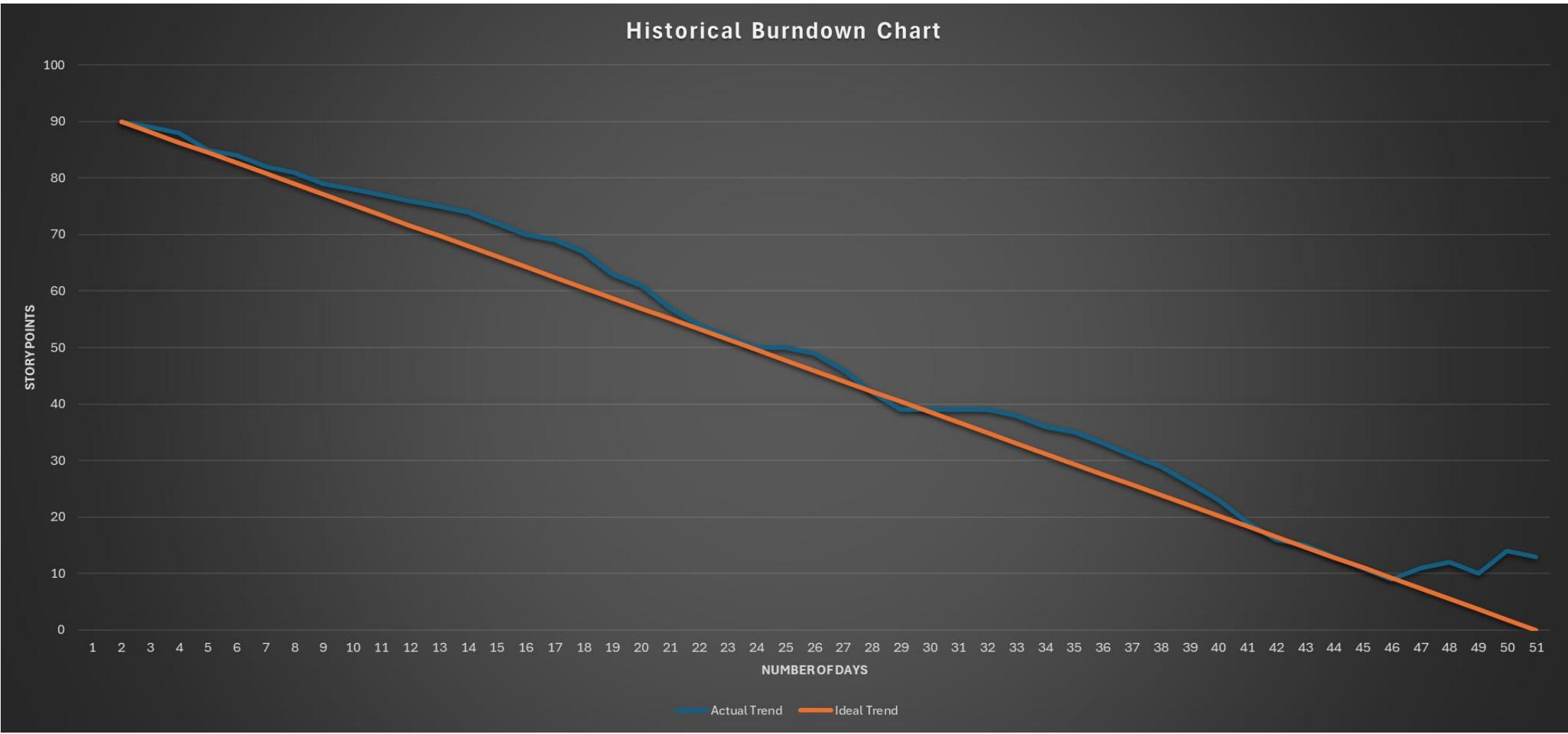


# HISTORICAL BURNDOWN CHART

Historical burndown chart = Sum(Sprint 1  
+ Sprint 2 + Sprint 3)



## Historical Burndown Chart



# AVERAGE BURNDOWN CHART

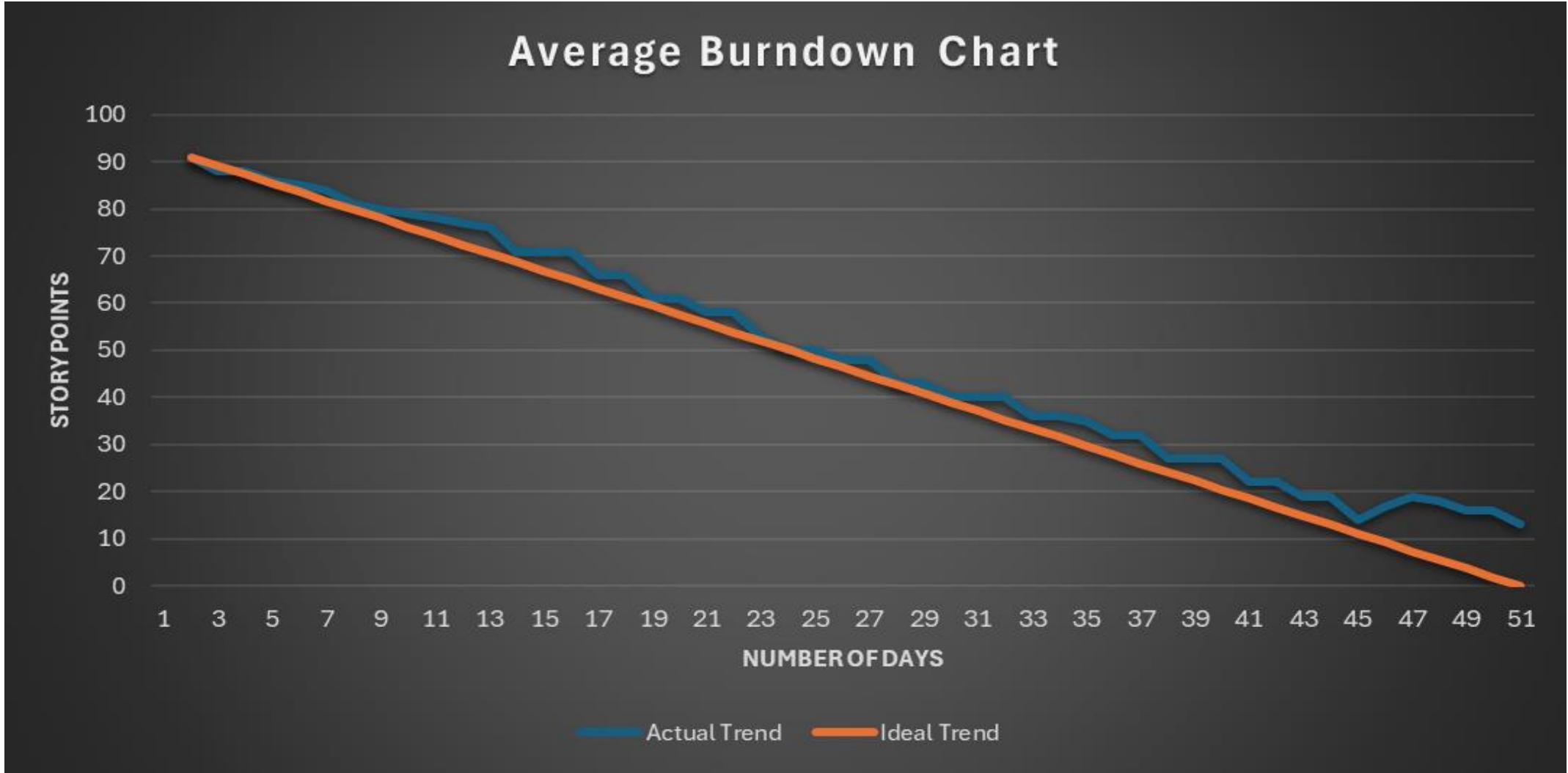
Average Burndown Rate = Total Variance / Total Number of Intervals

Where:

- Total Variance =  $\Sigma$  (Ideal Burndown - Actual Burndown) for all intervals
- Total Number of Intervals = Number of intervals (e.g., days, weeks) during the project duration. In our Case it is 3days.



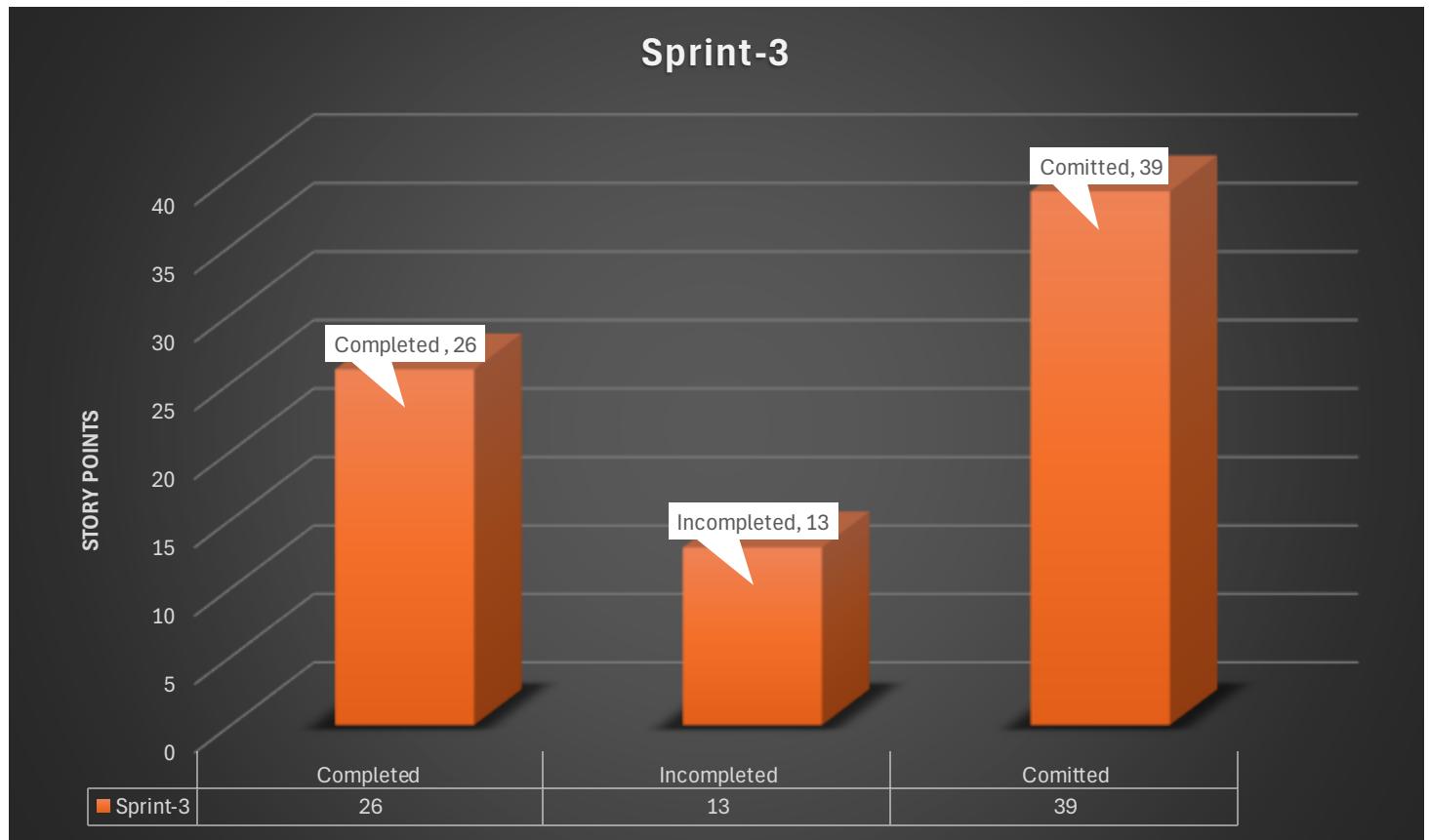
## Average Burndown Chart



# SPRINT -3 COMPLETED/COMMITTED RATIO



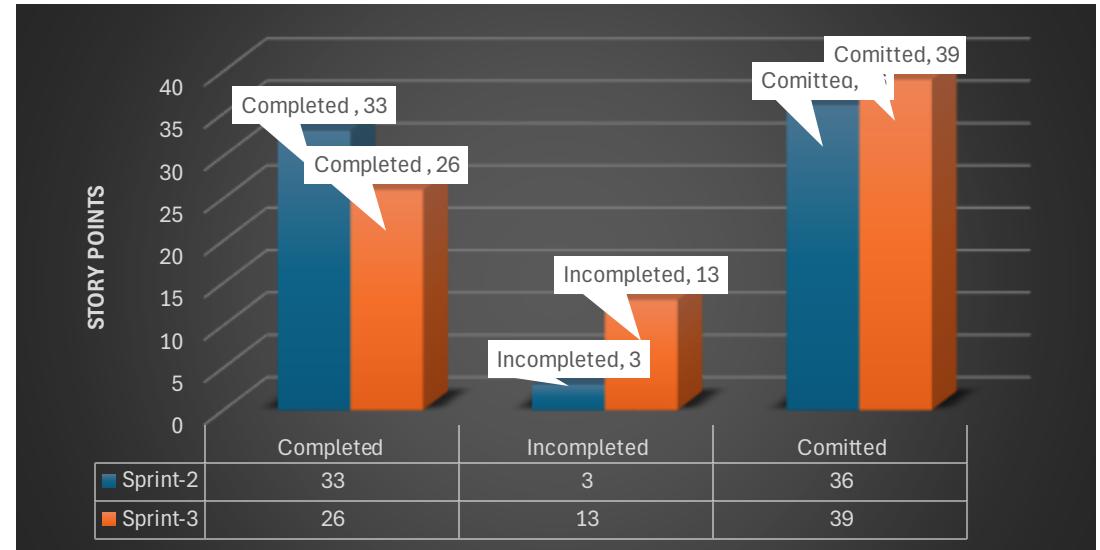
COMPLETED/COMMITTED RATIO FOR SPRINT – 3 IS  
 $26/39 = 66.67\%$



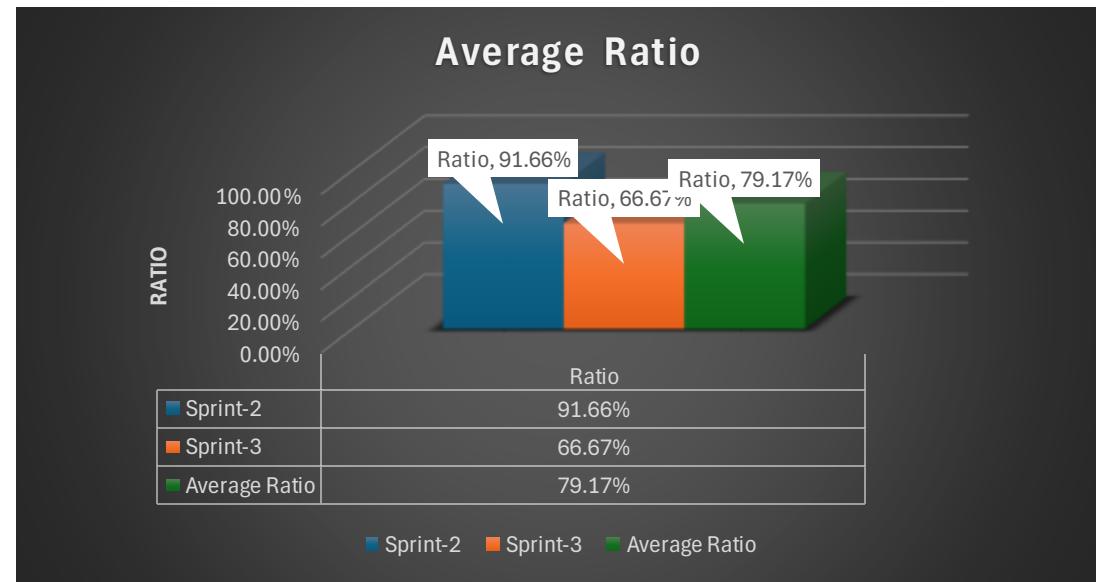
# AVERAGE COMPLETED/COMMITTED RATIO



AVERAGE RATIO =  
AVG(SPRINT-2 RATIO+SPRINT-3 RATIO)



AVERAGE RATIO IS - 79.17%

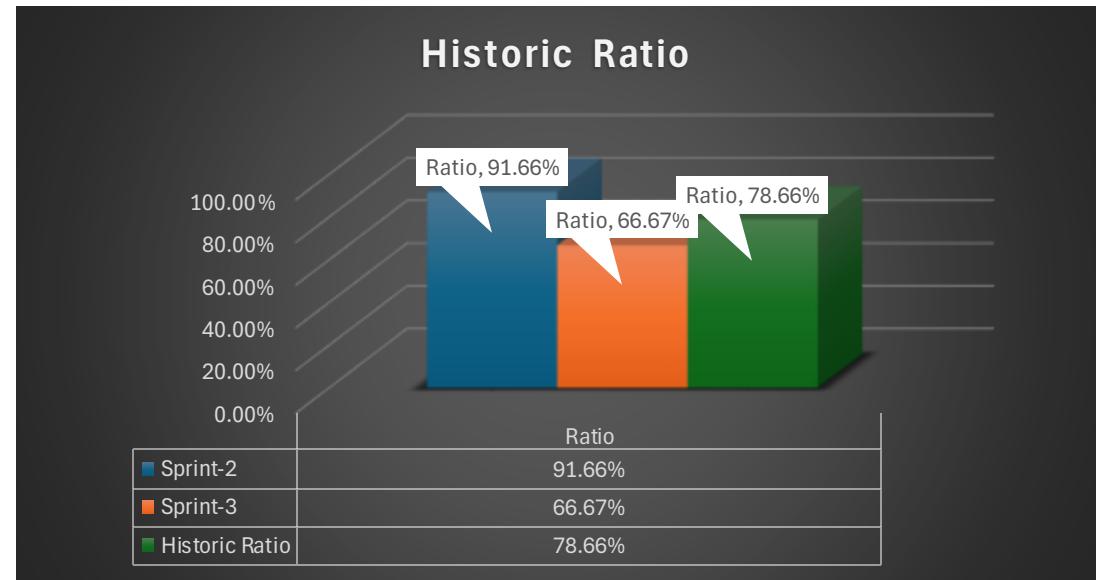
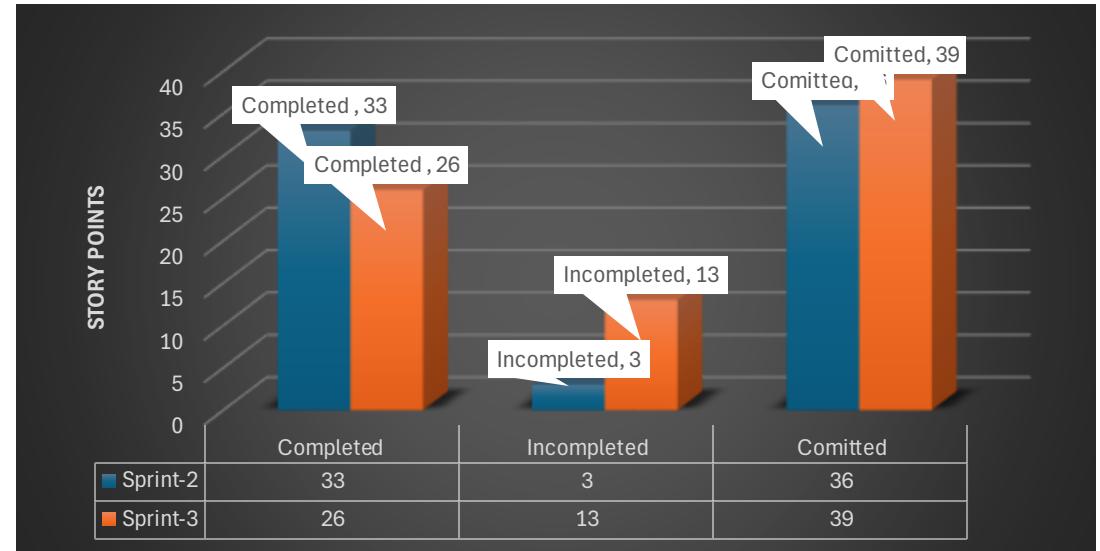


# HISTORIC COMPLETED/COMMITTED RATIO



**HISTORIC RATIO =**  
**TOTAL COMPLETED WORK**  
**TOTAL COMMITTED WORK**

**HISTORIC RATIO IS – 78.66%**



# RETROSPECTIVE



## Retrospective

### What went well

1. Team collaboration and completion of tasks on time  + 3	Improvement in sticking with work-agreement  + 3
Helpful Q&A  + 7	Feedback on each other's work was useful in completing the task more efficiently  + 5
Everyone was cooperative and time management was on point.  + 2	There was generous commitment from every teammate to assigned responsibilities which lead to successful task completion on time  + 3
Everybody learning and contributing on all the parts of the project.  + 1	Time was managed very well and everyone was cooperative  + 2
We conducted Daily standups to discuss the tasks assigned, to track progress and take updates on any blockers  + 7	

### What can be improved

Updating JIRA actively about task status  + 6	Taking Individual initiatives to start working on tasks and let the team members know if there are any dependencies in the tasks.  + 6
People that require assistance with blockers should reach out for help early, and not when the deadline is approaching.  + 6	task estimation  + 3
Constantly updating GIT (Version Control) as soon as you are finished working on specific part of the project.  + 2	fewer meetings and more work.  + 1
lesser meeting and focus on work  + 5	

### Action Items

Commit your work as soon as you are finished working into the version control (GIT).  + 4	Constantly communicating with other team members for the dependencies without delaying till the end of the estimated time for that task.  + 4
Each team member identifies tasks they can take initiative on, based on the project goals, their skills, and current priorities.  + 5	Story points should not exceed capacity of the group when estimating while planning  + 5
Question on Blockers should be asked in daily standup rather than to wait for weekly review meetings  + 7	Updating work and seeing others willingness to complete the work  + 7

# SPRINT-4 BACKLOG



Task ID	Story Points	User Story	Status	Sprint
MS_1	3	As a user, I want to be able to upload a picture representing my current mood From the Already Saved Images in the gallery	In Progress	Sprint - 3
		Criteria, The user can upload an image file. Supported image formats include JPG, PNG, and GIF.		
MS_8	5	As a user, I want to be able to search for specific songs or artists.	In Progress	Sprint - 3
		Criteria, Users can search for songs or artists using a search bar. Search results should be relevant and displayed in a clear manner.		
MS_23	5	As a user, I want the application to provide recommendations based on my current mood, even if I'm not logged in or don't have an account.	In Progress	Sprint - 3
		Criteria, Users should be able to access mood-based recommendations without requiring authentication or account creation.		
MS_24	3	As a user, I want the application to store uploaded pictures for future reference.	To Do	sprint -4
		Criteria, Uploaded pictures should be saved securely in the user's account. Users should be able to access their stored pictures from their profile or settings.		
MS_25	3	As a user, I want the ability to delete uploaded pictures if needed.	To Do	sprint -4
		Criteria, Users can delete uploaded pictures from their account. Deletion should be confirmed with a confirmation prompt to prevent accidental deletion.		
MS_21	3	As a user, I want the application to display loading indicators or progress bars to indicate when mood analysis and song retrieval are in progress.	To Do	sprint -4
		Criteria, Users should see visual cues indicating that the application is processing their request. Loading indicators should be displayed prominently and disappear once processing is complete.		

Task ID	Story Points	User Story	Status	Sprint
MS_5	5	<p>As a user, I want to be able to save songs I like to a playlist for future listening.</p> <p>Criteria, Users should have the option to add songs to a personal playlist. The playlist should persist across sessions for registered users.</p>	To Do	sprint -4
MS_10	3	<p>As a user, I want to be able to create and customize multiple playlists for different moods or occasions.</p> <p>Criteria, Users can create new playlists and give them custom names. Users can add or remove songs from playlists.</p>	To Do	sprint -4
MS_11	5	<p>As a user, I want the application to provide additional information about recommended songs</p> <p>Criteria, Users can access additional information about recommended songs by clicking on them. Information should be presented in a clear and concise manner.</p>	To Do	sprint -4
MS_15	5	<p>As a user, I want to be able to provide feedback on recommended songs or report any issues encountered while using the application.</p> <p>Criteria, Users should have a feedback mechanism to submit comments or report problems.</p>	To Do	sprint -4
MS_17	3	<p>As a user, I want to be able to clear my mood selection and start over if I'm not satisfied with the recommended songs.</p> <p>Criteria, Users should have the option to clear their mood selection and return to the initial state. Clearing the mood selection should reset the recommended songs.</p>	To Do	sprint -4
MS_20	3	<p>As a user, I want the application to provide clear instructions and guidance on how to use its features effectively.</p> <p>Criteria, The application should include help documentation or tooltips to explain its functionality.</p>	To Do	sprint -4

# PROJECT DEMO



# APP SCREENSHOTS



# Landing Page

Mood Sphere

GET STARTED

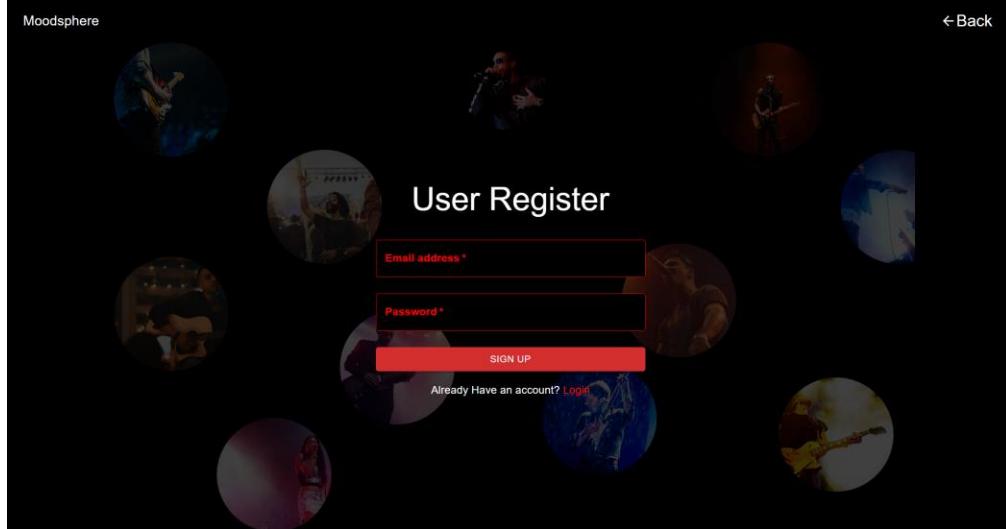
MoodSphere

Discover Your Music

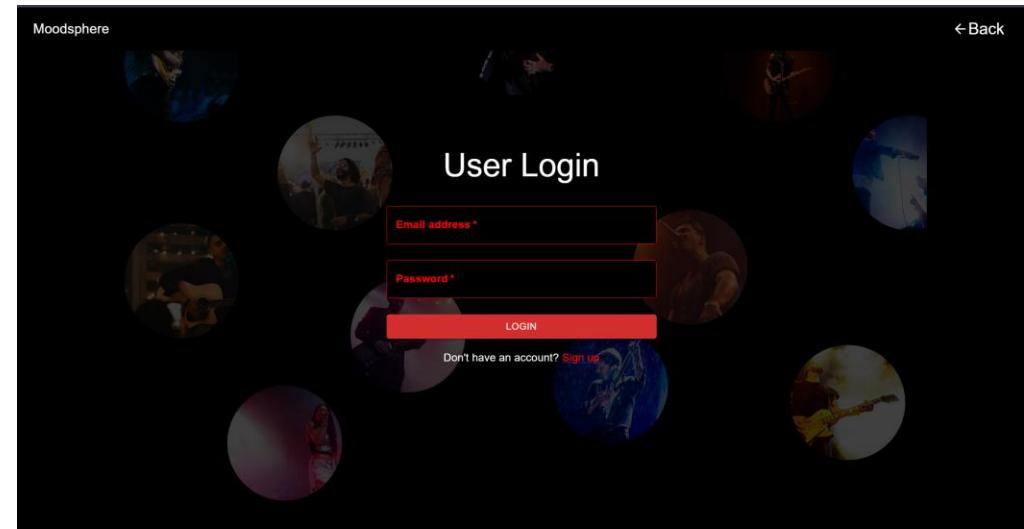
Get Songs based on your emotions in Realtime.

► GET STARTED    G LOGIN/SIGN UP WITH GOOGLE    E LOGIN WITH EMAIL

# Signup



# Login



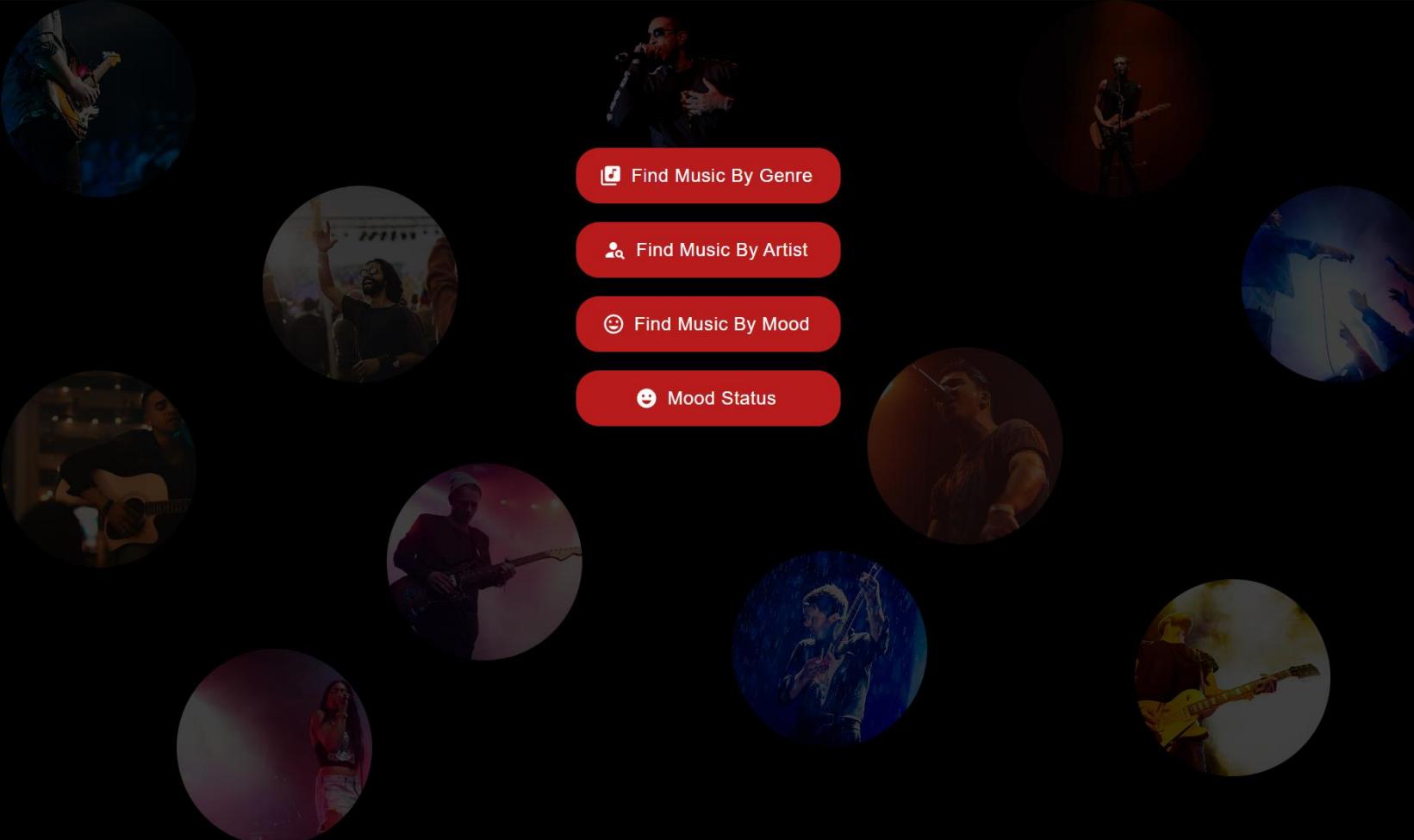
# Home Page

MoodSphere

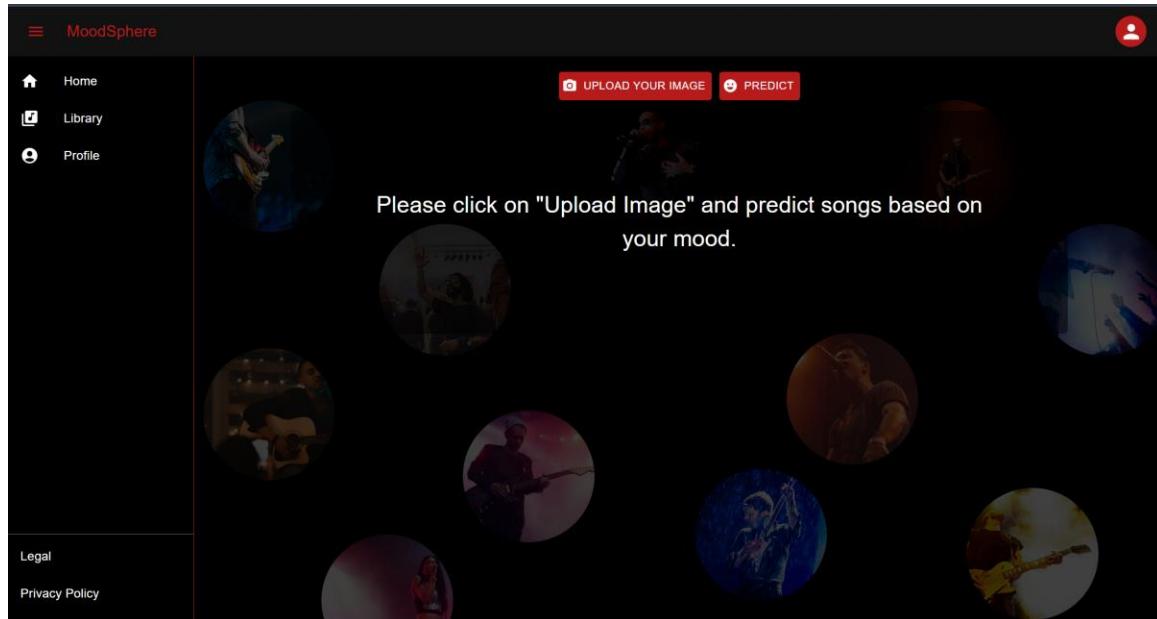
Home

Library

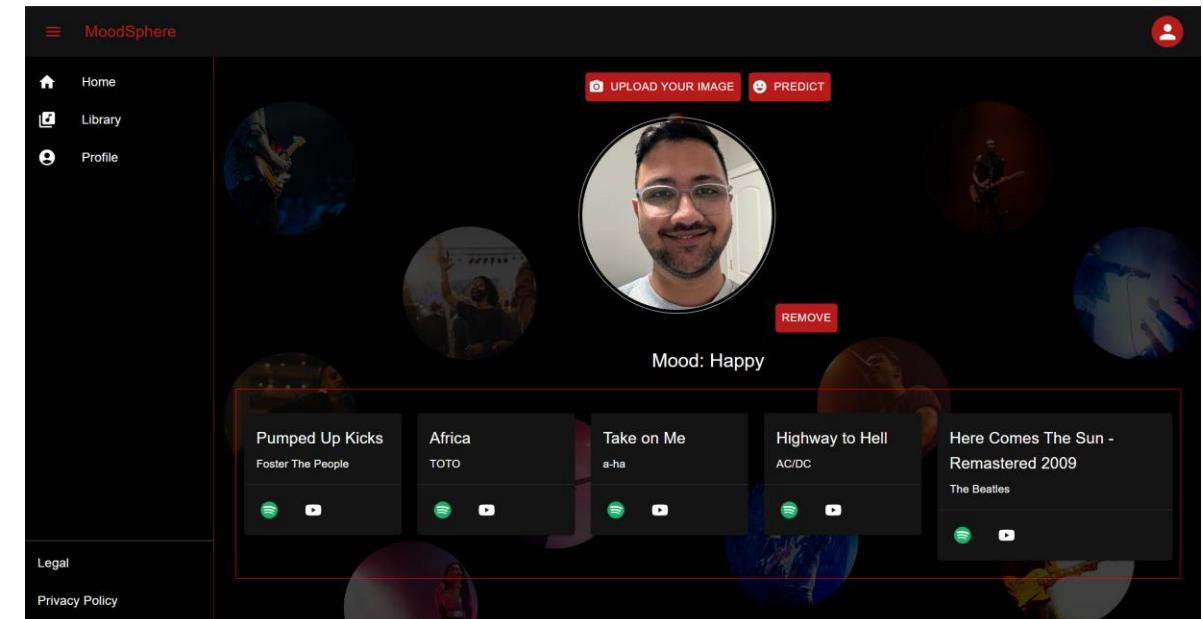
Profile



# Mood Prediction and Song Recommendation



The screenshot shows the home page of the MoodSphere application. At the top right is a user profile icon. Below it are three buttons: "UPLOAD YOUR IMAGE" with a camera icon, "PREDICT" with a smiley face icon, and a red "LOGOUT" button. On the left is a vertical navigation bar with "MoodSphere" at the top, followed by "Home", "Library", and "Profile". At the bottom are links for "Legal" and "Privacy Policy". The main area features a dark background with several circular thumbnails of people in various musical settings (guitarists, singers). A central text box reads: "Please click on \"Upload Image\" and predict songs based on your mood.".



The screenshot shows the predictions page after a user has uploaded an image and predicted their mood. At the top right is a user profile icon. Below it are three buttons: "UPLOAD YOUR IMAGE" with a camera icon, "PREDICT" with a smiley face icon, and a red "LOGOUT" button. On the left is a vertical navigation bar with "MoodSphere" at the top, followed by "Home", "Library", and "Profile". At the bottom are links for "Legal" and "Privacy Policy". The main area shows a circular profile picture of a man with glasses and a beard. Below it, the text "Mood: Happy" is displayed. A red box highlights a row of four song cards: "Pumped Up Kicks" by Foster The People, "Africa" by TOTO, "Take on Me" by a-ha, and "Highway to Hell" by AC/DC. To the right, another row of song cards is partially visible: "Here Comes The Sun - Remastered 2009" by The Beatles. Each card includes the song title, artist, a Spotify icon, and a YouTube icon.

# API SLIDES



## API Endpoint: /songs-by-artist

- This API endpoint “/songs-by-artist” is designed to recommend songs based on a provided artist name.
- The API endpoint accepts POST requests and expects to receive a JSON payload containing an artist name. It processes this input to find and return a list of songs associated with the specified artist.
- Upon receiving the request, it attempts to parse the JSON to extract the artist's name.
- If no artist name is provided, it responds with an error and a 400-status code. If an artist name is given, it then uses the get\_songs\_by\_artist function to retrieve the corresponding songs

```
def get_songs_by_artist(artist_name):
    songs_by_artist = Music_Player[Music_Player['artist'] == artist_name]
    if songs_by_artist.empty:
        return []
    return songs_by_artist[['name', 'artist']].values.tolist()

def get_songs_by_genre(artist_name):
    songs_by_artist = Music_Player[Music_Player['artist'] == artist_name]
    if songs_by_artist.empty:
        return []
    return songs_by_artist[['name', 'artist']].values.tolist()

#API for recommending songs based on artists
@app.route('/songs-by-artist', methods=['POST'])
def artist():
    try:
        data = request.get_json()
        artist_name = data.get('artist_name')

        if not artist_name:
            return jsonify({'error': 'Artist name not provided'}), 400

        songs = get_songs_by_artist(artist_name)
        if not songs:
            return jsonify({'message': 'No songs found for the provided artist'})

        return jsonify({'songs': songs}), 200

    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

## API Endpoint: /predict

- The function `recommend_songs` takes a predicted class of mood from an image and maps it to a corresponding mood category. For example, 'Disgust' is mapped to 'Sad', and combinations like ['Happy', 'Sad'] or ['Fear', 'Angry'] are mapped to 'Happy' or 'Calm' respectively. Any other mood prediction results in the 'Energetic' category.
- Once the mood is determined, the system filters the songs from the `Music_Player` dataset that match the mood category. Then it sorts these songs by their popularity in descending order and returns the top 5 as a dictionary in the 'records' format.

```
# Function to recommend songs based on predicted class
def recommend_songs(pred_class):
    if pred_class == 'Disgust':
        mood = 'Sad'
    elif pred_class in ['Happy', 'Sad']:
        mood = 'Happy'
    elif pred_class in ['Fear', 'Angry']:
        mood = 'Calm'
    else:
        mood = 'Energetic'

    # Filter music based on predicted mood and sort by popularity
    recommended_songs = Music_Player[Music_Player['mood'] == mood]
    recommended_songs = recommended_songs.sort_values(by="popularity", ascending=False).head(5).to_dict('records')

    return recommended_songs

# Define route for prediction
@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get the uploaded image file from the request
        file = request.files.get('image')
        if not file:
            return jsonify({'error': 'No file uploaded'}), 400
        print(request.headers)
        # Read the image file as an array
        img = Image.open(io.BytesIO(file.read()))
        img = img.resize((150, 150)) # Resize image to match model's expected sizing
        img_array = np.asarray(img)
        img_array = np.expand_dims(img_array, axis=0)
        img_array = img_array / 255.0 # Normalize pixel values
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

## API Endpoint: /predict

- Before making a prediction, the code asserts that the shape of the pre-processed image array matches the expected input shape for the model, which is (1, 150, 150, 3). If the shape does not match, an assertion error with the message "Unexpected image shape" will be raised.
- The response is constructed as a JSON object with two fields: 'prediction', containing the name of the predicted mood class, and 'recommended\_songs', containing the list of songs recommended by the system. This response is then returned to the client. If an exception occurs during the process, an error response with a 400-status code and the exception message is returned.

```
# Make sure image shape matches the expected input shape
assert img_array.shape == (1, 150, 150, 3), "Unexpected image shape"

# Perform prediction using your model
prediction = model.predict(img_array)
predicted_class_index = np.argmax(prediction)
predicted_class = class_names[predicted_class_index]

prediction = model.predict(img_array)
predicted_class_index = np.argmax(prediction)
predicted_class = class_names[predicted_class_index]

recommended_songs = recommend_songs(predicted_class)

response = {
    'prediction': predicted_class,
    'recommended_songs': recommended_songs
}

return jsonify(response)

except Exception as e:
    return jsonify({'error': str(e)}), 400
```

# GITHUB LINK

[HTTPS://GITHUB.COM/  
HTMW/2024S-DEV-  
DYNASTY/WIKI](https://github.com/HTMW/2024S-DEV-DYNASTY/WIKI)



# LIVE APPLICATION DEMO LINK

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=00UBHMGBNQG&FEATURE=YOUTU.BE](https://www.youtube.com/watch?v=00UBHMGBNQG&feature=youtu.be)





# THANK YOU

DevDynasty