Final Year Project 1

Coordinator: Dr.Aarij Mahmood Hussaan

Project Name: Universal Recommendation System

Supervisor: Rukhsana Majeed

Fyp Group 1:

Group Members:

Muhammad Ali Ammar Naseer (54353) (Leader)
Hassaan Ahmed (60211)
Hafsa Amin (60209)
Abdul Moiz (54357)

Project Description:

Our group is developing a universal recommendation system to provide personalized recommendations to a large user base. This system is designed to help users find relevant items based on their preferences, behavior, and other factors. With the overwhelming amount of data and information available, it can be challenging for users to find what they need quickly and efficiently. Our recommendation system addresses this problem by offering personalized suggestions that save users time and effort while improving their overall experience.

Our system consists of four unique features: a content-based movie recommendation system, a book recommendation system using collaborative filtering, a music recommendation system using collaborative filtering and cosine similarity, and a fashion recommendation system based on deep learning CNN. Each feature serves a specific purpose in providing personalized recommendations to users.

We are using Python, Django, and Flask for the backend and HTML and CSS for the front end of our recommendation system. By implementing these four features, we aim to provide a comprehensive and personalized recommendation system that will improve user engagement, satisfaction, and conversion rates for businesses.

Project's Main Features:

This project aims to build a multiple-category recommendation system that consists of four features. These features include a content-based movie recommendation system, a book recommendation system, a music recommendation system, and a fashion recommendation system. Each feature will serve a unique purpose in providing personalized recommendations to the users.

1. Movie Recommendation System

The movie recommendation system will use content-based filtering to recommend movies similar to a given movie. The system will represent each movie as a vector and use cosine similarity to find movies that are similar to the input movie. The system will also display the top 250 movies based on their rating, which will be updated using an API. The system will collect data on the movies, represent each movie as a vector, and use the updated data to retrain the model and predict the new movie ratings. Based on these predicted ratings, the system will sort and display the top 250 movies to the user.

2. Book Recommendation System

The book recommendation system using collaborative filtering will provide users with recommendations based on their input book name. To ensure the quality of recommendations, the system will only consider books that have received at least 250 votes from users in the dataset. By filtering out unpopular books, the system aims to recommend only the most popular and highly rated books to the user.

3. Music Recommendation System

The music recommendation system employs a collaborative filtering technique that analyzes users' past behavior and preferences to recommend new songs. The system computes the similarity between a user's musical taste and other users' preferences using cosine similarity as the similarity metric.

Cosine similarity measures the cosine of the angle between two vectors, where each vector represents a user's musical preference. The closer the angle between the two vectors, the higher the cosine similarity score between them. The system then scores each candidate song based on how similar it is to the user's preference vector, and the system recommends songs with the highest scores.

In other words, if a song has a high cosine similarity score, it means that the song is highly correlated with the user's musical taste, and the user is more likely to find it interesting. The system recommends the songs with the highest scores to the user, as they are more likely to match the user's interests.

4. Fashion Recommendation System

The fashion recommendation system based on deep learning CNN utilizes a reverse image search approach to recommend visually similar products to the user. The user can upload an image of a fashion-related item, such as suits, watches, shoes, and so on. The system processes the image using a convolutional neural network (CNN) to identify the features and patterns in thepicture.

Then, the system searches its database for products that have similar features and patterns to the user's uploaded image. It recommends a maximum of 5 products that are visually similar to the

user's image. This approach of the fashion recommendation system is called reverse image search as it uses the image as a search query to find similar products.

This fashion recommendation system is particularly useful for users who want to find similar designs or styles to the one in their image. By using deep learning CNN, the system can accurately identify the features and patterns in the image and find visually similar products, making it easier for users to discover fashion items that match their preferences.

RESEARCH APPROACHES:

How will this project be made?

This project is planned to be developed with a combination of Python, Django, and Flask for the backend and HTML and CSS for the frontend. It comprises four main features, each with its own approach to providing personalized recommendations to users. The movie recommendation system will use content-based filtering, the book recommendation system will use collaborative filtering, the music recommendation system will use collaborative filtering and cosine similarity, and the fashion recommendation system will use deep learning CNN for reverse image search.

Why will this project be made?

This project is being made to help users find relevant items based on their preferences, behavior, and other factors, which can save them time and effort while improving their overall experience. With the vast amount of data and information available, it can be challenging for users to find what they need quickly and efficiently.

The main purpose of this project is to provide a personalized recommendation system to a large user base across different categories such as movies, books, music, and fashion. The overwhelming amount of data available can make it challenging for users to find what they need quickly and efficiently. The project also aims to improve user engagement and satisfaction for businesses by providing them with accurate recommendations for their products.

What are the approaches to construct this project?

The construction of this project involves the implementation of various approaches and techniques in order to create an effective and efficient recommendation system. Here are some of the key approaches:

- **Content-based filtering:** This approach involves analyzing the features and characteristics of items, such as movies, books, or music, and making recommendations based on their similarities. For instance, the movie recommendation system in this project uses content-based filtering to recommend movies that are similar to a given movie.

- **Collaborative filtering:** This approach involves analyzing the behavior and preferences of users and making recommendations based on their similarities with other users. The book and music recommendation systems in this project utilize collaborative filtering techniques to make personalized recommendations based on the user's past behavior and preferences.

- **Deep learning:** The fashion recommendation system in this project employs deep learning techniques, specifically convolutional neural networks (CNNs), to identify the features and patterns in fashion-related images uploaded by users. The system uses reverse image search to recommend visually similar products to the user.

- **API integration:** The movie recommendation system in this project integrates an API to retrieve and display the top 250 movies based on their rating. The system regularly updates its data to retrain the model and predict new movie ratings.

- **Front-end development:** The recommendation system's front-end development uses HTML and CSS to provide an intuitive and user-friendly interface that enhances the user's overall experience.

Overall, the approaches used in this project focus on providing personalized recommendations to users based on their behavior, preferences, and other factors. The combination of content-based filtering, collaborative filtering, deep learning, API integration, and front-end development creates a comprehensive and effective recommendation system that improves user engagement and satisfaction.

Functional Requirements:

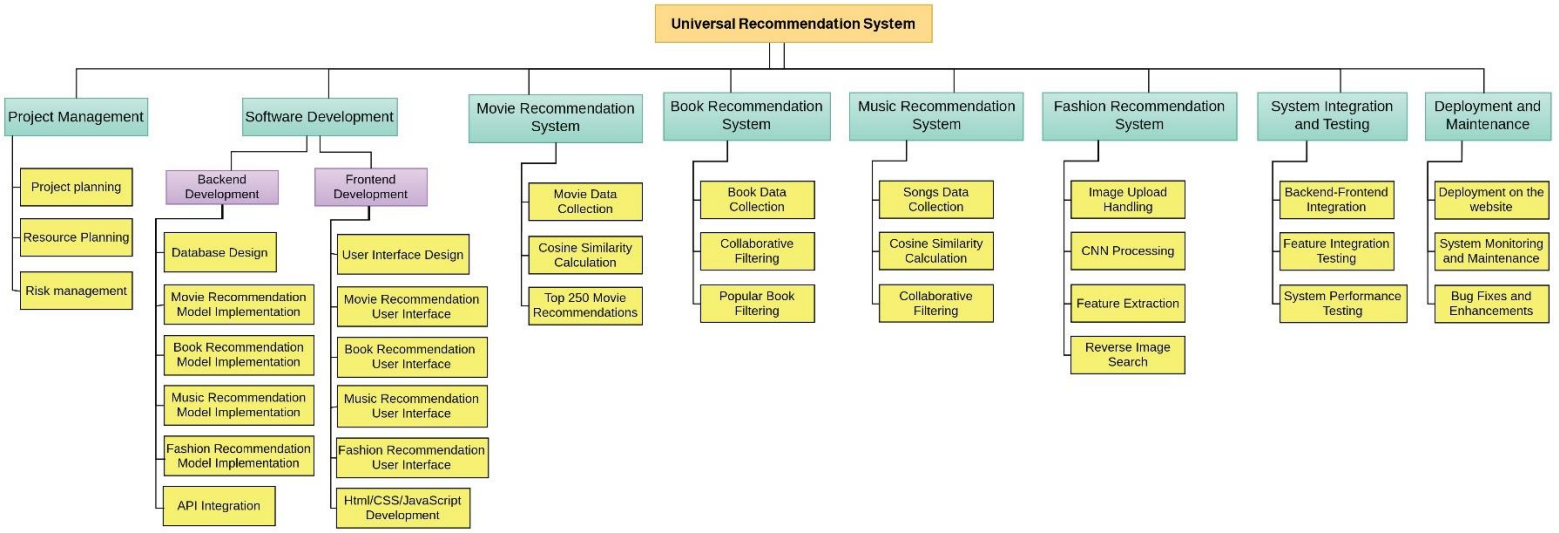
- The system recommends movies similar to the input movie using content-based filtering.
- An updated list of the top 250 movies, ranked according to user ratings and obtained via an API, is displayed by the system.
- The book recommendation system should be able to provide users with book recommendations based on their input book name.
- Only books that have received at least 250 votes from users in the dataset are considered by the book recommendation system.
- The music recommendation system should compute the similarity between a user's musical taste and other users' preferences using cosine similarity as the similarity metric.
- To suggest new songs to users based on their past behavior and preferences, the music recommendation system uses collaborative filtering.
- The fashion recommendation system should be able to recommend visually similar products to the user based on their uploaded image.
- By analyzing uploaded images, the fashion recommendation system suggests visually similar products to users, utilizing deep learning CNN.

- A maximum of five visually similar products to the user's uploaded image is recommended by the fashion recommendation system.

Non-Functional Requirements:

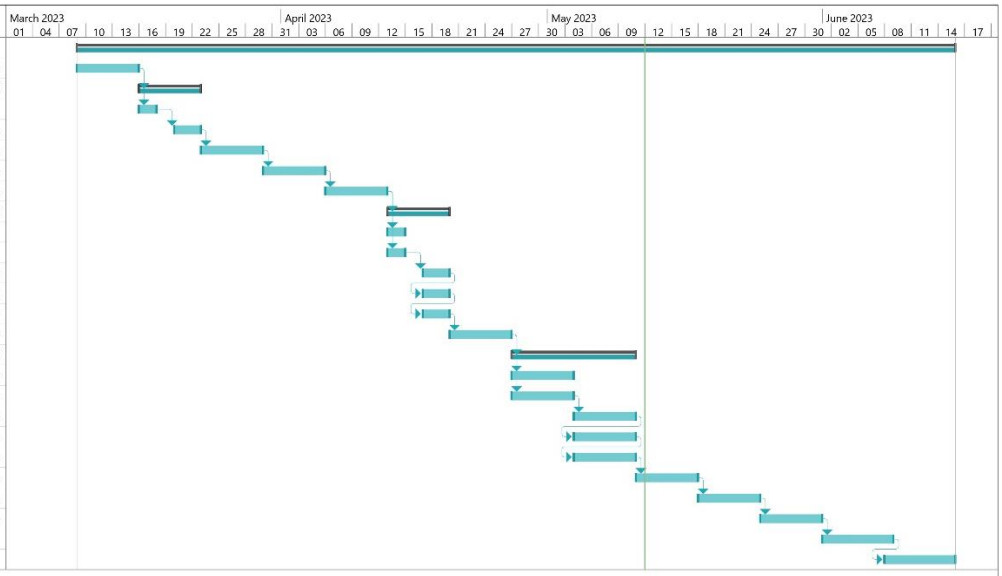
- The system should be highly personalized and provide recommendations based on individual preferences and behavior.
- The system has the ability to handle a large user base and is designed to be scalable.
- The system should establish a direct connection with the dataset through API and synchronize the data in real-time. This ensures that the system always has the most up-to-date information, allowing for accurate and personalized recommendations for users.
- A user-friendly interface is provided by the system to ensure ease of use and a superior user experience.
- The system should provide recommendations quickly and efficiently.

Work Breakdown Structure (WBS):



Gantt Chart FYP 1:

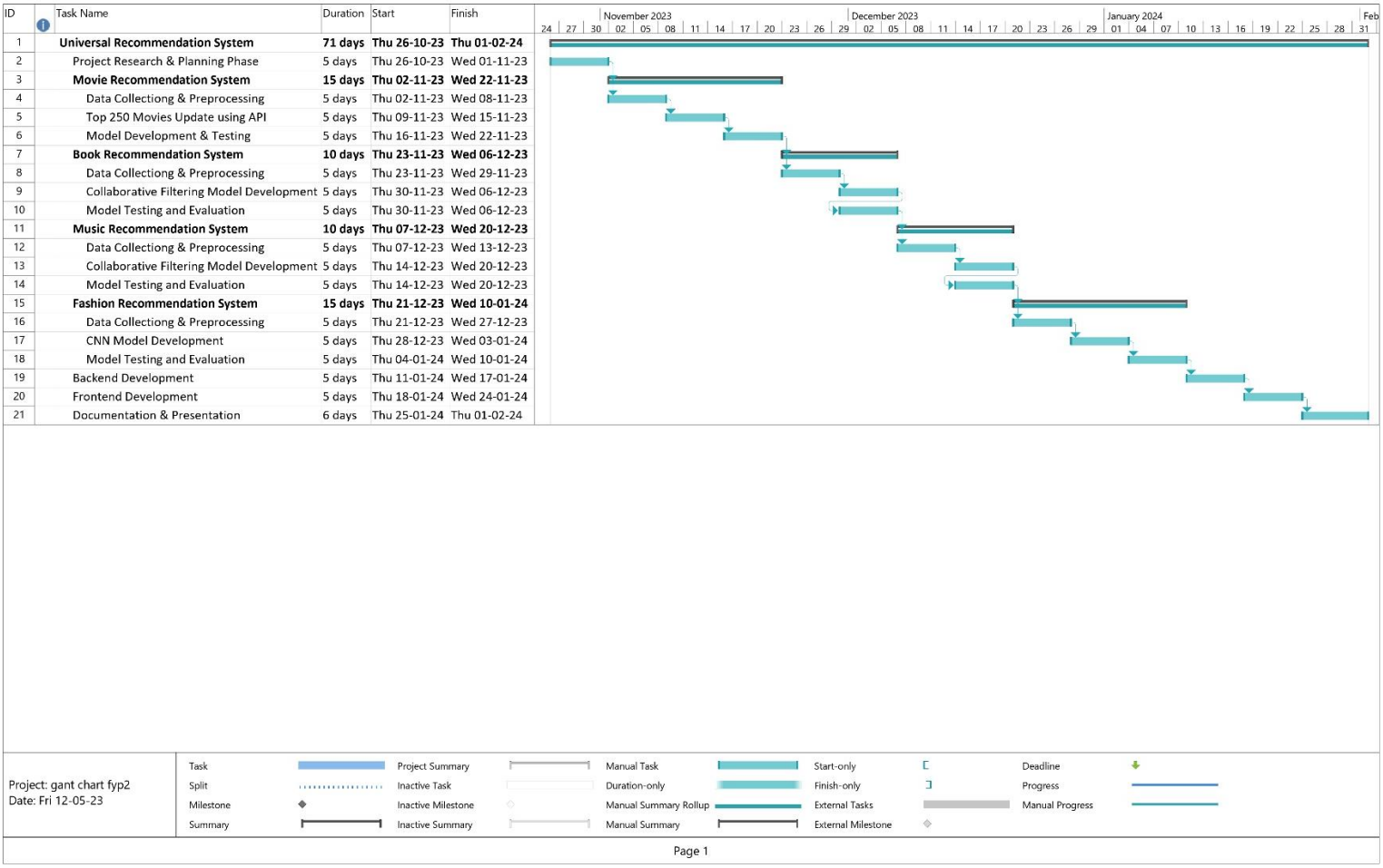
ID	Task Name	Duration	Start	Finish
1	**Universal Recommendation System**	**71 days**	**Thu 09-03-23**	**Thu 15-06-23**
2	Project Research	5 days	Thu 09-03-23	Wed 15-03-23
3	**Planning**	**5 days**	**Thu 16-03-23**	**Wed 22-03-23**
4	Background Research	2 days	Thu 16-03-23	Fri 17-03-23
5	Brainstorming Idea	3 days	Mon 20-03-23	Wed 22-03-23
6	Idea Approval	5 days	Thu 23-03-23	Wed 29-03-23
7	Supervisor Discussion	5 days	Thu 30-03-23	Wed 05-04-23
8	Project Proposal Submission	5 days	Thu 06-04-23	Wed 12-04-23
9	**Project Requirements**	**5 days**	**Thu 13-04-23**	**Wed 19-04-23**
10	Title	2 days	Thu 13-04-23	Fri 14-04-23
11	Description	2 days	Thu 13-04-23	Fri 14-04-23
12	Supervisor	3 days	Mon 17-04-23	Wed 19-04-23
13	Group Members	3 days	Mon 17-04-23	Wed 19-04-23
14	Features	3 days	Mon 17-04-23	Wed 19-04-23
15	Research Approach	5 days	Thu 20-04-23	Wed 26-04-23
16	**Detailed Plan**	**10 days**	**Thu 27-04-23**	**Wed 10-05-23**
17	Functional	5 days	Thu 27-04-23	Wed 03-05-23
18	Non-Functional	5 days	Thu 27-04-23	Wed 03-05-23
19	Work Breakdown Structure	5 days	Thu 04-05-23	Wed 10-05-23
20	Gantt Chart	5 days	Thu 04-05-23	Wed 10-05-23
21	Division of Tasks	5 days	Thu 04-05-23	Wed 10-05-23
22	1st Iteration of Design	5 days	Thu 11-05-23	Wed 17-05-23
23	2nd Iteration of Design	5 days	Thu 18-05-23	Wed 24-05-23
24	3rd Iteration of Design	5 days	Thu 25-05-23	Wed 31-05-23
25	Poster Design	6 days	Thu 01-06-23	Thu 08-06-23
26	Final Report	6 days	Thu 08-06-23	Thu 15-06-23



Project: Gantt chart
Date: Fri 12-05-23

Task	Inactive Task	Manual Summary Rollup	External Milestone	Baseline
Split	Inactive Milestone	Manual Summary	Deadline	Baseline Milestone
Milestone	Inactive Summary	Start-only	Path Successor Milestone Task	Baseline Summary
Summary	Manual Task	Finish-only	Path Successor Summary Task	Progress
Project Summary	Duration-only	External Tasks	Path Successor Normal Task	Manual Progress

Gantt Chart FYP 2:



ID	Task Name	Duration	Start	Finish
1	**Universal Recommendation System**	**71 days**	**Thu 26-10-23**	**Thu 01-02-24**
2	Project Research & Planning Phase	5 days	Thu 26-10-23	Wed 01-11-23
3	**Movie Recommendation System**	**15 days**	**Thu 02-11-23**	**Wed 22-11-23**
4	Data Collectiong & Preprocessing	5 days	Thu 02-11-23	Wed 08-11-23
5	Top 250 Movies Update using API	5 days	Thu 09-11-23	Wed 15-11-23
6	Model Development & Testing	5 days	Thu 16-11-23	Wed 22-11-23
7	**Book Recommendation System**	**10 days**	**Thu 23-11-23**	**Wed 06-12-23**
8	Data Collectiong & Preprocessing	5 days	Thu 23-11-23	Wed 29-11-23
9	Collaborative Filtering Model Development	5 days	Thu 30-11-23	Wed 06-12-23
10	Model Testing and Evaluation	5 days	Thu 30-11-23	Wed 06-12-23
11	**Music Recommendation System**	**10 days**	**Thu 07-12-23**	**Wed 20-12-23**
12	Data Collectiong & Preprocessing	5 days	Thu 07-12-23	Wed 13-12-23
13	Collaborative Filtering Model Development	5 days	Thu 14-12-23	Wed 20-12-23
14	Model Testing and Evaluation	5 days	Thu 14-12-23	Wed 20-12-23
15	**Fashion Recommendation System**	**15 days**	**Thu 21-12-23**	**Wed 10-01-24**
16	Data Collectiong & Preprocessing	5 days	Thu 21-12-23	Wed 27-12-23
17	CNN Model Development	5 days	Thu 28-12-23	Wed 03-01-24
18	Model Testing and Evaluation	5 days	Thu 04-01-24	Wed 10-01-24
19	Backend Development	5 days	Thu 11-01-24	Wed 17-01-24
20	Frontend Development	5 days	Thu 18-01-24	Wed 24-01-24
21	Documentation & Presentation	6 days	Thu 25-01-24	Thu 01-02-24

Division of Tasks:

Muhammad Ali Ammar Naseer (54353) (Group Leader)

- Responsible for leading and managing the group.
- Coordinates with the supervisor and the coordinator.
- Work on the music recommendation system, including data collection and preprocessing.
- Develops the code for collaborative filtering using cosine similarity.
- Computes the similarity between a user's musical taste and other users' preferences.
- Scores each candidate song based on how similar it is to the user's preference vector.
- Testing and debugging the music recommendation system.

Hassaan Ahmed (60211)

- Work on the content-based movie recommendation system, including data collection, datapreprocessing, and vector representation of movies.
- Develops the code to represent each movie as a vector.
- Implement the cosine similarity algorithm for finding similar movies based on theirattributes.
- Work on updating the Top 250 movies rating using an API and retraining the model.
- Testing and debugging the book recommendation system.

Hafsa Amin (60209)

- Work on the fashion recommendation system based on deep learning CNN, including datacollection and preprocessing.
- Implement the code for the reverse image search approach using convolutional neuralnetworks.
- Ensure that the system accurately identifies the features and patterns in the user's uploadedimage.
- Recommend visually similar products to the user and work on testing and evaluating thefashion recommendation system.
- Testing and debugging the fashion recommendation system.

Abdul Moiz (54357)

- Work on the book recommendation system, including data collection and preprocessing.
- Develops the code for collaborative filtering to recommend books.
- Filters out unpopular books from the dataset to ensure quality recommendations.
- Works on retraining the model to predict new book ratings.

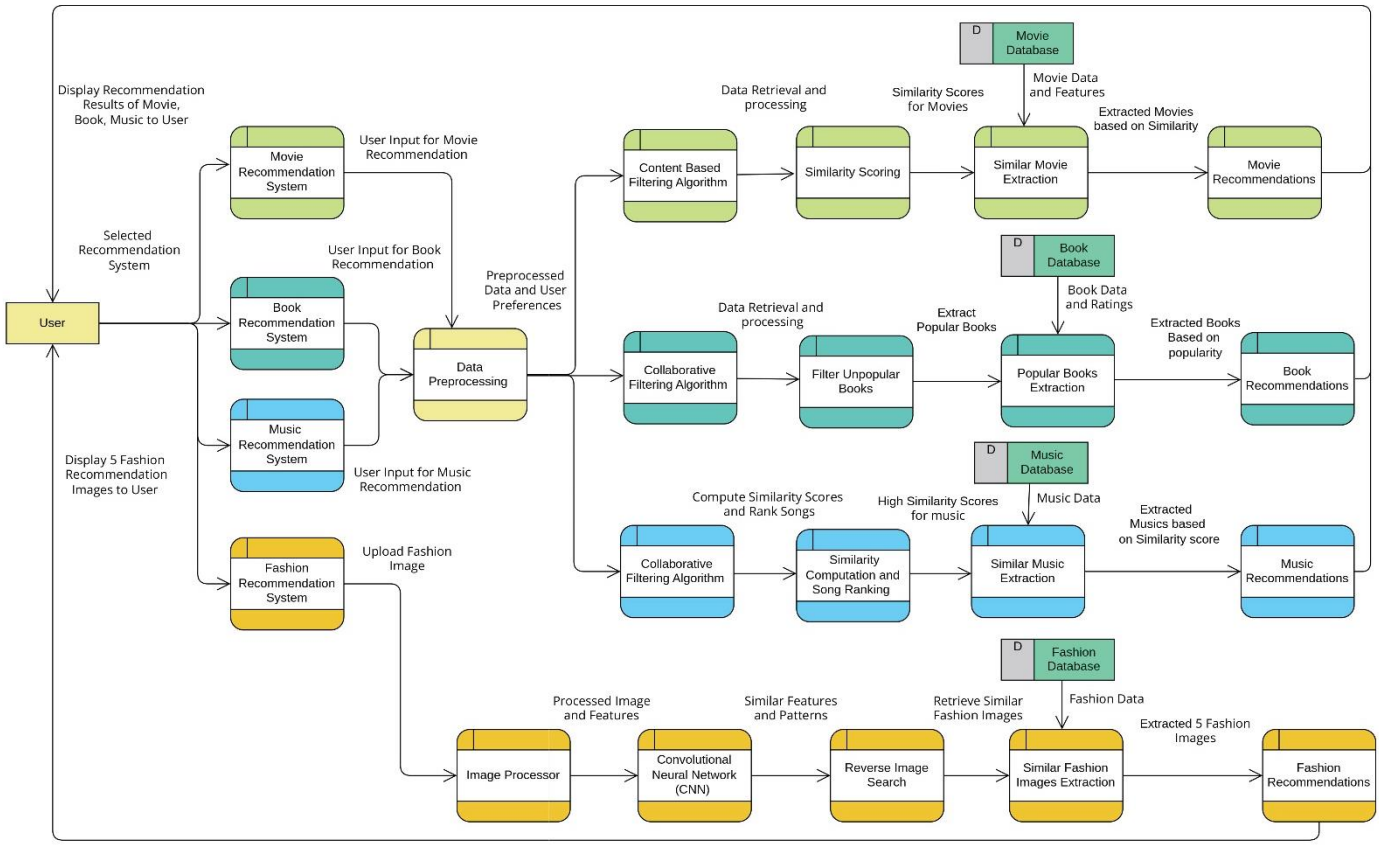
All group members will also contribute to the following:

- Work on the integration of the four recommendation systems into a single website.
- Writing the system's backend code using Python, Django, and Flask.
- Writing the system's frontend code using HTML and CSS.
- Testing and debugging the system.
- Prepare and deliver presentations on the progress of the project to the supervisor and coordinator.

SOFTWARE DESIGN

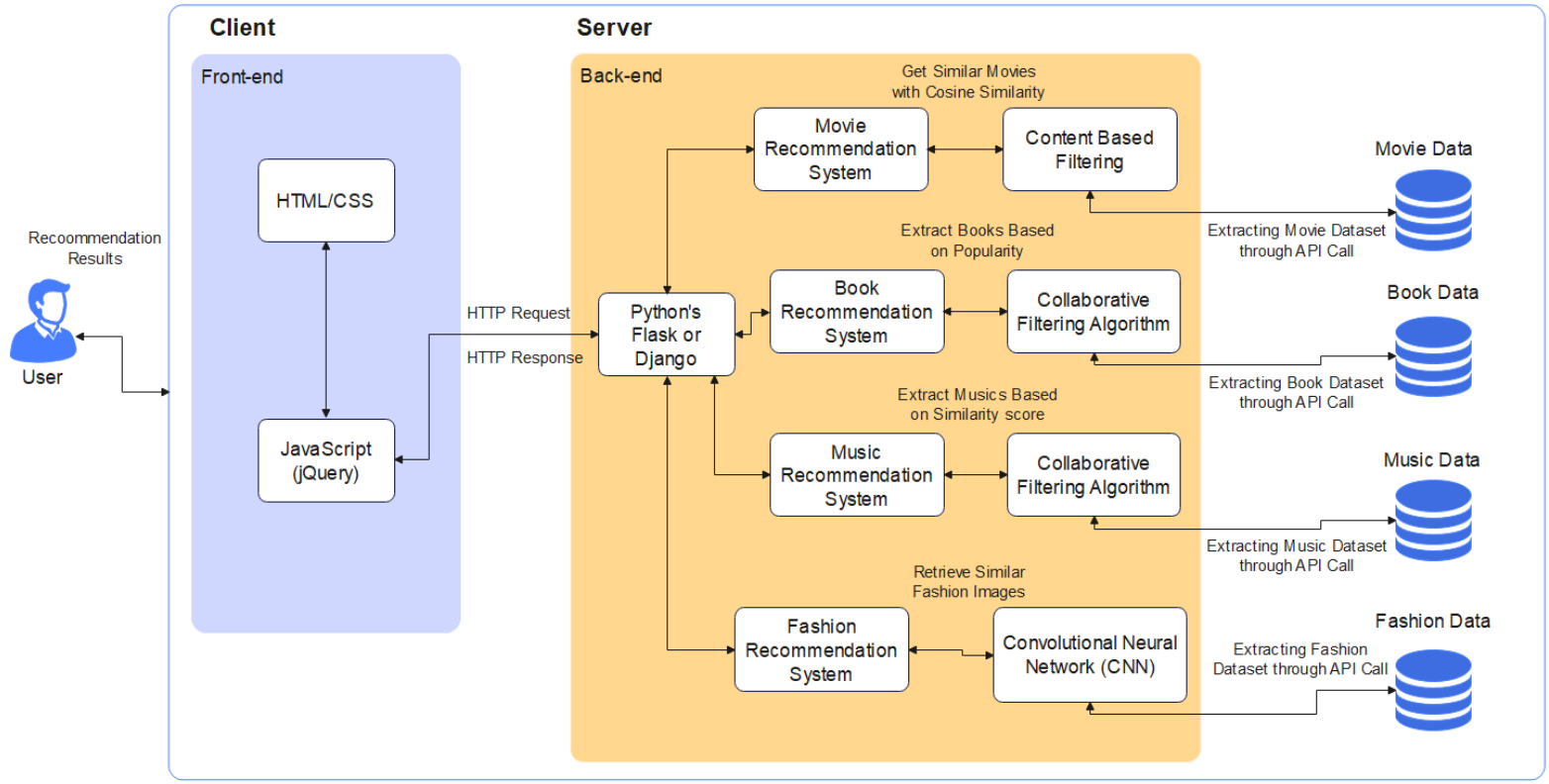
1st ITERATION OF DESIGN:

DATA FLOW DIAGRAM:



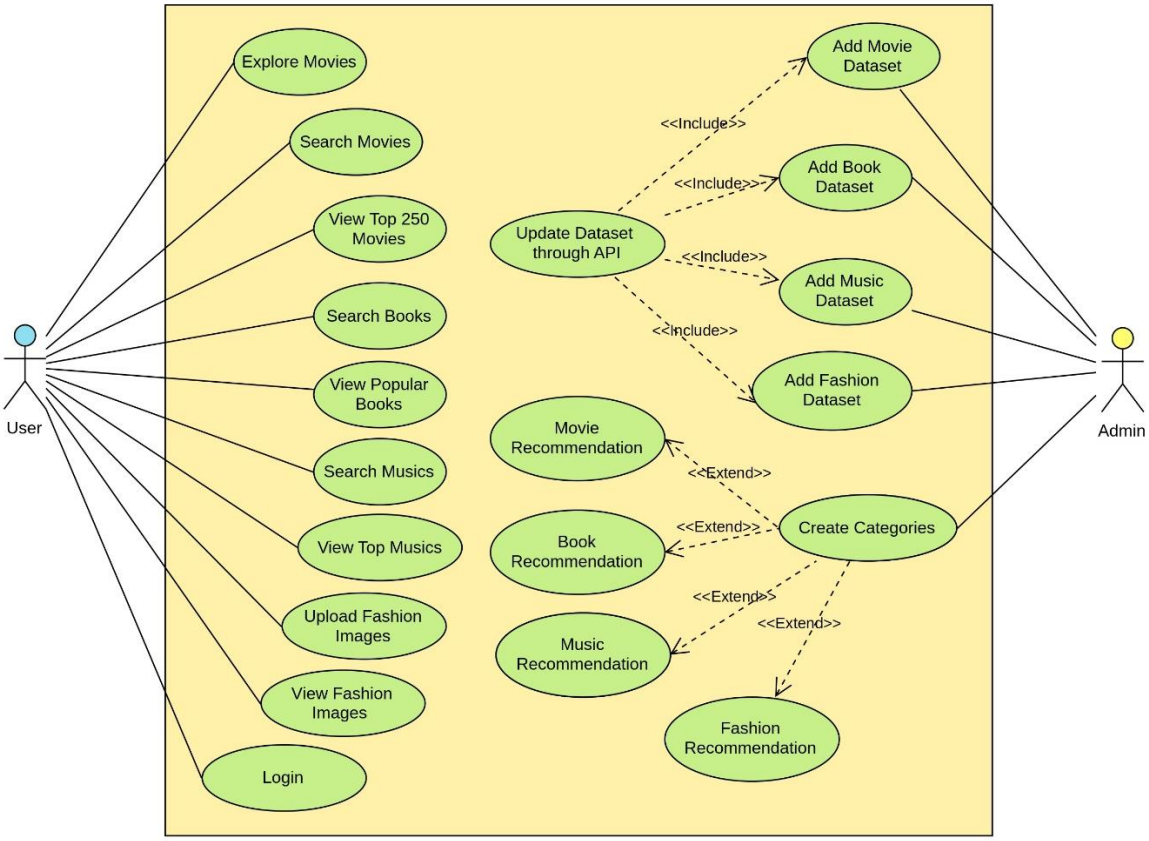
2nd ITERATION OF DESIGN:

ARCHITECTURE DIAGRAM:



3rd ITERATION OF DESIGN:

USE CASE DIAGRAM:



NARRATIVE:

Actors: User, Admin

Use Case Name:	Explore Movies	
Actor:	User	
Use Case Description:	The actor can explore a section that displays content based movie recommendation to actor.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor selects the "Movie Recommendation" option from the main menu. Step 4: The actor scrolls through the movie recommendations.	Step 2: The system presents a section with movie recommendations based on user preferences. Step 3: The system displays information about each movie, including title, genre, release year, and a brief summary.
Precondition:	The actor should be enter inside the movie recommendation system option.	
Postcondition:	The user has viewed the movie recommendations and can proceed to take further actions based on their preferences.	

Use Case Name:	Search Movies	
Actor:	User	
Use Case Description:	The actor can search for movies within the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor enters keywords, movie titles and genres into the search field. Step 3: The actor views the movie recommendation results.	Step 2: The system performs a search based on the actor's input and displays the recommendation results.
Precondition:	The actor should be present inside the movie recommendation system option.	
Postcondition:	The actor has obtained search results based on their query.	

Use Case Name:	View Top 250 Movies	
Actor:	User	
Use Case Description:	The actor can view top 250 movies within Movie Recommendation based on user's rating.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor view the "Top 250 Movies" within the movie recommendation section. Step 3: The actor scrolls through the list of top 250 movies.	Step 2: The system displays the movie titles and genres for each movie in the list.
Precondition:	The actor should be present inside the movie recommendation system option.	
Postcondition:	The actor has viewed the top 250 movies	

Use Case Name:	Search Books	
Actor:	User	
Use Case Description:	The actor can search for books based on their preferences.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor enters book title or author's name into the search field. Step 3: The actor views the book recommendation results.	Step 2: The system performs a search based on the actor's input and displays the recommendation results.
Precondition:	The actor should be present inside the book recommendation system option.	
Postcondition:	The actor has obtained search results based on their query.	

Use Case Name:	View Popular Books	
Actor:	User	
Use Case Description:	The actor can view a section that displays recommendation of popular books.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor selects the "Book Recommendation" option from the main menu. Step 3: The actor scrolls through the book recommendations.	Step 2: The system presents a section with book recommendations based on popularity.
Precondition:	The actor should be present inside the book recommendation system option.	
Postcondition:	The actor has viewed the popular books.	

Use Case Name:	Search Musics	
Actor:	User	
Use Case Description:	The actor can search for music based on their preferences and user's musical taste.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor enters keywords, song titles, artists and genres into the search field. Step 3: The actor views the music recommendation results.	Step 2: The system performs a search based on the actor's input and displays the recommendation results.
Precondition:	The actor should be present inside the music recommendation system option.	
Postcondition:	The actor has obtained search results based on their query.	

Use Case Name:	View Top Musics	
Actor:	User	
Use Case Description:	The actor can explore a section that displays the recommendation of top music tracks.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor selects the "Music Recommendation" option from the main menu. Step 3: The actor scrolls through the music recommendations.	Step 2: The system presents a section displaying the top music tracks based on factors such as popularity or user ratings.
Precondition:	The actor should be present inside the music recommendation system option.	
Postcondition:	The actor has viewed the top music tracks.	

Use Case Name:	Upload Fashion Images	
Actor:	User	
Use Case Description:	The actor can upload an image of a fashion-related item and find visually similar products	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor selects the "Fashion Recommendation" option from the main menu. Step 2: The actor selects the fashion images they want to upload from their local device or storage. Step 4: The actor is waiting for response from fashion recommendation.	Step 3: The system processes the uploaded fashion image.
Precondition:	The actor should be present inside the fashion recommendation system option.	
Postcondition:	The actor awaits fashion recommendation results.	

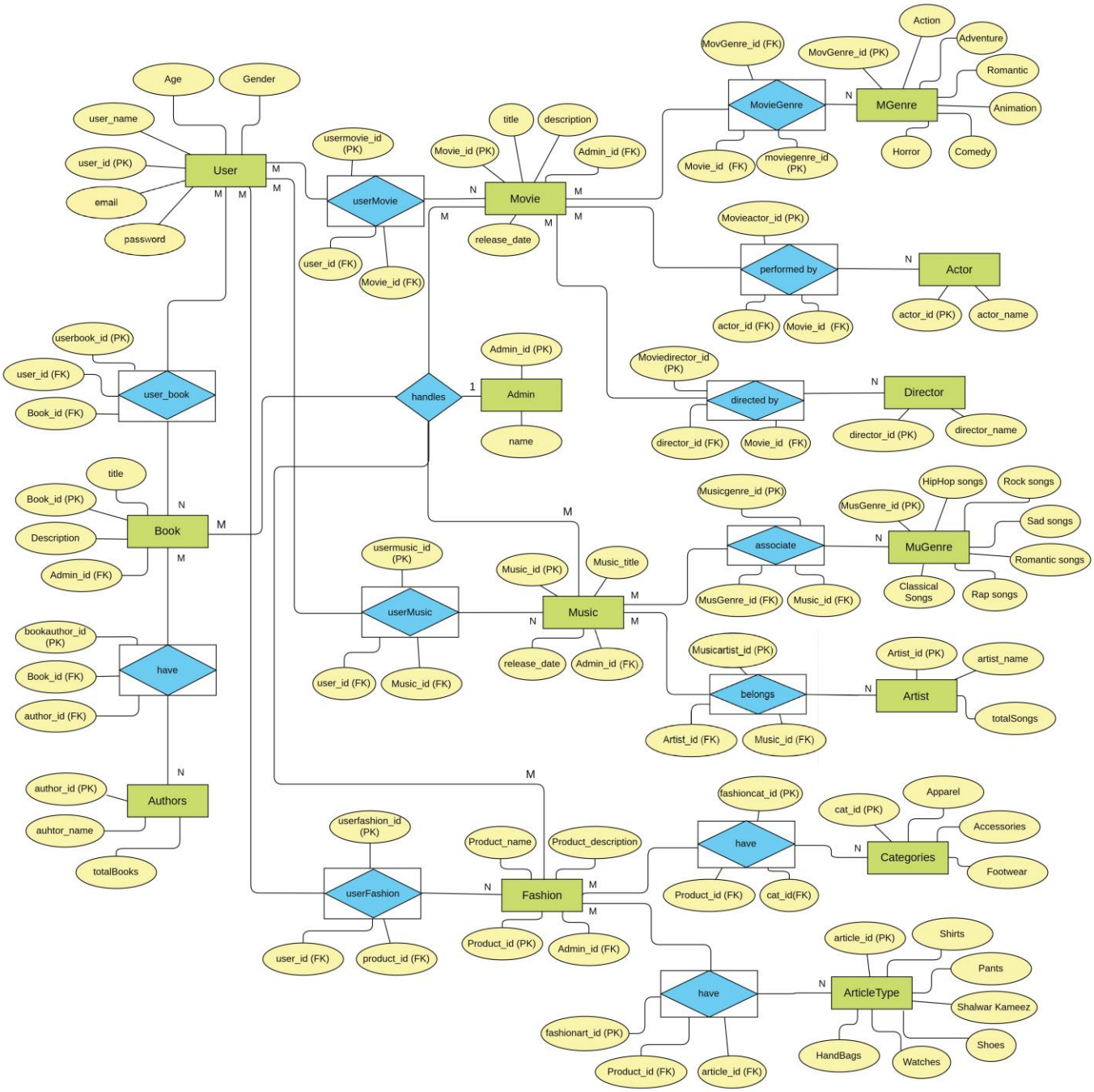
Use Case Name:	View Fashion Images	
Actor:	User	
Use Case Description:	The actor can explore a section that displays the recommendation of similar fashion images.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 3: The actor views the fashion recommendation results.	Step 1: The system presents recommendation in the form of grid of maximum 5 similar images. Step 2: The system displays the fashion images with details such as image title.
Precondition:	The actor should be present inside the fashion recommendation system option.	
Postcondition:	The actor has viewed the 5 visually similar images.	

Use Case Name:	Login	
Actor:	User	
Use Case Description:	This use case allow user to login into the system to access the relevant functions.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor enters the email and password and submits them to the system. Step 3: The actor can access all the recommendation systems.	Step 2: The system validates the actor's information, if a user exists, then the actor moves to the homepage, otherwise, it shows an error.
Precondition:	The actor has to have a valid account.	
Postcondition:	The system displays the relevant homepage.	

Use Case Name:	Create Categories	
Actor:	Admin	
Use Case Description:	The actor can create categories of movie, book. Music and fashion recommendation and handle them.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The actor can manage categories of movie, book, and music and fashion recommendation.	Step 2: The system handles the recommendation system algorithms which actor build for each recommendation system.
Precondition:	The actor should have access to the admin panel.	
Postcondition:	The actor manages the categories of recommendation system.	

Use Case Name:	Dataset Management	
Actor:	Admin	
Use Case Description:	The actor can manage datasets, including updating datasets of movie, book, and music and fashion recommendation.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 2: The actor can update the existing datasets to ensure that they provide an accurate recommendation from the up-to-date information.	Step 1: The system presents a list of existing datasets which is used for recommendation systems.
Precondition:	The actor should have access to the admin panel.	
Postcondition:	The datasets are successfully managed according to the admin's actions, ensuring accurate and up-to-date information.	

ER DIAGRAM:


User
Age
Gender
user_name
user_id (PK)
email
password
userMovie
usermovie_id (PK)
user_id (FK)
Movie_id (FK)
Movie
Movie_id (PK)
title
description
Admin_id (FK)
release_date
MovieGenre
MovGenre_id (FK)
MovGenre_id (PK)
Movie_id (FK)
moviegenre_id (PK)
MGenre
Action
Adventure
Romantic
Animation
Horror
Comedy
performed by
Movieactor_id (PK)
actor_id (FK)
Movie_id (FK)
Actor
actor_id (PK)
actor_name
directed by
Moviedirector_id (PK)
director_id (FK)
Movie_id (FK)
Director
director_id (PK)
director_name
handles
Admin
Admin_id (PK)
name
user_book
userbook_id (PK)
user_id (FK)
Book_id (FK)
Book
title
Book_id (PK)
Description
Admin_id (FK)
have
bookauthor_id (PK)
Book_id (FK)
author_id (FK)
Authors
author_id (PK)
auhtor_name
totalBooks
userMusic
usermusic_id (PK)
user_id (FK)
Music_id (FK)
Music
Music_id (PK)
Music_title
release_date
Admin_id (FK)
associate
Musicgenre_id (PK)
MusGenre_id (FK)
Music_id (FK)
MuGenre
MusGenre_id (PK)
HipHop songs
Rock songs
Sad songs
Romantic songs
Classical Songs
Rap songs
belongs
Musicartist_id (PK)
Artist_id (FK)
Music_id (FK)
Artist
Artist_id (PK)
artist_name
totalSongs
userFashion
userfashion_id (PK)
user_id (FK)
product_id (FK)
Fashion
Product_name
Product_description
Product_id (PK)
Admin_id (FK)
have
fashioncat_id (PK)
Product_id (FK)
cat_id(FK)
Categories
cat_id (PK)
Apparel
Accessories
Footwear
have
fashionart_id (PK)
Product_id (FK)
article_id (FK)
ArticleType
article_id (PK)
Shirts
Pants
Shalwar Kameez
HandBags
Watches
Shoes