Tribhuvan University

Institution of Science and Technology

A Final Year Project Report On

ANIME RECOMMENDATION SYSTEM USING KNN BASED COLLABORATING FILTERING ALGORITHHM

Under the Supervision of

Shikha Sharma

Full Time Faculty

Orchid International College

Submitted To:

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

ORCHID INTERNATIONAL COLLEGE

In partial fulfillment for the Bachelor's Degree in Computer Science and Information Technology

Submitted By:

Sanjita Tiwari (23887/076)

Utkarsha Bhandari (23905/076)

Vidit Pandey (23906/076)

March, 2024

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Sanjita Tiwari (23887/076), Utkarsha Bhandari (23905/076) and Vidit Pandey (23906/076) entitled "Anime Recommendation System Using KNN based Collaborative Filtering" In partial fulfillment of requirements for the degree of B.Sc. in Computer Science and Information Technology processed for the evaluation.

…………………………

Shikha Sharma

Project Supervisor,

Department of CSIT

Orchid International College

Bijaychowk, Gaushala

ORCHID INTERNATIONAL College

LETTER OF APPROVAL

This is to certify that this project prepared by Sanjita Tiwari (23887/076), Utkarsha Bhandari (23905/076) and Vidit Pandey (23906/076) entitled “**Anime Recommendation System Using KNN based Collaborative Filtering**” in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

…………………………. **Shikha Sharma** Project Supervisor, Department of Computer Science and IT Orchid International College Bijaychowk, Gaushala	………………………. **Er. Dhiraj Kumar Jha** Program Coordinator Department of Computer Science and IT Orchid International College Bijaychowk, Gaushala
……………………….. **External Examiner** Central Department of Computer Science and IT Tribhuvan University Kritipur, Nepal	

ACKNOWLEDGMENT

We would like to express our deepest appreciation to all those who provided us with the possibility to complete this project. Special gratitude to Project Supervisor, Dhiraj Kumar Jha, whose guidance, assistance, and encouragement helped us to coordinate our project and guided us on the correct path to maintain its progress.

We are extremely grateful to the Department of Computer Science and Information Technology for providing us with this opportunity to undertake this project and for providing all necessities for its completion. We are thankful to all the teachers from the department involved in the project work for their support and suggestions during proposals and presentations.

Last but not least, our thanks and appreciation go to all our friends and colleagues who helped and supported us continuously through different parts of the project, directly and indirectly.

Sanjita Tiwari (TU Exam Roll No: 23887/076)

Utkarsha Bhandari (TU Exam Roll No: 23905/076)

Vidit Pandey (TU Exam Roll No: 23906/076)

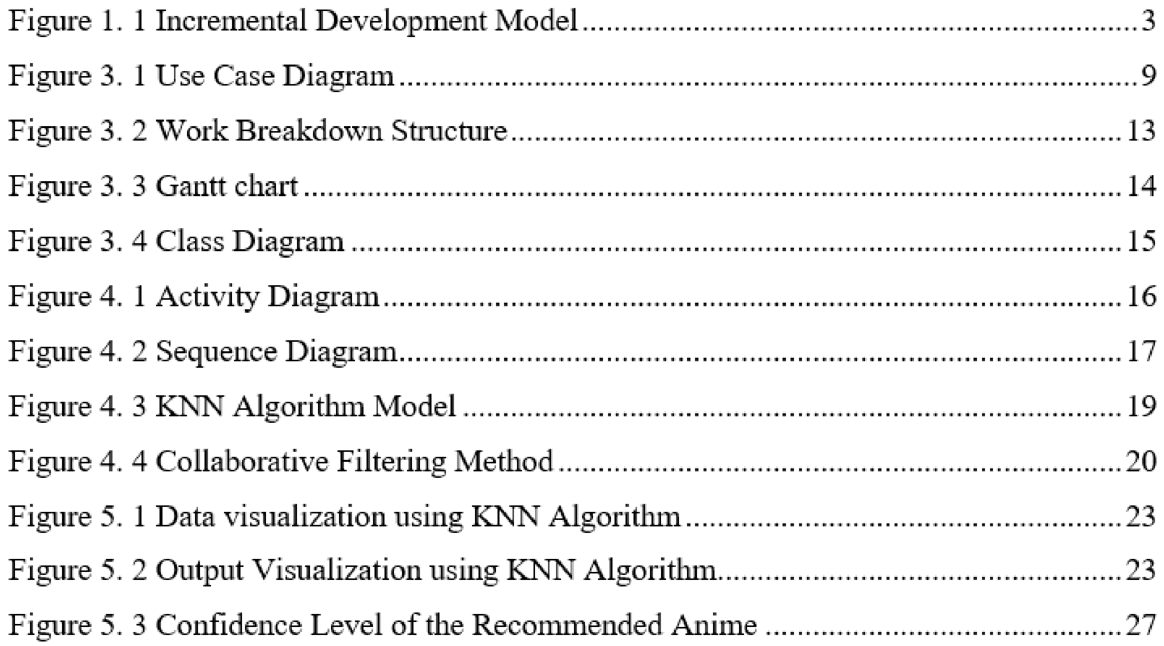
ABSTRACT

The Anime Recommendation System is a project that aims at addressing the demands for the genre of anime and providing effective recommendations to the user. Identifying the multiple factors, influencing anime choices, including rating of the anime, number of users, and variance in ratings, the project employs machine-learning algorithms, specifically K-Nearest Neighbor (KNN) and Collaborative Filtering. With a dataset comprising 16,300 records of Anime of different genres, the system objective is to recommend different anime based on the rating of the anime provided by the user, on which further recommendations are based upon. During implementation, KNN algorithms are built and rigorously evaluated for confidence level in terms of different anime recommendations provided by the system to the user determining the most effective recommendation list. The user interface is created using HTML and CSS for easy interaction. Django application has been used for web framework, featuring functionalities such as user registration and login, along with validation of the user credentials. To store data, SQLite database management system is employed. The system not only provides user with new recommendation and list of anime but also does so with about 79-81% confidence. As a result, the project seeks to enhance user experience by incorporating additional functionalities such as providing the score of the listed recommendations offering better and effective choice among the recommended list. By combining multiple datasets, which includes anime information, user ratings, and individual anime's rating, it contributes in making the system more reliable and efficient.

Keywords: K Nearest Neighbor Algorithm, Collaborative Filtering Algorithm, Anime recommendations.

TABLE OF CONTENTS

LIST OF FIGURES



LIST OF TABLES

LIST OF ABBREVIATIONS

KNN	K Nearest Neighbor
CF	Collaborative Filtering
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
DF	Data Frame
UML	Unified Modeling Language

CHAPTER 1: INTRODUCTION

1.1 Introduction

In recent times popularity of the genre anime has skyrocketed. For being an animated show, it caters to the wide variety of audience with multiple genres and compelling storyline among various other entertainment platforms. Anime has grown into a huge entertainment platform grossing over thousands of shows. 36% (or 2.88 billion) of viewers worldwide enjoy watching anime. Anime is the 3rd most in-demand subgenre worldwide, with a 5.5% demand in entertainment market where over 100 million households globally watched at least 1 anime title on Netflix [1]. Despite the abundance of information available to users over the internet, there are often limited sources on next recommendation of anime shows for them based upon current anime they've watched and its ratings. Hence needing a system to effectively recommend anime to the users. Thus, machine learning algorithms, specifically the K Nearest Neighbor and Collaborative Filtering, were built as system to figure out effective recommendation of anime shows to the user.

This project employs data sourced from the Anime Recommendation Database 2020 [2]. In total, three different datasets were used which includes Anime Information dataset with 16300 records, User dataset with 320000 users' anime records and Rating dataset with users' rating on every anime. This comprehensive dataset contains various attributes including anime ID, title, user ID, user ratings, overall scores, and supplementary information. These data points serve as inputs for training machine learning algorithm. This model is operationalized via a web-based application developed with HTML, CSS, Python and Django, leveraging SQLite as the relational database management system to ensure efficient data storage and retrieval. Additionally, the system incorporates user-friendly functionalities such as registration and login, along with an interactive user interface for searching anime recommendations.

1.2 Problem Statement

In the realm of entertainment consumption, particularly within the anime community, there exists a growing need for personalized and accurate recommendations. Existing recommendation systems often fall short in providing tailored suggestions that align with the unique tastes and preferences of individual users. Consequently, users are provided with choices that may not resonate with their interests, leading to frustration and disengagement.

To address this challenge, the development of an advanced anime recommendation system is essential. This system aims to leverage machine learning algorithm and technique to analyze user behavior, preferences, and historical interactions with anime content. By integrating multiple datasets, this system endeavors to enhance its recommendation capabilities.

The recommendation system aims to provide tailored suggestions that match users' preferences, enriching their viewing experience.

1.3 Objectives

- To build a machine-learning model using KNN algorithm for anime recommendation system.
- To load and preprocess the datasets, filter out data based on specific criteria and merge datasets to combine information.

1.4 Scope and Limitation

The project aims to implement a system that can take input from users and generate recommendations list based for the user based on the rating of the show provided to the system.

The literature study conducted by others on the same topic has the following limitations:

- The studies reviewed had no recommendations based upon the rating of the anime shows.
- The studies reviewed failed to consider the specific anime preferences of individual users.

1.5 Development Methodology

The project adopts a plan-driven incremental development approach, facilitating the construction of software systems component by component. This model enables the clear delineation of final requirements from the outset. It entails creating an initial version, obtaining feedback, and iteratively refining the software through successive versions until the desired system is achieved. Consequently, the incremental methodology progresses the system in a series of increments or versions, with each iteration augmenting the functionality of the preceding one.

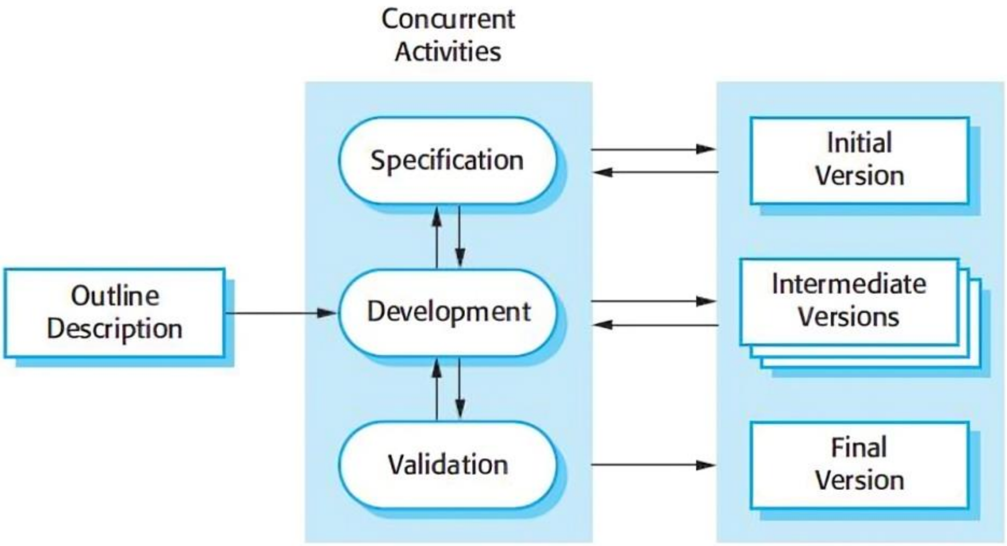


Figure 1.1 Incremental Development Model

1.6 Report Organization

Report organization describes the steps by which the project’s report has been organized. The project follows the following steps:

- Introduction
- Background Study
- Literature Review
- System Analysis
- System Design
- Implementation
- Testing
- Conclusion

CHAPTER 2: LITERATURE REVIEW

2.1 Background Study

Anime, originating from Japan, has gained a global following due to its rich storytelling, unique art styles, and diverse genres. However, the increasing volume of available anime poses a challenge for viewers in discovering content aligned with their preferences. To address this, anime recommendation systems have emerged, leveraging machine learning algorithms to assist users in finding relevant content tailored to their tastes.

Recommendation systems aim to enhance user engagement and satisfaction by offering personalized suggestions. Traditional methods like manual browsing or keyword searches often fail to provide efficient recommendations. By employing data analytics, machine learning algorithms, and user feedback, recommendation systems offer an effective way for users to explore and engage with anime content.

Approaches to anime recommendation systems include collaborative filtering, content-based filtering, K Nearest Neighbor (KNN) and hybrid methodology. Collaborative filtering (CF) analyzes user interactions to generate recommendations based on similar user preferences. Content-based filtering examines anime characteristics like genre and art style to suggest similar content. KNN takes the provided input and generates recommendation through calculating the nearest distance of the provided input. Hybrid systems combine both approaches for more accurate and diverse recommendations.

Successful anime recommendation systems depend on comprehensive datasets and ongoing user feedback. These systems have the potential to revolutionize content discovery, driving greater engagement with anime platforms. However, challenges such as data sparsity and algorithmic bias must be addressed to ensure effectiveness in catering to the diverse preferences of anime enthusiasts worldwide.

2.2 Literature Review

Understanding user behavior in anime recommendation systems is paramount for optimizing user engagement and satisfaction. Despite the growing popularity of anime, there remains a lack of insight among platform administrators regarding the reasons behind users' decisions to forego recommended titles. This lack of understanding poses a significant challenge for the effective management of anime recommendation systems, as it hampers efforts to tailor recommendations to individual user preferences and interests. By leveraging machine learning algorithms, it becomes possible to identify users' preferences and recommend tailored suggestions to the users. Just as demand forecasting and revenue management are intertwined in the hospitality industry, so too are user engagement and recommendation systems in the medium of anime streaming platforms. Through the application of machine learning techniques, recommendation systems can enhance user satisfaction and maximize effective recommendations to the users by delivering personalized suggestions that resonate with each user's unique tastes and preferences.

To build and develop domain expertise for the project, a careful analysis was conducted on several research papers published in specialized areas such as machine learning algorithms of these system. The insights gained from these publications significantly contributed to enhancing understanding of the project's domain, operations, and system functionality, as mentioned below.

A study performed by A S Girsang, B Al Faruq, H R Herlianto, and S Simbolon [3] explores a collaborative recommendation system for anime films, leveraging collaborative filtering on a Kaggle dataset of 73,516 users and 12,294 anime. Utilizing the Alternating Least Squares (ALS) method, it matches user histories to suggest anime, aiming to assist millions in finding desired content. The study discusses problem formulations, normalization for new users, similarity matrices, and top movie recommendations based on feature differences. Model evaluations showcase the effectiveness of collaborative filtering, presenting a simple yet efficient anime recommendation system.

Furthermore, the effectiveness of the collaborative filtering model is rigorously evaluated through various metrics and benchmarks. These evaluations serve to demonstrate the practical viability and efficiency of the anime recommendation system proposed in the study. By presenting empirical evidence of its performance and efficacy, the research underscores the potential of collaborative filtering as a simple yet potent tool in the realm of anime recommendation. In summary, the study contributes valuable insights into the development and implementation of collaborative recommendation systems tailored for anime enthusiasts. By leveraging collaborative filtering techniques and innovative methodologies, it seeks to empower users in their quest for personalized and relevant anime content.

Another significant contribution in the medium of anime recommendation systems comes from a Nuurshadieq, Agung Toto Wibowo [4], which introduced a novel approach utilizing deep neural networks. This method intricately integrates both user-specific information and detailed attributes of anime content, placing particular emphasis on textual synopsis analysis. Through rigorous evaluations employing Root Mean Square Error (RMSE) as a metric, this approach demonstrated highly promising results, surpassing conventional methods such as Singular Value Decomposition (SVD) and K-Nearest Neighbors (KNN). Notably, the integration of anime-related data into the recommendation process significantly enhanced the accuracy and relevance of the suggested titles. Moreover, the inclusion of textual information, especially from the synopsis, emerged as a crucial factor in improving recommendation quality, even in scenarios with limited data availability. This innovative approach represents a significant advancement in anime recommendation systems, offering a more robust and effective method for generating recommendations.

Furthermore, research by Shardul Rudraksha1, Vipul Munot, Satyadev Mishra, Swapnil Misal, Yogita Kandhare, Abhilasha Kulkarni [5] extensively explored recommendation systems across various domains, ranging from movies and books to music and news. Content-based filtering and collaborative filtering have emerged as prominent techniques in this field. Content-based filtering relies on user profiles and item attributes to recommend relevant items, while collaborative filtering suggests items based on similarities between users or items. However, both approaches have inherent limitations; content-based filtering may suffer from inaccuracies in item attributes, while collaborative filtering may struggle with new users or items. To address these challenges, researchers have proposed hybrid recommendation systems, which combine elements of both approaches to achieve better performance. These hybrid systems offer various approaches, including weighted hybridization, switching hybridization, mixed hybridization, feature combination, and cascade. These methods aim to optimize recommendation accuracy and overcome the shortcomings of individual filtering techniques. In the context of anime recommendation, the hybrid approach presents a promising solution to provide users with more efficient and accurate anime suggestions, leveraging both content-based and collaborative filtering mechanisms. Through an analysis of different hybridization methods, this study seeks to contribute to the ongoing research in enhancing recommendation systems for anime enthusiasts.

CHAPTER 3: SYSTEM ANALYSIS

3.1 System Analysis

3.1.1 Requirement Analysis

The goal of the project is to build an anime recommendation system using KNN based collaborative filtering that can effectively recommend the list of anime with their respective scores/ratings.

i. Functional Requirements

The anime recommendation system using KNN based collaborative filtering must perform following functions:

- Register into the website.
- Login into the website
- Search Anime
- Recommend Anime

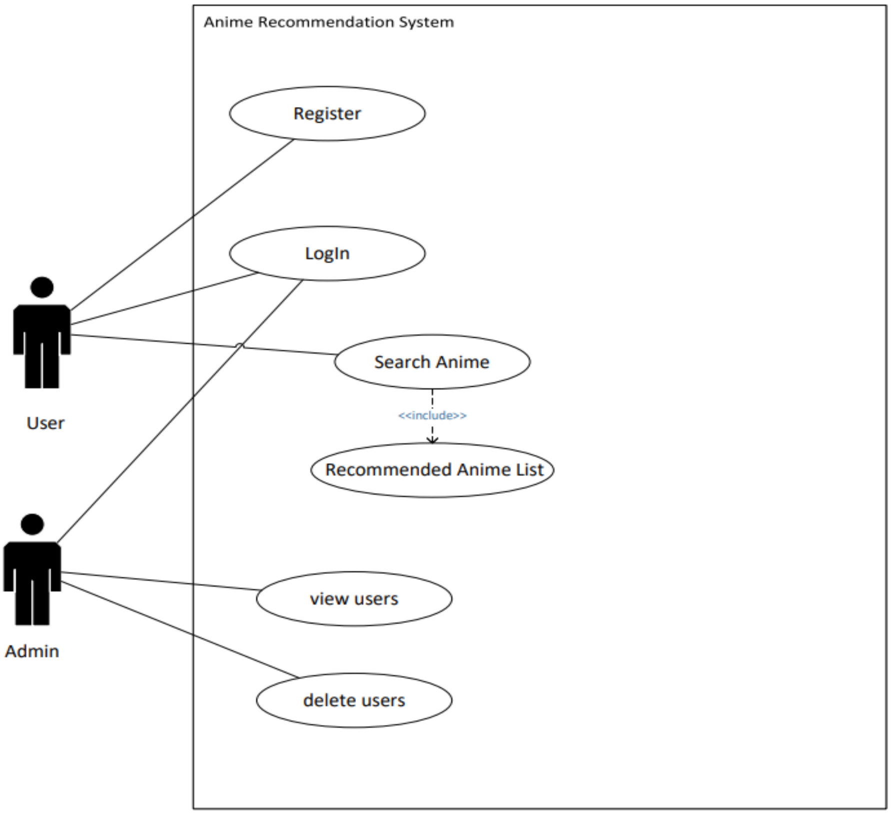


Figure 3. 1 Use Case Diagram

Use Case Description:

Table 3. 1 Use Case for User Registration

Use-Case Identifier	UC1- Register into website
Primary Actor	User
Secondary Actor	None
Description	User registers and enters into the system
Pre-Condition	User should not have been registered previously
Success Scenario	Successful user registration and redirect to the login page
Failure Scenario	User is not registered

Table 3. 2 Use Case for User Login

Use-Case Identifier	UC2- Login into website
Primary Actor	User
Secondary Actor	None
Description	User login with their credentials
Pre-Condition	User is registered and has login credentials
Success Scenario	User is logged in successfully into the system
Failure Scenario	User receives a notification indicating login has failed

Table 3. 3 Use Case for User Input

Use-Case Identifier	UC3- Enter Data
Primary Actor	User
Secondary Actor	None
Description	User enters title of the anime they want recommendation for
Pre-Condition	User had successfully logged into the system
Success Scenario	User enter correct anime title
Failure Scenario	User enters wrong anime title

Table 3. 4 Use Case for Anime Recommendation

Use-Case Identifier	UC4-Anime Recommendation
Primary Actor	User
Secondary Actor	None
Description	User clicks show recommendation button
Pre-Condition	User has entered anime title correctly.
Success Scenario	List of recommended anime is displayed on the screen
Failure Scenario	“No recommendation available” message is displayed

ii. Non-Functional Requirements

Non-functional requirements define the operational characteristics and quality attributes of a system, such as reliability, scalability, and security. They focus on how the system behaves rather than specific features or functionalities. These criteria are essential for guiding the design and evaluation of the system to meet stakeholders' expectations and ensure overall effectiveness. The non- functional requirements for this project are shown as:

Reliability: The system should recommend the anime as per the user input. The system be reliable and should not crash or malfunction.

Performance: The recommendation time should be as minimum as possible making the system more efficient and responsive.

Flexibility: The system should be flexible for the addition of various new anime shows and their information for further recommendation.

Usability: This ensures that users can interact with the system seamlessly and without unnecessary complexity, enhancing overall usability.

3.1.2. Feasibility Analysis

Feasibility analysis assesses the practicality of a project by evaluating its technical, economic, and operational viability within given constraints. Its goal is to determine whether the project is feasible and viable for implementation. Following are the feasibility analysis that were conducted during the system analysis phase.

i. Technical

Technical feasibility describes about the hardware and software components that are required by the system development methodology. Technical feasibility ensures that the hardware can handle K Nearest Neighbour based Collaborative Filtering algorithm. It should be compatible with necessary software tools (Jupyter Notebook, Django) and libraries. It also encompasses the data availability required for the project.

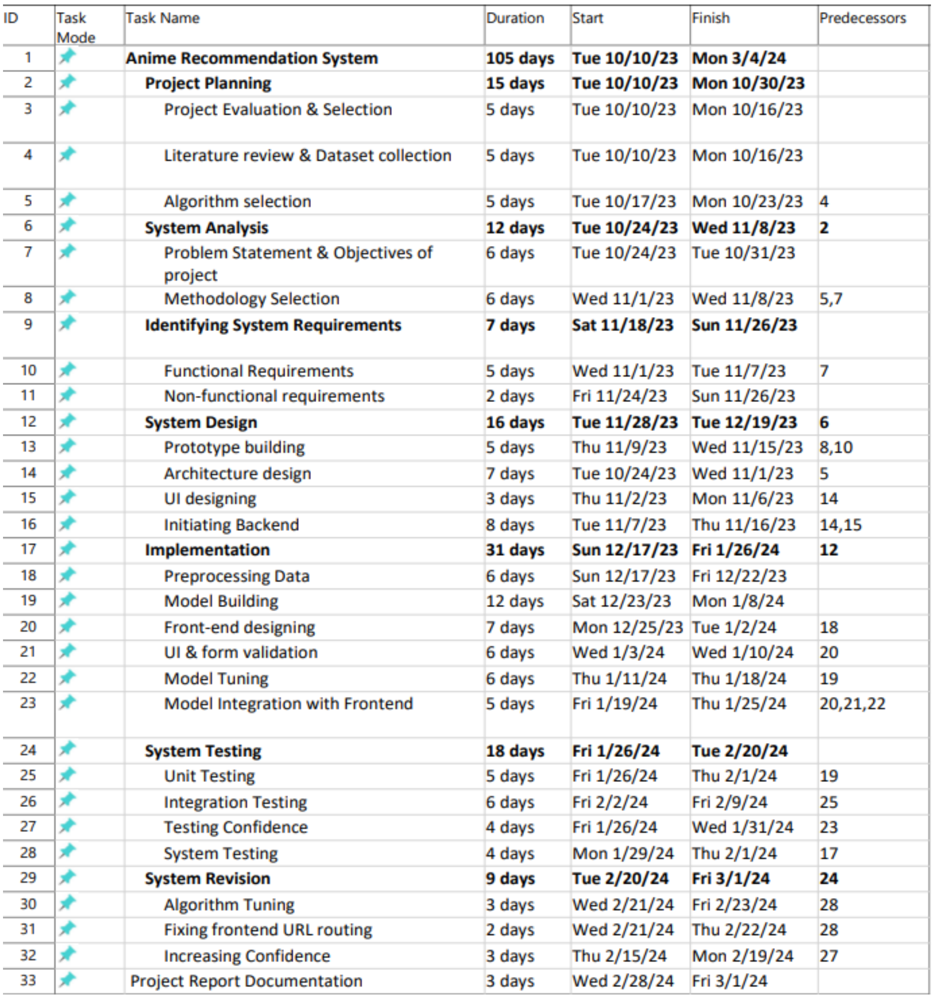
ii. Operational

Operational feasibility for an anime recommendation system as a project involves assessing whether the system can be effectively integrated into existing standalone applications. This includes evaluating factors such as user interface design, ease of navigation and system performance with different devices. Overall, the goal is to ensure that the system can be successfully deployed and used by anime enthusiasts, meeting their needs and expectations effectively.

iii. Schedule

The schedule feasibility evaluation for this project, projected to last around 5 months, indicates a practical timeline for its development and implementation. The timeline corresponds to the expected timeframe for completing essential tasks and meeting project goals.

The project's work breakdown structure is as follows:



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1	📌	**Anime Recommendation System**	**105 days**	**Tue 10/10/23**	**Mon 3/4/24**	
2	📌	**Project Planning**	**15 days**	**Tue 10/10/23**	**Mon 10/30/23**	
3	📌	Project Evaluation & Selection	5 days	Tue 10/10/23	Mon 10/16/23	
4	📌	Literature review & Dataset collection	5 days	Tue 10/10/23	Mon 10/16/23	
5	📌	Algorithm selection	5 days	Tue 10/17/23	Mon 10/23/23	4
6	📌	**System Analysis**	**12 days**	**Tue 10/24/23**	**Wed 11/8/23**	**2**
7	📌	Problem Statement & Objectives of project	6 days	Tue 10/24/23	Tue 10/31/23	
8	📌	Methodology Selection	6 days	Wed 11/1/23	Wed 11/8/23	5,7
9	📌	**Identifying System Requirements**	**7 days**	**Sat 11/18/23**	**Sun 11/26/23**	
10	📌	Functional Requirements	5 days	Wed 11/1/23	Tue 11/7/23	7
11	📌	Non-functional requirements	2 days	Fri 11/24/23	Sun 11/26/23	
12	📌	**System Design**	**16 days**	**Tue 11/28/23**	**Tue 12/19/23**	**6**
13	📌	Prototype building	5 days	Thu 11/9/23	Wed 11/15/23	8,10
14	📌	Architecture design	7 days	Tue 10/24/23	Wed 11/1/23	5
15	📌	UI designing	3 days	Thu 11/2/23	Mon 11/6/23	14
16	📌	Initiating Backend	8 days	Tue 11/7/23	Thu 11/16/23	14,15
17	📌	**Implementation**	**31 days**	**Sun 12/17/23**	**Fri 1/26/24**	**12**
18	📌	Preprocessing Data	6 days	Sun 12/17/23	Fri 12/22/23	
19	📌	Model Building	12 days	Sat 12/23/23	Mon 1/8/24	
20	📌	Front-end designing	7 days	Mon 12/25/23	Tue 1/2/24	18
21	📌	UI & form validation	6 days	Wed 1/3/24	Wed 1/10/24	20
22	📌	Model Tuning	6 days	Thu 1/11/24	Thu 1/18/24	19
23	📌	Model Integration with Frontend	5 days	Fri 1/19/24	Thu 1/25/24	20,21,22
24	📌	**System Testing**	**18 days**	**Fri 1/26/24**	**Tue 2/20/24**	
25	📌	Unit Testing	5 days	Fri 1/26/24	Thu 2/1/24	19
26	📌	Integration Testing	6 days	Fri 2/2/24	Fri 2/9/24	25
27	📌	Testing Confidence	4 days	Fri 1/26/24	Wed 1/31/24	23
28	📌	System Testing	4 days	Mon 1/29/24	Thu 2/1/24	17
29	📌	**System Revision**	**9 days**	**Tue 2/20/24**	**Fri 3/1/24**	**24**
30	📌	Algorithm Tuning	3 days	Wed 2/21/24	Fri 2/23/24	28
31	📌	Fixing frontend URL routing	2 days	Wed 2/21/24	Thu 2/22/24	28
32	📌	Increasing Confidence	3 days	Thu 2/15/24	Mon 2/19/24	27
33	📌	Project Report Documentation	3 days	Wed 2/28/24	Fri 3/1/24	

Figure 3. 2 Work Breakdown Structure

Here is the Gantt chart for the Anime Recommendation System:

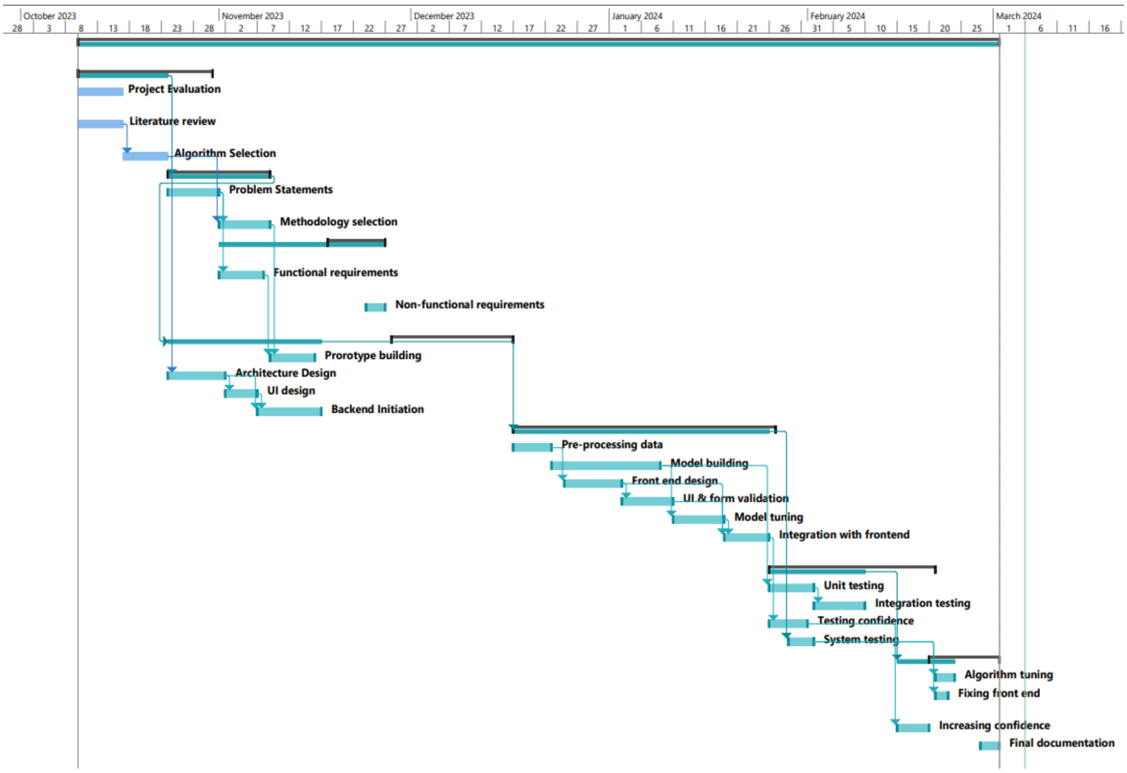


Figure 3. 3 Gantt chart

User Requirements

The system designed will be user friendly and easy to navigate. The user should provide title of the anime in the search bar.

System Requirements

The system requirements include software and hardware requirements that are required to run the program efficiently.

Software Requirements:

- Operating System: Windows, MAC OS
- Web browser: any web browser

Hardware Requirements

- Processor: Intel i3 or above
- Processor speed: 1.0 GHz or above
- RAM: 4 GB or above

3.1.3 Analysis

During the analysis phase, the project's requirements will be organized. Data modeling and preprocessing techniques will be employed for this purpose.

i. Class Diagram

A class diagram visually depicts the static structure and relationships within a system, including classes, attributes, methods, and associations. Crucial in object-oriented design, it provides a blueprint for software architecture.

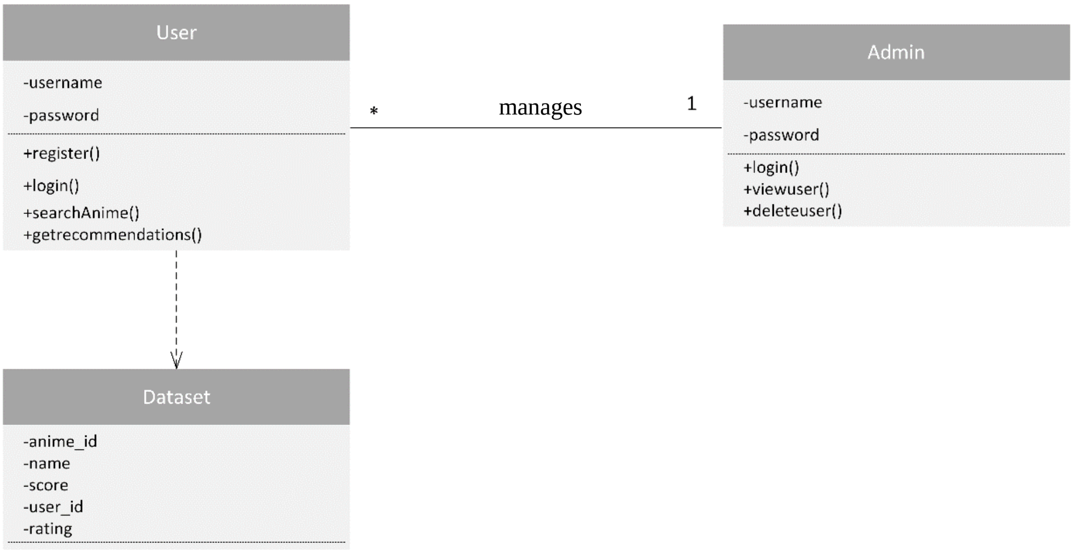


Figure 3. 4 Class Diagram

The above given is a class diagram that shows the interaction between users and administrators of a system, likely a database system. The diagram shows that users can register and login to the system using a username and password. They can then search for anime, get recommendations, and view their ratings. The diagram also shows that administrators can manage users, including deleting them. They can also view all of the data in the database, including the anime ID, name, score, user ID, and rating. Overall, the diagram shows that users have limited access to the system, while administrators have full access.

CHAPTER 4: SYSTEM DESIGN

System design involves creating a blueprint that outlines the architecture, components, interfaces, and data of a system to meet specified requirements. It aims to ensure the resulting system is scalable, reliable, and fulfills user needs effectively. It includes:

4.1 Activity Diagram

An Activity Diagram serves as a visual representation of a workflow, illustrating the sequence of activities and decision points along with the flow of control between them. Commonly employed in software engineering, business process modeling, and similar domains, these diagrams offer insight into complex processes. They afford a high-level overview of system behavior, facilitating the detection of possible inefficiencies or areas for improvement. In an activity diagram, activities are depicted as nodes, while transitions between activities are represented by arrows.

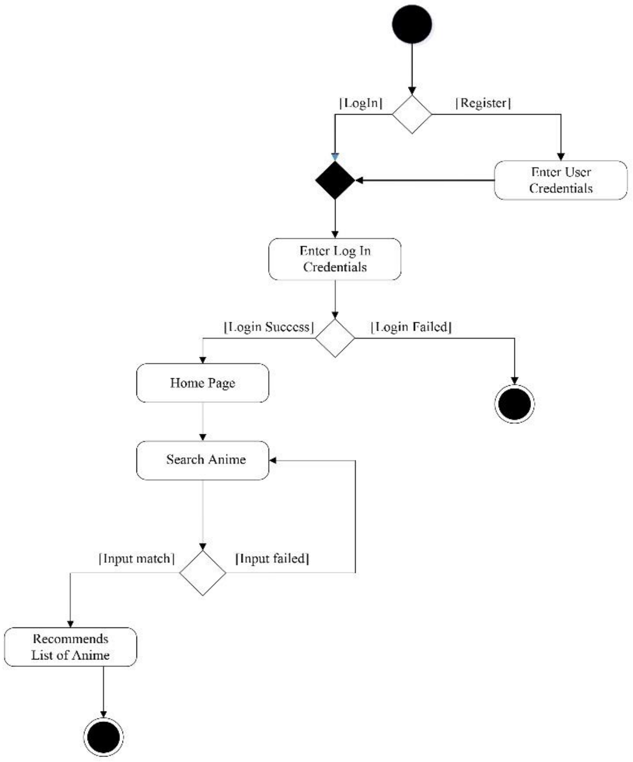


Figure 4. 1 Activity Diagram

This activity diagram maps out how users find anime recommendations on a website. It begins with users logging in with existing credentials or registering for a new account. Once logged in, they can search for anime by entering a title or keyword. The system then consults its database for matches. If a match is found, the user is rewarded with a curated list of similar anime to explore, expanding their anime horizons. However, if the search comes up empty, the user receives a "search failed" message, prompting them to adjust their search terms.

4.2 Sequence Diagram

A sequence diagram, a type of UML diagram, portrays the interactions among system elements over time. It depicts the chronological sequence of interactions, emphasizing the spatial order of communications exchanged between objects. In sequence diagrams, objects are depicted as vertical rectangles, and messages are shown as horizontal arrows connecting lifelines. These messages are labeled to indicate the information transmitted, while the arrangement of arrows reflects their sequential occurrence.

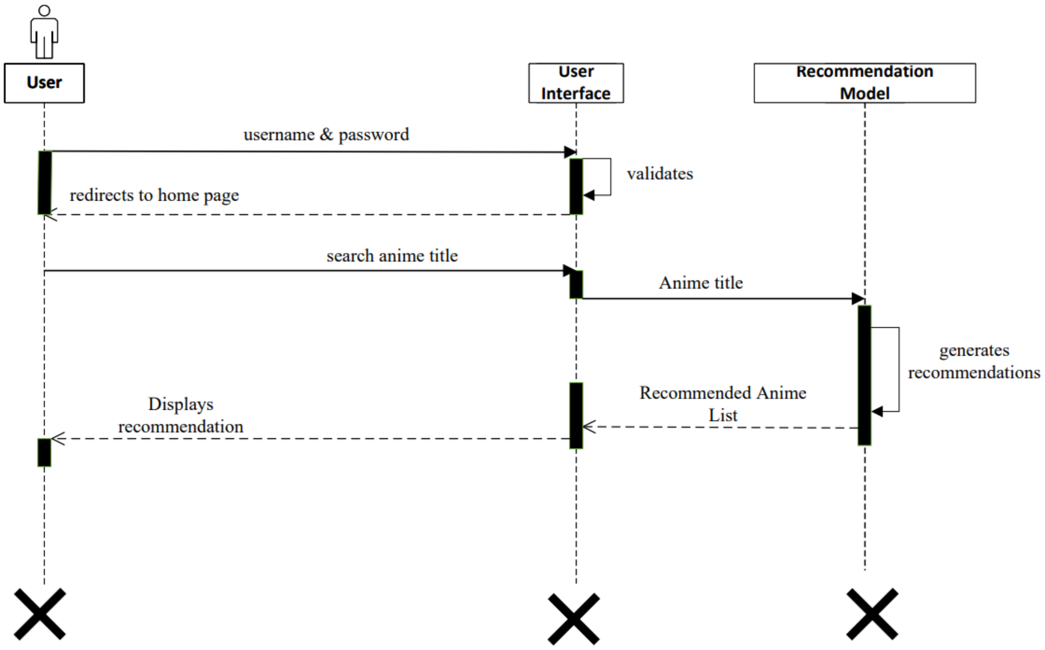


Figure 4. 2 Sequence Diagram

In the above sequence diagram, it shows how a user goes through to find anime recommendations on a website. It starts with the user entering their login credentials, username and password. Following successful validation, the system directs the user to the homepage. This homepage provides the user with a search function where they can enter the title of a desired anime. The system then checks its database to see if there’s a match. If a match is found, the website generates a recommended list of similar anime titles, offering the user a broader selection of choices. However, if the search yields no results, the user is informed of the unsuccessful search attempt. This allows them to either refine their search query or browse the website's collection for other anime titles.

4.2 Algorithm Details

To build the anime recommendation system model, we began by generating a pivot table from the available datasets, incorporating all essential features. Following data preprocessing within the pivot table, we stored the information in a sparse matrix for efficient handling of large datasets. Implementing the KNN (K-Nearest Neighbors) algorithm, we facilitated the search process. When a search index is applied, the algorithm identifies all non-zero values for that index and calculates distances between them and other values. Subsequently, the algorithm forms nearest neighbors and clusters. Upon calling for a specific anime, the algorithm examines these clusters, identifying nearest neighbors, and stores their values as recommendations. Additionally, it calculates distances to gauge the system's efficiency and accuracy.

K Nearest Neighbour:

The k-nearest neighbors (KNN) algorithm is a straightforward yet effective machine learning approach used for both classification and regression tasks. In KNN, classification involves assigning a class label to a new data point based on the majority vote of its k-nearest neighbors in the feature space, while regression predicts the target value by averaging the values of its k-nearest neighbors. During recommendation, the algorithm calculates the distance between the new data point and all other points in the training set using metrics like Euclidean or Manhattan distance. It then selects the k-nearest neighbors and determines the outcome based on their collective input. It doesn't construct a model during training but instead compares the new data point to the stored training data at recommendation time. While KNN is intuitive and versatile, the choice of the parameter k significantly impacts its performance, and it can be computationally expensive with large datasets due to the need to calculate distances between data points. Nevertheless, KNN remains a valuable tool for various classification and regression tasks.

Manhattan Distance Formula = $| x_2-x_1 | + | y_2-y_1 |$

Euclidean Distance Formula = $\sqrt{(x_2-x_1)^2-(y_2-y_1)^2}$

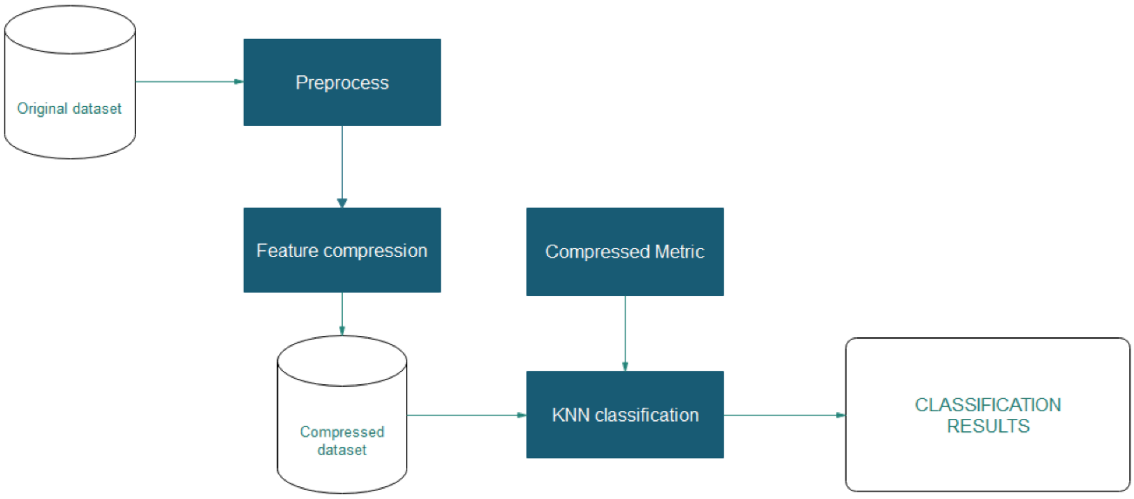


Figure 4. 3 KNN Algorithm Model

Collaborative Filtering Algorithm:

Collaborative filtering is a recommendation system technique that analyzes user interactions and item preferences to make personalized suggestions. It operates by identifying similarities between users or items based on their past interactions, such as ratings or purchases. In user-based collaborative filtering, recommendations are made by finding users with similar preferences and suggesting items they have liked. Conversely, item-based collaborative filtering recommends items that are similar to ones the user has previously shown interest in. By leveraging the collective wisdom of users, collaborative filtering enhances recommendation accuracy and user satisfaction across various platforms and domains.

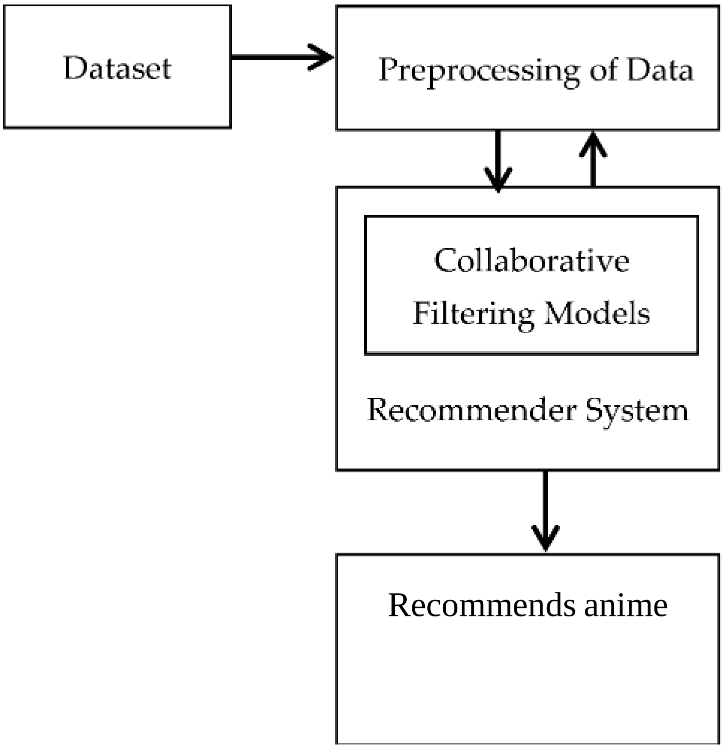


Figure 4. 4 Collaborative Filtering Method

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1 Tools Used

In the implementation phase of the project, the following tools were employed for project development and design of the system:

Development Tools

Tools	**Description**
Django	Web framework for building web applications.
Jupyter Notebook	Interactive environment for application development
Visual Studio Code	Lightweight source code editor.
Python	Primary programming language for development
HTML/CSS	Markup and styling for the frontend
Git and GitHub	Version control and collaboration platform

Design and Documentation Tools

Design and documentation tools were essential in creating the system architecture, user interfaces, and project documents.

Tools	**Description**
MS Visio	Diagramming and vector graphics application tool.
MS Project	Project management software for planning, tracking, and managing projects.
MS Word	Word processing tool for project documents

Dependencies

Package	**Description**
NumPy	For numerical computing and array manipulation.
Matplotlib	For versatile data visualizations.
Pandas	For data manipulation and analysis
Pickle	For serializing and deserializing python objects into byte streams

Database Platform

Tools	**Description**
SQLite	Lightweight, serverless, self-contained SQL database engine.

5.1.2 Implementation Details of Module

Description:

This module implements K-nearest neighbors (KNN) based collaborative filtering a to recommend anime to users. It operates by grouping users into clusters according to their ratings for individual shows and calculating distances between them to evaluate accuracy. By leveraging these clusters, the algorithm effectively identifies users with similar preferences and suggests anime that they may enjoy. Additionally, the module includes functions for model training, indexing searches, and cluster creation, facilitating efficient recommendation generation and system optimization.

Data cluster using KNN Algorithm:

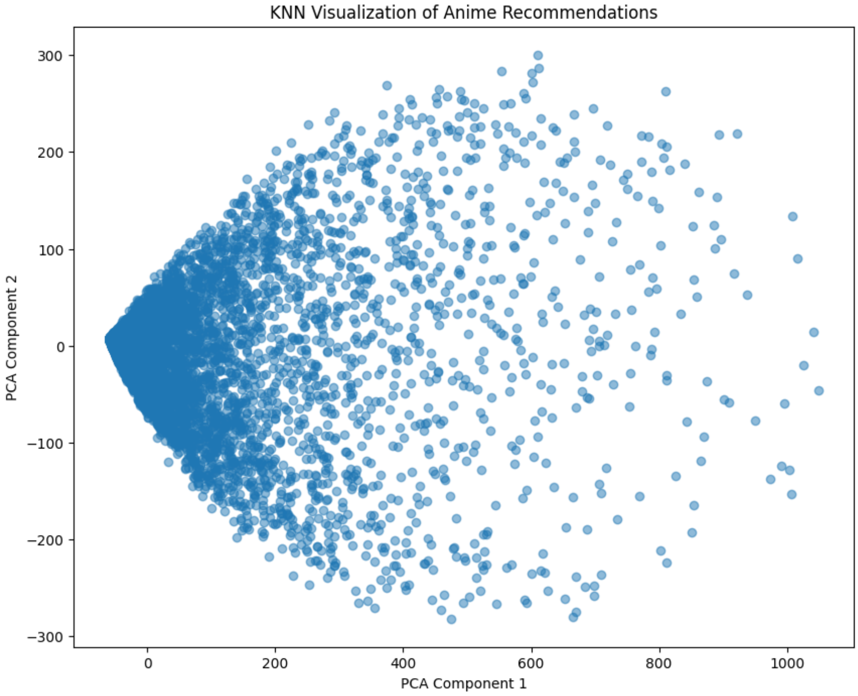


Figure 5. 1 Data visualization using KNN Algorithm

Output Cluster Using KNN Algorithm

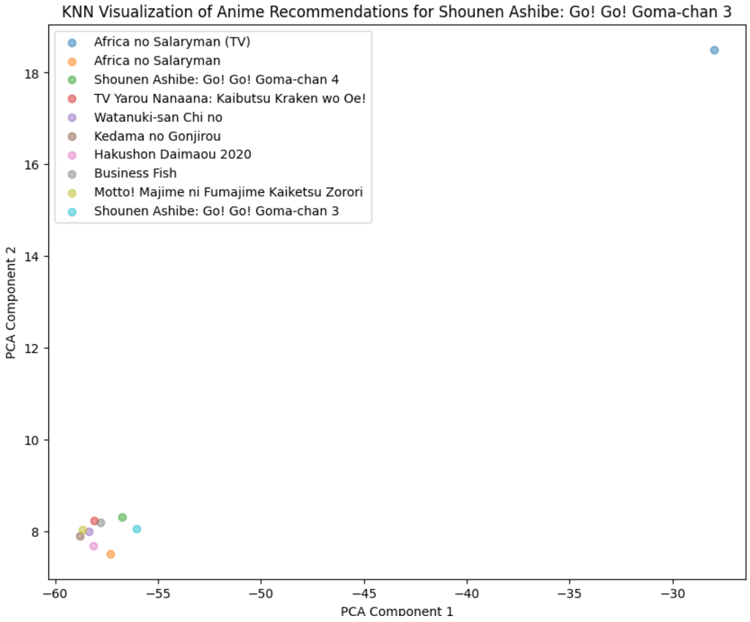


Figure 5. 2 Output Visualization using KNN Algorithm

5.2 Testing

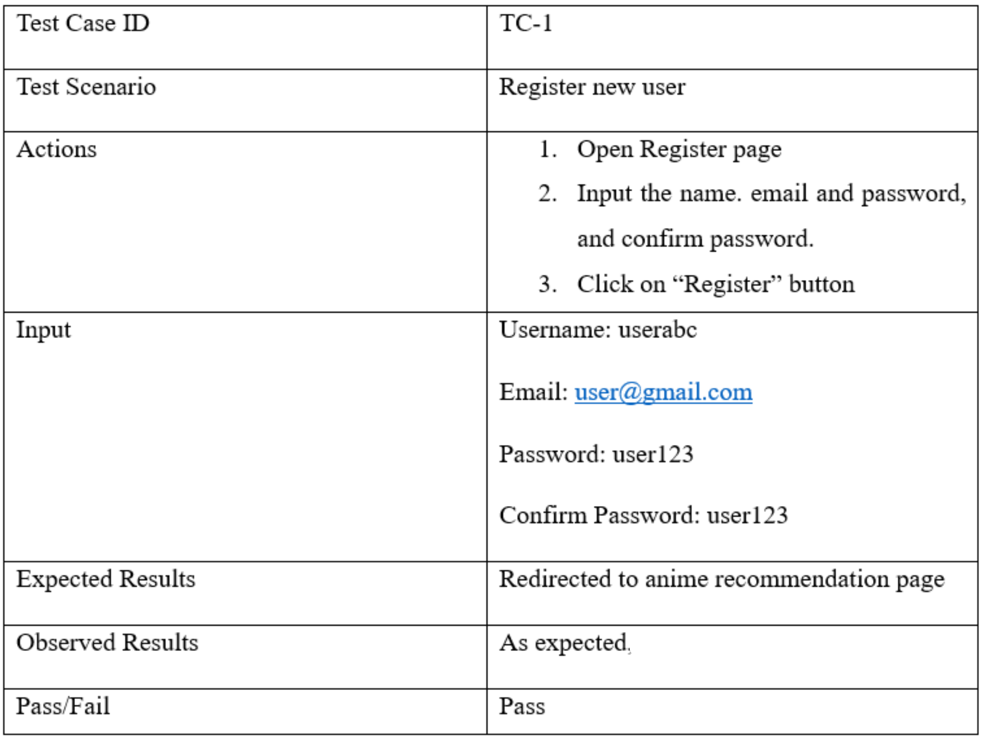
Testing involves assessing a software application or system to detect differences between anticipated and observed behavior, guaranteeing adherence to predefined criteria. The main goals of testing in a software project encompass:

- **Verification**: Testing verifies that the software meets the specified requirements and functions correctly according to its design and intended behavior.
- **Validation**: Testing confirms that the software aligns with user requirements and expectations, effectively fulfilling its designated functions.
- **Identification of Defects**: Testing aids in pinpointing flaws, glitches, or anomalies within the software that could result in inaccurate or unforeseen outcomes.
- **Quality Assurance**: Testing is a crucial component of quality assurance processes, helping to ensure that the software meets quality standards and adheres to best practices.
- **Risk Mitigation**: Testing aids in recognizing and addressing risks linked to the software, including security vulnerabilities, performance concerns, or usability challenges.

5.2.1 Test Cases for Unit Testing

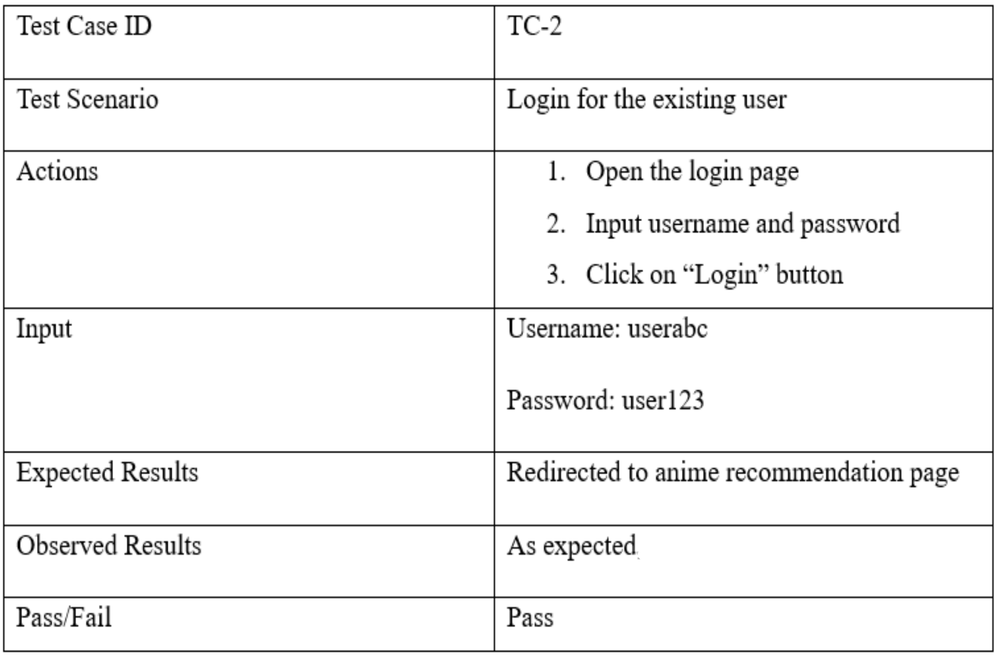
Unit testing is an essential practice in software development, allowing developers to verify the correctness of each unit's behavior in isolation. By testing units independently, developers can detect and fix defects early in the development process, leading to more robust and maintainable software. Additionally, unit testing facilitates easier debugging and refactoring, as issues can be localized to specific units rather than being intertwined with the broader system functionality. Overall, unit testing contributes to the overall quality and reliability of software systems.

Table 5. 1 Unit Test Case for Registration



Test Case ID	TC-1
Test Scenario	Register new user
Actions	1. Open Register page 2. Input the name. email and password, and confirm password. 3. Click on “Register” button
Input	Username: userabc Email: user@gmail.com Password: user123 Confirm Password: user123
Expected Results	Redirected to anime recommendation page
Observed Results	As expected.
Pass/Fail	Pass

Table 5. 2 Unit Test Case for Login

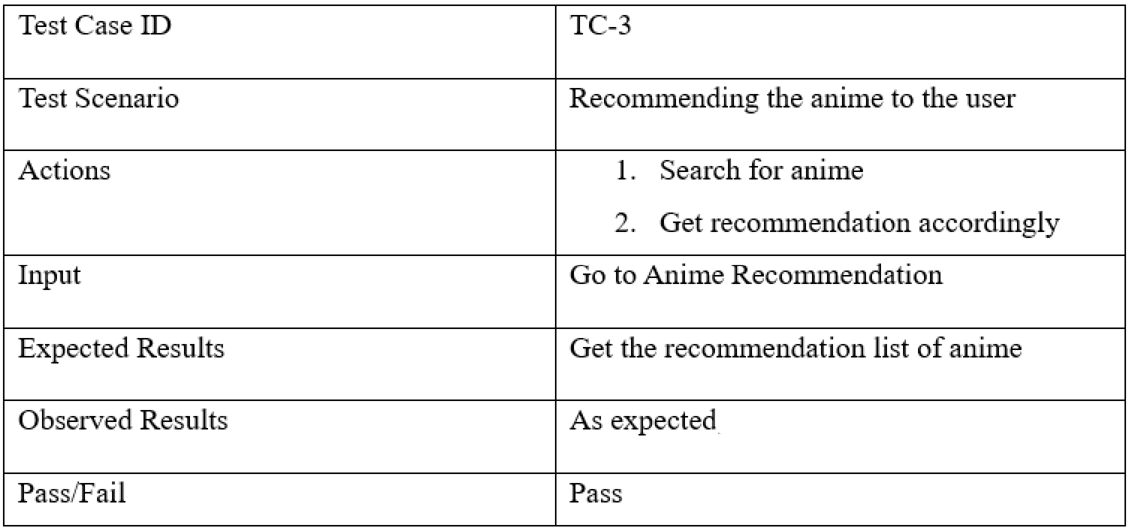


Test Case ID	TC-2
Test Scenario	Login for the existing user
Actions	1. Open the login page 2. Input username and password 3. Click on “Login” button
Input	Username: userabc Password: user123
Expected Results	Redirected to anime recommendation page
Observed Results	As expected.
Pass/Fail	Pass

5.2.2 Test Cases for system testing

System testing is a level of software testing where a complete and integrated software system is tested as a whole. Unlike unit testing, which focuses on testing individual components or units in isolation, system testing evaluates the behavior and performance of the entire system in a real or simulated environment.

Table 5. 3 System Test Case

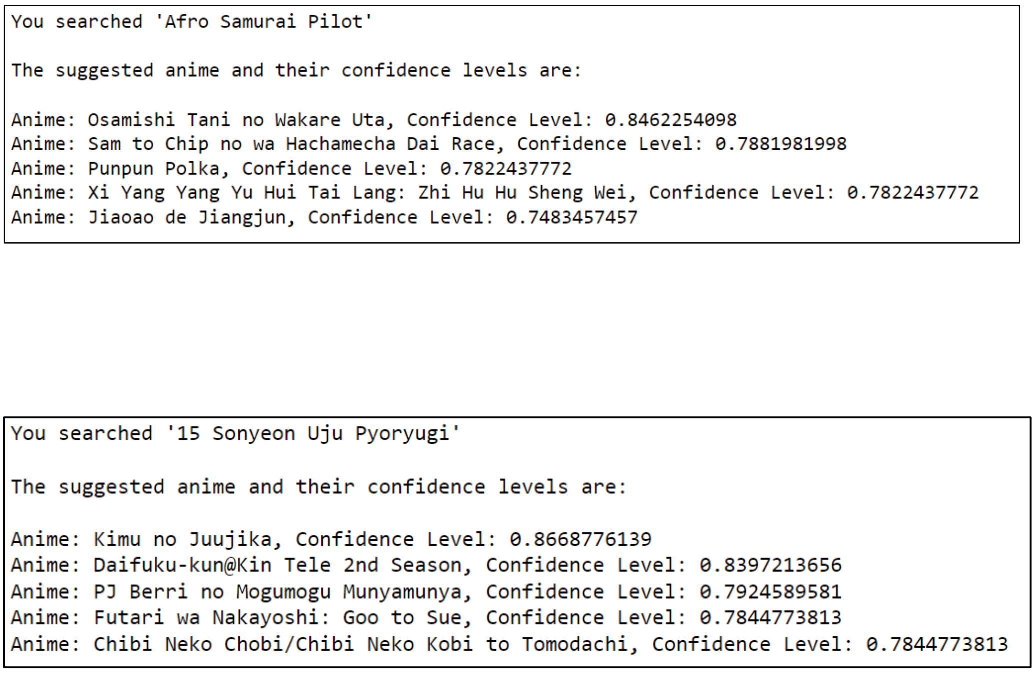


Test Case ID	TC-3
Test Scenario	Recommending the anime to the user
Actions	1. Search for anime 2. Get recommendation accordingly
Input	Go to Anime Recommendation
Expected Results	Get the recommendation list of anime
Observed Results	As expected
Pass/Fail	Pass

5.3 Result Analysis

Result analysis is the phase where the effectiveness of a trained model is assessed to evaluate its capability in making precise predictions. It's critical in determining the model's ability to generalize with unseen data and whether it aligns with anticipated outcomes.

To assess the accuracy and efficiency of the model, an evaluation of its confidence level was conducted. The confidence level, a key metric in determining the reliability of the model's recommendations, was meticulously analyzed. Through comprehensive examination, it was determined that the confidence level ranged between 79% and 81%. This range serves as a crucial indicator of the model's reliability and effectiveness in providing accurate anime recommendations. By meticulously scrutinizing the confidence level, the model ensures robustness and enhances user trust in its suggested recommendations.



```
You searched 'Afro Samurai Pilot'

The suggested anime and their confidence levels are:

Anime: Osamishi Tani no Wakare Uta, Confidence Level: 0.8462254098
Anime: Sam to Chip no wa Hachamecha Dai Race, Confidence Level: 0.7881981998
Anime: Punpun Polka, Confidence Level: 0.7822437772
Anime: Xi Yang Yang Yu Hui Tai Lang: Zhi Hu Hu Sheng Wei, Confidence Level: 0.7822437772
Anime: Jiaoao de Jiangjun, Confidence Level: 0.7483457457
```

```
You searched '15 Sonyeon Uju Pyoryugi'

The suggested anime and their confidence levels are:

Anime: Kimu no Juujika, Confidence Level: 0.8668776139
Anime: Daifuku-kun@Kin Tele 2nd Season, Confidence Level: 0.8397213656
Anime: PJ Berri no Mogumogu Munyamunya, Confidence Level: 0.7924589581
Anime: Futari wa Nakayoshi: Goo to Sue, Confidence Level: 0.7844773813
Anime: Chibi Neko Chobi/Chibi Neko Kobi to Tomodachi, Confidence Level: 0.7844773813
```

Figure 5.3 Confidence Level of the Recommended Anime

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

6.1 Conclusion

In summary, the anime recommendation system employs user preferences and anime ratings through the KNN-based Collaborative Filtering algorithm to generate a curated list of anime shows along with their corresponding scores. It utilizes the Django web framework for user registration and login, and HTML/CSS for the user interface, including input fields. Upon entering valid login credentials and providing input, the system generates a list of recommendations based on the user's anime ratings. Notably, it not only presents the recommendations but also displays the ratings of the suggested anime, enhancing user decision-making. This system represents a personalized approach to anime recommendations, offering valuable insights into its effectiveness and areas for improvement.

6.2 Future Recommendation

- The system should respond promptly to user queries and provide recommendations quickly to enhance the user experience.
- Design the system to handle a growing user base and an expanding library of anime titles without compromising performance.
- Ensure the reliability of the recommendation algorithm, minimizing errors and inaccuracies in suggested anime titles

REFERENCES

[1] Matija Ferjan, " headphonesaddict: 20+ Anime Statistics & Facts," 2024. Available: https://headphonesaddict.com/anime-statistics/

[2] Hernan Valdivieso, "Kaggle," 2020. [Online]. Available: https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020?resource=download.

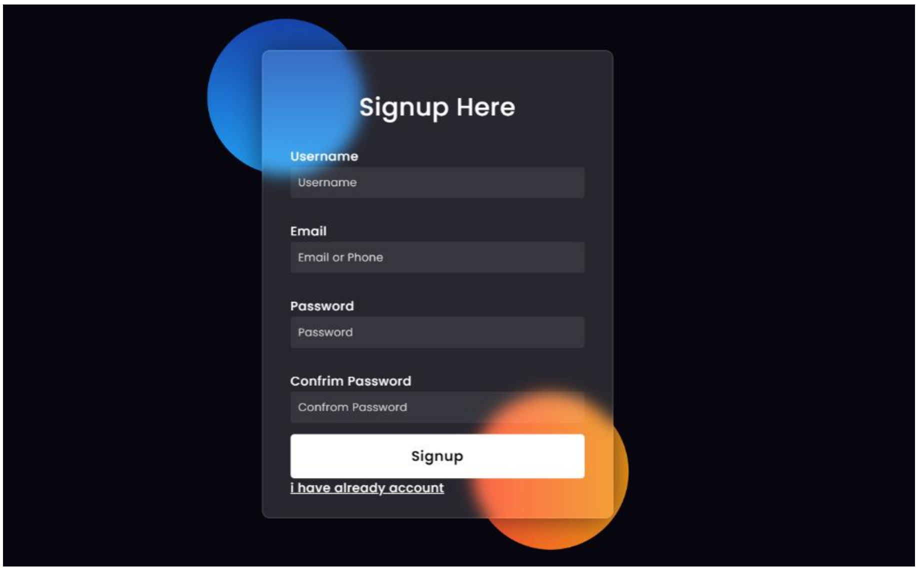
[3] A S Girsang, B Al Faruq, H R Herlianto, and S Simbolon, “Collaborative Recommendation System in Users of Anime Films,”, Jakarta, Indonesia, 2020.

[4] Nuurshadieq, Agung Toto Wibowo, “Leveraging Side Information to Anime Recommender System using Deep learning,” School of Computing Telkom University Bandung, Indonesia, 2020.

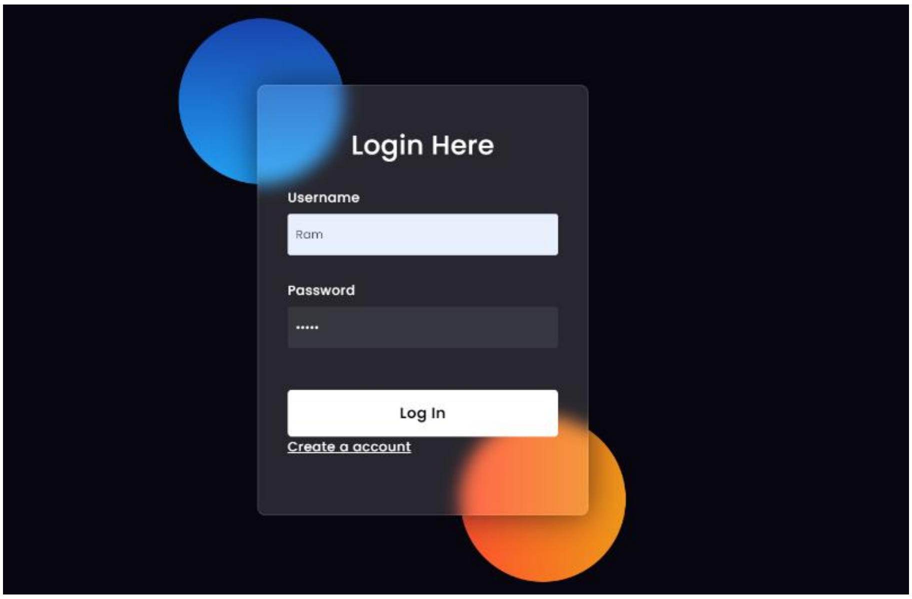
[5] Shardul Rudraksha1, Vipul Munot, Satyadev Mishra, Swapnil Misal, Yogita Kandhare, Abhilasha Kulkarni, “Hybrid Approach to Build Anime Recommendation System,”, Karvenagar, Pune, India, 2019.

APPENDICES

User Registration Page



User Login Page



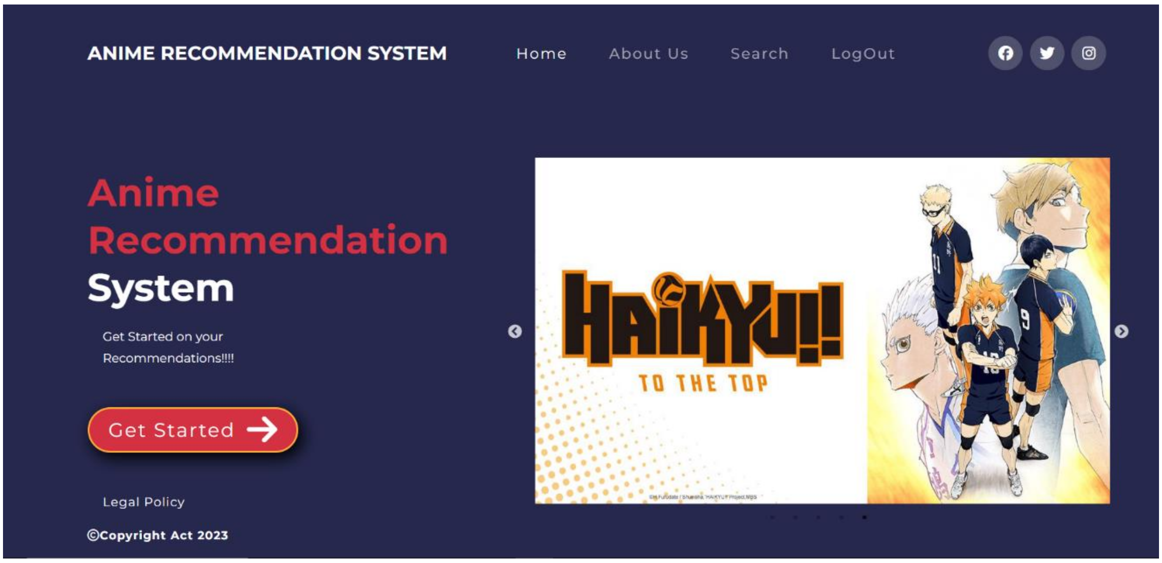
Username

Ram

Password

•••••

Home Page:

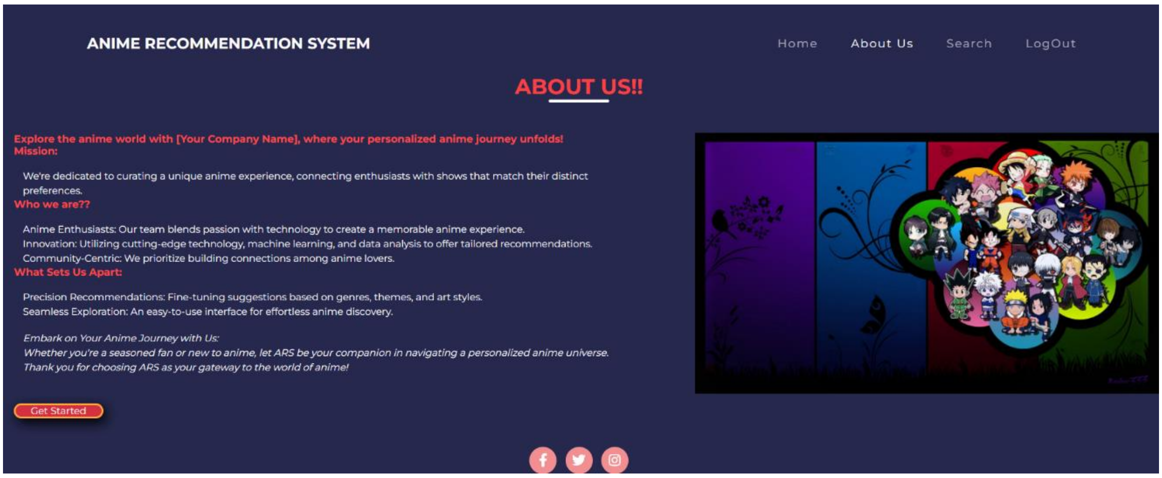


Anime Recommendation System

Get Started on your Recommendations!!!!

©Copyright Act 2023

About us Page:



ABOUT US!!

Explore the anime world with [Your Company Name], where your personalized anime journey unfolds!
Mission:

We're dedicated to curating a unique anime experience, connecting enthusiasts with shows that match their distinct preferences.

Who we are??

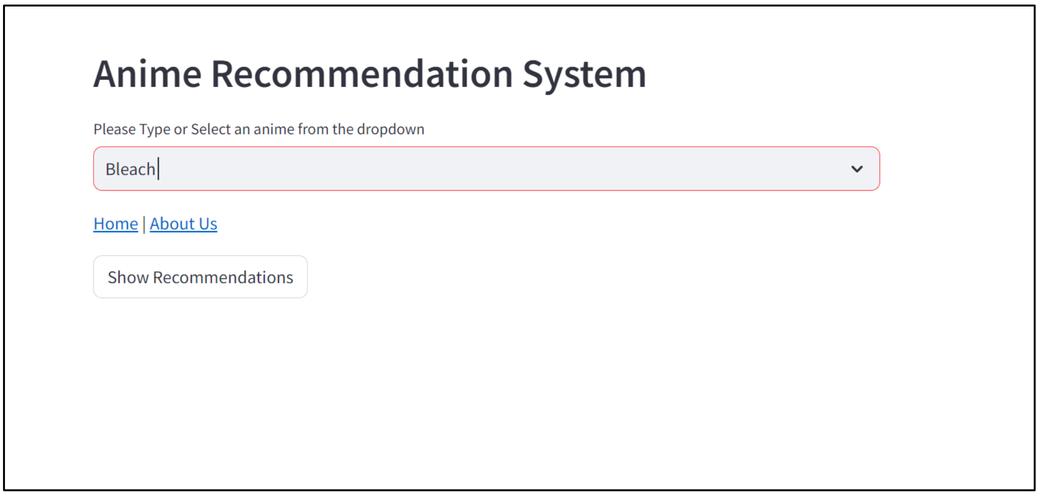
Anime Enthusiasts: Our team blends passion with technology to create a memorable anime experience.
Innovation: Utilizing cutting-edge technology, machine learning, and data analysis to offer tailored recommendations.
Community-Centric: We prioritize building connections among anime lovers.

What Sets Us Apart:

Precision Recommendations: Fine-tuning suggestions based on genres, themes, and art styles.
Seamless Exploration: An easy-to-use interface for effortless anime discovery.

Embark on Your Anime Journey with Us:
Whether you're a seasoned fan or new to anime, let ARS be your companion in navigating a personalized anime universe.
Thank you for choosing ARS as your gateway to the world of anime!

Search Interface:

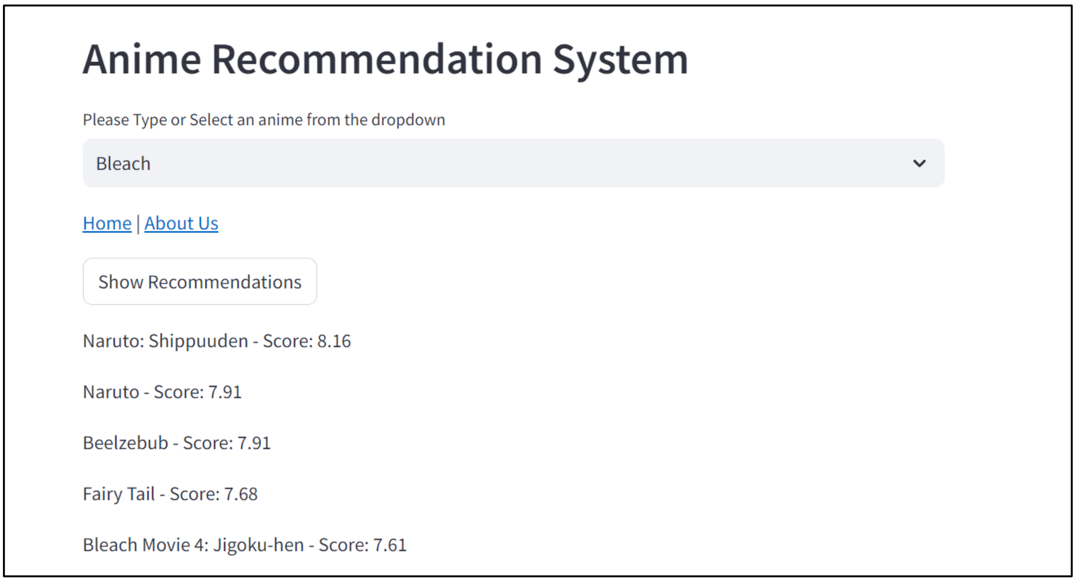


Anime Recommendation System

Please Type or Select an anime from the dropdown

Bleach

Recommendations Generated:



Anime Recommendation System

Please Type or Select an anime from the dropdown

Bleach

Naruto: Shippuuden - Score: 8.16

Naruto - Score: 7.91

Beelzebub - Score: 7.91

Fairy Tail - Score: 7.68

Bleach Movie 4: Jigoku-hen - Score: 7.61