MoviX App
(Movie Recommendation application)

A PROJECT REPORT
For
Major Project (KCA353)
Session (2023-24)

Submitted by
Akarsh Shekhar Singh
2200290140017

Deepanshi Sharma
2200290140052

Harsh Kain
2200290140077

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Shruti Mam
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(MAY 2024)

DECLARATION

I hereby declare that the work presented in report entitled “MoviX App” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Akarsh Shekhar Singh

Roll No.: 2200290140017

Name: Deepanshi Sharma

Roll No.: 2200290140052

Name: Hrash Kain

Roll No.: 2200290140077

(Candidate Signature)

CERTIFICATE

Certified that **Akarsh Shekhar Singh 2200290140017, Deepanshi Sharma 2200290140052, Harsh Kain 2200290140077** has/ have carried out the project work having "**MoviX App**" (**Major Project-KCA353**) for **Master of Computer Application** from Dr A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:24 /05/2024

Akarsh Shekhar Singh (2200290140017)
Deepanshi Sharma (2200290140052)
Harsh Kain (2200290140077)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 24/05/2024

Ms. Shruti Mam
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr Arun Kumar Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

"MoviX App" is basically a movie recommendation application that predicts the user's choices and then suggest them the more accurate results by using ChatGpt integration to the application. We have a variety of varied applications of this recommendation systems in which we can be used over the years and now used in various online platforms the basic content of all these platforms are basically different types of movies such as action thriller romantic or maybe your eCommerce website any platform of social media having a professional website such as LinkedIn . For example, when we use Instagram we can see the previous stories that on the feed of the people we follow so here we can see that the Instagram can monitor our interaction with the various people are our past activities and then it just suggest kind of other related stories of some other accounts that have done some same kind of activity previously or currently. Quite a few times is recommender system also keep improving the activities of a bunch of users based on the activities they have scroll through you attempted. For example, on Flipkart when we buy some laptop or any mobile phone then it simply suggests mobile cover tempered glass for mobile or buy USB type C adaptor or type A adaptor for the laptop also. Safed enhancements in the recommender systems users get good recommendation all the time and it keeps on improving as we move forward in the 21st century and they make almost accurate solutions. In case of clash of any e App Music any music platform or any educational then use a simply deny using the app in addition to this the companies have to focus on their recommendation system which is more Complex than it seems. Every user has different preferences and different choices based upon their different type of activities sometime mood also so in case of music's while playing, travelling, running aur after having some fight in relationships etc.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, Ms. Shruti Mam (Assistant Professor) for their guidance, help and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions have guided me a lot in completing this project successfully.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Akarsh Shekhar Singh (2200290140017)

Deepanshi Sharma (2200290140052)

Harsh Kain (2200290140077)

TABLE OF CONTENTS

	Page No.
Declaration	ii
Certificate	iii
Abstract	iv
Acknowledgements	v
Table of contents	vi
List of tables	viii
List of figures	ix
Chapter 1: Introduction	10-12
1.1 Overview	10
1.2 Functionality	10
1.3 Objectives	11
1.4 Scope	12
1.5 Hardware / Software used in project	13
Chapter 2: Feasibility Study	14-17
2.1 Technical feasibility	14
2.2 Operational feasibility	15
2.3 Economic feasibility	15
2.4 Behavioral feasibility	17
Chapter 3: System requirements	18-20
3.1 Functional requirements	18
3.2 Non-functional requirements	19
3.3 Performance Requirements	18
Chapter 4: System Analysis	21-23
4.1 Requirement Analysis	21
4.2 Stakeholder Analysis	21
4.3 Use Case Analysis	22
4.4 Data Analysis	22

4.5 Process Analysis 22
4.6 Technical Analysis 23
4.7 Risk Analysis 23
4.8 Feasibility Analysis 23

Chapter 5: System Design 25-38

5.1 Primary Design phase 25
5.2 Secondary design phase 25
5.3 Use case diagram 27
5.4 ER Diagram 29-32
5.5 Data Flow Diagram 33-38
5.5.1 Level-0 DFD 33
5.5.2 Level-1 DFD 33
5.5.3 Level-2 DFD 35
5.6 Design Goals 37

Chapter 6: Testing 39-42

6.1 Testing 39
6.2 Testing result 42
6.3 The Steps in Software Testing 42

Chapter 7: Conclusion 50

7.1 Future enhancement 50

Chapter 8: References 51

LIST OF TABLES

Table No.	Name of Table	Page
1.1	Hardware Requirement	11
1.2	Software Requirement	12

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
5.1	Use case diagram	20
5.2	ER- diagram	21
5.3	Level 0 DFD Diagram	22
5.4	Level 1 DFD Diagram	23
5.5	Level 2 DFD Diagram	24
7.1	Home Page	33
7.2	Login Page	34
7.3	Sign up Page	35
7.4	Trending Movie Page	36
7.5	Movie Detail Page	37
7.6	Movie Page	38
7.7	TV Show Page	39

Chapter 1

INTRODUCTION

1.1 Overview

MoviX is an innovative movie recommendation application designed to enhance users' entertainment experience by offering personalized movie suggestions tailored to their preferences. With an intuitive user interface and powerful recommendation algorithms, MoviX aims to revolutionize the way users discover and enjoy movies.

MoviX represents the next generation of movie recommendation applications, leveraging advanced technology to deliver personalized and engaging movie suggestions to users. By combining powerful recommendation algorithms with social integration and intuitive features, MoviX aims to redefine the way people discover and enjoy movies in the digital age.

1.2 FUNCTIONALITY

- **Personalized Recommendations:** MoviX utilizes advanced machine learning algorithms to analyse users' viewing history, ratings, and preferences to generate personalized movie recommendations.

- **User Profiles:** Users can create profiles where they can input their movie preferences, ratings, and watch history. This allows MoviX to continuously refine its recommendations based on individual user tastes.

- **Search and Discovery:** MoviX offers robust search functionality, allowing users to explore a vast database of movies based on genres, actors, directors, and other criteria. Users can also discover new movies through curated lists and trending recommendations.

- **Watchlist and Favourite's:** Users can save movies to their watchlist for future viewing and mark their favourites movies for easy access.

- **Social Integration:** MoviX allows users to connect with friends, share movie recommendations, and see what movies their friends are watching and enjoying.

- **Reviews and Ratings:** Users can read reviews and ratings from other users to help them make informed decisions about which movies to watch.

- **Notifications:** MoviX sends notifications to users about new movie releases, personalized recommendations, and updates from friends.

- **Cross-Platform Compatibility:** MoviX is available across multiple platforms, including iOS, Android, and web browsers, ensuring seamless access for users across different devices.

1.3 OBJECTIVES

- Develop a user-friendly movie recommendation application.
- Implement advanced machine learning algorithms for personalized movie suggestions.
- Enable users to create profiles and input their movie preferences.
- Provide robust search and discovery features for exploring a vast movie database.
- Facilitate social interaction by allowing users to connect with friends and share recommendations.
- Incorporate features for saving favourites movies and creating watchlists.
- Deliver notifications for new movie releases and personalized recommendations.
- Ensure cross-platform compatibility for seamless access across multiple devices.

1.4 SCOPE

The scope of MoviX encompasses the development of a comprehensive movie recommendation application that employs advanced machine learning algorithms to provide personalized

suggestions based on user preferences. It includes features such as user profiles, robust search and discovery functionalities, social integration for sharing recommendations, and the ability to save favourite movies and create watchlists. The application aims to enhance users' movie discovery experience by delivering tailored recommendations and facilitating social interaction. Additionally, MoviX will ensure cross-platform compatibility to enable seamless access across various devices, ultimately aiming to redefine how users discover and enjoy movies in the digital era.

1.5 Hardware / Software Requirement

Hardware Requirement :

S. N.	Description
1	PC with 5 GB or more Hard disk.
2	PC with 2 GB RAM.
3	PC with core i3 or above processor.

Software Requirements:

S. N.	Description	Type
1	Operating System	Windows 10 or 11 or Ubuntu 18.04 or above
2	Language	JavaScript
3	Front End	React 17
4	IDE	Google Collab, VS Code
5	Browser	Chrome, Firefox, Edge

CHAPTER 2

FEASIBILITY STUDY

After doing the project MoviX App (movie recommendation application), study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time. The Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily made based on the future upcoming requirements.

2.1 Economic Feasibility

This is a very important aspect to be considered while developing a project. We decided. the technology based on minimum possible costfactor.

- All hardware and software costs must be borne by the organization.
- Overall, we have estimated that the benefits the organization is goingto receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

2.2 Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS) and checked if everything was possible using different type of frontend and backend platforms.

2.3 Operation Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, proper training has been conducted to let the essenceof the system to the users so that they feel comfortable with the new system.

2.4 Behavioural Feasibility

The behavioural feasibility of our proposed e-commerce project is highly promising, as it aligns seamlessly with the contemporary attitudes and preferences of potential users. Market research indicates a widespread acceptance of online shopping, and our platform caters to this trend by offering a user-friendly interface, responsive design for mobile devices, and robust security measures. The project addresses user concerns through clear communication of security protocols and trust-building elements such as secure payment gateways. Additionally, our commitment to providing excellent customer support and feedback mechanisms ensures a positive user experience, building trust and credibility. With a focus on cultural adaptability, intuitive on boarding processes, and integration with social media platforms, the project aims to not only meet but exceed user expectations, fostering a strong and loyal customer base in the competitive e-commerce landscape.

2.5 Risk Analysis:

- Identification of potential risks (e.g., competition, technical issues, legal disputes).
- Assessment of risk severity and likelihood.
- Mitigation strategies.

Certainly! When conducting a risk analysis for a movie recommendation website, it's important to consider various potential risks that could impact the project's feasibility. Here are some key risks to consider: Competition: Risk: Competition from established movie recommendation platforms could make it challenging to attract users. Mitigation: Conduct thorough market research to identify unique features and target niche audiences. Focus on offering superior user experience and personalized recommendations.

2.6 Legal and Ethical Considerations:

Identify any legal requirements (e.g., copyright laws, data privacy regulations) and ensure compliance. Consider ethical implications, such as data security and user privacy, and develop appropriate measures to address them.

2.7 User Adoption and Retention:

Risk: Difficulty in attracting and retaining users due to competition or lack of engagement features. Mitigation: Invest in marketing and user acquisition strategies, offer compelling incentives for users to sign up and remain active (e.g., personalized recommendations, exclusive content), and continuously optimize the user experience based on feedback and analytics.

2.7 Conclusion:

- Summary of findings.
- Recommendation on the feasibility of the project.
- Next steps and further research if needed.

By conducting a thorough feasibility study, you can make informed decisions about the viability and potential success of the movie recommendation website project.

2.8 Appendices:

- Additional data, charts, or references used in the study.

Remember to tailor your feasibility study to the specific requirements and objectives of your movie recommendation website project. Conducting thorough research and analysis will help you make informed decisions and increase the likelihood of success.

CHAPTER 3

SYSTEM REQUIREMENTS

The MoviX app requires a modern smartphone or tablet running iOS 11 or later, or Android 8.0 or later. The app relies on a stable internet connection for real-time movie recommendations, updates, and social sharing features. Users must have sufficient storage space on their device to download movies for offline viewing. Additionally, the app's performance is optimized for devices with at least 2GB of RAM and a quad-core processor. The app is designed to be user-friendly and accessible, adhering to accessibility guidelines for users with disabilities.

3.1 Functional Requirements:

- **User Registration:** Users should be able to create an account on the MoviX app.
- **User Profile Creation:** Users should be able to create and update their profiles, including movie preferences and watch history.
- **Movie Recommendation:** The app should provide personalized movie recommendations based on user preferences, ratings, and watch history.
- **Movie Search:** Users should be able to search for movies by title, genre, actor, director, and other criteria.
- **Watchlist:** Users should be able to add movies to a watchlist for future viewing.
- **Rating and Reviews:** Users should be able to rate movies and leave reviews.
- **Social Sharing:** Users should be able to share movie recommendations and reviews on social media platforms.
- **Notifications:** The app should send notifications for new movie releases, personalized recommendations, and updates.
- **Offline Viewing:** The app should allow users to download movies for offline viewing.
- **Multiple Devices:** Users should be able to access their account and preferences across multiple devices.

3.2 Non-Functional Requirements:

- **Performance:** The app should respond quickly to user interactions and provide smooth navigation.
- **Security:** The app should ensure the security of user data, including personal information and watch history.
- **Scalability:** The app should be able to handle many users and movie data.
- **Compatibility:** The app should be compatible with various devices and operating systems.
- **User Interface:** The app should have an intuitive and user-friendly interface.
- **Reliability:** The app should be always reliable and available for use.
- **Data Backup:** The app should regularly backup user data to prevent loss.
- **Compliance:** The app should comply with relevant regulations and standards for data privacy and security.
- **Accessibility:** The app should be accessible to users with disabilities, following accessibility guidelines.
- **Performance Tracking:** The app should track user interactions and preferences to improve movie recommendations over time.

3.3 PERFORMANCE REQUIREMENT

- **Response Time:** The MoviX app should respond to user interactions (e.g., movie searches, recommendations) within 1 second.
- **Loading Time:** The app should load movie lists and details within 3 seconds.
- **Scalability:** The app should be able to handle a minimum of 10,000 simultaneous users without significant performance degradation.
- **Database Performance:** Database queries should execute within 100 milliseconds to ensure smooth operation.
- **Offline Mode:** Downloading movies for offline viewing should not take more than 5 minutes per movie.
- **Notification Delivery:** Notifications should be delivered to users within 10 seconds of being triggered.
- **Search Speed**: Search results should be displayed to the user within 2 seconds of initiating the search.
- **Concurrency Handling**: The app should handle at least 100 concurrent user sessions without crashing or experiencing significant slowdowns.
- **Update Frequency:** Movie recommendations should be updated at least once a day to reflect user preferences and new releases.
- **Compatibility:** The app should be compatible with a variety of devices and screen sizes, ensuring consistent performance across different platforms.

CHAPTER 4

SYSTEM ANALYSIS

The system analysis phase for a movie recommendation website involves a detailed examination of the requirements, functionalities, and architecture needed to develop an effective and user-friendly platform. Here's a breakdown of the key components of this phase:

4.1 Requirement Analysis

4.1.1 Functional Requirements:

- **User Authentication:** Users should be able to sign up, log in, log out, and recover passwords.
- **Movie Search:** Implement a robust search feature with filters and sorting options.
- **Recommendation Engine:** Develop an algorithm that suggests movies based on user **preferences, viewing history, and ratings.**
- **Movie Details:** Provide comprehensive information about each movie, including synopsis, cast, director, genre, release date, and user reviews.
- **Rating and Reviews:** Allow users to rate movies and write reviews.
- **Watchlist:** Enable users to create, manage, and share watchlists.
- **User Profiles:** users customize their profiles and set preferences.

4.1.2 Non-Functional Requirements:

- **Performance:** Ensure fast load times and responsive interactions.
- **Scalability:** Design the system to handle a growing number of users and data.
- **Security:** Protect user data with encryption and secure authentication mechanisms.
- **Usability:** Create an intuitive and easy-to-navigate interface.
- **Compatibility:** Ensure the website works across various devices and browsers.

4.2 Stakeholder Analysis

- **Users:** Identify different user personas such as casual viewers, movie enthusiasts, and critics. Understand their needs and preferences.
- **Administrators:** Define the roles and responsibilities of website administrators, including content management, user support, and system maintenance.
- **Developers:** Outline the technical requirements and tools needed for development.
- **Partners:** Identify potential partners for data sources (e.g., movie databases) and third-party integrations (e.g., social media).

4.3 Use Case Analysis

- Develop detailed use cases for each functional requirement. For example:

Search Movie Use Case:

- User enters a search query.
- System displays a list of movies matching the query.
- User applies filters to refine the search results.

Rate Movie Use Case:

- User selects a movie and submits a rating.
- System updates the movie's average rating and adjusts recommendations accordingly.

4.4 Data Analysis

- **Data Sources:** Identify reliable sources for movie data (e.g., IMDb).
- **Data Storage:** Design the database schema to store user data, movie details, ratings, and reviews.
- **Data Flow:** Map out how data will flow through the system from user input to processing by the recommendation engine and back to the user interface.

4.5 Process Analysis

- **Workflow Diagrams:** Create diagrams to illustrate the workflow for key processes like user registration, movie search, recommendation generation, and review submission.
- **Sequence Diagrams:** Develop sequence diagrams to detail the interactions between different system components for various use cases.

4.6 Technical Analysis

- **Technology Stack:** Choose the appropriate technologies for frontend (e.g., React, Angular), backend (e.g., Node.js), database (e.g., MySQL, MongoDB), and recommendation engine (e.g., collaborative filtering, content-based filtering).
- **Architecture:** Design the system architecture, including client-server interactions, API design, and data storage solutions.
- **Integration:** Plan for integration with third-party services and APIs for additional functionalities like social sharing and external movie databases.

4.7 Risk Analysis

- **Identify Risks:** Determine potential risks such as data breaches, system downtime, and inaccurate recommendations.
- **Mitigation Strategies:** Develop strategies to mitigate these risks, including robust security measures, regular system backups, and continuous improvement of the recommendation algorithm.

4.8 Feasibility Analysis

- **Technical Feasibility:** Assess whether the chosen technology stack can support the system requirements.
- **Operational Feasibility:** Ensure the system can be maintained and operated efficiently.
- **Economic Feasibility:** Evaluate the cost of development, operation, and maintenance against the expected benefits and revenue.

CHAPTER 5

SYSTEM DESIGN

Designing a movie recommendation website involves structuring the system to efficiently handle user interactions, content management, and recommendation algorithms. Here's a basic system design for such a website:

User Interface (UI):

- The UI should be intuitive, visually appealing, and responsive across different devices (desktop, mobile, tablet).
- Include features for user registration, login, profile management, and browsing/searching for movies.
- Implement personalized recommendations based on user preferences, viewing history, ratings, and other relevant data.

Backend Services:

User Authentication:

- Implement secure authentication mechanisms such as OAuth or JWT tokens for user login and registration.

Database Management:

- Use a relational or NoSQL database to store user profiles, movie metadata, ratings, and other relevant data.

Content Management:

- Develop functionalities for adding, editing, and categorizing movie content. Include features for content moderation and approval.

Recommendation Engine:

- Build algorithms to generate personalized movie recommendations based on user behaviour, collaborative filtering, content-based filtering, or hybrid approaches.

Integration with External APIs:

- Integrate with external APIs for accessing movie metadata, trailers, reviews, and streaming platforms (if applicable).

Infrastructure:

- Web Server: Deploy a web server (e.g., Apache, Nginx) to handle HTTP requests and serve static and dynamic content.

Application Server:

- Use a server-side framework (e.g., Django, Flask for Python; Express.js for Node.js) to handle application logic and interact with the database.

Scalability:

- Design the system to scale horizontally by adding more servers or using cloud services (e.g., AWS, Google Cloud) to handle increased traffic and workload.

APIs and Microservices:

- Design RESTful APIs for communication between frontend and backend components.
- Implement microservices architecture for modularization and scalability. Separate services for user management, content management, recommendation engine, etc.

Caching and Performance Optimization:

- Implement caching mechanisms (e.g., Memcached, Redis) to store frequently accessed data and improve system performance.

- Optimize database queries, use indexing, and employ other performance tuning techniques to reduce latency and improve response times.

Security:

- Implement security best practices such as input validation, output encoding, and protection against common vulnerabilities (e.g., XSS, CSRF).
- Use HTTPS for secure communication, encrypt sensitive data at rest and in transit, and implement proper access controls and authorization mechanism!

Monitoring and Analytics:

- Set up monitoring tools (e.g., Prometheus, Grafana) to track system performance, detect anomalies, and troubleshoot issues.
- Use analytics tools (e.g., Google Analytics, mix panel) to gather user insights, track user engagement, and evaluate the effectiveness of recommendation algorithms.

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following in the following two steps:

5.1 Primary Design Phase:

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

5.2 Secondary Design Phase:

In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

1. Design various blocks for overall system processes.
2. Design smaller, compact, and workable modules in each block.
3. Design various database structures.
4. Specify details of programs to achieve desired functionality.
5. Design the form of inputs, and outputs of the system.
6. Perform documentation of the design.

5.3 Use Case Diagram

A user case diagram for a movie recommendation project includes users searching for movies, viewing recommendations, rating movies, creating profiles, and saving favourites. The system processes user data, updates recommendations, and manages movie databases to improve accuracy and user experience.

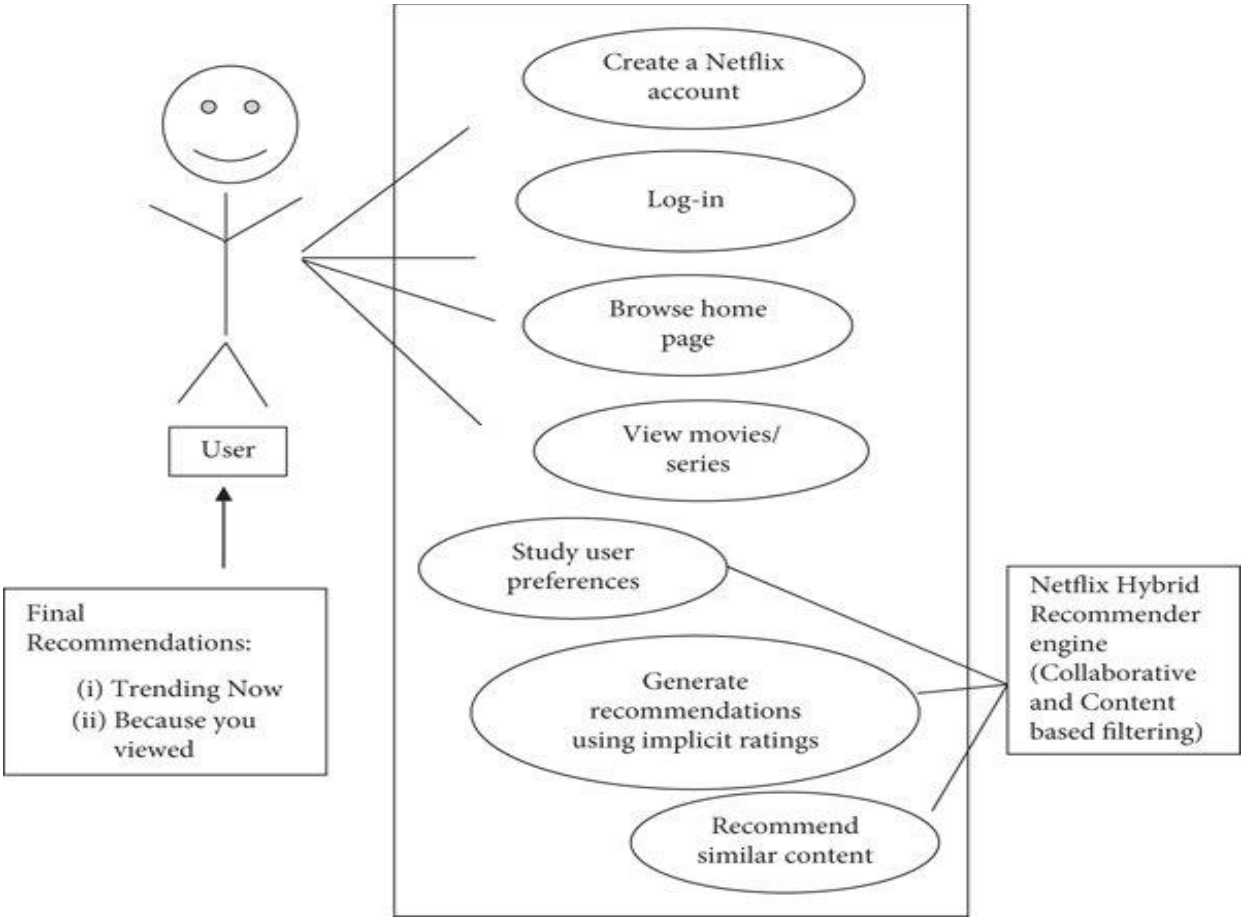

Figure 1: Use Case Diagram

5.4 ER Diagram

An Entity-Relationship (ER) diagram is a visual representation of the relationships among entities in a database. It typically includes entities (such as people, objects, or concepts), attributes that describe these entities, and the relationships between entities**.**

Here's how you might design an E-R diagram for such a website:

Entities:

- **User:**
  Represents registered users of the website. Attributes may include UserID (primary key), Username, Password (hashed), Email, and other user profile information.
- **Movie:**
  Represents individual movies available on the platform. Attributes may include MovieID (primary key), Title, ReleaseYear, Genre, Director, Actors, Description, Ratings, etc.
- **Genre:**
  Represents different genres that movies belong to. Attributes may include GenreID (primary key) and GenreName.
- **Actor:**
  Represents actors involved in movies. Attributes may include ActorID (primary key), FirstName, LastName, etc.
- **Director:**
  Represents directors of movies. Attributes may include DirectorID (primary key), FirstName, LastName, etc.

Rating:

Represents user ratings for movies. Attributes may include RatingID (primary key), UserID (foreign key), MovieID (foreign key), RatingValue, Timestamp, etc.

Recommendation:

Represents recommendations generated for users. Attributes may include RecommendationID (primary key), UserID (foreign key), MovieID (foreign key), Score, Timestamp, etc.

Relationships:

- **User-Movie (Rated):** A user can rate multiple movies, and a movie can receive ratings from multiple users. This is a many-to-many relationship.
- **User-Recommendation:** A user can receive multiple recommendations, and each recommendation is generated for a single user. This is a one-to-many relationship.

Movie-Genre:

A movie can belong to multiple genres, and each genre can have multiple movies. This is a many-to-many relationship.

- **Movie-Actor:** A movie can have multiple actors, and an actor can participate in multiple movies. This is a many-to-many relationship.
- **Movie-Director:** A movie can have only one director, but a director can direct multiple movies. This is a one-to-many relationship.

Attributes:

- Each entity has its own set of attributes, as described above. Attributes are the properties or characteristics of an entity.

Keys:

- Each entity has a primary key (underlined) which uniquely identifies each record in the table.
- Foreign keys are used to establish relationships between entities. They are attributes in one entity that refer to the primary key of another entity.

Cardinality and Modality:

- Cardinality describes the maximum number of times an instance in one entity can be associated with instances in another entity.
- Modality describes the minimum number of times an instance in one entity must be associated with instances in another entity.

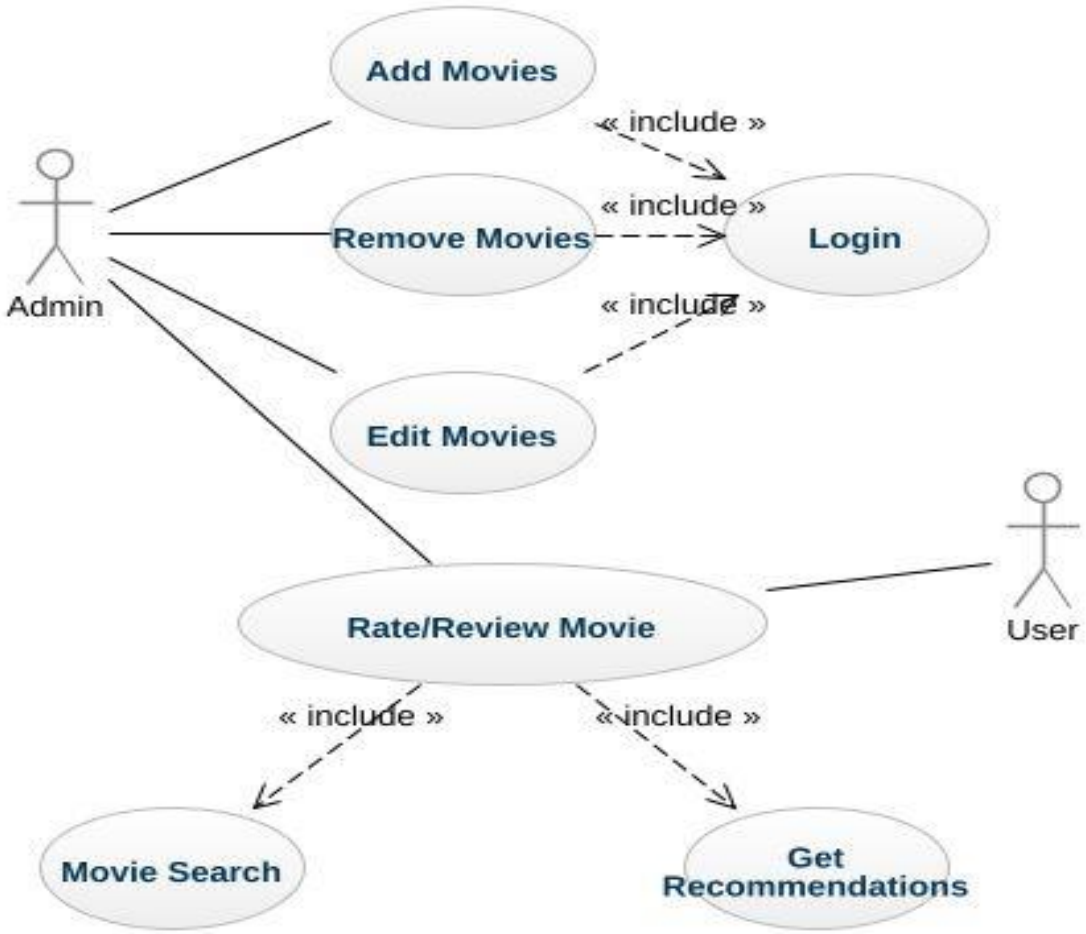

Figure 2: ER Diagram

5.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It shows how data moves between processes, data stores, and external entities. Let's break down how a DFD could be applied to a movie recommendation website:

External Entities:

Users:

- Users interact with the website to perform actions such as registering, logging in, searching for movies, rating movies, and receiving recommendations.

Content Providers:

- External entities providing movie metadata, trailers, reviews, and other content that the website integrates.

Processes:

User Registration/Login:

- Processes for users to register for a new account or log in to an existing account.

Search and Browse:

- Processes for users to search for movies based on various criteria such as genre, release year, actor, director, etc.

Rating Movies:

- Processes for users to rate movies they've watched, providing feedback that will influence the recommendation engine.

Recommendation Engine:

- The core process responsible for generating personalized movie recommendations based on user preferences, ratings, and viewing history.

Content Management:

- Processes for adding, editing, and categorizing movie content, as well as moderating user-generated content such as reviews and comments.

Data Stores:

User Profiles:

- Stores user data such as usernames, passwords (hashed), email addresses, preferences, and ratings.

Movie Database:

- Stores metadata for movies including titles, genres, release years, actors, directors, ratings, and other relevant information.

User Ratings:

- Stores user ratings for movies, linking users to the movies they've rated and the corresponding ratings.

Recommendations:

- Stores generated recommendations for users, linking users to recommended movies.

Content Cache:

- Stores frequently accessed movie metadata, trailers, and other content to improve performance.

Data Flow:

User Registration/Login Flow:

- Data flows from the user interface to the user profiles data store when users register or log in. Authentication tokens may flow back to the user interface upon successful login.

Search and Browse Flow:

- User queries flow from the user interface to the movie database, and search results flow back to the user interface for display.

Rating Movies Flow:

- User ratings flow from the user interface to the user ratings data store for storage and analysis by the recommendation engine.

Recommendation Generation Flow:

- User profiles, movie metadata, and user ratings flow to the recommendation engine, which processes this data to generate personalized recommendations. The generated recommendations flow back to the user interface for display.

Content Management Flow:

- New movie data and user-generated content flow from the user interface to the content management processes and data stores for moderation and storage.

In summary, a Data Flow Diagram for a movie recommendation website illustrates how users interact with the system to search for movies, rate them, and receive personalized recommendations, as well as how data flows between different processes and data stores to facilitate these interactions.

A Data Flow Diagram (DFD) illustrates how data flows through a system, showing the processes, data stores, and external entities that interact with the system. It consists of processes that transform data, data stores that hold data, data flows that represent the movement of data between processes and stores, and external entities that interact with the system. DFDs are useful for understanding system requirements and designing new systems or improving existing ones.

5.5.1 Zero Level DFD

A Zero Level Data Flow Diagram (DFD), also known as a Context Diagram, provides a high-level overview of a system. It depicts the system as a single process with its interactions with external entities such as users, other systems, or external databases. This diagram outlines the overall system boundary, main process, and data flows without detailing internal processes or data stores.

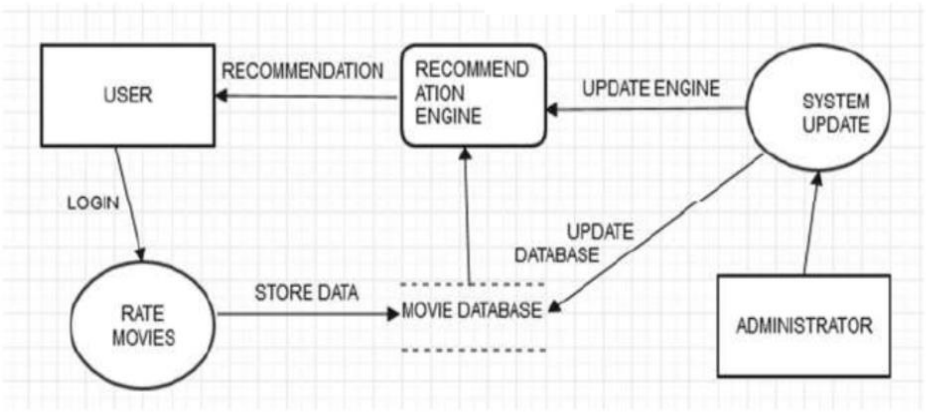

Figure 3: Zero Level DFD

5.5.2 First level DFD

A Level 1 Data Flow Diagram (DFD) breaks down the main process into sub-processes, showing detailed data flows between these processes, data stores, and external entities. It visualizes how data is processed and transferred within the system, providing a clearer understanding of its functionality and interactions.

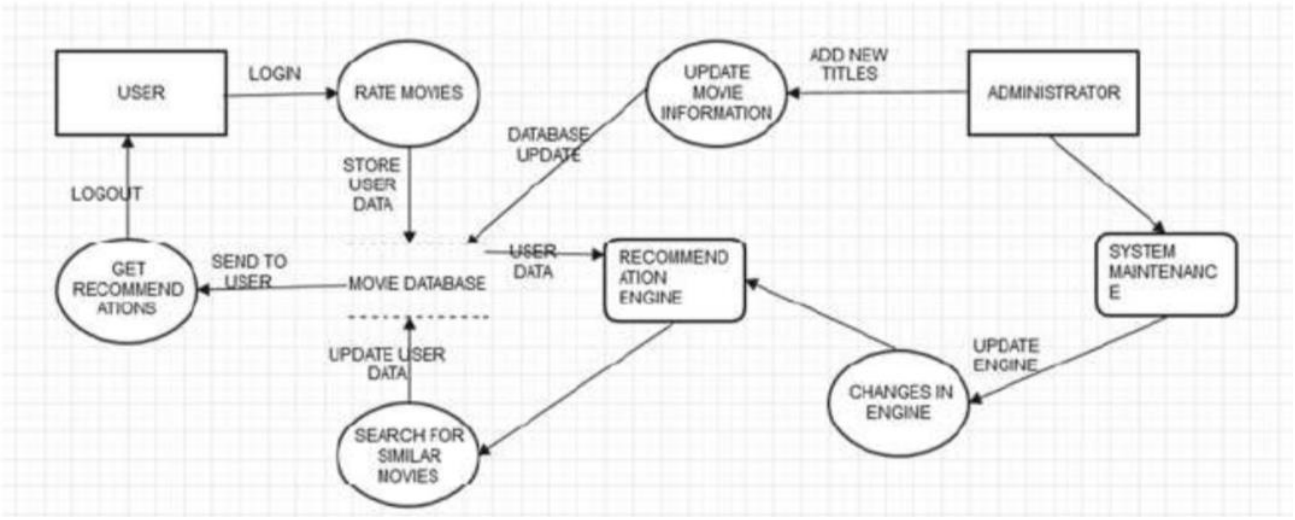

Figure 4: First Level DFD

5.5.3 Second level DFD

A Second Level Data Flow Diagram (DFD) provides even more detail by breaking down the processes identified in the Level 1 DFD into more granular sub-processes. This diagram shows specific data flows between these sub-processes, data stores, and external entities, offering a deeper understanding of how each part of the system operates and interacts within the broader context**.**

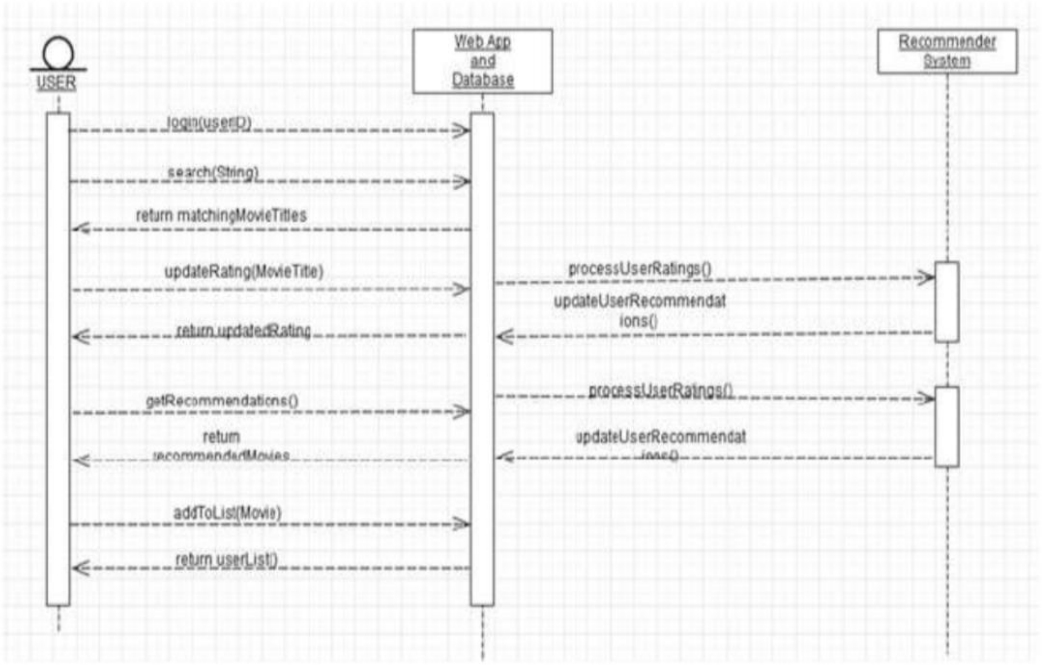

Figure 5: Second Level DFD

5.6 Design Goal

The primary design goal of a movie recommendation system website is to create an intuitive, personalized, and engaging user experience that keeps users returning to the platform.

1. **Personalized Recommendations:** The system should leverage advanced algorithms to analyse user preferences, viewing history, ratings, and behaviours to provide highly accurate and relevant movie suggestions. This personalization increases user satisfaction and engagement.
2. **User-Friendly Interface:** The website should have a clean, intuitive interface that makes it easy for users to search for movies, view recommendations, and navigate through various sections.
3. **Comprehensive Movie Information:** Each movie should have a detailed page with information such as synopsis, cast, director, genre, release date, and user reviews. This helps users make informed choices about what to watch.
4. **Interactive Features:** The website should include interactive elements such as user ratings, reviews, and the ability to create and share watchlists. Social features like commenting, sharing recommendations with friends, and following other users' activity can enhance community engagement.
5. **Performance and Scalability:** The platform must be designed to handle many users and data efficiently. Fast load times, seamless streaming integration, and reliable performance are crucial.
6. **Cross-Platform Compatibility:** Ensuring the website is responsive and works well on various devices, including desktops, tablets, and smartphones, is essential for accessibility and user convenient.

CHAPTER 6

Testing

6.1 Testing

A movie recommendation website testing module involves several key components to ensure the platform functions correctly and provides a seamless user experience. Here's a detailed outline of the testing module:

1. Functional Testing

- **User Authentication:** Test sign-up, login, logout, and password recovery processes.

Search Functionality: Verify the search feature works accurately, including filtering and sorting options.

- **Recommendation Algorithm:** Check the accuracy and relevance of movie recommendations based on user preferences and history.
- **Movie Details Page**: Ensure all information (synopsis, cast, ratings, reviews) displays correctly.
- **Rating and Review Submission:** Test the ability for users to rate and review movies and see their own and others' reviews.
- **Watchlist Management:** Verify users can add, remove, and manage movies in their watchlists.
- **Profile Management:** Check functionalities for updating user profiles, including personal information and preferences.

2. Usability Testing

- **Navigation:** Ensure the website's navigation is intuitive and user-friendly.

- **Interface Design:** Check the visual appeal, layout consistency, and readability of the website.
- **Mobile Compatibility:** Test the website's responsiveness and functionality on various devices (desktop, tablets, smartphones).

3. Performance Testing

- **Load Time:** Measure the website's load times under different conditions.
- **Scalability:** Test the website's performance under high user load to ensure it can scale effectively.
- **Stress Testing:** Determine the website's behaviours under extreme conditions to identify breaking points.

4. Security Testing

- **Data Protection:** Verify encryption of user data, especially sensitive information like passwords.
- **Vulnerability Scanning:** Check for common vulnerabilities such as SQL injection, XSS, and CSRF attacks.
- **Authorization:** Ensure proper access control for different user roles and data access.

5. Compatibility Testing

- **Cross-Browser Compatibility:** Test the website on different browsers (Chrome, Firefox, Safari, Edge) to ensure consistent behaviours.
- **Platform Compatibility:** Verify the website works correctly across various operating systems (Windows, macOS, Linux).

6. Regression Testing

- **New Features:** Ensure new features do not break existing functionality.
- **Bug Fixes:** Verify that bug fixes are effective and do not introduce new issues.

7. Integration Testing

- **Third-Party Services:** Check the integration with external APIs and services (e.g., social media sharing, streaming services).
- **Database Integration:** Test interactions between the website and the database for data retrieval and storage accuracy.

8. User Acceptance Testing (UAT)

- **Real User Testing:** Conduct testing sessions with real users to gather feedback on the website's functionality and user experience.
- **Feedback Implementation:** Incorporate feedback from UAT to make necessary adjustments and improvements.

By covering these aspects, the testing module ensures that the movie recommendation website is reliable, secure, user-friendly, and capable of providing a high-quality user experience.

6.2 Test Results

Functionality: All critical functions such as room search, booking, and payment processing were found to be working correctly.

Usability: The user interface was intuitive, and users were able to complete tasks without difficulty.

Compatibility: The application was tested on multiple devices and browsers, and no compatibility issues were found.

Performance: The application responded well under normal and peak loads, with acceptable response times.

Security: No major security vulnerabilities were found, and user data was protected.

Regression: New updates did not introduce any regressions, and existing functionality remained intact.

6.3 The Steps In Software Testing

The steps involved during Unit testing are as follows:

a. Preparation of the test cases.

b. Preparation of the possible test data with all the validation checks.

c. Complete code review of the module.

d. Actual testing done manually.

e. Modifications done for the errors found during testing.

f. Prepared the test result scripts.

SCREEN SHOTS OF MovieX App

Home page:

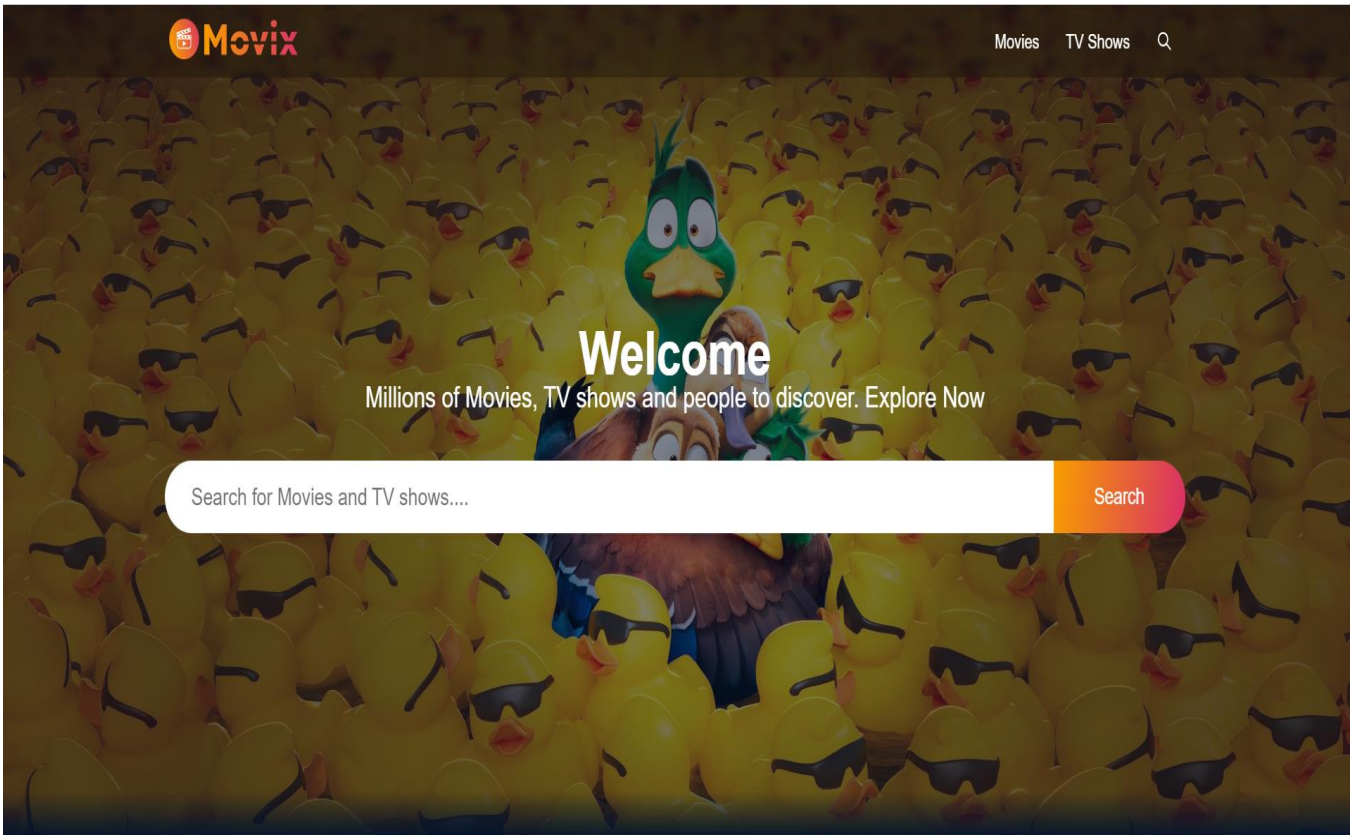

Figure 6: Home Page

User Login page:

User login to his/her account by filling username/email id with password.

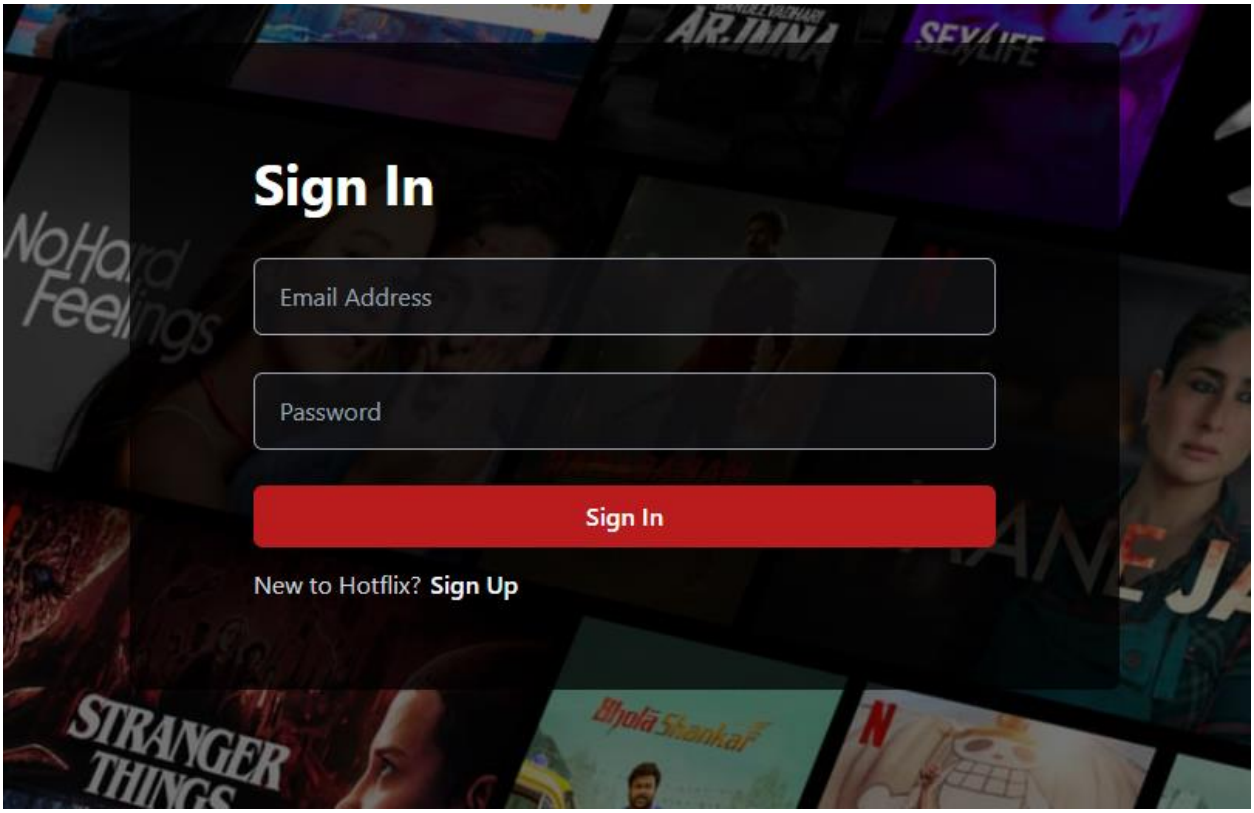

Figure 7: User Login Page

User Registration page:

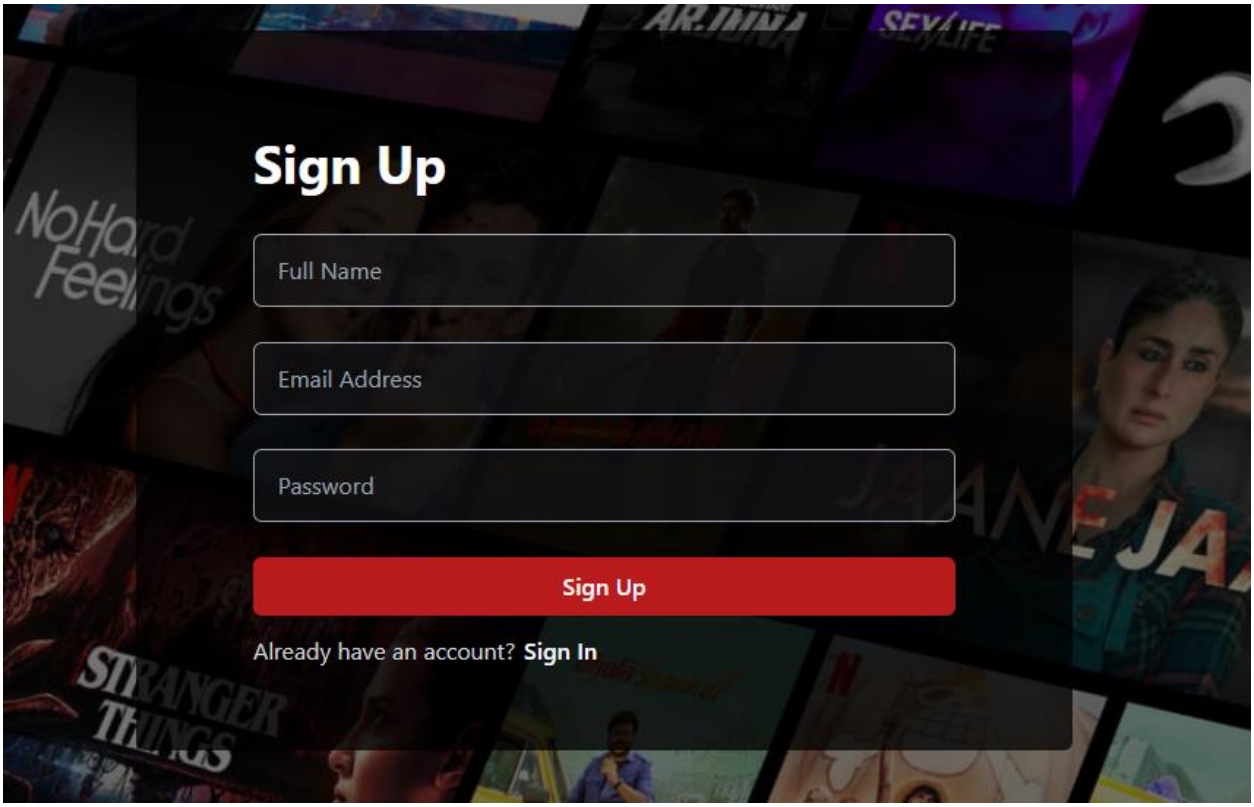

Figure 8: User Registration Page

Trending Movie Page:

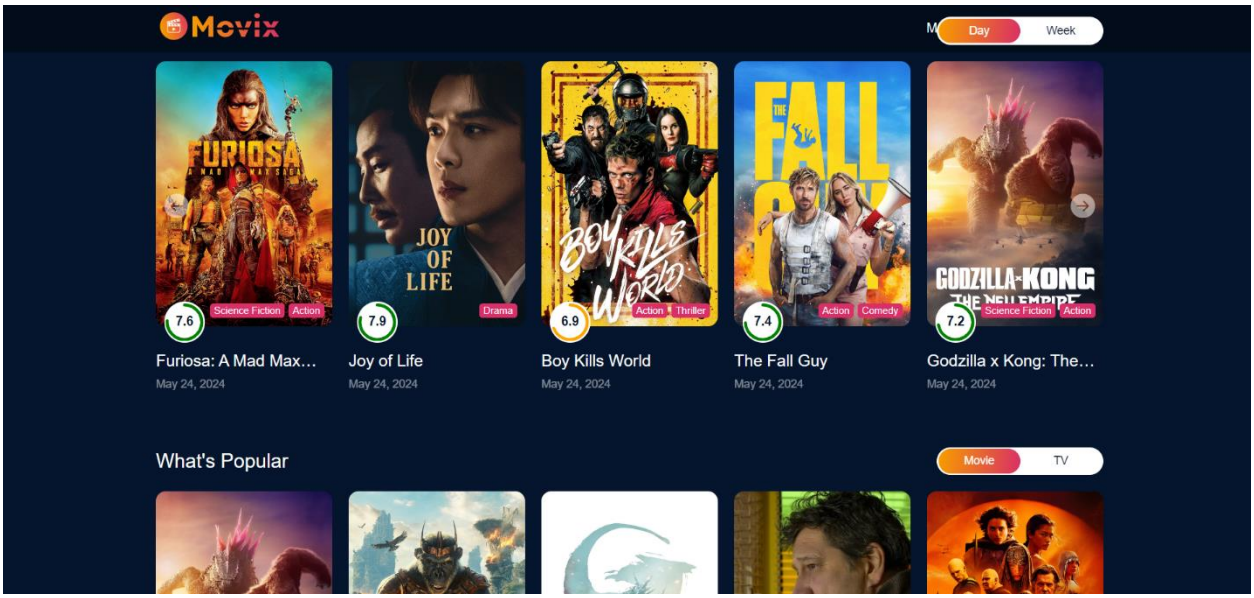

Figure 9: Trending Movie Page

Movie Detail Page:

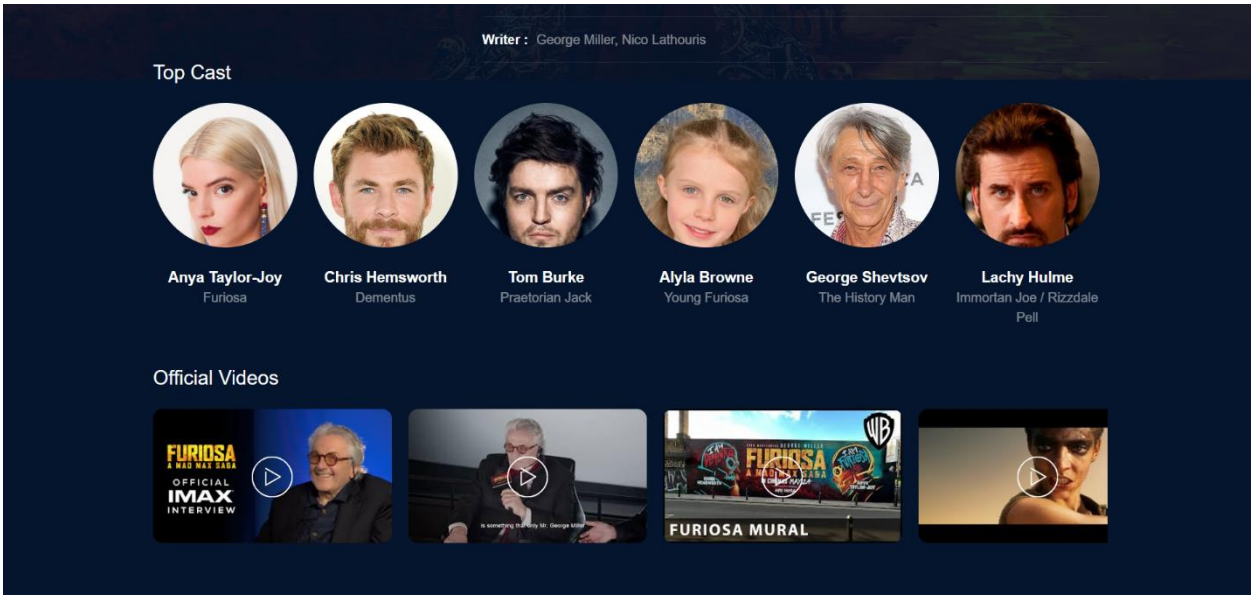

Figure 10: Movie Detail Page

Movie Page:

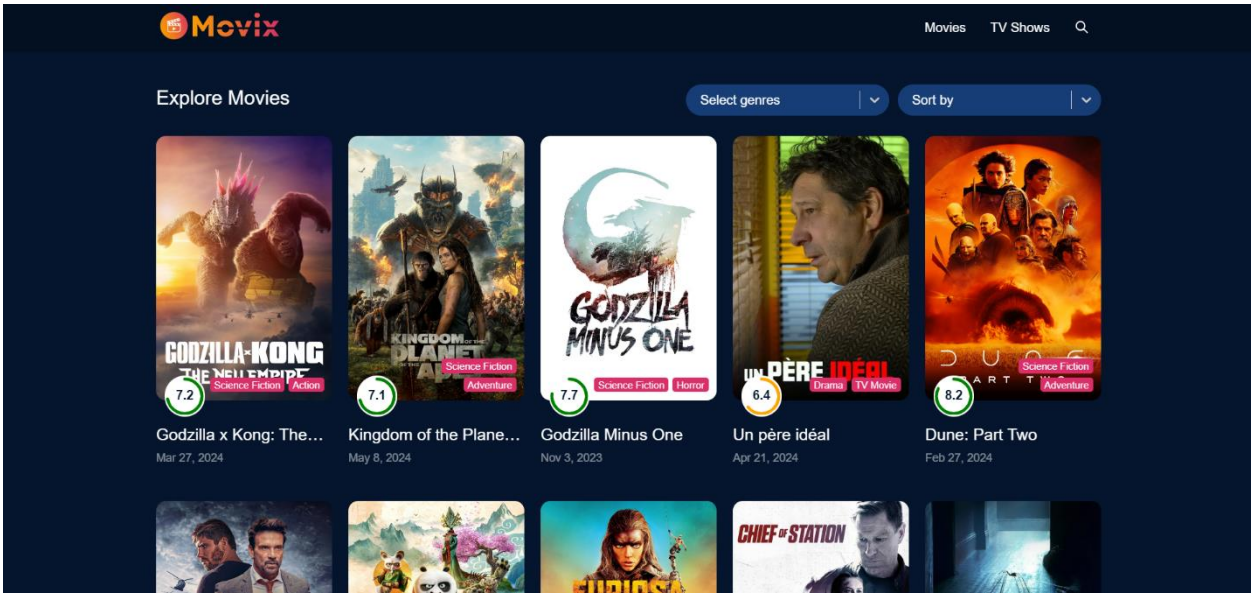

Figure 11: Movie Page

TV Shows page:

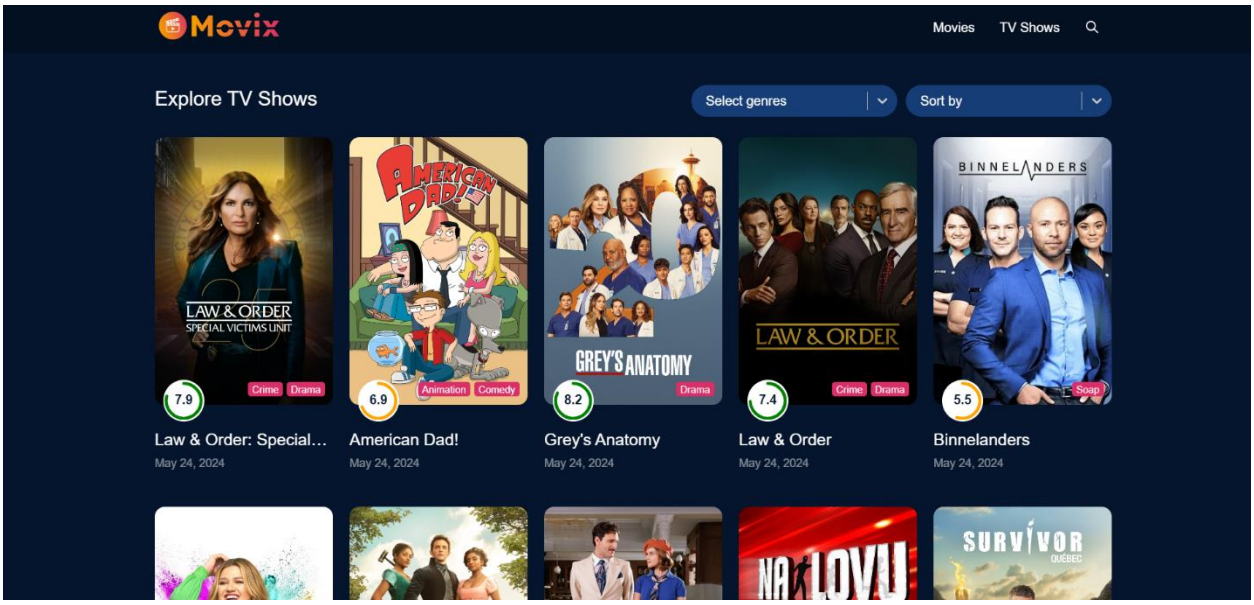

Figure 13: TV shows Page

CHAPTER 7

CONCLUSION

In conclusion, selecting a website for recommending movies in a college project involves careful consideration of various factors such as user interface, content diversity, recommendation algorithms, and user feedback mechanisms. Platforms like IMDb, Rotten Tomatoes, and Letterboxed offer comprehensive movie databases, user ratings, and personalized recommendations, making them suitable choices for such a project. Ultimately, the chosen website should align with project objectives and provide a seamless user experience for movie enthusiasts.

FUTURE ENHANCEMENT

Future enhancements for the movie recommendation website in the college project could include advanced recommendation algorithms, personalized user profiles, social features like reviews, integration with streaming platforms, and mobile app development for wider accessibility and improved user engagement.

CHAPTER 8

REFERENCES

- https://www.w3schools.com/php/
- https://www.javatpoint.com/html-tutorial
- https://www.canva.com/graphs/er-diagrams/
- https://ijcrt.org/papers/IJCRT22A6549.pdf
- https://github.com/
- https://www.tutorialspoint.com/java/
- https://www.geeksforgeeks.org/php-tutorial/?ref=lbp