

TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

MADAN BHANDARI MEMORIAL COLLEGE

*A Project Proposal On*

“MOVIE RECOMMENDER SYSTEM”

Submitted To:

Mr. Ram Kumar Basnet

DEPARTMENT OF CSIT

MADAN BHANDARI MEMORIAL COLLEGE

*In partial fulfilment of the requirements for the degree of*

*Bachelors of Science in Computer Science and Information Technology (B.Sc. CSIT)*

Submitted By:

Yugal Poudel

Siddhanth Shrestha

Famous Dhungana

Devraj Silwal

July, 2025

# Table of Content

[*Table of Content i*](#_Toc202650474)

[*List of Figures ii*](#_Toc202650475)

[1. Introduction 1](#_Toc202650476)

[2. Problem Statement 1](#_Toc202650477)

[3. Objectives 2](#_Toc202650478)

[4. Methodology 2](#_Toc202650479)

[4.1 Requirement Identification 2](#_Toc202650480)

[4.1.1 Study of Existing System / Literature Review 2](#_Toc202650481)

[4.1.2 Requirement Analysis 3](#_Toc202650482)

[4.2 Feasibility Study 4](#_Toc202650483)

[4.2.1 Technical Feasibility 4](#_Toc202650484)

[4.2.2 Operational Feasibility 4](#_Toc202650485)

[4.2.3 Economic Feasibility 4](#_Toc202650486)

[4.2.4 Schedule Feasibility 4](#_Toc202650487)

[4.3 High-Level Design of System 5](#_Toc202650488)

[4.3.1 Methodology of the Proposed System 5](#_Toc202650489)

[4.3.2 System Flowchart / Working Mechanism 5](#_Toc202650490)

[4.3.3 Description of Algorithms 7](#_Toc202650491)

[5. Expected Outcome 8](#_Toc202650492)

[6. References 8](#_Toc202650493)

# List of Figures

[Figure 1: Gantt Chart 5](#_Toc202650110)

[Figure 2: System Flowchart 6](#_Toc202650111)

# 1. Introduction

In the digital entertainment industry, users are often overwhelmed by the vast number of movies available across different platforms. Choosing a movie that aligns with one’s taste can become a tedious task. A movie recommender system provides users with personalized movie suggestions based on their preferences, making content discovery easier and more enjoyable.

This project aims to design and implement a movie recommendation web application that uses content-based filtering to suggest movies based on genres and user interactions. The system will demonstrate the application of software engineering principles including requirement gathering, system modeling, implementation, and testing within the scope of an academic project.

# 2. Problem Statement

With the rise of streaming platforms and online movie databases, users have access to an overwhelming number of movies across various genres and languages. While this provides a wide range of choices, it also makes it difficult for users to select a movie that matches their personal preferences.

Existing systems often provide generic lists based on popularity or manual filtering options, which do not always lead to satisfying recommendations. As a result, users spend a significant amount of time searching for something to watch.

There is a clear need for an intelligent movie recommender system that can analyze user preferences and behavior to suggest movies tailored to individual tastes. This project aims to address this problem by developing a user-friendly recommendation system that helps users quickly find movies they are likely to enjoy.

# 3. Objectives

The main objective of this project is to build a simple and effective movie recommender system that suggests movies based on user preferences. The system will be developed using basic software engineering concepts.

**Specific objectives include:**

* To create a web-based platform for users to register and log in.
* To allow users to search, browse and enjoy movies.
* To suggest movies using content-based filtering.
* To design a simple and user-friendly interface.
* To apply core software engineering principles during development.
* To manage the project using basic planning and scheduling tools.

# 4. Methodology

## 4.1 Requirement Identification

This phase focuses on understanding the needs of the users and studying existing solutions to define clear system requirements.

### 4.1.1 Study of Existing System / Literature Review

Several movie recommendation platforms exist today, such as Netflix, IMDb, and Amazon Prime Video. These platforms use complex recommendation algorithms like **collaborative filtering and deep learning** to provide personalized suggestions. However, their implementations require large datasets and advanced computing resources.

For this project, we focus on a simpler approach suitable for academic purposes, primarily using **content-based filtering techniques**. Content-based filtering recommends movies based on features such as genre, language, and user ratings. This approach is effective for smaller datasets and easier to implement, making it appropriate for the scope of this project.

Research articles and tutorials on recommendation systems emphasize the importance of understanding user preferences and movie attributes to generate meaningful recommendations. By reviewing existing systems and literature, this project aims to implement a basic but functional recommender system while applying software engineering principles.

### 4.1.2 Requirement Analysis

This phase clearly defines the essential functions the system must perform, along with the quality attributes it must possess to meet user expectations and operate smoothly.

**Functional Requirements:**

* Allow users to register and securely log in to the system
* Enable browsing of movies filtered by genre, language, and release year
* Provide search functionality to find movies by title or keywords
* Generate personalized movie recommendations based on user preferences

**Non-Functional Requirements:**

* Ensure the user interface is intuitive, responsive, and easy to navigate
* Implement secure authentication and protect user data privacy
* Maintain fast system response times for browsing and recommendation generation
* Design the system to be scalable for handling multiple simultaneous users
* Ensure compatibility with major web browsers and devices

## 4.2 Feasibility Study

This study evaluates the practicality of implementing the movie recommender system from different perspectives.

### 4.2.1 Technical Feasibility

The proposed system will be developed using commonly available technologies such as HTML, CSS, JavaScript for the frontend, Python (Flask or Django) or PHP for the backend, and MySQL for the database. These technologies are well-documented, widely supported, and suitable for academic projects, making the development technically feasible.

### 4.2.2 Operational Feasibility

The system will be accessible through any modern web browser, requiring no special software installation. It’s simple and intuitive design ensures ease of use for end users without any prior training. The operational requirements are minimal and can be met within the existing academic environment.

### 4.2.3 Economic Feasibility

As an academic project, the development incurs no direct software licensing or hardware costs. All necessary tools and frameworks are open source and free to use. Thus, the project is economically viable.

### 4.2.4 Schedule Feasibility

The project is planned to be completed within twelve weeks, divided into phases including requirement gathering, design, implementation, testing, and report preparation. The extended timeline allows for thorough development and testing. This schedule is realistic and achievable given the project scope and available resources.

*Figure 1: Gantt Chart*

## 4.3 High-Level Design of System

This section describes how the system will work at a high level, including its architecture, process flow, and core algorithm.

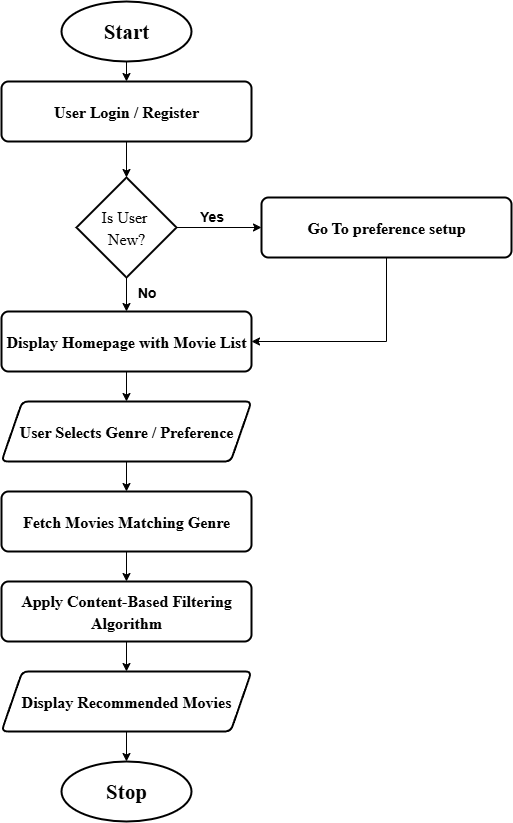
### 4.3.1 Methodology of the Proposed System

The system will follow a **content-based filtering** approach. This means the recommendation engine will suggest movies similar in content (e.g., genre) to those the user has previously liked or selected.

The system will be developed using a **modular and layered architecture**, separating the user interface, application logic, and database for better maintainability and scalability.

### 4.3.2 System Flowchart / Working Mechanism

The following flowchart illustrates the overall workflow of the movie recommender system. It shows how users interact with the system from login to receiving personalized movie recommendations based on their selected preferences.



*Figure 2: System Flowchart*

### 4.3.3 Description of Algorithms

The recommendation logic in this system is based on **content-based filtering**, which suggests movies similar to the user's selected preferences.

**How it works:**

* Each movie is described by features such as genre, language, and release year.
* When a user selects a genre or preference, the system creates a **feature vector** based on those inputs.
* It compares this vector with the feature vectors of all available movies.
* A **similarity score** is calculated (for example, using cosine similarity or a basic matching count).
* The top-scoring movies are selected and shown as recommendations.

**Why content-based filtering?**

* It works well without needing prior user ratings.
* It is suitable for new users (cold-start problem is handled through selected preferences).
* It is simple to implement and interpret, especially in small-scale academic projects.

This approach allows us to provide relevant movie suggestions using a straightforward algorithm that balances simplicity with effectiveness.

# 5. Expected Outcome

Upon successful completion of the project, the following outcomes are expected:

* A functional web-based movie recommender system that suggests movies based on user preferences.
* A clean and responsive user interface that allows easy navigation, search, and interaction.
* Implementation of content-based filtering to generate personalized recommendations.
* A structured and well-documented software project that follows software engineering principles.
* A final project report including requirement analysis, system design, testing, and conclusion.

This system will serve as a practical demonstration of how software engineering methodologies can be applied to build real-world applications.

# 6. References

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. Springer. <https://doi.org/10.1007/978-0-387-85820-3>

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.  
<https://doi.org/10.1007/978-3-319-29659-3>

Sharma, A., & Pathak, N. (2020). *Movie recommendation system using content-based filtering*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(1), 22–26.