

NLP Mini Project Report on
“Binge Recommendation System”

Submitted in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering
In
Computer Engineering
By

DEVIKA POTDAR (76)
APARNA PRASAD (77)
SAMIKSHA SHEREGAR (84)

Under the Guidance of

Prof. RUCHA PATWARDHAN



Department of Computer Engineering
Watumull Institute of Engineering and Technology
Ulhasnagar - 421003

UNIVERSITY OF MUMBAI
Academic Year 2025-2026

Approval Sheet

This Mini Project Report entitled "*Binge Recommendation System*" submitted by "*DEVIKA POTDAR*" (76), "*APARNA PRASAD*" (77), "*SAMIKSHA SHEREGAR*" (84) is approved for the partial fulfilment of the requirement for the award of the degree of *Bachelor of Engineering* in *Computer Engineering* from *University of Mumbai*.

Prof. _____
(Guide)

Place: Ulhasnagar

Date:

CERTIFICATE

This is to certify that the mini project entitled "*Binge Recommendation System*" submitted by "*Devika Potdar*" (76), "*Aparna Prasad*" (77), "*Samiksha Sheregar*" (84) for the partial fulfilment of the requirement for award of a degree *Bachelor of Engineering* in *Computer Engineering*, to the University of Mumbai, is a Bonafide work carried out during academic year 2025-2026.

Prof.

(Guide)

Place: Ulhasnagar

Date: / /25

Declaration

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

DEVIKA POTDAR (76) _____
APARNA PRASAD (77) _____
SAMIKSHA SHEREGAR (84) _____

Place: Ulhasnagar

Date: / /25

Abstract

Binge Recommendation is a web application designed to simplify finding movies and TV shows. It addresses the problem of too many content choices on streaming platforms. The system provides quick, relevant suggestions without needing user history, solving the “cold-start” issue.

By analysing metadata like genres, directors, cast, and plot summaries, the app uses Natural Language Processing (NLP) with Term Frequency–Inverse Document Frequency (TF-IDF) vectorization to create numerical representations. Cosine similarity ranks the most similar content, providing ten curated recommendations. Each recommendation shows title, genre, and description.

The platform is built with Python and Streamlit, offering a lightweight, interactive, and user-friendly web interface. Pandas and NumPy handle data manipulation and calculations efficiently. Preprocessing with regular expressions keeps input clean for accurate modelling.

The dataset comes from Kaggle’s Netflix Movies and TV Shows collection. Content-based filtering demonstrates how AI can generate clear and efficient recommendations. Users get personalized suggestions in a simple and structured way.

Future plans include integrating collaborative filtering for hybrid recommendations and expanding to more streaming platforms. Features like filtering by genre or year and “more like this” options are planned. It combines NLP, machine learning, and an easy interface to make content discovery smooth and personalized.

Content

Sr. No.	Topic	Page No
1.	Introduction	1-3
1.1	Overview	1
1.2	Introduction	1
1.3	Objective	1
1.4	Scope	2
1.5	Purpose	3
2.	Problem Definition	4
2.1	Problem Statement	4
2.2	Technology	4
3.	Hardware and Software Specification	5
3.1	Hardware Specification	5
3.2	Software Specification	5
4.	Design	6-8
4.1	Explanation	6
4.2	Steps in the program	6
4.3	Flowchart	8
5.	Result	12 - 14
5.1	Screen Shots of test result or output	12
6.	Future Scope and enhancements	10
7.	Conclusion	11
8.	Bibliography	12
	Acknowledgment	13

Chapter 1: Introduction

1.1 : Overview

In today's digital streaming world, platforms like Netflix, Amazon Prime, and Disney+ host thousands of movies and TV shows. While this abundance provides unlimited entertainment options, it also creates a choice overload problem users often spend more time deciding what to watch rather than enjoying the content. To tackle this, recommendation systems have become essential for simplifying user decisions and improving overall viewing experience.

The **Binge Recommendation System** is a content-based movie and TV show recommendation web application that provides intelligent and personalized suggestions. Unlike collaborative systems that rely on user history or ratings, Binge Recommendation uses content similarity analysing the metadata of titles, including genres, directors, cast, and descriptions to generate recommendations.

The system uses **NLP** techniques such as **TF-IDF vectorization** and **cosine similarity** to represent text numerically and calculate relationships between shows and movies. Built using **Python and Flask**, with a **beautiful light brown theme**, the web application provides an interactive, user-friendly experience with filters, search, and a "Surprise Me!" option for random suggestions.

This project shows how **Artificial Intelligence (AI) and NLP** can enhance digital entertainment by helping users quickly find content that matches their preferences even without user history making streaming more efficient and enjoyable.

1.2 : Objective

The main objective of this project is to create a user-friendly, intelligent recommendation system that assists users in discovering movies and TV shows efficiently. The key goals are:

- To implement a **content-based filtering** system using **TF-IDF** and **cosine similarity**.
- To recommend titles based on **metadata features** such as genre, cast, director, and description.
- To build an interactive **Flask web application** that provides real-time, responsive, and engaging user experience.
- To overcome the **cold-start problem** by generating recommendations even for new users without viewing history.
- To use **NLP preprocessing** (tokenization, cleaning, and vectorization) for accurate content similarity measurement.

By achieving these objectives, Binge Recommendation bridges data science and web design to deliver an effective, attractive, and intelligent movie recommendation platform.

1.3 : Scope

The scope of the **Binge Recommendation System** focuses on content-based filtering using NLP and machine learning techniques. The project includes:

- Using the **Netflix Movies and TV Shows dataset** from Kaggle as the data source.
- Extracting and preprocessing metadata like genres, cast, country, director, and description.
- Using **TF-IDF Vectorization** to represent text numerically and **cosine similarity** to calculate content similarity.
- Displaying the top 10 recommended titles with details such as title, genre, description, and similarity score.
- Building a **Flask-based web interface** with Bootstrap and Cascading Style Sheets (CSS) for interactive, mobile-responsive design.
- Adding advanced features such as **genre and country filtering**, **random “Surprise Me!” suggestions**, and **smart dropdowns** for movie selection.

The current version uses a static dataset and English-language content. However, it sets a strong base for future improvements such as hybrid recommendation, multilingual support, and integration with live streaming APIs.

1.4: Purpose

The purpose of this project is to simplify the user experience of discovering entertainment content online. By implementing AI-powered recommendations, the system reduces the time and effort users spend browsing through endless options.

For streaming services, it demonstrates how **machine learning and NLP** can increase viewer engagement and satisfaction. For learners, it serves as a hands-on example of how data science concepts can be combined with web development.

Overall, Binge Recommendation aims to make watching movies and shows easier, faster, and more enjoyable through smart, personalized, and well-designed recommendations.

Chapter 2: Problem Definition

2.1: Problem Statement

With the rapid growth of digital streaming services, users face the problem of overwhelming content availability. Deciding what to watch next becomes time-consuming and frustrating. Traditional systems that rely on user ratings or histories fail for new users and new titles due to the cold-start problem.

The **Binge Recommendation System** addresses this by focusing on *content-based filtering* rather than user-based data. It analyses each title's features genre, cast, director, and description to find and rank similar content. Using **TF-IDF vectorization** and **cosine similarity**, the system can suggest relevant titles without needing prior watch data.

Additionally, many existing systems lack modern interfaces or visual appeal. To solve this, a **Flask-based web app** with an elegant **light brown theme** is developed. The app includes dynamic filters, real-time search, and a "Surprise Me!" button to make the experience interactive and user-friendly.

2.2: Technology Used

- **Programming Language:** Python 3.x
- **Framework:** Flask (for backend and web routes)
- **Libraries & Tools:**
 - scikit-learn – for TF-IDF vectorization and cosine similarity
 - pandas & NumPy – for data handling and analysis
 - regex (Regular Expressions) – for text preprocessing
 - Matplotlib, Seaborn – for data visualization
- **Frontend:** HTML, CSS, Bootstrap 5
- **Theme Colors:** Light brown palette (#F5E6D3, #E8D5C0, #2C1810)
- **Dataset:** Netflix Movies and TV Shows dataset from Kaggle
- **Development Tools:** Jupyter Notebook, VS Code
- **Deployment:** Flask local server (<http://localhost:5000>)

Chapter 3: Hardware and Software Specification

3.1: Hardware Specification

- Processor: Intel Core i3 or higher
- RAM: Minimum 4 GB
- Storage: At least 1 GB free space
- Internet Connection: For dataset and dependencies download

3.2: Software Specification

- **Operating System:** Windows 10 / macOS / Linux
- **Python Version:** Python 3.x
- **Libraries:** Flask, pandas, NumPy, scikit-learn, regex, matplotlib, seaborn
- **Tools Used:** VS Code / Jupyter Notebook
- **Browser:** Google Chrome / Edge for running the web app

Chapter 4: Design

4.1: Explanation

The Binge Recommendation System is designed to simplify the process of discovering movies and TV shows through a structured, step-by-step approach that combines data preprocessing, NLP-based feature extraction, and web-based interaction.

The system begins by loading and cleaning the Netflix dataset using pandas. Missing data is handled, and unnecessary characters are removed through regular expression preprocessing. Then, the project focuses on transforming textual and categorical features such as genres, cast, directors, and descriptions into a numerical form that can be analyzed mathematically.

The cleaned and combined text data is represented using TF-IDF vectorization, which helps identify the importance of words across all titles. Using cosine similarity, the model then compares these TF-IDF vectors to find the top 10 most similar titles to a given movie or show.

All these backend operations are integrated with a Flask web application, providing an interactive, visually appealing, and easy-to-use interface. The web interface allows users to search for movies, apply filters such as genre or country, and view personalized recommendations in real time. Additionally, the “Surprise Me!” feature provides random movie suggestions for users who want spontaneous viewing options.

The design ensures that each component from data handling to recommendation logic and user interface works cohesively to deliver fast, relevant, and engaging content suggestions.

4.2 Steps in the Program:

1. **Dataset Collection:**

The dataset used in this project is the Netflix Movies and TV Shows dataset from Kaggle, containing fields like title, cast, director, country, release year, rating, and description.

2. **Data Loading and Cleaning:**

Using pandas, the dataset is loaded, and missing values in columns like director, cast, country, and rating are filled with “Unknown.” Regular expressions are used to remove special symbols, punctuation, and extra spaces from the text fields.

3. **Feature Engineering:**

New columns such as processed cast, processed_director, and processed_genres are created by cleaning and formatting metadata. These features are combined with the cleaned description into a single text field called “soup.”

4. **TF-IDF Vectorization:**

The TF-IDF Vectorizer from scikit-learn converts the “soup” text into numerical feature vectors, representing the significance of each word relative to all titles.

5. **Cosine Similarity Calculation:**

The system computes the cosine similarity matrix to measure how close each movie or show is to every other title in the dataset.

6. Recommendation Function:

A Python function retrieves the index of the selected movie, sorts other titles by similarity score, and displays the top 10 most relevant recommendations with their details.

7. Flask Application Integration:

The backend logic (recommendation engine) is integrated into a Flask web app. The *app.py* file defines routes for the homepage, recommendation display, and the “Surprise Me!” feature.

8. User Interface (Templates):

Using Bootstrap 5 and custom CSS, the web interface includes a clean light brown theme with a movie dropdown, filter options for genre/country, and responsive movie cards.

9. Testing and Execution:

The application runs locally via Flask at <http://localhost:5000>. Users can enter or select a movie, and the system instantly generates personalized recommendations.

10. Output Display:

The final recommendations are presented with key details such as title, genre, description, similarity score, and more. The UI also supports random discovery via the “Surprise Me!” button.

4.3: Flowchart

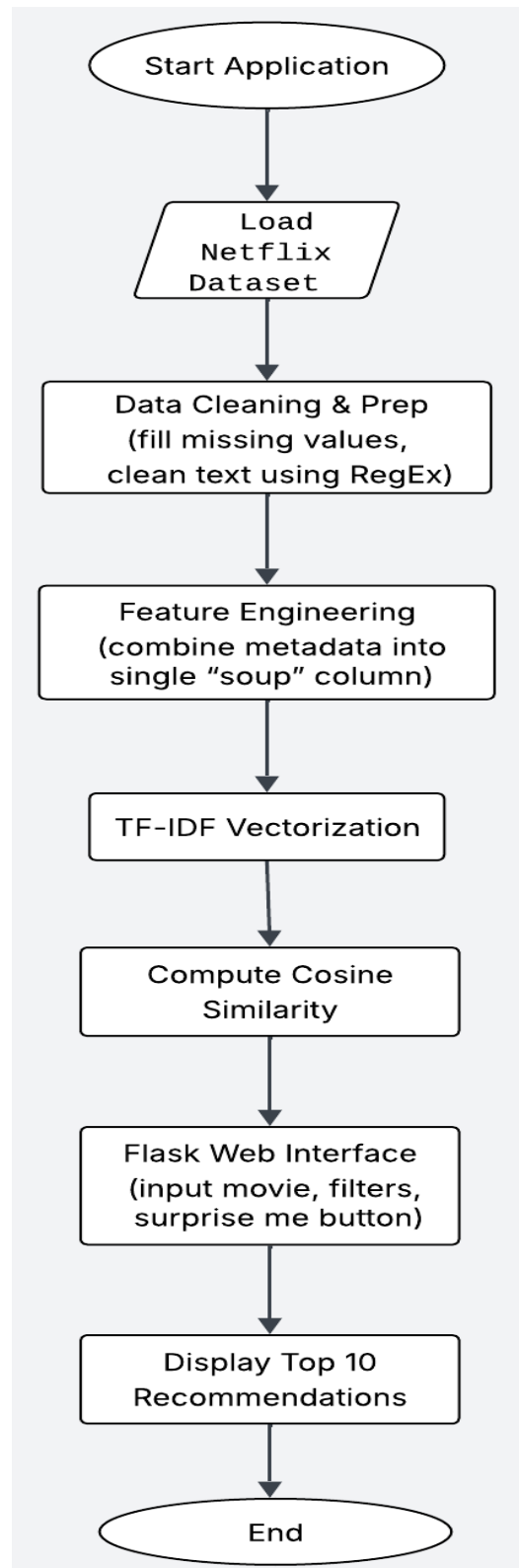
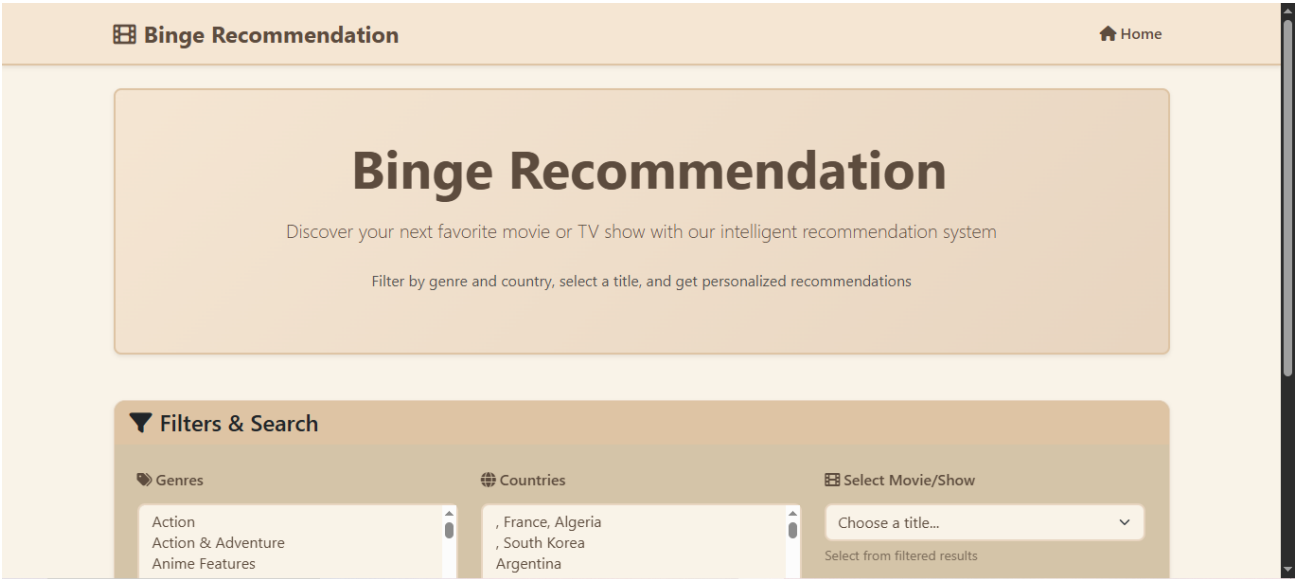


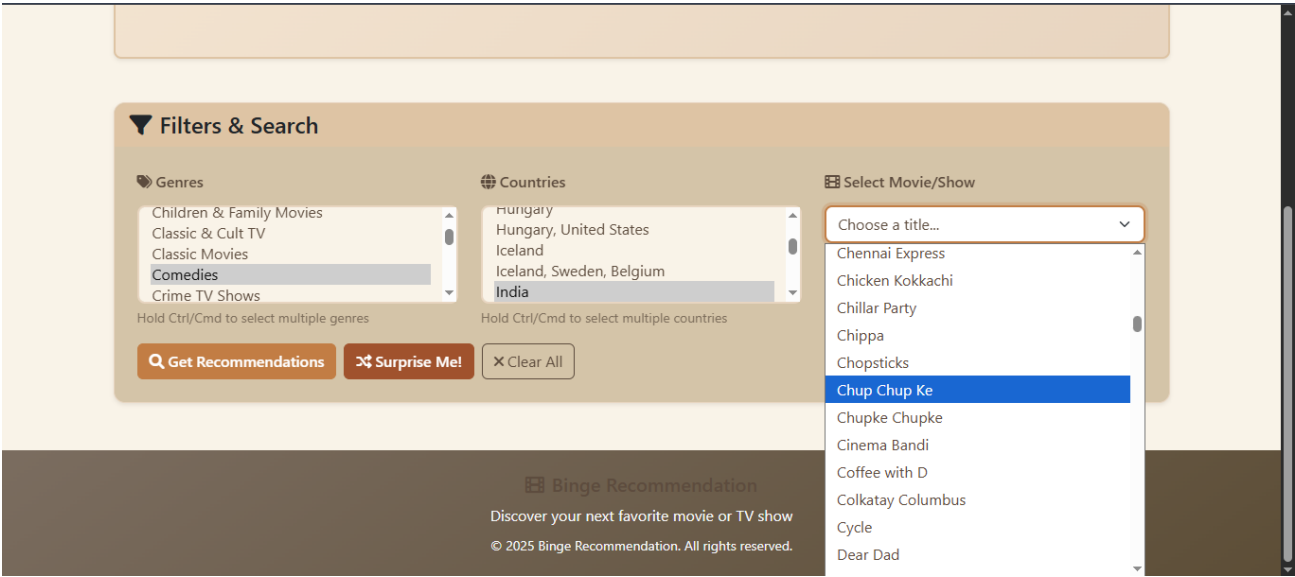
Fig: 4.3.1

Chapter 5: Result

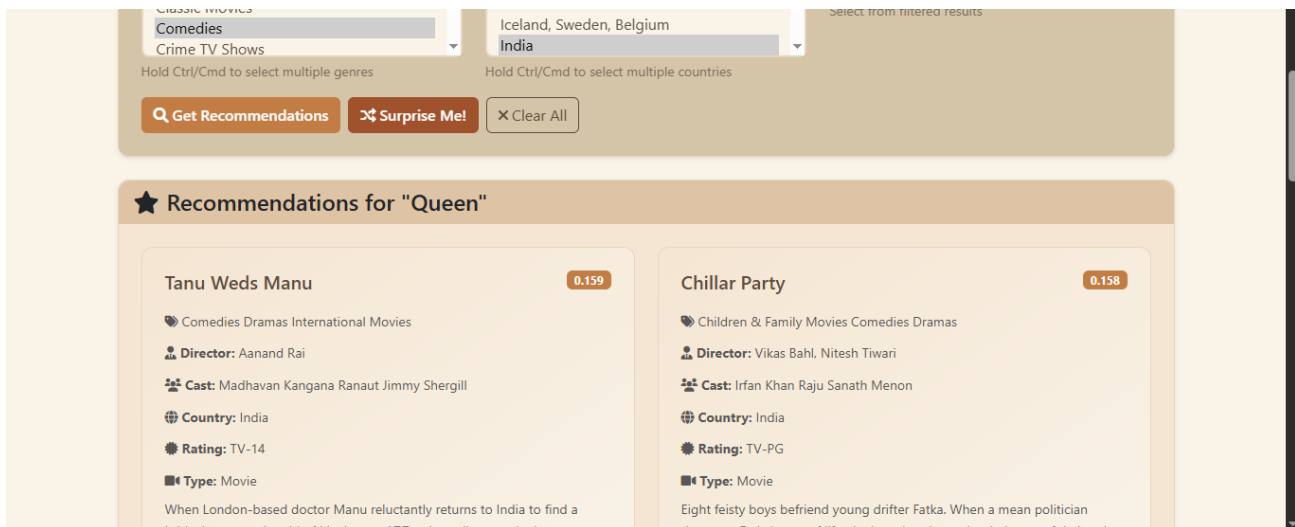
5.1 Test Results



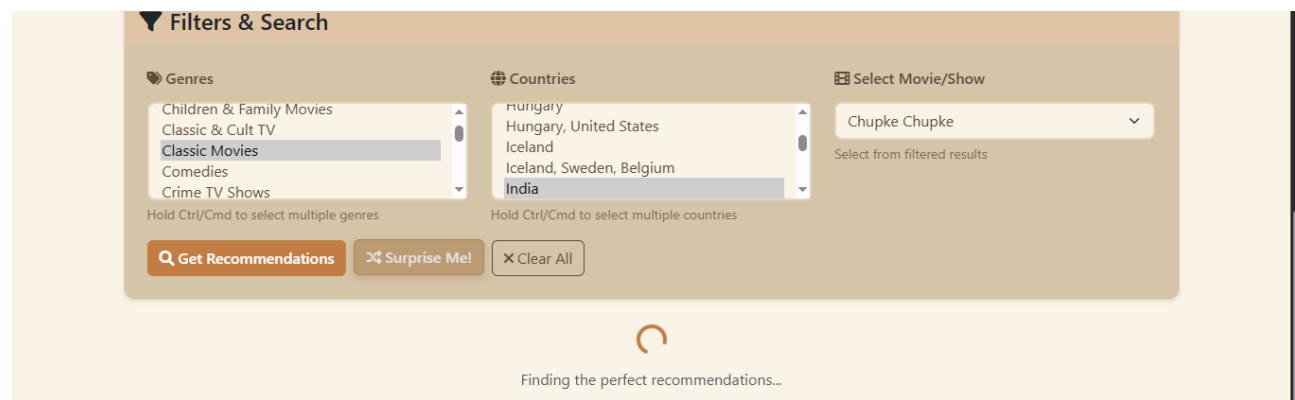
5.1.1 Clean light brown themed homepage with filters and search form



5.1.2 Responsive movie selection dropdown dynamically filtered by genre and country



5.1.3 Recommendation cards with similarity scores, genres, cast, director, country, description



5.1.4 Loading spinner animations 4 during recommendation generation



5.1.5 "Surprise Me!" button resulting in spontaneous random movie suggestions

Chapter 6: Future Scope and Enhancements

The **Binge Recommendation System** can be further improved in several ways to make it more powerful, accurate, and user-friendly:

1. **Hybrid Recommendation:**

Future versions can combine **content-based and collaborative filtering** to create a hybrid model. This approach will consider both movie features and user viewing patterns, allowing for more personalized and relevant recommendations.

2. **Multilingual Support:**

The system can be extended to work with **non-English datasets**, enabling recommendations for movies and shows in multiple languages. This would make the platform useful for a wider global audience.

3. **User Profiles:**

Adding a **login and profile system** would allow users to create accounts, save favourite movies, and maintain personalized watchlists for future recommendations.

4. **Real-Time Data Integration:**

Future updates can connect the app with **live streaming APIs** to fetch the latest movies and series automatically, keeping recommendations up-to-date with current releases.

5. **Emotion & Sentiment Analysis:**

Integrating **emotion detection** could help match movies to users' moods for example, suggesting comedies when a user feels sad or thrillers when they seek excitement.

6. **Cloud Deployment:**

The system can be hosted on **Amazon Web Services (AWS) or Heroku**, making it publicly accessible from anywhere and improving performance, scalability, and user reach.

Chapter 7: Conclusion

The Binge Recommendation System uses NLP and machine learning to suggest movies and TV shows. It employs TF-IDF and cosine similarity for accurate content-based recommendations without user watch history. Flask ensures a smooth, interactive, and visually appealing interface.

Features include genre filters, “Surprise Me!” random suggestions, and a responsive design for easy navigation. Users can quickly find content that matches their preferences. The system makes the experience simple and enjoyable.

AI helps reduce decision fatigue and speeds up content discovery by guiding users to relevant movies and shows. Python combined with web technologies makes the platform practical and user-friendly. It demonstrates intelligent, real-world applications.

The system supports future upgrades like hybrid recommendation models, user personalization, and real-time data integration. These enhancements ensure continuous improvement and adaptability. The platform remains relevant in the evolving digital entertainment landscape.

Chapter 8: Bibliography

1. Kaggle – Netflix Movies and TV Shows Dataset
2. Flask Documentation – <https://flask.palletsprojects.com>
3. scikit-learn Documentation – <https://scikit-learn.org>
4. Bootstrap 5 – <https://getbootstrap.com>
5. W3Schools – CSS Color Palettes and Styling
6. GeeksforGeeks – Movie Recommendation System using Python
7. DigitalOcean – Structuring Flask Applications
8. Color-Hex – Light Brown Color Palette Reference
9. Medium – Building Content-Based Recommendation Systems
10. Stack Overflow – Flask and Python Best Practices

Acknowledgement

We take great pleasure in presenting this mini project report on “**BINGE RECOMMENDATION SYSTEM.**” We would like to express our sincere gratitude to our guide, **Prof. Rucha Patwardhan**, Department of Computer Engineering, Watumull Institute, for her constant support, technical guidance, and insightful suggestions that have greatly contributed to the successful completion of this project.

We are thankful to **Prof. Ranjana Singh**, Head of the Department, Computer Engineering, Watumull Institute, for her encouragement during the progress reviews and for providing valuable inputs that helped shape this report.

We extend our sincere thanks to **Prof. Avinash V. Gondal**, Principal of Watumull Institute, for his continued support and encouragement throughout the duration of the project.

We would also like to thank all the faculty and staff members of Watumull Institute of Engineering and Technology for their help and cooperation during the course of this work.

Lastly, we express our heartfelt gratitude to our colleagues and friends at Watumull Institute for their encouragement, support, and valuable feedback during this project journey.

DEVIKA POTDAR (76)

APARNA PRASAD (77)

SAMIKSHA SHEREGAR (84)