

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



A project report
On
“Recherche de Cinema”

[Code No: COMP 207]
(For partial fulfilment of II Year/ II Semester in Computer Engineering)

Submitted By:
Rishav Kumar Paudel (35)
Chuddant Shrestha (43)
Manjit Singh (47)
Subash Chandra Yadav (58)

Submitted To:
Mr. Suman Shrestha
Department of Computer Science and Engineering

Submission Date: 2025/11/16

Abstract

Cinema is an integral part of our lives. Over Nine Thousand and Five Hundred movies are created annually and as the saying goes, ‘with abundance comes trouble’, people go through a tough cycle of finding a great movie to watch, view it and find another one like such. Platforms like Letterboxd and IMDB may provide users an ability to track or browse movies, they do not provide them personalized suggestions based on the user’s taste and preferences. Our project proposes to develop a dedicated web platform that would leverage artificial intelligence to recommend users the perfect movie that they would watch based on their preferences in genres. The system will analyse different data points such as genres, casts, directors and other metadata to generate personalized recommendations and assign scores. Additional Features include, dashboard viewing, Automated Speech Recognition or ASR to enhance the usability of the platform.

Our tech stack will include MongoDB, Node.js, React.js, python, Tailwind CSS and etc. depending on the future requirements of the project.

Our aim is to learn these essential skills while developing a project that will align with our personal interests and for the users worldwide. By providing accessible and actionable information, ‘Recherche de Cinema’ aims to provide cinephiles and etc. a feature based rich platform that would enhance their movie viewing experience and us developers a chance in expanding our skills in web development and parts of machine learning as it is essential in the learning curve of Computer Engineering and Data Analysis.

Keywords: *Artificial Intelligence, Web Development, Automated Speech Recognition, Machine Learning, Data Analysis, Metadata, MongoDB, Node.js, python, Tailwind CSS, React*

Table of Contents

ABSTRACT.....	1
TABLE OF CONTENTS.....	2
LIST OF FIGURES	3
ACRONYMS/ABBREVIATIONS.....	4
CHAPTER 1 INTRODUCTION.....	5
1.1 BACKGROUND	5
1.2 OBJECTIVES.....	5
1.3 MOTIVATION AND SIGNIFICANCE	6
1.4 EXPECTED OUTCOMES	6
CHAPTER 2 RELATED WORKS/EXISTING WORKS.....	8
2.1 LETTERBOXD	8
2.2 IMDB (INTERNET MOVIE DATABASE).....	8
CHAPTER 3 PROCEDURE AND METHODS.....	10
3.1 DEVELOPMENT PROCEDURE	10
3.1.1 REQUIREMENT ANALYSIS AND PLANNING.....	10
3.1.2 SYSTEM DESIGN.....	10
3.1.3 TESTING AND DEBUGGING	11
3.1.4 DOCUMENTATION.....	12
3.2 SYSTEM ARCHITECTURE	12
3.3 SYSTEM TABLE	12
CHAPTER 4 SYSTEM REQUIREMENT SPECIFICATION	14
4.1 SOFTWARE REQUIREMENTS	14
4.1.1 FUNCTIONAL REQUIREMENTS	14
4.1.2 NON-FUNCTIONAL REQUIREMENTS	14
4.2 HARDWARE SPECIFICATIONS	15
4.2.1 SERVER-SIDE REQUIREMENTS:	15
4.2.2 CLIENT-SIDE REQUIREMENTS:.....	15
CHAPTER 5 PROJECT PLANNING AND SCHEDULING.....	16
5.1 TASKS	16
CHAPTER 6 REFERENCES.....	17

List of Figures

Figure 1: Letterboxd	8
Figure 2: IMDb	9
Figure 3: System Architecture Design	12
Figure 4: System Table	13
Figure 5: Gantt-Chart.....	16

Acronyms/Abbreviations

AI: Artificial Intelligence

ASR: Automated Speech Recognition

ML: Machine Learning

DB: Database

UI: User Interface

UX: User Experience

API: Application Programming Interface

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

JSON: Javascript Object Notation

MERN: MongoDB, Express, React, Node

IDE: Integrated Development Environment

RAM: Random Access Memory

CPU: Central Processing Unit

Chapter 1 Introduction

1.1 Background

After the rise in popularity of Letterboxd, we were left with an idea to make a similar feature rich platform that would also aid the feature of personalized recommendations that is not present in Letterboxd for users across the world. With an enormous and continuously expanding number of films across different streaming platform and regions, users more often than not struggle in finding what to watch and track it.

Most common movie tracking websites are Letterboxd and IMDB, while they provide a huge catalogue of movies for tracking and reviewing movies, they lack in providing personalized suggestions through data analysis and a voice search system. As a result of this, the user is left with a tedious typical cycle of browsing the catalogue, checking reviews, watching a film and finding another one.

On the other note, there is a huge rise in popularity of data science and artificial intelligence across the world, we believe that not working in these topics as computer science students leaves us out behind the modern up and coming developers.

So by working on this project, we aim to develop a rich feature-based platform that not only lets the users maintain lists, user ratings and reviews but also intelligently curates the recommendations based on their unique tastes and data. We will also get a chance to work in data analysis and furnish our skills in web development.

1.2 Objectives

The major objectives of our project are listed in the next page:

- Develop AI based recommendation system that generates personalized suggestions for users based on their preferences.
- Design a feature rich web platform that provides great UI and stable backend
- Implement Automated Speech Recognition (ASR) to enable voice search.
- Integrate machine learning and data analysis.

1.3 Motivation and Significance

With the rapid growth of digital streaming platforms and the increasing volume of films produced annually, people struggle with an overwhelming task of selecting what to watch. Existing platforms such as Letterboxd and IMDB offer updated and extensive catalogue but they do not provide personalized suggestions tailored to individual's tastes. This results in a repetitive cycle of browsing, reviewing catalogues and finding a suitable movie to watch.

At the same time, the global shift towards data driven technologies, machine learning and artificial intelligence, it has transformed how modern systems provide personalized content. As Computer Engineering students, it is essential to learn these skills to stay aligned and up to date with the modern industry requirements. This project helps us merge our interests in cinema and learning these new technologies to create a rich feature filled platform for users worldwide.

1.4 Expected Outcomes

- A fully functional web program that allows users to browse, track, review and manage their movie lists.
- Personalized movie recommendation system

- An Automated Speech Recognition Feature
- A dynamic user dashboard.

Chapter 2 Related Works/Existing Works

The related works are listed below:

2.1 Letterboxd

Letterboxd is a social platform that helps users track, rate and review movies. It also provides a feature of adding any movie to the watchlist from their wide catalogue. While they emphasize on community engagement, they lack the feature of providing personalized recommendations based on the user's preferences.

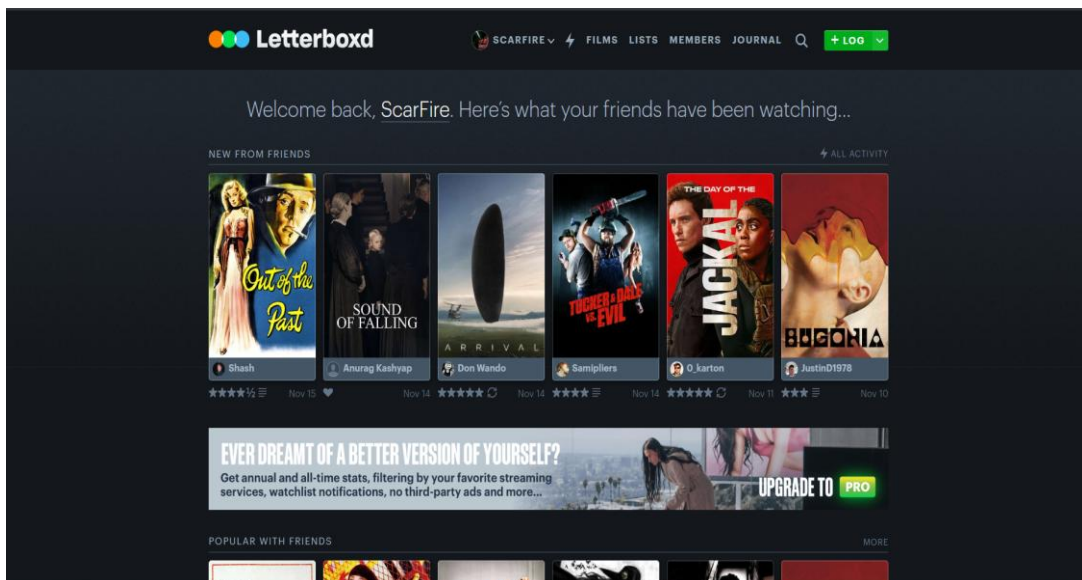


Figure 1: Letterboxd

2.2 IMDb (Internet Movie Database)

Internet Movie Database or IMDb is a comprehensive database of movies, TV shows and any kind of entertainment media offering detailed metadata such as cast, directors, genres etc. IMDb also has the same issue of lack of personalization in creating recommendations for the users. Additionally, much like Letterboxd it also lacks a voice-based search system.

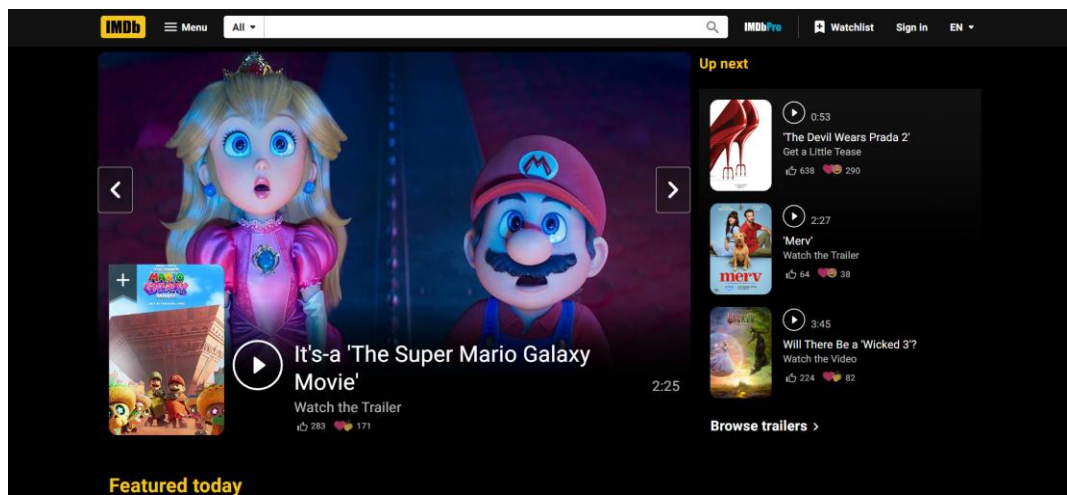


Figure 2: IMDb

Chapter 3 Procedure and Methods

This section showcases the step-by-step approach we will follow to complete this project. This project follows a traditional software development that is going through four major steps: Planning, Design, Development and Testing/Deployment.

3.1 Development Procedure

3.1.1 Requirement Analysis and Planning

- Gather necessary requirements and features that are to be implemented such as movie-catalogue, login system, API keys etc.
- Conduct research on existing platforms to find the gaps to be filled.
- Define functional and non-functional requirements to the platform
- Plan the overall system architecture, system stack and developmental goals.

3.1.2 System Design

- **Front-End Design:** We first plan to work on the front-end on the platform. We will sketch out the different pages of webpage in paper and design the front-end accordingly with the help of Tailwind CSS and React.js. The UI will emphasize simplicity, clarity and a beautiful webpage such that the user will seamlessly interact with the different services of the platform.
- **Back-End Design:** There will be a definite database that will log different user data using the help of MongoDB. This phase of system design will integrate core features such as user registration, preference management and product recommendation workflows.

The backend will be built with the help of Node's runtime environment. A machine learning model will analyse the product information and generalize personalized results for users.

- **Voice-Search System:** We will integrate browser-compatible ASR(e.g: Web Speech API or any JavaScript-based ASR library). The search bar will contain a microphone icon that users will click to initiate voice input. Once clicked, the browser will send a request to access the permission to the microphone. After that, the Web Speech API will listen to the voice and convert it to text which will be automatically inserted into the search bar.
- **AI/ML Integration:** We will use a hybrid recommendation system. It analyses user preferences, watch history and movie metadata to generate personalized movie suggestions. The AI/ML module will be developed with the help of Python and will be connected to Node.js and the recommendations stored in MongoDB for fast retrieval.

3.1.3 Testing and Debugging

- Comprehensive testing will be conducted to ensure platform stability and correctness.
- Unit testing and manual testing will be used to verify each component.
- The team will actively debug issues, address performance bottlenecks and ensure app meets privacy and security standards.

3.1.4 Documentation

- We will document our code and developmental progress online in GitHub.
- We will make some final adjustments and then publicize the project online using vercel.

3.2 System Architecture

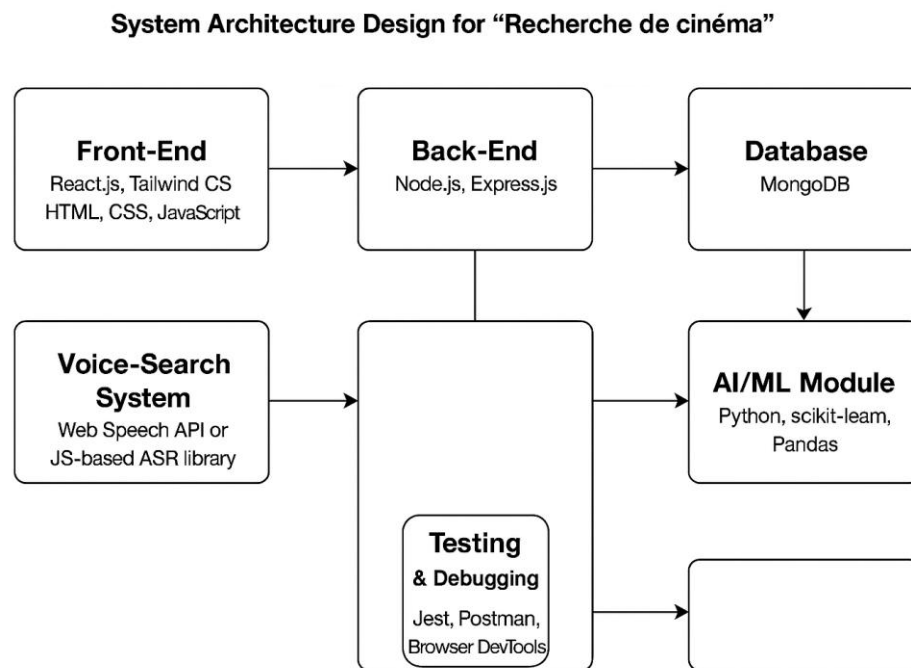


Figure 3: System Architecture Design

3.3 System Table

The system table for this platform is given below.

Module	Description	Tools Used
Front-End	UI/UX Development	React and CSS
Back-End	Handling Server Side Logics	Node.js and Express.js
Database	Stores user data, movie metadata	MongoDB
Data Analysis and Reporting	Tracks user behaviour for recommendations	Python

Figure 4: System Table

Chapter 4 System Requirement Specification

4.1 Software Requirements

4.1.1 Functional Requirements

- **Visibility:** Users should be able to access the website without signing in.
- **Browsing:** Users should be able to browse through the movie catalogue.
- **Voice Search:** Users should be able to voice search.
- **Recommendation Engine:** The system must analyse user history and recommend accordingly.
- **Database:** User's data must be stored in a database.

4.1.2 Non-Functional Requirements

- **Performance:** The system should load pages under 3 seconds.
- **Scalability:** The system must support at least 10 thousand users.
- **Reliability:** The system must be reliable and maintain user privacy.
- **Compatibility:** The system must be compatible.

4.2 Hardware Specifications

4.2.1 Server-Side Requirements:

- **Processor:** Minimum 2.5 GHz Multi-Core processor (intel i5/i7).
- **RAM:** 8 GB DDR3 RAM or Higher
- **Network:** Stable and High Speed Connection
- **Development Tools:** VS Code Text Editor/ WebStorm/ Any preferred IDE
- **Operating System:** Linus Based/ MacOS/ Windows.

4.2.2 Client-Side Requirements:

- **Processor:** Minimum 1.5 GHz.
- **RAM:** 2 GB RAM or higher.
- **Browser:** Latest version of Chromium Based browser or any.

Chapter 5 Project Planning and Scheduling

We are intending to work a day every week. If we manage to stay regular and disciplined to the schedule, we can finish this in 3 months of time with ease. The Gantt-Chart is given below.

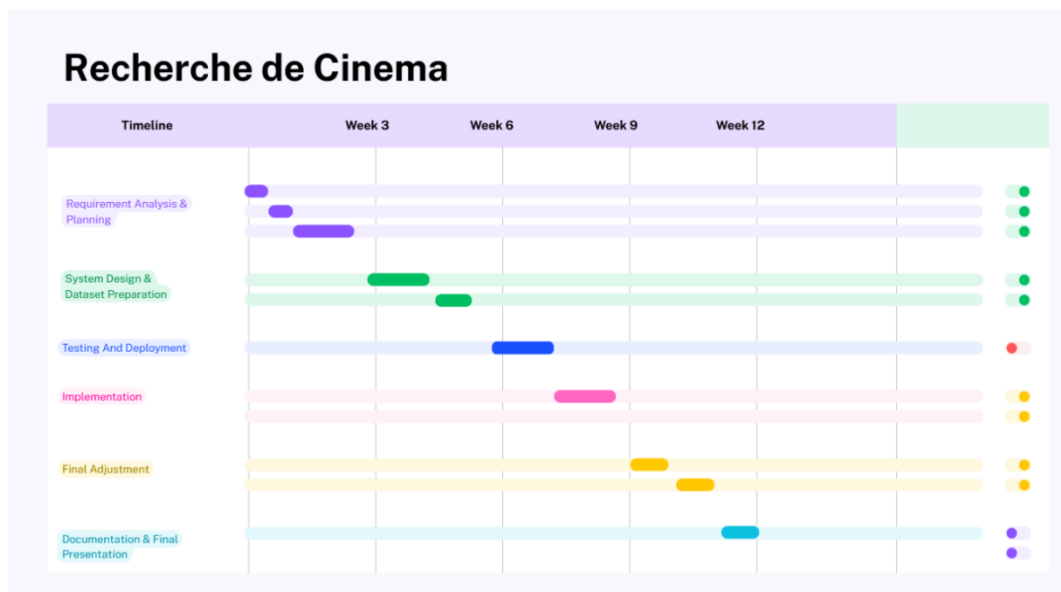


Figure 5: Gantt-Chart

5.1 Tasks

- Requirement Analysis and Planning
- System Design and Dataset Preparation
- Testing and Deployment
- Final Adjustment
- Documentation and Final Presentation

Chapter 6 References

React. (n.d.). *React documentation*. Retrieved from <https://react.dev/>

Node.js. (n.d.). *Node.js documentation*. Retrieved from <https://nodejs.org/en/docs/>

Express. (n.d.). *Express.js - Node.js web application framework*. Retrieved from <https://expressjs.com/>

Mozilla Developer Network. (n.d.). *Web Speech API*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

MongoDB, Inc. (n.d.). *MongoDB Documentation*. Retrieved from MongoDB Docs [MongoDB+1](#)

Tailwind Labs. (n.d.). *Tailwind CSS*. Retrieved from <https://tailwindcss.com/Tailwind CSS>

GeeksforGeeks. (2025, September 22). *Recommendation System in Python*. Retrieved from <https://www.geeksforgeeks.org/machine-learning/recommendation-system-in-python/>

