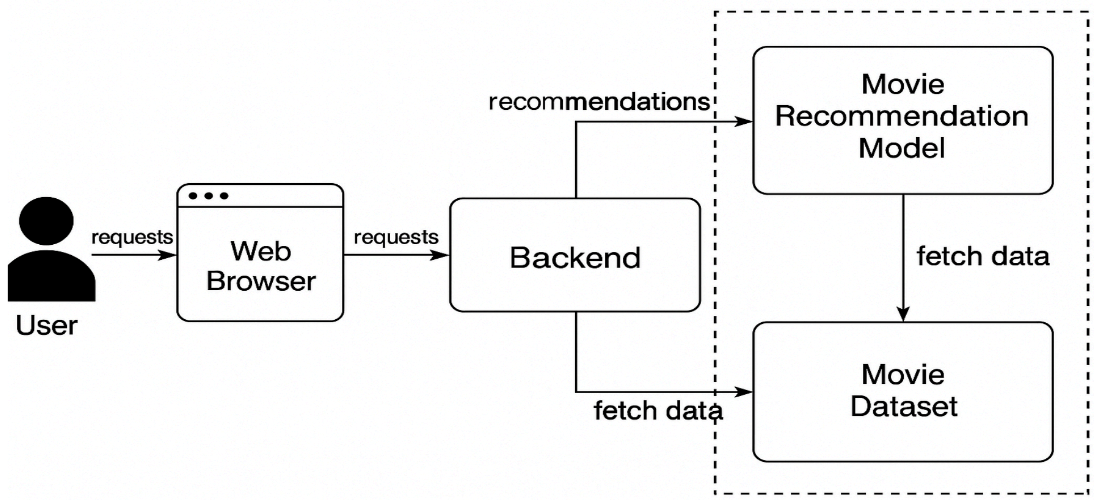
MOVIE RECOMMENDATION SYSTEM

1. Introduction

The Movie Recommendation System is a comprehensive web-based application designed to provide personalized movie suggestions to users based on their preferences, viewing habits, and interactive feedback. Developed as a full-stack project, it incorporates principles from Natural Language Processing (NLP), machine learning, and software engineering. The application's architecture is divided into three primary components: frontend (React.js), backend (FastAPI), and model integration (Python-based recommendation engine).

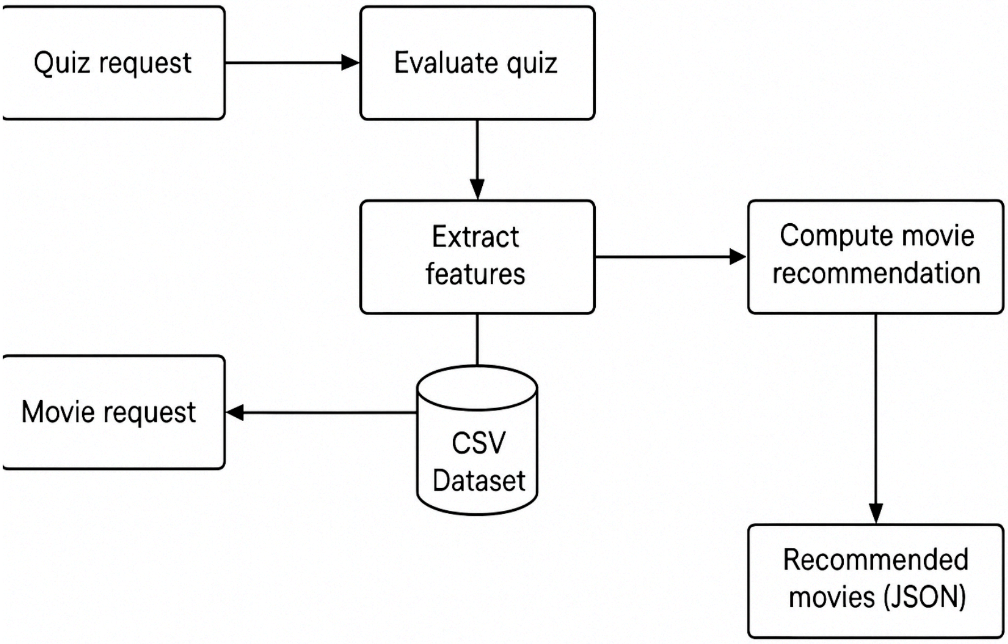


System Architecture System

2. System Architecture

The system adopts a modular and scalable architecture with distinct layers responsible for user interaction, data handling, and intelligent recommendation generation.

• Frontend: Built using React.js, it provides an interactive user interface for searching, exploring, and receiving personalized movie recommendations.
• Backend: Developed using FastAPI, it serves as a bridge between the frontend and the recommendation model. It manages API endpoints, processes user queries, and performs NLP-driven computations.
• Model: A Python-based recommendation engine that utilizes NLP techniques to analyze metadata and compute similarities between movies based on content attributes.



3. Recommendation Model (NLP-Based)

The core of the system's intelligence lies in its recommendation model, implemented in Python. The model uses metadata from 'data.csv' and processes it using NLP-based algorithms to identify patterns and similarities. Key steps include:

1. Data Preprocessing: Missing values in textual fields such as genres, keywords, tagline, cast, and director are replaced with blanks.
2. Feature Combination: Textual features are concatenated to form a composite description for each movie.
3. TF-IDF Vectorization: The model employs the Term Frequency-Inverse Document Frequency (TF-IDF) technique, a core NLP method that quantifies the significance of words relative to the dataset.
4. Cosine Similarity: Once vectorized, cosine similarity is computed between movie feature vectors to identify closely related titles.

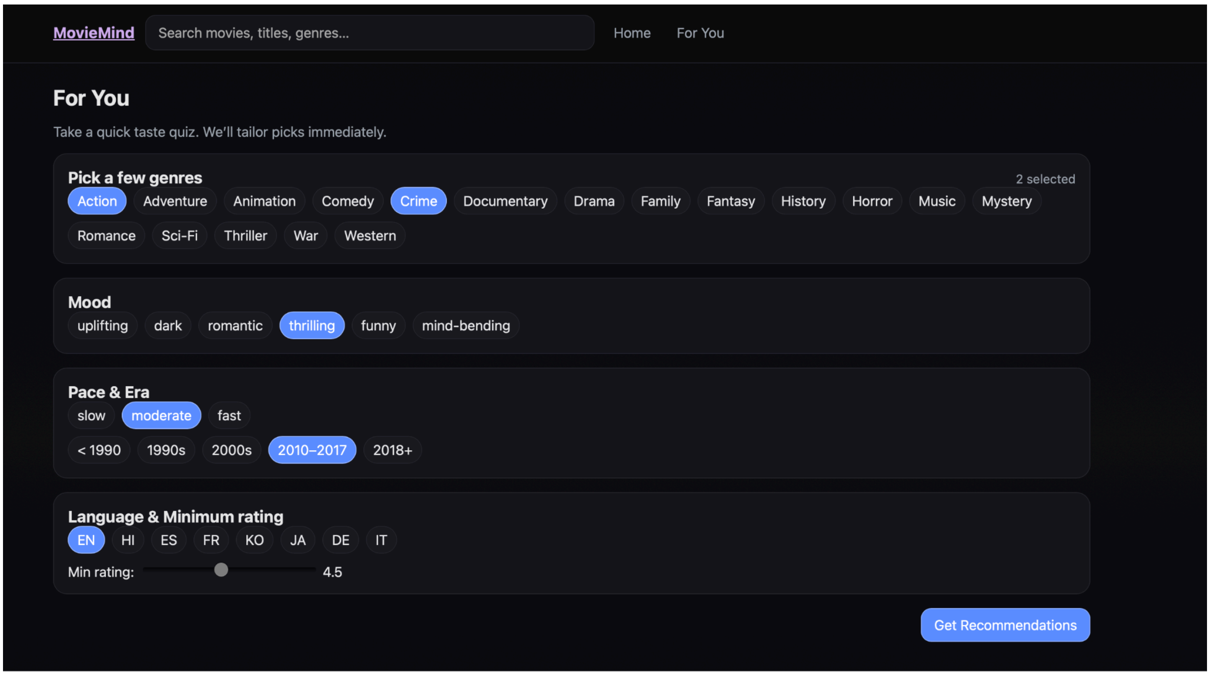
This combination of TF-IDF and cosine similarity allows the model to semantically relate movies even if they differ in genre representation or title wording.

4. Personalized Quiz-Based Recommendation ('For You' Page)

A major enhancement introduced in this system is the 'For You' page, designed to capture user preferences through a short quiz. This quiz dynamically evaluates the user's movie taste and mood, and provides curated recommendations using hybrid logic that merges NLP similarity with user-defined parameters.

The quiz includes attributes such as:
• Genres – e.g., Action, Drama, Romance
• Mood – e.g., Uplifting, Dark, Thrilling
• Pace – e.g., Slow, Moderate, Fast
• Era – e.g., Classic, 2000s, Recent
• Languages – e.g., English, Hindi, French
• Minimum Rating – Numerical threshold for filtering high-rated movies

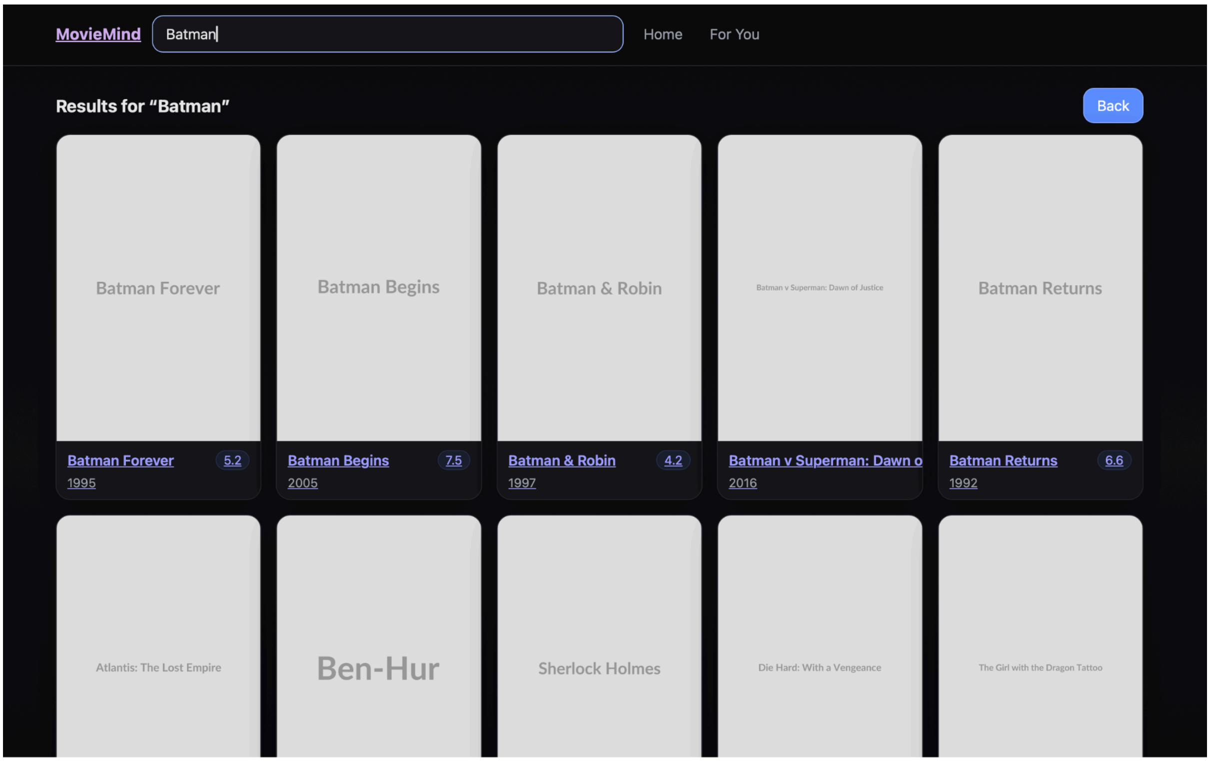
Once the quiz is submitted, the backend filters the dataset using the chosen parameters, constructs a weighted query based on mood and genre keywords, and retrieves results using the same TF-IDF vector space. This approach blends semantic relevance (from NLP) with rule-based filtering to achieve personalized output.



5. Frontend Design and User Interface

The frontend interface is built using React.js, designed with modern aesthetics and responsive CSS styling. The interface allows seamless navigation between pages and integrates real-time API communication using Axios.

Key pages include:
• Home Page – Displays trending and popular movies.
• For You Page – Hosts the personalized quiz and displays tailored recommendations.
• Movie Detail Page – Provides a summary, genres, and cast information for each title.



6. Role of NLP in the Application

Natural Language Processing (NLP) plays a pivotal role in this project. It transforms descriptive textual information about movies into structured numerical representations. The following NLP techniques are applied:

• TF-IDF Vectorization – Identifies the relative importance of words within the combined metadata fields.
• Cosine Similarity – Measures the semantic closeness between movie descriptions.
• Keyword Matching – Supports quiz-based recommendation by recognizing mood and genre-related terms.

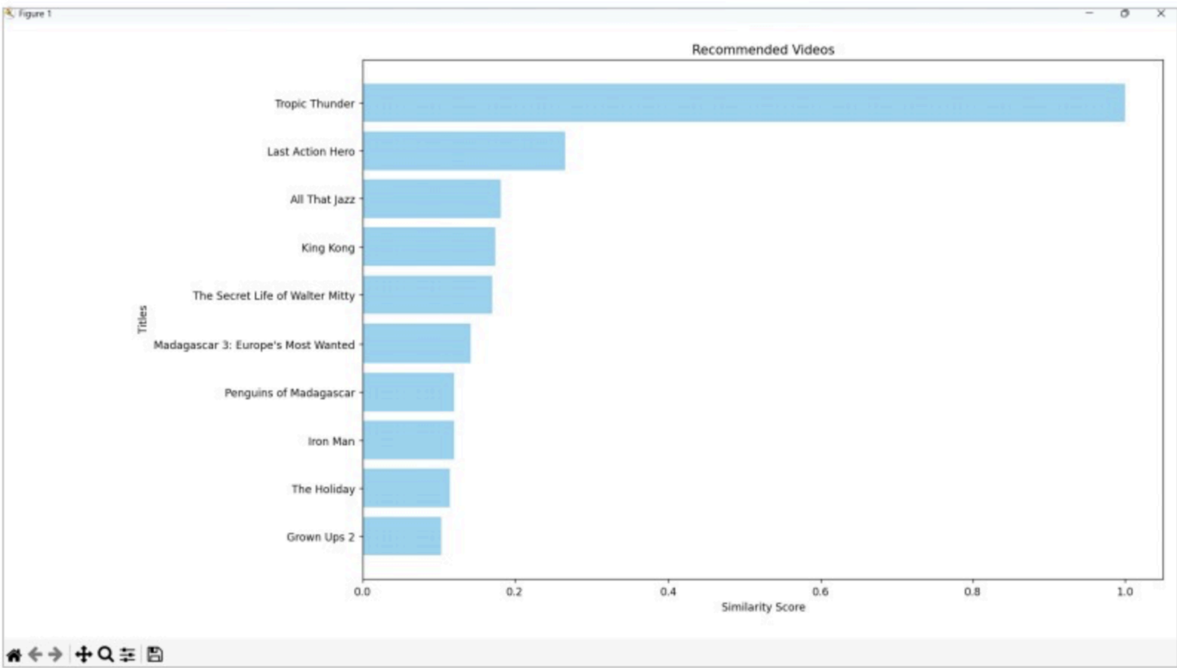
These techniques collectively enable the system to interpret movie characteristics beyond simple metadata, allowing more human-like recommendations that account for thematic and contextual nuances.

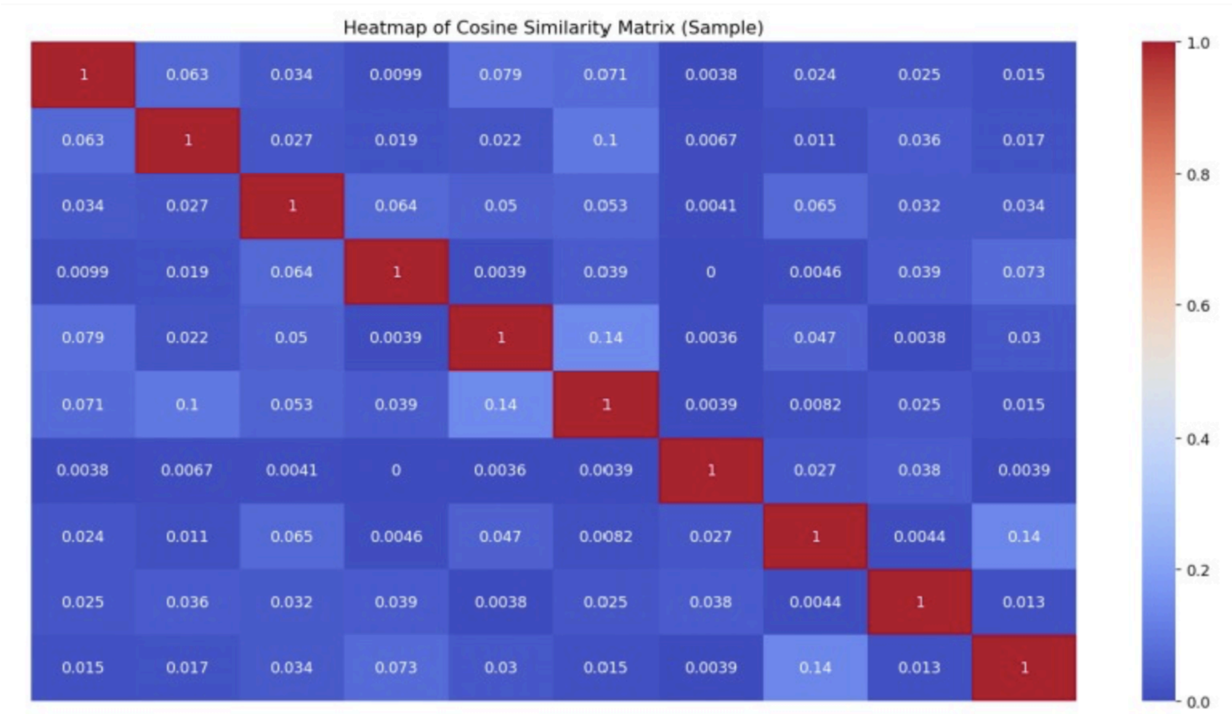
7. Results

The performance and functionality of the Movie Recommendation System were evaluated through a series of experiments and visual analyses. The results primarily demonstrate how effectively the NLP-based model captures semantic similarities between movies and produces accurate, context-aware recommendations.

To visualize the relationship between movies within the dataset, a cosine similarity heatmap was generated. This heatmap provides a graphical interpretation of the similarity matrix produced during model computation, highlighting clusters of movies that share comparable themes, genres, or cast members.

In addition to visualization, the terminal output showcases how the system retrieves the top recommended titles for a given movie query, listing them in descending order of similarity. This confirms that the TF-IDF and cosine similarity pipeline functions as expected and that user queries are accurately matched to relevant titles.





8. Conclusion

The Movie Recommendation System effectively combines web technologies and artificial intelligence to create a smart, user-friendly platform. Through the integration of NLP techniques, the system can analyze and compare movies based on textual metadata, providing results that align closely with user interests. The 'For You' quiz component introduces an element of interactivity and personalization, further enriching the user experience.

This project serves as a practical example of applying NLP and software engineering concepts to deliver real-world, data-driven solutions.

-Harsh Prajapati	(2201087)
-Nehal Jain	(2201129)
-Piyush Upadhyay	(2201143)
-Sai Arvind Karthik	(2201173)
-Sanchayan Saha	(2201175)