# Movie Recommendation System

## 1. Title Page / Header

Project Title: Semantic Movie Recommendation System

Student Participant's Name: Keshara Gunathilaka, University of Ruhuna.

Mentor's Name & Affiliation: Mr. Arshardh Ifthikar, Technical Lead, WSO2.

IEEE ELEVATE Program: IEEE ELEVATE 2025

Project Links:
- GitHub Repository: https://github.com/KesharaGunathilaka/Movie-Recommendation-System

## 2. Introduction

This project builds a movie recommendation system using a content-based approach. We use a Kaggle movie dataset with the features Title, Plot, Year, Director, and Cast. Each movie is represented by its own content structured fields and unstructured text. The system recommends films that are similar in content to a user's query or a given movie. We preprocess the dataset, normalize text, and derive vector representations from these fields to measure similarity and generate relevant suggestions.

The goal of this project was to build a recommendation system that can handle both title-based queries and natural language searches. This approach makes the system more interactive and closer to how users naturally search for movies.

The project is relevant to both academic and industry applications as recommender systems are critical in platforms like Netflix, YouTube, and Amazon. By using sentence transformers (BERT-based embeddings), the project demonstrates how NLP can enhance user personalization.

## 3. Project Execution & Methodology

**Planned milestones:**
1. Dataset preparation and preprocessing.
2. Baseline recommender using TF-IDF + cosine similarity.

3. Upgrade to semantic embeddings using sentence-transformers.
4. Development of Flask API backend.
5. Development of React Vite frontend with Tailwind.

**Tools, platforms, and resources:**
- Python (Pandas, NumPy, scikit-learn, SentenceTransformers, Flask)
- Frontend: React Vite + TailwindCSS
- Backend: Python Flask
- Version control: GitHub

**Adjustments made:**
First, TF-IDF was considered sufficient, but results for natural queries were weak. The project upgraded to semantic embeddings (all-mpnet-base-v2) which provided much stronger results.

## 4. Contributions & Teamwork

This was an individual project.
**Contributions include:**
- Designed preprocessing pipeline for movie dataset.
- Implemented both TF-IDF baseline and semantic embedding recommender.
- Built a Flask API to recommend endpoint.
- Designed and developed an attractive frontend using React Vite + TailwindCSS.

## 5. Mentor Engagement

- Number of meetings with mentor: 6
- Mode of communication: Google Meet

- Key feedback: Mentor emphasized improving recommendation quality when results were not satisfactory. He suggested applying feature engineering, like generating keywords from the movie plot and exploring better representations beyond plain TF-IDF. Mentor also recommended using a vectorized database approach for more accurate similarity matching.

- Impact: This guidance shaped the system design by adding plot-based keyword features and adopting semantic embeddings with a vector database. As a result, the recommender achieved more meaningful suggestions and could better handle natural language queries.

## 6. Key Outcomes & Learnings

**Major deliverables:**
- A fully working movie recommendation system.
- Flask backend exposing REST API.
- React Vite frontend with attractive UI.

**Technical Results:**
- Ability to recommend movies based on titles, actors, directors, genres, and descriptive natural language queries.
- Embedding model used: all-mpnet-base-v2

**Challenges & solutions:**
- Weak results with TF-IDF, then switched to sentence embeddings.

**Skills gained:**
- Technical: Handling large dataset, Data Preprocessing, NLP embeddings, Flask API design, React + Tailwind UI development.
- Professional: Project structuring, mentor collaboration, and problem-solving.

## 7. Conclusion & Future Work

The project successfully achieved its goal of building a semantic movie recommender with an engaging user interface. Compared to the initial TF-IDF approach, semantic embeddings greatly enhanced recommendation quality.

Future improvements could include:
- Adding hybrid recommender features (collaborative + content-based).
- User personalization based on history and ratings.

This project was an enriching experience, combining NLP, backend, and frontend development, and aligning with the objectives of the IEEE ELEVATE program.

## 8. References / Appendices

- Hugging Face SentenceTransformers: https://www.sbert.net/
- Dataset - https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots