**MovieMeter: Movie Recommendation Website**

**Team Members:** Hunter Smith, Guillermo Flores, Darien Walker

# Project Idea

MovieMeter – An AI/ML-powered movie recommendation system that suggests personalized movies to users.

The system will consider:

* Initial User Questions
* Feedback on previously watched movies
* Ongoing Interaction

# Project Goal

The goal of this project is to design and implement an AI system that addresses a real-world challenge by combining intelligent decision-making with user-centered design. Specifically, the project aims to:

* **Clearly define the problem space**: Many streaming platforms overwhelm users with options, making it difficult to find movies that truly match their tastes.
* **Propose a structured AI solution**: A movie recommendation system that adapts to user preferences through interaction and feedback.
* **Demonstrate the system** through a prototype website that simulates real-world use cases.

# Why This Project?

* **Relevance** – Personalized recommendations are central to platforms like Netflix, Hulu, and YouTube. Building this system provides practical insight into real-world AI applications.
* **Feasibility** – A recommendation system can be reasonably designed and prototyped within the semester using available datasets (e.g., MovieLens, TMDB API).
* **Educational Value** – This project enhances our understanding of AI through problem definition, model design, evaluation, and tradeoff analysis between accuracy, scalability, and usability.

# How (Methodology & Approach)

We will design the AI system using a structured component breakdown:

## Problem Definition

* + Scope: Provide personalized movie recommendations via a website interact
  + User needs: Users want relevant, accurate movie suggestions without wasting time searching.
  + Constraints: need for complete movie dataset, which is continuously expanding, time to implement project.

## System Design (Conceptual)

* + Input Data:
* Movie metadata: Name, Genres, Ratings, Cast, Year, Etc.
* User data: User IDs and User ratings or interactions (likes, watches, skips), statistics
  + Processing/Model
* Hybrid Approach:

Content-Based Filtering: Matches movies based on genres, actors, and keywords.

Collaborative Filtering: Identifies patterns from similar users’ preferences.

* Optional: Incorporate NLP for analyzing user-written feedback.
  + Output: What kind of decision, prediction, or recommendation will be provided?
* Personalized movie recommendations ranked by relevance
* Adaptive suggestions that change as user feedback accumulates.

## Example System User Flow

1. User interacts with the website → answers initial questions or provides feedback.
2. Input data is sent to the back-end system.
3. AI models (hybrid recommendation engine) process the input.
4. The system outputs personalized recommendations.
5. User actions (watch, like/dislike) feed back into the system to improve future suggestions.

## Implementation Plan

* + Data Collection: Use the MovieLens dataset or TMDB API for movie metadata.
  + Preprocessing: Clean and format data for model input (genres, ratings, user IDs).
  + Model Training: Split data into training and test sets, Train the model(s), Tune hyperparameters. Implement baseline collaborative filtering + content-based recommendation.
  + Website Prototype: Develop a simple user interface to display questions, recommendations, and feedback options.

## Evaluation Plan

* + **Success Criteria**:
* Accuracy of recommendations
* User satisfaction based on feedback (in test scenarios)
* Usability of the interact
  + Testing Plan: Simulate multiple user profiles with different preferences and test how well recommendations adapt.
  + Plan for testing with scenarios or case studies.
  + Use metrics like RMSE, MAE (for rating)
  + Precision, Recall, F1-score, MAP, NDCG (for ranking/recommendations)

# Tools & Technologies

* VS Code
* Git / GitHub
* Python
* HTML/CSS/JavaScript
* TMDb API / MovieLens dataset
* Jupyter Notebook

# Work Distribution

Darien Walker will be responsible for conducting the research and background study. This includes reviewing existing movie recommendation systems, comparing popular algorithms such as collaborative filtering, content-based filtering, and hybrid approaches, and identifying strengths and weaknesses in each. Darien will also investigate available datasets and APIs such as MovieLens and TMDb to determine which resources best fit the project’s scope and timeline. His research will form the foundation of the system’s design and ensure that the team selects the most effective methods for implementation.

Hunter Smith will take primary responsibility for drafting and compiling the final project report. This will involve integrating contributions from all team members into a cohesive, well-structured document that clearly explains the project’s goals, methodology, system architecture, implementation process, evaluation metrics, and results. Hunter will also edit the report for clarity, technical accuracy, and formatting to meet academic standards.

Guillermo Flores will lead the creation of the final presentation slides. He will design a professional and visually engaging slide deck that summarizes the key points of the report, including project goals, methodology, system design, results, and lessons learned. Guillermo will ensure that the slides highlight technical details in a way that is accessible to the audience, using visuals such as diagrams, flowcharts, and graphs.

In addition to their individual responsibilities, the entire team will collaborate on the system design and documentation. Together, they will outline the architecture of the recommendation system, describe the flow of data between components, and create diagrams that show how user input is processed to generate personalized recommendations.

All team members will also share responsibility for the implementation of the prototype website and AI recommendation model. This includes coding the back-end system in Python, integrating the recommendation algorithms, building the front-end user interface, and connecting the system to the chosen dataset or API. Team members will divide coding tasks based on their strengths (Ex: Hunter primarily focusing front0end design) but will work together to ensure seamless integration across components.

Finally, testing and presentation preparation will be carried out collaboratively. Each member will help test the system using different user scenarios to evaluate the accuracy and usability of recommendations. For the presentation, all three members will rehearse their roles to ensure smooth delivery, with each person presenting the parts of the project that align with their contributions.

# Citations

“Non-Commercial, Personalized Movie Recommendations.” *MovieLens*, [movielens.org](https://movielens.org/)/. Accessed 26 Sept. 2025.

“Getting Started.” *The Movie Database (TMDB)*, [developer.themoviedb.org/reference/intro/getting-started](https://developer.themoviedb.org/reference/intro/getting-started). Accessed 26 Sept. 2025.