**Project Title: Movie Recommendation System**

**Objective:**

* The goal of this project is to build an intelligent movie recommender system that suggests similar movies based on user preferences.
* By leveraging NLP techniques and machine learning, the system helps users discover movies that match their interests.

**Working:**

* Loaded and cleaned the dataset by handling missing values, duplicates, and irrelevant columns.
* Combined key features like *overview, genres, keywords, cast,* and *crew* into a single text column.
* Applied **text normalization** — lowercasing, removing special characters, and tokenization.
* Used **stemming** to reduce words to their root form for better text uniformity.
* Converted text data into numerical form using Count Vectorization to measure the importance of words across movie descriptions.
* Computed cosine similarity between movie vectors to measure how closely movies relate to each other.
* Designed a recommendation function that fetches the top similar movies based on similarity scores.
* Built an interactive user interface with Streamlit for easy user interaction.
* Added dropdown search to select a movie and display top recommendations.
* Displayed movie posters, overviews, and details in visually appealing expandable cards.
* Integrated a background theme and navigation bar for an engaging user experience.

**Outcome:**

* Developed a fully functional movie recommender web application capable of displaying personalized movie suggestions.
* Improved user engagement by including visual elements like posters and overviews.
* Enhanced the recommendation accuracy using Count Vectorizer + cosine similarity approach.

**Skills Demonstrated:**

* Python Programming
* Data Preprocessing & Cleaning
* Natural Language Processing (NLP)
* Machine Learning (Cosine Similarity, TF-IDF)
* Streamlit Web App Development
* API Integration (Fetching posters from TMDb API)
* Data Visualization and User Interface Design

**Tools & Technologies Used:**

* **Languages:** Python
* **Libraries:** Pandas, NumPy, Scikit-learn, NLTK, Streamlit, Requests
* **Techniques:** Count Vectorization, Cosine Similarity
* **Visualization:** Streamlit Components and Layouts
* **Platform:** Streamlit Cloud / Localhost