**🧾 🎯 Project Title: MOVIE RECOMMENDATION WEB APP  
📅 Project Timeline:** August 2023 – October 2023  
🎥 YouTube Demo: [https://youtu.be/YgA\_rae6ojw](https://youtu.be/YgA_rae6ojw?utm_source=chatgpt.com)  
📦 GitHub Source Code: [https://github.com/IvanSicaja/2023.08.01\_GitHub\_Movie-Recommendation-Web-App](https://github.com/IvanSicaja/2023.08.01_GitHub_Movie-Recommendation-Web-App?utm_source=chatgpt.com)  
----------------------------------------------------------------------------------------------------------------

🏷️ My Personal Profiles: ⬇︎  
🎥 Video Portfolio: To be added  
📦 GitHub Profile: <https://github.com/IvanSicaja>  
🔗 LinkedIn: <https://www.linkedin.com/in/ivan-si%C4%8Daja-832682222>  
🎥 YouTube: <https://www.youtube.com/@ivan_sicaja>  
----------------------------------------------------------------------------------------------------------------

### 📚🔍 Project description: ⬇︎⬇︎⬇︎

### 💡 App Purpose

The purpose is to build a **recommendation app** that applies **content-based similarity search** in this case **movie similarity search.**

### 🧠 How It Works

From the **movie database** which consists of columns like **movie description, title, genre, vote rating**, the company has created a **subset** of the most important columns in the dataset.  
From the dataset subset the **bag of words** is created with the usage of the **scikit-learn Python module**. Every relevant word is **tokenized** and **lemmatized**. **Nonrelevant words** (e.g. „the“ etc..) and **stop characters** are excluded from the bag of words. As the final dataset a **.csv** and **.pkl file** are created. Every row of the **.csv file** is represented by the **vector** of the **word tokens** for every **movie summary subset**.

As the **recommendation function** uses the **cosine similarity principle** which makes the calculation efficient even if it is difficult to visualize the **1000+ dimensional vector space**, the math works. After the **most similar movies** from the selected movie info are forwarded to the **frontend** with corresponding **movie posters**.  
**Python** is used for the **frontend** and for the **backend**. The application uses **cosine similarity** on **tokenized and lemmatized movie descriptions** to recommend **five similar movies**. It retrieves **movie posters dynamically** via the **The Movie Database (TMDB) API**, and the **Streamlit frontend** provides a simple **dropdown** for movie selection and displays **recommendations** with **posters in columns**.

### ⚠️ Note

The project is based on a **subset** of the **original dataset** and demonstrates the **functionality** of a **recommendation system**. **Real-world applications** would require **larger datasets** and **optimization** for **scalability**.

### 🔧 Tech Stack

**Python, Natural Language Processing (NLP), Pandas, scikit-learn, Streamlit, Requests, Pickle, Linux, API usage**

### 📸 Project Snapshot

Not available.

### 🎥 Video Demonstration

Not available.

### 📣 Hashtags Section

**# #MovieRecommendation #Python #Streamlit #NLP #MachineLearning #CosineSimilarity #Pandas #ScikitLearn #MovieApp #RecommendationSystem #DataScience #AI #Frontend #Backend #APIs #TMDB #GitHub #FullStackDevelopment #Linux #ArtificialIntelligence #MLProject**