Source Code :

import streamlit as st

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

# Load Data

movies = pd.read\_csv("movies.csv")

# TF-IDF processing

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(movies['Genres'])

cosine\_sim = cosine\_similarity(tfidf\_matrix, tfidf\_matrix)

# Title lookup

movie\_indices = pd.Series(movies.index, index=movies['Title'])

# Recommend function

def get\_recommendations(title, top\_n=10):

idx = movie\_indices[title]

sim\_scores = list(enumerate(cosine\_sim[idx]))

sim\_scores = sorted(sim\_scores, key=lambda x: x[1], reverse=True)

sim\_scores = sim\_scores[1:top\_n + 1]

movie\_indices\_top = [i[0] for i in sim\_scores]

return movies['Title'].iloc[movie\_indices\_top]

# --- Streamlit UI ---

st.set\_page\_config(page\_title="ˆ’z`˘˜蘯F~—蘹MovieMatch AI", layout="centered")

st.title("ˆ˘z`F~˜’蘹—蘯 MovieMatch AI")

st.markdown("""

Welcome to \*\*MovieMatch AI\*\* — your personal movie recommender system

powered by artificial intelligence!

Enter a movie title you like, and we'll suggest similar ones just for you.

""")

# Movie input (instead of selectbox)

selected\_movie = st.text\_input("Enter the name of a movie you like:")

if st.button(".\_蘇蘄Recommend"):

if selected\_movie:

st.subheader("You might also enjoy:")

try:

recommendations = get\_recommendations(selected\_movie)

for i, title in enumerate(recommendations, 1):

st.write(f"{i}. {title}")

except KeyError:

st.error("Sorry, we couldn't find the movie. Please try again with a different

title.")

else:

st.error("Please enter a movie name.")