2. MOVIE TITLE RECOMMENDATION SYSTEM

EX.N0 : 2	
	DEVELOP A MOVIE TITLE RECOMMENDATION SYSTEM
DATE : /0 /2025	

AIM:

To write a program to Develop a movie Title recommendation system

ALGORITHM:

Step 1:   Start

Step 2: Import necessary NLP libraries.

Step 3: Load the movie rating dataset.

Step 4: Create a pivot table: users vs. movies with ratings.

Step 5: Compute cosine similarity between movie vectors.

Step 6: Define a function to get top similar movies.

Step 7: Get recommendations for a given movie.
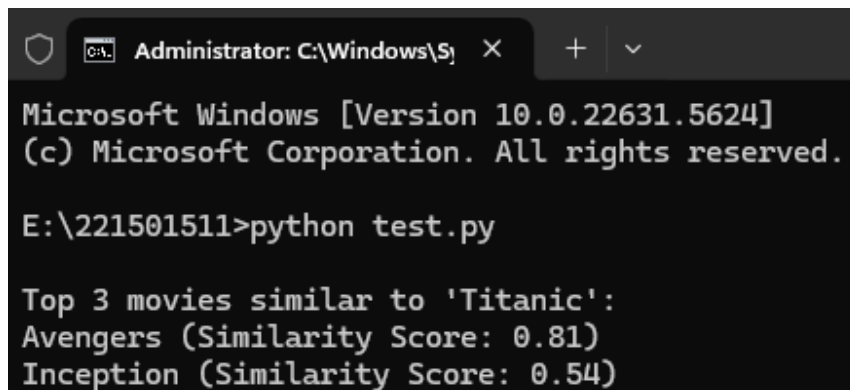
PROGRAM:

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import StandardScaler
data = {
'UserID': [1, 1, 1, 2, 2, 3, 3, 4, 4],
'MovieTitle': [
'Avengers', 'Inception', 'Titanic',
'Avengers', 'Titanic',
'Inception', 'Titanic',
'Avengers', 'Inception' ],
```

'Rating': [5, 4, 3, 4, 4, 5, 2, 3, 4] }

df = pd.DataFrame(data)

user_movie_matrix = df.pivot_table(index='UserID', columns='MovieTitle', values='Rating').fillna(0)

movie_matrix = user_movie_matrix.T

similarity_matrix = cosine_similarity(movie_matrix)

similarity_df = pd.DataFrame(similarity_matrix, index=movie_matrix.index, columns=movie_matrix.index)

def recommend_movies(movie_name, top_n=3):

print(f"\nTop {top_n} movies similar to '{movie_name}':")

if movie_name not in similarity_df.columns:

print("Movie not found in the dataset.")

return

recommendations = similarity_df[movie_name].sort_values(ascending=False)[1:top_n+1]

for title, score in recommendations.items():

print(f"{title} (Similarity Score: {score:.2f})")

recommend_movies("Titanic")

OUTPUT:

RESULT:

Thus a program to develop a movie Title recommendation system has been executed successfully.