K.RAMAKRISHNAN
COLLEGE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
SAMAYAPURAM, TRICHY-621 112

Practical Record Note

Name : RAGHUL S

Register Number : 2303811710421126

Subject code/name : Laboratory

Programme :

K.RAMAKRISHNAN

COLLEGE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

SAMAYAPURAM, TRICHY-621 112

Certified that this is a bonafide record of work done by **RAGHUL S** of _____ Semester in **Python Programming - I Year - II Sem - Project Module** Laboratory during the academic year 2023-2024

Staff Incharge **Head of the Department**

Submitted for the Practical exam held on:

Internal Examiner
Date:

External Examiner
Date:

Aim:

Project Module.

Program:

CTP28132.py

```
# Sample movie database
movies = [
    {"title": "The Shawshank Redemption", "genre": "Drama", "year": 1994,
"rating": 9.3},
    {"title": "The Godfather", "genre": "Crime", "year": 1972, "rating": 9.2},
    {"title": "The Dark Knight", "genre": "Action", "year": 2008, "rating": 9.0},
    {"title": "Pulp Fiction", "genre": "Crime", "year": 1994, "rating": 8.9},
    {"title": "Forrest Gump", "genre": "Drama", "year": 1994, "rating": 8.8},
    {"title": "Inception", "genre": "Action", "year": 2010, "rating": 8.8},
    {"title": "The Matrix", "genre": "Sci-Fi", "year": 1999, "rating": 8.7},
    {"title": "The Lord of the Rings: The Return of the King", "genre":
"Fantasy", "year": 2003, "rating": 8.9},
    {"title": "Fight Club", "genre": "Drama", "year": 1999, "rating": 8.8},
    {"title": "Interstellar", "genre": "Sci-Fi", "year": 2014, "rating": 8.6},
]

def get_user_preference():
    print("Welcome to the Movie Recommendation System!")
    genre = input("Enter your preferred genre (Drama, Crime, Action, Sci-Fi,
Fantasy): ").strip()
    min_year = int(input("Enter the minimum year of release: ").strip())
    min_rating = float(input("Enter the minimum rating (0.0 to 10.0): ").strip())

    return genre, min_year, min_rating

def recommend_movie(movies, genre, min_year, min_rating):
    filtered_movies = [
        movie for movie in movies
        if movie["genre"].lower() == genre.lower() and movie["year"] >= min_year
and movie["rating"] >= min_rating
    ]

    if not filtered_movies:
        return "Sorry, no movies match your preferences."

    # Sort by rating and then by year
    filtered_movies.sort(key=lambda x: (-x["rating"], -x["year"]))

    # Recommend the top movie
    top_movie = filtered_movies[0]
    return f"We recommend you watch '{top_movie['title']}' ({top_movie['year']})
- Rating: {top_movie['rating']}"

# Main program
genre, min_year, min_rating = get_user_preference()
recommendation = recommend_movie(movies, genre, min_year, min_rating)
print(recommendation)
```


Output:

Test case - 1

User Output
Hello World

```
Hello World
```

Result:

Thus the above program is executed successfully and the output has been verified