Movie Recommendation by Prabhat

Importing the Library

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [7]:
```
#Loading the Dataset
rating  = pd.read_csv('ml-100k/u.data',sep='\t', names=['user_id','item_id',
print(rating)
```

```
       user_id  item_id  rating  timestamp
0          196      242       3  881250949
1          186      302       3  891717742
2           22      377       1  878887116
3          244       51       2  880606923
4          166      346       1  886397596
...        ...      ...     ...        ...
99995      880      476       3  880175444
99996      716      204       5  879795543
99997      276     1090       1  874795795
99998       13      225       2  882399156
99999       12      203       3  879959583

[100000 rows x 4 columns]
```

In [25]:
```
movies = pd.read_csv('ml-100k/u.item',sep='|',header=None, encoding='latin-1
                                'unknown', 'Action', 'Adventure', 'Animation',
                                'Crime', 'Documentary', 'Drama', 'Fantasy', 'Fi
                                'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thr

# print(movies)
```

```
      item_id                                     title release_date  \
0           1                          Toy Story (1995)  01-Jan-1995
1           2                          GoldenEye (1995)  01-Jan-1995
2           3                         Four Rooms (1995)  01-Jan-1995
3           4                         Get Shorty (1995)  01-Jan-1995
4           5                            Copycat (1995)  01-Jan-1995
...       ...                                       ...          ...
1677     1678                       Mat' i syn (1997)  06-Feb-1998
1678     1679                        B. Monkey (1998)  06-Feb-1998
1679     1680                    Sliding Doors (1998)  01-Jan-1998
1680     1681                     You So Crazy (1994)  01-Jan-1994
1681     1682  Scream of Stone (Schrei aus Stein) (1991)  08-Mar-1996

      video_release_date                                         IMDb_URL
\
0                    NaN  http://us.imdb.com/M/title-exact?Toy%20Story%2...
1                    NaN  http://us.imdb.com/M/title-exact?GoldenEye%20(...
2                    NaN  http://us.imdb.com/M/title-exact?Four%20Rooms%...
3                    NaN  http://us.imdb.com/M/title-exact?Get%20Shorty%...
4                    NaN  http://us.imdb.com/M/title-exact?Copycat%20(1995)
...                  ...                                               ...
1677                 NaN  http://us.imdb.com/M/title-exact?Mat%27+i+syn+...
1678                 NaN  http://us.imdb.com/M/title-exact?B%2E+Monkey+(...
1679                 NaN      http://us.imdb.com/Title?Sliding+Doors+(1998)
1680                 NaN  http://us.imdb.com/M/title-exact?You%20So%20Cr...
1681                 NaN  http://us.imdb.com/M/title-exact?Schrei%20aus%...

      unknown  Action  Adventure  Animation  Children  ...  Fantasy  \
0           0       0          0          1         1  ...        0
1           0       1          1          0         0  ...        0
2           0       0          0          0         0  ...        0
3           0       1          0          0         0  ...        0
4           0       0          0          0         0  ...        0
...       ...     ...        ...        ...       ...  ...      ...
1677        0       0          0          0         0  ...        0
1678        0       0          0          0         0  ...        0
1679        0       0          0          0         0  ...        0
1680        0       0          0          0         0  ...        0
1681        0       0          0          0         0  ...        0

      Film-Noir  Horror  Musical  Mystery  Romance  Sci-Fi  Thriller  War  \
0             0       0        0        0        0       0         0    0
1             0       0        0        0        0       0         1    0
2             0       0        0        0        0       0         1    0
3             0       0        0        0        0       0         0    0
4             0       0        0        0        0       0         1    0
...         ...     ...      ...      ...      ...     ...       ...  ...
1677          0       0        0        0        0       0         0    0
1678          0       0        0        0        1       0         1    0
1679          0       0        0        0        1       0         0    0
1680          0       0        0        0        0       0         0    0
1681          0       0        0        0        0       0         0    0

      Western
0           0
1           0
```

```
2           0
3           0
4           0
...       ...
1677        0
1678        0
1679        0
1680        0
1681        0

[1682 rows x 24 columns]
```

In [19]:
```
users = pd.read_csv('ml-100k/u.user',sep='|',names = ['user_id','age','gende
print(users)
```

```
     user_id  age gender     occupation zip-code
0          1   24      M     technician    85711
1          2   53      F          other    94043
2          3   23      M         writer    32067
3          4   24      M     technician    43537
4          5   33      F          other    15213
..       ...  ...    ...            ...      ...
938      939   26      F        student    33319
939      940   32      M  administrator    02215
940      941   20      M        student    97229
941      942   48      F      librarian    78209
942      943   22      M        student    77841

[943 rows x 5 columns]
```

In [31]:
```
# Merging the movies with rating
merged = pd.merge(rating,movies[['item_id','title']],on='item_id')
print(merged)
```

```
       user_id  item_id  rating  timestamp                         title
0          196      242       3  881250949                  Kolya (1996)
1          186      302       3  891717742      L.A. Confidential (1997)
2           22      377       1  878887116           Heavyweights (1994)
3          244       51       2  880606923    Legends of the Fall (1994)
4          166      346       1  886397596           Jackie Brown (1997)
...        ...      ...     ...        ...                           ...
99995      880      476       3  880175444  First Wives Club, The (1996)
99996      716      204       5  879795543     Back to the Future (1985)
99997      276     1090       1  874795795                 Sliver (1993)
99998       13      225       2  882399156         101 Dalmatians (1996)
99999       12      203       3  879959583              Unforgiven (1992)

[100000 rows x 5 columns]
```

In [57]:
```
## Taking the input from the user for ratings
def get_user_rating():
    print("Rate the movie:\n")
    user_ratings = {}
    while True:
        movie_title = input("Enter the movie title or type 'done': ").strip(
        if movie_title.lower()=='done':
            break
```

```
        ratings = input(f"Rate: {movie_title} on a scale of 1 to 5").strip()
        user_ratings[movie_title] = ratings
    return user_ratings
user_ratings = get_user_rating()
```

Rate the movie:

In [59]:
```
## similiarity calculation
from sklearn.metrics.pairwise import cosine_similarity

user_movie_matrix = merged.pivot_table(index = 'user_id',columns = 'title',v
similiarity = cosine_similarity(user_movie_matrix)
similiarity_df = pd.DataFrame(similiarity,index=user_movie_matrix.index,colu
print(similiarity_df.head(7))
```

```
user_id       1         2         3         4         5         6         7
\
user_id
1        1.000000  0.168937  0.048388  0.064561  0.379670  0.429682  0.44309
7
2        0.168937  1.000000  0.113393  0.179694  0.073623  0.242106  0.10860
4
3        0.048388  0.113393  1.000000  0.349781  0.021592  0.074018  0.06742
3
4        0.064561  0.179694  0.349781  1.000000  0.031804  0.068431  0.09150
7
5        0.379670  0.073623  0.021592  0.031804  1.000000  0.238636  0.37473
3
6        0.429682  0.242106  0.074018  0.068431  0.238636  1.000000  0.49352
9
7        0.443097  0.108604  0.067423  0.091507  0.374733  0.493529  1.00000
0

user_id       8         9         10   ...       934       935       936  \
user_id                                ...
1        0.320079  0.078385  0.377733  ...  0.372213  0.119860  0.269860
2        0.104257  0.162470  0.161273  ...  0.147095  0.310661  0.363328
3        0.084419  0.062039  0.066217  ...  0.033885  0.043453  0.167140
4        0.188060  0.101284  0.060859  ...  0.054615  0.036784  0.133619
5        0.248930  0.056847  0.201427  ...  0.340183  0.080580  0.095284
6        0.202514  0.184997  0.554851  ...  0.384703  0.112464  0.187093
7        0.285815  0.146092  0.488501  ...  0.459442  0.114526  0.113190

user_id       937       938       939       940       941       942       94
3
user_id
1        0.193343  0.197949  0.118722  0.315064  0.149086  0.181612  0.39943
2
2        0.410725  0.322713  0.231096  0.228793  0.162911  0.175273  0.10673
2
3        0.071288  0.126278  0.026758  0.164539  0.102899  0.136757  0.02699
0
4        0.196561  0.146058  0.030202  0.196858  0.152041  0.171538  0.05875
2
5        0.081053  0.148607  0.071612  0.239955  0.139595  0.153799  0.31394
1
6        0.220179  0.138685  0.112729  0.354454  0.145268  0.312264  0.27761
2
7        0.120105  0.153818  0.104394  0.330926  0.060175  0.285273  0.39556
0

[7 rows x 943 columns]
```

Recommend Movie

In [61]:
```
# Get the most similiar users
user_id = 1 # lets take this as example
similiar_user = similiarity_df[user_id].sort_values(ascending=False).iloc[1:
# Get the top-rated movie
recommendations = user_movie_matrix.loc[similiar_user].mean(axis=0).sort_val
```

```
# Filter the movie ,the user has already rated
recommendations  = recommendations[~recommendations.index.isin(user_ratings.
print(recommendations.head(5))
```

```
title
Star Wars (1977)                  5.0
Raiders of the Lost Ark (1981)    5.0
Aliens (1986)                     4.8
Empire Strikes Back, The (1980)   4.8
Citizen Kane (1941)               4.8
dtype: float64
```

In [67]:
```
# Visuals
top_movie = recommendations.head(7)
plt.figure(figsize=(8,6))
top_movie.plot(kind='bar',color='skyblue')
plt.title(f"Top five Movie Recommendation for user {user_id}")
plt.xlabel("Movie Titles")
plt.ylabel("Predicted Ratings")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```