```
import pandas as pd
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import re
```

```
df = pd.read_csv("/content/myntra.csv")
```

```
df
```

	ID	Name	Price	MRP	Rating	Total	Discount	Seller
0	1	Men Solid Oversized Cotton	532	1299.0	4.1	5300	59	Difference of Opinion
1	2	Men Cotton Pure Cotton T-shirt	274	499.0	4.2	25400	45	Roadster
2	3	Women Pure Cotton T-shirt	551	1199.0	4.5	3400	54	DILLINGER
3	5	Printed Round Neck Pure Cotton T-shirt	494	899.0	4.2	3800	45	Roadster
4	6	Boys Pack of 5 T-shirt	699	4995.0	4.1	1200	86	HELLCAT
...	...	...	...	...	...	...	...	...
20243	20246	Boys Slim Fit Jeans	714	1099.0	4.4	139	35	TALES & STORIES
20244	20247	Floral Linen Cotton Shirt	719	1799.0	4.3	9	60	U.S. Polo Assn. Kids
20245	20248	Infants Pack Of 3 T-shirts	1259	1399.0	0.0	0	10	Ed-a-Mamma Baby
20246	20249	Boys Casual Cotton Shirt	769	1099.0	4.5	12	30	Ed-a-Mamma
20247	20250	Boys Batman Printed T-shirt	314	899.0	3.8	52	65	YK Justice League

20248 rows × 8 columns

```
# Data Preprocessing
# Handling missing values
print("\nMissing values before cleaning:")
print(df.isnull().sum())
# Drop rows with missing values
df.dropna(inplace=True)
```

```
Missing values before cleaning:
ID          0
Name        0
Price       0
MRP         0
Rating      0
Total       0
Discount    0
Seller      0
dtype: int64
```

```
# Clean the data
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = text.lower()
    return text
# Apply cleaning function to the 'name' column
df['Name'] = df['Name'].apply(clean_text)
# Display the first few rows of the cleaned dataframe
print("\nCleaned data:")
print(df.head())
```

```
Cleaned data:
   ID                                Name  Price     MRP  Rating  Total  \
0   1             men solid oversized cotton    532  1299.0     4.1   5300
1   2          men cotton pure cotton tshirt    274   499.0     4.2  25400
2   3               women pure cotton tshirt    551  1199.0     4.5   3400
3   5  printed round neck pure cotton tshirt    494   899.0     4.2   3800
4   6                   boys pack of  tshirt    699  4995.0     4.1   1200

   Discount                Seller
0        59  Difference of Opinion
1        45               Roadster
2        54              DILLINGER
3        45               Roadster
4        86                HELLCAT
```

```
# Feature Engineering
# Convert text features into numerical representations
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(df['Name'])

# Compute Item Similarity
item_similarity = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

```
# Recommendation Generation
def get_similar_items(item_idx, top_n=5):
    sim_scores = list(enumerate(item_similarity[item_idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    top_similar_items = sim_scores[1:top_n+1]  # Exclude the item itself
    return top_similar_items
```

```
# Example: Recommend similar items for the item at index 0
print('item:')
print(df.iloc[0]['Name'])
similar_items = get_similar_items(0)
for idx, score in similar_items:
    print(f"Similar Item: {df.iloc[idx]['Name']} (Similarity Score: {score})")
```

```
item:
men solid oversized cotton
Similar Item: men solid oversized cotton (Similarity Score: 1.0)
Similar Item: men solid oversized cotton (Similarity Score: 1.0)
Similar Item: cotton oversized tshirt (Similarity Score: 0.7597237790035756)
Similar Item: cotton oversized tshirt (Similarity Score: 0.7597237790035756)
Similar Item: cotton oversized tshirt (Similarity Score: 0.7597237790035756)
```

```
Start coding or generate with AI.
```