Future Interns Task 3

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [9]:
```

df = pd.read_csv('student_feedback.csv')
```

In [10]:
```
df
```

Out[10]:

	S/N	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	recommen ba rel
0	1	340	5	2	7	6	9	2	1	
1	2	253	6	5	8	6	2	1	2	
2	3	680	7	7	6	5	4	2	3	
3	4	806	9	6	7	1	5	9	4	
4	5	632	8	10	8	4	6	6	9	
...	...	...	...	...	...	...	...	...	...	
996	997	55	8	7	6	2	5	7	7	
997	998	913	5	5	6	5	6	7	6	
998	999	199	9	5	8	3	8	1	1	
999	1000	539	10	2	7	4	3	4	10	
1000	1001	759	7	2	4	2	1	5	9	

1001 rows × 10 columns

In [24]:
```
df_cleaned = df.drop(columns=["S/N", "Student ID"])
```

In [27]:
```
df_cleaned
```

Out[27]:

	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance
0	5	2	7	6	9	2	1	8
1	6	5	8	6	2	1	2	9
2	7	7	6	5	4	2	3	1
3	9	6	7	1	5	9	4	6
4	8	10	8	4	6	6	9	9
...	...	...	...	...	...	...	...	...
996	8	7	6	2	5	7	7	9
997	5	5	6	5	6	7	6	1
998	9	5	8	3	8	1	1	2
999	10	2	7	4	3	4	10	1
1000	7	2	4	2	1	5	9	9

1001 rows × 8 columns

In [28]:
```
average_ratings = df_cleaned.mean().sort_values(ascending=False)
```

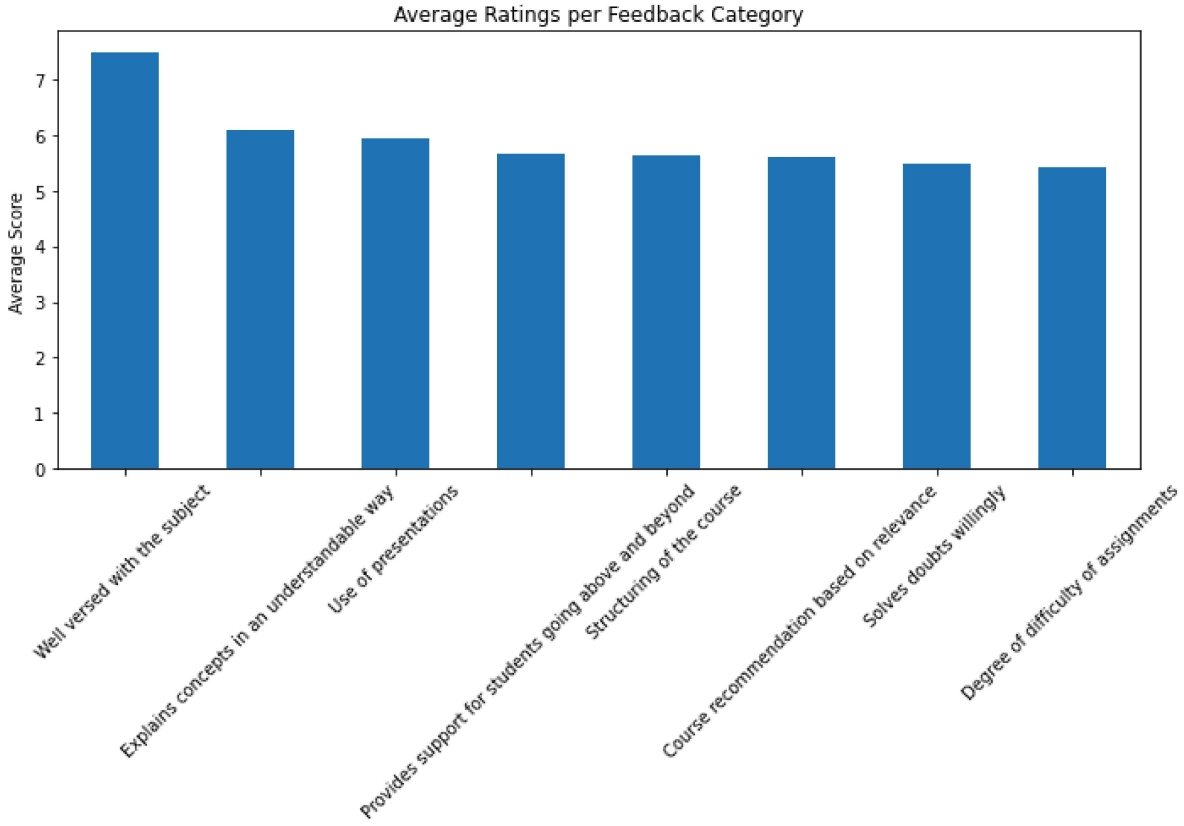
In [29]:
```
average_ratings
```

Out[29]:
```
Well versed with the subject                          7.497502
Explains concepts in an understandable way            6.081918
Use of presentations                                  5.942058
Provides support for students going above and beyond  5.662338
Structuring of the course                             5.636364
Course recommendation based on relevance              5.598402
Solves doubts willingly                               5.474525
Degree of difficulty of assignments                   5.430569
dtype: float64
```

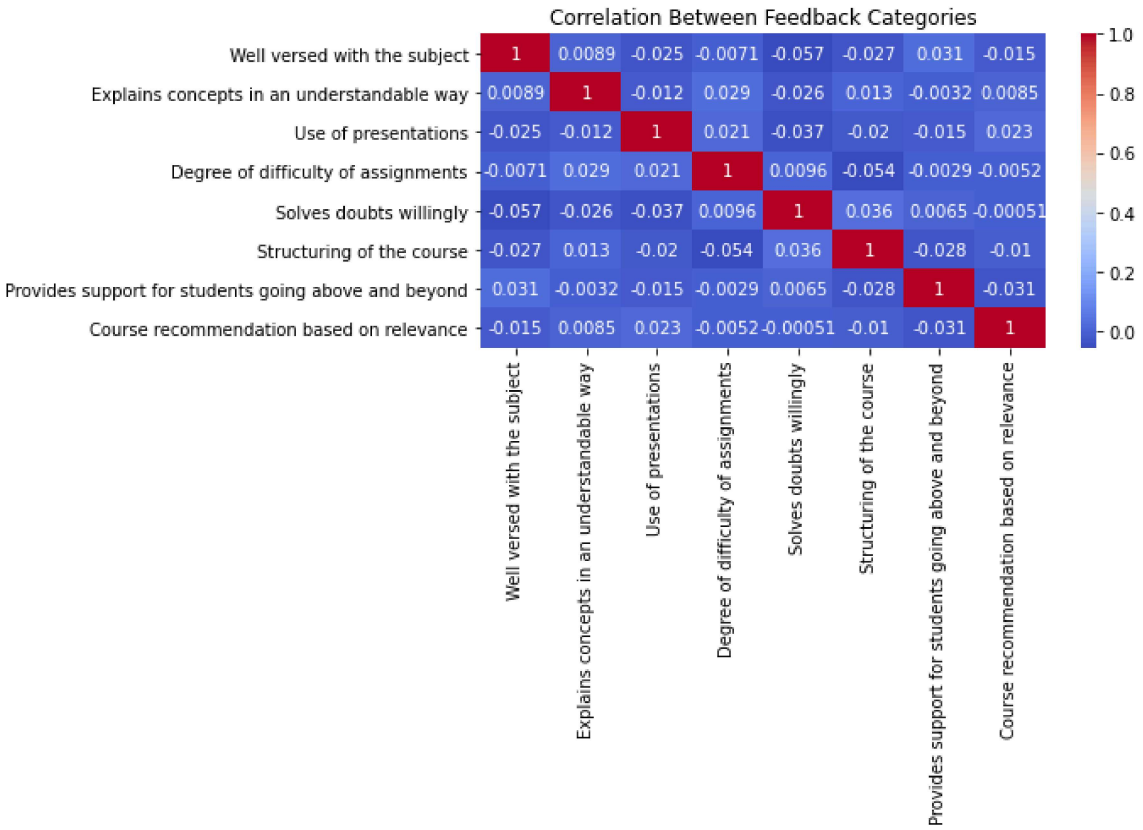
In [32]:

```
plt.figure(figsize=(10, 7))
average_ratings.plot(kind='bar')
plt.title("Average Ratings per Feedback Category")
plt.ylabel("Average Score")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



In [33]:
```
plt.figure(figsize=(10, 7))
sns.heatmap(df_cleaned.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Between Feedback Categories")
plt.tight_layout()
plt.show()
```

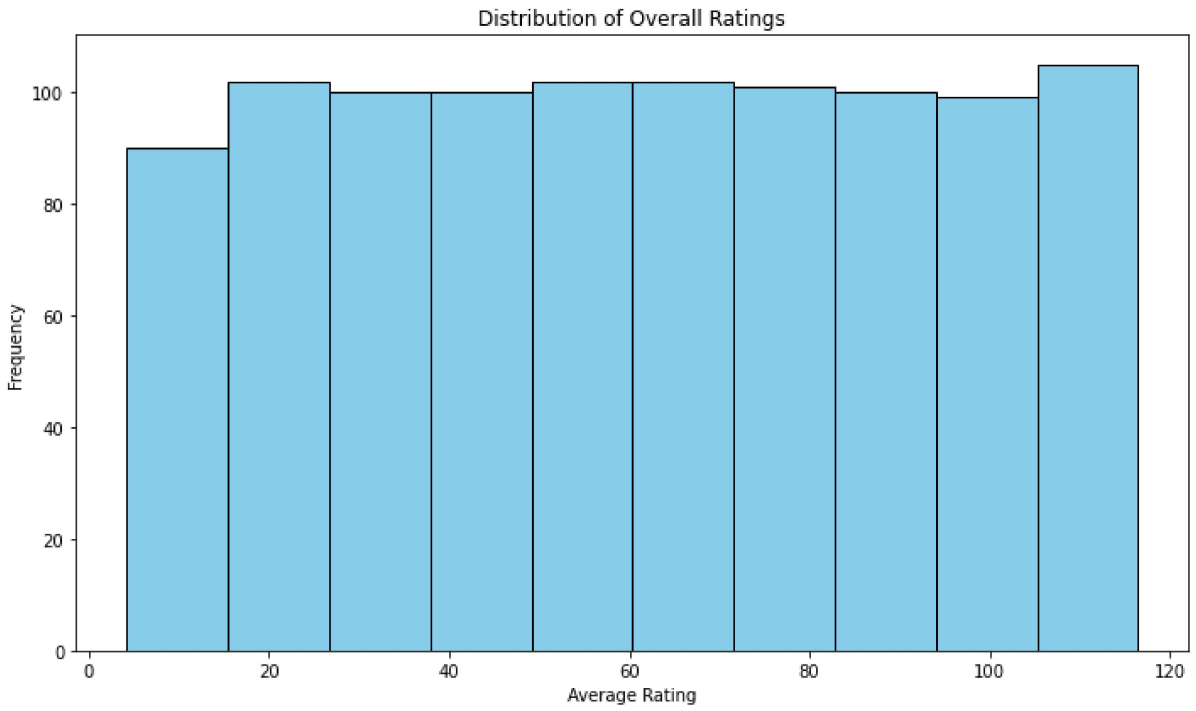


In [40]:
```
df["Overall Rating"] = df.drop(columns=["Student ID"]).mean(axis=1)

```

In [45]:
```
df["Overall Rating"].plot.hist(bins=10, color='skyblue', edgecolor='black', figsize=(10, 6
plt.title("Distribution of Overall Ratings")
plt.xlabel("Average Rating")
plt.tight_layout()
plt.show()

```



In [46]:
```
top_3 = average_ratings.head(3)
bottom_3 = average_ratings.tail(3)

```

In [47]:
```
top_3
```

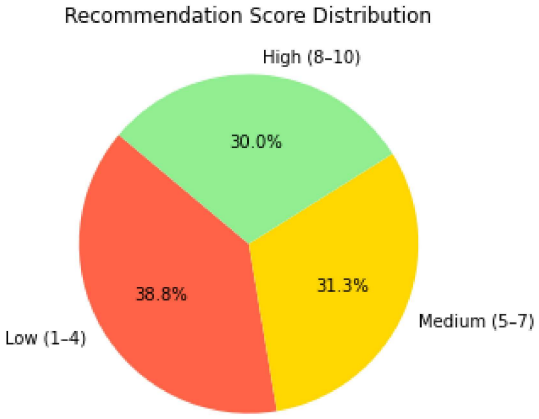
Out[47]:
```
Well versed with the subject                 7.497502
Explains concepts in an understandable way    6.081918
Use of presentations                          5.942058
dtype: float64
```

In [48]:
```
bottom_3
```

Out[48]:
```
Course recommendation based on relevance    5.598402
Solves doubts willingly                     5.474525
Degree of difficulty of assignments         5.430569
dtype: float64
```

In [52]:
```
df["Course recommendation based on relevance"].value_counts(
    bins=[0, 4, 7, 10], sort=True
).plot.pie(
    labels=["Low (1-4)", "Medium (5-7)", "High (8-10)"],
    autopct='%1.1f%%',
    startangle=140,
    colors=["tomato", "gold", "lightgreen"]
)
plt.title("Recommendation Score Distribution")
plt.ylabel("")
plt.tight_layout()
plt.show()

```



In [53]:
```
df["Overall Rating"].describe()

```

Out[53]:
```
count    1001.000000
mean       60.924853
std        32.127932
min         4.222222
25%        33.111111
50%        61.111111
75%        88.555556
max       116.444444
Name: Overall Rating, dtype: float64
```

In [ ]:
```

```