In [1]:
```
# Step 1: Load data
import pandas as pd

df = pd.read_csv(r"student_feedback.csv")
df.head(10)  # First 10 rows
```

Out[1]:

	S.No	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuri of t cou
0	0	340	5	2	7	6	9	
1	1	253	6	5	8	6	2	
2	2	680	7	7	6	5	4	
3	3	806	9	6	7	1	5	
4	4	632	8	10	8	4	6	
5	5	832	7	2	7	8	3	
6	6	772	9	3	5	2	10	
7	7	961	9	8	7	4	4	
8	8	814	6	5	8	6	4	
9	9	863	5	9	4	7	9	

In [11]:
```
# 1. Check column indexes and names
for i, col in enumerate(df.columns):
    print(i, col)
```

```
0 S.No
1 Student ID
2 Well versed with the subject
3 Explains concepts in an understandable way
4 Use of presentations
5 Degree of difficulty of assignments
6 Solves doubts willingly
7 Structuring of the course
8 Provides support for students going above and beyond
9 Course recommendation based on relevance
```

In [17]:
```
#2. Select only feedback columns (excluding Serial No.)
feedback_cols = df.columns[1:]  # selects columns from 3rd to last
df_feedback = df[feedback_cols]  # now df_feedback is defined
```

In [18]:
```
df_feedback.head()
```

Out[18]:

	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	Pr su stu b
0	340	5	2	7	6	9	2	
1	253	6	5	8	6	2	1	
2	680	7	7	6	5	4	2	
3	806	9	6	7	1	5	9	
4	632	8	10	8	4	6	6	

In [19]:
```
df_feedback.describe()
```

Out[19]:

	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly
count	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000
mean	500.000000	7.497502	6.081918	5.942058	5.430569	5.474525
std	289.108111	1.692998	2.597168	1.415853	2.869046	2.874648
min	0.000000	5.000000	2.000000	4.000000	1.000000	1.000000
25%	250.000000	6.000000	4.000000	5.000000	3.000000	3.000000
50%	500.000000	8.000000	6.000000	6.000000	5.000000	6.000000
75%	750.000000	9.000000	8.000000	7.000000	8.000000	8.000000
max	1000.000000	10.000000	10.000000	8.000000	10.000000	10.000000

In [15]:
```
df.tail()
```

Out[15]:

	S.No	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Struc
996	996	55	8	7	6	2	5	
997	997	913	5	5	6	5	6	
998	998	199	9	5	8	3	8	
999	999	539	10	2	7	4	3	
1000	1000	759	7	2	4	2	1	

In [20]:
```
# Check for missing values
df_feedback.isnull().sum()
```

Out[20]:
```
Student ID                                                0
Well versed with the subject                              0
Explains concepts in an understandable way                0
Use of presentations                                      0
Degree of difficulty of assignments                       0
Solves doubts willingly                                   0
Structuring of the course                                 0
Provides support for students going above and beyond      0
Course recommendation based on relevance                  0
dtype: int64
```

In [3]:
```
import pandas as pd

# Load your CSV file into a DataFrame
df = pd.read_csv("student_feedback.csv")  # Make sure the file path is correct

# Step 1: Select only the rating columns (skip ID, S.No)
rating_columns = df.columns[2:]

# Step 2: Convert ratings to sentiment buckets
sentiment_results = {}

for col in rating_columns:
    sentiments = df[col].apply(
        lambda x: "Positive" if x >= 8 else ("Negative" if x <= 5 else "Neutral")
    )
    sentiment_counts = sentiments.value_counts()
    sentiment_results[col] = sentiment_counts

# Step 3: Create summary DataFrame
sentiment_summary = pd.DataFrame(sentiment_results).T.fillna(0).astype(int)
```

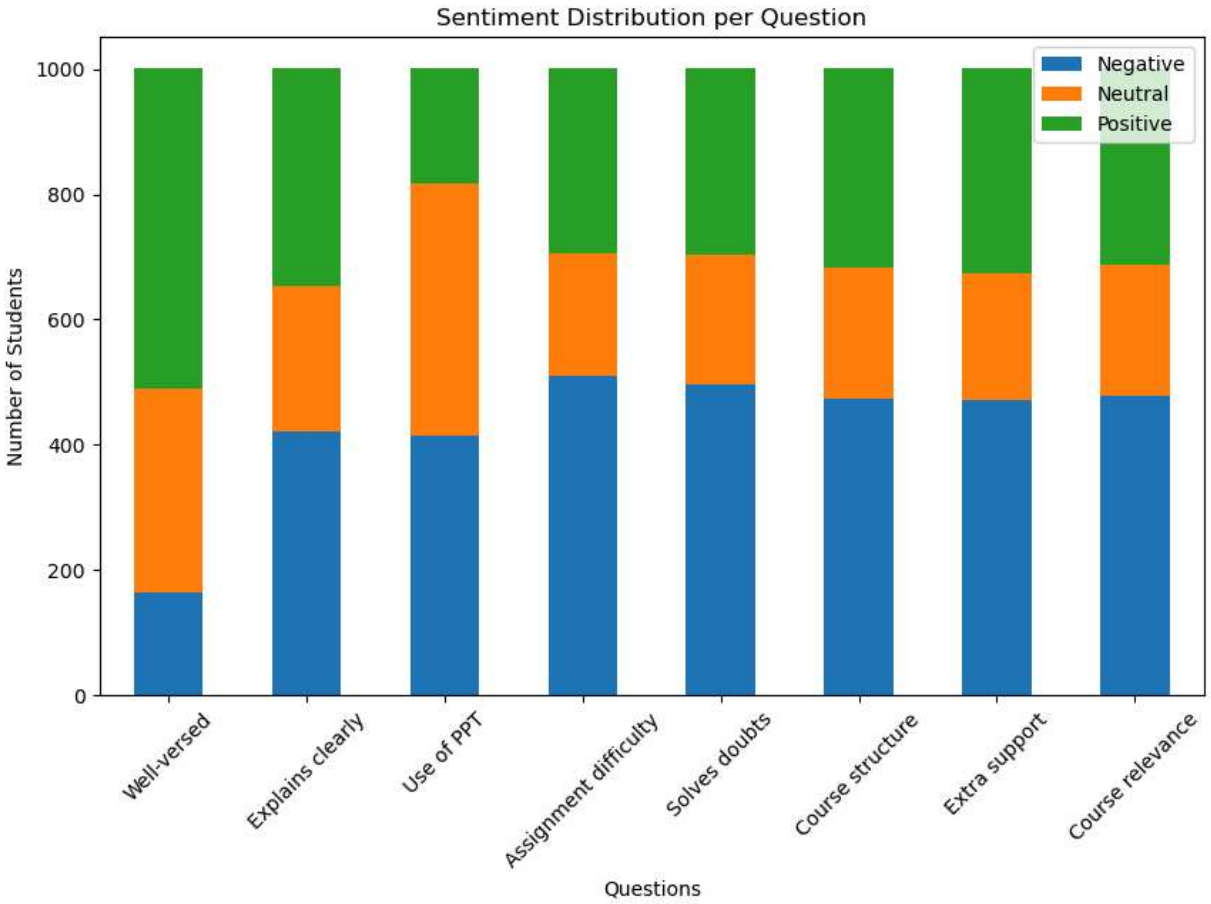
```
print("Sentiment Analysis Summary:")
print(sentiment_summary)
```

```
Sentiment Analysis Summary:
                                                  Negative  Neutral  \
Well versed with the subject                           165      324
Explains concepts in an understandable way             422      232
Use of presentations                                   414      404
Degree of difficulty of assignments                    509      197
Solves doubts willingly                                496      207
Structuring of the course                              474      209
Provides support for students going above and b...     472      202
Course recommendation based on relevance               478      210

                                                  Positive
Well versed with the subject                           512
Explains concepts in an understandable way             347
Use of presentations                                   183
Degree of difficulty of assignments                    295
Solves doubts willingly                                298
Structuring of the course                              318
Provides support for students going above and b...     327
Course recommendation based on relevance               313
```

In [5]:
```
short_names = [
    "Well-versed",
    "Explains clearly",
    "Use of PPT",
    "Assignment difficulty",
    "Solves doubts",
    "Course structure",
    "Extra support",
    "Course relevance"
]
import matplotlib.pyplot as plt
# Replace index (long question names) with short names
sentiment_summary.index = short_names[:len(sentiment_summary)]

# Now plot with short names
sentiment_summary.plot(
    kind="bar", stacked=True, figsize=(10,6),
    title="Sentiment Distribution per Question"
)
plt.ylabel("Number of Students")
plt.xlabel("Questions")
plt.xticks(rotation=45)
plt.show()
```



In [21]:
```
# 3. Select only the feedback questions (drop Student ID column too)
feedback_questions = df_feedback.columns[1:]  # exclude Student ID
```

In [22]:
```
# Show frequency of each rating per question
for col in feedback_questions:
    print(df_feedback[col].value_counts())
```

```
Well versed with the subject
9     182
8     177
6     172
5     165
10    153
7     152
Name: count, dtype: int64
Explains concepts in an understandable way
9     124
6     118
2     114
7     114
10    113
8     110
4     106
3     103
5      99
Name: count, dtype: int64
Use of presentations
4    218
7    208
6    196
5    196
8    183
Name: count, dtype: int64
Degree of difficulty of assignments
6     112
3     107
9     107
2     105
1     100
4     100
5      97
8      95
10     93
7      85
Name: count, dtype: int64
Solves doubts willingly
6     110
2     105
1     105
8     102
9     101
4     101
7      97
5      96
10     95
3      89
Name: count, dtype: int64
Structuring of the course
10    123
6     109
4     103
1     102
7     100
```

```
8     100
9      95
2      94
5      88
3      87
Name: count, dtype: int64
Provides support for students going above and beyond
7     114
8     113
9     107
10    107
4     102
3     100
1      98
6      88
2      87
5      85
Name: count, dtype: int64
Course recommendation based on relevance
9     111
7     106
6     104
4     103
1     102
10    102
8     100
2      94
5      90
3      89
Name: count, dtype: int64
```

In [30]:
```python
# 4. Calculate average rating per question
# Assuming df_feedback has Student ID + 8 feedback questions
feedback_questions = df_feedback.columns[1:]  # exclude Student ID

# Calculate average ratings per question
avg_ratings = df_feedback[feedback_questions].mean()

# View the result
print(avg_ratings)
```

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

In [34]:
```python
avg_ratings = df_feedback[feedback_questions].mean()
```

In [31]:
```python
# 4. Calculate average rating per question
high_satisfaction = avg_ratings[avg_ratings >= 8]  # adjust threshold if scale is 1
print("High satisfaction areas:")
```

```
print(high_satisfaction)

# Identify areas for improvement
low_satisfaction = avg_ratings[avg_ratings < 7]  # adjust threshold
print("\nAreas for improvement:")
print(low_satisfaction)
```

```
High satisfaction areas:
Series([], dtype: float64)

Areas for improvement:
Explains concepts in an understandable way              6.081918
Use of presentations                                    5.942058
Degree of difficulty of assignments                     5.430569
Solves doubts willingly                                 5.474525
Structuring of the course                               5.636364
Provides support for students going above and beyond    5.662338
Course recommendation based on relevance                5.598402
dtype: float64
```

In [37]:
```
# 6. Generate recommendations automaticallyrecommendations = {}
for question, rating in avg_ratings.items():
    if rating >= 8:
        recommendations[question] = "Keep current approach; students satisfied."
    elif rating < 7:
        recommendations[question] = "Needs improvement; consider taking student sug
    else:
        recommendations[question] = "Moderate satisfaction; monitor and improve if

print("\nRecommendations per Question:")
for q, rec in recommendations.items():
    print(f"{q}: {rec}")
```

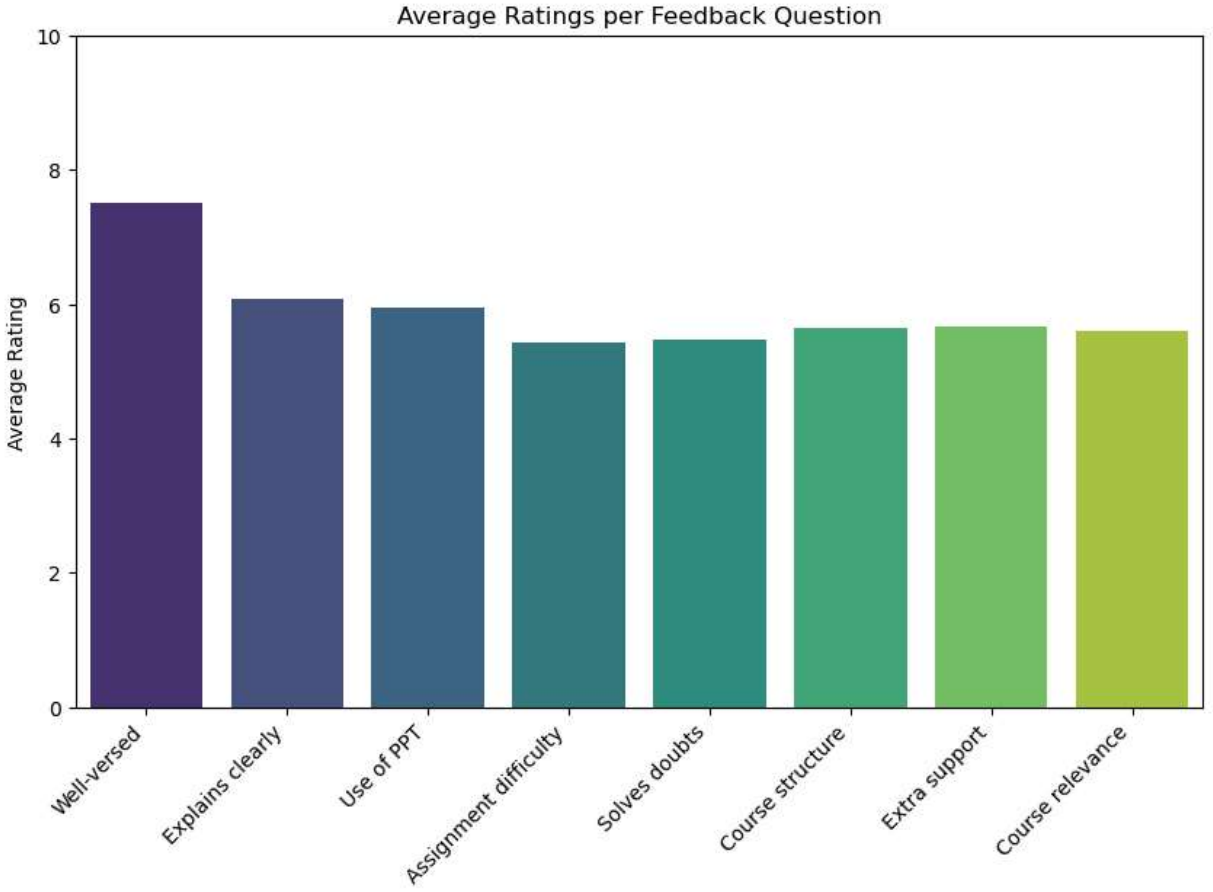
```
Recommendations per Question:
Well versed with the subject: Moderate satisfaction; monitor and improve if needed.
Explains concepts in an understandable way: Needs improvement; consider taking stude
nt suggestions or providing additional support.
Use of presentations: Needs improvement; consider taking student suggestions or prov
iding additional support.
Degree of difficulty of assignments: Needs improvement; consider taking student sugg
estions or providing additional support.
Solves doubts willingly: Needs improvement; consider taking student suggestions or p
roviding additional support.
Structuring of the course: Needs improvement; consider taking student suggestions or
providing additional support.
Provides support for students going above and beyond: Needs improvement; consider ta
king student suggestions or providing additional support.
Course recommendation based on relevance: Needs improvement; consider taking student
suggestions or providing additional support.
```

In [27]:
```
short_names = [
    "Well-versed",
    "Explains clearly",
    "Use of PPT",
    "Assignment difficulty",
    "Solves doubts",
    "Course structure",
```

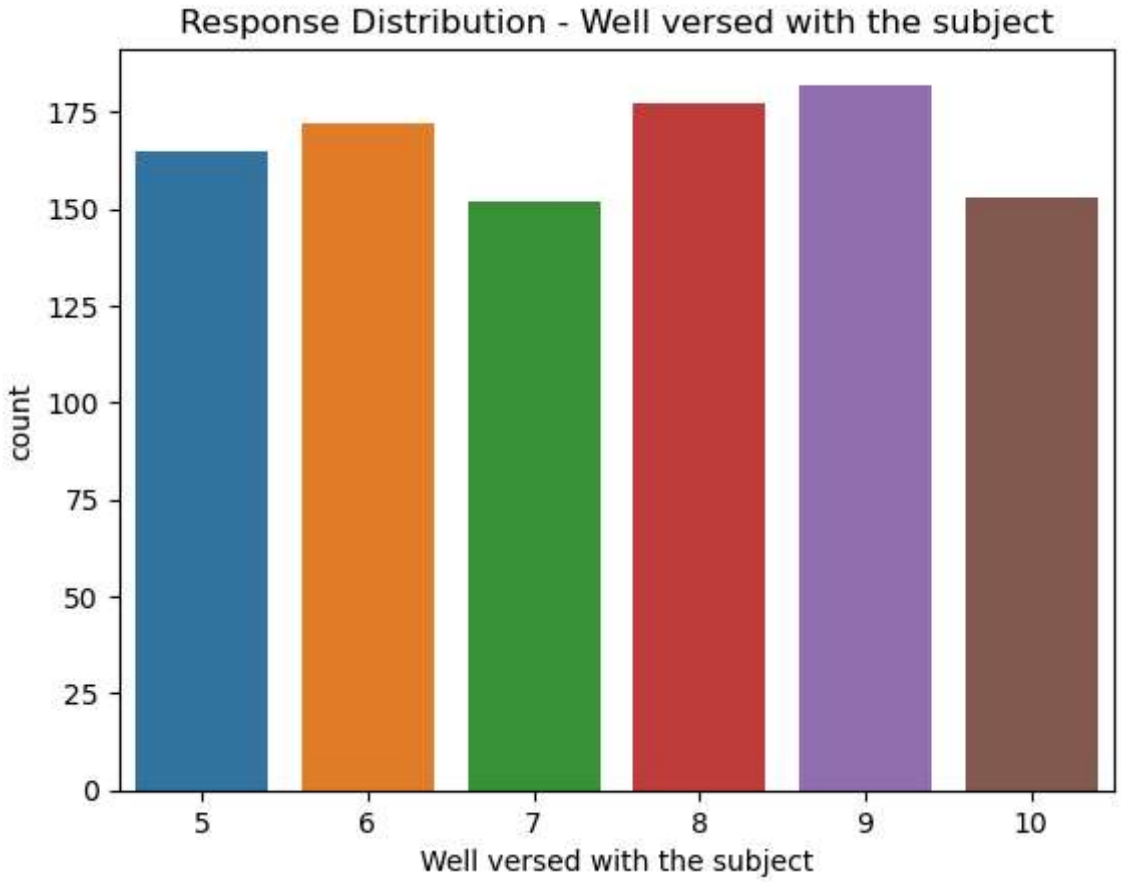
```
    "Extra support",
    "Course relevance"
]
```

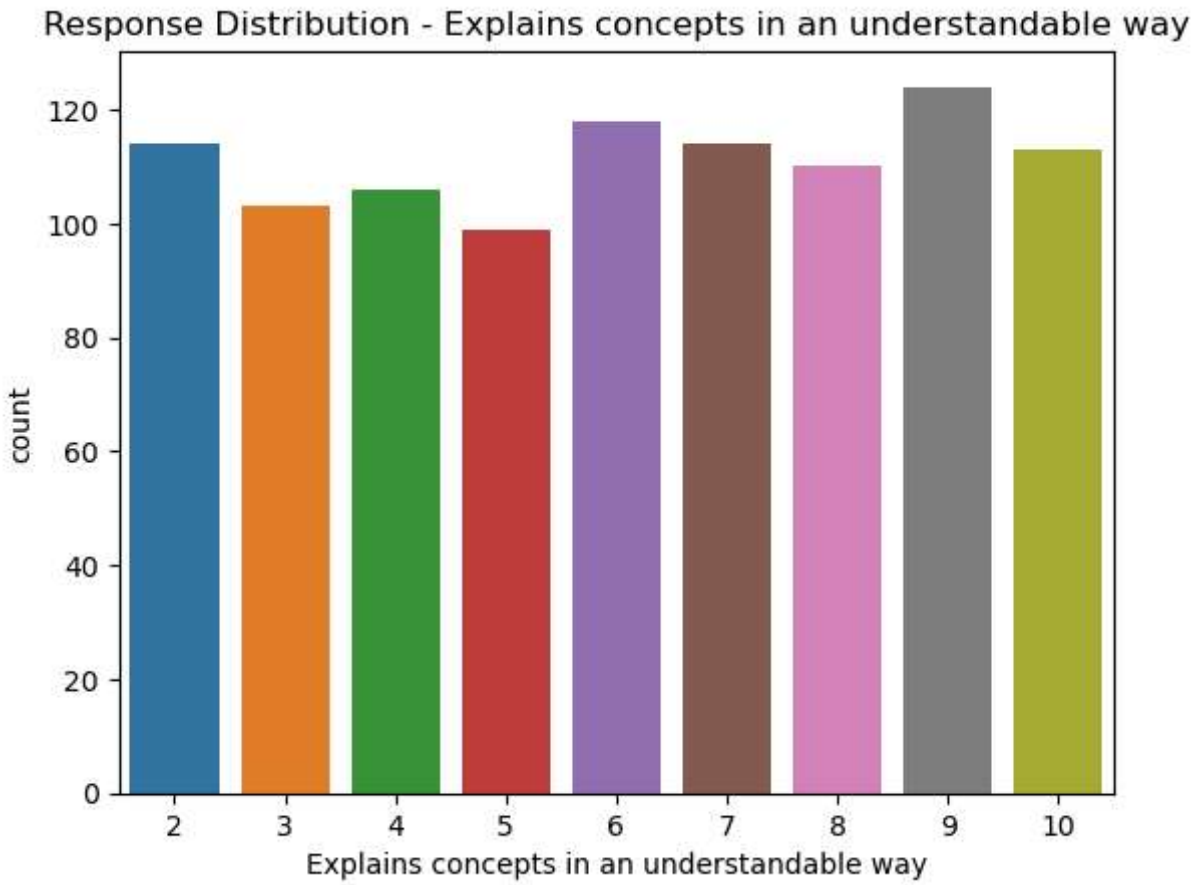
In [28]:
```
# 7. Bar plot of average ratings
plt.figure(figsize=(10,6))
sns.barplot(x=short_names, y=avg_ratings.values, palette="viridis")
plt.xticks(rotation=45, ha='right')
plt.ylabel("Average Rating")
plt.title("Average Ratings per Feedback Question")
plt.ylim(0, 10)  # optional: keep scale consistent
plt.show()
```

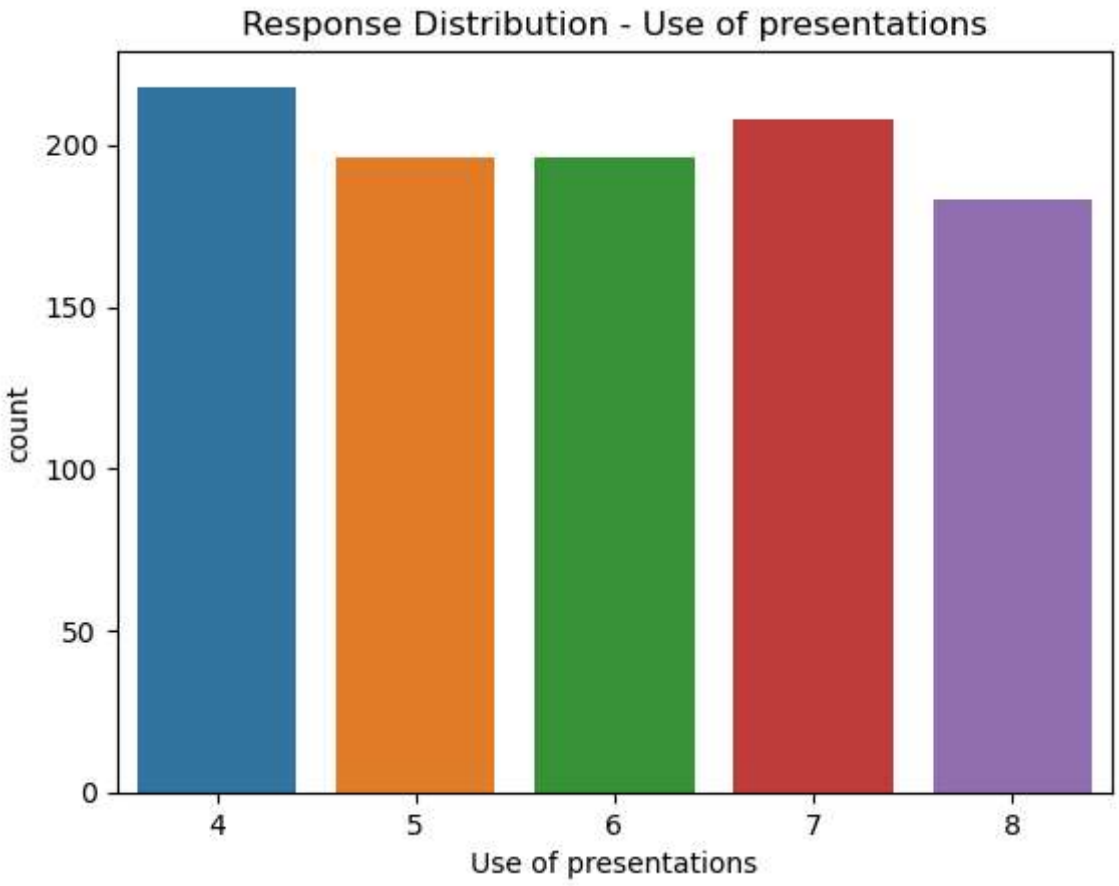
```
C:\Users\Teju\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1765: FutureWarning: u
nique with argument that is not not a Series, Index, ExtensionArray, or np.ndarray i
s deprecated and will raise in a future version.
  order = pd.unique(vector)
```

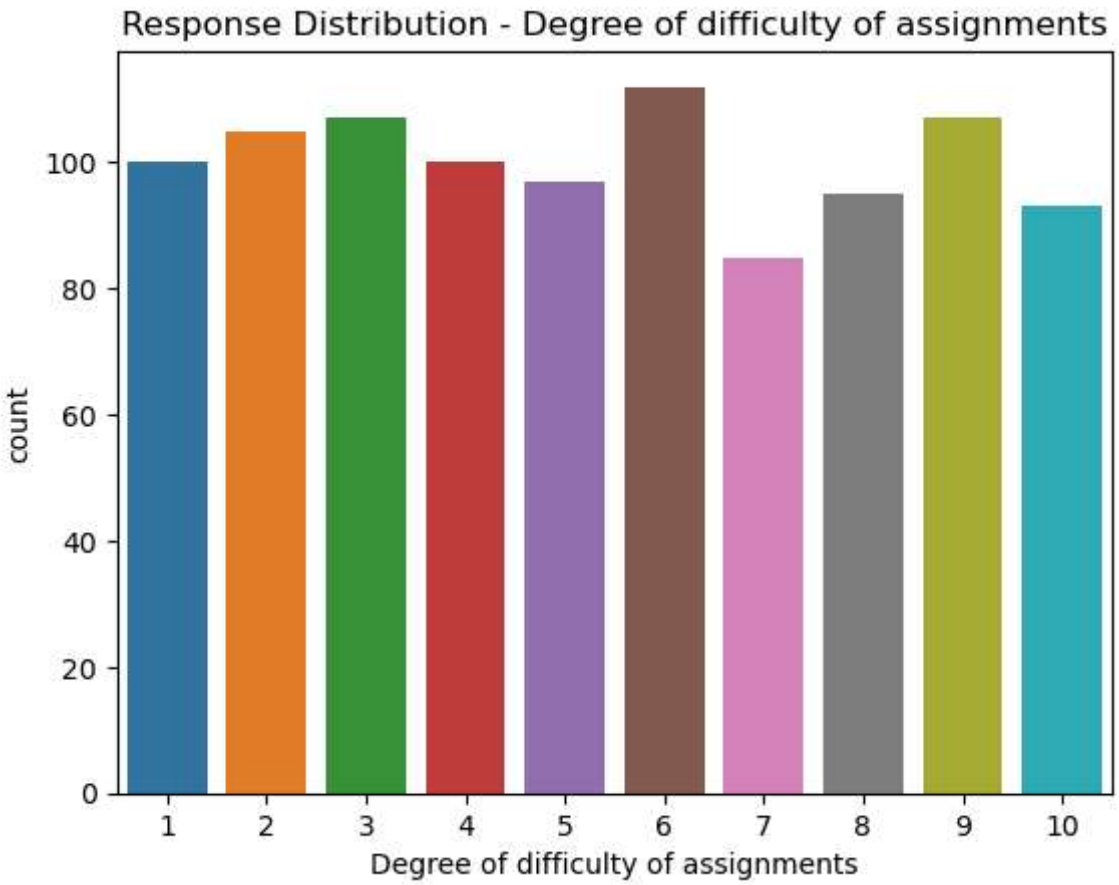


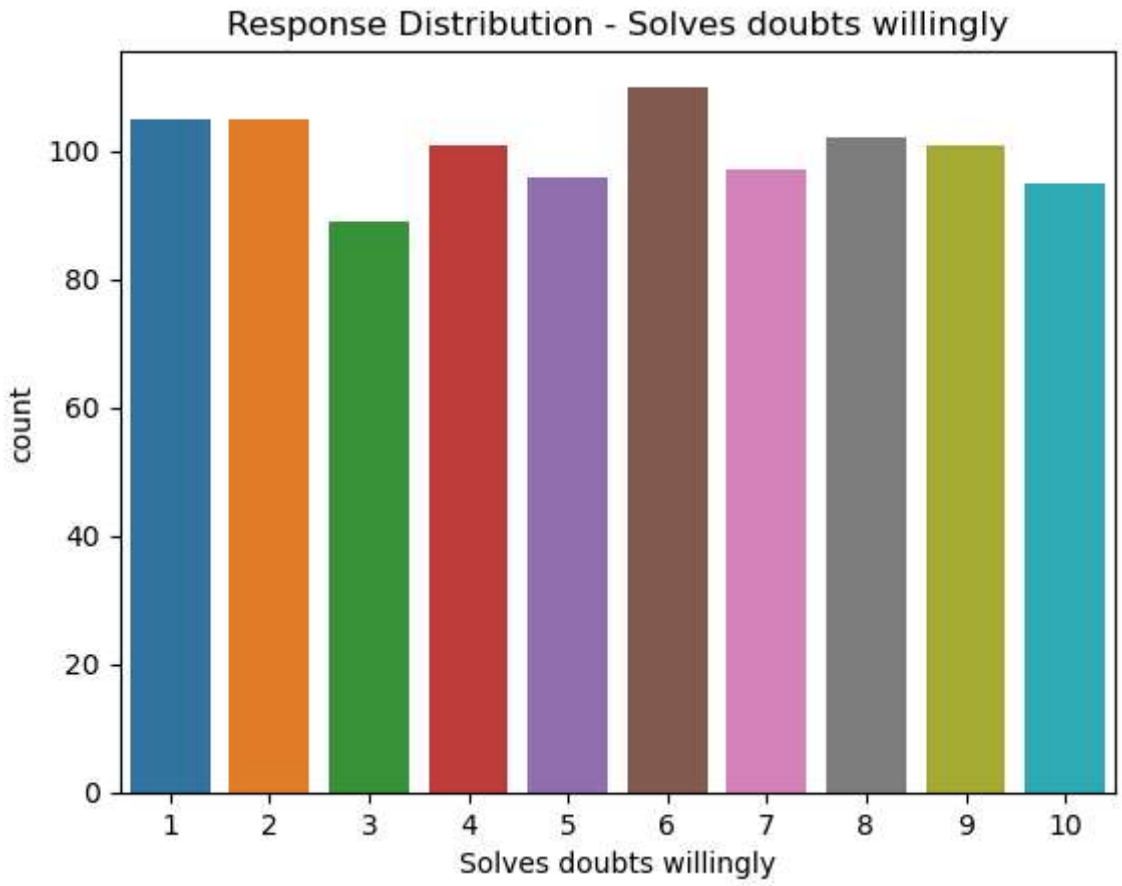
In [29]:
```
for col in feedback_questions:
    sns.countplot(x=df_feedback[col])
    plt.title(f"Response Distribution - {col}")
    plt.show()
```

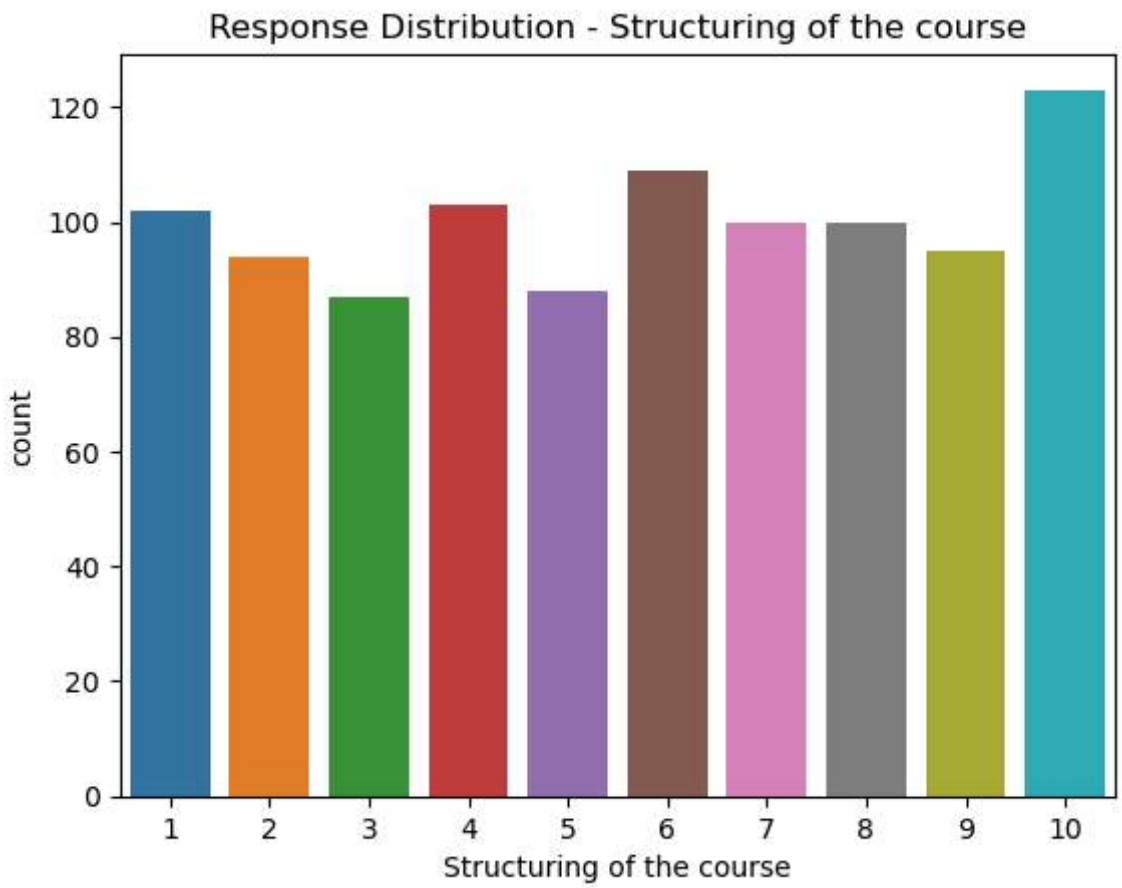

Response Distribution - Well versed with the subject
count
0
25
50
75
100
125
150
175
5
6
7
8
9
10
Well versed with the subject

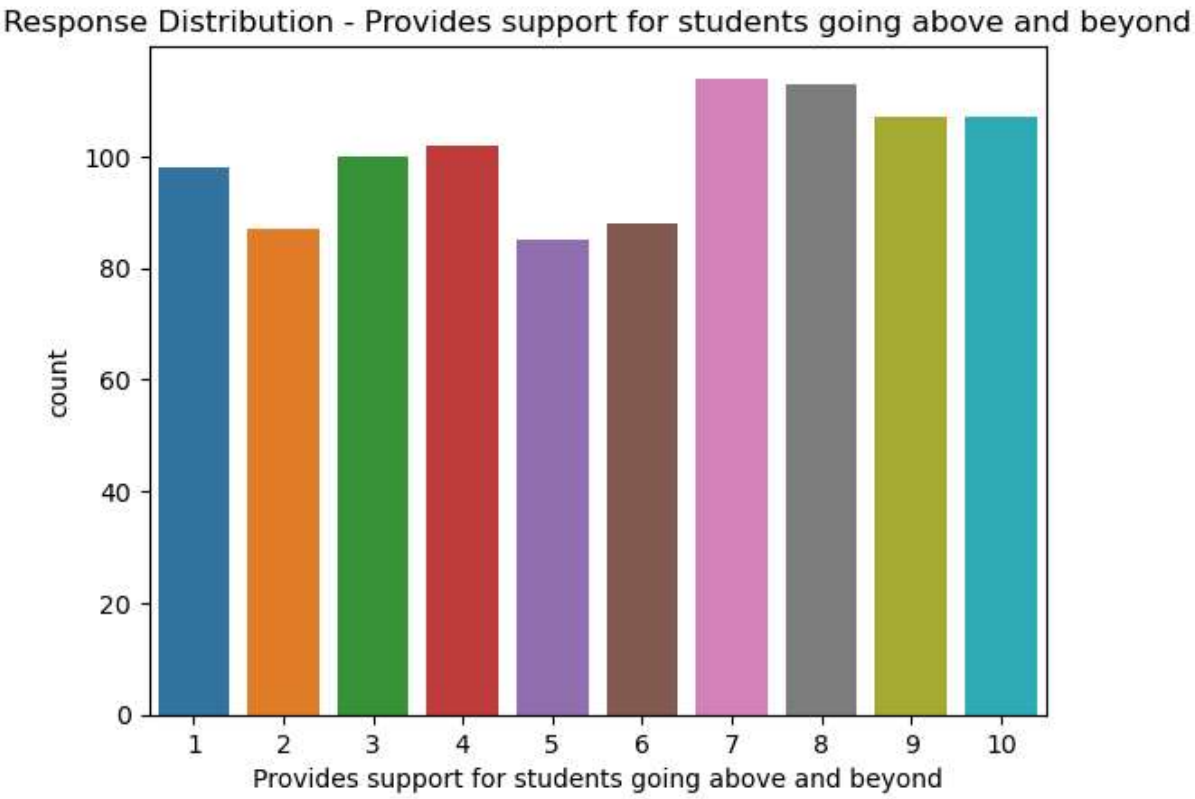

Response Distribution - Explains concepts in an understandable way
count
0
20
40
60
80
100
120
2
3
4
5
6
7
8
9
10
Explains concepts in an understandable way

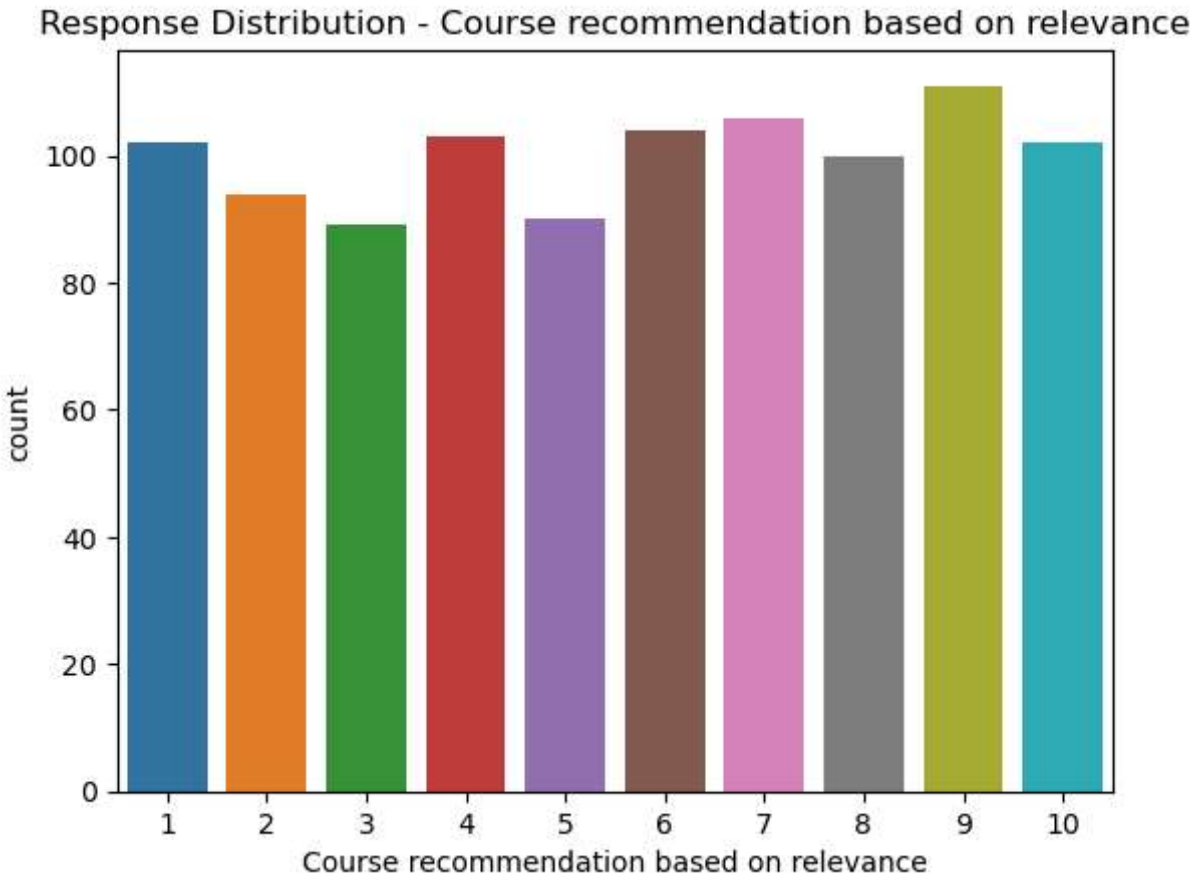

Response Distribution - Use of presentations
count
0
50
100
150
200
4
5
6
7
8
Use of presentations


Response Distribution - Degree of difficulty of assignments
count
0
20
40
60
80
100
1
2
3
4
5
6
7
8
9
10
Degree of difficulty of assignments


Response Distribution - Solves doubts willingly
count
0
20
40
60
80
100
1
2
3
4
5
6
7
8
9
10
Solves doubts willingly


Response Distribution - Structuring of the course
count
0
20
40
60
80
100
120
1
2
3
4
5
6
7
8
9
10
Structuring of the course





In [38]:
```
# Step 6: Create summary table
summary_data = []

for question, rating in avg_ratings.items():
```

```
    if rating >= 8:
        status = "High Satisfaction"
        rec = "Keep current approach; students satisfied."
    elif rating < 7:
        status = "Needs Improvement"
        rec = "Needs improvement; consider extra support or adjustments."
    else:
        status = "Moderate"
        rec = "Monitor and improve if needed."

    summary_data.append({
        "Question": question,
        "Average Rating": round(rating, 2),
        "Status": status,
        "Recommendation": rec
    })

summary_df = pd.DataFrame(summary_data)

# Display summary table
print("\nSummary Table:")
print(summary_df)
```

```
Summary Table:
                                            Question  Average Rating  \
0                        Well versed with the subject            7.50
1         Explains concepts in an understandable way            6.08
2                                Use of presentations            5.94
3               Degree of difficulty of assignments            5.43
4                             Solves doubts willingly            5.47
5                           Structuring of the course            5.64
6  Provides support for students going above and ...            5.66
7           Course recommendation based on relevance            5.60

              Status                                     Recommendation
0           Moderate                       Monitor and improve if needed.
1  Needs Improvement  Needs improvement; consider extra support or a...
2  Needs Improvement  Needs improvement; consider extra support or a...
3  Needs Improvement  Needs improvement; consider extra support or a...
4  Needs Improvement  Needs improvement; consider extra support or a...
5  Needs Improvement  Needs improvement; consider extra support or a...
6  Needs Improvement  Needs improvement; consider extra support or a...
7  Needs Improvement  Needs improvement; consider extra support or a...
```

In [ ]: