In [17]:
```
# import python libraries
import numpy as np
import pandas as pd
import matplotlib as mlt
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from textblob import TextBlob
from wordcloud import WordCloud
import re
```

Data Cleaning

In [18]:
```
# import csv file
df = pd.read_csv(r"D:\Project\Student Feedback CSV File.csv", encoding= 'unicode
```

In [19]:
```
# checking if file has stored or not
df.shape
```

Out[19]: (29, 81)

In [20]:
```
# displaying the table
df.head()
```

Out[20]:

	ï»¿Timestamp	StudentName	RollNumber	Class	Semester	Clarity_Prof_Seema	Subl
0	16-08-2025	Madhura Vahile	69	TY	Semester 5	4	
1	16-08-2025	Vansh Rathod	54	TY	Semester 5	4	
2	16-08-2025	Gaurav Bajaj	4	TY	Semester 5	4	
3	16-08-2025	Daneshwari Kaplish	17	TY	Semester 5	5	
4	16-08-2025	Aryan Pol	50	TY	Semester 5	4	

5 rows × 81 columns

In [21]:
```
# Getting column information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29 entries, 0 to 28
Data columns (total 81 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   ï»¿Timestamp                    29 non-null     object
 1   StudentName                     29 non-null     object
 2   RollNumber                      29 non-null     int64
 3   Class                           29 non-null     object
 4   Semester                        29 non-null     object
 5     Clarity_Prof_Seema            29 non-null     int64
 6     SubKnowledge_Prof_Seema       29 non-null     int64
 7     Interaction_Prof_Seema        29 non-null     int64
 8    DoubtSolving_Prof_Seema        29 non-null     int64
 9     Punctuality_Prof_Seema        29 non-null     int64
 10   Comments_Prof_Seema            9 non-null      object
 11    Clarity_Prof_Ashwini          29 non-null     int64
 12    SubKnowledge_Prof_Ashwini     29 non-null     int64
 13    Interaction_Prof_Ashwini      29 non-null     int64
 14  DoubtSolving_Prof_Ashwini       29 non-null     int64
 15    Punctuality_Prof_Ashwini      29 non-null     int64
 16  Comments_Prof_Ashwini           5 non-null      object
 17    Clarity_Mrs_Vaishali          29 non-null     int64
 18    SubKnowledge_Mrs_Vaishali     29 non-null     int64
 19    Interaction_Mrs_Vaishali      28 non-null     float64
 20  DoubtSolving_Mrs_Vaishali       29 non-null     int64
 21    Punctuality_Mrs_Vaishali      29 non-null     int64
 22  Comments_Mrs_Vaishali           4 non-null      object
 23    Clarity_Mrs_Anita             29 non-null     int64
 24    SubKnowledge_Mrs_Anita        29 non-null     int64
 25    Interaction_Mrs_Anita         28 non-null     float64
 26     DoubtSolving_Mrs_Anita       29 non-null     int64
 27    Punctuality_Mrs_Anita         29 non-null     int64
 28  Comments_Mrs_Anita              3 non-null      object
 29    Clarity_Mrs_Sakshi            29 non-null     int64
 30    SubKnowledge_Mrs_Sakshi       29 non-null     int64
 31    Interaction_Mrs_Sakshi        28 non-null     float64
 32    DoubtSolving_Mrs_Sakshi       29 non-null     int64
 33    Punctuality_Mrs_Sakshi        29 non-null     int64
 34  Comments_Mrs_Sakshi             4 non-null      object
 35    Clarity_Mrs_Sonam             29 non-null     int64
 36    SubKnowledge_Mrs_Sonam        29 non-null     int64
 37    Interaction_Mrs_Sonam         28 non-null     float64
 38    DoubtSolving_Mrs_Sonam        29 non-null     int64
 39    Punctuality_Mrs_Sonam         29 non-null     int64
 40  Comments_Mrs_Sonam              4 non-null      object
 41    Clarity_Mr_Pradosh            29 non-null     int64
 42    SubKnowledge_Mr_Pradosh       29 non-null     int64
 43    Interaction_Mr_Pradosh        28 non-null     float64
 44   DoubtSolving_Mr_Pradosh        29 non-null     int64
 45    Punctuality_Mr_Pradosh        29 non-null     int64
 46  Comments_Mr_Pradosh             4 non-null      object
 47    Clarity_Ms_Rupali             29 non-null     int64
 48    SubKnowledge_Ms_Rupali        29 non-null     int64
 49    Interaction_Ms_Rupali         28 non-null     float64
 50  DoubtSolving_Ms_Rupali          29 non-null     int64
 51    Punctuality_Ms_Rupali         29 non-null     int64
 52  Comments_Ms_Rupali              4 non-null      object
 53    Clarity_Ms_Maitri             29 non-null     int64
 54    SubKnowledge_Ms_Maitri        29 non-null     int64
```

```
 55    Interaction_Ms_Maitri          27 non-null     float64
 56  DoubtSolving_Ms_Maitri           29 non-null     int64
 57    Punctuality_Ms_Maitri          29 non-null     int64
 58  Comments_Ms_Maitri               4 non-null      object
 59    Clarity_Mrs_Priya              29 non-null     int64
 60    SubKnowledge_Mrs_Priya         29 non-null     int64
 61    Interaction_Mrs_Priya          28 non-null     float64
 62  DoubtSolving_Mrs_Priya           29 non-null     int64
 63    Punctuality_Mrs_Priya          29 non-null     int64
 64  Comments_Mrs_Priya               2 non-null      object
 65    Clarity_Ms_Abha                29 non-null     int64
 66    SubKnowledge_Ms_Abha           29 non-null     int64
 67    Interaction_Ms_Abha            28 non-null     float64
 68  DoubtSolving_Ms_Abha             29 non-null     int64
 69    Punctuality_Ms_Abha            29 non-null     int64
 70  Comments_Ms_Abha                 2 non-null      object
 71    Library_Resources              29 non-null     int64
 72  ClassroomEnvironment             29 non-null     int64
 73  Computer/ITLabWiFi               29 non-null     int64
 74  CleanlinessHygiene               29 non-null     int64
 75  SportsExtracurricular            29 non-null     int64
 76  FacilitiesSuggestions            4 non-null      object
 77  CollegeExperience                29 non-null     int64
 78  CollegeRecommendation            29 non-null     object
 79  Fav_Aspects_College              8 non-null      object
 80   Suggestions                     5 non-null      object
dtypes: float64(9), int64(53), object(19)
memory usage: 18.5+ KB
```

In [22]:
```
# Checking if table contains null or missing values or not
pd.isnull(df)
```

Out[22]:

	ï»¿Timestamp	StudentName	RollNumber	Class	Semester	Clarity_Prof_Seema	Sul
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
5	False	False	False	False	False	False	
6	False	False	False	False	False	False	
7	False	False	False	False	False	False	
8	False	False	False	False	False	False	
9	False	False	False	False	False	False	
10	False	False	False	False	False	False	
11	False	False	False	False	False	False	
12	False	False	False	False	False	False	
13	False	False	False	False	False	False	
14	False	False	False	False	False	False	
15	False	False	False	False	False	False	
16	False	False	False	False	False	False	
17	False	False	False	False	False	False	
18	False	False	False	False	False	False	
19	False	False	False	False	False	False	
20	False	False	False	False	False	False	
21	False	False	False	False	False	False	
22	False	False	False	False	False	False	
23	False	False	False	False	False	False	
24	False	False	False	False	False	False	
25	False	False	False	False	False	False	
26	False	False	False	False	False	False	
27	False	False	False	False	False	False	
28	False	False	False	False	False	False	

29 rows × 81 columns

In [23]:
```
# displaying total null values per column
pd.isnull(df).sum()
```

Out[23]:
```
ï»¿Timestamp              0
StudentName               0
RollNumber                0
Class                     0
Semester                  0
                        ..
FacilitiesSuggestions    25
CollegeExperience         0
CollegeRecommendation     0
Fav_Aspects_College      21
 Suggestions             24
Length: 81, dtype: int64
```

In [24]:
```
# removing extra spaces of column names
df.columns = df.columns.str.strip()
```

In [25]:
```
# filling space into null values or missing values of text columns
text_cols = ["FacilitiesSuggestions", "Fav_Aspects_College", "Suggestions"]
df[text_cols] = df[text_cols].fillna("")
```

In [26]:
```
# checking for null values
pd.isnull(df).sum()
```

Out[26]:
```
ï»¿Timestamp            0
StudentName               0
RollNumber                0
Class                     0
Semester                  0
                       ..
FacilitiesSuggestions     0
CollegeExperience         0
CollegeRecommendation     0
Fav_Aspects_College       0
Suggestions               0
Length: 81, dtype: int64
```

In [27]:
```
# getting column names
df.columns
```

Out[27]:
```
Index(['ï»¿Timestamp', 'StudentName', 'RollNumber', 'Class', 'Semester',
       'Clarity_Prof_Seema', 'SubKnowledge_Prof_Seema',
       'Interaction_Prof_Seema', 'DoubtSolving_Prof_Seema',
       'Punctuality_Prof_Seema', 'Comments_Prof_Seema', 'Clarity_Prof_Ashwini',
       'SubKnowledge_Prof_Ashwini', 'Interaction_Prof_Ashwini',
       'DoubtSolving_Prof_Ashwini', 'Punctuality_Prof_Ashwini',
       'Comments_Prof_Ashwini', 'Clarity_Mrs_Vaishali',
       'SubKnowledge_Mrs_Vaishali', 'Interaction_Mrs_Vaishali',
       'DoubtSolving_Mrs_Vaishali', 'Punctuality_Mrs_Vaishali',
       'Comments_Mrs_Vaishali', 'Clarity_Mrs_Anita', 'SubKnowledge_Mrs_Anita',
       'Interaction_Mrs_Anita', 'DoubtSolving_Mrs_Anita',
       'Punctuality_Mrs_Anita', 'Comments_Mrs_Anita', 'Clarity_Mrs_Sakshi',
       'SubKnowledge_Mrs_Sakshi', 'Interaction_Mrs_Sakshi',
       'DoubtSolving_Mrs_Sakshi', 'Punctuality_Mrs_Sakshi',
       'Comments_Mrs_Sakshi', 'Clarity_Mrs_Sonam', 'SubKnowledge_Mrs_Sonam',
       'Interaction_Mrs_Sonam', 'DoubtSolving_Mrs_Sonam',
       'Punctuality_Mrs_Sonam', 'Comments_Mrs_Sonam', 'Clarity_Mr_Pradosh',
       'SubKnowledge_Mr_Pradosh', 'Interaction_Mr_Pradosh',
       'DoubtSolving_Mr_Pradosh', 'Punctuality_Mr_Pradosh',
       'Comments_Mr_Pradosh', 'Clarity_Ms_Rupali', 'SubKnowledge_Ms_Rupali',
       'Interaction_Ms_Rupali', 'DoubtSolving_Ms_Rupali',
       'Punctuality_Ms_Rupali', 'Comments_Ms_Rupali', 'Clarity_Ms_Maitri',
       'SubKnowledge_Ms_Maitri', 'Interaction_Ms_Maitri',
       'DoubtSolving_Ms_Maitri', 'Punctuality_Ms_Maitri', 'Comments_Ms_Maitri',
       'Clarity_Mrs_Priya', 'SubKnowledge_Mrs_Priya', 'Interaction_Mrs_Priya',
       'DoubtSolving_Mrs_Priya', 'Punctuality_Mrs_Priya', 'Comments_Mrs_Priya',
       'Clarity_Ms_Abha', 'SubKnowledge_Ms_Abha', 'Interaction_Ms_Abha',
       'DoubtSolving_Ms_Abha', 'Punctuality_Ms_Abha', 'Comments_Ms_Abha',
       'Library_Resources', 'ClassroomEnvironment', 'Computer/ITLabWiFi',
       'CleanlinessHygiene', 'SportsExtracurricular', 'FacilitiesSuggestions',
       'CollegeExperience', 'CollegeRecommendation', 'Fav_Aspects_College',
       'Suggestions'],
      dtype='object')
```

In [28]:
```
# statistical calculations for numeric values
df[['Clarity_Prof_Seema', 'SubKnowledge_Prof_Seema','Interaction_Prof_Seema', 'D
       'Punctuality_Prof_Seema', 'Clarity_Prof_Ashwini','SubKnowledge_Prof_Ashwi
       'DoubtSolving_Prof_Ashwini', 'Punctuality_Prof_Ashwini', 'Clarity_Mrs_Vai
       'DoubtSolving_Mrs_Vaishali', 'Punctuality_Mrs_Vaishali','Clarity_Mrs_Anit
       'Punctuality_Mrs_Anita',  'Clarity_Mrs_Sakshi','SubKnowledge_Mrs_Sakshi',
       'DoubtSolving_Mrs_Sakshi', 'Punctuality_Mrs_Sakshi','Clarity_Mrs_Sonam',
       'Interaction_Mrs_Sonam', 'DoubtSolving_Mrs_Sonam','Punctuality_Mrs_Sonam'
       'DoubtSolving_Mr_Pradosh', 'Punctuality_Mr_Pradosh','Clarity_Ms_Rupali',
       'Punctuality_Ms_Rupali',  'Clarity_Ms_Maitri','SubKnowledge_Ms_Maitri', '
       'Clarity_Mrs_Priya', 'SubKnowledge_Mrs_Priya', 'Interaction_Mrs_Priya',
       'DoubtSolving_Mrs_Priya', 'Punctuality_Mrs_Priya', 'Clarity_Ms_Abha', 'Su
       'DoubtSolving_Ms_Abha', 'Punctuality_Ms_Abha', 'Library_Resources', 'Clas
       'CollegeExperience',
       ]].describe()
```

Out[28]:

	Clarity_Prof_Seema	SubKnowledge_Prof_Seema	Interaction_Prof_Seema	DoubtSo
count	29.000000	29.000000	29.000000	
mean	4.379310	4.310345	4.206897	
std	0.902924	1.003688	0.901559	
min	1.000000	1.000000	1.000000	
25%	4.000000	4.000000	4.000000	
50%	5.000000	5.000000	4.000000	
75%	5.000000	5.000000	5.000000	
max	5.000000	5.000000	5.000000	

8 rows × 61 columns

Exploratory Data Analysis

combining columns per teacher

In [29]:
```
teacher_name = "Prof_Seema"

# Rating columns for this teacher
rating_cols_Prof_Seema = [col for col in df.columns if teacher_name in col and n

print("Rating Columns:", rating_cols_Prof_Seema)
```

```
Rating Columns: ['Clarity_Prof_Seema', 'SubKnowledge_Prof_Seema', 'Interaction_Pr
of_Seema', 'DoubtSolving_Prof_Seema', 'Punctuality_Prof_Seema']
```

In [30]:
```
teacher_name = "Mrs_Sonam"

# Rating columns for this teacher
rating_cols_Mrs_Sonam = [col for col in df.columns if teacher_name in col and no

print("Rating Columns:", rating_cols_Mrs_Sonam)
```

```
Rating Columns: ['Clarity_Mrs_Sonam', 'SubKnowledge_Mrs_Sonam', 'Interaction_Mrs_
Sonam', 'DoubtSolving_Mrs_Sonam', 'Punctuality_Mrs_Sonam']
```

In [31]:
```
teacher_name = "Mrs_Anita"

# Rating columns for this teacher
rating_cols_Mrs_Anita = [col for col in df.columns if teacher_name in col and no

print("Rating Columns:", rating_cols_Mrs_Anita)
```

```
Rating Columns: ['Clarity_Mrs_Anita', 'SubKnowledge_Mrs_Anita', 'Interaction_Mrs_
Anita', 'DoubtSolving_Mrs_Anita', 'Punctuality_Mrs_Anita']
```

In [32]:
```
teacher_name = "Prof_Ashwini"

# Rating columns for this teacher
rating_cols_Prof_Ashwini = [col for col in df.columns if teacher_name in col and

print("Rating Columns:", rating_cols_Prof_Ashwini)
```

```
Rating Columns: ['Clarity_Prof_Ashwini', 'SubKnowledge_Prof_Ashwini', 'Interactio
n_Prof_Ashwini', 'DoubtSolving_Prof_Ashwini', 'Punctuality_Prof_Ashwini']
```

In [33]:
```
teacher_name = "Mrs_Sakshi"

# Rating columns for this teacher
rating_cols_Mrs_Sakshi = [col for col in df.columns if teacher_name in col and n

print("Rating Columns:", rating_cols_Mrs_Sakshi)
```

```
Rating Columns: ['Clarity_Mrs_Sakshi', 'SubKnowledge_Mrs_Sakshi', 'Interaction_Mr
s_Sakshi', 'DoubtSolving_Mrs_Sakshi', 'Punctuality_Mrs_Sakshi']
```

In [34]:
```
teacher_name = "Mr_Pradosh"

# Rating columns for this teacher
rating_cols_Mr_Pradosh = [col for col in df.columns if teacher_name in col and n

print("Rating Columns:", rating_cols_Mr_Pradosh)
```

```
Rating Columns: ['Clarity_Mr_Pradosh', 'SubKnowledge_Mr_Pradosh', 'Interaction_Mr
_Pradosh', 'DoubtSolving_Mr_Pradosh', 'Punctuality_Mr_Pradosh']
```

In [35]:
```
teacher_name = "Ms_Maitri"

# Rating columns for this teacher
rating_cols_Ms_Maitri = [col for col in df.columns if teacher_name in col and no

print("Rating Columns:", rating_cols_Ms_Maitri)
```

```
Rating Columns: ['Clarity_Ms_Maitri', 'SubKnowledge_Ms_Maitri', 'Interaction_Ms_M
aitri', 'DoubtSolving_Ms_Maitri', 'Punctuality_Ms_Maitri']
```

In [36]:
```
teacher_name = "Mrs_Priya"

# Rating columns for this teacher
rating_cols_Mrs_Priya = [col for col in df.columns if teacher_name in col and no

print("Rating Columns:", rating_cols_Mrs_Priya)
```

```
Rating Columns: ['Clarity_Mrs_Priya', 'SubKnowledge_Mrs_Priya', 'Interaction_Mrs_
Priya', 'DoubtSolving_Mrs_Priya', 'Punctuality_Mrs_Priya']
```

In [37]:
```
teacher_name = "Ms_Abha"

# Rating columns for this teacher
rating_cols_Ms_Abha = [col for col in df.columns if teacher_name in col and not

print("Rating Columns:", rating_cols_Ms_Abha)
```

```
Rating Columns: ['Clarity_Ms_Abha', 'SubKnowledge_Ms_Abha', 'Interaction_Ms_Abh
a', 'DoubtSolving_Ms_Abha', 'Punctuality_Ms_Abha']
```

In [38]:
```
teacher_name = "Ms_Rupali"
```

```
# Rating columns for this teacher
rating_cols_Ms_Rupali = [col for col in df.columns if teacher_name in col and no

print("Rating Columns:", rating_cols_Ms_Rupali)
```

```
Rating Columns: ['Clarity_Ms_Rupali', 'SubKnowledge_Ms_Rupali', 'Interaction_Ms_R
upali', 'DoubtSolving_Ms_Rupali', 'Punctuality_Ms_Rupali']
```

In [39]:
```
teacher_name = "Mrs_Vaishali"

# Rating columns for this teacher
rating_cols_Mrs_Vaishali = [col for col in df.columns if teacher_name in col and

print("Rating Columns:", rating_cols_Mrs_Vaishali)
```

```
Rating Columns: ['Clarity_Mrs_Vaishali', 'SubKnowledge_Mrs_Vaishali', 'Interactio
n_Mrs_Vaishali', 'DoubtSolving_Mrs_Vaishali', 'Punctuality_Mrs_Vaishali']
```

For Average Rating

In [40]:
```
# Create a dictionary: teacher_name -> rating columns list
teacher_ratings = {
    "Prof_Seema": rating_cols_Prof_Seema,
    "Mrs_Sonam": rating_cols_Mrs_Sonam,
    "Mrs_Anita": rating_cols_Mrs_Anita,
    "Prof_Ashwini": rating_cols_Prof_Ashwini,
    "Mrs_Sakshi": rating_cols_Mrs_Sakshi,
    "Mr_Pradosh": rating_cols_Mr_Pradosh,
    "Ms_Maitri": rating_cols_Ms_Maitri,
    "Mrs_Priya": rating_cols_Mrs_Priya,
    "Ms_Abha": rating_cols_Ms_Abha,
    "Ms_Rupali": rating_cols_Ms_Rupali,
    "Mrs_Vaishali": rating_cols_Mrs_Vaishali,
}

# Dictionary to store each teacher's DataFrame
teacher_dfs = {}

print(teacher_ratings)
```

{'Prof_Seema': ['Clarity_Prof_Seema', 'SubKnowledge_Prof_Seema', 'Interaction_Prof_Seema', 'DoubtSolving_Prof_Seema', 'Punctuality_Prof_Seema'], 'Mrs_Sonam': ['Clarity_Mrs_Sonam', 'SubKnowledge_Mrs_Sonam', 'Interaction_Mrs_Sonam', 'DoubtSolving_Mrs_Sonam', 'Punctuality_Mrs_Sonam'], 'Mrs_Anita': ['Clarity_Mrs_Anita', 'SubKnowledge_Mrs_Anita', 'Interaction_Mrs_Anita', 'DoubtSolving_Mrs_Anita', 'Punctuality_Mrs_Anita'], 'Prof_Ashwini': ['Clarity_Prof_Ashwini', 'SubKnowledge_Prof_Ashwini', 'Interaction_Prof_Ashwini', 'DoubtSolving_Prof_Ashwini', 'Punctuality_Prof_Ashwini'], 'Mrs_Sakshi': ['Clarity_Mrs_Sakshi', 'SubKnowledge_Mrs_Sakshi', 'Interaction_Mrs_Sakshi', 'DoubtSolving_Mrs_Sakshi', 'Punctuality_Mrs_Sakshi'], 'Mr_Pradosh': ['Clarity_Mr_Pradosh', 'SubKnowledge_Mr_Pradosh', 'Interaction_Mr_Pradosh', 'DoubtSolving_Mr_Pradosh', 'Punctuality_Mr_Pradosh'], 'Ms_Maitri': ['Clarity_Ms_Maitri', 'SubKnowledge_Ms_Maitri', 'Interaction_Ms_Maitri', 'DoubtSolving_Ms_Maitri', 'Punctuality_Ms_Maitri'], 'Mrs_Priya': ['Clarity_Mrs_Priya', 'SubKnowledge_Mrs_Priya', 'Interaction_Mrs_Priya', 'DoubtSolving_Mrs_Priya', 'Punctuality_Mrs_Priya'], 'Ms_Abha': ['Clarity_Ms_Abha', 'SubKnowledge_Ms_Abha', 'Interaction_Ms_Abha', 'DoubtSolving_Ms_Abha', 'Punctuality_Ms_Abha'], 'Ms_Rupali': ['Clarity_Ms_Rupali', 'SubKnowledge_Ms_Rupali', 'Interaction_Ms_Rupali', 'DoubtSolving_Ms_Rupali', 'Punctuality_Ms_Rupali'], 'Mrs_Vaishali': ['Clarity_Mrs_Vaishali', 'SubKnowledge_Mrs_Vaishali', 'Interaction_Mrs_Vaishali', 'DoubtSolving_Mrs_Vaishali', 'Punctuality_Mrs_Vaishali']}

In [41]:
```
# Dictionary to store each teacher's DataFrame
teacher_dfs = {}

for teacher, rating_cols in teacher_ratings.items():
    # Create DataFrame with only rating columns
    teacher_df = df[rating_cols].copy()

    # Store in dictionary
    teacher_dfs[teacher] = teacher_df
```

In [42]:
```
# Prof Seema's ratings only
teacher_dfs['Prof_Seema'].head(20)
```

Out[42]:

	Clarity_Prof_Seema	SubKnowledge_Prof_Seema	Interaction_Prof_Seema	DoubtSolvin
0	4	5	4	
1	4	4	4	
2	4	3	4	
3	5	5	5	
4	4	5	4	
5	1	1	1	
6	5	5	5	
7	5	5	5	
8	5	5	4	
9	4	4	4	
10	4	4	4	
11	5	5	5	
12	5	5	5	
13	4	4	4	
14	5	5	5	
15	5	5	5	
16	5	5	5	
17	3	3	3	
18	5	5	5	
19	5	5	5	

In [43]:

```
#Initialize a dictionary to store averages
avg_ratings = {}
#Loop through each teacher's DataFrame
for teacher, t_df in teacher_dfs.items():
    # Compute overall average rating for this teacher
    avg_ratings[teacher] = t_df.mean().mean()
#View results
for teacher, avg in avg_ratings.items():
    print(f"{teacher}: {avg:.2f}")
```

```
Prof_Seema: 4.26
Mrs_Sonam: 2.95
Mrs_Anita: 3.63
Prof_Ashwini: 3.91
Mrs_Sakshi: 4.15
Mr_Pradosh: 4.02
Ms_Maitri: 3.64
Mrs_Priya: 3.93
Ms_Abha: 3.67
Ms_Rupali: 3.84
Mrs_Vaishali: 3.96
```

In [44]:
```
#Convert to DataFrame for easier plotting
avg_ratings_df = pd.DataFrame(list(avg_ratings.items()),columns=['Teacher','Aver
avg_ratings_df
```

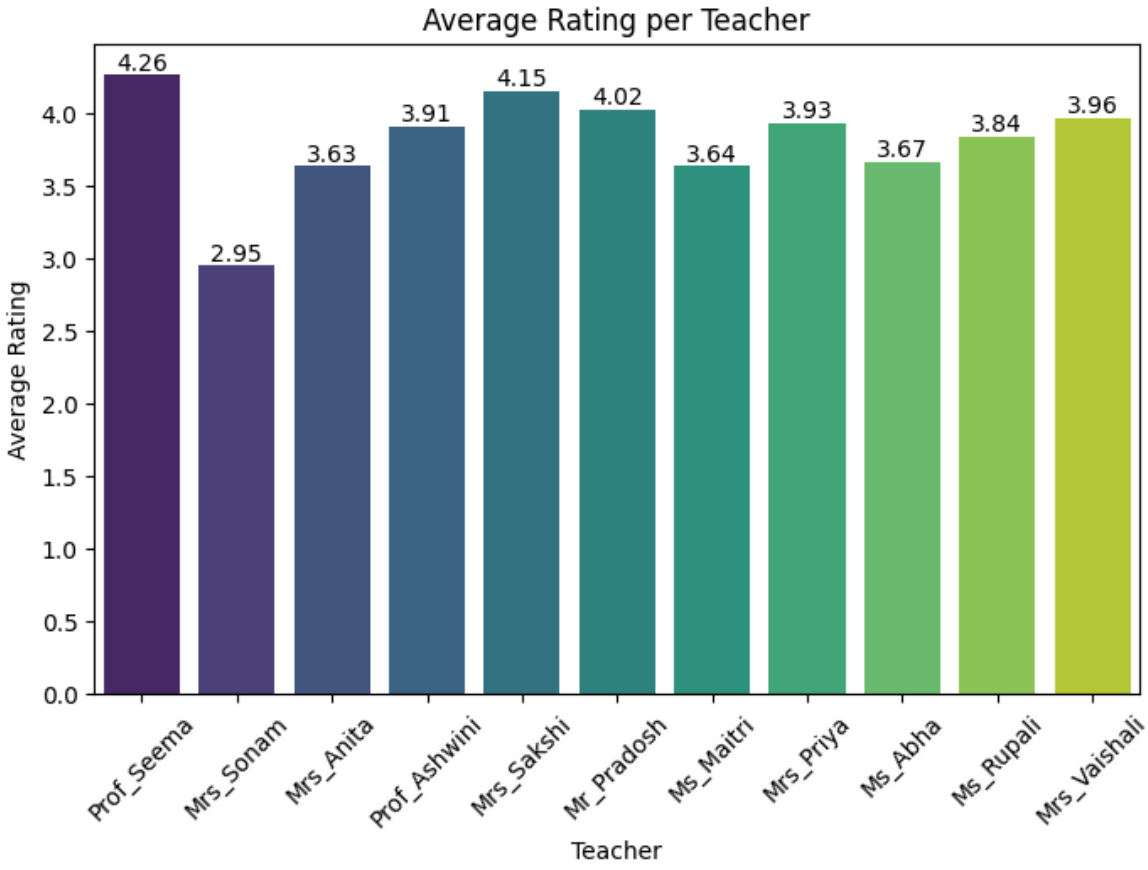
Out[44]:

	Teacher	Average_Rating
0	Prof_Seema	4.262069
1	Mrs_Sonam	2.950246
2	Mrs_Anita	3.633744
3	Prof_Ashwini	3.910345
4	Mrs_Sakshi	4.153448
5	Mr_Pradosh	4.021182
6	Ms_Maitri	3.635760
7	Mrs_Priya	3.930049
8	Ms_Abha	3.666010
9	Ms_Rupali	3.839409
10	Mrs_Vaishali	3.963547

Visualizing Average Ratings

In [45]:
```
plt.figure(figsize=(8,5))
ax = sns.barplot(x="Teacher",y="Average_Rating",data=avg_ratings_df,palette='vir
plt.title("Average Rating per Teacher")
plt.ylabel("Average Rating")
plt.xticks(rotation=45)


# Add data labels on top of each bar
for bars in ax.containers:
    ax.bar_label(bars,fmt='%.2f')

plt.show()
```

```
C:\Users\chaud\AppData\Local\Temp\ipykernel_14580\2330293619.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  ax = sns.barplot(x="Teacher",y="Average_Rating",data=avg_ratings_df,palette='vi
ridis')
```



- Prof. Seema has the highest average rating among all teachers, which indicates strong student satisfaction with her teaching.

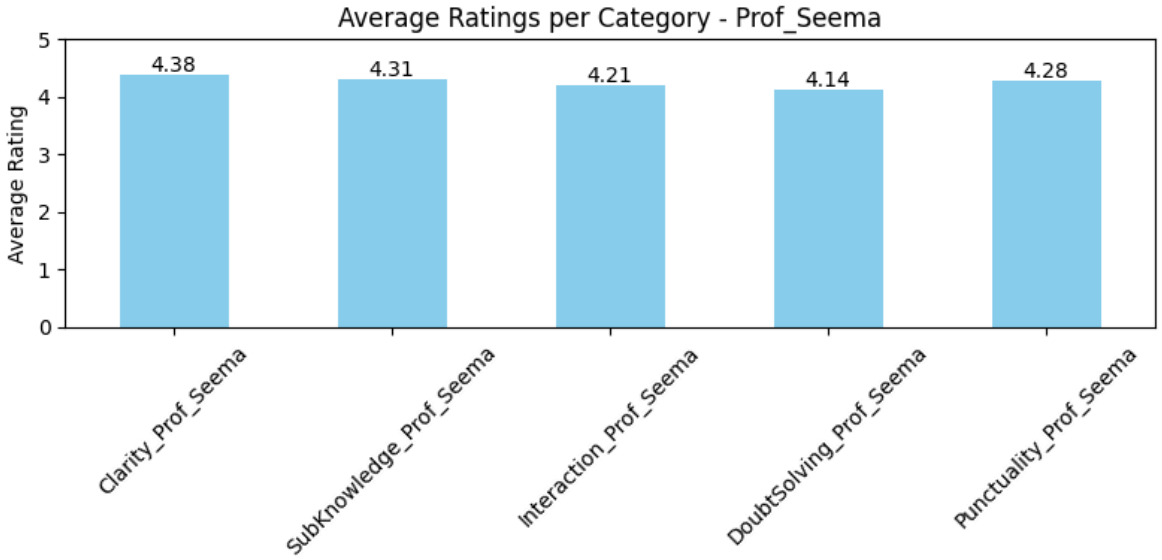
Calculating Average Rating per Category for Each Teacher

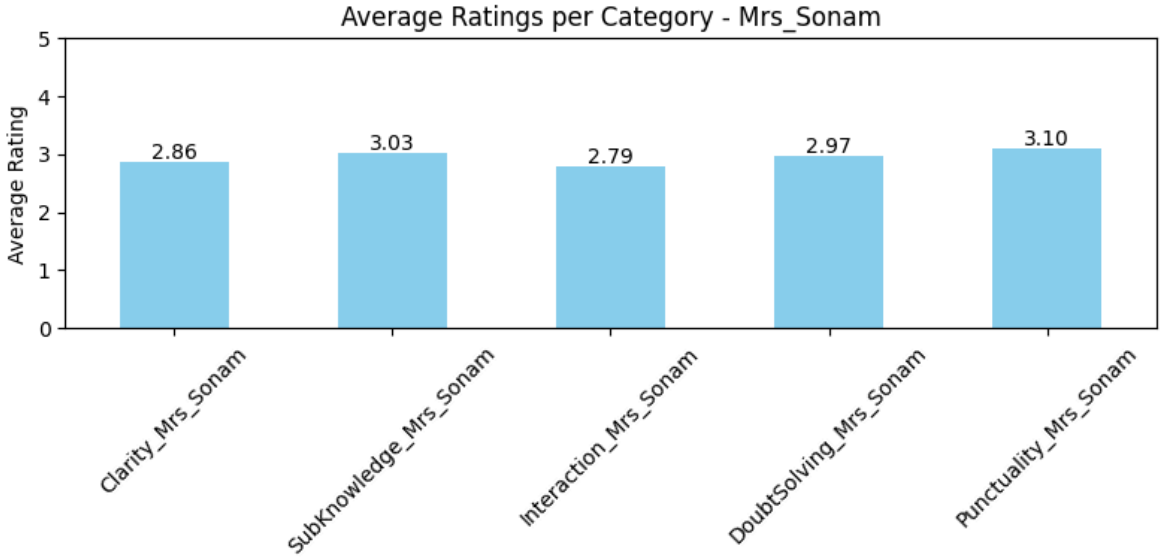
In [46]:
```
for teacher, t_df in teacher_dfs.items():
    if not t_df.empty:  # Only process if teacher has rating columns
        # Average per category
        avg_per_category = t_df.mean()

        # Display the averages
        print(f"\nAverage ratings for {teacher}:\n", avg_per_category)
```

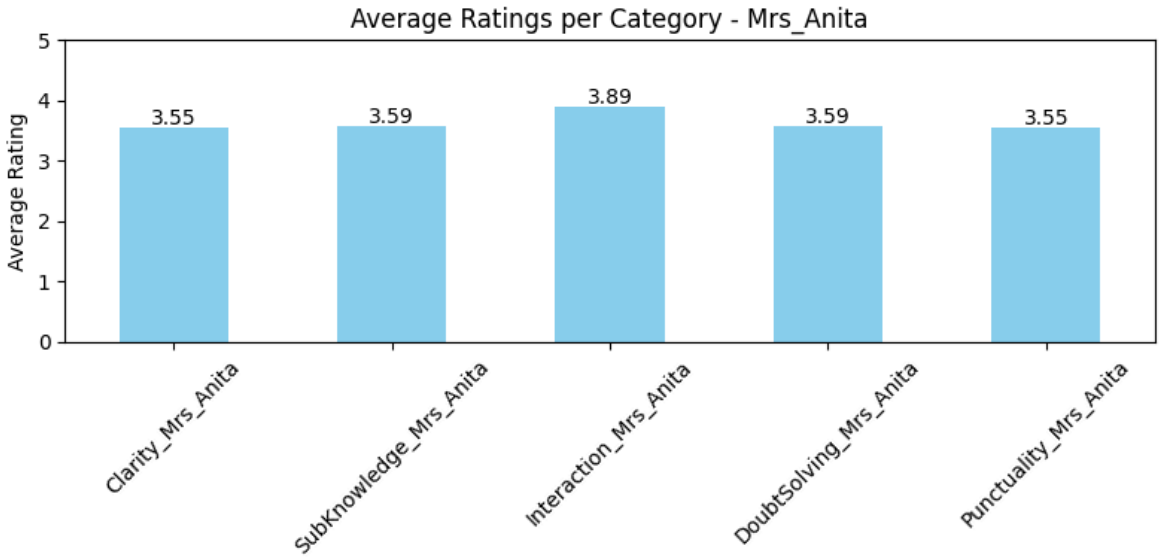
```
Average ratings for Prof_Seema:
 Clarity_Prof_Seema         4.379310
SubKnowledge_Prof_Seema    4.310345
Interaction_Prof_Seema     4.206897
DoubtSolving_Prof_Seema    4.137931
Punctuality_Prof_Seema     4.275862
dtype: float64

Average ratings for Mrs_Sonam:
 Clarity_Mrs_Sonam         2.862069
SubKnowledge_Mrs_Sonam    3.034483
Interaction_Mrs_Sonam     2.785714
DoubtSolving_Mrs_Sonam    2.965517
Punctuality_Mrs_Sonam     3.103448
dtype: float64

Average ratings for Mrs_Anita:
 Clarity_Mrs_Anita         3.551724
SubKnowledge_Mrs_Anita    3.586207
Interaction_Mrs_Anita     3.892857
DoubtSolving_Mrs_Anita    3.586207
Punctuality_Mrs_Anita     3.551724
dtype: float64

Average ratings for Prof_Ashwini:
 Clarity_Prof_Ashwini         3.896552
SubKnowledge_Prof_Ashwini    4.034483
Interaction_Prof_Ashwini     3.689655
DoubtSolving_Prof_Ashwini    3.896552
Punctuality_Prof_Ashwini     4.034483
dtype: float64

Average ratings for Mrs_Sakshi:
 Clarity_Mrs_Sakshi         4.068966
SubKnowledge_Mrs_Sakshi    4.172414
Interaction_Mrs_Sakshi     4.250000
DoubtSolving_Mrs_Sakshi    4.068966
Punctuality_Mrs_Sakshi     4.206897
dtype: float64

Average ratings for Mr_Pradosh:
 Clarity_Mr_Pradosh         4.137931
SubKnowledge_Mr_Pradosh    3.931034
Interaction_Mr_Pradosh     4.071429
DoubtSolving_Mr_Pradosh    3.965517
Punctuality_Mr_Pradosh     4.000000
dtype: float64

Average ratings for Ms_Maitri:
 Clarity_Ms_Maitri         3.586207
SubKnowledge_Ms_Maitri    3.689655
Interaction_Ms_Maitri     3.592593
DoubtSolving_Ms_Maitri    3.655172
Punctuality_Ms_Maitri     3.655172
dtype: float64

Average ratings for Mrs_Priya:
 Clarity_Mrs_Priya         3.862069
SubKnowledge_Mrs_Priya    3.965517
Interaction_Mrs_Priya     3.857143
```

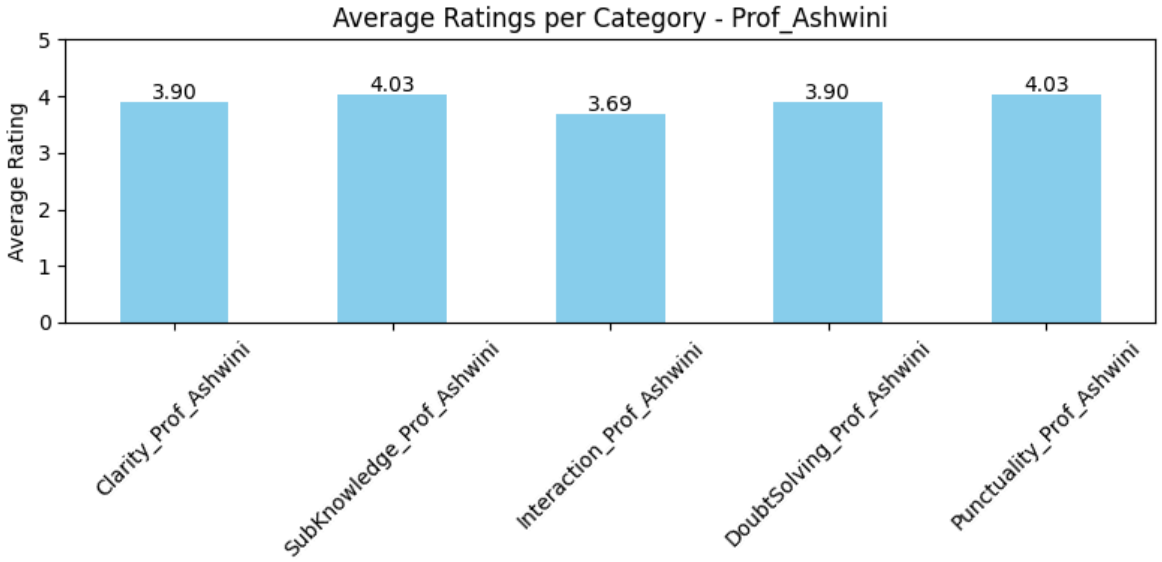
```
DoubtSolving_Mrs_Priya    3.965517
Punctuality_Mrs_Priya     4.000000
dtype: float64

Average ratings for Ms_Abha:
 Clarity_Ms_Abha         3.655172
SubKnowledge_Ms_Abha    3.689655
Interaction_Ms_Abha     3.571429
DoubtSolving_Ms_Abha    3.689655
Punctuality_Ms_Abha     3.724138
dtype: float64

Average ratings for Ms_Rupali:
 Clarity_Ms_Rupali         3.689655
SubKnowledge_Ms_Rupali    3.965517
Interaction_Ms_Rupali     3.714286
DoubtSolving_Ms_Rupali    3.965517
Punctuality_Ms_Rupali     3.862069
dtype: float64

Average ratings for Mrs_Vaishali:
 Clarity_Mrs_Vaishali         3.931034
SubKnowledge_Mrs_Vaishali    4.172414
Interaction_Mrs_Vaishali     3.714286
DoubtSolving_Mrs_Vaishali    3.758621
Punctuality_Mrs_Vaishali     4.241379
dtype: float64
```

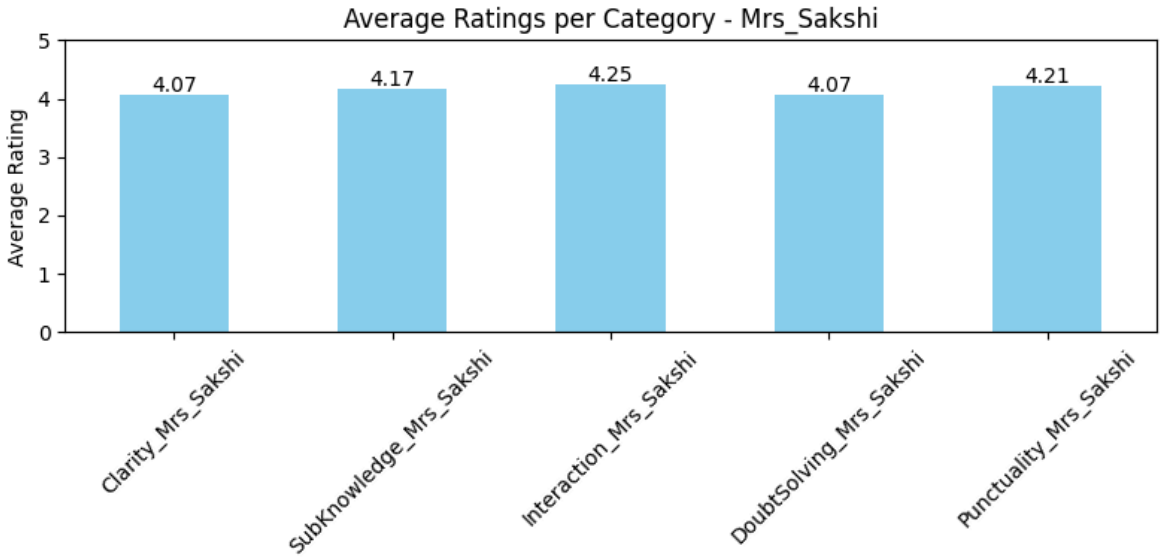
In [47]:
```python
for teacher, t_df in teacher_dfs.items():
    if not t_df.empty:
        avg_per_category = t_df.mean()

        # Plot
        plt.figure(figsize=(8,4))
        avg_per_category.plot(kind='bar', color='skyblue')
        plt.title(f"Average Ratings per Category - {teacher}")
        plt.ylabel("Average Rating")
        plt.ylim(0,5)  # Assuming rating scale 0-5
        plt.xticks(rotation=45)

        # Add data labels on top
        for i, v in enumerate(avg_per_category):
            plt.text(i, v + 0.05, f"{v:.2f}", ha='center', fontsize=10)

        plt.tight_layout()
        plt.show()
```

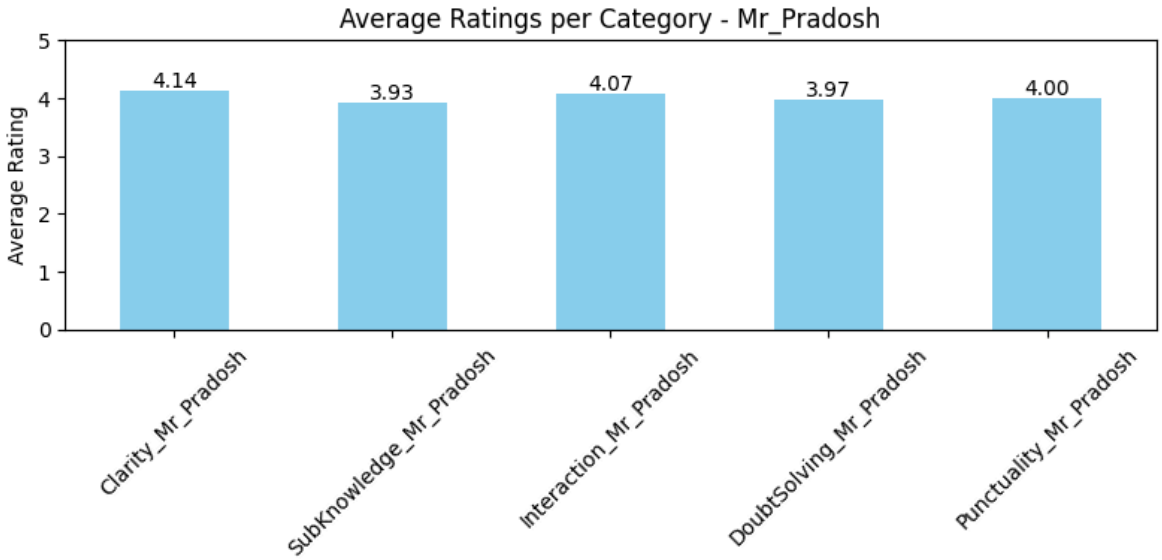

Average Ratings per Category - Prof_Seema
Average Rating
0
1
2
3
4
5
4.38
4.31
4.21
4.14
4.28
Clarity_Prof_Seema
SubKnowledge_Prof_Seema
Interaction_Prof_Seema
DoubtSolving_Prof_Seema
Punctuality_Prof_Seema

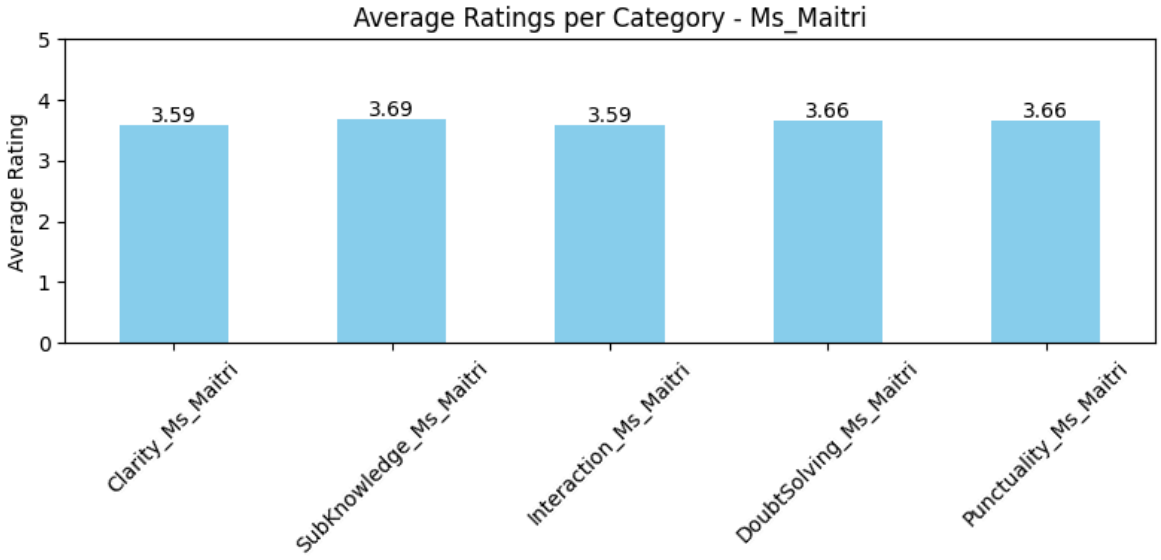

Average Ratings per Category - Mrs_Sonam
Average Rating
0
1
2
3
4
5
2.86
3.03
2.79
2.97
3.10
Clarity_Mrs_Sonam
SubKnowledge_Mrs_Sonam
Interaction_Mrs_Sonam
DoubtSolving_Mrs_Sonam
Punctuality_Mrs_Sonam

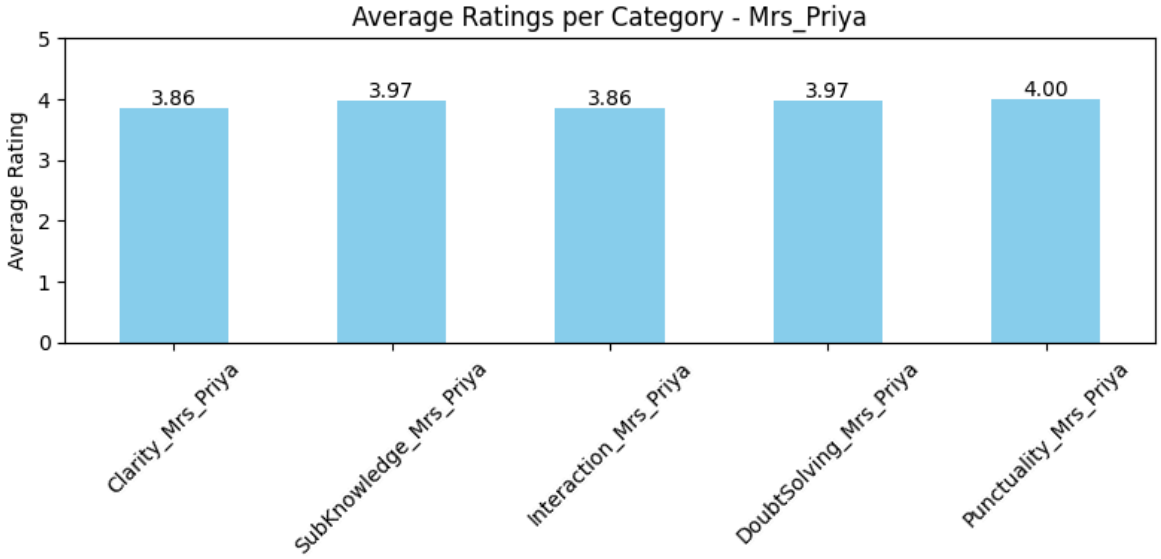

Average Ratings per Category - Mrs_Anita
Average Rating
0
1
2
3
4
5
3.55
3.59
3.89
3.59
3.55
Clarity_Mrs_Anita
SubKnowledge_Mrs_Anita
Interaction_Mrs_Anita
DoubtSolving_Mrs_Anita
Punctuality_Mrs_Anita

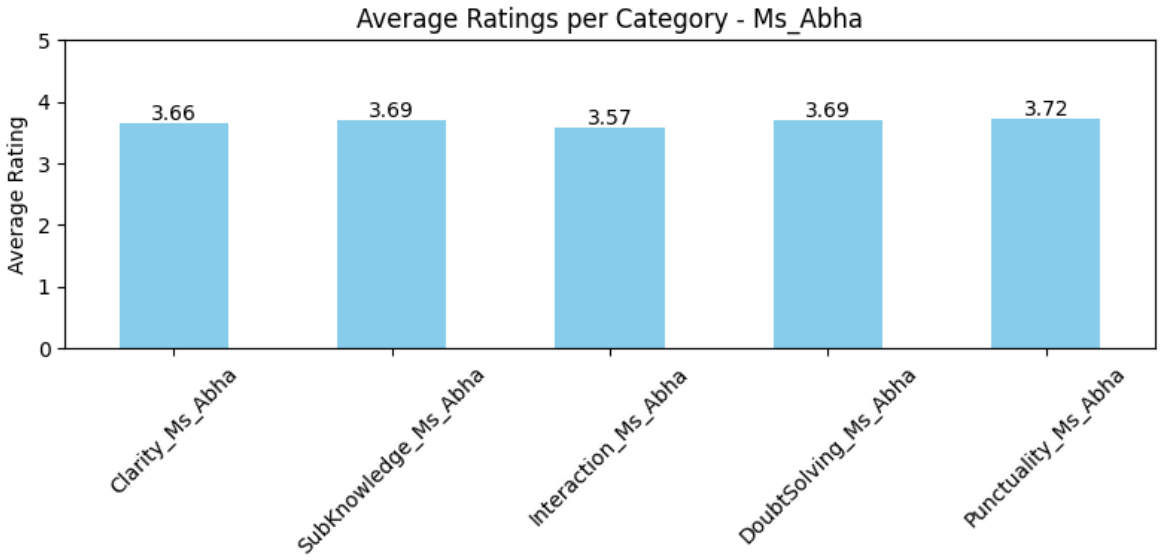

Average Ratings per Category - Prof_Ashwini
Average Rating
0
1
2
3
4
5
3.90
4.03
3.69
3.90
4.03
Clarity_Prof_Ashwini
SubKnowledge_Prof_Ashwini
Interaction_Prof_Ashwini
DoubtSolving_Prof_Ashwini
Punctuality_Prof_Ashwini

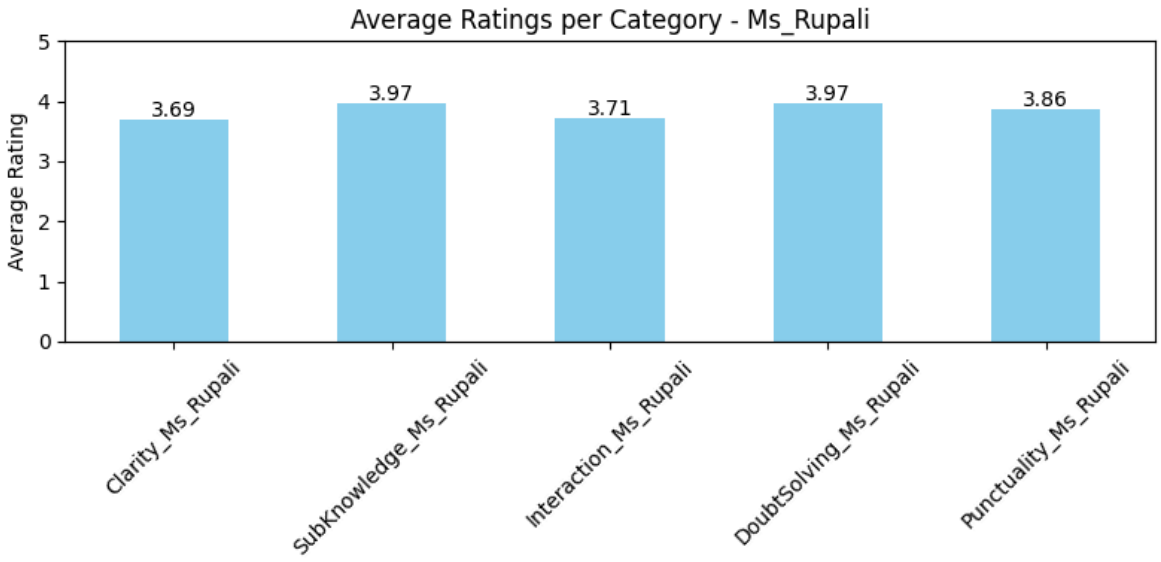

Average Ratings per Category - Mrs_Sakshi
Average Rating
0
1
2
3
4
5
4.07
4.17
4.25
4.07
4.21
Clarity_Mrs_Sakshi
SubKnowledge_Mrs_Sakshi
Interaction_Mrs_Sakshi
DoubtSolving_Mrs_Sakshi
Punctuality_Mrs_Sakshi

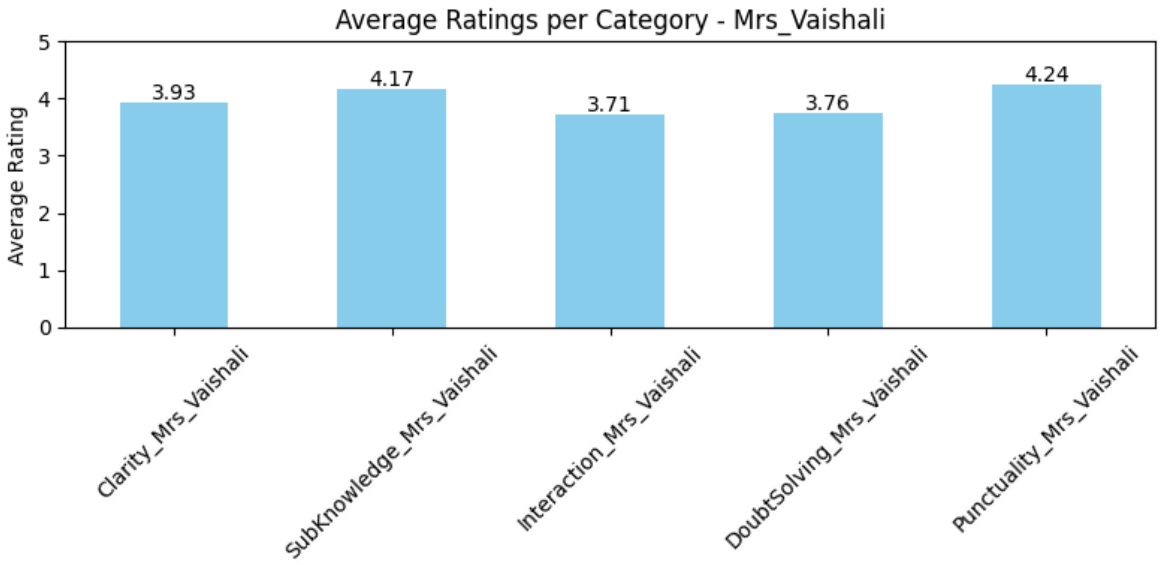

Average Ratings per Category - Mr_Pradosh
Average Rating
0
1
2
3
4
5
4.14
3.93
4.07
3.97
4.00
Clarity_Mr_Pradosh
SubKnowledge_Mr_Pradosh
Interaction_Mr_Pradosh
DoubtSolving_Mr_Pradosh
Punctuality_Mr_Pradosh


Average Ratings per Category - Ms_Maitri
Average Rating
0
1
2
3
4
5
3.59
3.69
3.59
3.66
3.66
Clarity_Ms_Maitri
SubKnowledge_Ms_Maitri
Interaction_Ms_Maitri
DoubtSolving_Ms_Maitri
Punctuality_Ms_Maitri


Average Ratings per Category - Mrs_Priya
Average Rating
0
1
2
3
4
5
3.86
3.97
3.86
3.97
4.00
Clarity_Mrs_Priya
SubKnowledge_Mrs_Priya
Interaction_Mrs_Priya
DoubtSolving_Mrs_Priya
Punctuality_Mrs_Priya


Average Ratings per Category - Ms_Abha
Average Rating
0
1
2
3
4
5
3.66
3.69
3.57
3.69
3.72
Clarity_Ms_Abha
SubKnowledge_Ms_Abha
Interaction_Ms_Abha
DoubtSolving_Ms_Abha
Punctuality_Ms_Abha



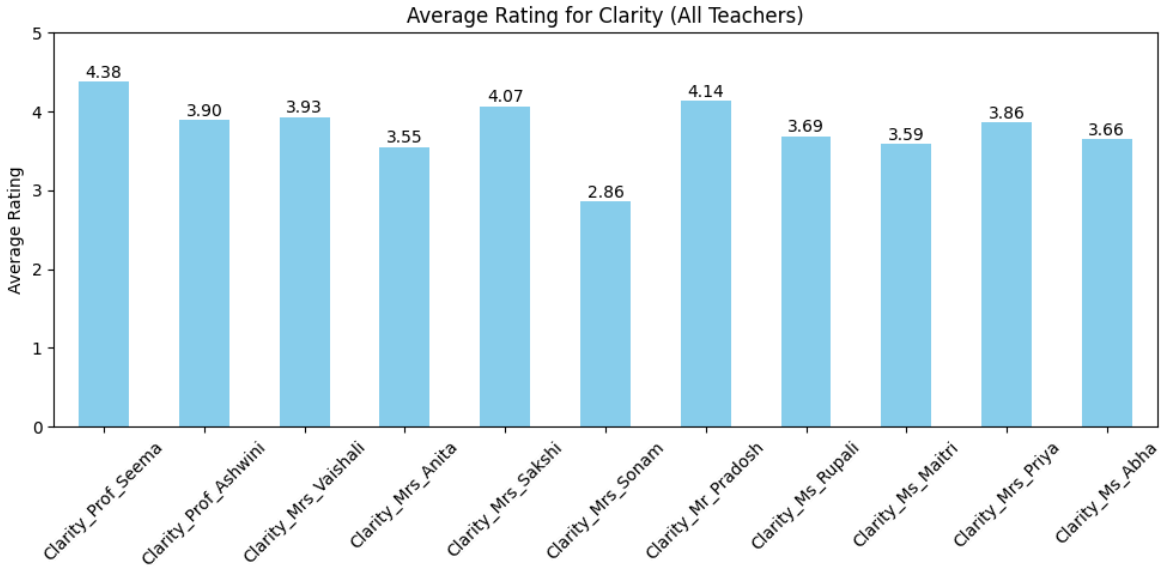


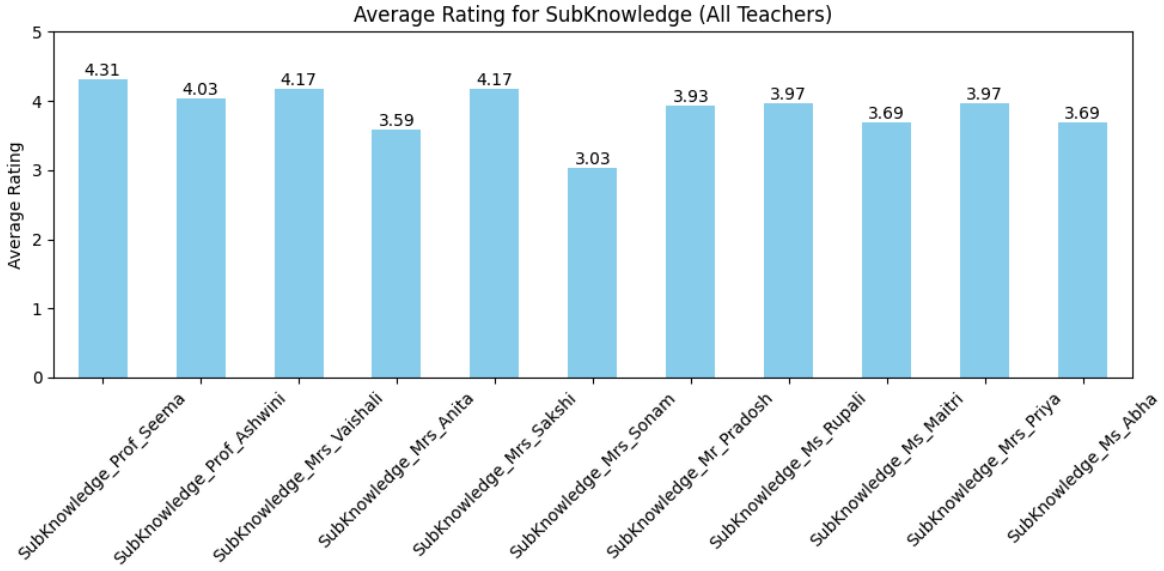
- Prof. Seema and Ms. Sakshi show consistent excellence with ratings above 4 across all categories, reflecting strong overall teaching quality. In contrast, Ms. Sonam's ratings are mostly below 3, indicating student dissatisfaction across multiple aspects.

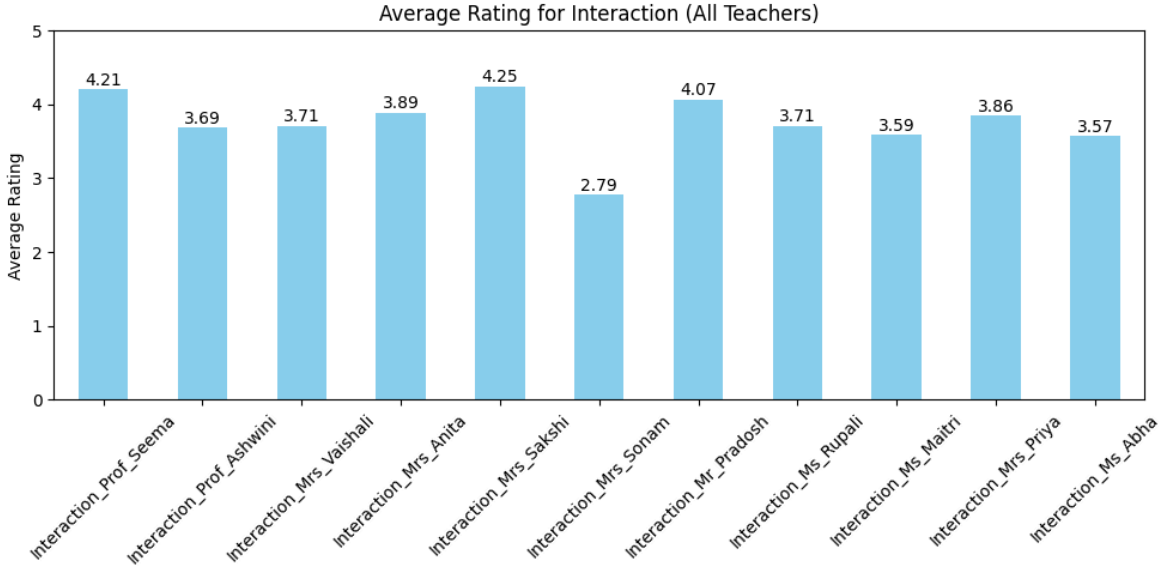
Calculating Average Rating per Teacher for Each Category

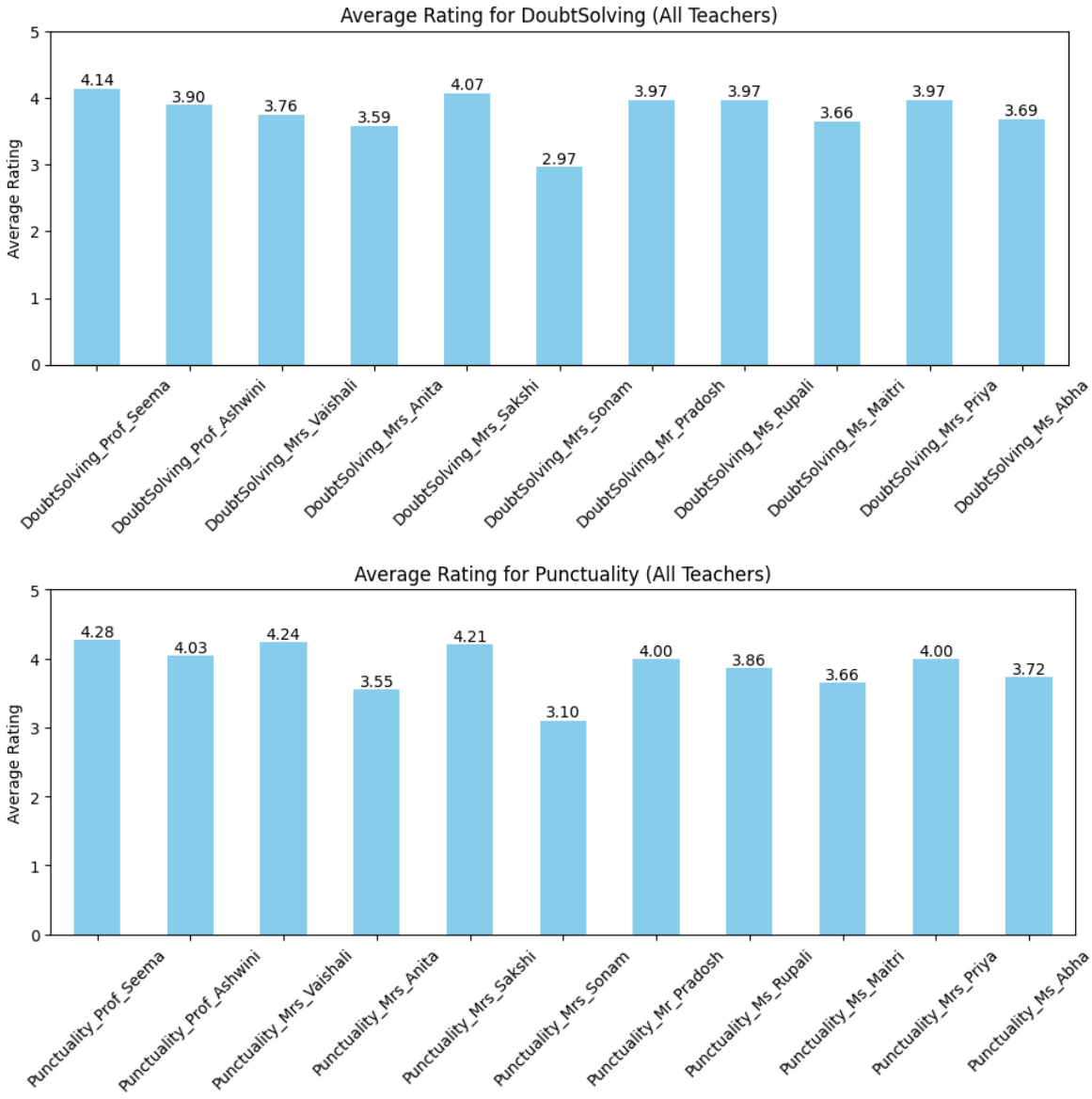
In [48]:
```
categories = ["Clarity", "SubKnowledge", "Interaction", "DoubtSolving", "Punctua

for cat in categories:
    cat_cols = [col for col in df.columns if col.startswith(cat)]
    cat_avg = df[cat_cols].mean()

    plt.figure(figsize=(10,5))
    cat_avg.plot(kind='bar', color='skyblue')
    plt.title(f"Average Rating for {cat} (All Teachers)")
    plt.ylabel("Average Rating")
    plt.ylim(0,5)
    plt.xticks(rotation=45)
```

```
    for i, v in enumerate(cat_avg):
        plt.text(i, v + 0.05, f"{v:.2f}", ha='center', fontsize=10)

    plt.tight_layout()
    plt.show()
```









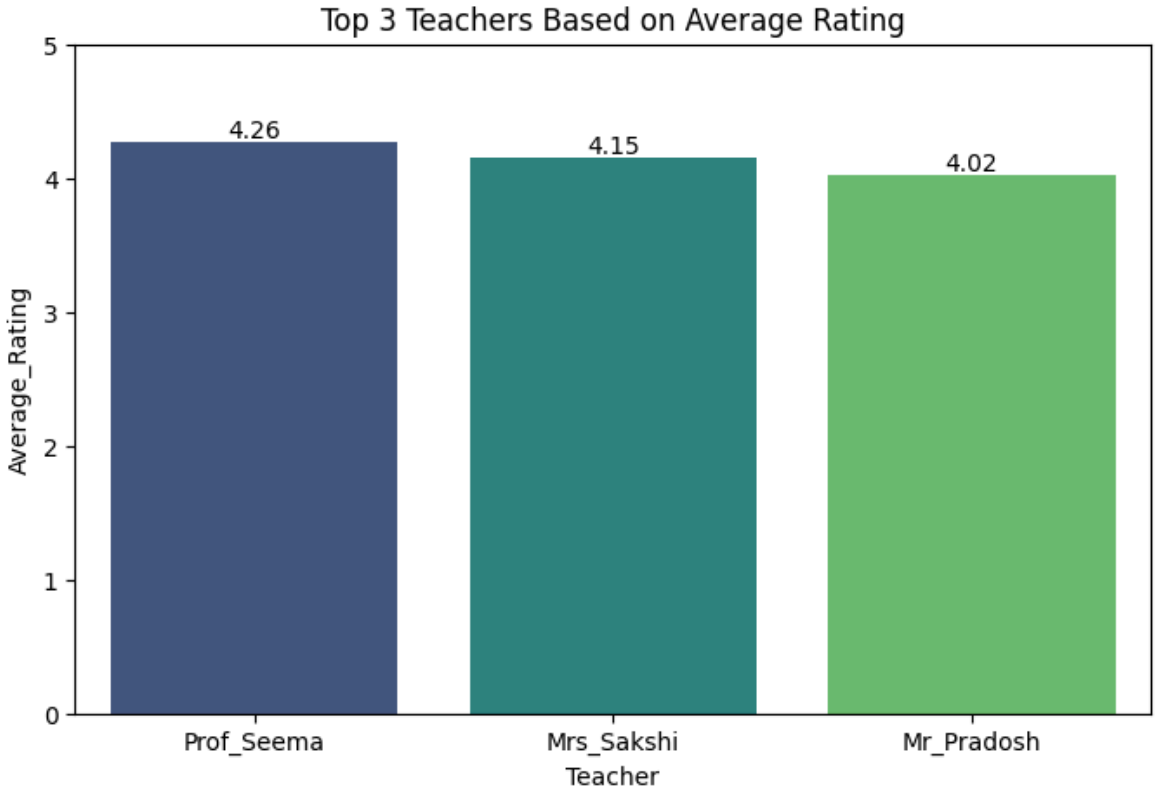
- Prof. Seema leads in most categories, showing excellence in subject knowledge, doubt solving, and punctuality, making her the top-performing teacher overall. Mr. Prashosh, Prof. Ashwini, and Ms. Sakshi perform strongly in specific areas like subject knowledge and interaction, reflecting their unique strengths. Ms. Sakshi particularly excels in interaction, highlighting her strong student engagement compared to peers.

Top 3 Teachers

In [52]:
```
#: Sort and get top 3 teachers
top3_teachers = avg_ratings_df.sort_values(by="Average_Rating",ascending=False).
top3_teachers

#Visualize top 3 teachers
plt.figure(figsize=(8,5))
ax = sns.barplot(x="Teacher",y="Average_Rating",data=top3_teachers,palette='viri
plt.title("Top 3 Teachers Based on Average Rating")
plt.ylim(0,5)
```

```
#adding data labels
for bars in ax.containers:
    ax.bar_label(bars,fmt="%.2f")

plt.show()
```

```
C:\Users\chaud\AppData\Local\Temp\ipykernel_14580\4184670461.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  ax = sns.barplot(x="Teacher",y="Average_Rating",data=top3_teachers,palette='vir
idis')
```

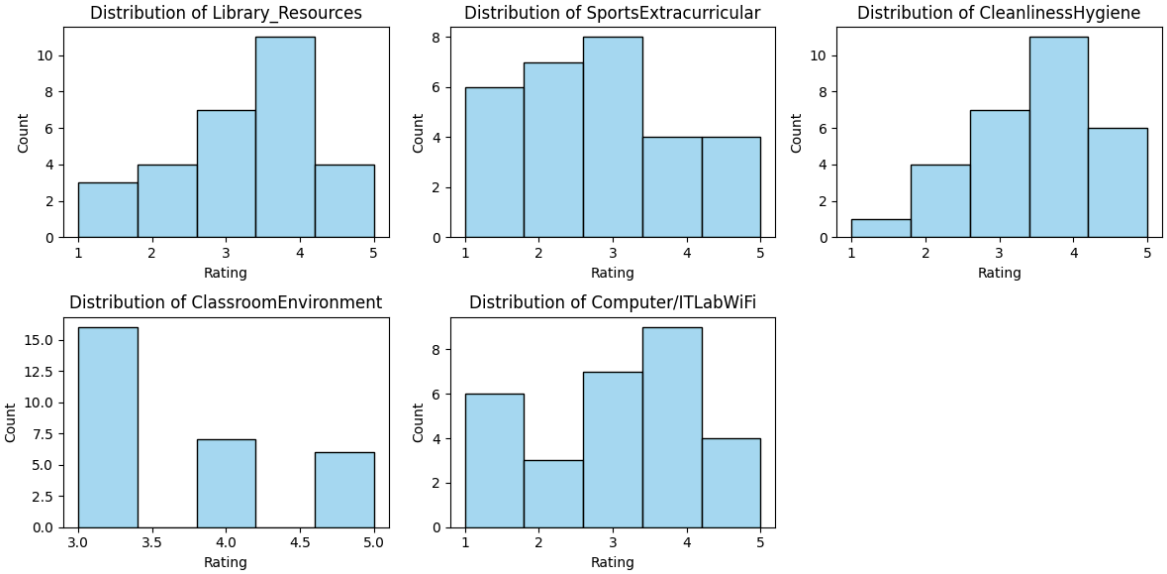


- Prof. Seema ranks first with the highest overall performance across most categories. Ms. Sakshi secures second place with strong ratings, especially in interaction and doubt solving. Mr. Pradosh stands third, performing well in overall teaching quality and subject knowledge.

Visualizing Distribution per Facility

In [53]:
```
import matplotlib.pyplot as plt
import seaborn as sns

facilities = ['Library_Resources','SportsExtracurricular','CleanlinessHygiene','

plt.figure(figsize=(12,6))
for i, col in enumerate(facilities,1):
```

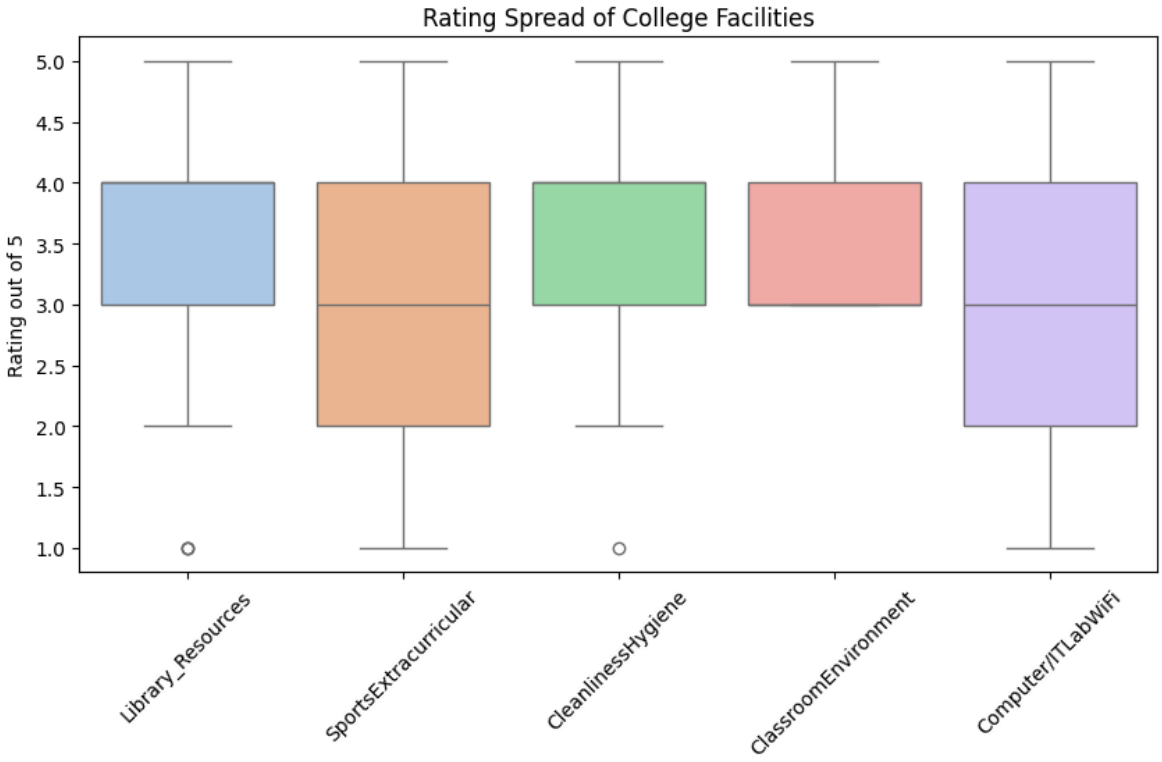
```
    plt.subplot(2,3,i)
    sns.histplot(df[col], bins=5, kde=False, color='skyblue')
    plt.title(f'Distribution of {col}')
    plt.xlabel('Rating')
    plt.ylabel('Count')
plt.tight_layout()
plt.show()
```



- Students rate the library, cleanliness, and computer lab highest (4), while sports and classroom environment lag (3–4), showing academic facilities are strong but extracurricular and learning spaces need improvement.

Boxplots to Check Spread and Outliers for Facilities

In [54]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df[facilities], palette='pastel')
plt.title("Rating Spread of College Facilities")
plt.ylabel("Rating out of 5")
plt.xticks(rotation=45)
plt.show()
```

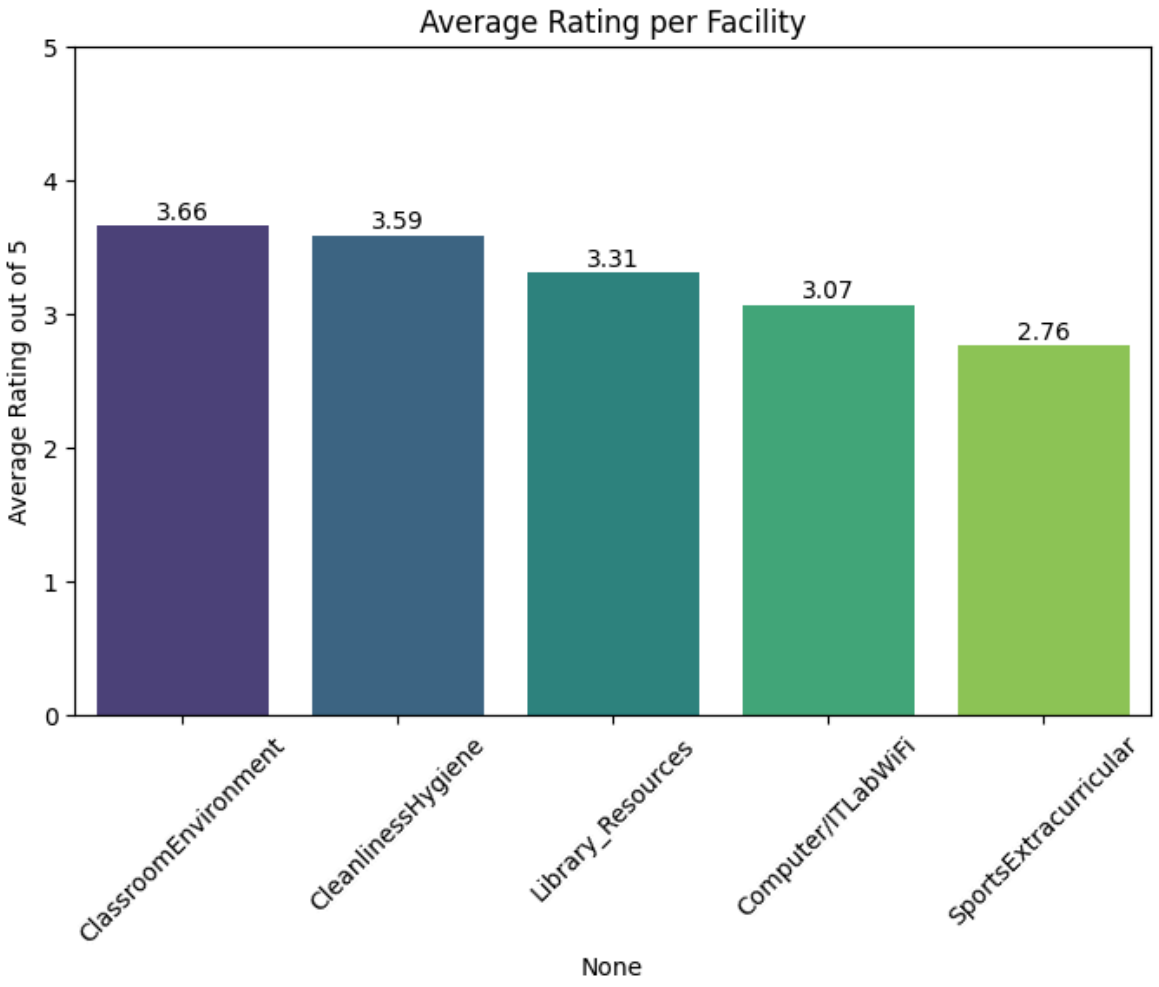


- Sports and extracurricular activities show high variability in ratings, indicating mixed student opinions, while classroom environment ratings are consistent, reflecting general agreement on its quality.

Average Rating per Facility

In [56]:
```
avg_facility_rating = df[facilities].mean().sort_values(ascending=False)
avg_facility_rating


plt.figure(figsize=(8,5))
sns.barplot(x=avg_facility_rating.index, y=avg_facility_rating.values, palette='
plt.title("Average Rating per Facility")
plt.ylabel("Average Rating out of 5")
plt.ylim(0,5)

# Add data labels
for i, v in enumerate(avg_facility_rating.values):
    plt.text(i, v + 0.05, f"{v:.2f}", ha='center', fontsize=10)

plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\chaud\AppData\Local\Temp\ipykernel_14580\3154315228.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=avg_facility_rating.index, y=avg_facility_rating.values, palette
='viridis')
```

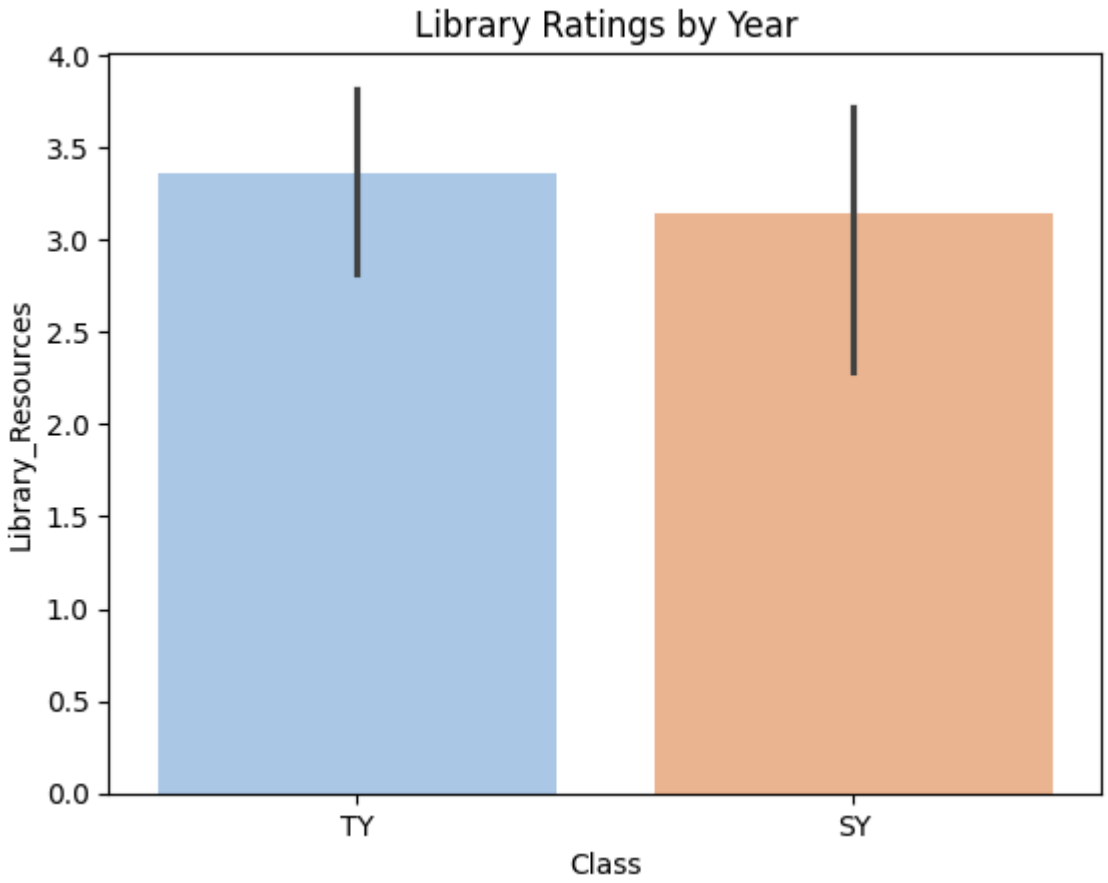


- Classroom environment has the highest average rating, followed by cleanliness and hygiene, while sports and extracurricular activities receive the lowest, highlighting strong academic facilities but weaker extracurricular support.

Grouped Analysis for Library Resources

In [57]:
```
sns.barplot(x='Class', y='Library_Resources', data=df, palette='pastel')
plt.title("Library Ratings by Year")
plt.show()
```

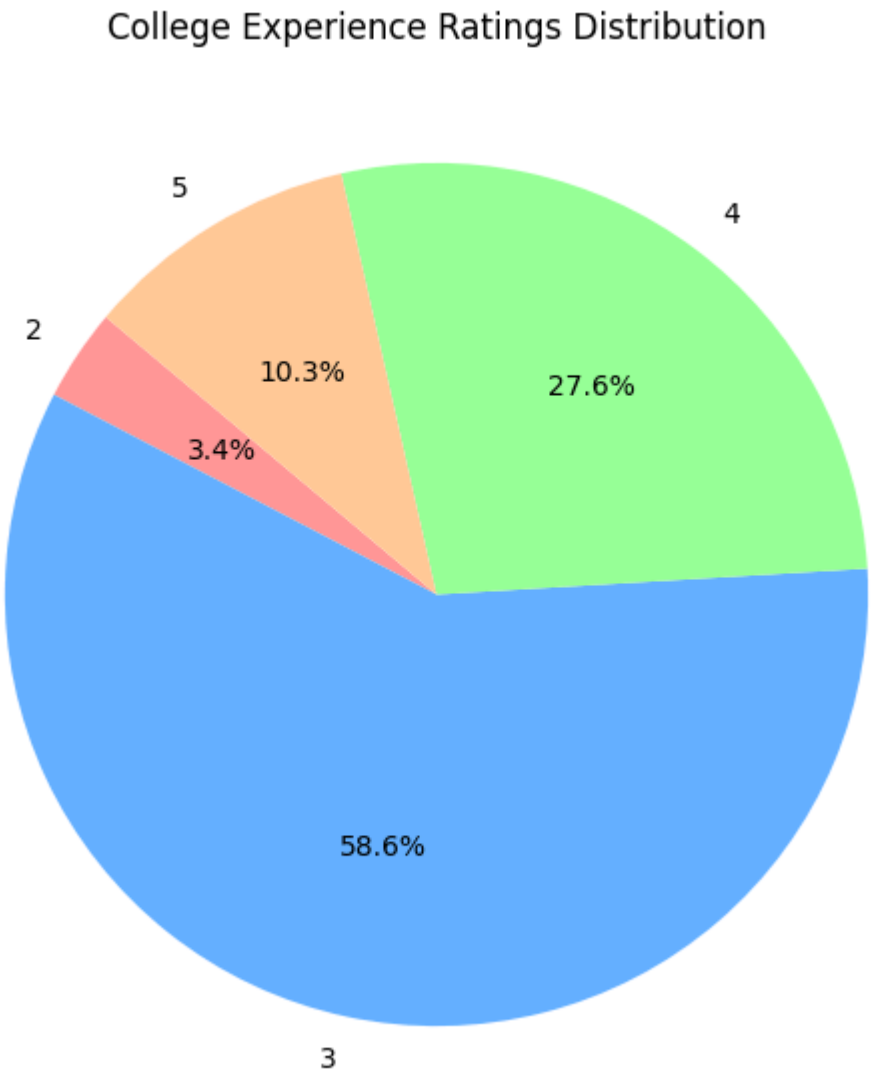
```
C:\Users\chaud\AppData\Local\Temp\ipykernel_14580\4005766780.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x='Class', y='Library_Resources', data=df, palette='pastel')
```



- Library resources are rated slightly higher by TY students compared to SY students, indicating that senior students find the resources more adequate or accessible.

College Experience

In [58]:
```
# Count number of students per rating
rating_counts = df['CollegeExperience'].value_counts().sort_index()
rating_counts

plt.figure(figsize=(7,7))
plt.pie(rating_counts,
        labels=rating_counts.index,
        autopct='%1.1f%%',   # shows percentage
        startangle=140,      # rotates chart for better view
        colors=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0'])

plt.title("College Experience Ratings Distribution")
plt.show()
```



- Most students have an average perception of their college experience, with a notable portion rating it positively, while very few express dissatisfaction, indicating generally moderate satisfaction.

Sentiment analysis for College Experience

In [ ]:
```
#Create a new column for sentiment
```

In [59]:
```
def experience_sentiment(rating):
    if rating <= 2:
        return 'Negative'
    elif rating == 3:
        return 'Neutral'
    else:  # 4 or 5
        return 'Positive'

# Apply function to the column
df['College_Experience_Sentiment'] = df['CollegeExperience'].apply(experience_se
```

```
# Check result
df[['CollegeExperience','College_Experience_Sentiment']].head(10)
```

Out[59]:

	CollegeExperience	College_Experience_Sentiment
0	3	Neutral
1	3	Neutral
2	3	Neutral
3	3	Neutral
4	4	Positive
5	3	Neutral
6	5	Positive
7	4	Positive
8	3	Neutral
9	3	Neutral

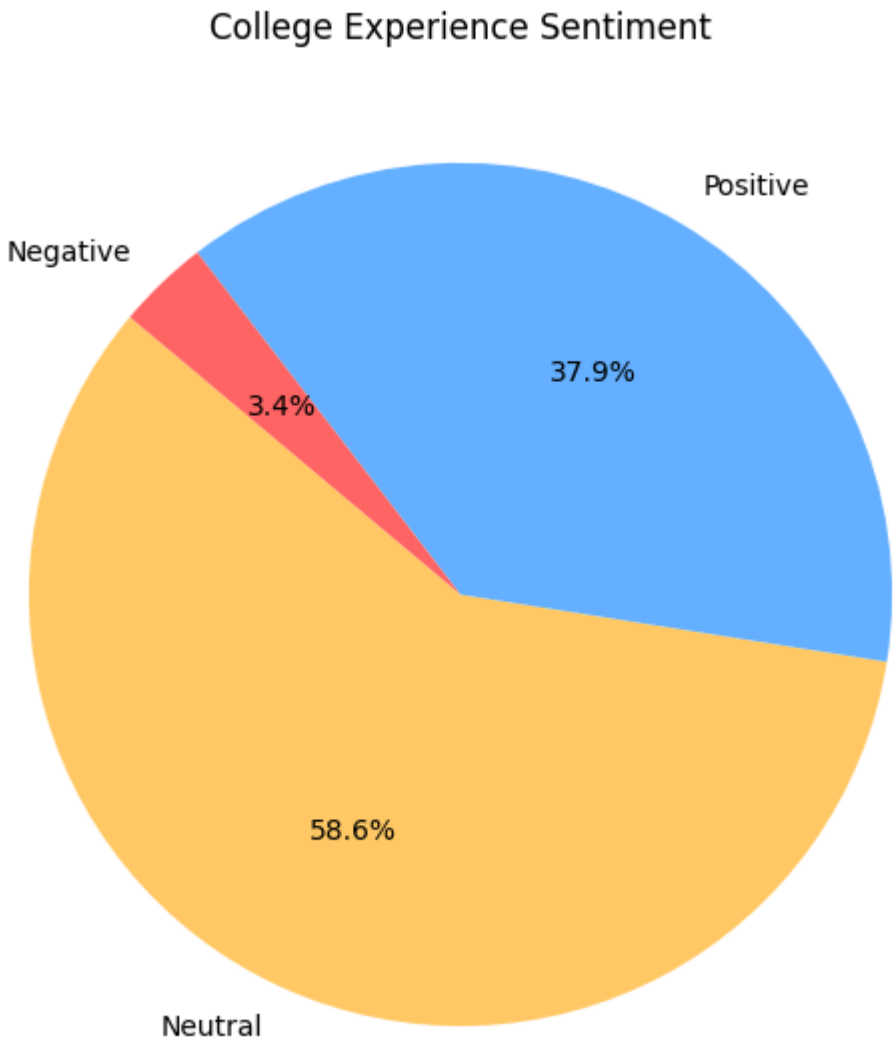
In [ ]:
```
#Count Positive / Neutral / Negative
```

In [60]:
```
sentiment_counts = df['College_Experience_Sentiment'].value_counts()
sentiment_counts
```

Out[60]:
```
College_Experience_Sentiment
Neutral     17
Positive    11
Negative     1
Name: count, dtype: int64
```

Visualizing sentiment (Pie Chart)

In [61]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(7,7))
plt.pie(sentiment_counts,
        labels=sentiment_counts.index,
        autopct='%1.1f%%',
        startangle=140,
        colors=['#ffcc66','#66b3ff','#ff6666']
        )  # Negative, Neutral, Positive


plt.title("College Experience Sentiment")
plt.show()
```

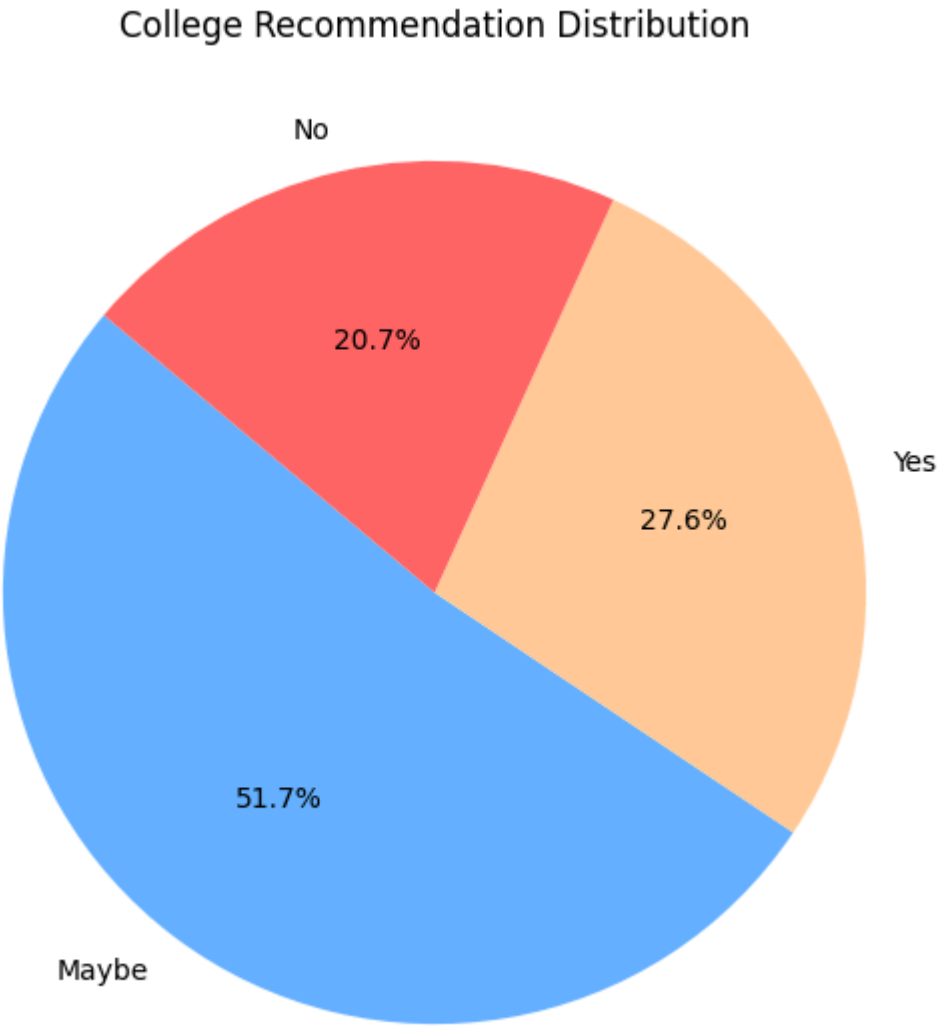


- Most students have a neutral college experience, a significant portion report a positive experience, and very few express a negative experience, indicating overall satisfactory student perception.

Overview of Recommendation Data

In [62]:
```
# Count how many students gave each response
recommendation_counts = df['CollegeRecommendation'].value_counts()
recommendation_counts

recommendation_percentage = df['CollegeRecommendation'].value_counts(normalize=T
recommendation_percentage

plt.figure(figsize=(7,7))
plt.pie(recommendation_counts,
        labels=recommendation_counts.index,
        autopct='%1.1f%%',
        startangle=140,
        colors=['#66b3ff','#ffcc99','#ff6666'])  # Yes, Maybe, No
```

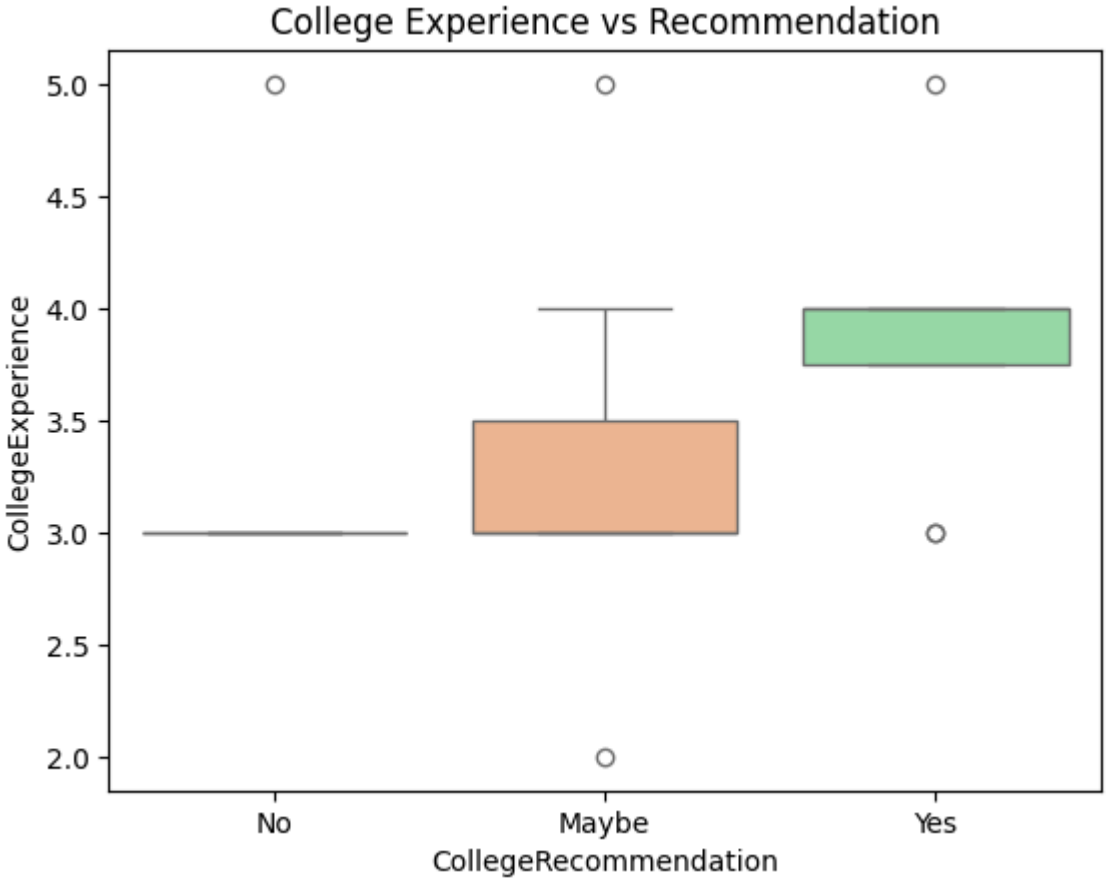
```
plt.title("College Recommendation Distribution")
plt.show()
```



- Most students are uncertain about recommending the college, with a majority choosing "maybe," some saying "yes," and a smaller portion saying "no," indicating mixed perceptions about endorsing the college.

In [63]:
```
# Example: College Experience vs Recommendation
sns.boxplot(x='CollegeRecommendation', y='CollegeExperience', data=df, palette='
plt.title("College Experience vs Recommendation")
plt.show()
```

```
C:\Users\chaud\AppData\Local\Temp\ipykernel_14580\2154549287.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x='CollegeRecommendation', y='CollegeExperience', data=df, palette
='pastel')
```



- Most students are uncertain about recommending the college, with a majority choosing "maybe," some saying "yes," and a smaller portion saying "no," indicating mixed perceptions about endorsing the college.

Sentiment Analysis for Comments for Teacher

In [65]:
```
# combining all comment columns for teachers
All_Comments = ['Comments_Prof_Seema','Comments_Mrs_Sonam','Comments_Mrs_Anita',
                'Comments_Mr_Pradosh','Comments_Ms_Maitri','Comments_Mrs_Priya','
                 'Comments_Mrs_Vaishali'
                ]

# Replace missing values with empty strings
for comment in All_Comments:
    df[comment] = df[comment].fillna("")

for c in All_Comments:
    df[c].isnull()
```

In [ ]:
```
#Define Cleaning Function
```

In [66]:
```
def clean_text(text):
    text = text.lower()  # Lowercase
```

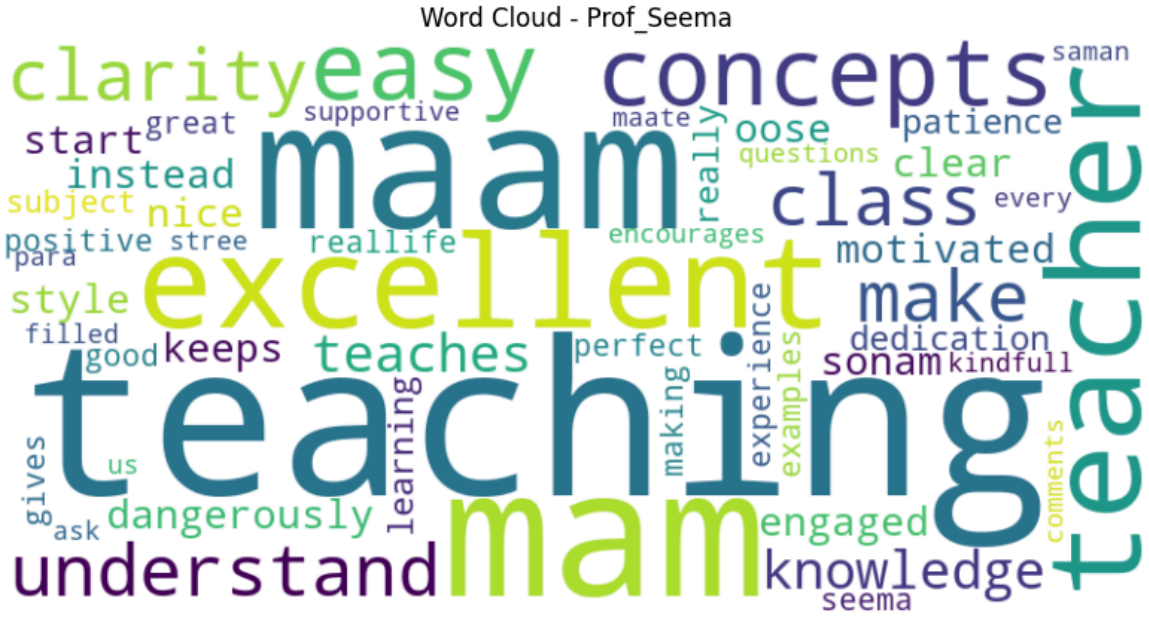
```
    text = re.sub(r'[^a-z\s]', '', text)  # remove punctuation/numbers
    text = re.sub(r'\s+', ' ', text)  # remove extra spaces
    return text
```

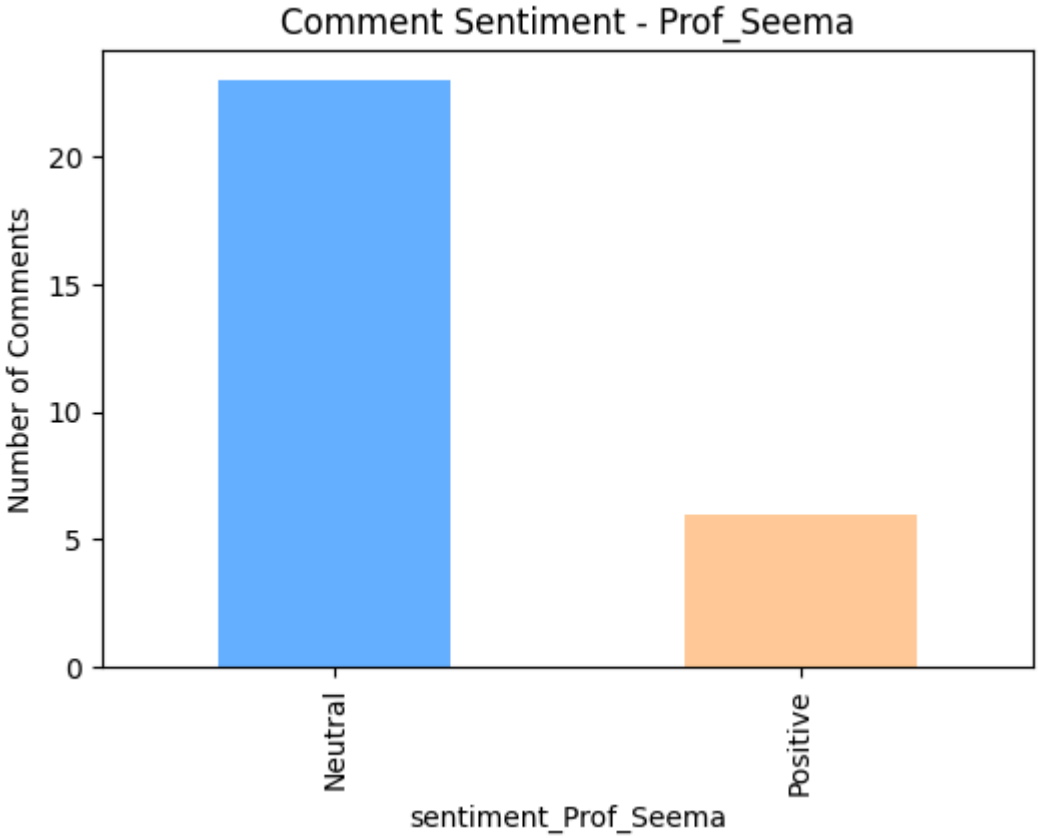
In [ ]:
```
#Define Sentiment Function
```

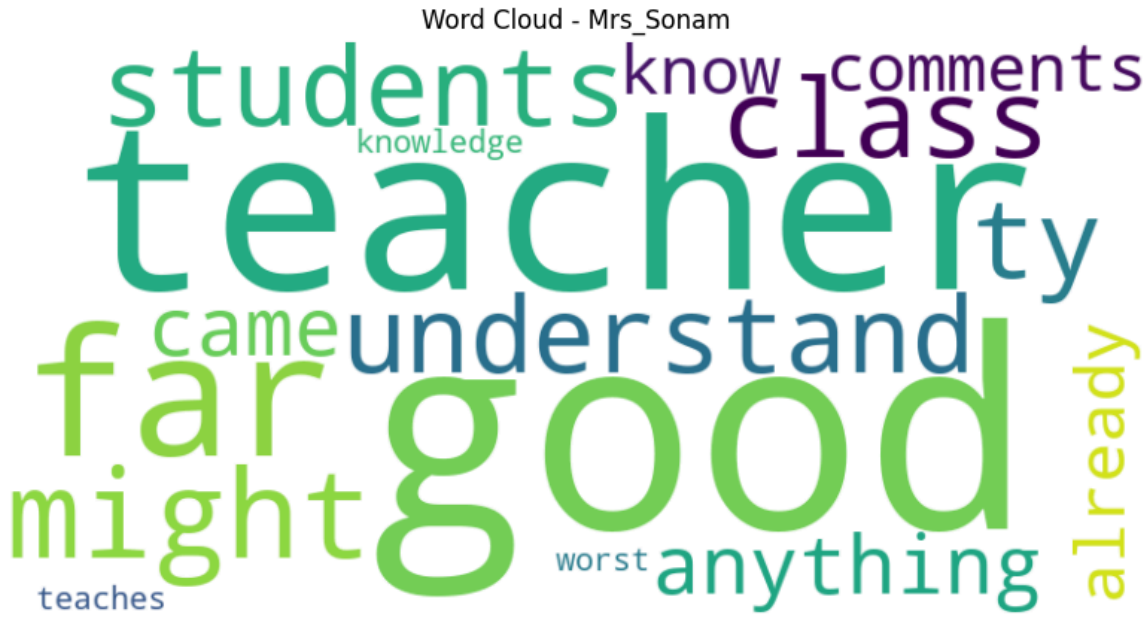
In [67]:
```
def get_sentiment(text):
    if text.strip() == "":
        return "Neutral"
    polarity = TextBlob(text).sentiment.polarity
    if polarity > 0.1:
        return "Positive"
    elif polarity < -0.1:
        return "Negative"
    else:
        return "Neutral"
```

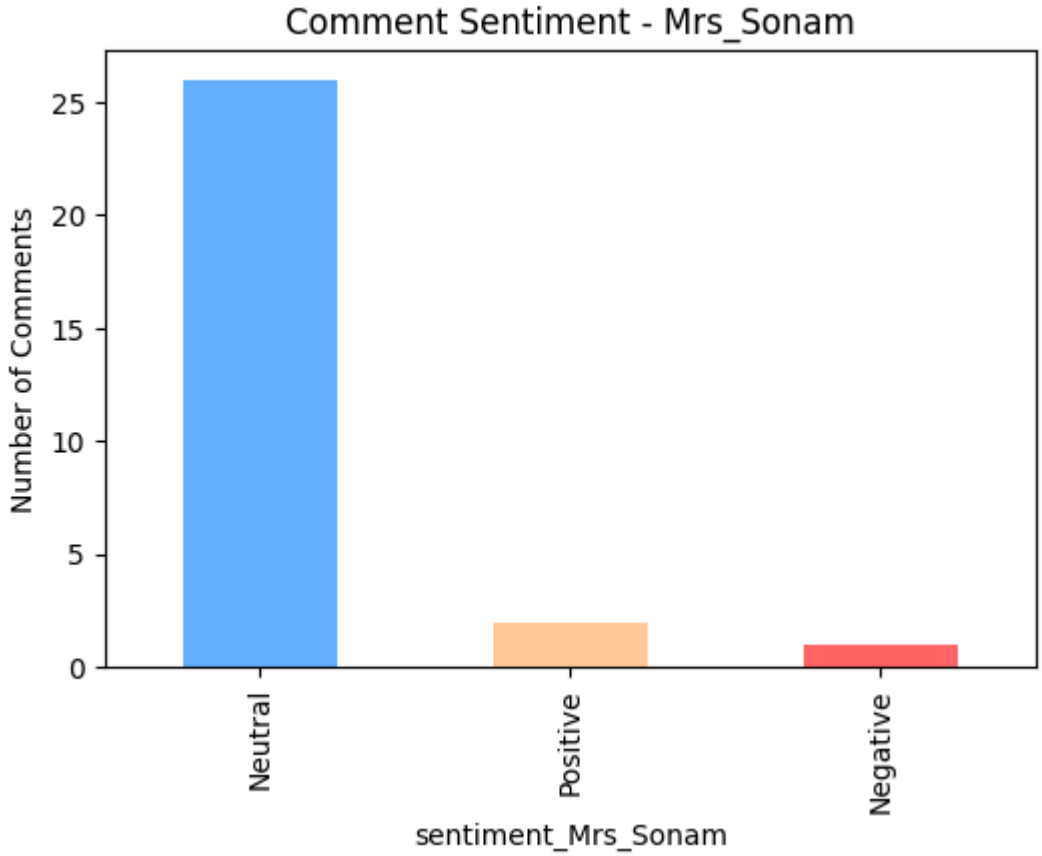
In [ ]:
```
#Loop Through All Teachers and Analyze
```

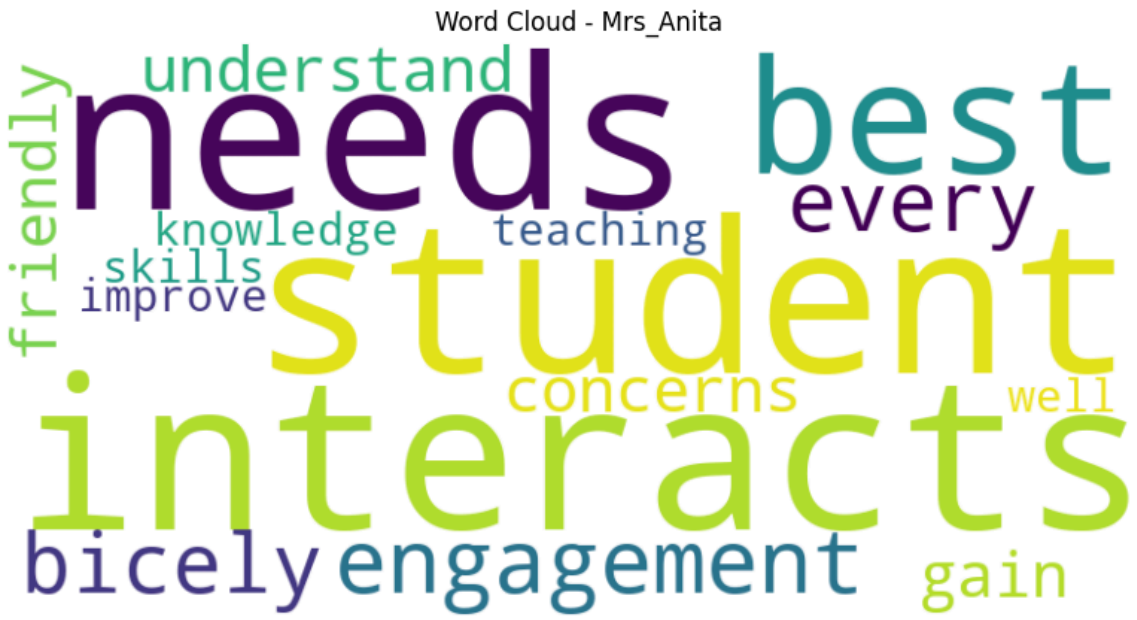
In [68]:
```
for col in All_Comments:
    teacher_name = col.replace("Comments_", "")

    # Clean comments
    df[f'clean_{teacher_name}'] = df[col].apply(clean_text)

    # Sentiment analysis
    df[f'sentiment_{teacher_name}'] = df[col].apply(get_sentiment)

    # Word Cloud
    all_comments = " ".join(df[f'clean_{teacher_name}'])
    wordcloud = WordCloud(width=800, height=400, background_color='white').gener

    plt.figure(figsize=(10,5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f"Word Cloud - {teacher_name}")
    plt.show()

    # Sentiment counts bar chart
    sentiment_counts = df[f'sentiment_{teacher_name}'].value_counts()
    plt.figure(figsize=(6,4))
    sentiment_counts.plot(kind='bar', color=['#66b3ff','#ffcc99','#ff6666'])
    plt.title(f"Comment Sentiment - {teacher_name}")
    plt.ylabel("Number of Comments")
    plt.show()
```

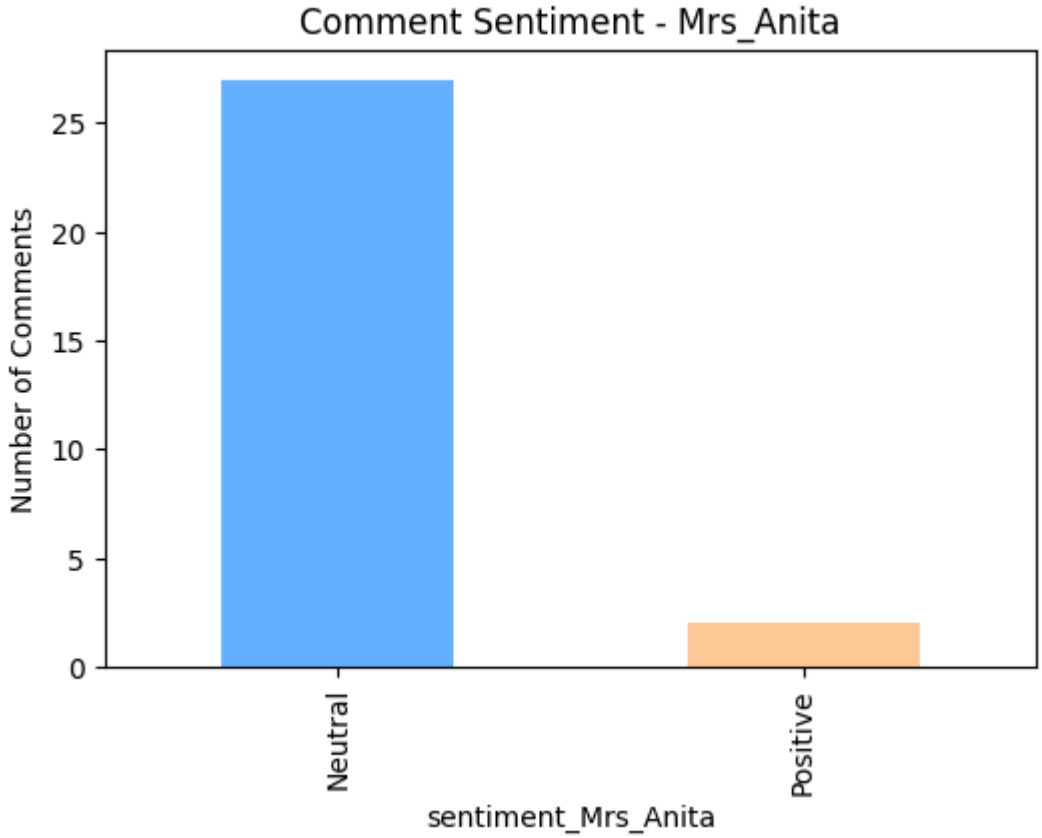

Word Cloud - Prof_Seema
clarity
easy
concepts
saman
start
great
supportive
maate
really
oose
patience
maam
instead
questions
clear
subject
nice
class
every
positive
stree
reallife
encourages
motivated
para
excellent
style
make
filled
keeps
teaches
perfect
dedication
good
sonam
kindfull
teacher
teaching
gives
us
learning
making
experience
examples
comments
ask
dangerously
engaged
mam
understand
knowledge
seema


Comment Sentiment - Prof_Seema
Number of Comments
0
5
10
15
20
Neutral
Positive
sentiment_Prof_Seema

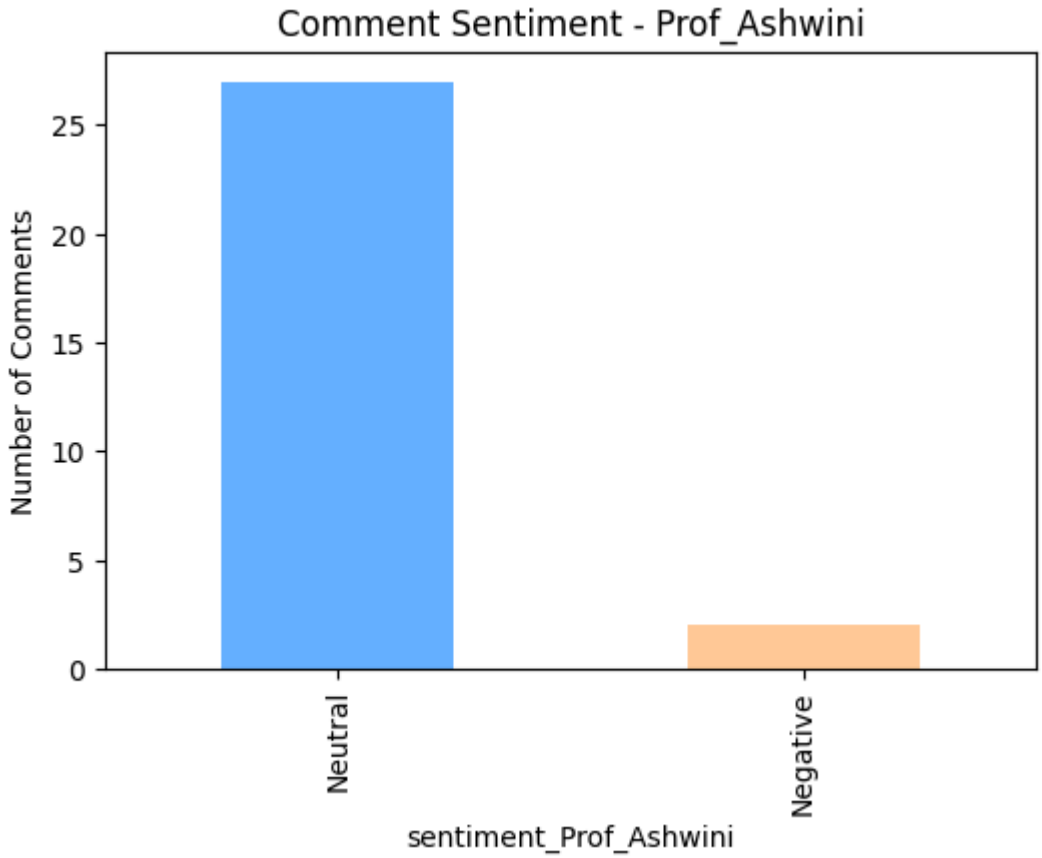

Word Cloud - Mrs_Sonam
students
know
comments
knowledge
class
teacher
ty
came
understand
far
good
might
already
worst
anything
teaches

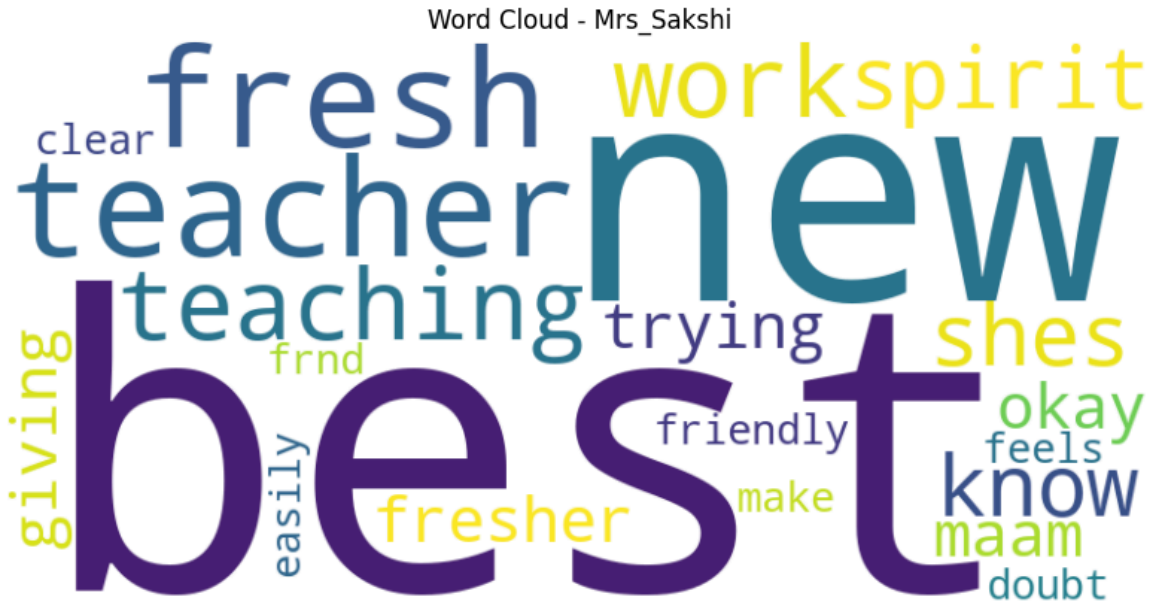

Comment Sentiment - Mrs_Sonam
Number of Comments
25
20
15
10
5
0
Neutral
Positive
Negative
sentiment_Mrs_Sonam

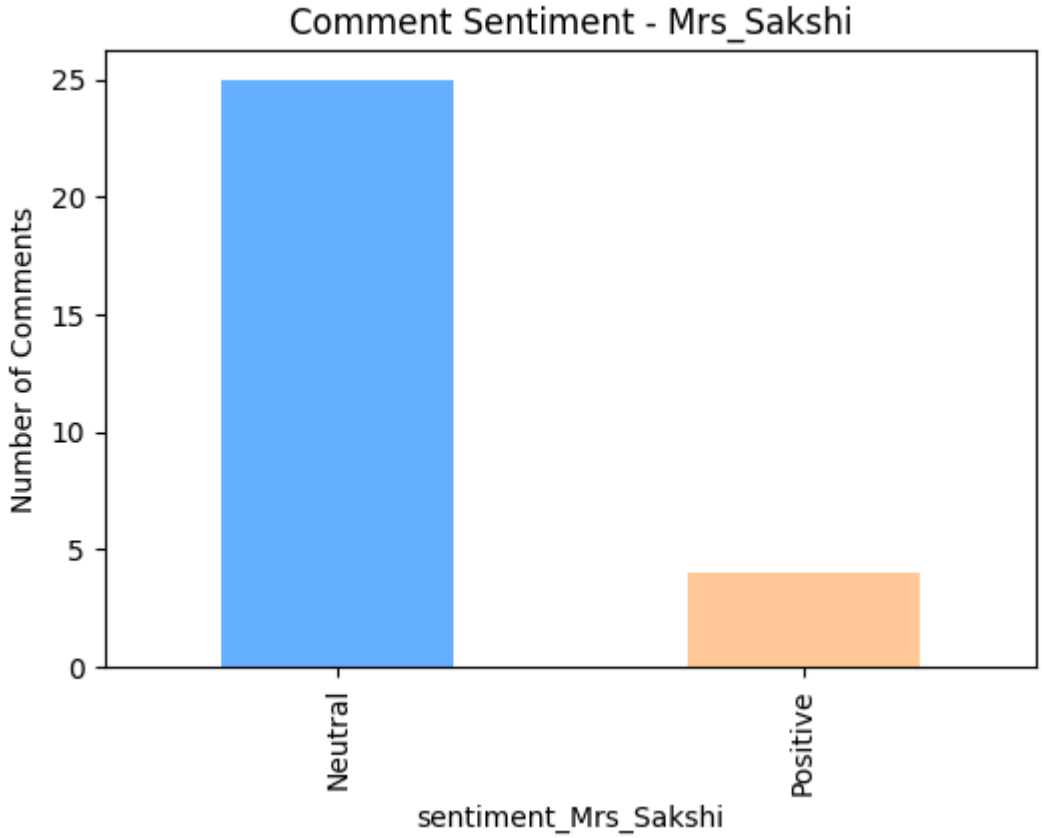

Word Cloud - Mrs_Anita
understand
needs
best
friendly
every
knowledge
teaching
skills
improve
student
concerns
well
interacts
bicely
engagement
gain


Comment Sentiment - Mrs_Anita
Number of Comments
25
20
15
10
5
0
Neutral
Positive
sentiment_Mrs_Anita

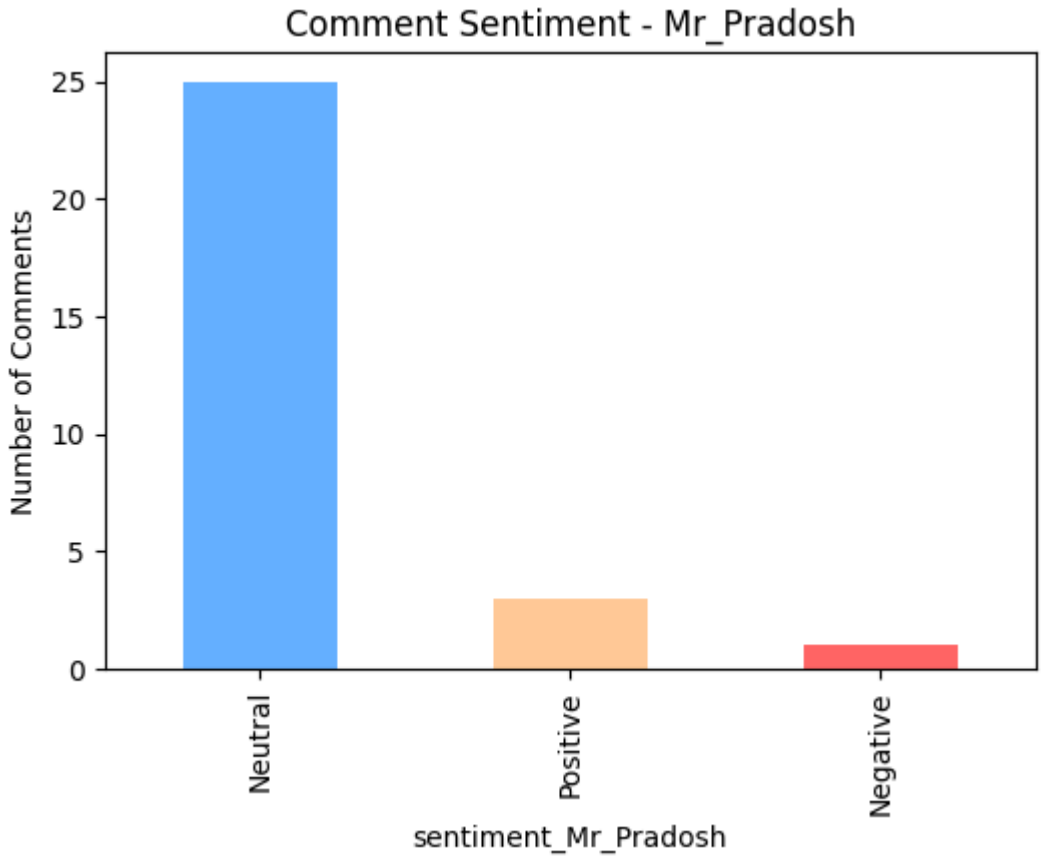

Word Cloud - Prof_Ashwini
feel
maam
doesnt
approach
helpful
subject
ashwini
scary
improve
average
first
way
strict
year
teaching
teach
didnt


Comment Sentiment - Prof_Ashwini
Number of Comments
25
20
15
10
5
0
Neutral
Negative
sentiment_Prof_Ashwini

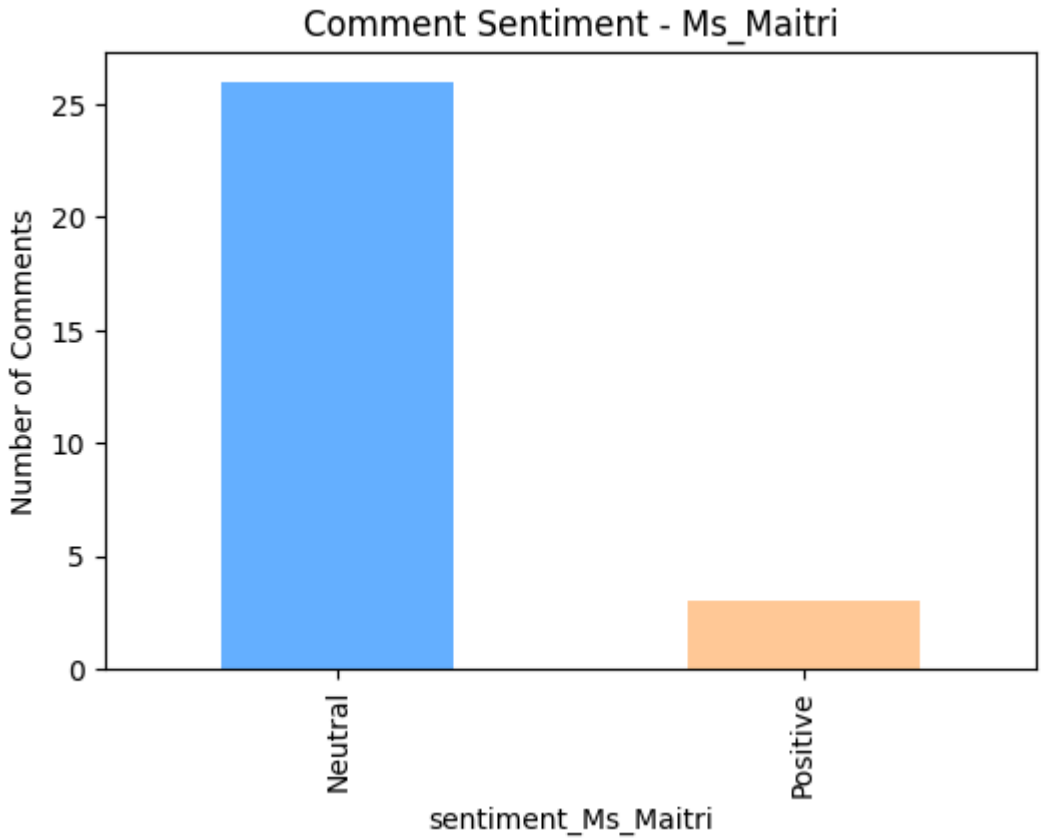

Word Cloud - Mrs_Sakshi
fresh
workspirit
clear
new
teacher
teaching
trying
shes
giving
frnd
best
okay
friendly
feels
know
easily
fresher
make
maam
doubt

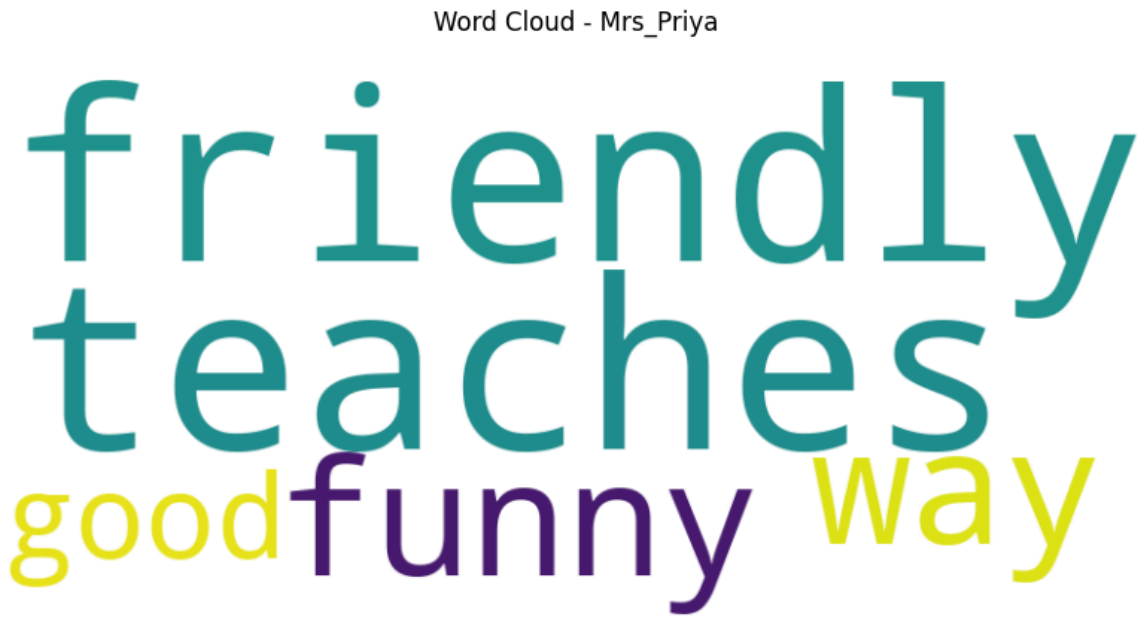

Comment Sentiment - Mrs_Sakshi
Number of Comments
25
20
15
10
5
0
Neutral
Positive
sentiment_Mrs_Sakshi

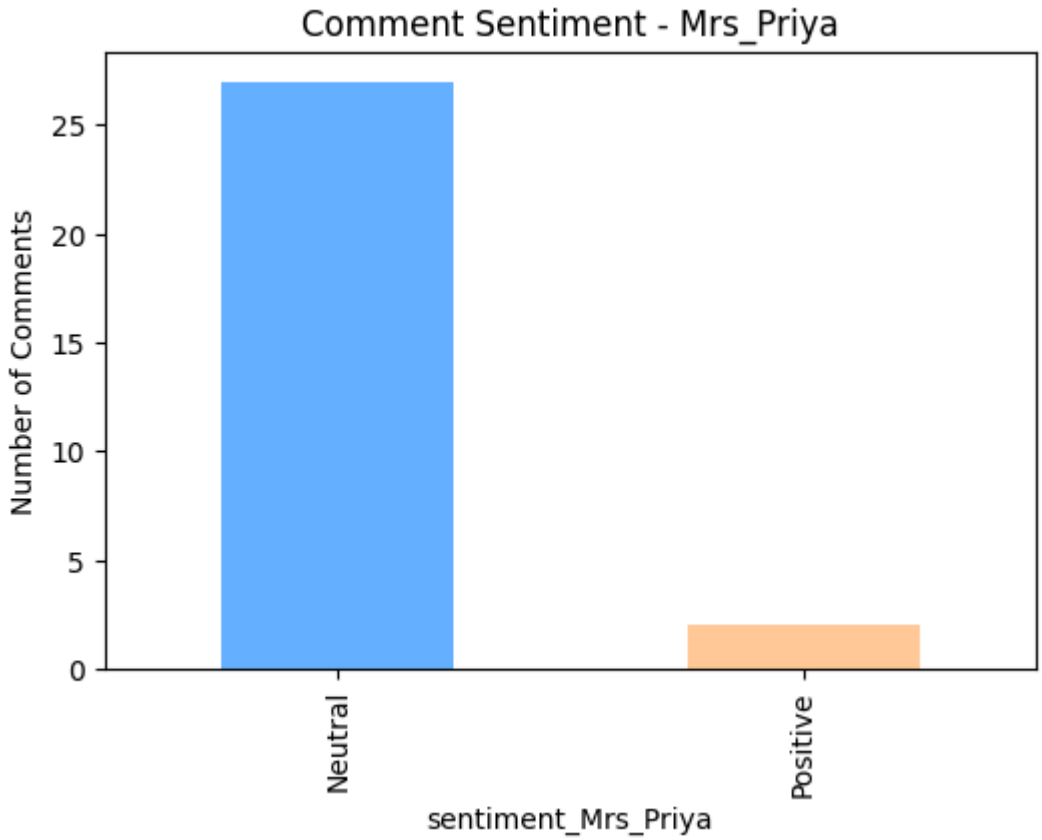

Word Cloud - Mr_Pradosh
average
engages
yet
hes
class
good
teacher
fun
learning
interacted
communication
dont
skill
friendly
much
environment
teaching


Comment Sentiment - Mr_Pradosh
Number of Comments
25
20
15
10
5
0
Neutral
Positive
Negative
sentiment_Mr_Pradosh

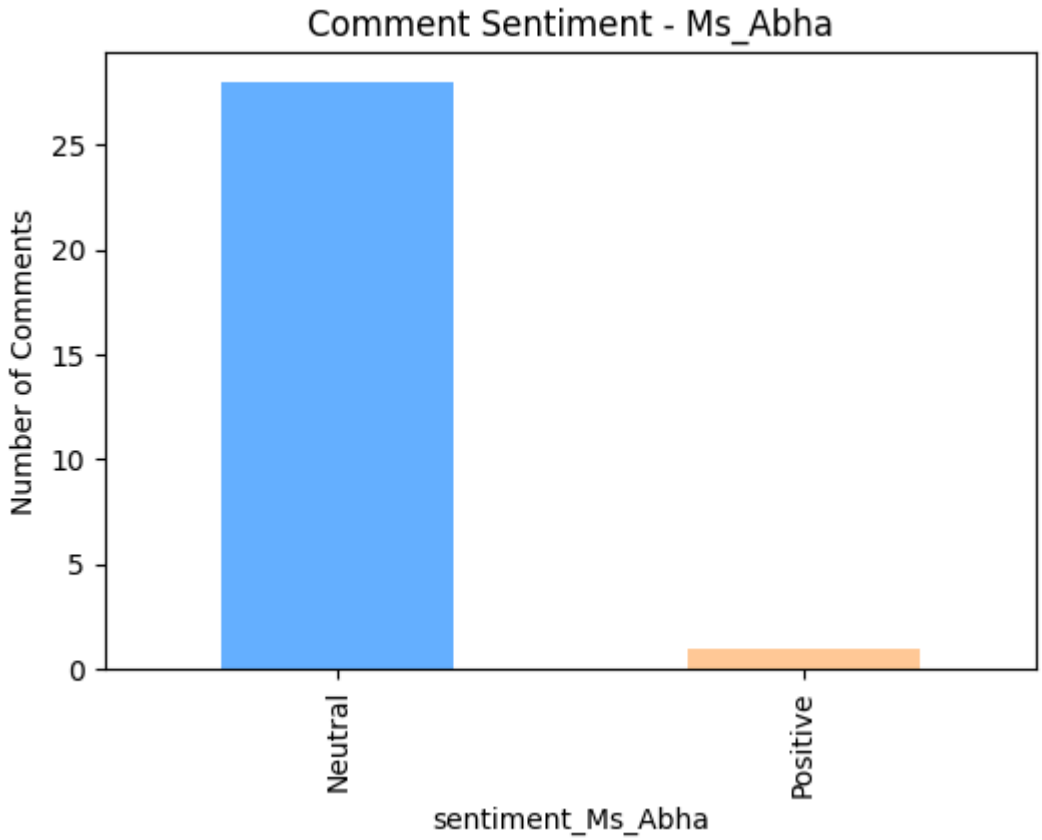
Word Cloud - Ms_Maitri

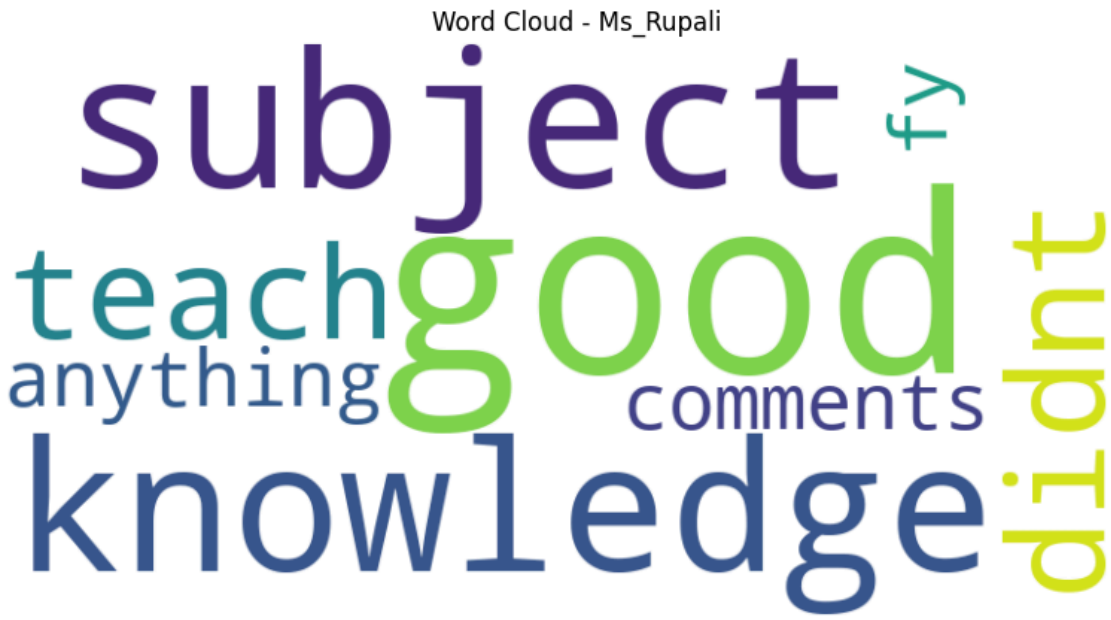


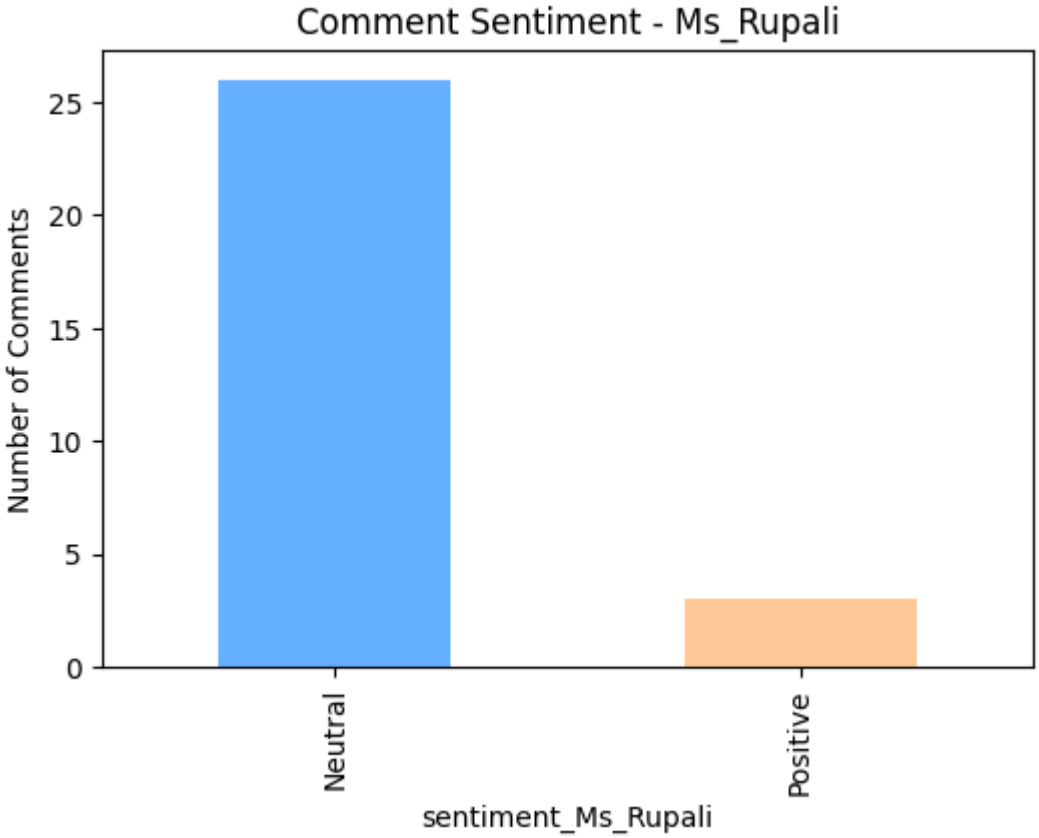

Word Cloud - Mrs_Priya
friendly
teaches
good
funny
way

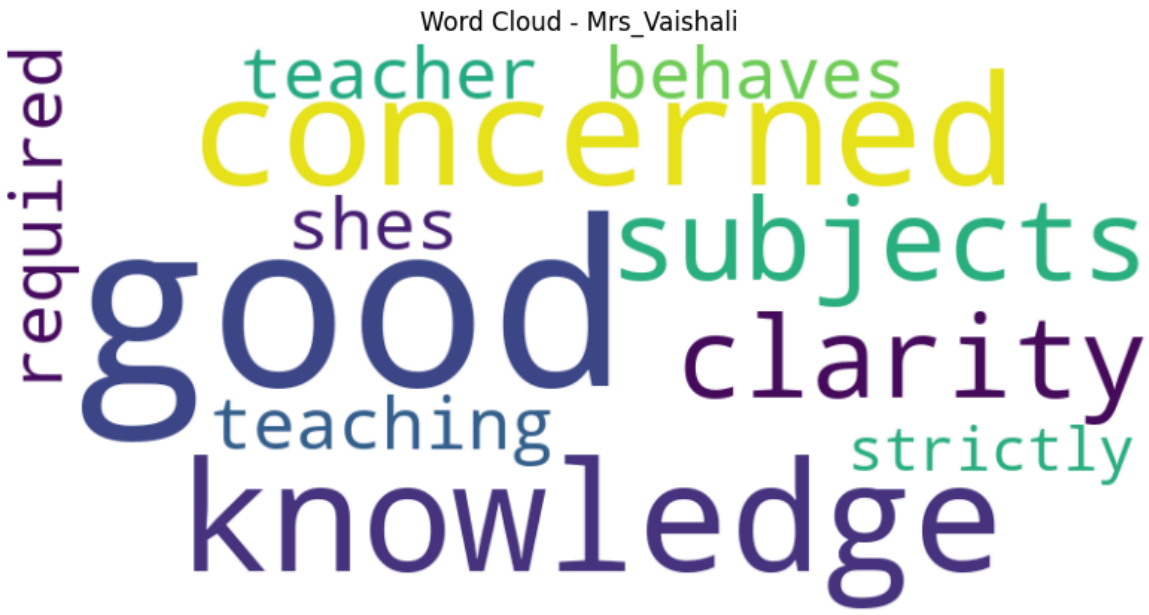

Comment Sentiment - Mrs_Priya
Number of Comments
0
5
10
15
20
25
Neutral
Positive
sentiment_Mrs_Priya

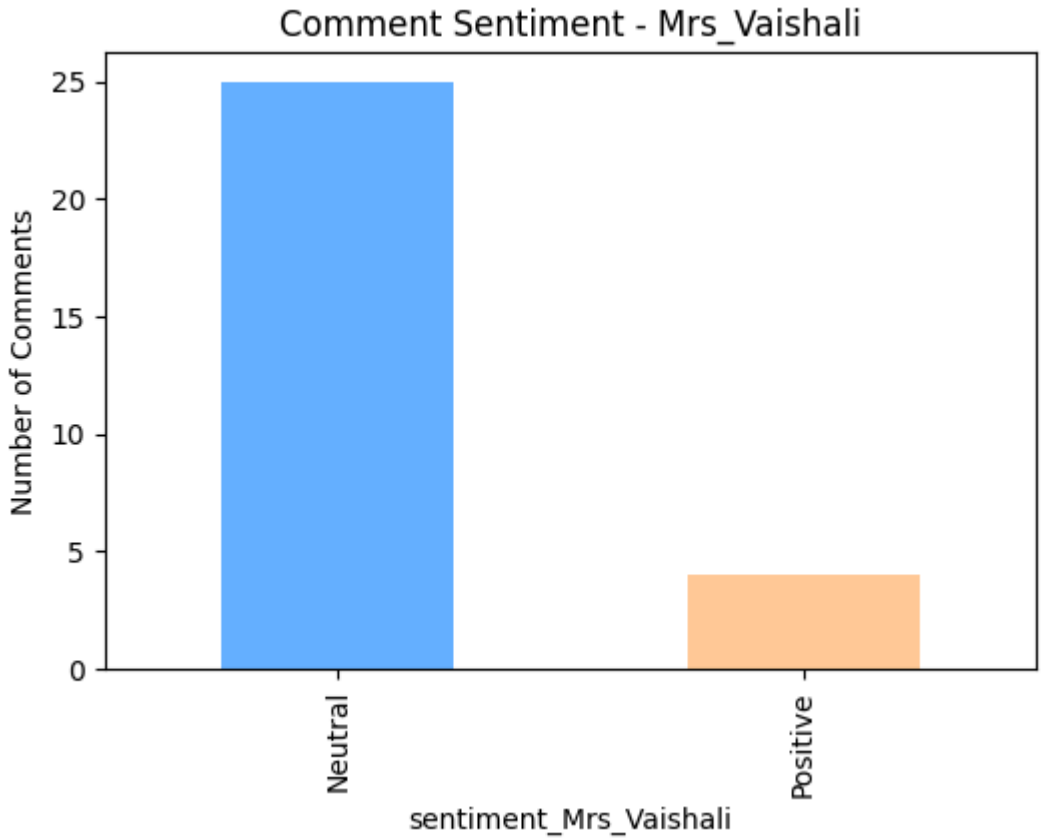

Word Cloud - Ms_Abha
contact
good
much


Comment Sentiment - Ms_Abha
Number of Comments
0
5
10
15
20
25
Neutral
Positive
sentiment_Ms_Abha


Word Cloud - Ms_Rupali
subject
fy
teach
good
anything
comments
didnt
knowledge


Comment Sentiment - Ms_Rupali
Number of Comments
25
20
15
10
5
0
Neutral
Positive
sentiment_Ms_Rupali





- Most teachers receive neutral or positive comments, while Ms. Sonam and Mr. Pradosh have a few negative comments, indicating generally favorable feedback with minor areas for improvement.

Student Feedback Analysis – Key Insights

---

1. Teacher Performance

- Prof. Seema is the top-performing teacher, excelling in most categories such as subject knowledge, doubt solving, and punctuality.
- Ms. Sakshi ranks second, performing particularly well in interaction and student engagement.
- Mr. Pradosh ranks third, showing strengths in subject knowledge but lower overall ratings compared to the top two.
- Ms. Sonam receives mostly lower ratings, highlighting the need for targeted improvement.

2. Category-wise Teacher Ratings

- Prof. Seema and Ms. Sakshi consistently score above 4 in most categories, reflecting balanced teaching quality.
- Ms. Sonam has ratings mostly below 3, suggesting widespread dissatisfaction in multiple teaching aspects.

3. Facilities Ratings

- Library, cleanliness, and computer lab are highly rated, indicating strong academic infrastructure.
- Sports and extracurricular activities receive lower ratings and show higher variability, suggesting mixed opinions and room for improvement.
- Classroom environment has the highest average rating and consistent feedback, reflecting general satisfaction in learning spaces.
- TY students rate library resources slightly higher than SY students, showing senior students find the facilities more adequate.

4. College Experience & Recommendation

- Most students have a neutral perception of their college experience, with a notable portion reporting positive experiences, and very few negative.
- Recommendations are mixed, with most students uncertain ("maybe") about endorsing the college, a smaller portion saying "yes," and some saying "no."

5. Sentiment Analysis

- Overall, teacher feedback is mostly neutral or positive.

- Ms. Sonam and Mr. Pradosh have a few negative comments, indicating minor areas for improvement.

Overall Insight:

Students are generally satisfied with teaching and academic facilities, with Prof. Seema and Ms. Sakshi standing out as top performers. Areas needing attention include sports, extracurricular activities, and improving the performance of lower-rated teachers. College experience and recommendations show moderate satisfaction, highlighting opportunities to enhance overall student engagement and perception.

Conclusion

The student feedback analysis reveals that overall teaching quality and academic facilities are well-received, with Prof. Seema and Ms. Sakshi emerging as top performers. Classroom environment, library, and computer lab are strong points, while sports and extracurricular activities need improvement. Most students report a neutral to positive college experience, and recommendations are mixed, indicating room to enhance engagement and satisfaction. Targeted support for lower-rated teachers and continued focus on student-centric facilities can further improve the overall college experience.
