In [1]:
```python
# eda_visuals.py

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

In [2]:
```python
# Load dataset
df = pd.read_csv('D:/AMITY/Semester_4/5. Major Project/mba-semester4-major-proje
```

In [3]:
```python
# Show basic info
print("\n--- Dataset Info ---")
print(df.info())
```

```
--- Dataset Info ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12492 entries, 0 to 12491
Data columns (total 21 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   age_group                    12492 non-null  object
 1   gender                       12492 non-null  object
 2   city                         12492 non-null  object
 3   toilet_cleanliness           12492 non-null  object
 4   toilet_safety                12492 non-null  object
 5   toilet_features              12492 non-null  object
 6   service_use                  12492 non-null  object
 7   service_use_freq             12492 non-null  object
 8   transport_satisfaction       12492 non-null  object
 9   transport_suggestions        12492 non-null  object
 10  park_visiting                12492 non-null  object
 11  park_visit_freq              12492 non-null  object
 12  park_amenities               12492 non-null  object
 13  park_issues                  12492 non-null  object
 14  transport_safety             12492 non-null  object
 15  park_suggestions             12492 non-null  object
 16  library_satisfaction         12492 non-null  object
 17  library_visit_freq           12492 non-null  object
 18  local_service_satisfaction   12492 non-null  object
 19  library_suggestions          12492 non-null  object
 20  local_service_suggestions    12492 non-null  object
dtypes: object(21)
memory usage: 2.0+ MB
None
```

In [4]:
```python
df.dtypes
```

Out[4]:
```
age_group                    object
gender                       object
city                         object
toilet_cleanliness           object
toilet_safety                object
toilet_features              object
service_use                  object
service_use_freq             object
transport_satisfaction       object
transport_suggestions        object
park_visiting                object
park_visit_freq              object
park_amenities               object
park_issues                  object
transport_safety             object
park_suggestions             object
library_satisfaction         object
library_visit_freq           object
local_service_satisfaction   object
library_suggestions          object
local_service_suggestions    object
dtype: object
```

In [5]:
```
# Check for missing values
print("\n--- Missing Values ---")
print(df.isnull().sum())
```

```
--- Missing Values ---
age_group                    0
gender                       0
city                         0
toilet_cleanliness           0
toilet_safety                0
toilet_features              0
service_use                  0
service_use_freq             0
transport_satisfaction       0
transport_suggestions        0
park_visiting                0
park_visit_freq              0
park_amenities               0
park_issues                  0
transport_safety             0
park_suggestions             0
library_satisfaction         0
library_visit_freq           0
local_service_satisfaction   0
library_suggestions          0
local_service_suggestions    0
dtype: int64
```

In [6]:
```
df.head(5)
```

Out[6]:

	age_group	gender	city	toilet_cleanliness	toilet_safety	toilet_features	servi
0	36-50	Male	Chennai	Poor	Somewhat unsafe	Comfortable seating areas	
1	71-95	Male	Kanakapura	Good	Somewhat safe	Air conditioning or heating	
2	36-50	Male	Mysuru	Good	Somewhat safe	Baby-changing facilities	
3	36-50	Male	Hyderabad	Fair	Neutral	Comfortable seating areas	
4	18-35	Male	Hyderabad	Fair	Neutral	Comfortable seating areas	

5 rows × 21 columns

In [7]:
```
# Basic descriptive stats
print("\n--- Descriptive Statistics ---")
df.describe(include='all')
```

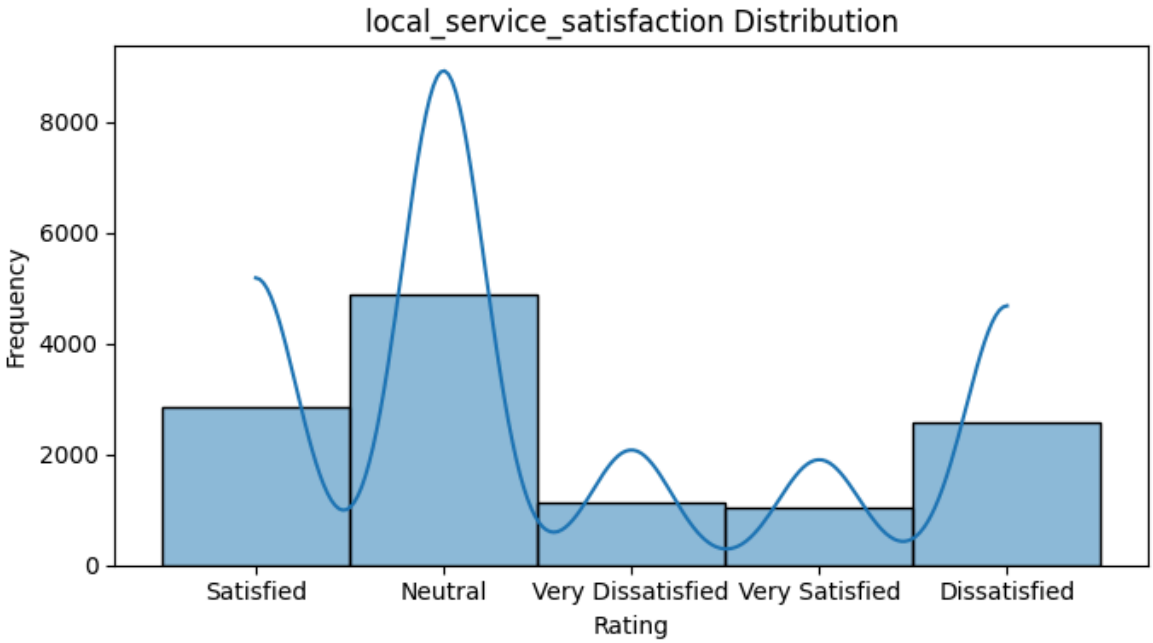
```
--- Descriptive Statistics ---
```

Out[7]:

	age_group	gender	city	toilet_cleanliness	toilet_safety	toilet_features
count	12492	12492	12492	12492	12492	12492
unique	4	2	8	4	5	6
top	18-35	Male	Kanakapura	Good	Somewhat safe	Comfortable seating areas
freq	5001	7224	3444	5256	4152	4032

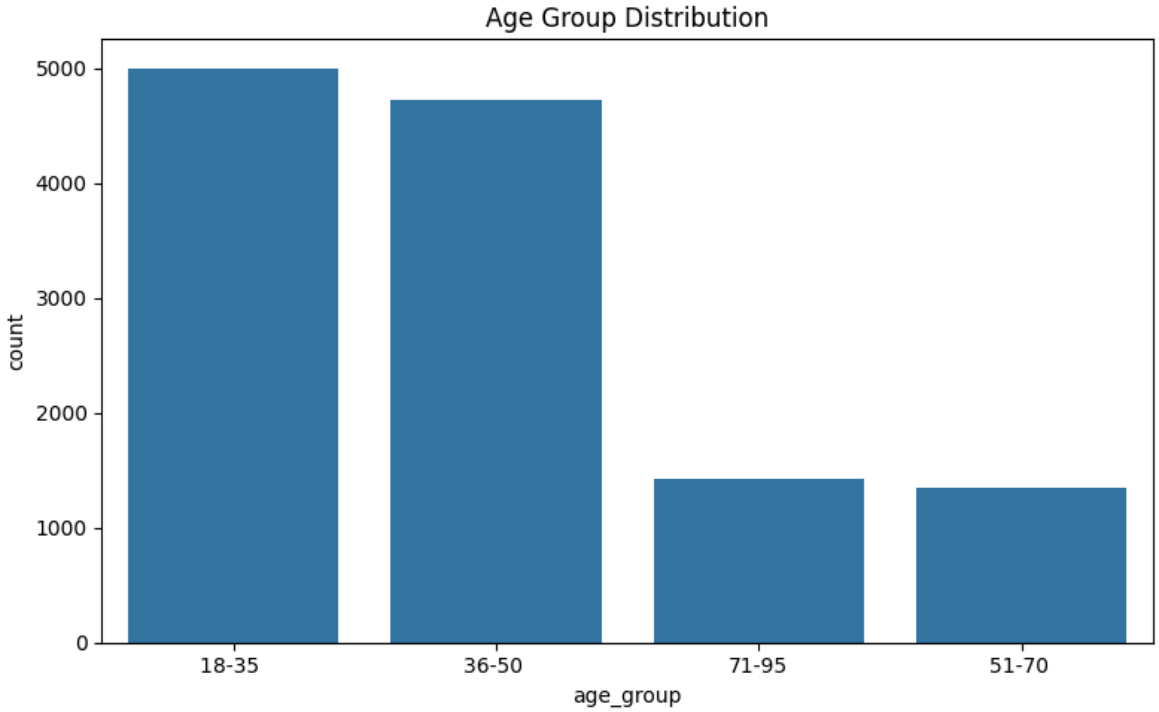
4 rows × 21 columns

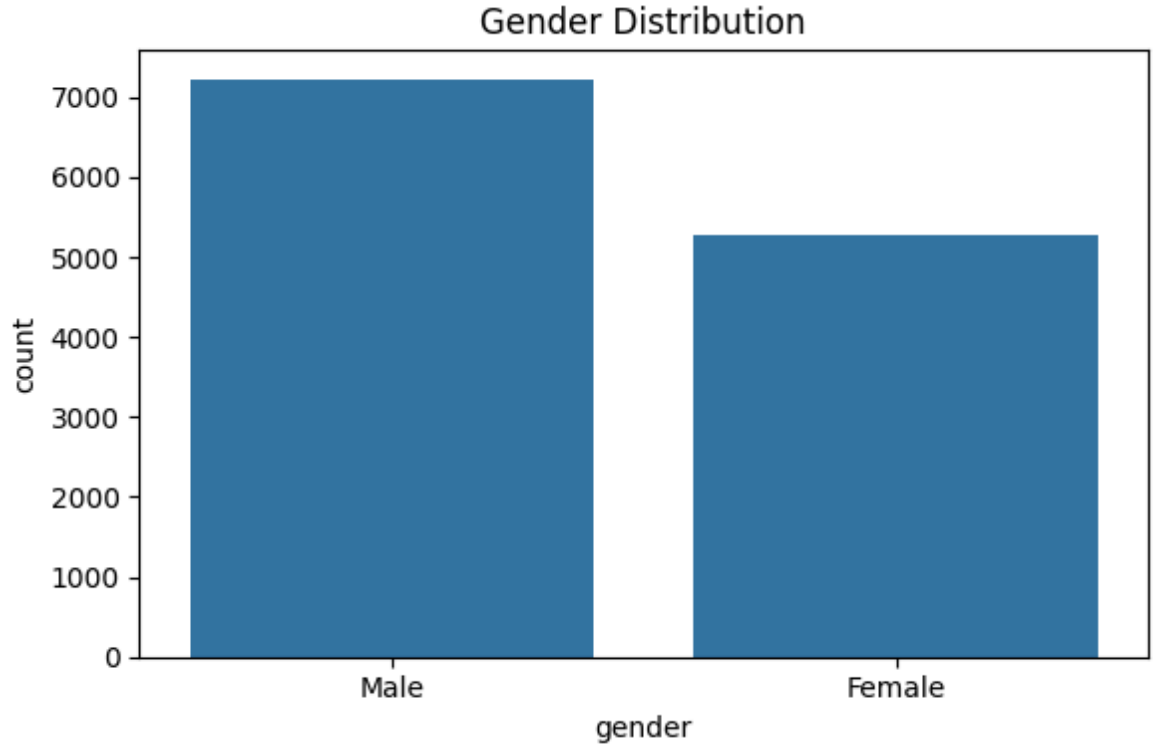
In [8]:
```
plt.figure(figsize=(7,4))
sns.histplot(df['local_service_satisfaction'], bins=8, kde=True)
plt.title('local_service_satisfaction Distribution')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

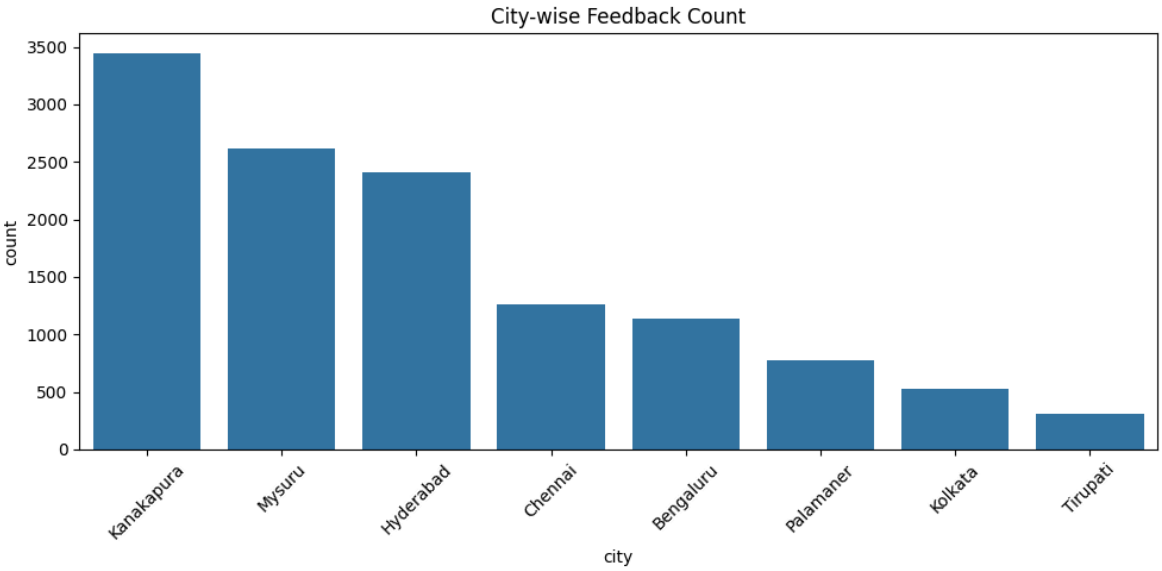


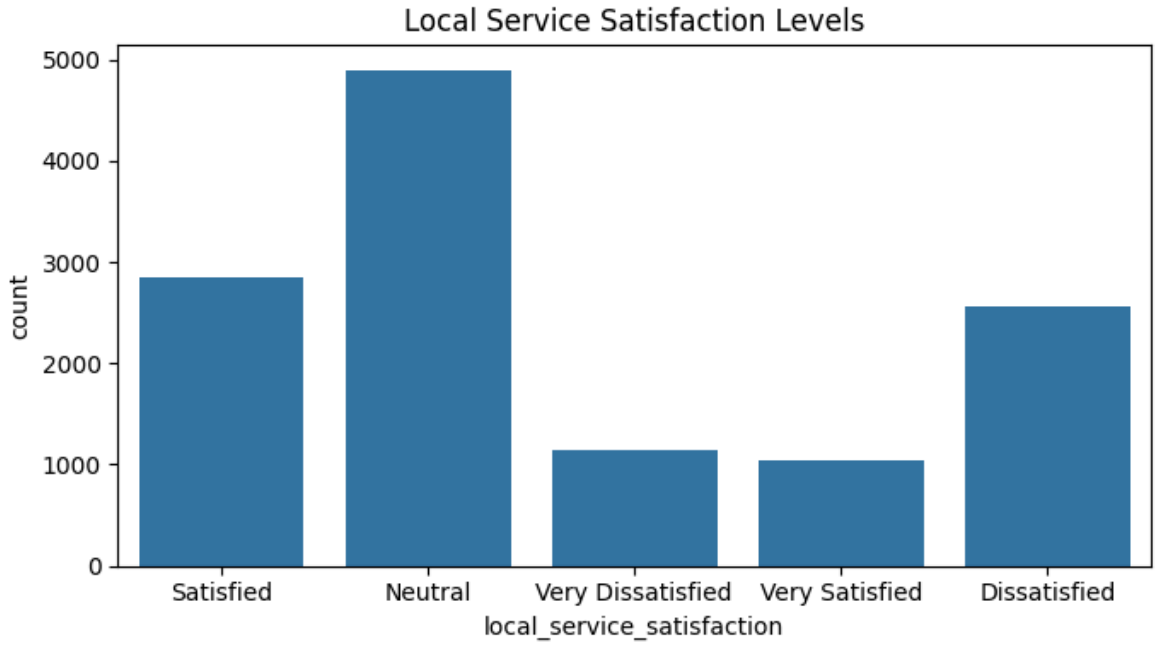
In [9]:

```
# 1. Age group distribution
plt.figure(figsize=(8,5))
sns.countplot(data=df, x='age_group', order=df['age_group'].value_counts().index
plt.title('Age Group Distribution')
plt.tight_layout()
plt.savefig("../data/exports/eda_plots/age_group_dist.png")
plt.show()

# 2. Gender distribution
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='gender')
plt.title('Gender Distribution')
plt.tight_layout()
plt.savefig("../data/exports/eda_plots/gender_dist.png")
plt.show()

# 3. City distribution
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='city', order=df['city'].value_counts().index)
plt.title('City-wise Feedback Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig("../data/exports/eda_plots/city_dist.png")
plt.show()

# 4. Service Satisfaction Distribution
plt.figure(figsize=(7,4))
sns.countplot(data=df, x='local_service_satisfaction')
plt.title('Local Service Satisfaction Levels')
plt.tight_layout()
plt.savefig("../data/exports/eda_plots/local_service_satisfaction.png")
plt.show()
```

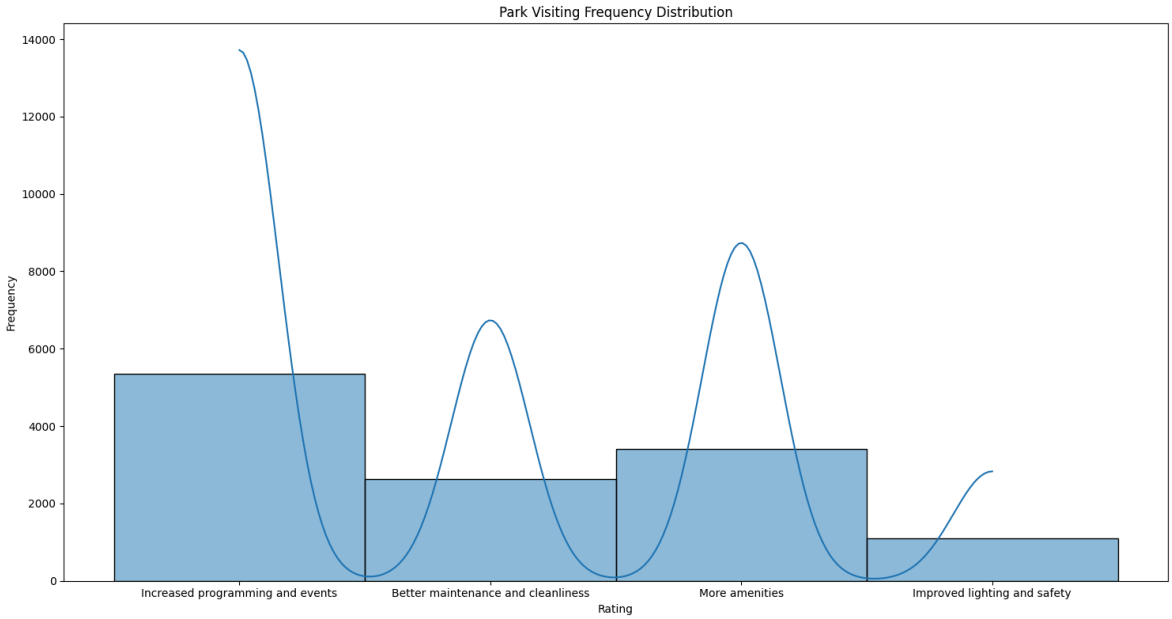

Age Group Distribution
5000
4000
3000
2000
1000
0
count
18-35
36-50
71-95
51-70
age_group


Gender Distribution
7000
6000
5000
4000
3000
2000
1000
0
count
Male
Female
gender

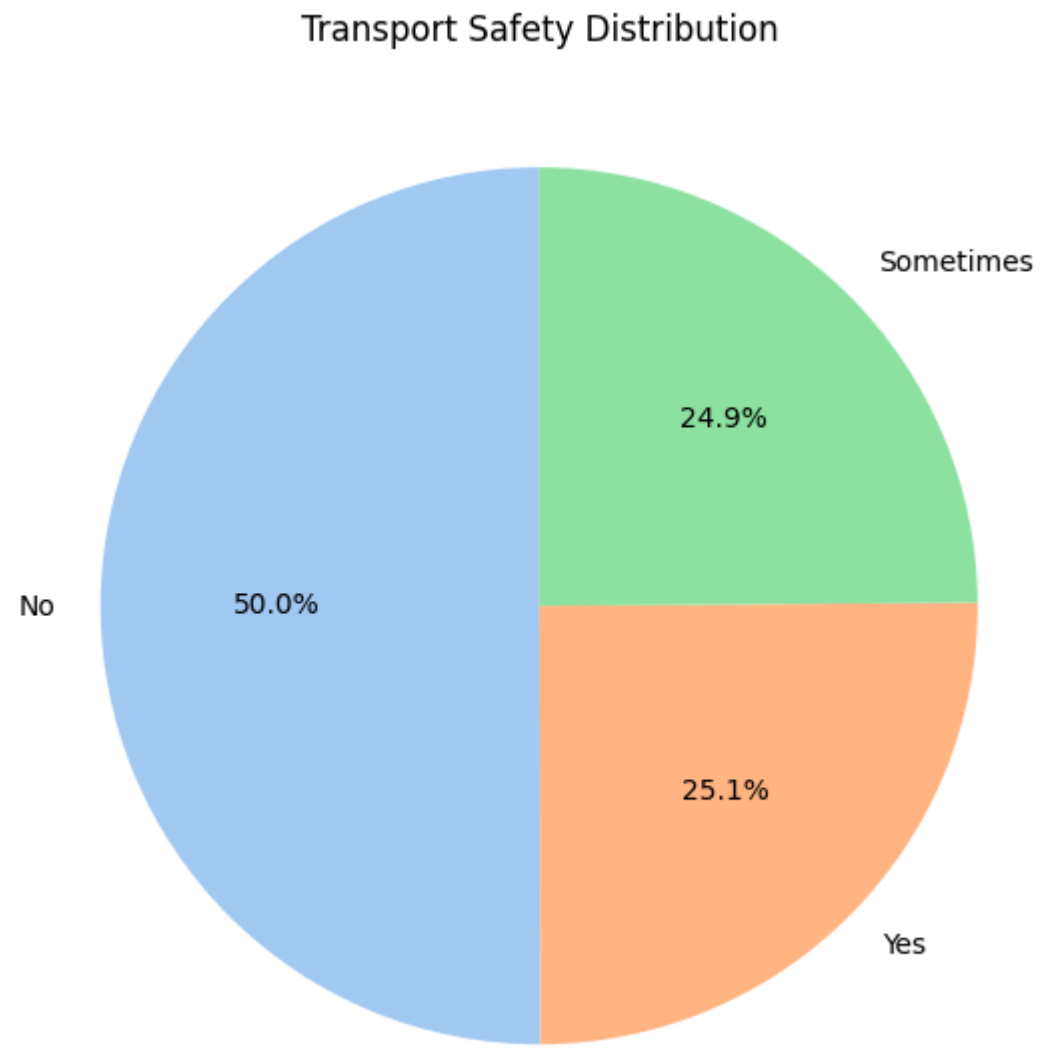




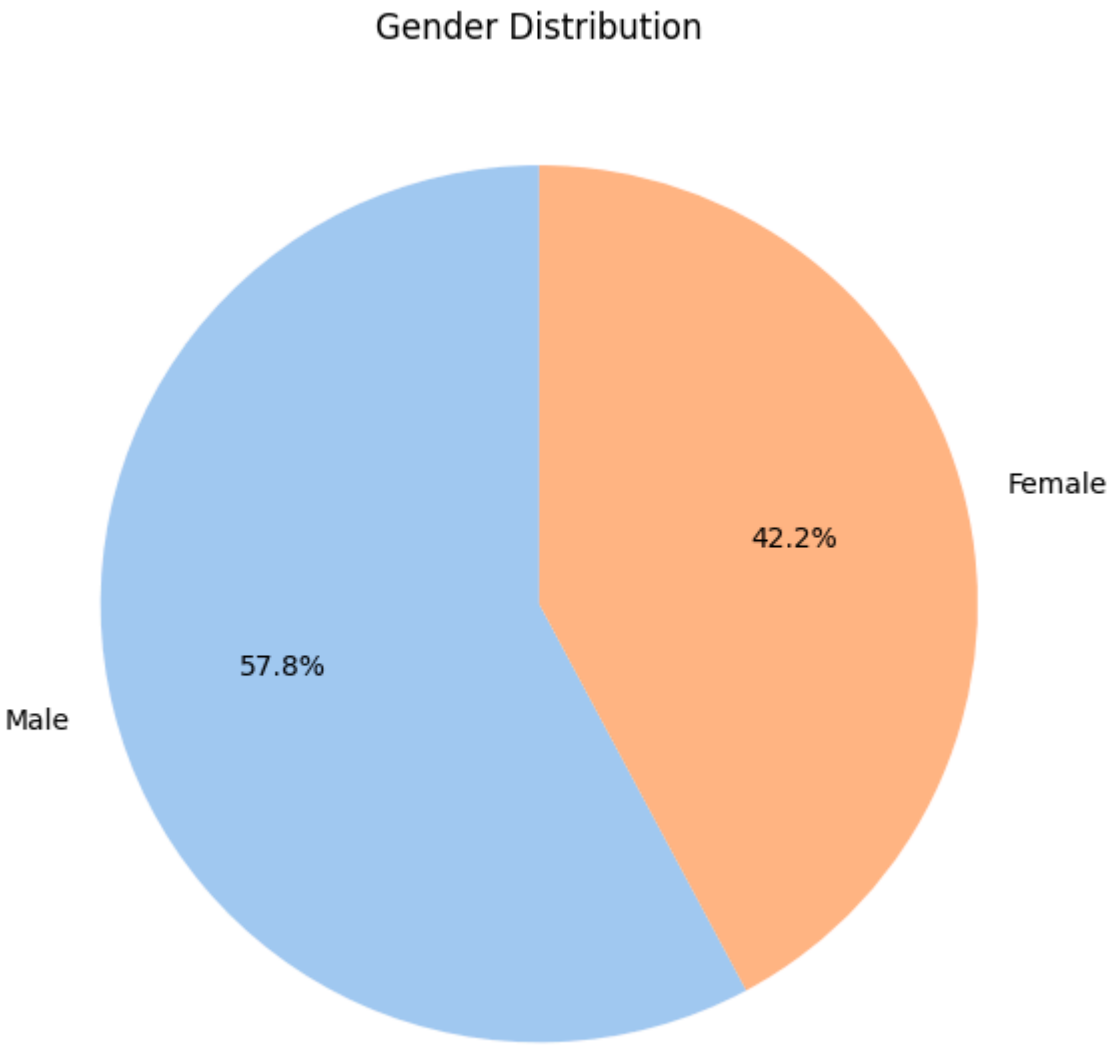
In [25]:
```
plt.figure(figsize=(15,8))
sns.histplot(df['park_visit_freq'], bins=8, kde=True)
plt.title('Park Visiting Frequency Distribution')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



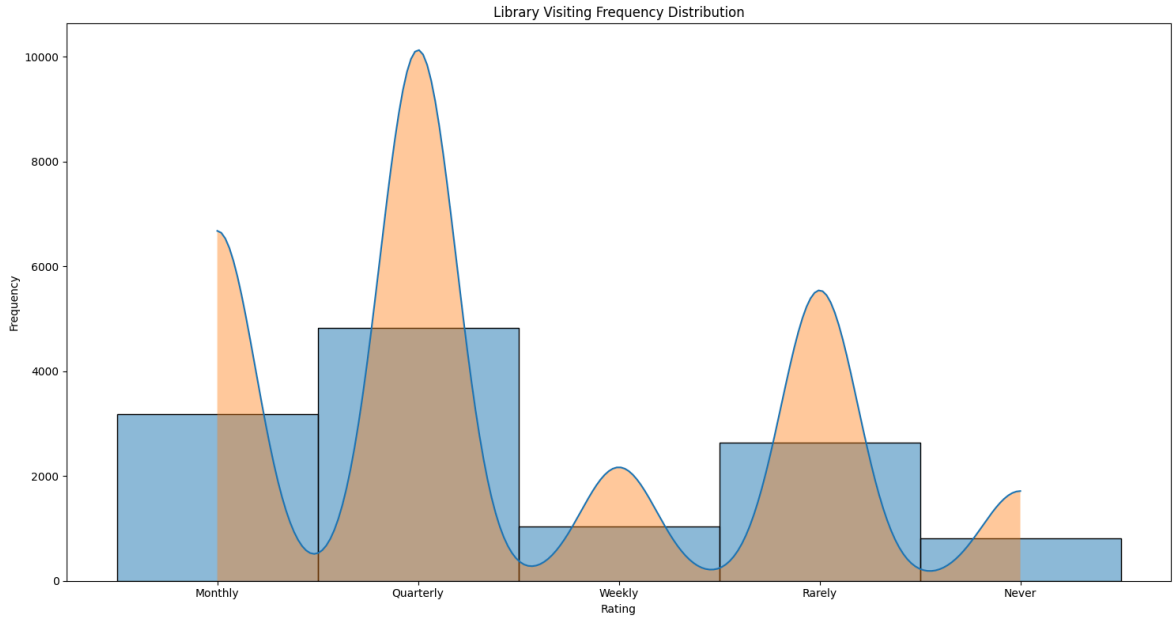
In [27]:
```
# Transport distribution as a pie chart
gender_counts = df['transport_safety'].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle
plt.title('Transport Safety Distribution')
plt.tight_layout()
plt.show()
```



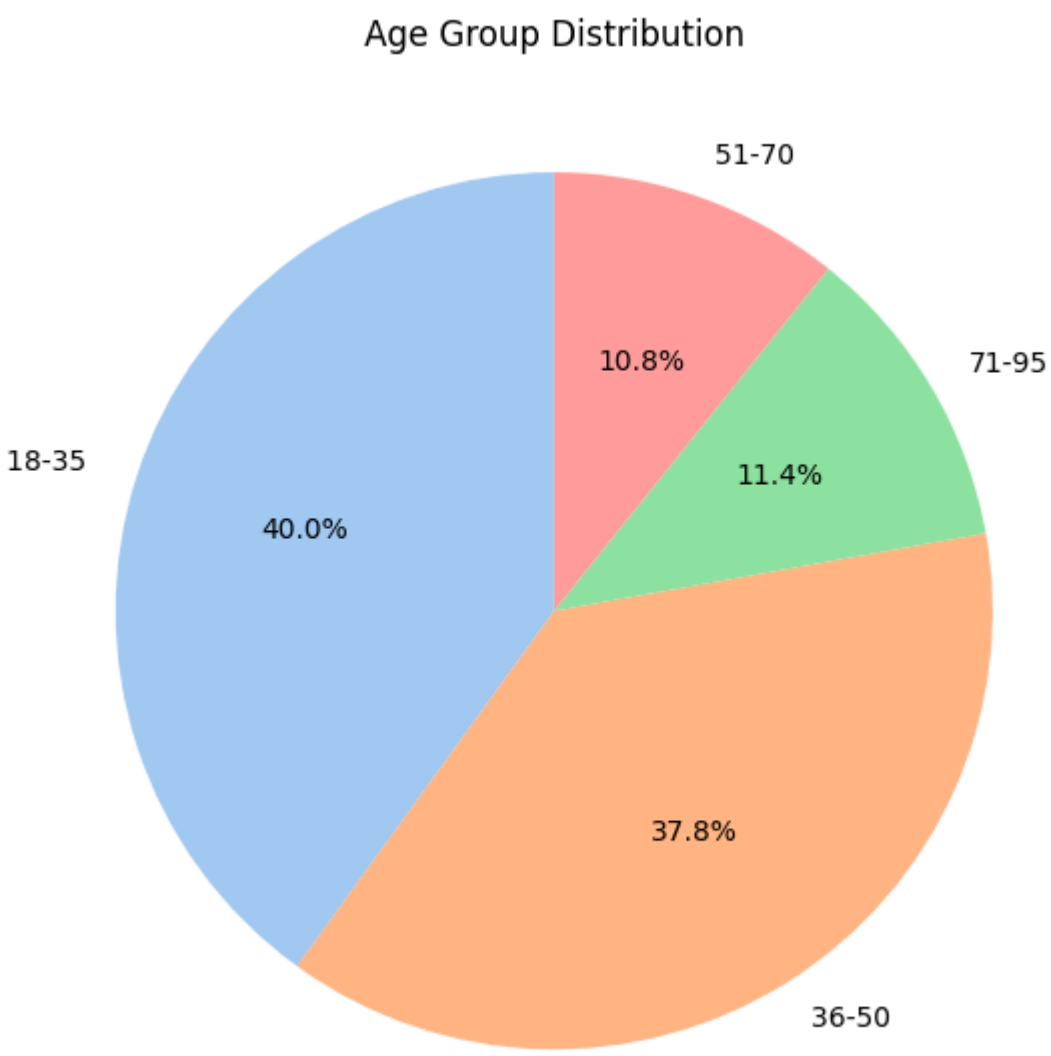
In [11]:
```
# Gender distribution as a pie chart
gender_counts = df['gender'].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle
plt.title('Gender Distribution')
plt.tight_layout()
plt.show()
```



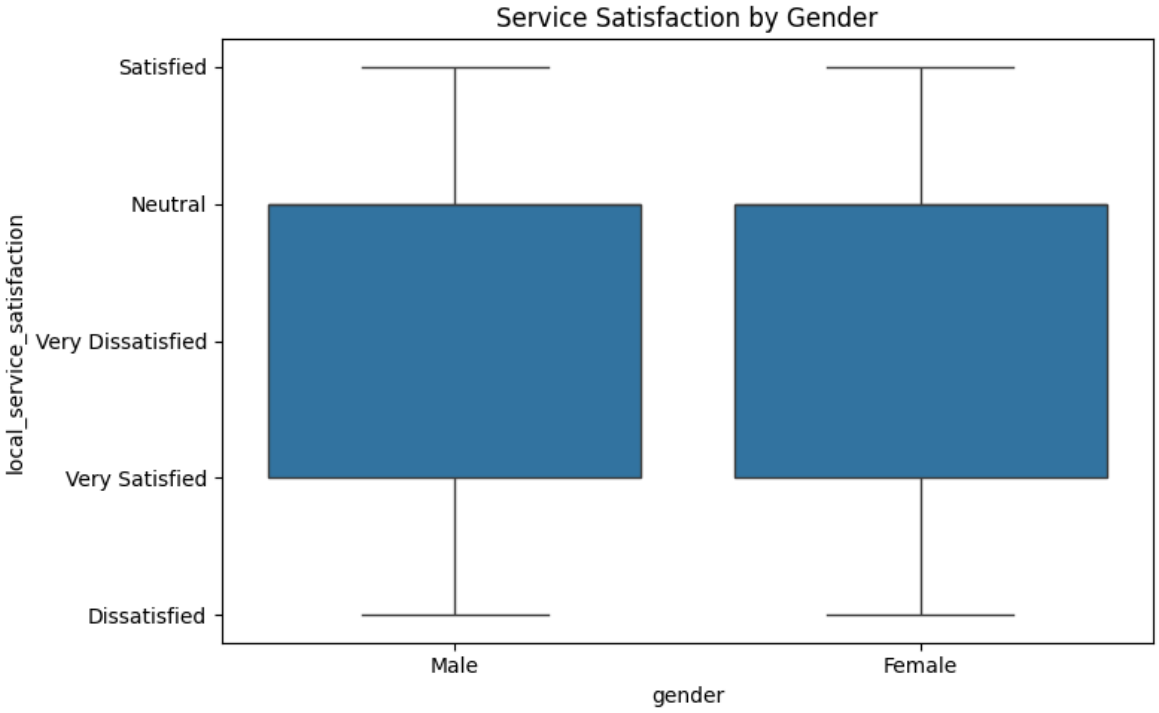
In [36]:
```
plt.figure(figsize=(15, 8))
sns.histplot(df['library_visit_freq'], bins=5, kde=True, fill=True, edgecolor='b

# Get the KDE line data
kde_line = plt.gca().lines[0]
kde_x, kde_y = kde_line.get_data()

# Fill the area under the KDE line
plt.fill_between(kde_x, kde_y, alpha=0.4)

plt.title('Library Visiting Frequency Distribution')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



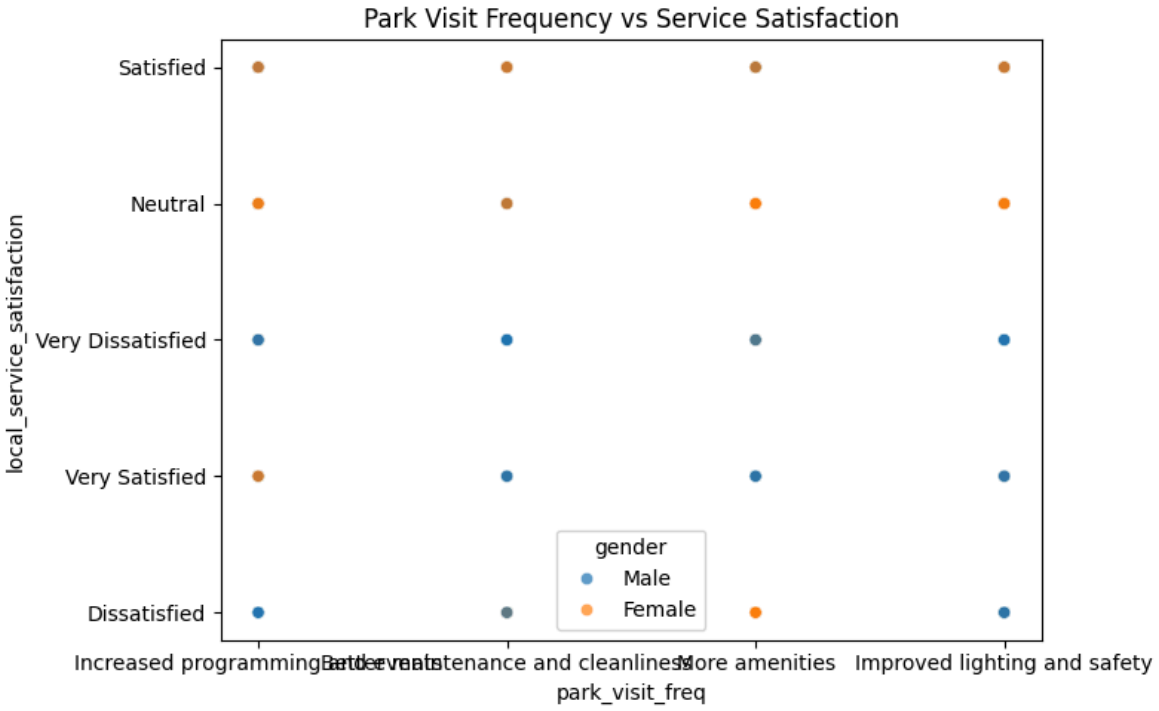
In [12]:
```
age_group_counts = df['age_group'].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(age_group_counts, labels=age_group_counts.index, autopct='%1.1f%%', star
plt.title('Age Group Distribution')
plt.tight_layout()
plt.show()
```



In [13]:
```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='gender', y='local_service_satisfaction')
plt.title('Service Satisfaction by Gender')
plt.tight_layout()
plt.show()
```

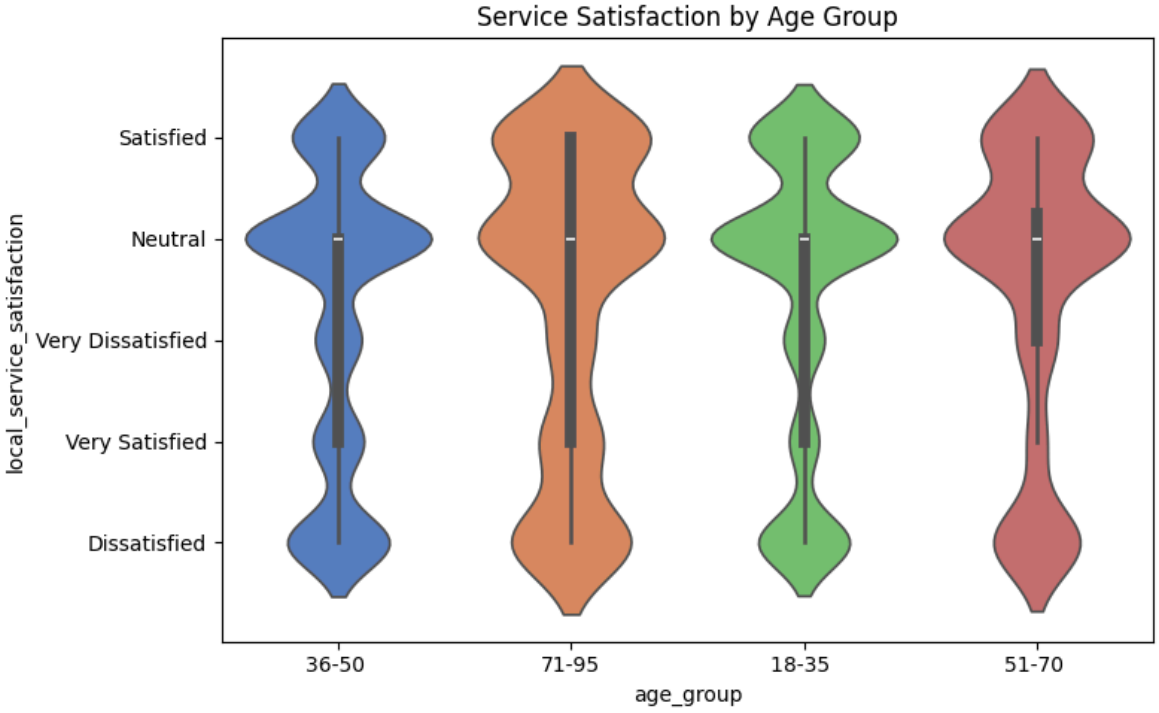


In [14]:
```
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='park_visit_freq', y='local_service_satisfaction', hu
plt.title('Park Visit Frequency vs Service Satisfaction')
plt.tight_layout()
plt.show()
```

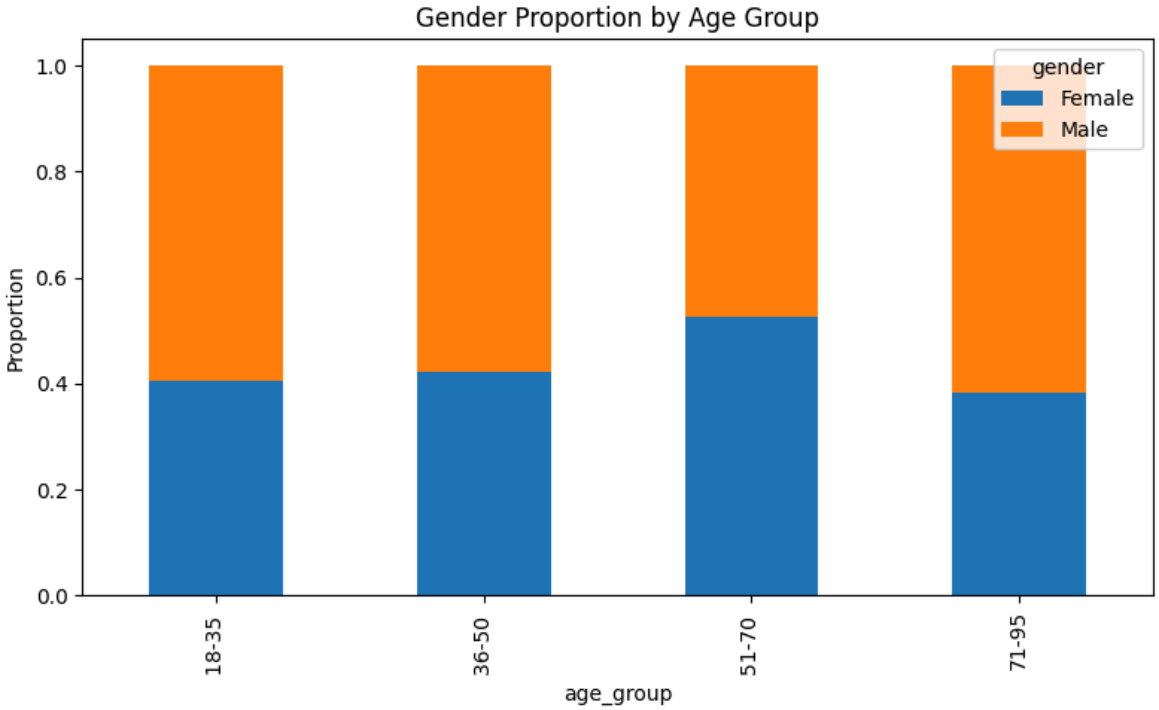


In [15]:
```
plt.figure(figsize=(8, 5))
sns.violinplot(data=df, x='age_group', y='local_service_satisfaction', palette='
plt.title('Service Satisfaction by Age Group')
plt.tight_layout()
plt.show()
```

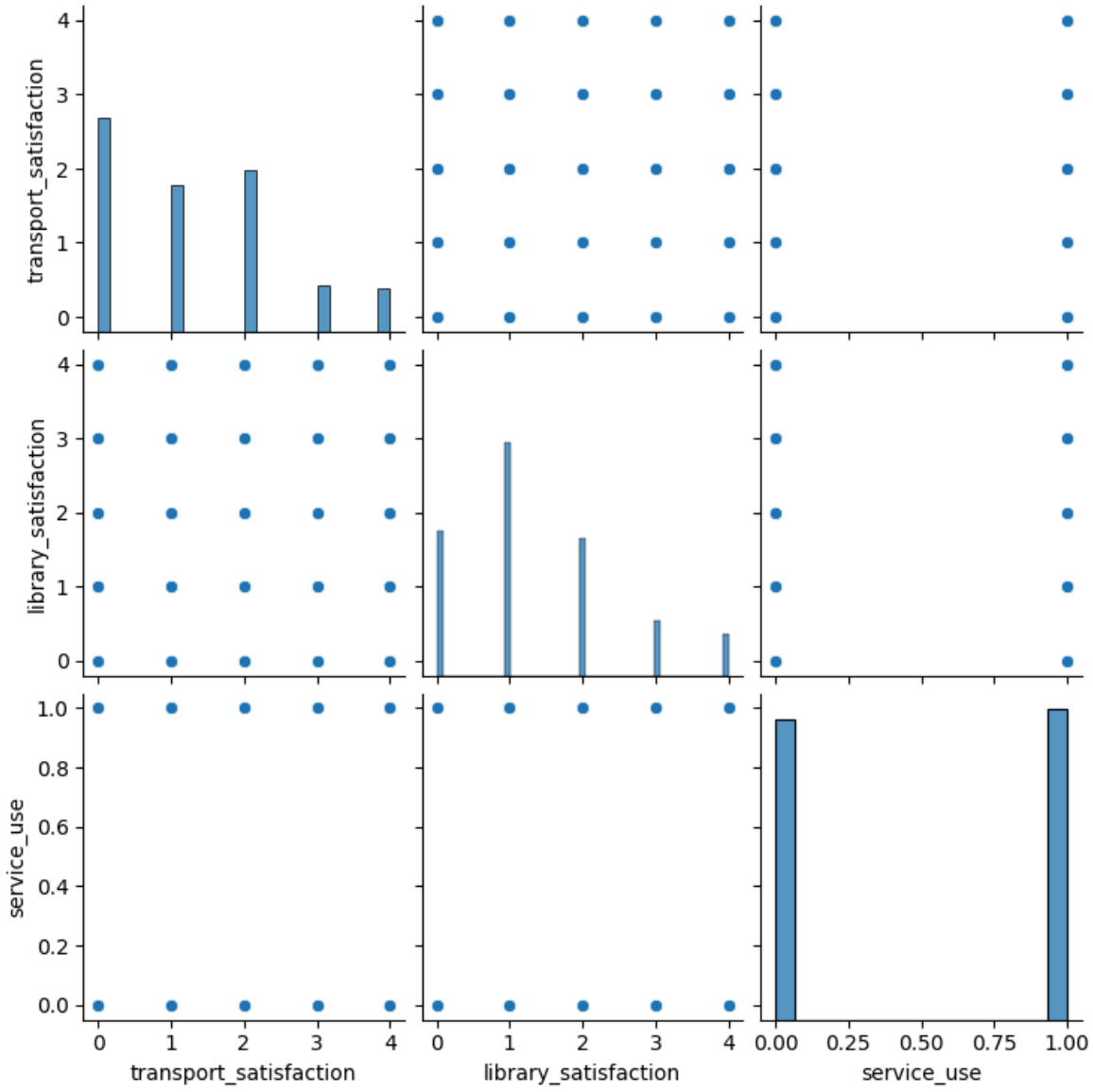
```
C:\Users\pcslg\AppData\Local\Temp\ipykernel_11496\1985317644.py:2: FutureWarning:

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.violinplot(data=df, x='age_group', y='local_service_satisfaction', palette
='muted')
```



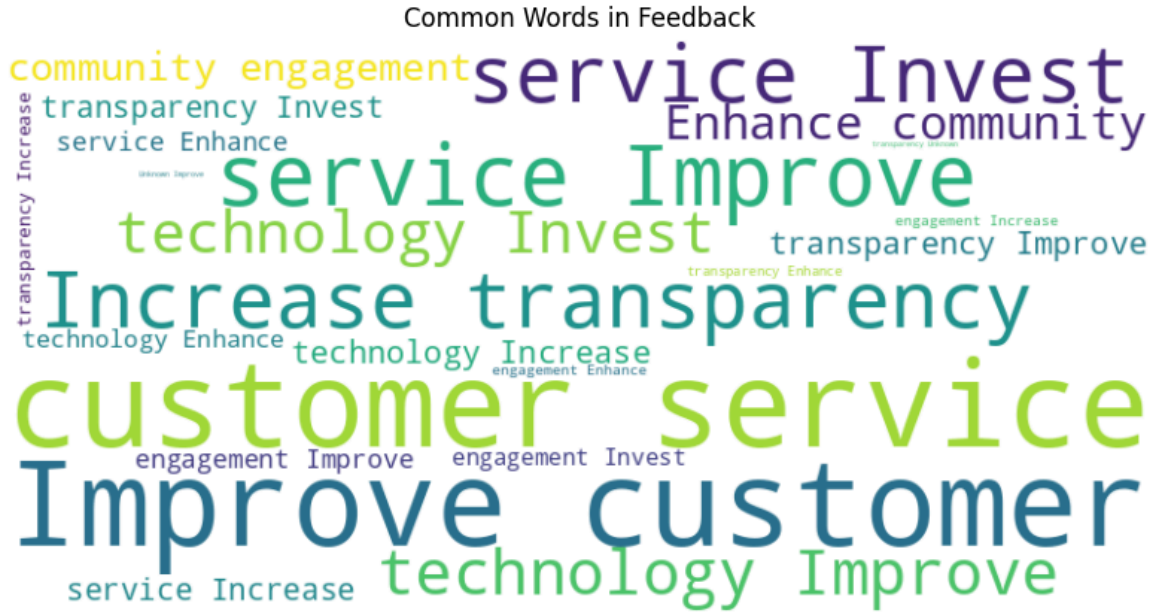
In [16]:
```
age_gender = pd.crosstab(df['age_group'], df['gender'], normalize='index')
age_gender.plot(kind='bar', stacked=True, figsize=(8, 5))
plt.title('Gender Proportion by Age Group')
plt.ylabel('Proportion')
plt.tight_layout()
plt.show()
```



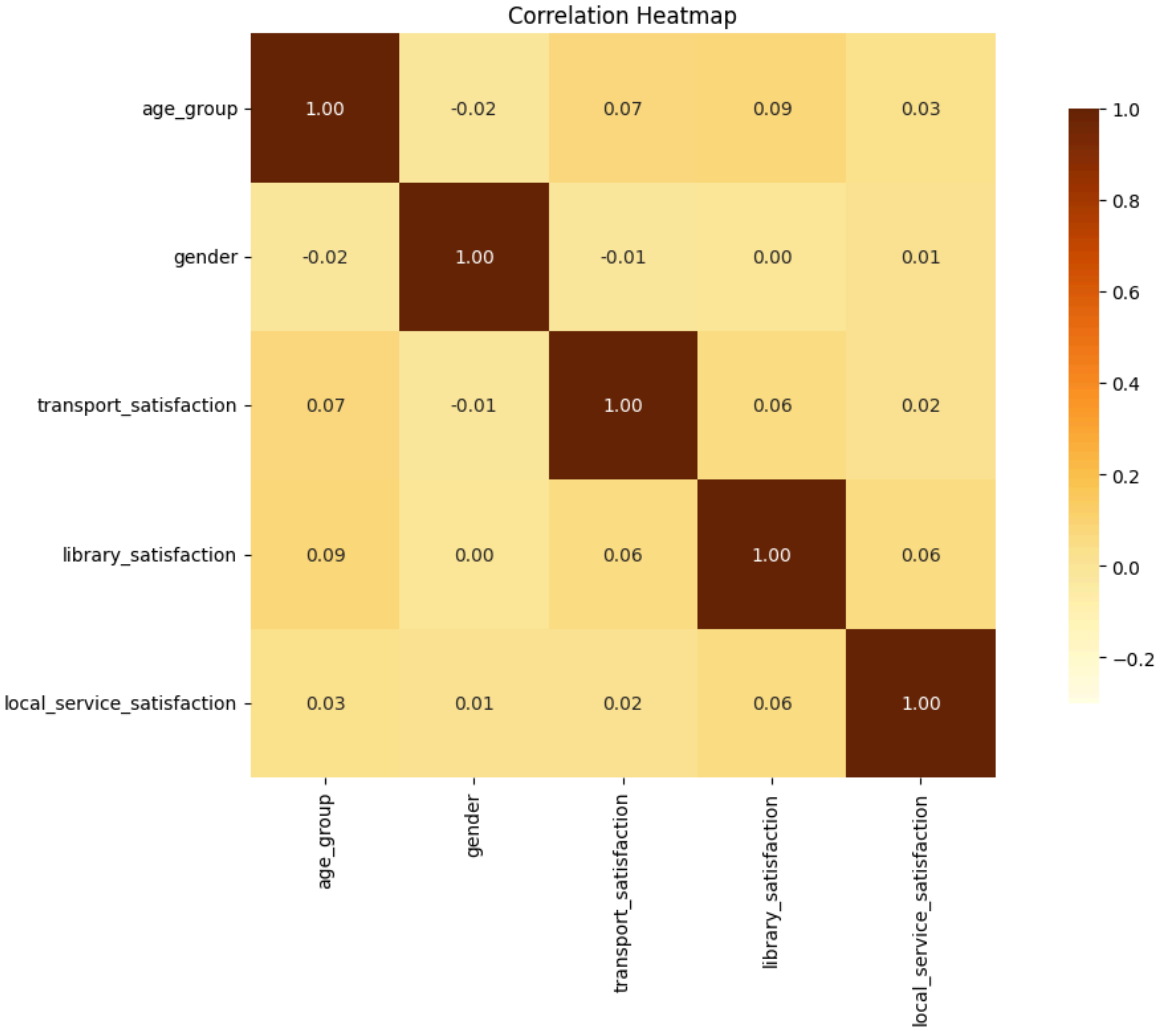
In [62]:
```
# Encode categorical variables (example: using LabelEncoder)
from sklearn.preprocessing import LabelEncoder

df_encoded = df.copy()
label_encoders = {}

for col in df.columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df[col])
    label_encoders[col] = le  # if you want to decode later

# Now create a pairplot
sns.pairplot(df_encoded[['transport_satisfaction', 'library_satisfaction', 'serv
plt.show()
```



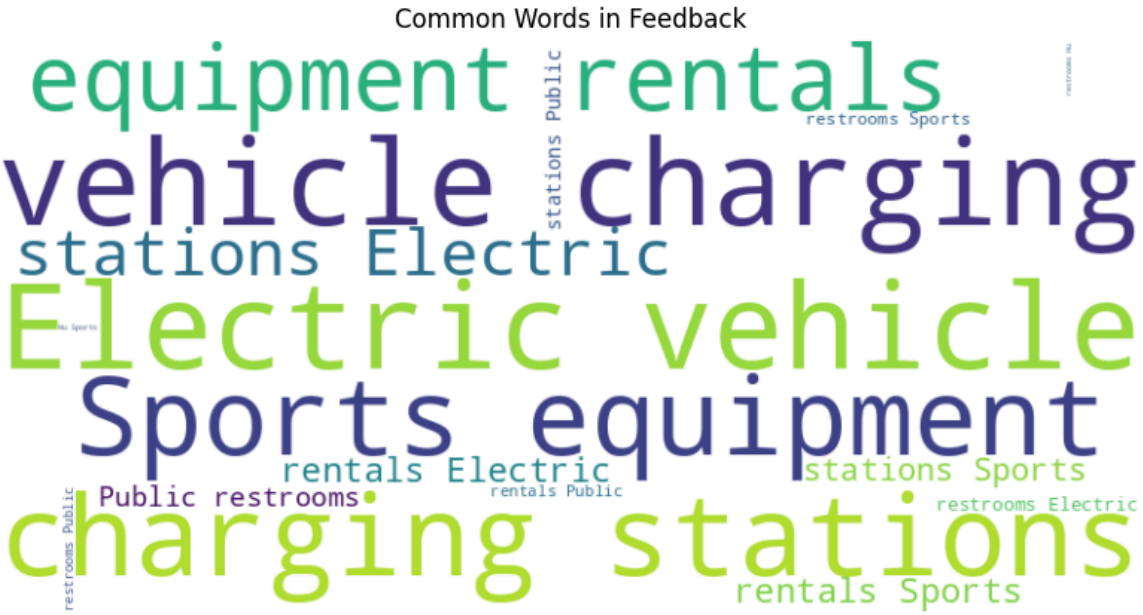
In [ ]:
```
# Word Cloud of Feedback Text (optional for fun)
text = ' '.join(df['local_service_suggestions'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Common Words in Feedback')
plt.show()
```



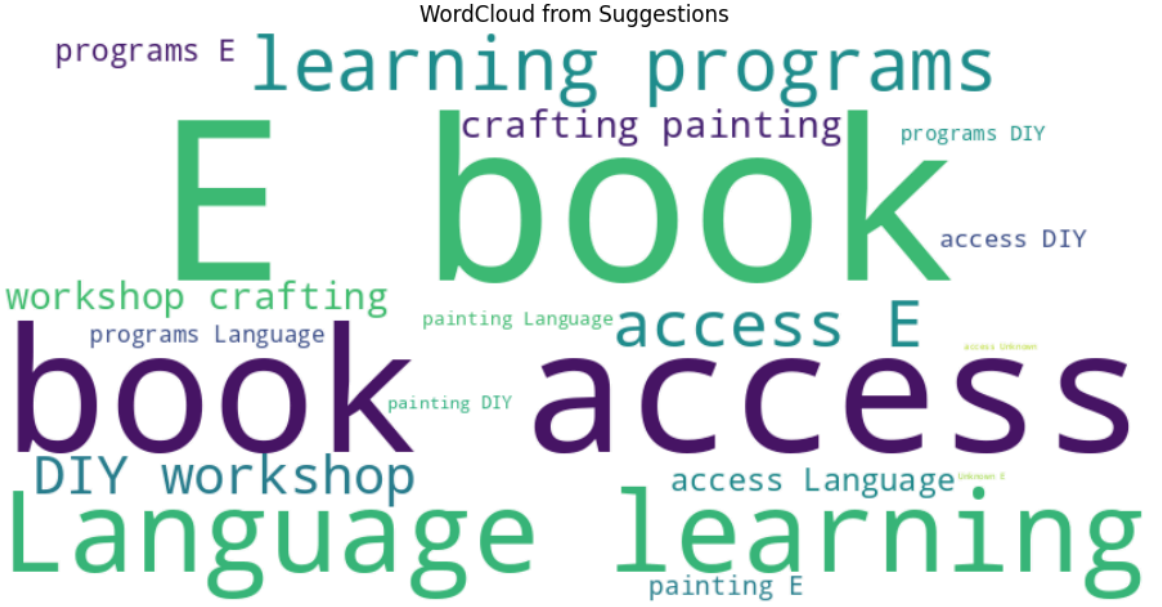
In [89]:
```
corr = df_encoded[['age_group', 'gender', 'transport_satisfaction', 'library_sat
plt.figure(figsize=(12, 8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='YlOrBr', square=True, cbar_kws={"
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



In [19]:
```
# Word Cloud of Feedback Text (optional for fun)
text = ' '.join(df['park_suggestions'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Common Words in Feedback')
plt.show()
```



In [20]:
```
# 5. WordCloud: all suggestion columns
# text_cols = ["transport_suggestions", "park_suggestions", "library_suggestions
# all_text = " ".join(df[col].fillna("") for col in text_cols)
text = ' '.join(df['library_suggestions'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('WordCloud from Suggestions')
plt.tight_layout()
plt.savefig("../data/exports/eda_plots/wordcloud.png")
plt.show()
```


WordCloud from Suggestions
programs E
learning programs
E book
crafting painting
programs DIY
access DIY
workshop crafting
painting Language
access E
programs Language
book access
painting DIY
DIY workshop
access Language
Language learning
painting E

In [ ]: