In [1]:
```python
import numpy as np
import pandas as pd
```

In [2]:
```python
df=pd.read_csv('onlinefoods.csv')
```

In [3]:
```python
df
```

Out[3]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	Pin code	Output	Feedback	Unnamed: 12
0	20	Female	Single	Student	No Income	Post Graduate	4	12.9766	77.5993	560001	Yes	Positive	Yes
1	24	Female	Single	Student	Below Rs.10000	Graduate	3	12.9770	77.5773	560009	Yes	Positive	Yes
2	22	Male	Single	Student	Below Rs.10000	Post Graduate	3	12.9551	77.6593	560017	Yes	Negative	Yes
3	22	Female	Single	Student	No Income	Graduate	6	12.9473	77.5616	560019	Yes	Positive	Yes
4	22	Male	Single	Student	Below Rs.10000	Post Graduate	4	12.9850	77.5533	560010	Yes	Positive	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...
383	23	Female	Single	Student	No Income	Post Graduate	2	12.9766	77.5993	560001	Yes	Positive	Yes
384	23	Female	Single	Student	No Income	Post Graduate	4	12.9854	77.7081	560048	Yes	Positive	Yes
385	22	Female	Single	Student	No Income	Post Graduate	5	12.9850	77.5533	560010	Yes	Positive	Yes
386	23	Male	Single	Student	Below Rs.10000	Post Graduate	2	12.9770	77.5773	560009	Yes	Positive	Yes
387	23	Male	Single	Student	No Income	Post Graduate	5	12.8988	77.5764	560078	Yes	Positive	Yes

388 rows × 13 columns

In [5]:
```python
df.dtypes
```

Out[5]:
```
Age                             int64
Gender                         object
Marital Status                 object
Occupation                     object
Monthly Income                 object
Educational Qualifications     object
Family size                     int64
latitude                      float64
longitude                     float64
Pin code                        int64
Output                         object
Feedback                       object
Unnamed: 12                    object
dtype: object
```

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

Out[6]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	Pin code	Output	Feedback	Unnamed: 12
0	20	Female	Single	Student	No Income	Post Graduate	4	12.9766	77.5993	560001	Yes	Positive	Yes
1	24	Female	Single	Student	Below Rs.10000	Graduate	3	12.9770	77.5773	560009	Yes	Positive	Yes
2	22	Male	Single	Student	Below Rs.10000	Post Graduate	3	12.9551	77.6593	560017	Yes	Negative	Yes
3	22	Female	Single	Student	No Income	Graduate	6	12.9473	77.5616	560019	Yes	Positive	Yes
4	22	Male	Single	Student	Below Rs.10000	Post Graduate	4	12.9850	77.5533	560010	Yes	Positive	Yes

In [7]:
```python
df.drop(columns='Unnamed: 12', inplace=True)
```

In [8]:
```python
df.head()
```

Out[8]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	Pin code	Output	Feedback
0	20	Female	Single	Student	No Income	Post Graduate	4	12.9766	77.5993	560001	Yes	Positive
1	24	Female	Single	Student	Below Rs.10000	Graduate	3	12.9770	77.5773	560009	Yes	Positive
2	22	Male	Single	Student	Below Rs.10000	Post Graduate	3	12.9551	77.6593	560017	Yes	Negative
3	22	Female	Single	Student	No Income	Graduate	6	12.9473	77.5616	560019	Yes	Positive
4	22	Male	Single	Student	Below Rs.10000	Post Graduate	4	12.9850	77.5533	560010	Yes	Positive

In [9]:
```python
df.shape
```

Out[9]:
```
(388, 12)
```

In [10]:
```python
df.describe()
```

Out[10]:

	Age	Family size	latitude	longitude	Pin code
count	388.000000	388.000000	388.000000	388.000000	388.000000
mean	24.628866	3.280928	12.972058	77.600160	560040.113402
std	2.975593	1.351025	0.044489	0.051354	31.399609
min	18.000000	1.000000	12.865200	77.484200	560001.000000
25%	23.000000	2.000000	12.936900	77.565275	560010.750000
50%	24.000000	3.000000	12.977000	77.592100	560033.500000
75%	26.000000	4.000000	12.997025	77.630900	560068.000000
max	33.000000	6.000000	13.102000	77.758200	560109.000000

In [11]:
```python
df.columns
```

Out[11]:
```
Index(['Age', 'Gender', 'Marital Status', 'Occupation', 'Monthly Income',
       'Educational Qualifications', 'Family size', 'latitude', 'longitude',
       'Pin code', 'Output', 'Feedback'],
      dtype='object')
```

In [12]:
```python
df.tail()
```

Out[12]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	Pin code	Output	Feedback
383	23	Female	Single	Student	No Income	Post Graduate	2	12.9766	77.5993	560001	Yes	Positive
384	23	Female	Single	Student	No Income	Post Graduate	4	12.9854	77.7081	560048	Yes	Positive
385	22	Female	Single	Student	No Income	Post Graduate	5	12.9850	77.5533	560010	Yes	Positive
386	23	Male	Single	Student	Below Rs.10000	Post Graduate	2	12.9770	77.5773	560009	Yes	Positive
387	23	Male	Single	Student	No Income	Post Graduate	5	12.8988	77.5764	560078	Yes	Positive

In [13]:
```python
print (df.count())
```

```
Age                           388
Gender                        388
Marital Status                388
Occupation                    388
Monthly Income                388
Educational Qualifications    388
Family size                   388
latitude                      388
longitude                     388
Pin code                      388
Output                        388
Feedback                      388
dtype: int64
```

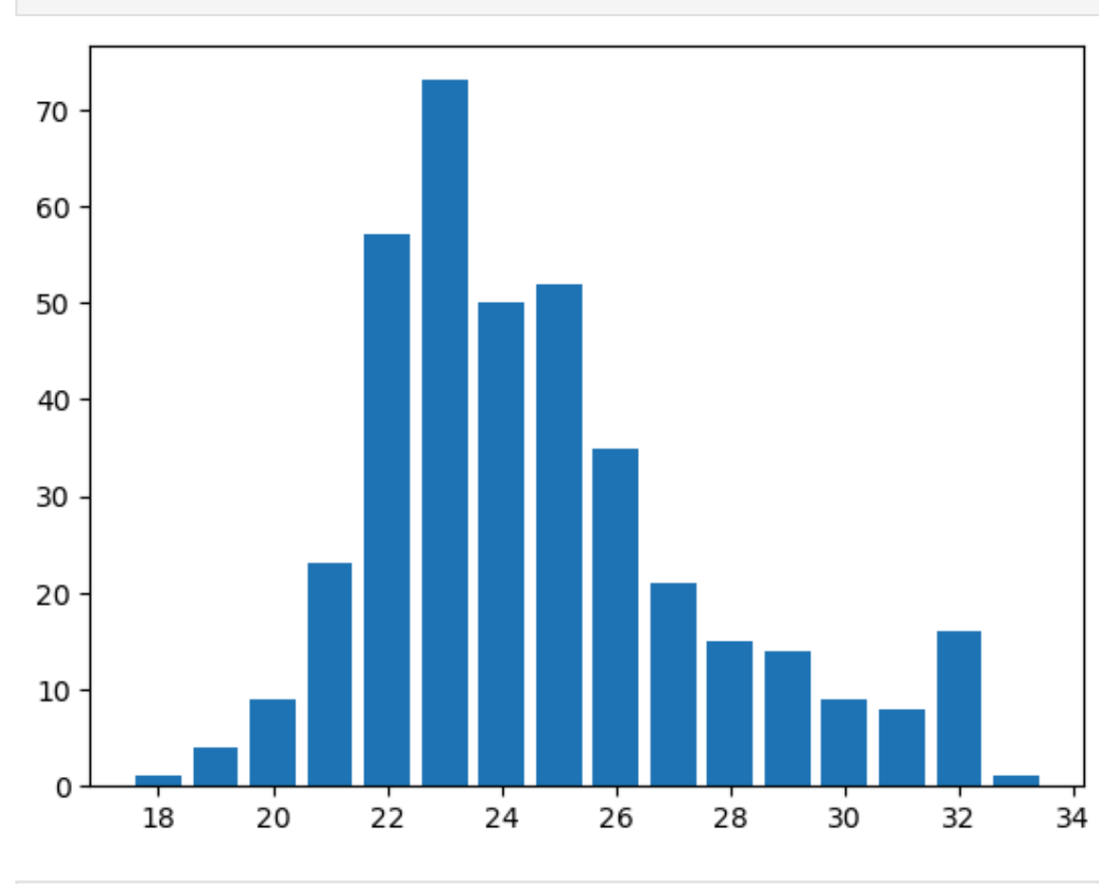
In [16]:
```python
df.isnull().sum()
```

Out[16]:
```
Age                           0
Gender                        0
Marital Status                0
Occupation                    0
Monthly Income                0
Educational Qualifications    0
Family size                   0
latitude                      0
longitude                     0
Pin code                      0
Output                        0
Feedback                      0
dtype: int64
```

In [17]:
```python
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')
```

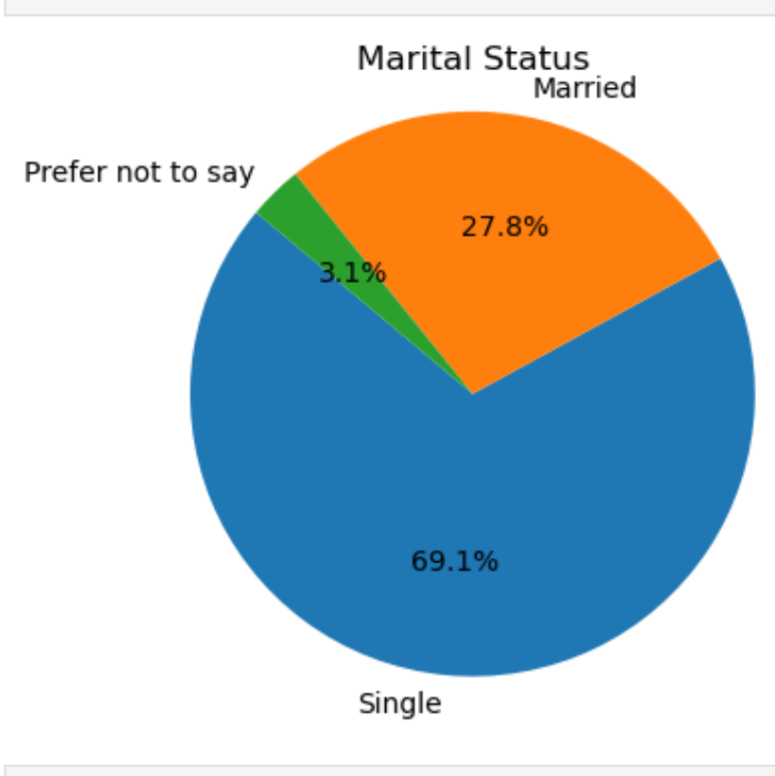
#distribution of individuals across different age groups

In [24]:
```python
age_counts =df['Age'].value_counts().sort_index()

plt.bar(age_counts.index, age_counts.values)
plt.show()
```

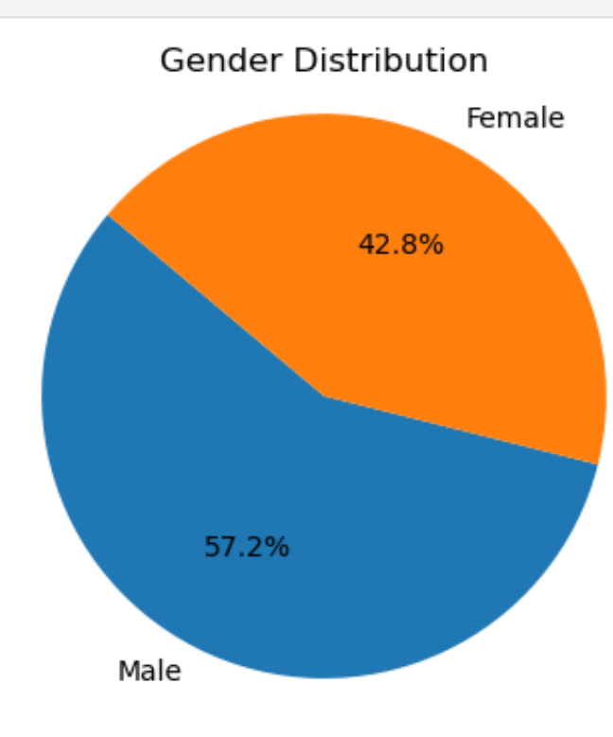


In [22]:
```python
import seaborn as sns
```

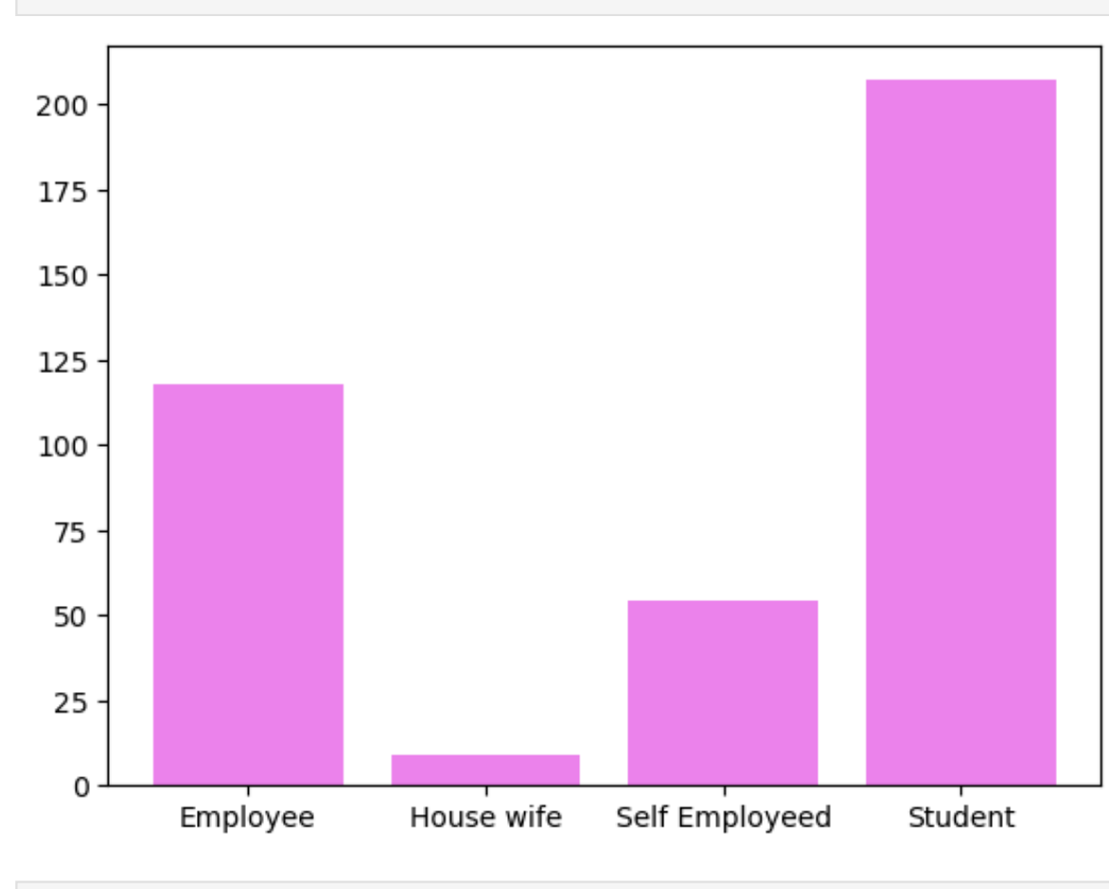
In [30]:
```python
# Count the occurrences of each Marital Status category
MaritalStatus_counts = df['Marital Status'].value_counts()

# Create a pie chart

plt.figure(figsize=(4, 4))
plt.pie(MaritalStatus_counts, labels=MaritalStatus_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Marital Status')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```



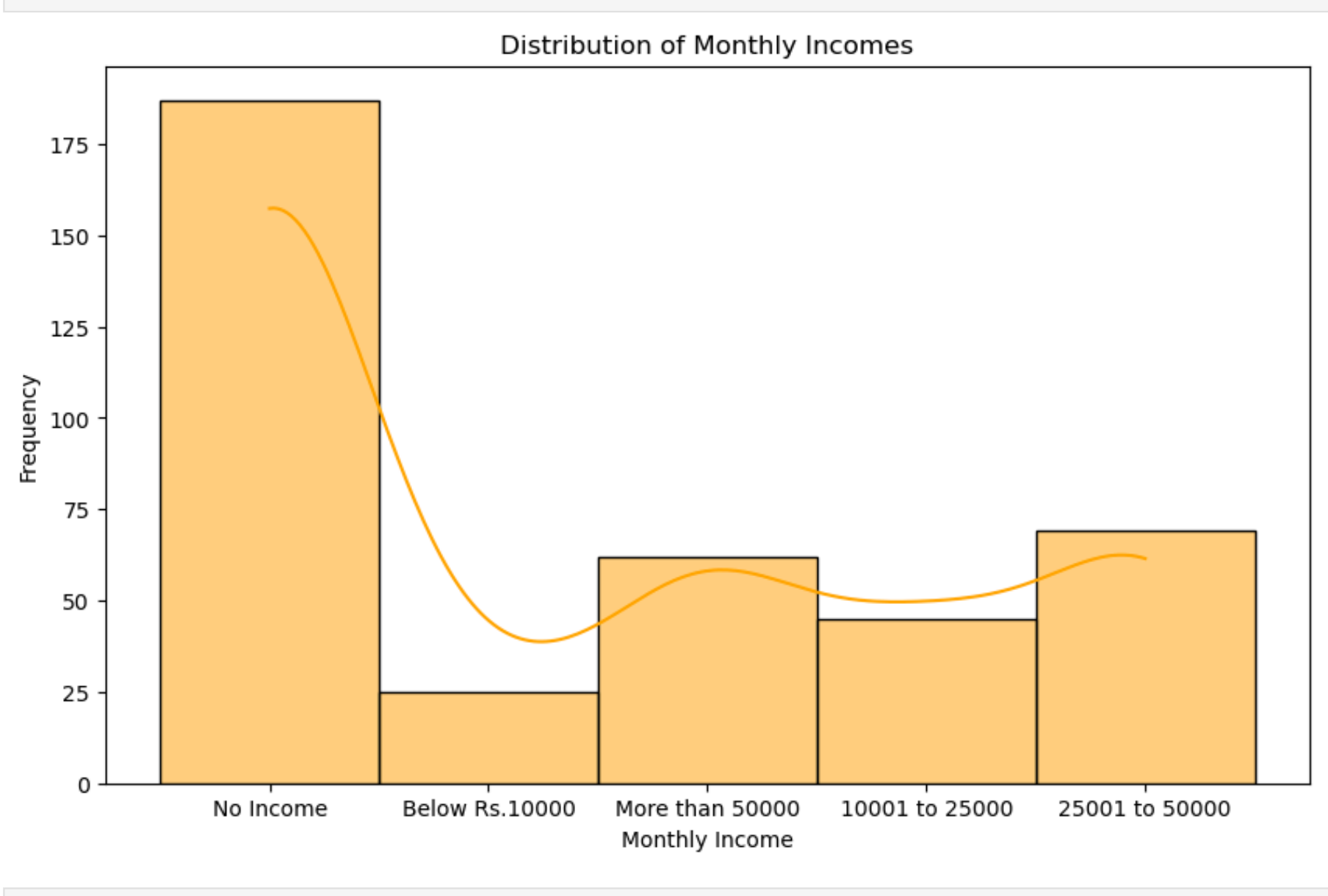
In [26]:
```python
# Count the occurrences of each gender category
gender_counts = df['Gender'].value_counts()

# Create a pie chart

plt.figure(figsize=(4, 4))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Gender Distribution')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```



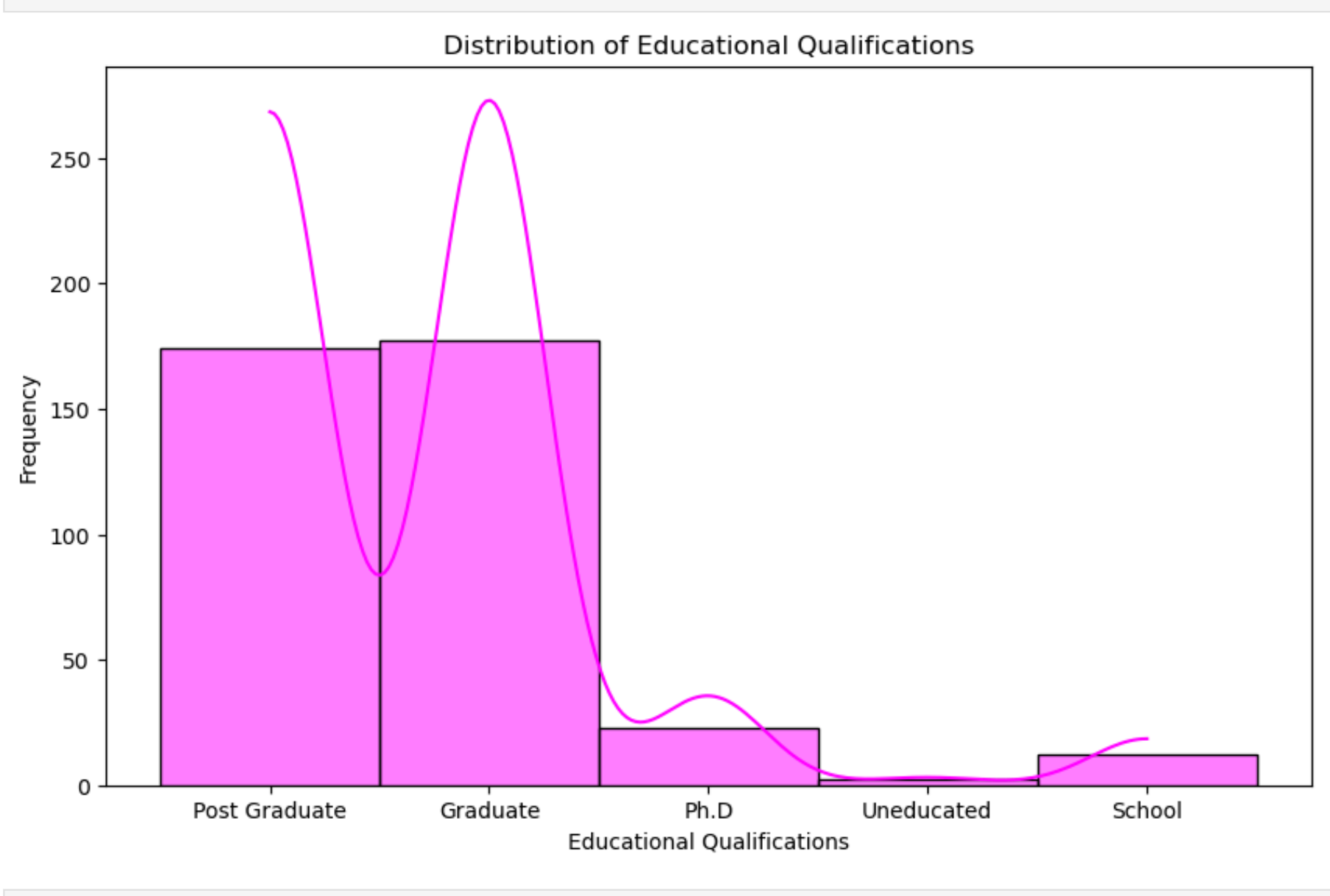
In [33]:
```python
Occupation_counts =df['Occupation'].value_counts().sort_index()

plt.bar(Occupation_counts.index, Occupation_counts.values,color='violet')
plt.show()
```



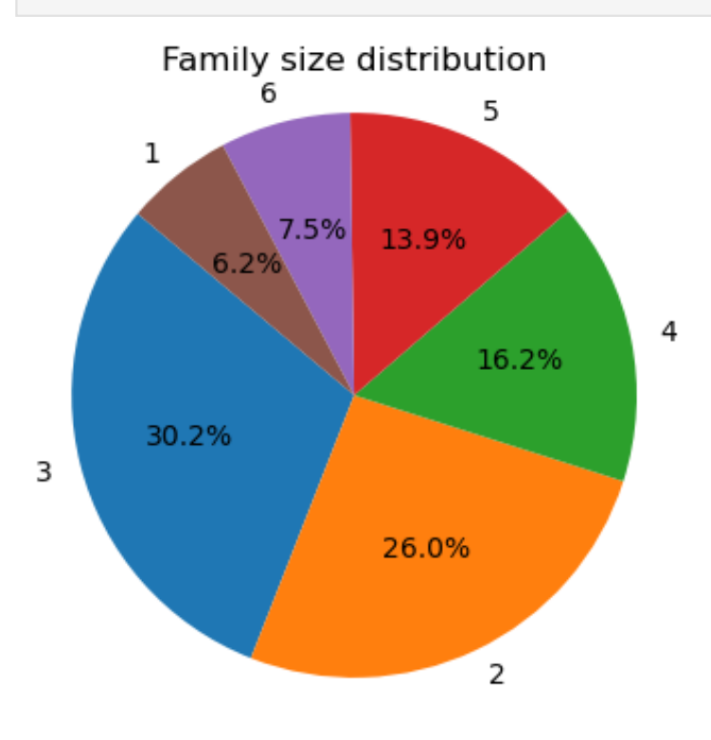
In [34]:
```python
# Histogram for Monthly Income distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['Monthly Income'], bins=20, kde=True,color='orange')
plt.title('Distribution of Monthly Incomes')
plt.xlabel('Monthly Income')
plt.ylabel('Frequency')
plt.show()
```



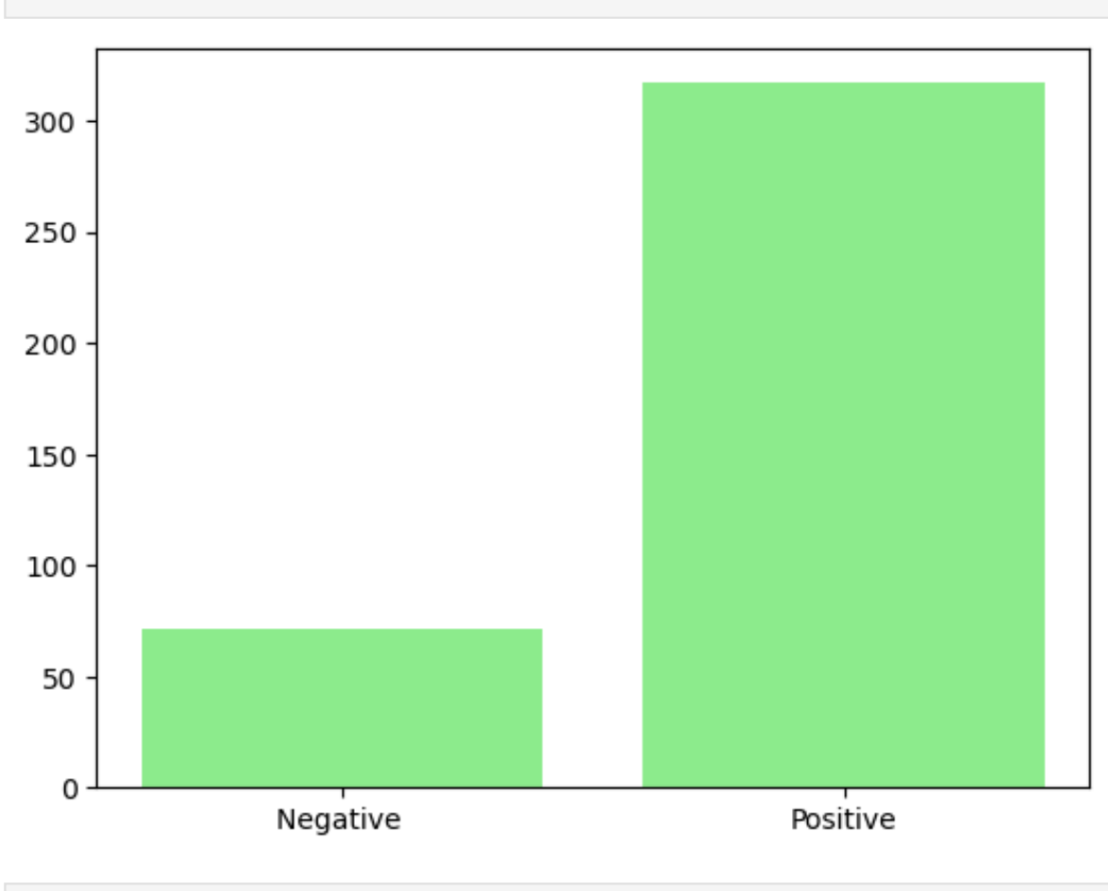
In [35]:
```python
# Histogram for Educational Qualifications distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['Educational Qualifications'], bins=20, kde=True,color='magenta')
plt.title('Distribution of Educational Qualifications')
plt.xlabel('Educational Qualifications')
plt.ylabel('Frequency')
plt.show()
```



In [38]:
```python
# Count the occurrences of each Family size  category
Familysize_counts = df['Family size'].value_counts()

# Create a pie chart

plt.figure(figsize=(4, 4))
plt.pie(Familysize_counts, labels=Familysize_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Family size distribution')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```

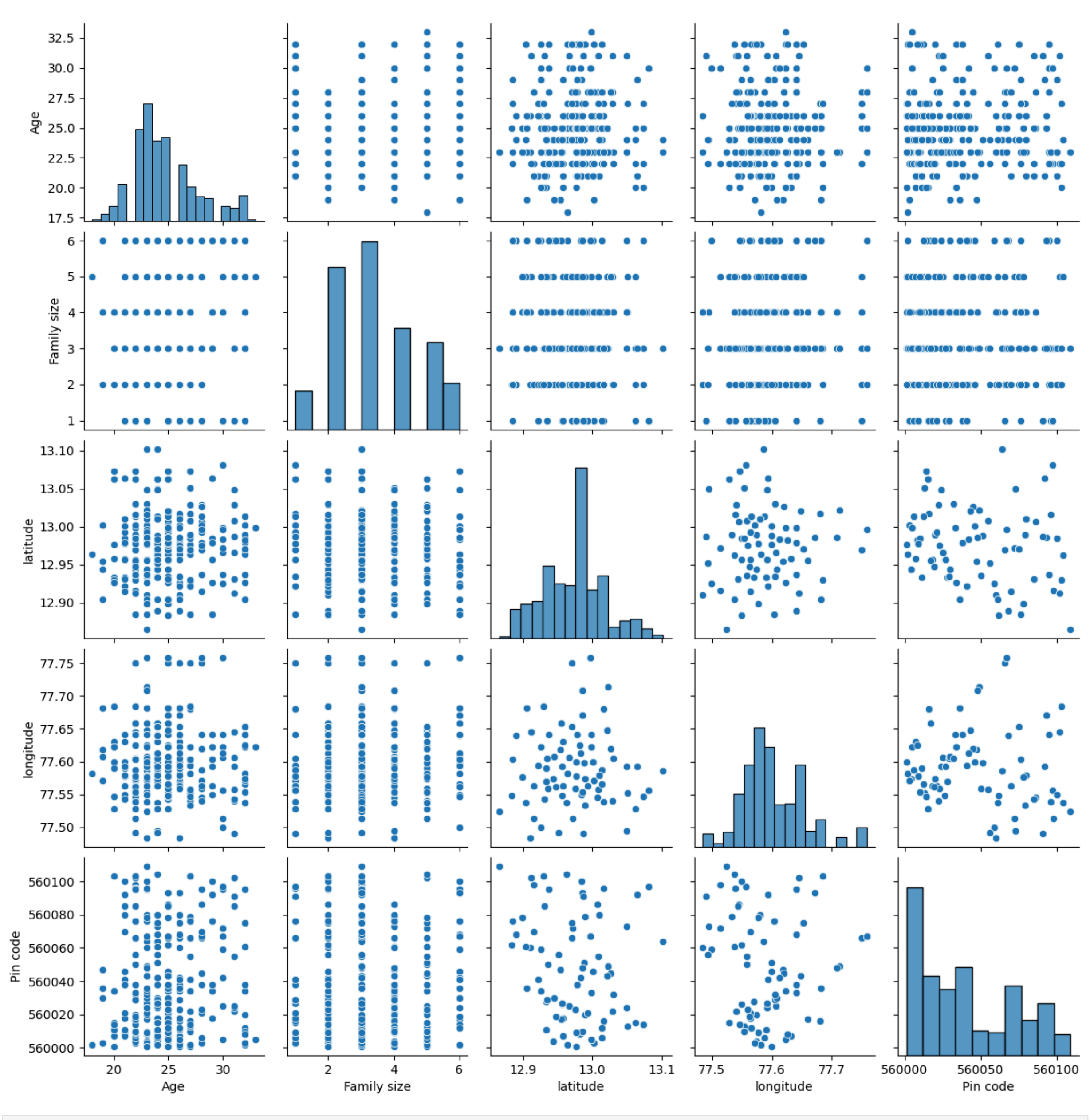


In [40]:
```python
Feedback_counts =df['Feedback'].value_counts().sort_index()

plt.bar(Feedback_counts.index, Feedback_counts.values,color='lightgreen')
plt.show()
```



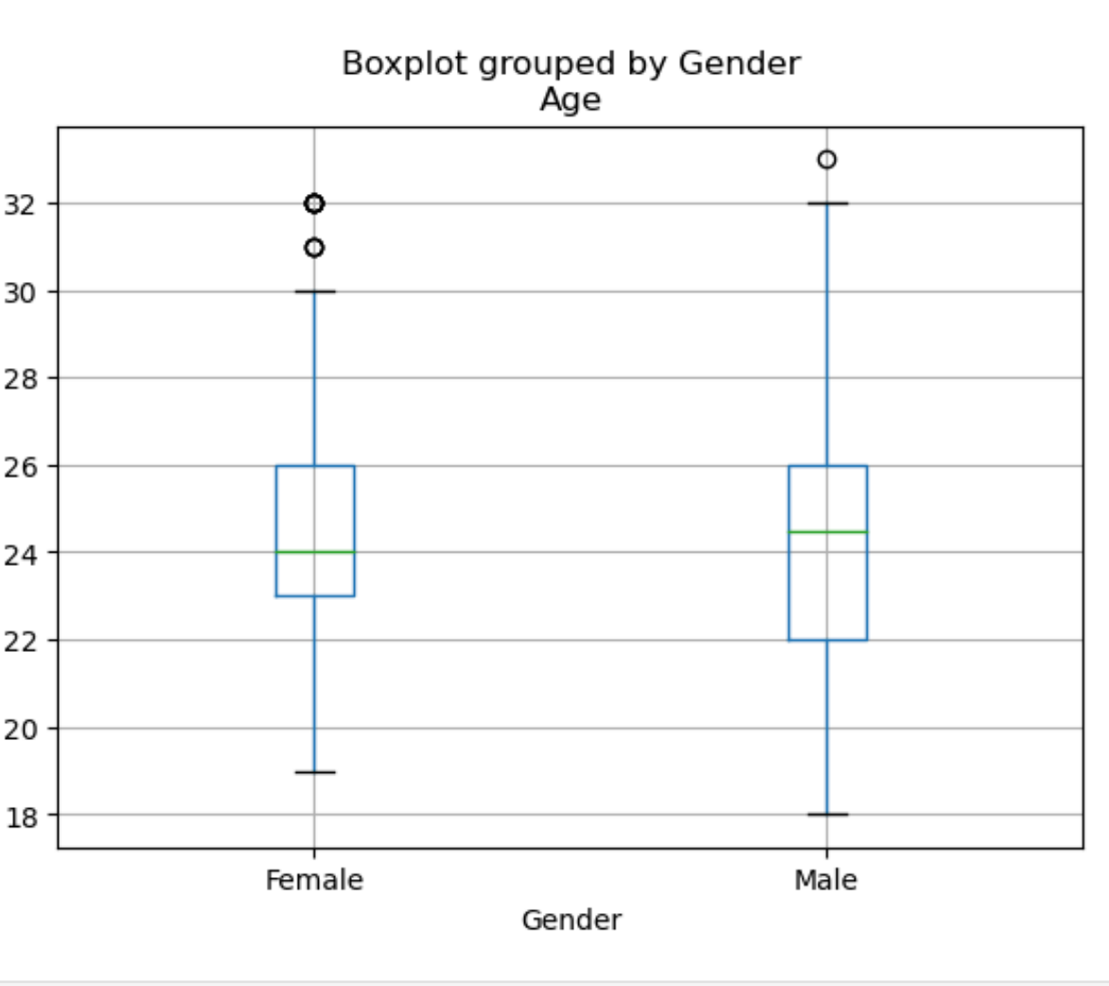
In [41]:
```python
sns.pairplot(df)
```

Out[41]:
```
<seaborn.axisgrid.PairGrid at 0x1b4d0fbda10>
```



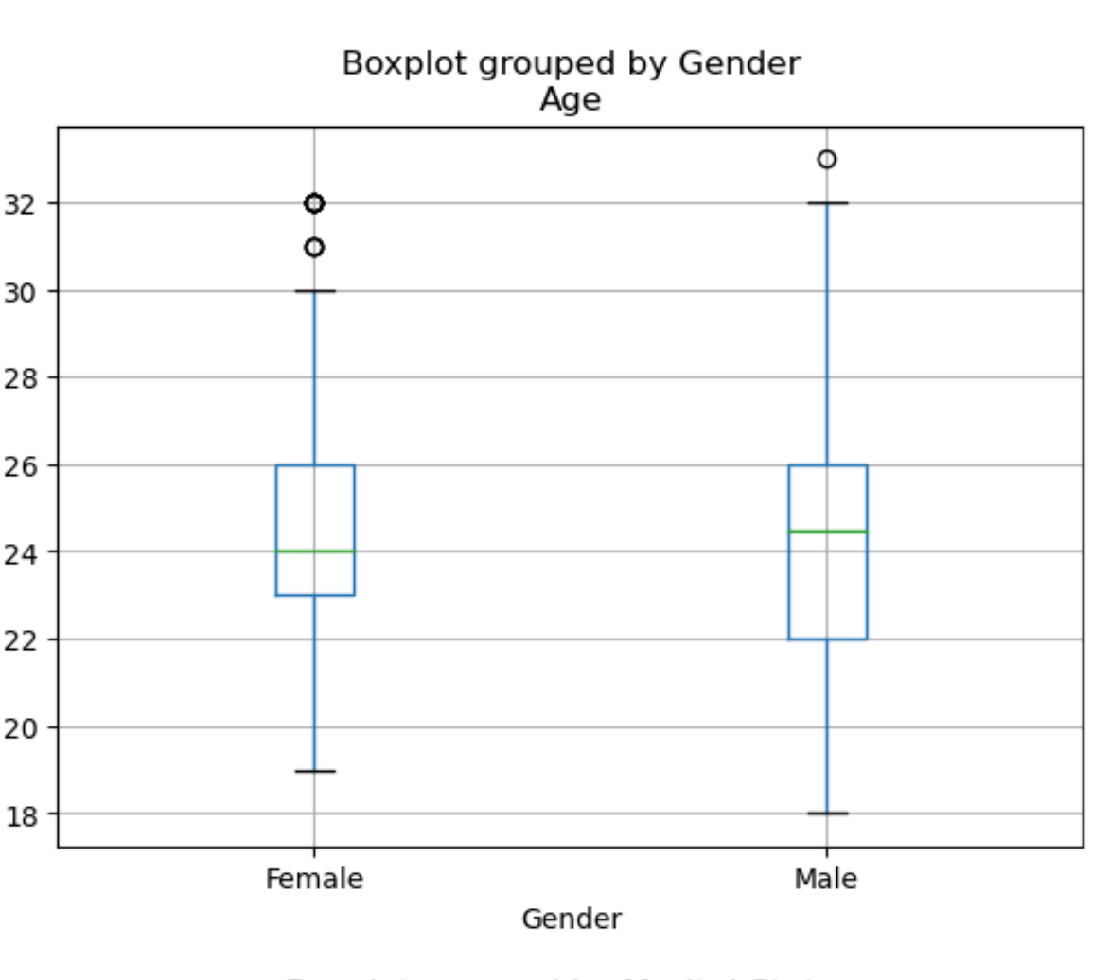
In [43]:
```python
df.boxplot(column='Age',by='Gender')
```

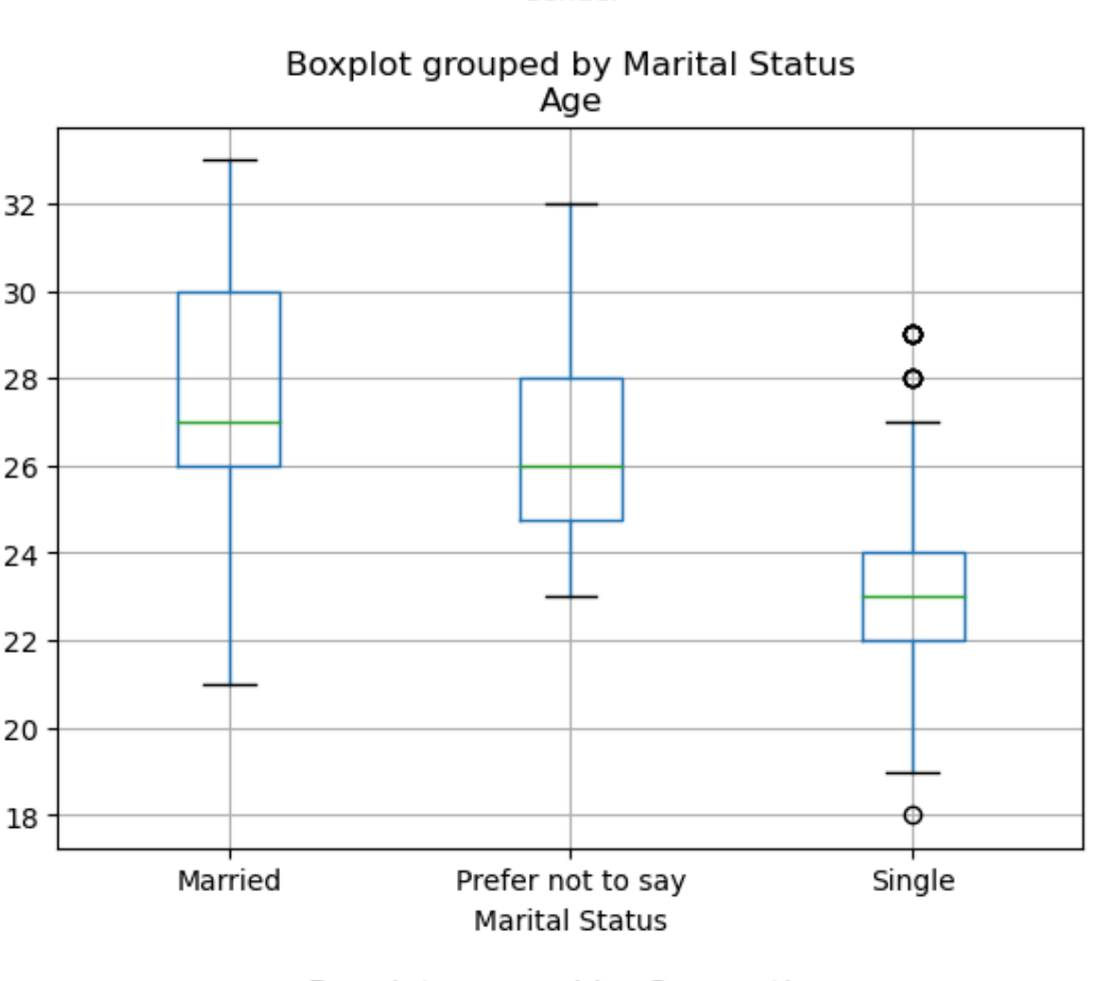
Out[43]:
```
<Axes: title={'center': 'Age'}, xlabel='Gender'>
```

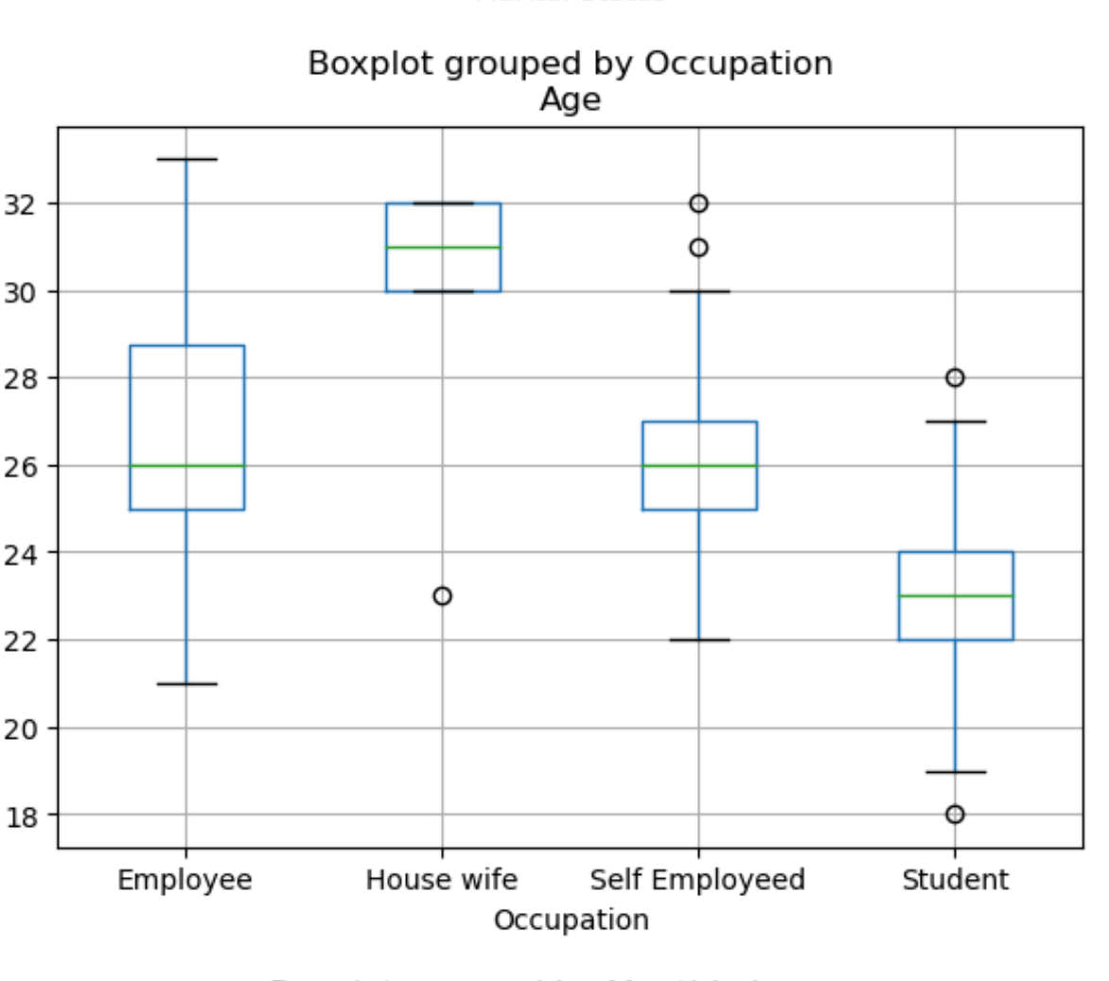


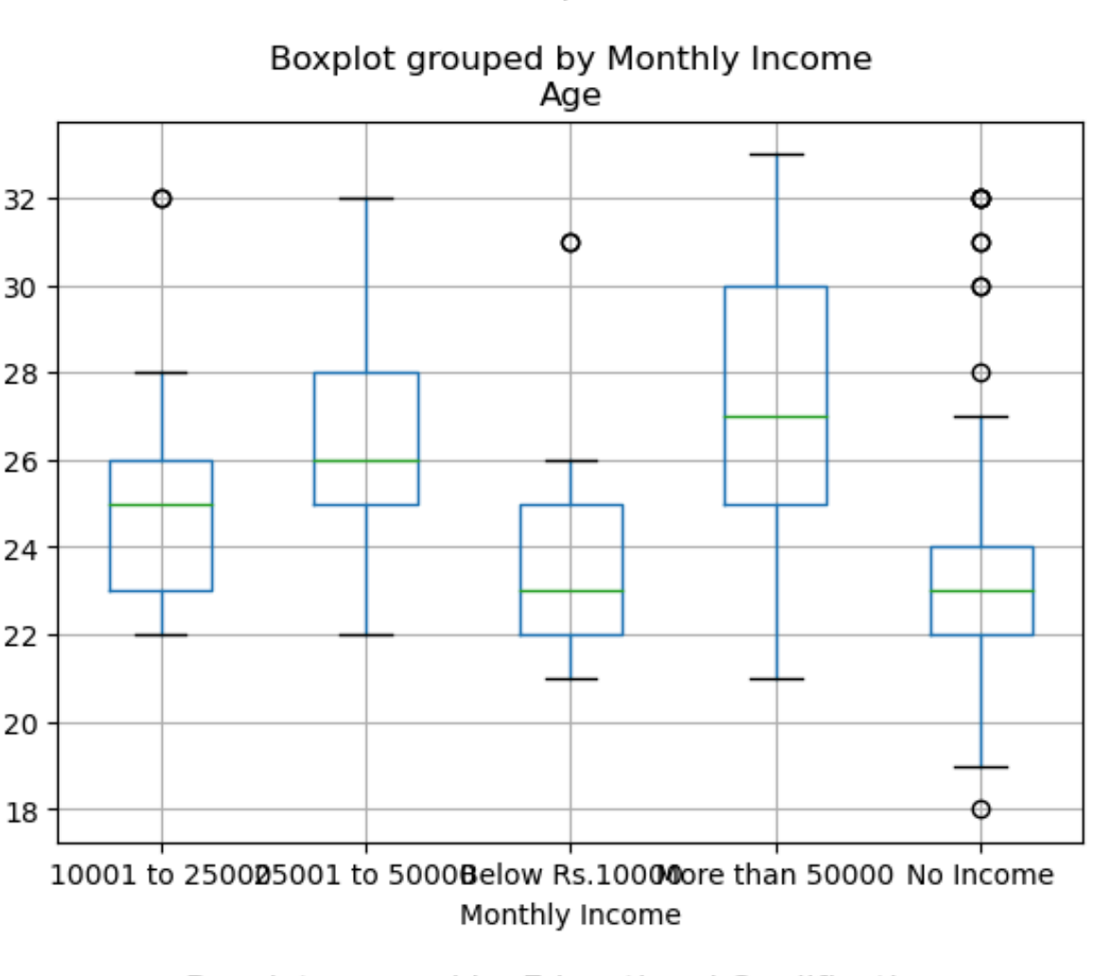
In [44]:
```python
df.boxplot(column='Age',by='Gender')
df.boxplot(column='Age',by='Marital Status')
df.boxplot(column='Age',by='Occupation')
df.boxplot(column='Age',by='Monthly Income')
df.boxplot(column='Age',by='Educational Qualifications')
df.boxplot(column='Age',by='Family size')
```

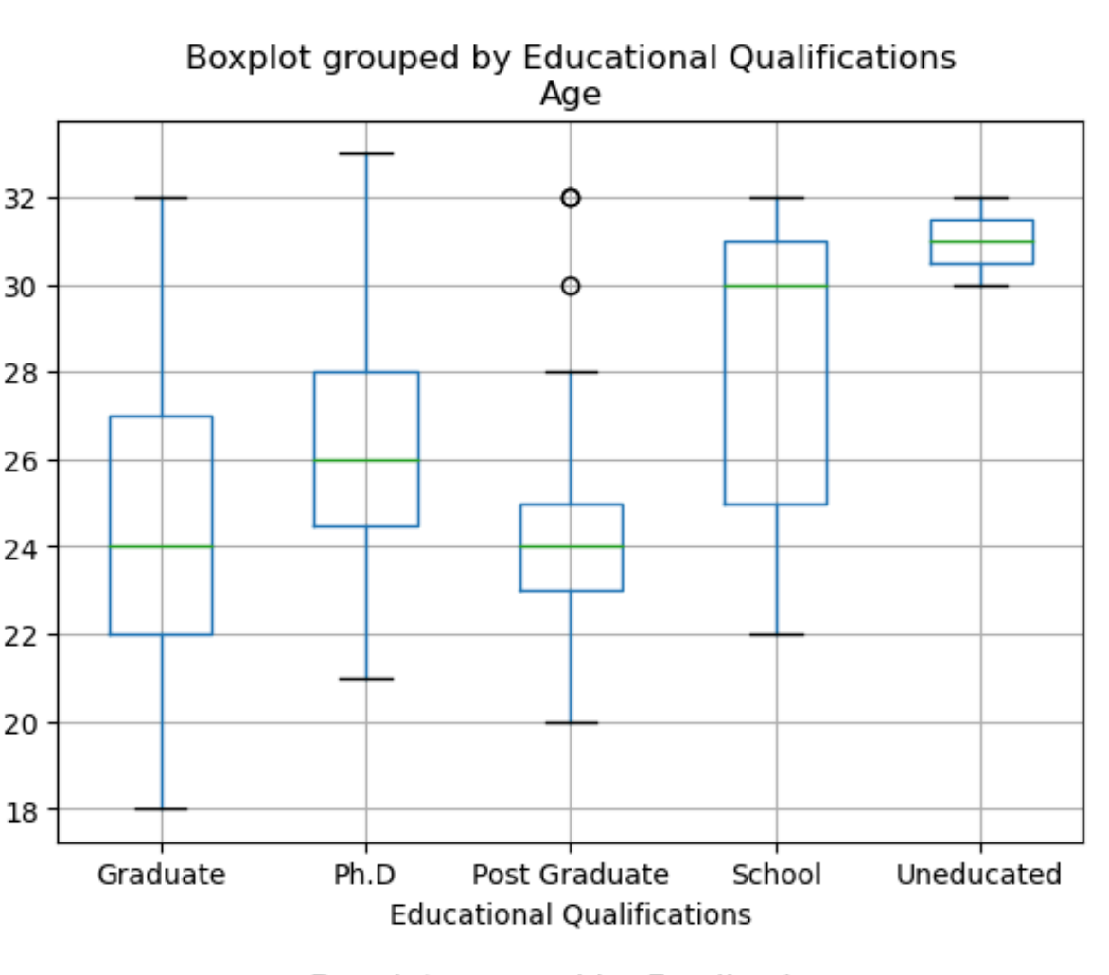
Out[44]:
```
<Axes: title={'center': 'Age'}, xlabel='Family size'>
```

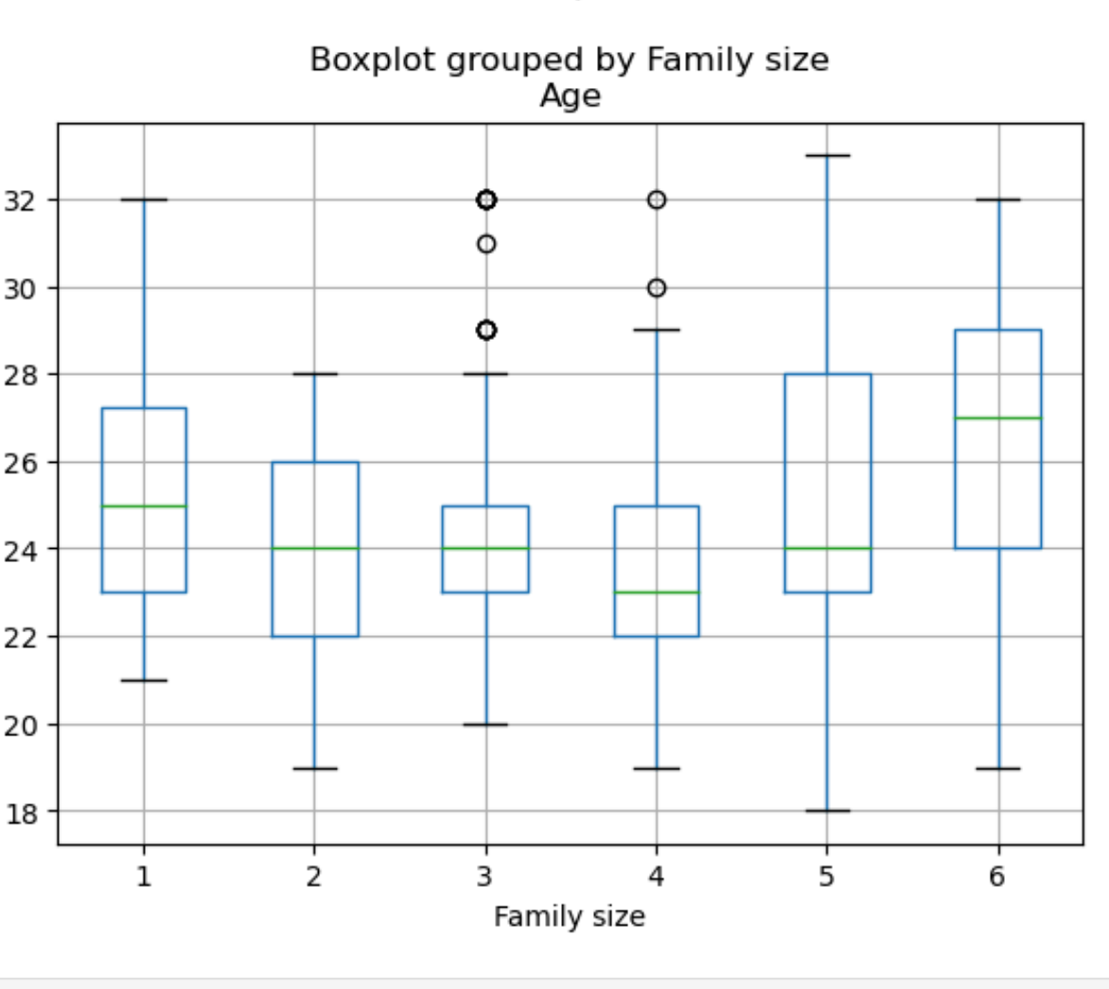




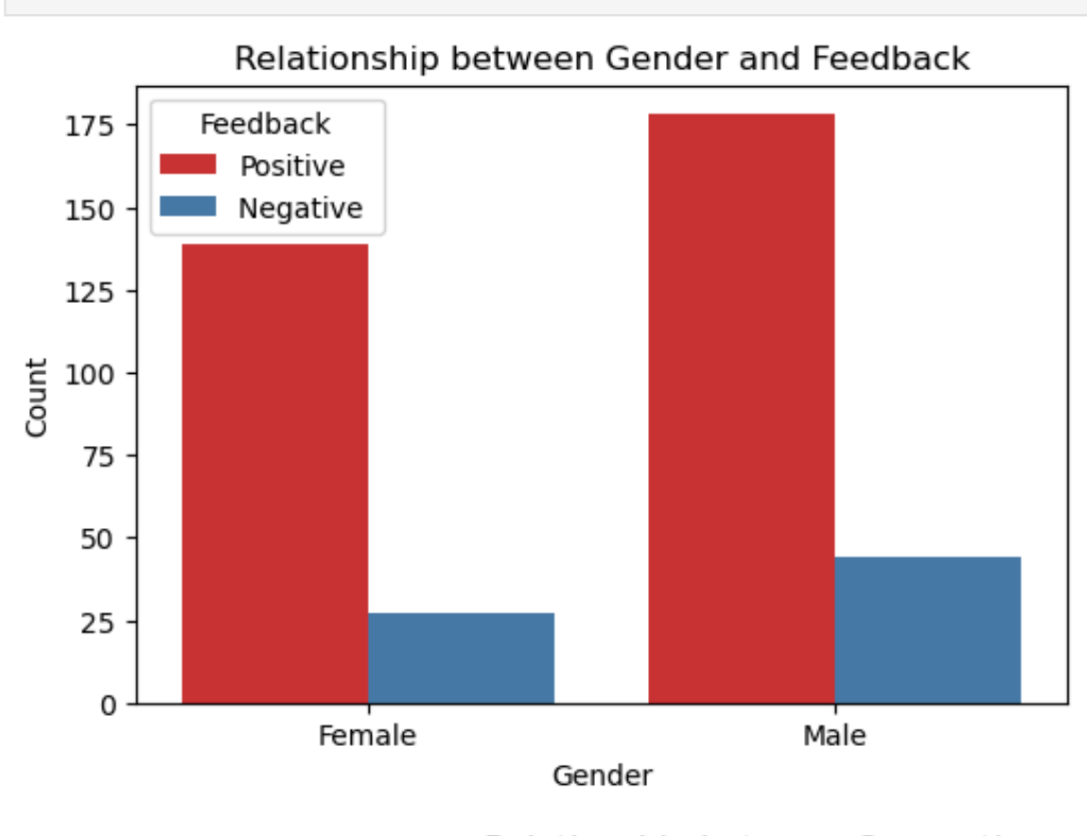


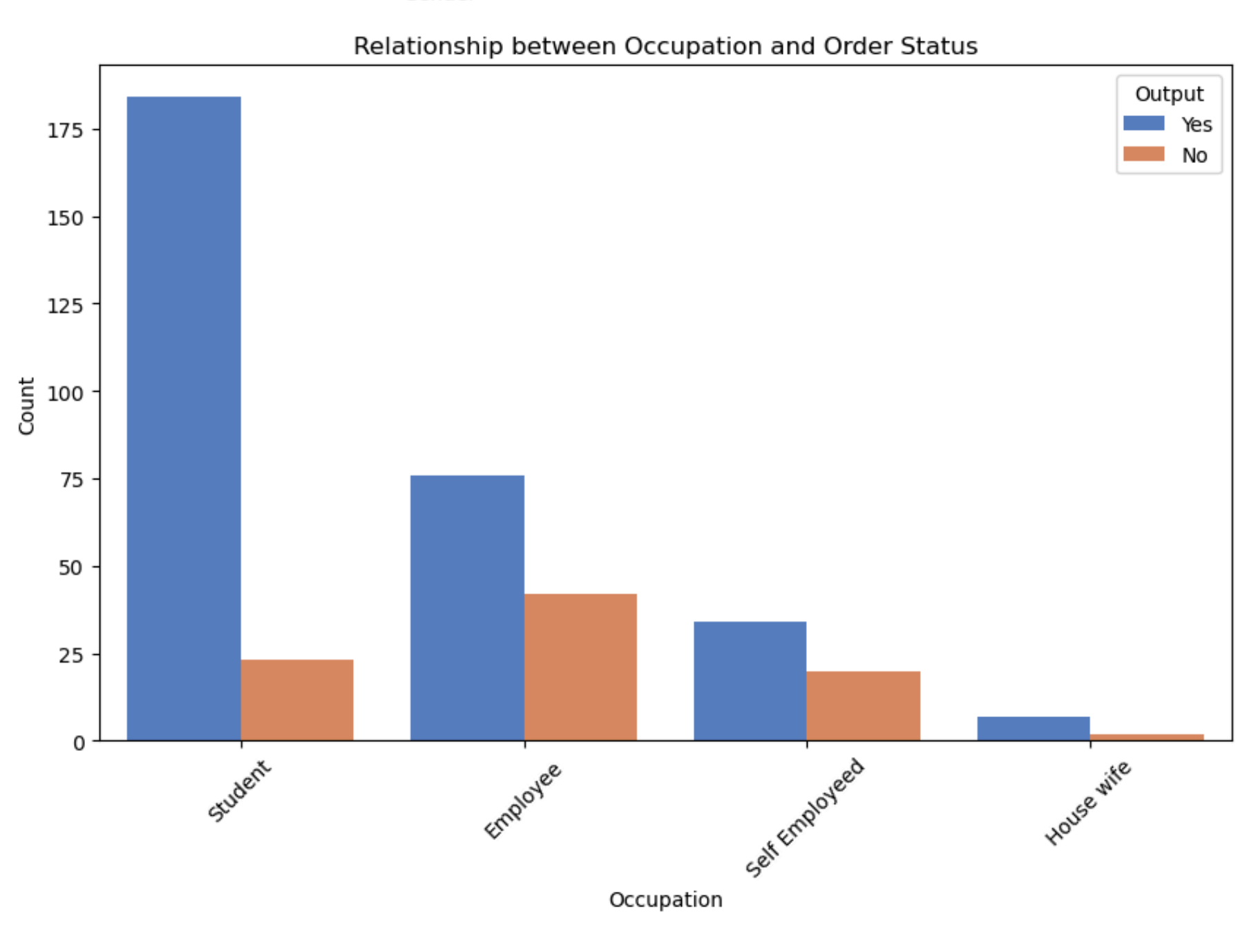






In [46]:
```python
#Bivariate Analysis#
# Relationship between Gender and Feedback
plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='Gender', hue='Feedback', palette='Set1')
plt.title('Relationship between Gender and Feedback')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

# Relationship between Occupation and Order Status
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Occupation', hue='Output', palette='muted')
plt.title('Relationship between Occupation and Order Status')
plt.xlabel('Occupation')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```





Hypothesis Testing

Can we use "Gender" to predict "Educational Qualifications" ?. Chi-Square Test for Independence can answer this question as "Gender" and "Educational Qualifications" are categorical features. Hypothesis Testing : Chi-Square Test Null Hypothesis (H0): "Gender" of the employee and "Educational Qualifications" are independent Alternate Hypothesis (H1): "Gender" of the employee and "Educational Qualifications" are not independent(dependent).

In [55]:
```python
df_chi_square = df.loc[:, ['Gender','Educational Qualifications']]
df_chi_square
```

Out[55]:

	Gender	Educational Qualifications
0	Female	Post Graduate
1	Female	Graduate
2	Male	Post Graduate
3	Female	Graduate
4	Male	Post Graduate
...	...	...
383	Female	Post Graduate
384	Female	Post Graduate
385	Female	Post Graduate
386	Male	Post Graduate
387	Male	Post Graduate

388 rows × 2 columns

In [ ]: