1.Import the Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

2.Importing the dataset

In [2]:

```
df=pd.read_csv("Titanic-Dataset.csv")
```

In [3]:

```
df
```

Loading [MathJax]/extensions/Safe.js

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

891 rows × 12 columns

In [4]:

```
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [5]:

```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [7]:

```
df.shape
```

Out[7]:

```
(891, 12)
```

In [8]:

```
df.info
```

Loading [MathJax]/extensions/Safe.js

Out[8]:
```
<bound method DataFrame.info of      PassengerId  Survived  Pclass  \
0              1         0       3   
1              2         1       1   
2              3         1       3   
3              4         1       1   
4              5         0       3   
..           ...       ...     ...   
886          887         0       2   
887          888         1       1   
888          889         0       3   
889          890         1       1   
890          891         0       3   

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1   
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1   
2                               Heikkinen, Miss. Laina  female  26.0      0   
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1   
4                             Allen, Mr. William Henry    male  35.0      0   
..                                                 ...     ...   ...    ...   
886                              Montvila, Rev. Juozas    male  27.0      0   
887                       Graham, Miss. Margaret Edith  female  19.0      0   
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1   
889                              Behr, Mr. Karl Howell    male  26.0      0   
890                                Dooley, Mr. Patrick    male  32.0      0   

     Parch            Ticket     Fare Cabin Embarked  
0        0         A/5 21171   7.2500   NaN        S  
1        0          PC 17599  71.2833   C85        C  
2        0  STON/O2. 3101282   7.9250   NaN        S  
3        0            113803  53.1000  C123        S  
4        0            373450   8.0500   NaN        S  
..     ...               ...      ...   ...      ...  
886      0            211536  13.0000   NaN        S  
887      0            112053  30.0000   B42        S  
888      2        W./C. 6607  23.4500   NaN        S  
889      0            111369  30.0000  C148        C  
890      0            370376   7.7500   NaN        Q  

[891 rows x 12 columns]>
```

In [9]:
```
df.describe()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

3.Checking for null values

In [11]:
```
df.isnull().any()
```

Loading [MathJax]/extensions/Safe.js

Out[11]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [12]:

```
df.isnull().sum()
```

Out[12]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [13]:

```
df["Age"].fillna(df["Age"].mean(),inplace=True)
```

In [16]:

```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

In [18]:

```
df.drop(["Cabin"],axis=1,inplace=True)
```

In [19]:

```
df.isnull().sum()
```

Out[19]:

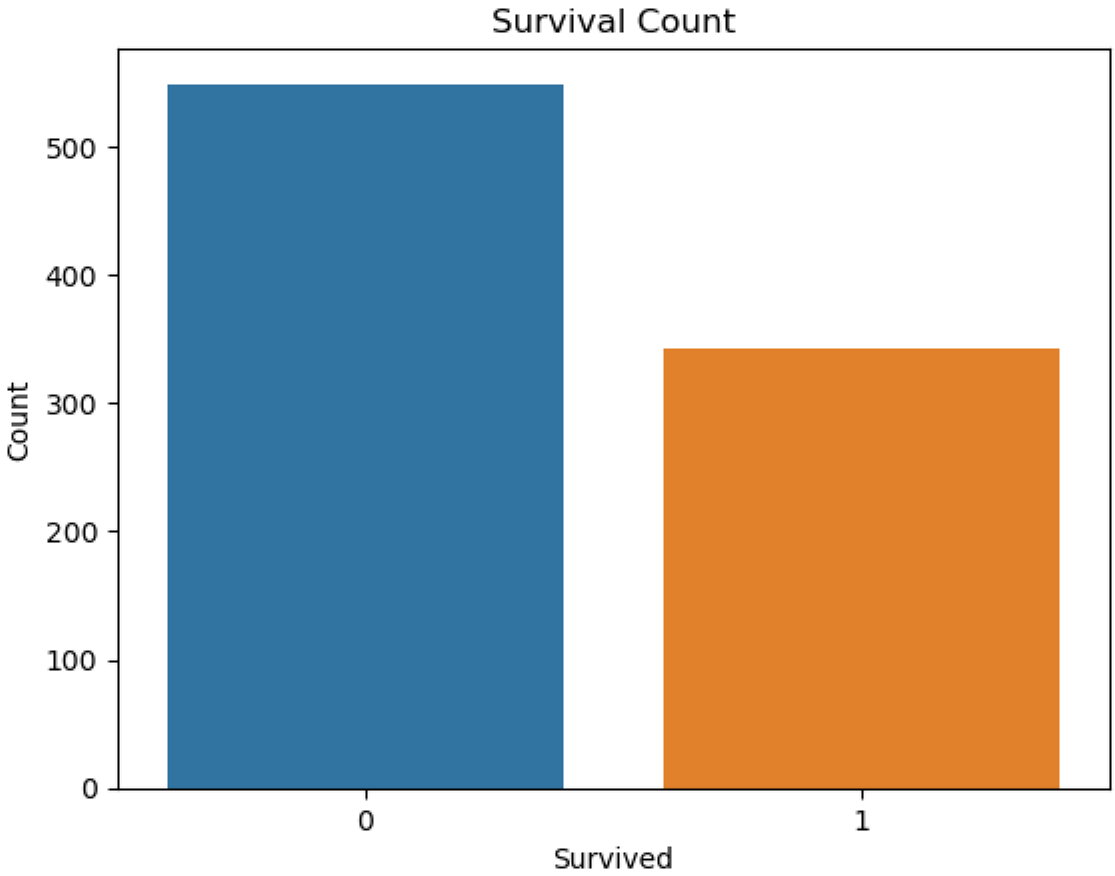
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

4.Data Visualization.

In [20]:

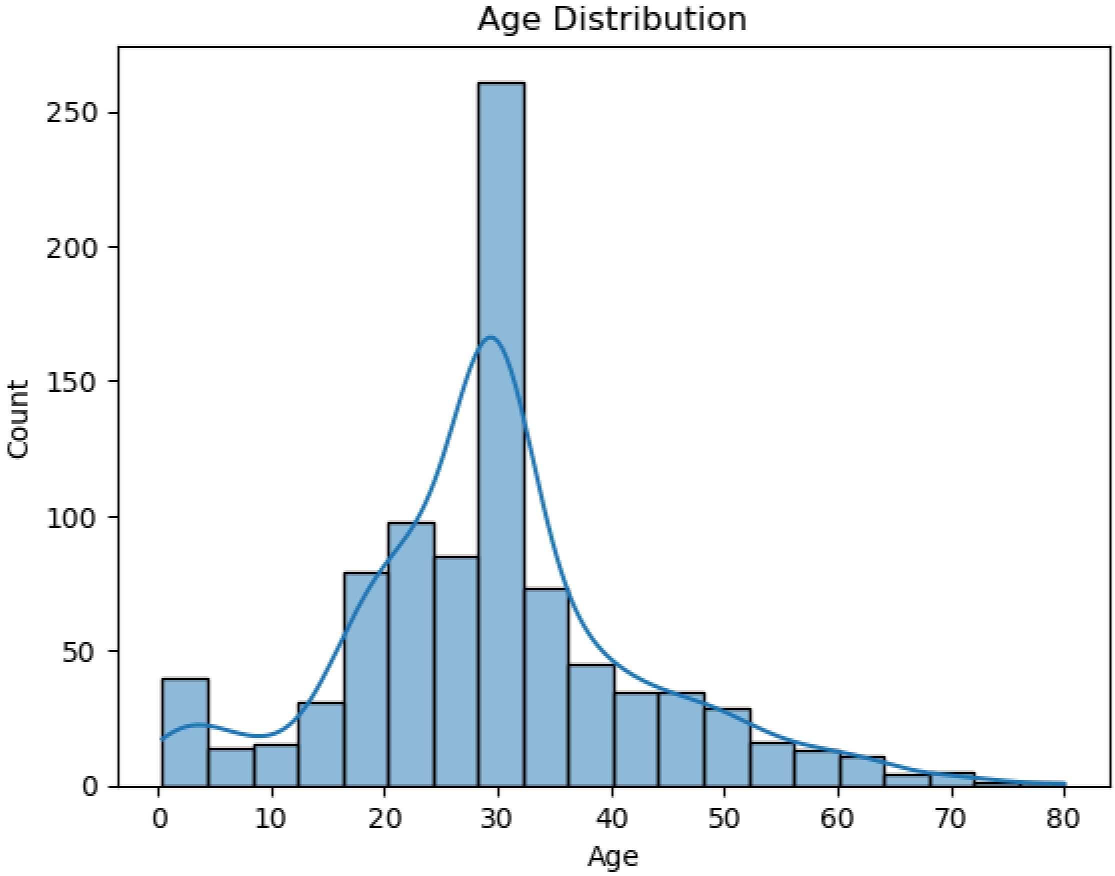
```
# Visualize the distribution of the 'Survived' column (0 = Not Survived, 1 = Survived)
sns.countplot(data=df, x='Survived')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```

Loading [MathJax]/extensions/Safe.js



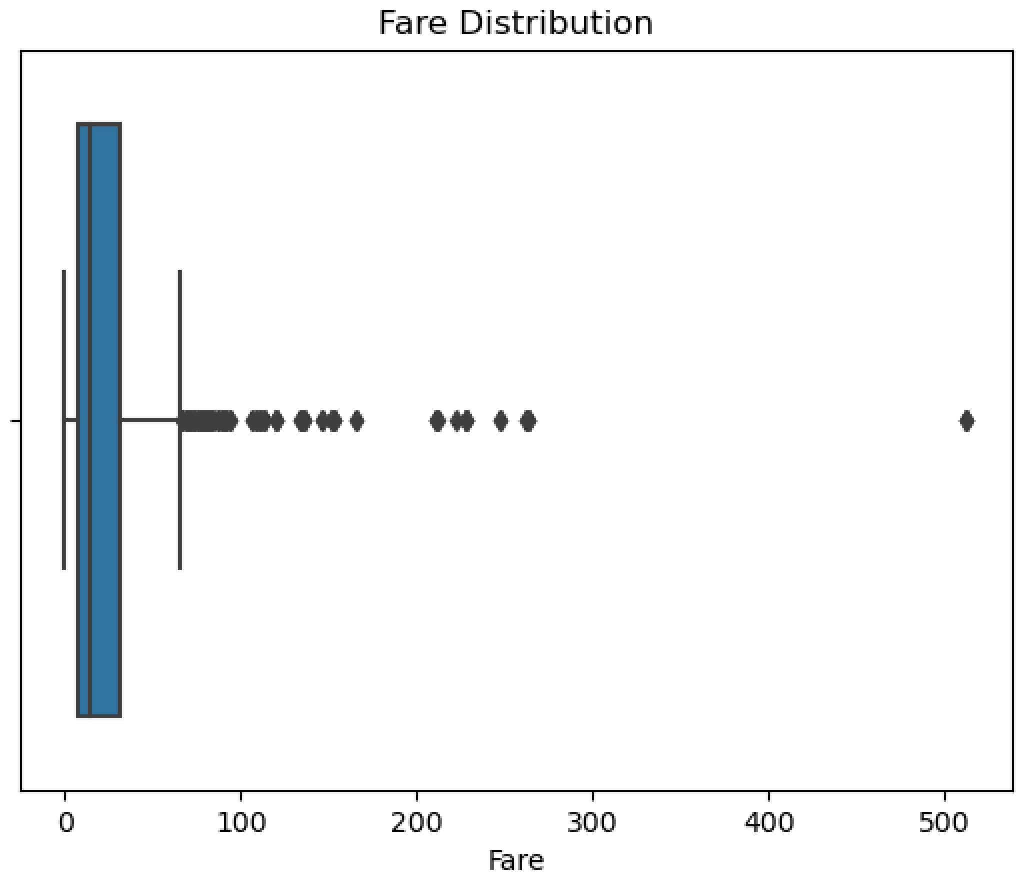
In [21]:

```
#Visualize the distribution of the 'Age' column
sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



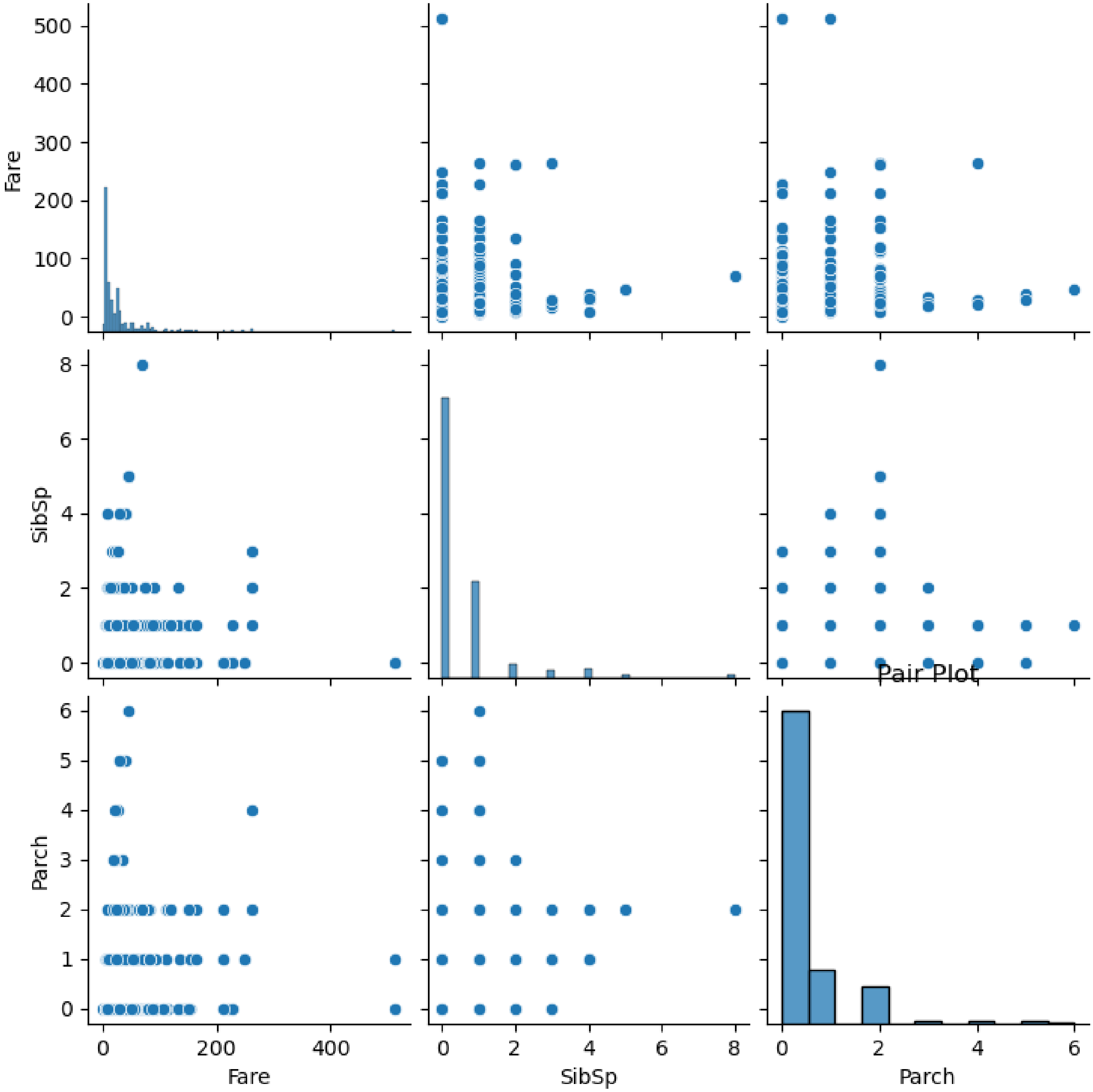
In [22]:

```
#Visualize the distribution of the 'Fare' column and detect outliers we will handle outl
sns.boxplot(data=df, x='Fare')
plt.title('Fare Distribution')
plt.xlabel('Fare')
plt.show()
```



In [23]:

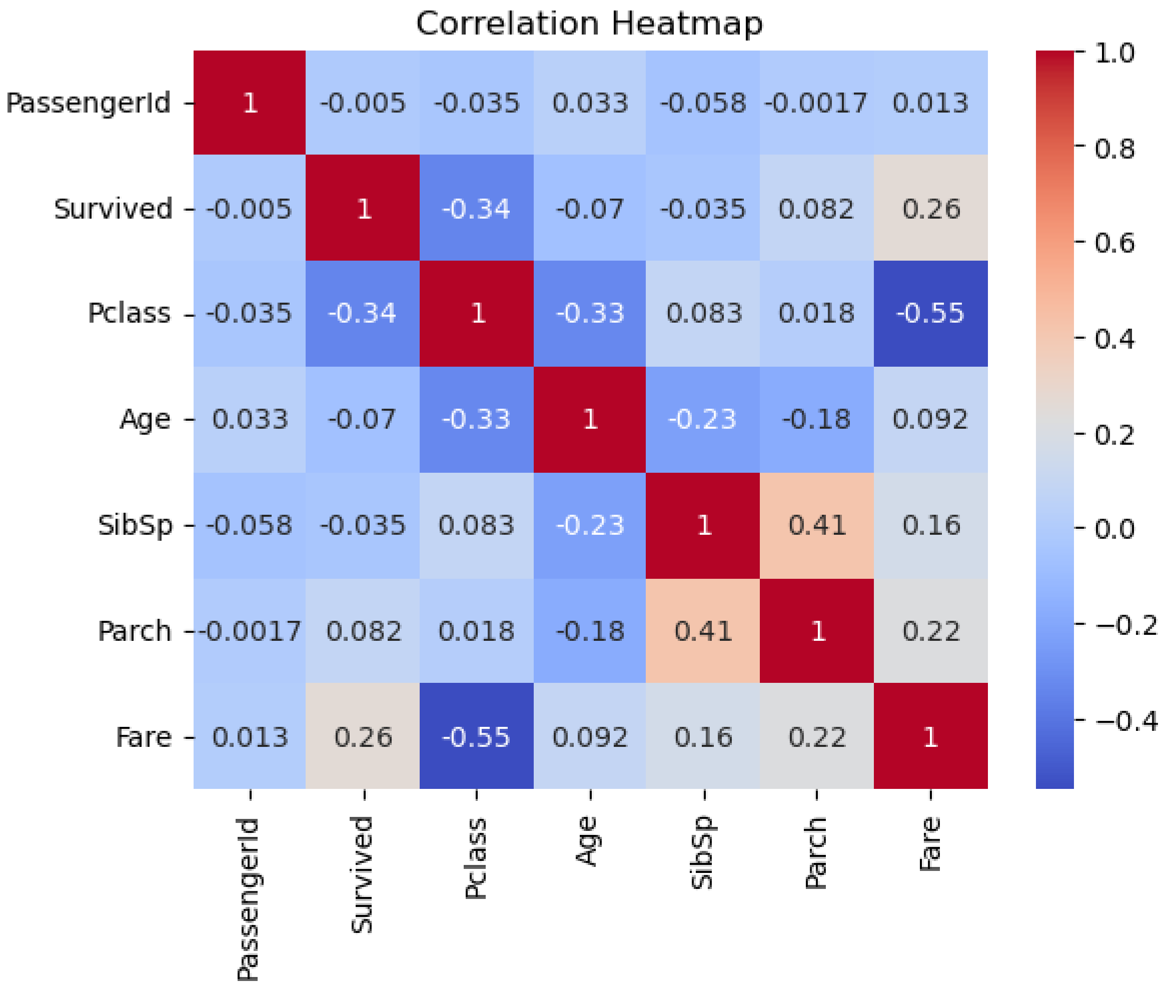
```
#Pair plot for selected numerical columns
sns.pairplot(data=df[['Fare', 'SibSp', 'Parch']])
plt.title('Pair Plot')
plt.show()
```



In [24]:

```
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True,cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
C:\Users\Praveen\AppData\Local\Temp\ipykernel_8032\554220597.py:1: FutureWarning: The de
fault value of numeric_only in DataFrame.corr is deprecated. In a future version, it wil
l default to False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  corr_matrix = df.corr()
```



5.Outlier Detection

In [25]:

```
z_scores = np.abs(stats.zscore(df['Age']))
max_threshold=3
outliers = df['Age'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
96     71.0
116    70.5
493    71.0
630    80.0
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

In [26]:

```
z_scores = np.abs(stats.zscore(df['Fare']))
max_threshold=3
outliers = df['Fare'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
27     263.0000
88     263.0000
118    247.5208
258    512.3292
299    247.5208
311    262.3750
341    263.0000
377    211.5000
380    227.5250
438    263.0000
527    221.7792
557    227.5250
679    512.3292
689    211.3375
700    227.5250
716    227.5250
730    211.3375
737    512.3292
742    262.3750
779    211.3375
Name: Fare, dtype: float64
```

In [27]:

```python
column_name = 'Fare'

# Calculate the first quartile (Q1) and third quartile (Q3)
Q1 = df[column_name].quantile(0.25)
Q3 = df[column_name].quantile(0.75)

# Calculate the IQR
IQR = Q3 - Q1

# Define the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter rows with values outside the IQR bounds
df_cleaned = df[(df[column_name] > lower_bound) & (df[column_name] <upper_bound)]

# Display the original and cleaned DataFrame sizes
print(f"Original DataFrame size: {df.shape}")
print(f"Cleaned DataFrame size: {df_cleaned.shape}")
df_cleaned
```

```
Original DataFrame size: (891, 11)
Cleaned DataFrame size: (775, 11)
```

Loading [MathJax]/extensions/Safe.js

Out[27]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	S
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	S
5	6	0	3	Moran, Mr. James	male	29.699118	0	0	330877	8.4583	Q
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	Q

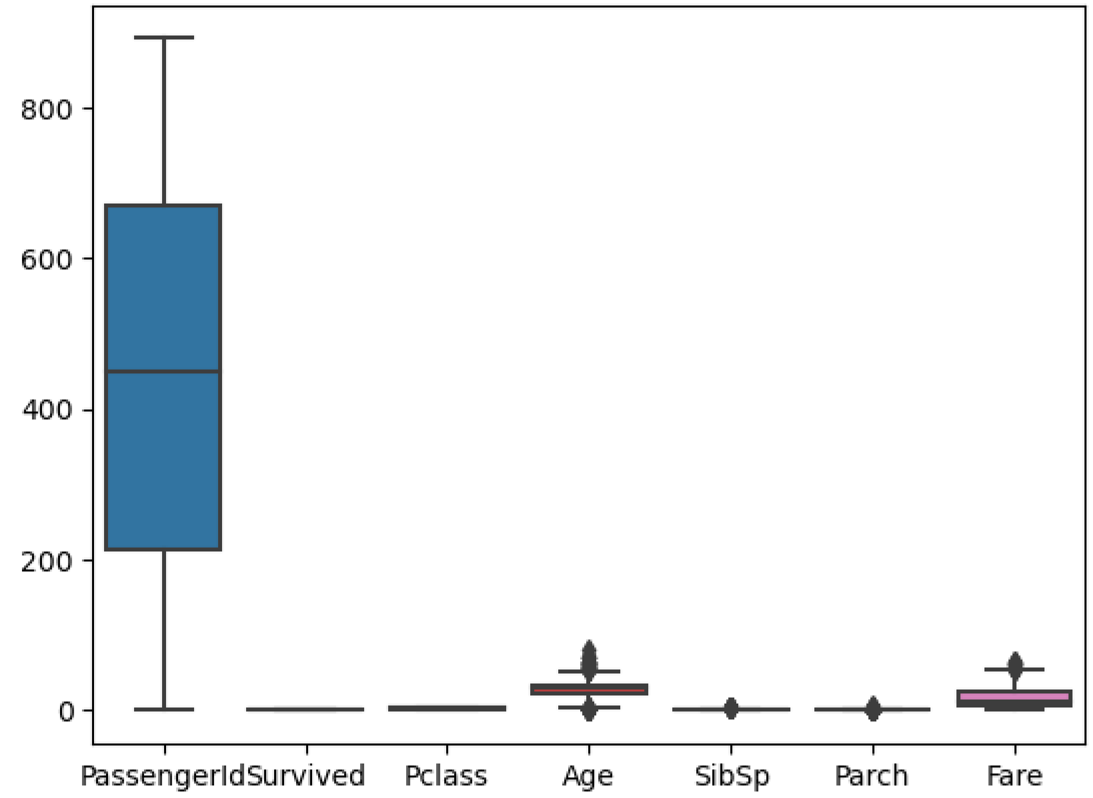
775 rows × 11 columns

In [28]:

```
sns.boxplot(df_cleaned)
```

Out[28]:

```
<Axes: >
```



In [29]:
```
df=df_cleaned
```

6.Splitting Dependent and Independent variables

In [34]:
```
x=df.iloc[:,[0,2,3,4,5,6,7,8]]
```

In [35]:
```
y=df.iloc[:,[1]]
```

In [36]:
```
x.head()
```

Out[36]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	S
2	3	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	S
4	5	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	S
5	6	3	Moran, Mr. James	male	29.699118	0	0	330877	8.4583	Q

In [38]:
```
x.shape
```

Out[38]:
```
(775, 10)
```

In [39]:
```
y.head()
```

Out[39]:

	Survived
0	0
2	1
3	1
4	0
5	0

In [40]:

```
y.shape
```

Out[40]:

```
(775, 1)
```

7.Perform Encoding

In [41]:

```
en = LabelEncoder() #using label encoding on sex
x['Sex'] = en.fit_transform(x['Sex'])
```

In [42]:

```
x.head()
```

Out[42]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	3	Braund, Mr. Owen Harris	1	22.000000	1	0	A/5 21171	7.2500	S
2	3	3	Heikkinen, Miss. Laina	0	26.000000	0	0	STON/O2. 3101282	7.9250	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.000000	1	0	113803	53.1000	S
4	5	3	Allen, Mr. William Henry	1	35.000000	0	0	373450	8.0500	S
5	6	3	Moran, Mr. James	1	29.699118	0	0	330877	8.4583	Q

In [43]:

```
#using one hot encoding on embarked
x = pd.get_dummies(x,columns=['Embarked'],drop_first=True)
```

In [44]:

```
x.head()
```

Out[44]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked_
0	1	3	Braund, Mr. Owen Harris	1	22.000000	1	0	A/5 21171	7.2500	0	
2	3	3	Heikkinen, Miss. Laina	0	26.000000	0	0	STON/O2. 3101282	7.9250	0	
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.000000	1	0	113803	53.1000	0	
4	5	3	Allen, Mr. William Henry	1	35.000000	0	0	373450	8.0500	0	
5	6	3	Moran, Mr. James	1	29.699118	0	0	330877	8.4583	1	

8. Feature Scaling

In [45]:

```
scale = StandardScaler()
x[['Age', 'Fare']] = scale.fit_transform(x[['Age', 'Fare']])
```

In [46]:

```
x.head()
```

Out[46]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embarked
0	1	3	Braund, Mr. Owen Harris	1	-0.556219	1	0	A/5 21171	-0.779117	0	
2	3	3	Heikkinen, Miss. Laina	0	-0.243027	0	0	STON/O2. 3101282	-0.729373	0	
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	0.461654	1	0	113803	2.599828	0	
4	5	3	Allen, Mr. William Henry	1	0.461654	0	0	373450	-0.720161	0	
5	6	3	Moran, Mr. James	1	0.046606	0	0	330877	-0.690071	1	

9. Splitting the data into Train and Test

In [47]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

In [48]:

```
x_train.shape
```

Out[48]:

```
(620, 11)
```

In [49]:

```
x_test.shape
```

Out[49]:
```
(155, 11)
```

In [50]:
```
y_train.shape
```

Out[50]:
```
(620, 1)
```

In [51]:
```
y_test.shape
```

Out[51]:
```
(155, 1)
```

In [52]:
```
x_train
```

Out[52]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Emb
881	882	3	Markun, Mr. Johann	1	0.305058	0	0	349257	-0.731524	0	
482	483	3	Rouse, Mr. Richard Henry	1	1.636122	0	0	A/5 3594	-0.720161	0	
131	132	3	Coelho, Mr. Domingos Fernandeo	1	-0.712814	0	0	SOTON/O.Q. 3101307	-0.793856	0	
283	284	3	Dorking, Mr. Edward Arthur	1	-0.791112	0	0	A/5. 10482	-0.720161	0	
173	174	3	Sivola, Mr. Antti Wilhelm	1	-0.634517	0	0	STON/O 2. 3101280	-0.729373	0	
...	...	...	...	...	...	...	...	...	...	...	
878	879	3	Laleff, Mr. Kristo	1	0.046606	0	0	349217	-0.731524	0	
211	212	2	Cameron, Miss. Clear Annie	0	0.461654	0	0	F.C.C. 13528	0.234198	0	
725	726	3	Oreskovic, Mr. Luka	1	-0.712814	0	0	315094	-0.675022	0	
643	644	3	Foo, Mr. Choong	1	0.046606	0	0	1601	2.850084	0	
790	791	3	Keane, Mr. Andrew "Andy"	1	0.046606	0	0	12460	-0.742269	1	

620 rows × 11 columns

In [53]:
```
y_train
```

Loading [MathJax]/extensions/Safe.js

```
Out[53]:
        Survived
 881           0
 482           0
 131           0
 283           1
 173           0
 ...         ...
 878           0
 211           1
 725           0
 643           1
 790           0

620 rows × 1 columns
```

```
In [54]: x_test
```

Out[54]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_Q	Embark
428	429	3	Flynn, Mr. James	1	0.046606	0	0	364851	-0.742269	1	
702	703	3	Barbara, Miss. Saiide	0	-0.869410	0	1	2691	-0.248199	0	
464	465	3	Maisner, Mr. Simon	1	0.046606	0	0	A/S 2816	-0.720161	0	
15	16	2	Hewlett, Mrs. (Mary D Kingcome)	0	2.027611	0	0	248706	-0.134280	0	
832	833	3	Saad, Mr. Amin	1	0.046606	0	0	2671	-0.780650	0	
...	...	...	...	...	...	...	...	...	...	...	
547	548	2	Padro y Manent, Mr. Julian	1	0.046606	0	0	SC/PARIS 2146	-0.291805	0	
560	561	3	Morrow, Mr. Thomas Rowan	1	0.046606	0	0	372622	-0.742269	1	
246	247	3	Lindahl, Miss. Agda Thorilda Viktoria	0	-0.321325	0	0	347071	-0.740427	0	
677	678	3	Turja, Miss. Anna Sofia	0	-0.869410	0	0	4138	-0.588120	0	
661	662	3	Badt, Mr. Mohamed	1	0.853143	0	0	2623	-0.780959	0	

155 rows × 11 columns

In [55]:

```
y_test
```

Loading [MathJax]/extensions/Safe.js

```
Out[55]:      Survived
         428         0
         702         0
         464         0
          15         1
         832         0
         ...       ...
         547         1
         560         0
         246         0
         677         1
         661         0

        155 rows × 1 columns
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js