In [2]:
```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Load Titanic dataset
df = pd.read_csv('Titanic.csv')  # Make sure titanic.csv is in the same folder

# Display the first 5 rows
df.head()
```

Out[2]:

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

In [3]:
```
df.tail()
```

Out[3]:

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

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

	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

In [7]:
```
df.shape
```

Out[7]:
```
(891, 12)
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
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 [10]:
```
df['Embarked'].value_counts()
```

Out[10]:
```
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

In [11]:
```
df['Pclass'].value_counts()
```

Out[11]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

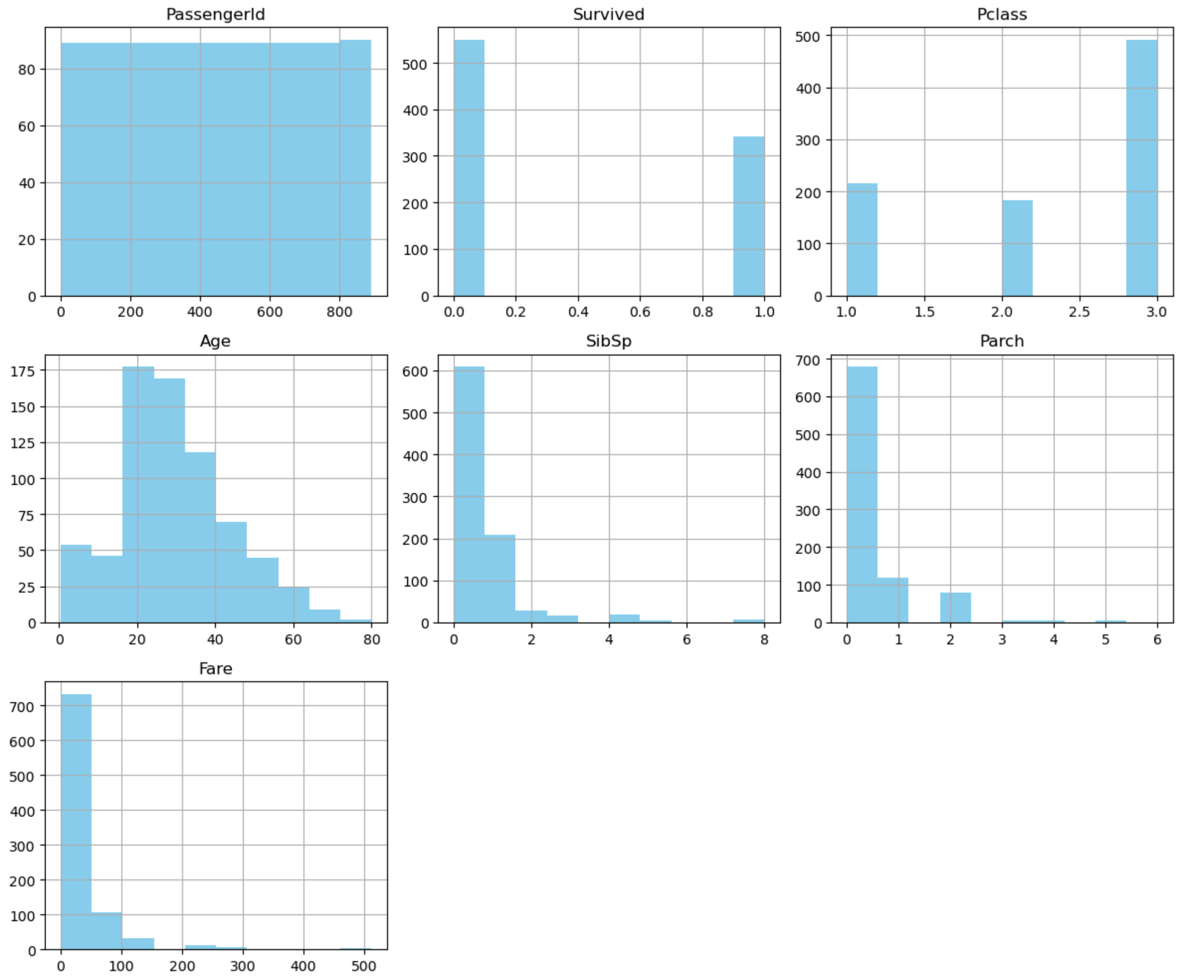
In [12]:
```
df['Sex'].value_counts()
```

Out[12]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

In [13]:
```
df['Survived'].value_counts()
```

Out[13]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

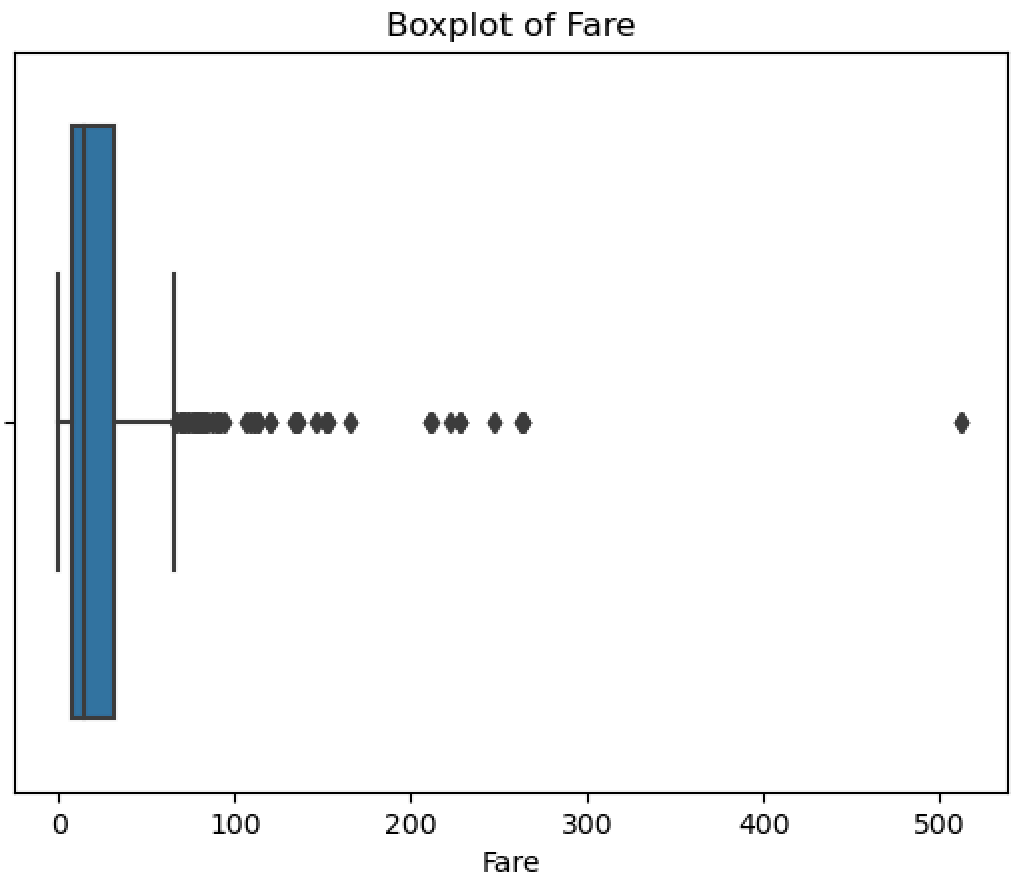
In [14]:
```
df.hist(figsize=(12, 10), color='skyblue')
plt.tight_layout()
plt.show()
```



📊 Histograms for Numerical Features

The histogram provides an overview of the distribution of numerical features such as Age, Fare, etc. It helps in identifying patterns like skewness, outliers, and the range of data.

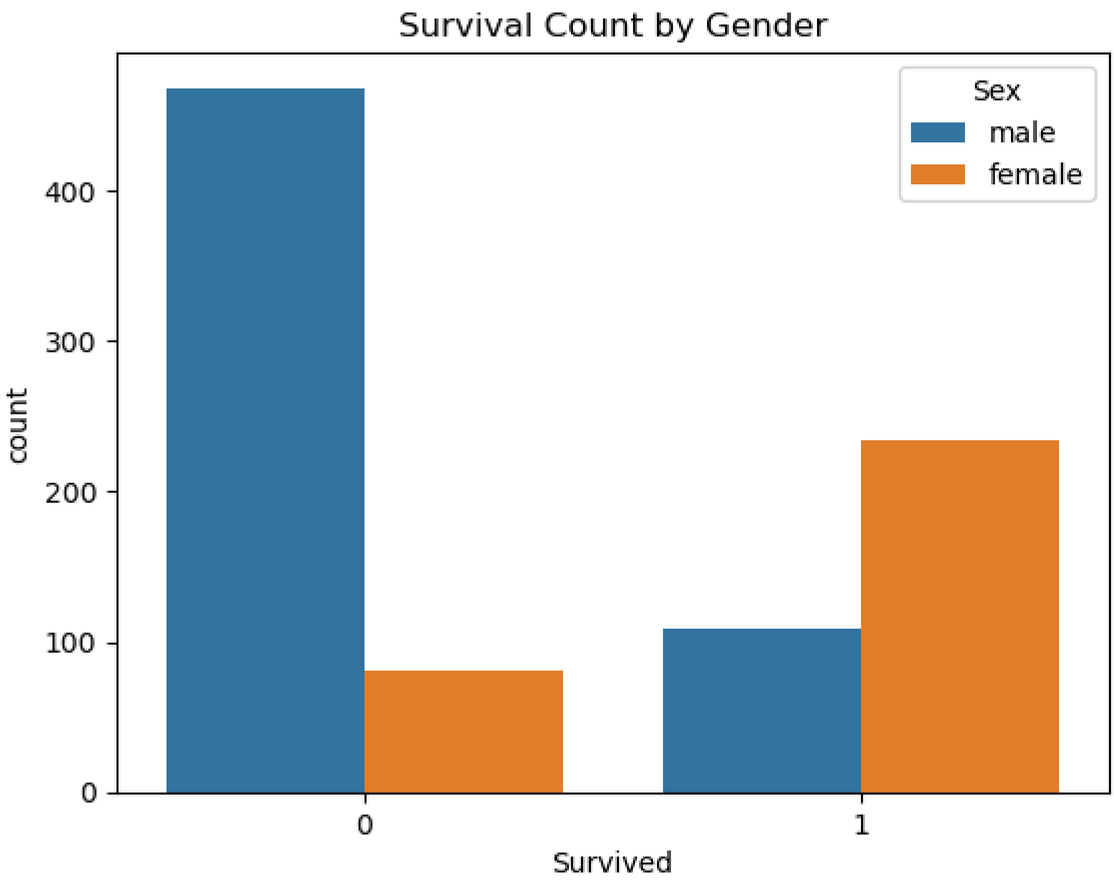
In [15]:
```
sns.boxplot(x=df['Fare'])
plt.title("Boxplot of Fare")
plt.show()
```



📦 Boxplots (To detect outliers)

The boxplot helps in detecting outliers in the 'Fare' feature. It displays the spread of the data and highlights values that are considered outliers, which can be further analyzed.

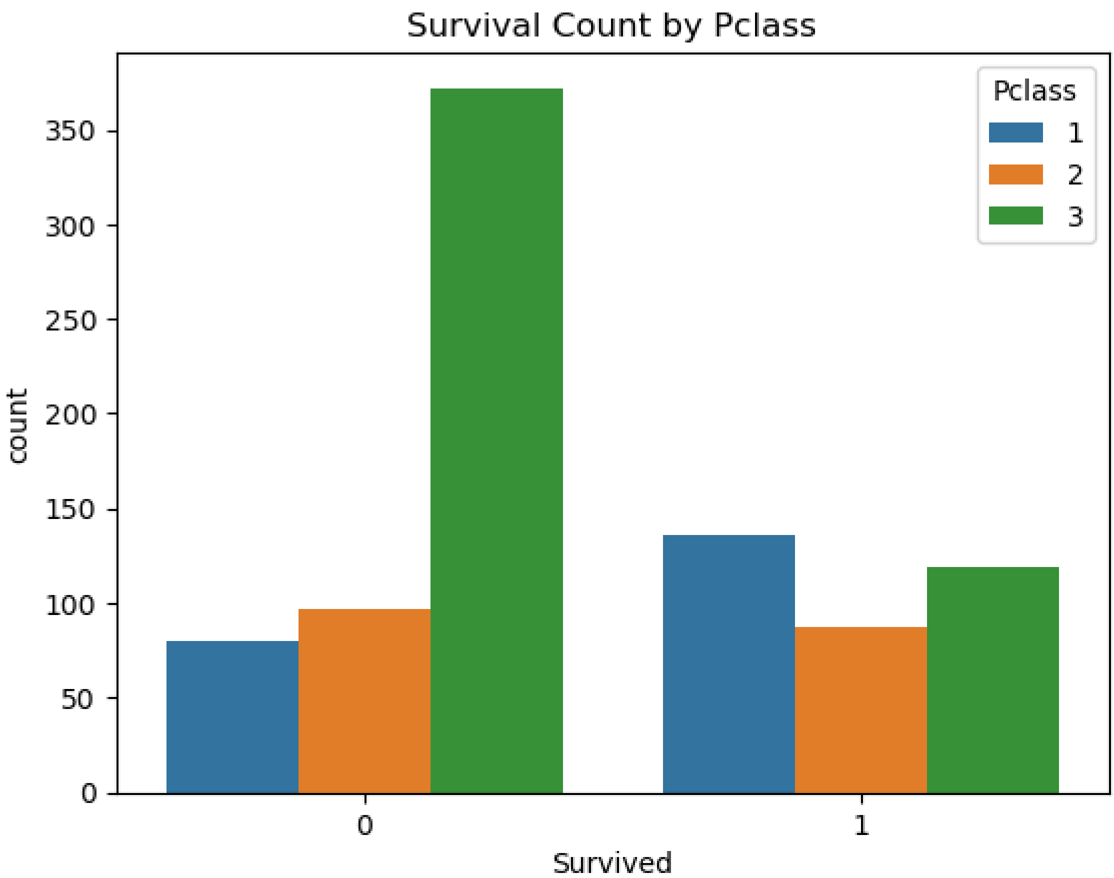
In [16]:
```
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title("Survival Count by Gender")
plt.show()
```



Survival by Sex

The countplot shows the survival count based on gender. This helps to analyze if gender played a role in the survival rates of passengers on the Titanic.

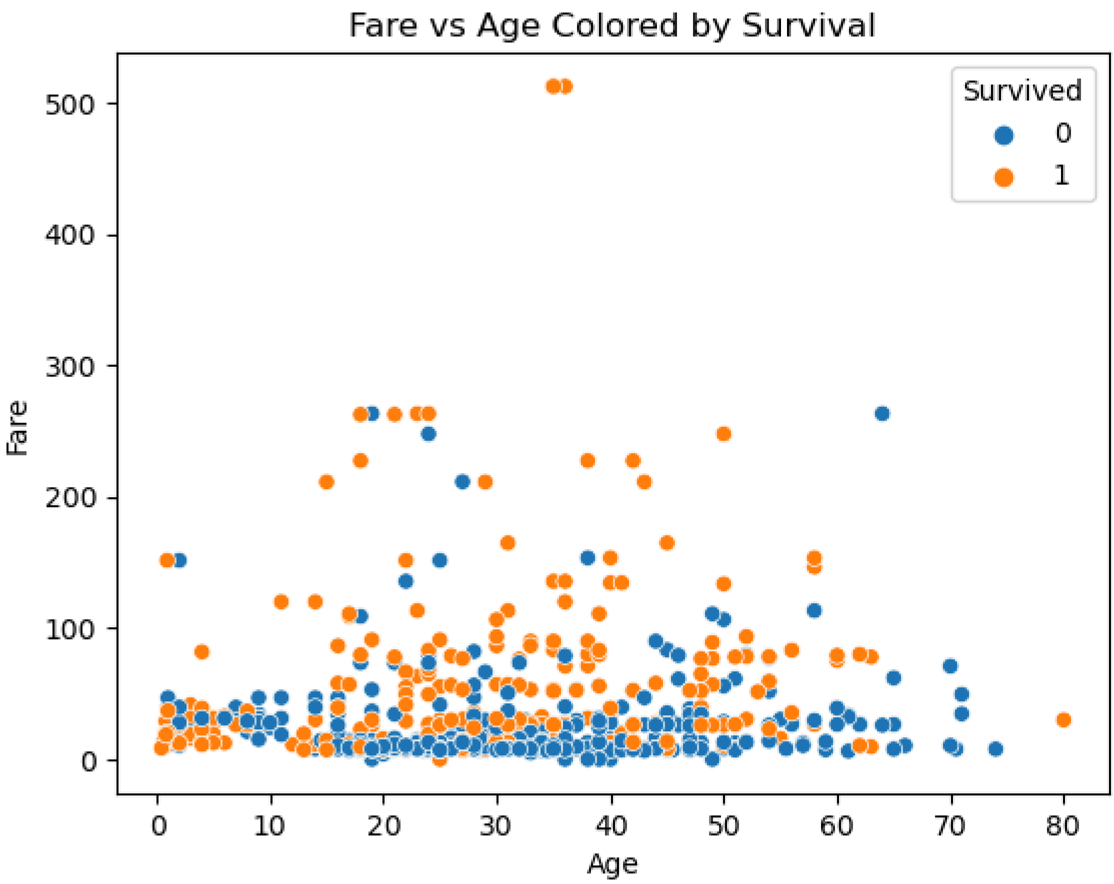
In [17]:
```
sns.countplot(x='Survived', hue='Pclass', data=df)
plt.title("Survival Count by Pclass")
plt.show()
```



🎓 Survival by Passenger Class

The countplot presents the survival count based on passenger class (Pclass). It illustrates how different classes influenced survival chances.

In [18]:
```
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title("Fare vs Age Colored by Survival")
plt.show()
```

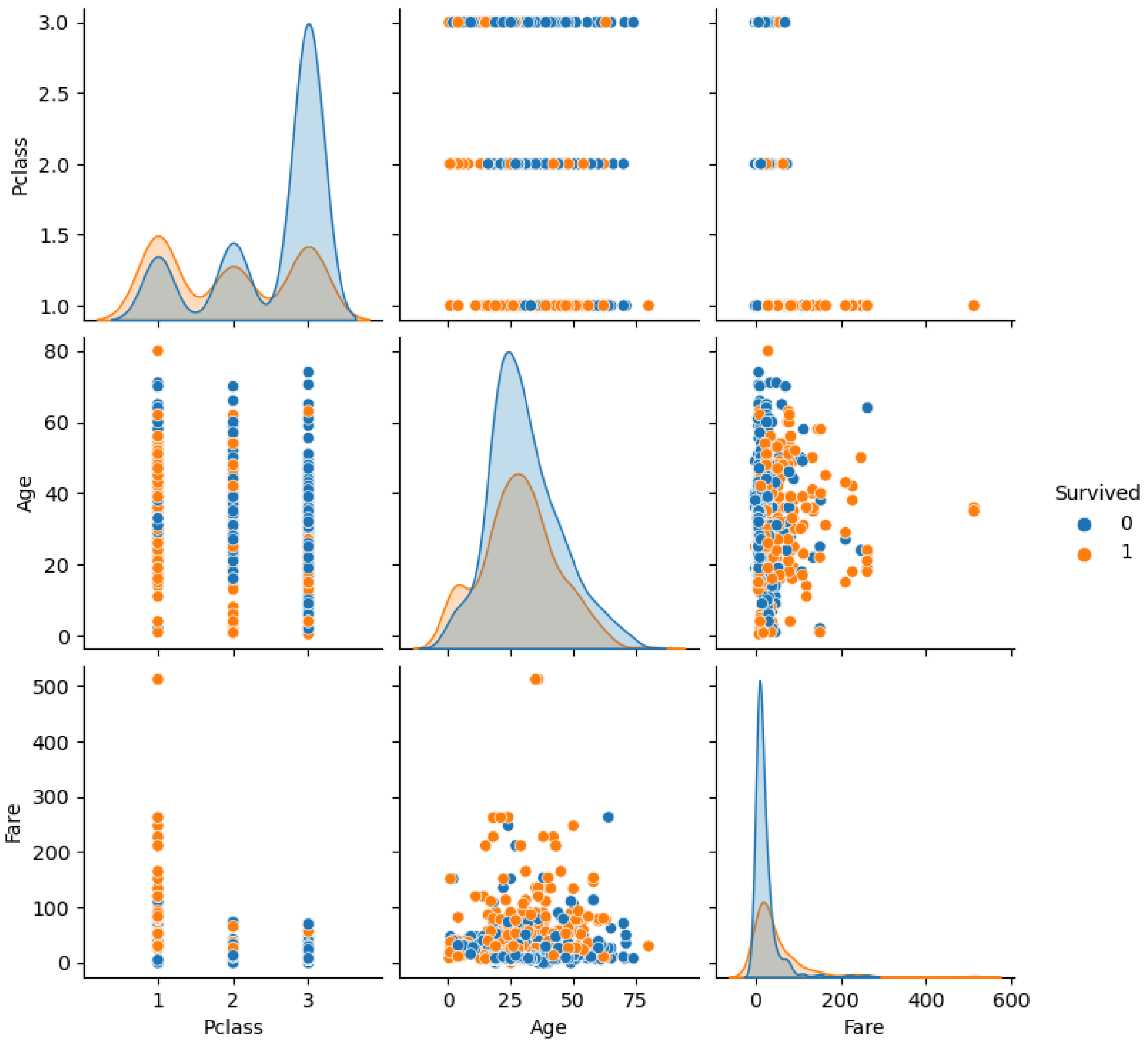


📈 Fare vs Age

The scatter plot demonstrates the relationship between the Age and Fare features, with points colored by survival status. This helps identify patterns between these variables and survival chances.

In [19]:
```
sns.pairplot(df[['Survived', 'Pclass', 'Age', 'Fare']], hue='Survived')
plt.show()
```

```
C:\Users\saura\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarn
ing: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

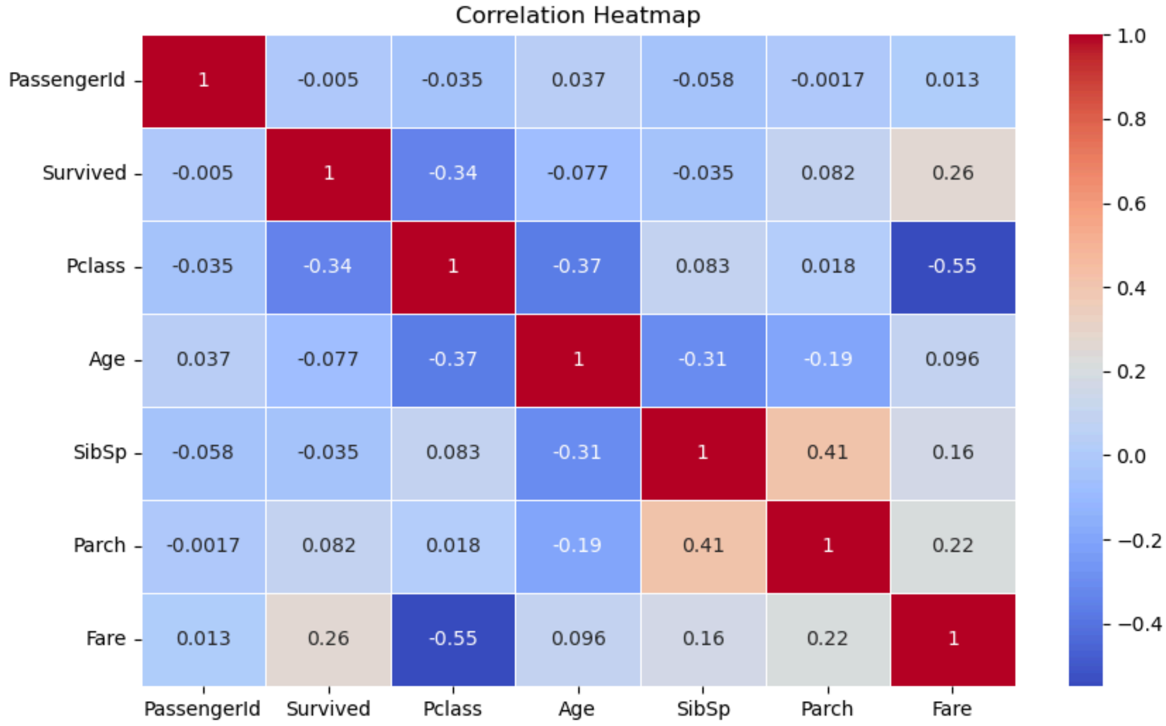


🎓 Pairplot (Relationships Across All Variables)

The pairplot displays the relationships across multiple features such as Survived, Pclass, Age, and Fare. It is useful for visualizing interactions between pairs of variables.

In [21]:
```
numeric_df = df.select_dtypes(include=['int64', 'float64'])

plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



🧭 Heatmap of Correlation Matrix

The heatmap shows the correlation between various numerical features in the dataset. Strong correlations (positive or negative) between features can be identified, which can guide further analysis or feature engineering.

🔍 Summary of Findings:

- Survival rate was higher for females compared to males.
- Passengers in 1st class had better chances of survival.
- Younger children had slightly higher survival.
- Fare prices had a wide range, with a few very high values (outliers).
- Missing values are present in 'Age' and 'Cabin'.

In [ ]: