In [8]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

In [12]:
```
df=pd.read_csv("train.csv")
```

In [14]:
```
df=df.drop(["PassengerId","Name","Ticket","Cabin"],axis=1)
```

In [18]:
```
df["Age"] = df["Age"].fillna(df["Age"].median())
df["Embarked"] = df["Embarked"].fillna(df["Embarked"].mode()[0])
```

In [20]:
```
df["Sex"] = df["Sex"].map({"male": 0,"female": 1})
df = pd.get_dummies(df,columns=["Embarked"])
```

In [22]:
```
X = df.drop("Survived", axis=1)
y = df["Survived"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [24]:
```
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Out[24]:

▾ RandomForestClassifier

```
RandomForestClassifier()
```

In [28]:
```
y_pred = model.predict(X_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.87      0.85       105
           1       0.80      0.76      0.78        74

    accuracy                           0.82       179
   macro avg       0.82      0.81      0.81       179
weighted avg       0.82      0.82      0.82       179
```

In [30]:
```
df.info()
df.describe()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Survived    891 non-null    int64
 1   Pclass      891 non-null    int64
 2   Sex         891 non-null    int64
 3   Age         891 non-null    float64
 4   SibSp       891 non-null    int64
 5   Parch       891 non-null    int64
 6   Fare        891 non-null    float64
 7   Embarked_C  891 non-null    bool
 8   Embarked_Q  891 non-null    bool
 9   Embarked_S  891 non-null    bool
dtypes: bool(3), float64(2), int64(5)
memory usage: 51.5 KB
```

Out[30]:

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          0
Embarked_C    0
Embarked_Q    0
Embarked_S    0
dtype: int64
```

In [32]:
```
df['Survived'].value_counts()
df['Sex'].value_counts()
df['Pclass'].value_counts()
```

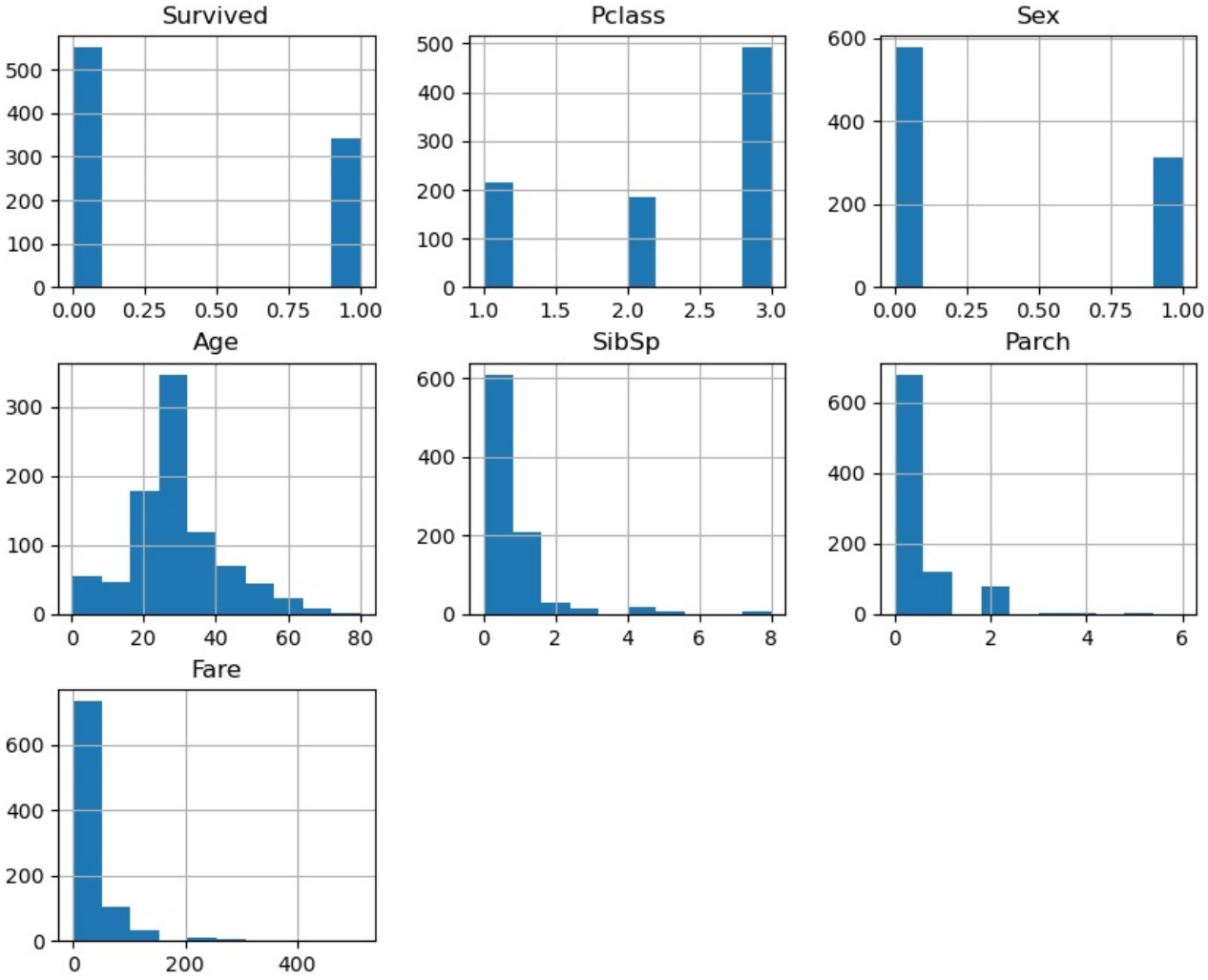
Out[32]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

In [44]:
```
import seaborn as sns
```

In [50]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [36]:
```
df.hist(figsize=(10,8))
```

Out[36]:
```
array([[<Axes: title={'center': 'Survived'}>,
        <Axes: title={'center': 'Pclass'}>,
        <Axes: title={'center': 'Sex'}>],
       [<Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'SibSp'}>,
        <Axes: title={'center': 'Parch'}>],
       [<Axes: title={'center': 'Fare'}>, <Axes: >, <Axes: >]],
      dtype=object)
```



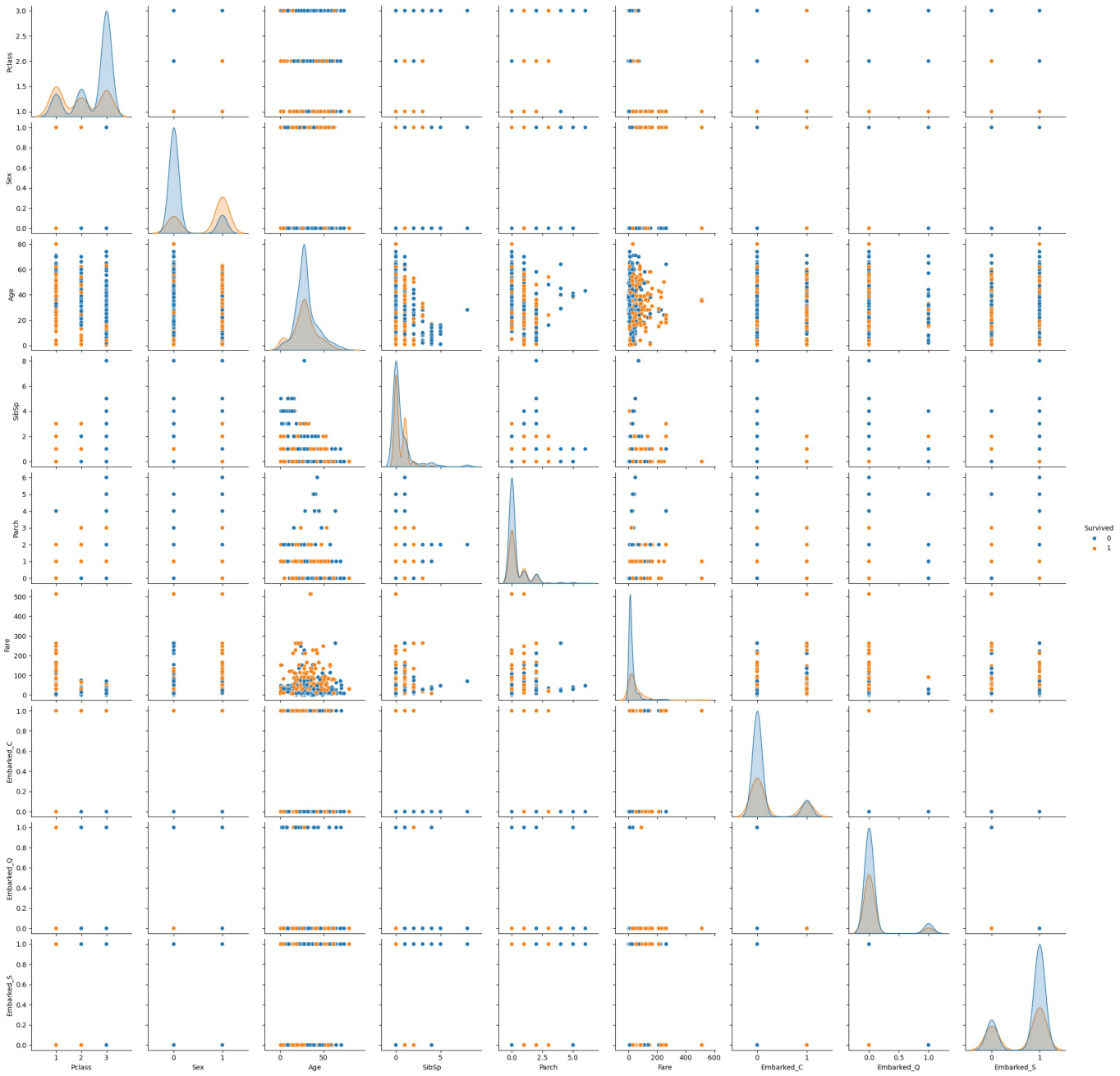

## Feature Distribution Overview

The histograms below show the distribution of key features in the Titanic dataset:

- *Survived*: More passengers did not survive.
- *Pclass*: Most were in 3rd class.
- *Sex*: More males than females.
- *Age*: Majority aged between 20–40.
- *SibSp*: Most had 0 or 1 sibling/spouse.
- *Parch*: Most had no parents/children aboard.
- *Fare*: Most passengers paid a lower fare.

In [54]:
```
sns.pairplot(df, hue='Survived')
```

Out[54]:
```
<seaborn.axisgrid.PairGrid at 0x2aa017ffc80>
```




## Pair Plot (by Survival)

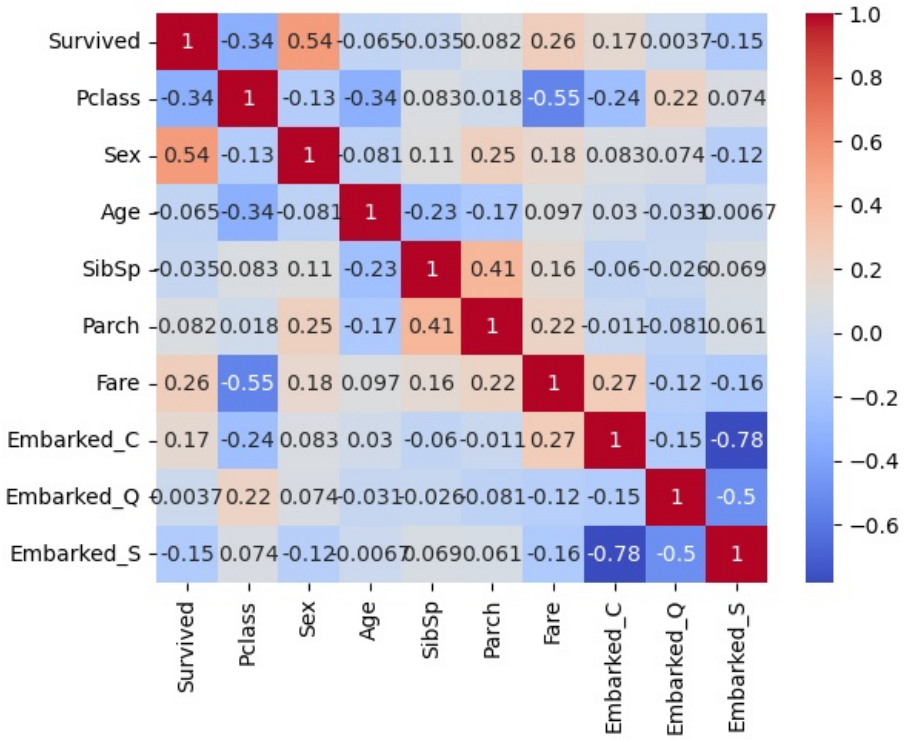
The pair plot below shows relationships between features colored by survival:

- *Clear separation* seen in features like Sex, Pclass, and Fare.
- Survivors (orange) tend to cluster at lower Pclass and higher Fare.
- *Age* and SibSp also show some visible differences by survival.

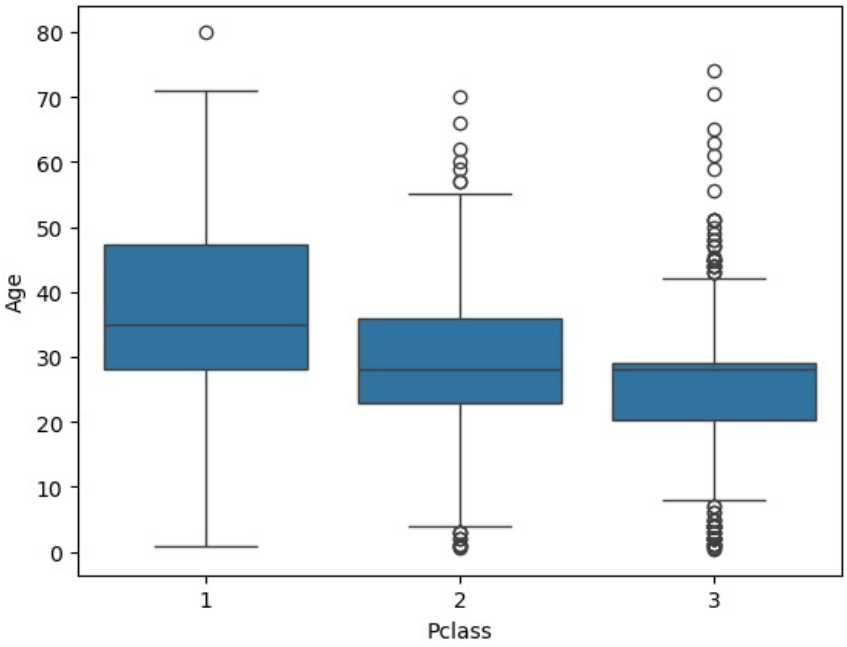
Useful for spotting feature interactions and patterns linked to survival.

In [56]:
```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

Out[56]: <Axes: >




In [52]:
```
sns.boxplot(x='Pclass', y='Age', data=df)
plt.show()
```

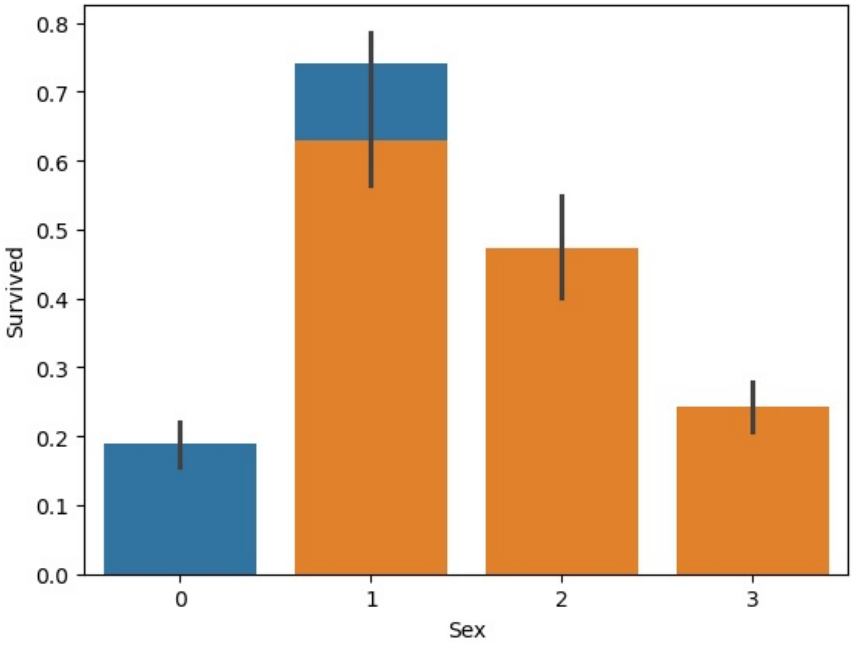



In [ ]:
```
### Observation: Age vs Pclass

This boxplot shows that passengers in 1st class (Pclass=1) tend to be older than those in 3rd class. There's als
```

In [71]:
```
sns.barplot(x='Sex', y='Survived',data=df)
sns.barplot(x='Pclass', y='Survived',data=df)
```

Out[71]:
```
<Axes: xlabel='Sex', ylabel='Survived'>
```




## Bar Plot of Survival Rate by Sex and Pclass

The bar plots below visualize the average survival rate based on two features from the Titanic dataset:

1. *Sex* (0 = Male, 1 = Female)
2. *Pclass* (1, 2, 3 = Passenger Classes)

*Observations*:

- Females had a higher survival rate than males.
- Passengers in *1st class* had the highest chance of survival.
- Survival probability decreased from *1st* to *3rd* class.

This analysis shows that both *gender* and *passenger class* were important factors affecting survival.

## Final Summary of Findings

- Females had a higher survival rate compared to males.
- Passengers in 1st class had better chances of survival than those in 2nd or 3rd class.
- Younger passengers were more likely to survive.
- Some variables showed skewed distributions (e.g., Age).
- There were visible patterns and correlations between Sex, Pclass, and Survival.
- No major multicollinearity issues were observed in the features.

These insights help us understand key factors that affected survival on the Titanic.

In [ ]: