In [1]:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, roc_curve, auc
```

In [2]:

```python
# Load dataset
df = pd.read_csv("/kaggle/input/titanic-survival-dataset/Titanic-Dataset.csv")
```

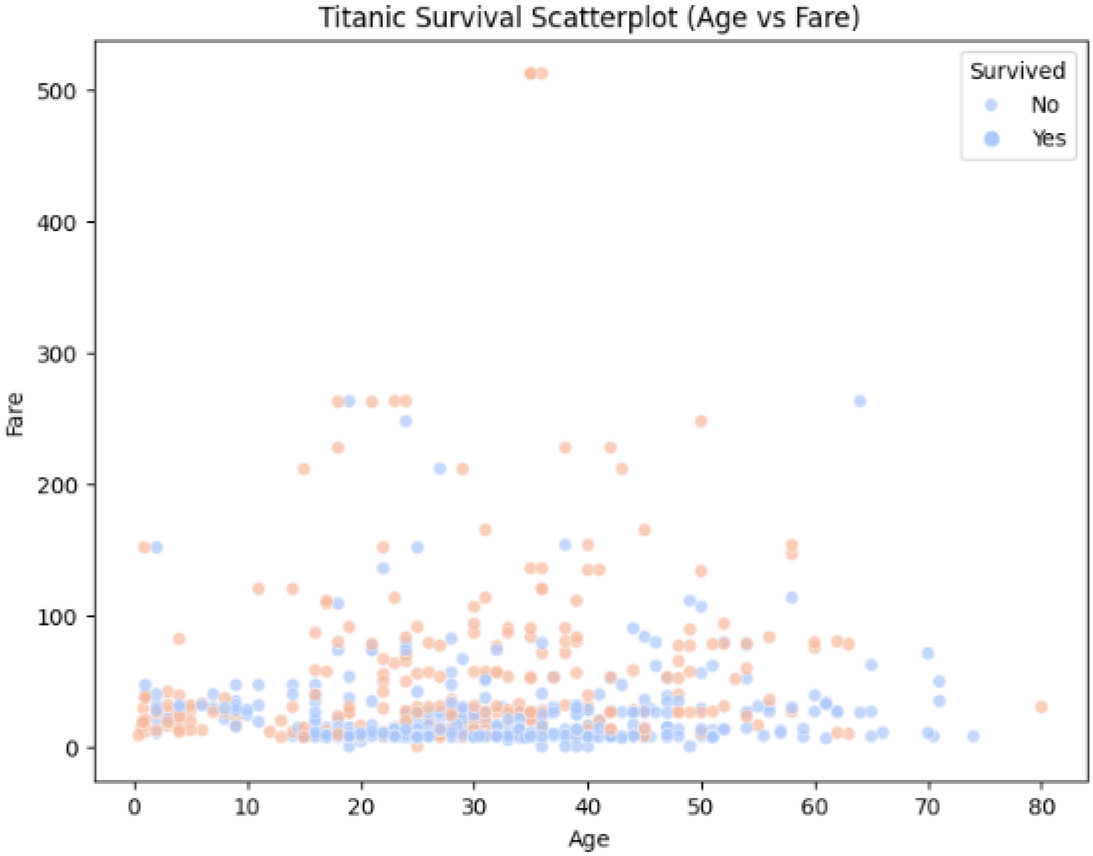
In [3]:

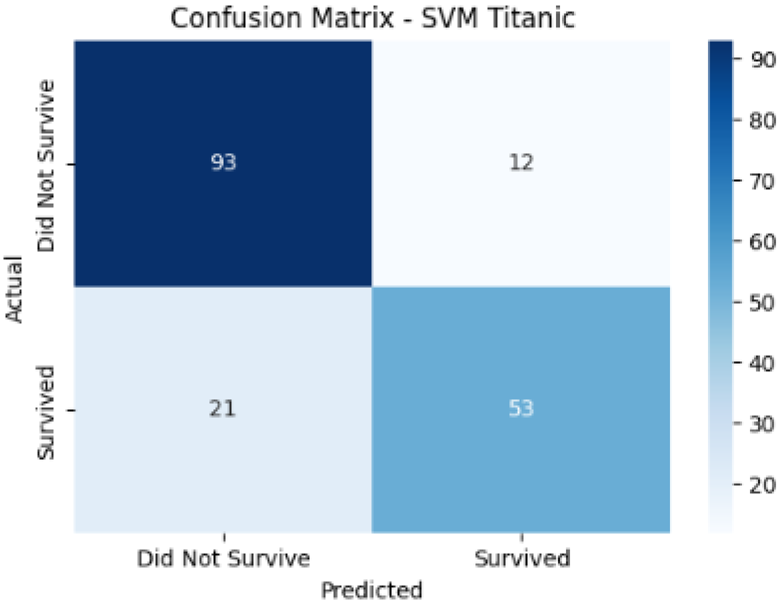
```python
# Select columns for prediction
features = ["Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked"]
target = "Survived"

X = df[features].copy()   # copy to avoid warnings
y = df[target]

# Handle missing values
X["Age"] = X["Age"].fillna(X["Age"].median())
X["Embarked"] = X["Embarked"].fillna(X["Embarked"].mode()[0])

# Encode categorical variables
le = LabelEncoder()
X["Sex"] = le.fit_transform(X["Sex"])
X["Embarked"] = le.fit_transform(X["Embarked"])

# Scale numerical features (convert to array here)
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build SVM classifier
svm_model = SVC(kernel="rbf", random_state=42)
svm_model.fit(X_train, y_train)

# Predictions
y_pred = svm_model.predict(X_test)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("SVM Model Accuracy:", accuracy)
```

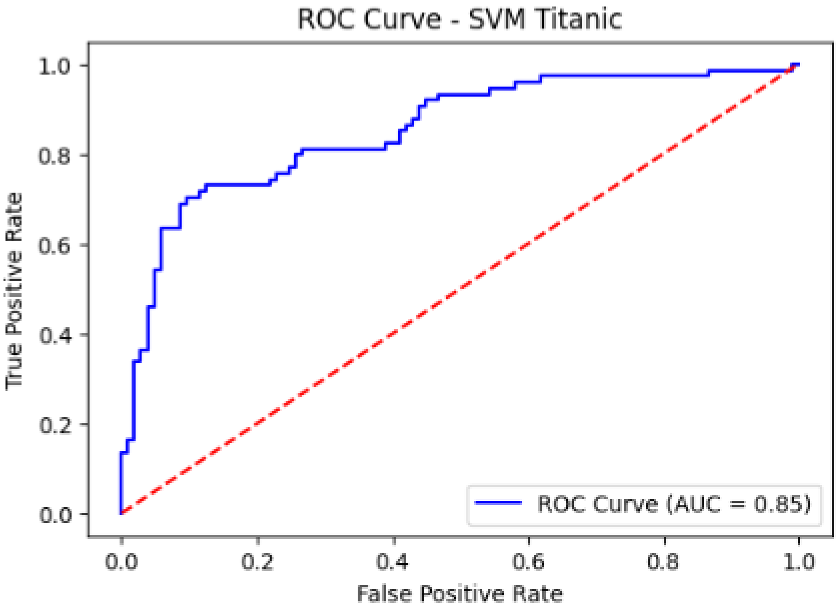
```
SVM Model Accuracy: 0.8156424581005587
```

In [4]:

```python
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x="Age", y="Fare", hue="Survived", palette="coolwarm", alpha=0.7)
plt.title("Titanic Survival Scatterplot (Age vs Fare)")
plt.xlabel("Age")
plt.ylabel("Fare")
plt.legend(title="Survived", labels=["No", "Yes"])
plt.show()

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=["Did Not Survive", "Survived"],
            yticklabels=["Did Not Survive", "Survived"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix - SVM Titanic")
plt.show()

# ROC Curve
y_scores = svm_model.decision_function(X_test)  # decision function for probabilities
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, color='blue', label=f"ROC Curve (AUC = {roc_auc:.2f})")
plt.plot([0,1], [0,1], color='red', linestyle="--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve - SVM Titanic")
plt.legend(loc="lower right")
plt.show()
```


Titanic Survival Scatterplot (Age vs Fare)
Survived
No
Yes
500
400
300
200
100
0
Fare
0
10
20
30
40
50
60
70
80
Age


Confusion Matrix - SVM Titanic
Did Not Survive
Survived
Actual
93
12
21
53
90
80
70
60
50
40
30
20
Did Not Survive
Survived
Predicted


ROC Curve - SVM Titanic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
ROC Curve (AUC = 0.85)