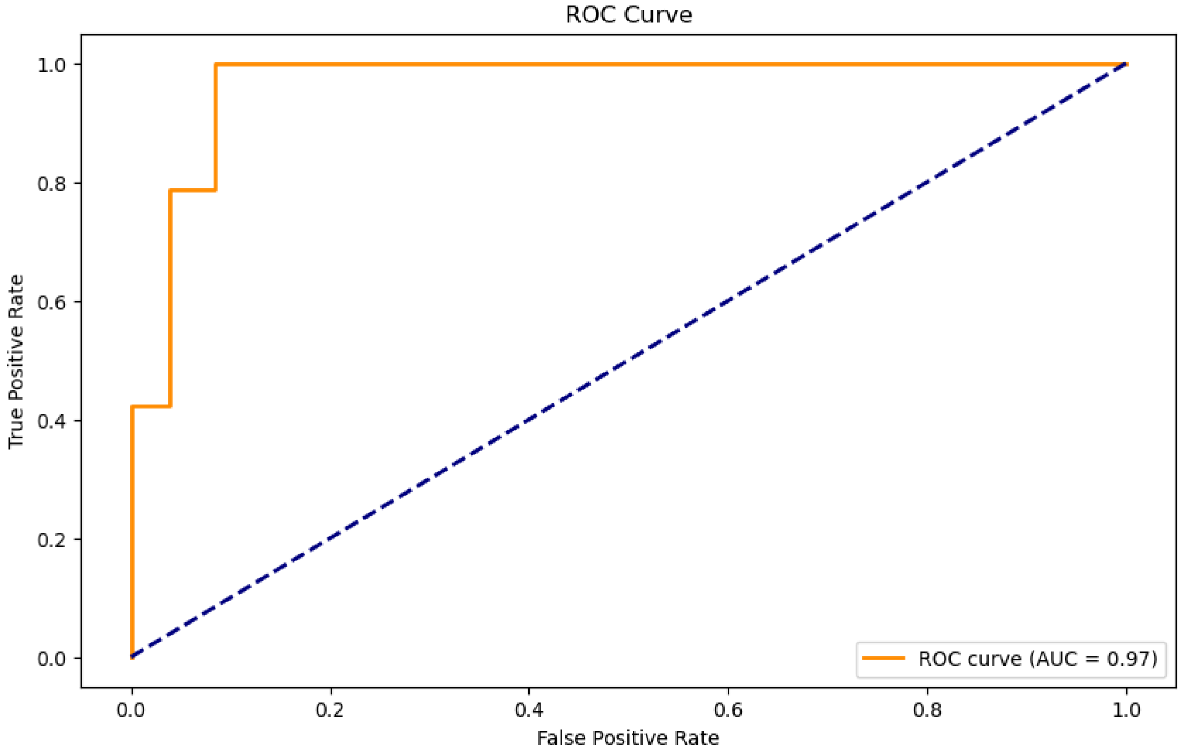
In [11]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import roc_curve, auc, classification_report, confusion
import joblib

# Load the saved logistic regression model
model = joblib.load('logreg_model.pkl')  # Load pre-trained model

# Load the new dataset
df = pd.read_csv("Healthcare_Support_Occupations.csv")
print(df.head())  # Inspect the dataset structure

# Convert 'Automatability' into binary labels (1 if >= 0.5, else 0)
df['Automatibility_Label'] = (df['Automatability'] >= 0.5).astype(int)
df.drop(columns=['Automatability'], inplace=True)  # Drop original column

# Encode 'Task Type' (1 -> 0 for Core, 2 -> 1 for Supplemental)
df['Task Type'] = df['Task Type'] - 1

# Convert 'Scale Name' to numeric values using LabelEncoder
label_encoder = LabelEncoder()
df['Scale Name'] = label_encoder.fit_transform(df['Scale Name'])

# Drop non-relevant columns
columns_to_drop = ["O*NET-SOC Code", "Task ID", "Task_x", "Title", "Category
df.drop(columns=columns_to_drop, errors='ignore', inplace=True)

# Handle missing values by replacing them with column means
df.fillna(df.mean(), inplace=True)

# Split data into features (X) and target (y)
X = df.drop(columns=["Automatibility_Label"])
y = df["Automatibility_Label"]

# Train-test split with stratification to maintain class balance
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Scale features for consistency with training data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Use the trained model for predictions
y_pred = model.predict(X_test_scaled)
y_prob = model.predict_proba(X_test_scaled)[:, 1]  # Get probabilities for 

# Evaluate the model with classification metrics
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Calculate and plot ROC curve and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(10, 6))
```

```
plt.plot(fpr, tpr, color="darkorange", lw=2, label=f"ROC curve (AUC = {roc_a
plt.plot([0, 1], [0, 1], color="navy", lw=2, linestyle="--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend(loc="lower right")
plt.show()

print(f"AUC Score: {roc_auc:.4f}")
```

```
  O*NET-SOC Code  Task ID                                             Task_
x  \
0     31-1121.00     4240  Maintain records of patient care, condition,
p...
1     31-1121.00     4240  Maintain records of patient care, condition,
p...
2     31-1121.00     4240  Maintain records of patient care, condition,
p...
3     31-1121.00     4240  Maintain records of patient care, condition,
p...
4     31-1121.00     4240  Maintain records of patient care, condition,
p...

   Scale Name  Data Value  Task_y  Task Type  Importance_x   Level_x  \
0           1        0.00    8138          1      3.079512  2.972683
1           1        0.08    8138          1      3.079512  2.972683
2           1        2.50    8138          1      3.079512  2.972683
3           1       12.18    8138          1      3.079512  2.972683
4           1       66.48    8138          1      3.079512  2.972683

   Importance_y  ...  Importance   Level  Tech Readiness  Tool Dependency
\
0      2.370571  ...    2.509808  2.2675        0.666667             32.0
1      2.370571  ...    2.509808  2.2675        0.666667             32.0
2      2.370571  ...    2.509808  2.2675        0.666667             32.0
3      2.370571  ...    2.509808  2.2675        0.666667             32.0
4      2.370571  ...    2.509808  2.2675        0.666667             32.0

    Context  Context (Categories 1-3)  Context (Categories 1-5)  \
0  2.559298                 33.333333                 19.999964
1  2.559298                 33.333333                 19.999964
2  2.559298                 33.333333                 19.999964
3  2.559298                 33.333333                 19.999964
4  2.559298                 33.333333                 19.999964

              Title Automatability                        Category
0  Home Health Aides       0.570689  Healthcare Support Occupations
1  Home Health Aides       0.570689  Healthcare Support Occupations
2  Home Health Aides       0.570689  Healthcare Support Occupations
3  Home Health Aides       0.570689  Healthcare Support Occupations
4  Home Health Aides       0.570689  Healthcare Support Occupations

[5 rows x 21 columns]

Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.96      0.85       382
           1       0.89      0.51      0.65       236

    accuracy                           0.79       618
   macro avg       0.83      0.74      0.75       618
weighted avg       0.81      0.79      0.77       618


Confusion Matrix:
[[367  15]
 [115 121]]
```




```
AUC Score: 0.9679
```