In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    confusion_matrix,
    classification_report,
    roc_curve,
    auc
)
```

In [2]:
```
# Step 2: Load Dataset
# Dataset: Social Network Ads (download from Kaggle)
# https://www.kaggle.com/datasets/rakeshrau/social-network-ads
data = pd.read_csv("Social_Network_Ads.csv")

print("Dataset Loaded Successfully")
print(data.head())
```

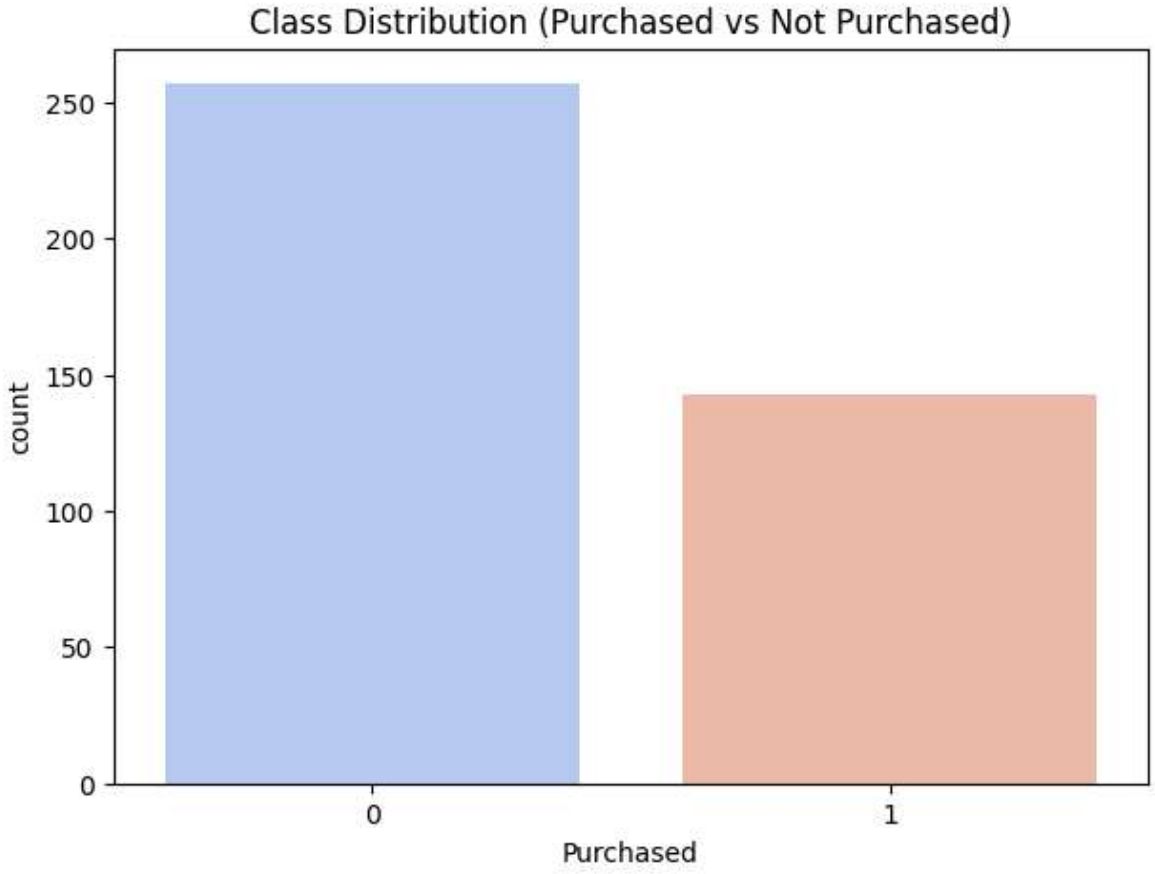
```
Dataset Loaded Successfully
    User ID  Gender  Age  EstimatedSalary  Purchased
0  15624510    Male   19            19000          0
1  15810944    Male   35            20000          0
2  15668575  Female   26            43000          0
3  15603246  Female   27            57000          0
4  15804002    Male   19            76000          0
```

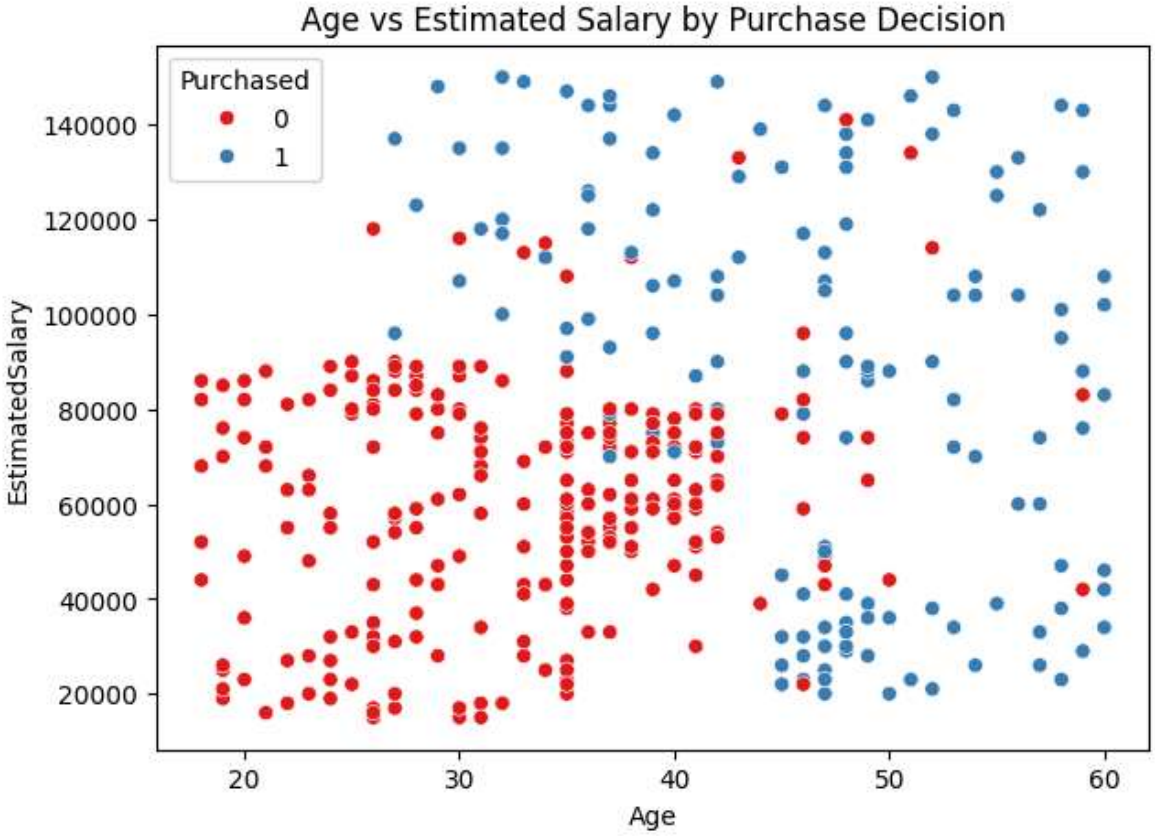
In [3]:
```
print("\nMissing Values:")
print(data.isnull().sum())
```

```
Missing Values:
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [4]:
```
# Step 3: Data Visualization
plt.figure(figsize=(7,5))
sns.countplot(x='Purchased', data=data, palette='coolwarm')
plt.title("Class Distribution (Purchased vs Not Purchased)")
plt.show()

plt.figure(figsize=(7,5))
sns.scatterplot(x='Age', y='EstimatedSalary', hue='Purchased',
                data=data, palette='Set1')
plt.title("Age vs Estimated Salary by Purchase Decision")
plt.show()
```

```
C:\Users\anves\AppData\Local\Temp\ipykernel_3940\4248253477.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x='Purchased', data=data, palette='coolwarm')
```





In [5]:
```python
# Step 4: Feature Selection & Train-Test Split
X = data[['Age', 'EstimatedSalary']]
y = data['Purchased']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)
```

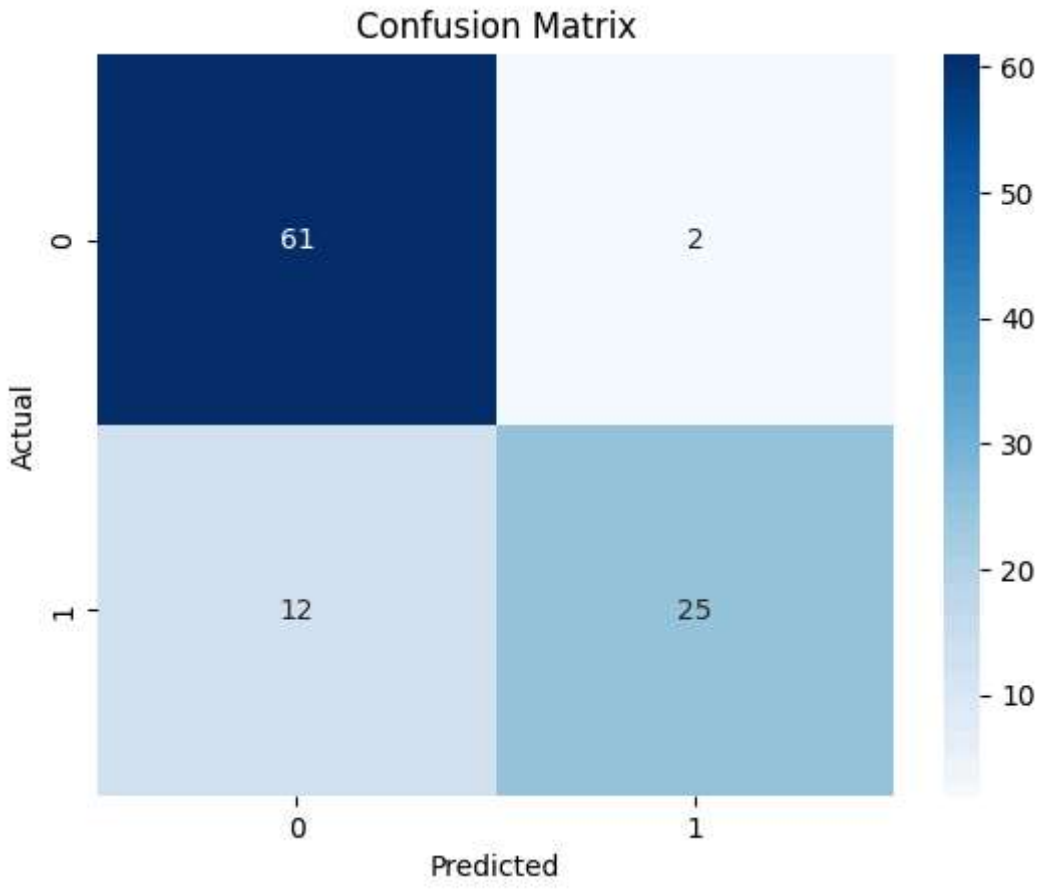
In [6]:
```
# Step 5: Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [7]:
```
# Step 6: Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

print("\nModel Training Complete")
print("Intercept:", model.intercept_)
print("Coefficients:", model.coef_)
```

```
Model Training Complete
Intercept: [-1.03148422]
Coefficients: [[1.92307917 1.07843614]]
```

In [8]:
```
# Step 7: Make Predictions
y_pred = model.predict(X_test_scaled)
y_prob = model.predict_proba(X_test_scaled)[:, 1]
```

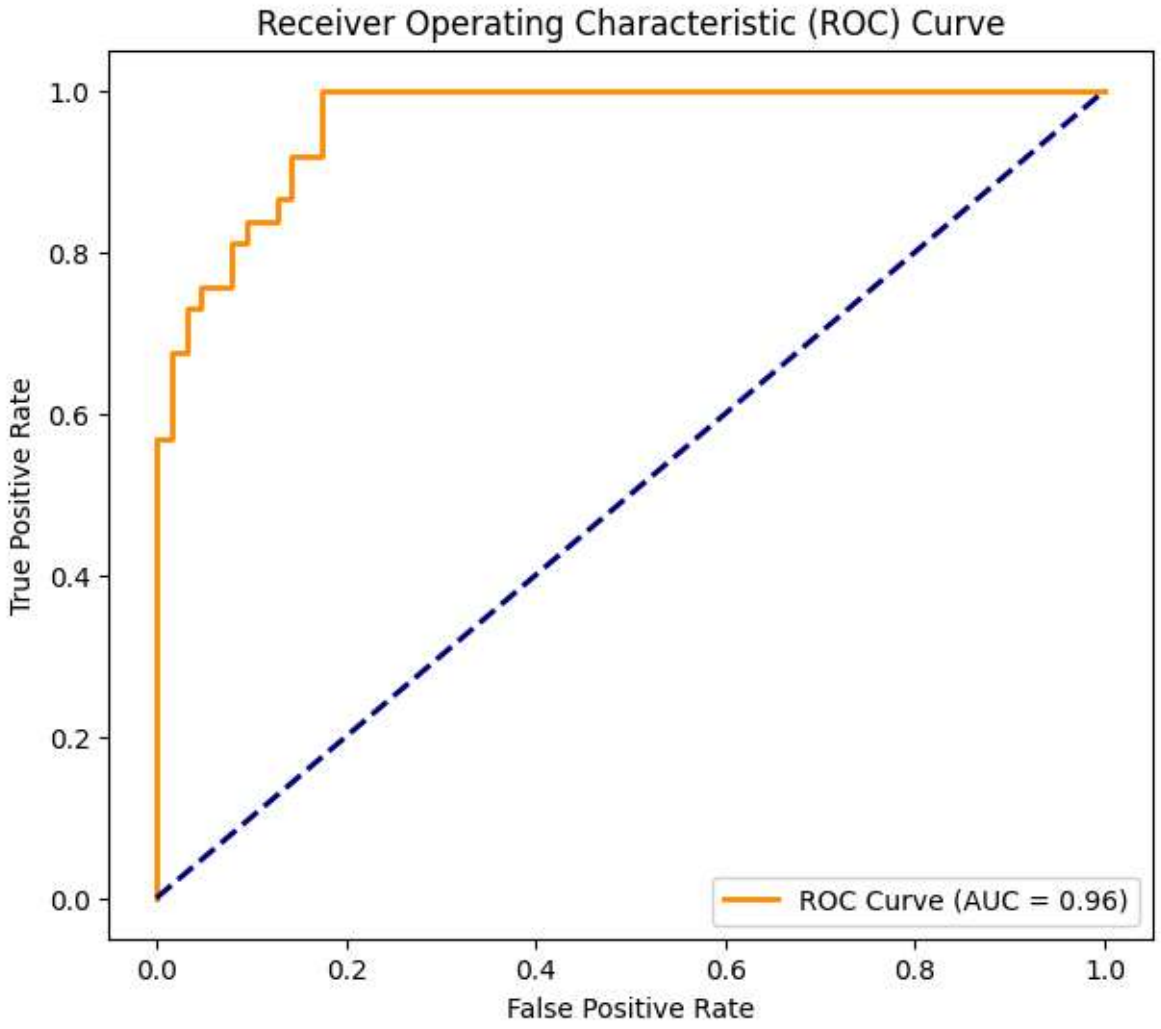
In [9]:
```
# Step 8: Evaluate Model
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



In [10]:
```
# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.97      0.90        63
           1       0.93      0.68      0.78        37

    accuracy                           0.86       100
   macro avg       0.88      0.82      0.84       100
weighted avg       0.87      0.86      0.85       100
```

In [11]:
```python
# Step 9: ROC-AUC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(7,6))
plt.plot(fpr, tpr, color='darkorange', lw=2,
         label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0,1], [0,1], color='navy', lw=2, linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Receiver Operating Characteristic (ROC) Curve")
plt.legend(loc="lower right")
plt.show()

print(f"\nROC-AUC Score: {roc_auc:.2f}")
```




```
ROC-AUC Score: 0.96
```