In [1]:

```
# Data libraries
import pandas as pd
import numpy as np

# Model and preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# Evaluation
from sklearn.metrics import confusion_matrix, accuracy_score, precision_s

# Visualization
import seaborn as sns
import matplotlib.pyplot as plt
```

In [27]:

```
# Load dataset
df = pd.read_csv("Desktop/datasets/Social_Network_Ads.csv")  # path to yo
df.head()
#df.tail()
```

Out[27]:

	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 [7]:

```
# Drop identifier column
df = df.drop("User ID", axis=1)

# Encode Gender
df["Gender"] = df["Gender"].map({"Male": 1, "Female": 0})

df.head()
```

Out[7]:

	Gender	Age	EstimatedSalary	Purchased
0	1	19	19000	0
1	1	35	20000	0
2	0	26	43000	0
3	0	27	57000	0
4	1	19	76000	0

In [9]:

```
X = df[["Age", "EstimatedSalary"]]  # Using numeric features only
y = df["Purchased"]
```

In [11]:

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

In [13]:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [15]:

```
model = LogisticRegression(random_state=42)
model.fit(X_train, y_train)
```

Out[15]:

```
▾ LogisticRegression
LogisticRegression(random_state=42)
```

In [17]:

```
y_pred = model.predict(X_test)
```

In [19]:

```
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Confusion Matrix:
 [[61  2]
 [12 25]]
```

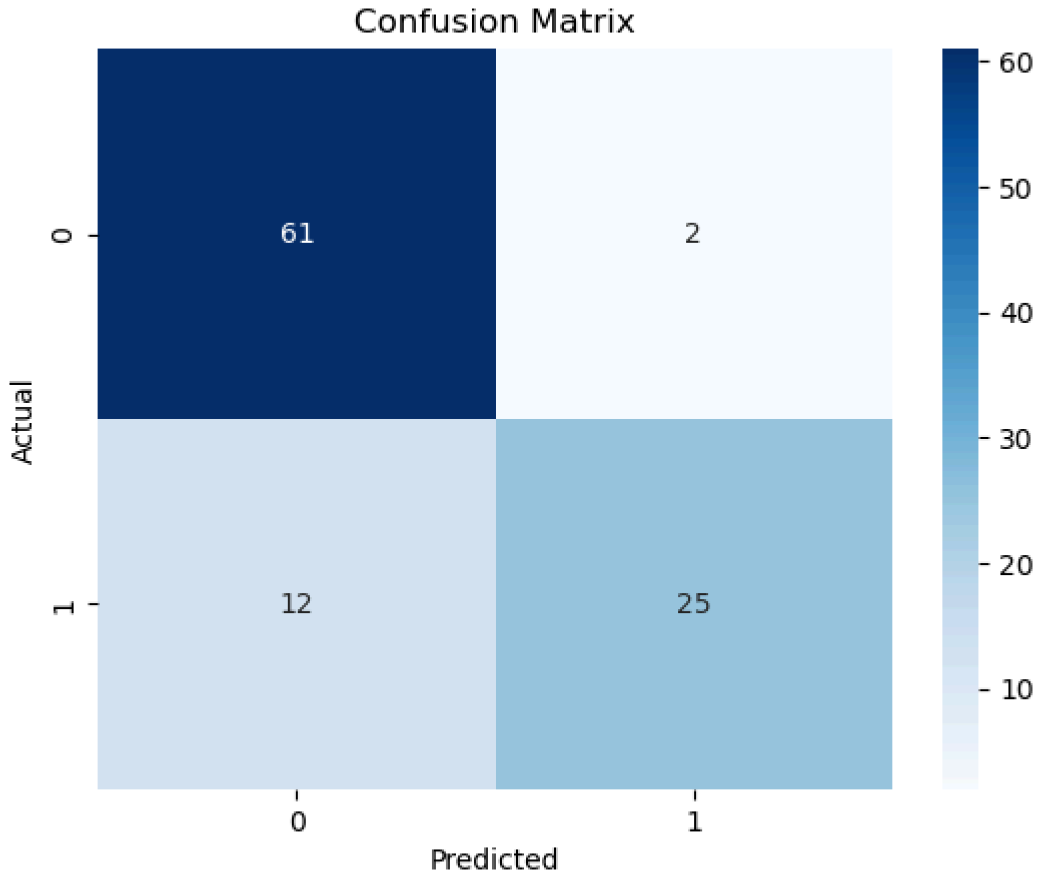
In [21]:

```
# Matrix values
TN, FP, FN, TP = cm.ravel()

# Metrics
accuracy = accuracy_score(y_test, y_pred)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print(f"True Positive (TP): {TP}")
print(f"False Positive (FP): {FP}")
print(f"True Negative (TN): {TN}")
print(f"False Negative (FN): {FN}")
print(f"Accuracy: {accuracy:.4f}")
print(f"Error Rate: {error_rate:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

```
True Positive (TP): 25
False Positive (FP): 2
True Negative (TN): 61
False Negative (FN): 12
Accuracy: 0.8600
Error Rate: 0.1400
Precision: 0.9259
Recall: 0.6757
```

In [23]:

```
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



In [29]:

```
###test
```

In [35]:

```
from sklearn.metrics import f1_score

# Calculate F1 Score
f1 = f1_score(y_test, y_pred)
print(f"F1 Score: {f1:.4f}")
```

```
F1 Score: 0.7812
```

In [ ]: