In [28]:
```
data = pd.read_csv("spam_fixed.csv")
```

In [29]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_m


data.rename(columns={'v1': 'Label', 'v2': 'EmailText'}, inplace=True)
data = data[['Label', 'EmailText']]
data.dropna(inplace=True) # Drop rows with missing values that came from the e

X = data['EmailText']
y = data['Label']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, randc

print(f"Total entries: {data.shape[0]}")
print(f"Training set size: {len(X_train)}")
print(f"Test set size: {len(X_test)}")

# 2. Text Vectorization

count_vectorizer = CountVectorizer()

# Fit the vectorizer
X_train_vectors = count_vectorizer.fit_transform(X_train)
X_train_vectors = X_train_vectors.toarray()

# Transform the test data
X_test_vectors = count_vectorizer.transform(X_test)
X_test_vectors = X_test_vectors.toarray()

print("\nVectorization Complete.")
print(f"Shape of X_train_vectors: {X_train_vectors.shape}")



nb_model = MultinomialNB()
nb_model.fit(X_train_vectors, y_train)

print("\nModel Training Complete using MultinomialNB.")


y_pred = nb_model.predict(X_test_vectors)

# Evaluate performance and calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred, zero_division=0)
```

```
conf_matrix = confusion_matrix(y_test, y_pred)

print("\nModel Evaluation Results:")
print(f"Accuracy: {accuracy:.4f}")
print("\nClassification Report:\n", report)
print("\nConfusion Matrix:\n", conf_matrix)
```

```
Total entries: 5572
Training set size: 4457
Test set size: 1115

Vectorization Complete.
Shape of X_train_vectors: (4457, 7691)

Model Training Complete using MultinomialNB.

Model Evaluation Results:
Accuracy: 0.9839

Classification Report:
               precision    recall  f1-score   support

         ham       0.98      1.00      0.99       965
        spam       0.99      0.89      0.94       150

    accuracy                           0.98      1115
   macro avg       0.98      0.95      0.96      1115
weighted avg       0.98      0.98      0.98      1115


Confusion Matrix:
 [[963   2]
 [ 16 134]]
```

In [ ]: