```
# Phase-3 Project: Revolutionizing Customer Support with an Intelligent Chatbot for Automated Assistance

# Step 1: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re, string

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay

# Step 2: Load/Create Dataset
data = {
    "query": [
        "How do I reset my password?",
        "Why is my bill so high?",
        "App keeps crashing",
        "Change my subscription plan",
        "I can't make calls"
    ],
    "category": ["Technical", "Billing", "Technical", "General", "Technical"]
}
df = pd.DataFrame(data)

# Step 3: Data Preprocessing
def clean_text(text):
    text = text.lower()
    text = re.sub(f"[{re.escape(string.punctuation)}]", "", text)
    return re.sub(r"\d+", "", text)

df['clean_query'] = df['query'].apply(clean_text)
df['query_length'] = df['query'].apply(len)
df['label'] = LabelEncoder().fit_transform(df['category'])

# Step 4: Feature Engineering
tfidf = TfidfVectorizer(max_features=100)
X_tfidf = tfidf.fit_transform(df['clean_query']).toarray()
X = np.hstack((X_tfidf, df[['query_length']].values))
y = df['label']

# Step 5: Model Building
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Step 6: Evaluation
print(classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
```