**Development of a Generative AI-based**

**Quality Management System (QMS)**

**for Airline Travel Call Centers:**

**Introduction:**

The objective of this project is to create a web application tailored for airline travel call centers, aiming to enhance quality management through the utilization of Generative AI technologies. The system will analyze call recordings to extract valuable insights, leveraging the Lyzr AI API for its generative capabilities. By transcribing audio files and extracting meaningful data from these transcriptions, the application will facilitate the evaluation of call center operations and agent performance.

**Key Objectives:**

1. Call Recording Analysis: Utilize Generative AI to transcribe audio recordings from airline travel call centers accurately.
2. Insight Extraction: Extract meaningful insights from transcriptions to assess call center operations and agent performance effectively.
3. KPI Calculation: Implement algorithms to calculate various Key Performance Indicators (KPIs) crucial for evaluating call quality and overall performance.
4. User-Friendly Interface: Design an intuitive web interface for easy navigation and interaction with the system.

**Libraries:**

# Install AssemblyAI library

!pip install assemblyai

# Install SpaCy library

!pip install spacy

# Install Transformers library

!pip install transformers

# Download the SpaCy English language model

!python -m spacy download en\_core\_web\_sm

# Install the AssemblyAI library

!pip install assemblyai

---------------------------------------------------------------------------------------------------------------

import zipfile

import os

# Path to the zip file

zip\_file\_path = "/content/ML Task.zip"

# Directory to extract the files

extracted\_dir = "/content/"

# Create the directory if it doesn't exist

os.makedirs(extracted\_dir, exist\_ok=True)

# Open the zip file

with zipfile.ZipFile(zip\_file\_path, 'r') as zip\_ref:

# Extract all files in the zip file

zip\_ref.extractall(extracted\_dir)

# List all extracted files

extracted\_files = os.listdir(extracted\_dir)

# Filter audio files (assuming they have .mp3 extension)

audio\_files = [file for file in extracted\_files if file.endswith(".mp3")]

# Print the list of audio files

print("Extracted audio files:")

for audio\_file in audio\_files:

print(os.path.join(extracted\_dir, audio\_file))

---------------------------------------------------------------------------------------------------------------

# Set up your AssemblyAI API key

import assemblyai as aai

aai.settings.api\_key = "781c430905c2481e929b18528f5d2411"

# Initialize the transcriber

transcriber = aai.Transcriber()

# Transcribe the audio file

transcript = transcriber.transcribe(r"/content/Call Data Sample/sample\_call\_1.mp3")

# transcript = transcriber.transcribe("./my-local-audio-file.wav")

# Print the transcribed text

print(transcript.text)

---------------------------------------------------------------------------------------------------------------

import matplotlib.pyplot as plt

# Call-Level KPIs

call\_resolution\_rate = 75 # Example value

# Pie chart for Call Resolution Rate

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

plt.pie([call\_resolution\_rate, 100 - call\_resolution\_rate], labels=["Resolved", "Not Resolved"], autopct='%1.1f%%')

plt.title("Call Resolution Rate")

plt.show()

---------------------------------------------------------------------------------------------------------------

import matplotlib.pyplot as plt

# Call-Level KPIs

average\_handle\_time = 25 # Example value

# Bar chart for Average Handle Time (AHT)

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

plt.bar(["AHT"], [average\_handle\_time], color='skyblue')

plt.ylabel("Minutes")

plt.title("Average Handle Time (AHT)")

plt.show()

---------------------------------------------------------------------------------------------------------------

import matplotlib.pyplot as plt

# Call-Level KPIs

customer\_satisfaction\_score = 85 # Example value

# Donut chart for Customer Satisfaction Score (CSAT)

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

plt.pie([customer\_satisfaction\_score, 100 - customer\_satisfaction\_score], labels=["Satisfied", "Not Satisfied"], autopct='%1.1f%%', colors=['lightgreen', 'lightcoral'])

plt.title("Customer Satisfaction Score (CSAT)")

plt.show()

---------------------------------------------------------------------------------------------------------------

import matplotlib.pyplot as plt

# Call-Level KPIs

error\_rate = 3 # Example value

# Bar chart for Error Rate

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

plt.bar(["Error Rate"], [error\_rate], color='salmon')

plt.ylabel("Percentage")

plt.title("Error Rate")

plt.show()

---------------------------------------------------------------------------------------------------------------

**Conclusion:**

The development of a Generative AI-based Quality Management System tailored for airline travel call centers represents a significant step towards improving call quality, enhancing agent performance, and optimizing operations. By leveraging advanced technologies and insightful data analysis, the system aims to empower call center managers with the tools and insights necessary to deliver exceptional customer service and drive organizational success.