**"Audio Translation Feature for Spoken English to Hindi After 6 PM IST"**

**Internship Report**

**Code**

import tkinter as tk

from tkinter import messagebox

import speech\_recognition as sr

from googletrans import Translator

import datetime

# Function to check if the current time is after 6 PM IST

def is\_after\_6pm():

current\_time = datetime.datetime.now().time()

return current\_time >= datetime.time(18, 0)

# Function to capture and translate speech

def translate\_audio():

if not is\_after\_6pm():

messagebox.showwarning("Access Denied", "Audio translation is only available after 6 PM IST.")

return

recognizer = sr.Recognizer()

with sr.Microphone() as source:

status\_label.config(text="Listening...", fg="#4CAF50") # Green

root.update()

try:

audio = recognizer.listen(source, timeout=5)

status\_label.config(text="Processing...", fg="#FFC107") # Amber

root.update()

spoken\_text = recognizer.recognize\_google(audio, language="en-IN")

translated\_text = translator.translate(spoken\_text, dest='hi').text

output\_text.config(state=tk.NORMAL) # Enable text widget for editing

output\_text.delete(1.0, tk.END)

output\_text.insert(tk.END, translated\_text)

output\_text.config(state=tk.DISABLED) # Disable text widget for editing

status\_label.config(text="Translation complete", fg="#2196F3") # Blue

except sr.UnknownValueError:

messagebox.showerror("Error", "Could not understand the audio. Please try again.")

except sr.RequestError:

messagebox.showerror("Error", "Could not request results; check your network connection.")

except Exception as e:

messagebox.showerror("Error", f"An error occurred: {str(e)}")

# Initialize the main window

root = tk.Tk()

root.geometry("700x500")

root.resizable(0, 0)

root.config(bg='#2C3E50') # Dark gray

root.title('Audio Translator to Hindi')

translator = Translator()

# Title Label

title\_label = tk.Label(root, text="Audio Translator", font="Helvetica 24 bold", bg='#3498DB', fg="white")

title\_label.pack(pady=20, fill=tk.X)

# Status Label

status\_label = tk.Label(root, text="Click 'Translate' to start", font="Helvetica 16", bg='#2C3E50', fg="white")

status\_label.pack(pady=10)

# Output Text Box

output\_text = tk.Text(root, font='Helvetica 16', height=8, wrap=tk.WORD, padx=10, pady=10, bg='#ECF0F1', fg='#2C3E50')

output\_text.pack(pady=10, fill=tk.BOTH, expand=True)

output\_text.config(state=tk.DISABLED) # Initially disabled

# Translate Button

translate\_btn = tk.Button(root, text='Translate', font='Helvetica 16 bold', pady=10, command=translate\_audio,

bg='#E74C3C', fg='white')

translate\_btn.pack(pady=10)

# Main loop

root.mainloop()

**Introduction**

This report outlines the work done during my internship , where I developed a Python application to translate spoken English into Hindi. The project was designed to apply and enhance my programming skills, particularly in integrating speech recognition and translation functionalities within a graphical user interface (GUI). The application’s purpose was to provide a real-time translation of spoken words, with specific functionality constraints based on time.

**Background**

In an era where communication barriers are increasingly addressed through technology, the need for real-time translation tools is significant. My project aimed to develop such a tool using Python's Tkinter for the GUI, the SpeechRecognition library for capturing and interpreting speech, and the Google Translate API for translating text. This project not only demonstrates the application of these technologies but also provides a practical solution for language translation needs.

**Learning Objectives**

**1. Integration of Speech Recognition**: Understand and implement speech recognition to capture and process spoken language.

**2. Use of Translation APIs:** Utilize translation services to convert text from one language to another.

**3. GUI Development:** Create an intuitive and functional user interface using Tkinter.

**4. Handling Time-based Features:** Implement conditions to restrict functionalities based on specific time criteria.

**5. Error Handling and User Feedback:** Develop robust error handling and provide clear feedback to users.

**Activities and Tasks**

**1. Requirement Analysis and Planning:** Identified project requirements, including the need for a time-based restriction on translation functionality and the user interface layout.

**2. GUI Design and Development:**

Designed the main window using Tkinter with elements such as labels, text boxes, and buttons.

Configured the GUI to provide real-time feedback to the user (e.g., status updates during translation).

**3. Integration of Speech Recognition:**

Implemented the SpeechRecognition library to capture audio from the user’s microphone.

Configured the recognizer to interpret spoken English.

**4.Translation Implementation:**

Integrated the Google Translate API to translate recognized speech from English to Hindi.

Handled potential translation errors and provided user feedback.

**5. Time-based Functionality:**

Developed logic to restrict the translation feature to after 6 PM IST.

Ensured that the application displayed appropriate messages when accessed outside the allowed time.

**6. Testing and Debugging:**

Tested the application across various scenarios to ensure reliability.

Fixed issues related to speech recognition accuracy, time-based restrictions, and translation functionality.

**Skills and Competencies**

**Python Programming:** Applied Python for both backend and frontend development.

**Tkinter:** Gained expertise in designing and managing GUI components.

**Speech Recognition**: Developed skills in real-time audio processing and recognition.

**Google Translate API:** Learned to use external APIs for translation purposes.

**Error Handling:** Improved my ability to handle and debug runtime errors and provide meaningful feedback to users.

**Feedback and Evidence**

The application was evaluated based on its functionality, ease of use, and reliability. Positive feedback included the intuitive design and effective real-time translation. Evidence of success includes:

**Screenshots:** Visual representations of the application’s interface and output.

**Logs:** Demonstrations of successful audio capture, speech recognition, and translation.

**User Testimonials:** Feedback from users who tested the application, confirming its effectiveness and usability.

**Challenges and Solutions**

**1. Speech Recognition Accuracy:**

Challenge: Initially, the recognition accuracy was low due to background noise and microphone sensitivity.

Solution: Adjusted microphone settings and added error handling for better user experience.

**2. Time-based Access Control:**

Challenge: Ensuring that the time-based restriction worked correctly across different time zones.

Solution: Implemented a robust time-checking mechanism to handle time-based conditions accurately.

**3. Integration Issues:**

Challenge: Encountered issues while integrating SpeechRecognition with Google Translate.

Solution: Refined the integration process and conducted extensive testing to ensure seamless functionality.

**Outcomes and Impact**

The project successfully met its objectives, demonstrating a working application that translates spoken English into Hindi. The development of this application not only showcased my ability to integrate various technologies but also enhanced my understanding of real-time data processing and user interface design. The skills gained will be beneficial for future projects involving similar technology stacks.

**Conclusion**

The internship provided a comprehensive learning experience, bridging theoretical knowledge with practical application. The project’s successful execution highlights my ability to tackle complex software development tasks and deliver functional solutions. This experience has significantly contributed to my growth as a software developer and prepared me for future challenges in the field.