

REALTIME SRT SYSTEM FOR CUSTOMER SUPPORT AUTOMATION

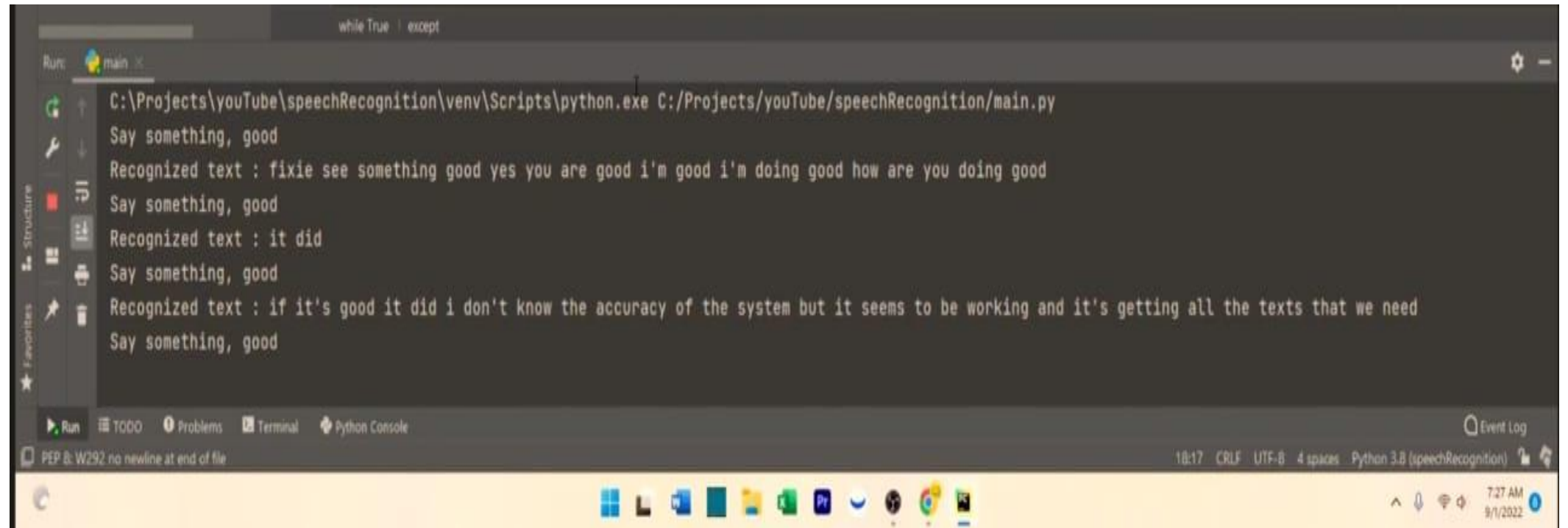
INTRODUCTION:

- A Real-time SRT System for customer support automation typically involves Speech Recognition Technology (SRT) to transcribe spoken words into text instantly. It can enhance customer service by allowing automated assistants to respond dynamically based on real-time speech input. Such a system can integrate Natural Language Processing (NLP), AI-driven chatbot systems, and databases to ensure seamless customer interactions. Here's a simple Python-based example using `speech_recognition` and `NLTK` for real-time voice-to-text processing in a customer support scenario:

PROGRAM:

- `import speech_recognition as sr`
- `#Initialize recognizerrecognizer = sr.Recognizer()`
- `# Load the audiofile`
- `audio_file =`
- `"path_to_your_audio_file.wav" # Replace with your actual file path`
- `with sr.AudioFile(audio_file) as`
- `source:`
- `recognizer.adjust_for_ambient_noise(source) # Helps with noisy backgrounds print("Transcribing...")`
- `audio = recognizer.record(source)`
- `# Read the entire audio file# Recognize the speech using PocketSphinx (offline)`
- `try:`
- `text =`
- `recognizer.recognize_sphinx(audio)`
- `# Offline recognition`
- `print("Transcription:", text)`
- `except sr.UnknownValueError:`
- `print("Sorry, could not understand the audio.")`
- `except sr.RequestError:`
- `print("PocketSphinx is not working properly.")`

OUTPUT:



The screenshot shows a Python IDE with a dark theme. The main editor window displays the output of a program. The output consists of several lines of text, including the command to run the program and the recognized text from a speech recording. The IDE interface includes a sidebar with 'Structure' and 'Favorites' views, a bottom toolbar with 'Run', 'TODO', 'Problems', 'Terminal', and 'Python Console' buttons, and a status bar at the bottom showing the file encoding and the current Python interpreter.

```
while True | except  
Run: main.py  
C:\Projects\youTube\speechRecognition\venv\Scripts\python.exe C:/Projects/youTube/speechRecognition/main.py  
Say something, good  
Recognized text : fixie see something good yes you are good i'm good i'm doing good how are you doing good  
Say something, good  
Recognized text : it did  
Say something, good  
Recognized text : if it's good it did i don't know the accuracy of the system but it seems to be working and it's getting all the texts that we need  
Say something, good
```

18:17 CRLF UTF-8 4 spaces Python 3.8 (speechRecognition)

7:27 AM 9/1/2022

CONCLUSION:

How It Works:

- 1.Speech Recognition:** Captures and converts customer speech into text using Google's speech API.
 - 2.Automated Response:** Matches the query against predefined responses and provides a relevant answer.
 - 3.Continuous Learning:** This can be enhanced with **machine learning** to offer smarter responses over time.
- Would you like enhancements, such as **database integration or voice-based authentication?** 