

LEVIATHAN

Morse Code Converter

ARTIFICIAL INTELLIGENCE
PROJECT



Member

001202200158
Almika Abu
Ghiffari

001202200174
Clement Delano
Samosir

001202200164
Indah Novianti
Setyoningrum

001202200182
Rafly Athallah
Pasya

001202200071
Satria Ibnu Pamungkas

Introduction:

The application, which utilizes speech recognition AI, demonstrates the integration of AI technology. By incorporating the `speech_recognition` library, the program highlights AI's ability to understand and interpret human speech, allowing users to interact with it through spoken commands.

The program also showcases AI's adaptability by processing recognized speech and executing contextually appropriate actions, as shown in its greeting and farewell responses. Furthermore, the application's ability to convert text into

Morse code and generate corresponding sounds illustrates AI's versatility in processing and transforming data across different modalities. In summary, this application's combination of AI components enables it to recognize, process, and respond to human speech, enhancing its usability and overall functionality



The AI Algorithm Used in Our Project

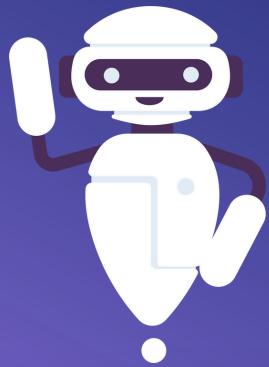
The speech recognition functionality in my project relies on the Google Web Speech API, which utilizes advanced artificial intelligence techniques, particularly deep learning algorithms.

While the exact details of the algorithms used within the API are proprietary and not publicly disclosed, it's widely understood that deep neural networks (DNNs) play a significant role.

These deep learning algorithms are trained on large datasets of speech samples to learn patterns and relationships between audio features and corresponding text transcriptions. This enables the system to accurately transcribe spoken words into text, even in noisy environments or with various accents.

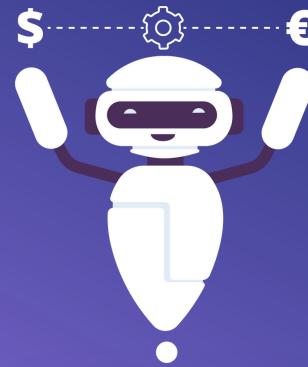


THE COMMANDS USED:



EXTERNAL MODULES AND LIBRARIES:

- 'speech_recognition' : Used for speech recognition and converting speech input into text.
- 'MorseCodePy' : Library providing functionalities to encode and decode text into Morse code, as well as play Morse code as sound.
- 'pyttsx3' : Used for speech synthesis, converting text into speech.



FUNCTION 'CAPTURE_VOICE_INPUT()':

- Utilizes the 'speech_recognition' module to capture speech input from the user via the microphone.

FUNCTION 'SPEAK(TEXT)':

- Uses the 'pyttsx3' library to vocalize the given text using speech synthesis.



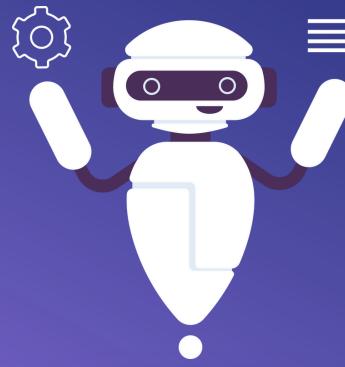
FUNCTION 'CONVERT_VOICE_TO_TEXT(AUDIO)':

- Employs 'speech_recognition' to convert speech input into text.

FUNCTION

'ENCODE_TO_MORSE(TEXT)':

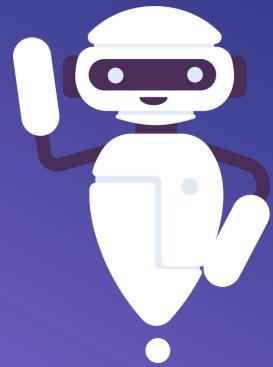
- Utilizes 'MorseCodePy' to convert text into Morse code.



FUNCTION 'PROCESS_VOICE_COMMAND'(TEXT):

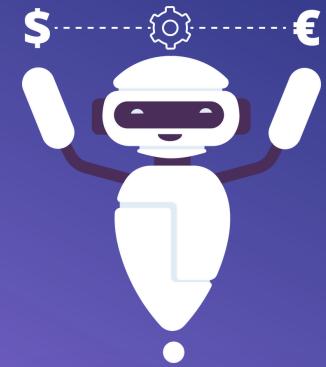
- Uses the recognized text to generate a response or decide whether the program should terminate.
- Processes the recognized speech command converted into text.

THE COMMANDS USED:



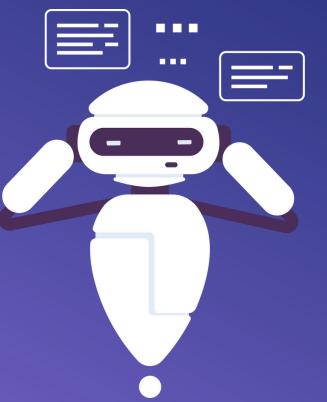
FUNCTION
`'MORSE_TO_SOUND'(TEXT):`

- Plays sound based on the given Morse code using 'MorseCodePy.'



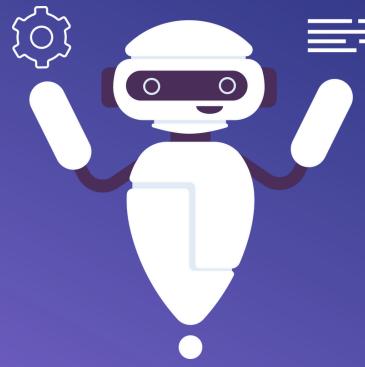
FUNCTION
`DECODE_MORSE_INPUT
(CIPHERT):`

- Decodes the given Morse code message into text using 'MorseCodePy.'



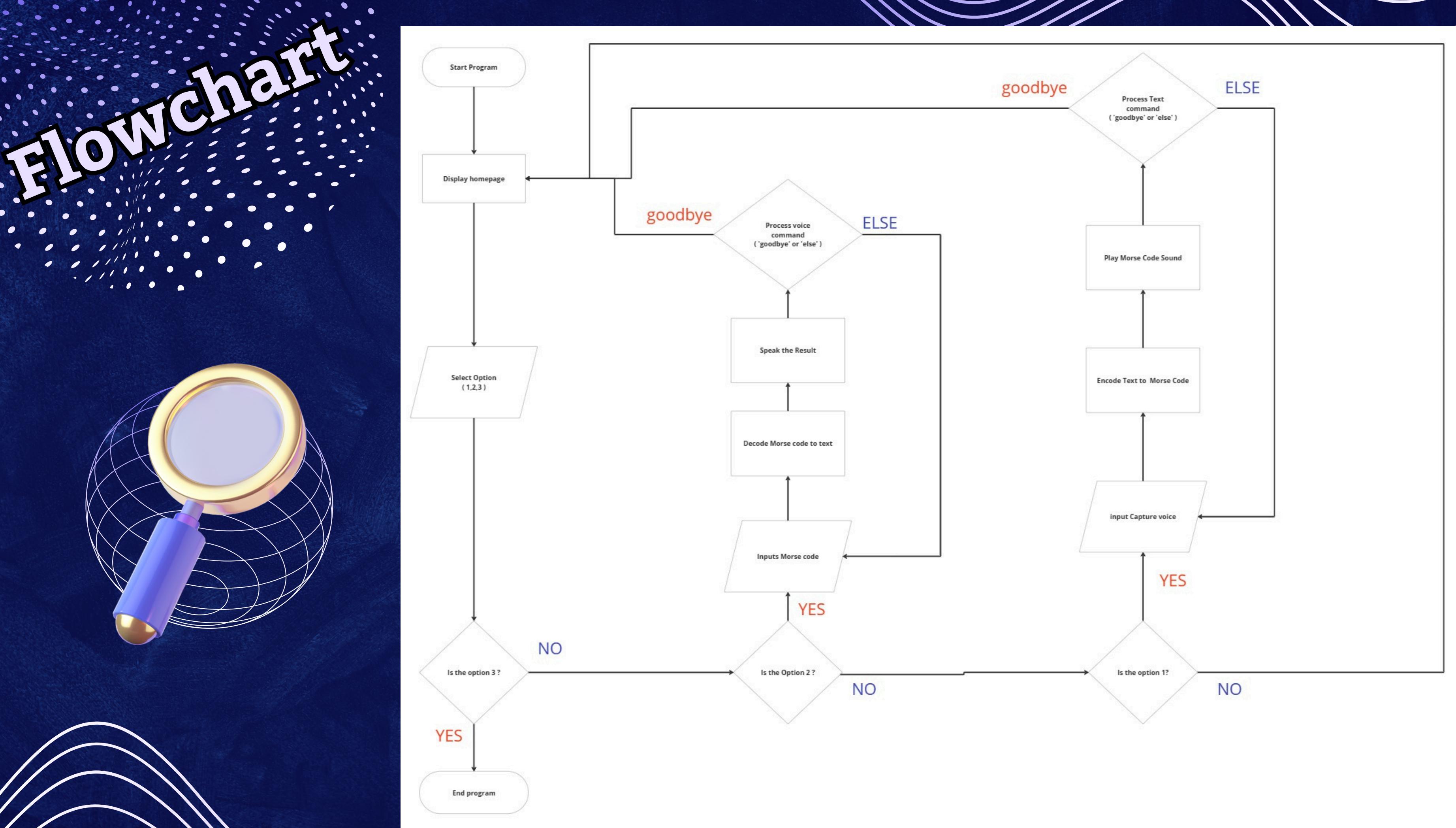
FUNCTION 'HOMEPAGE()':

- Prints the main menu of the program and prompts the user to select an option.



FUNCTION 'MAIN()':

- The main function of the program containing the main loop to execute the program.
- Processes user inputs and calls appropriate functions based on user's choice.



THANK YOU!

