Here’s my complete Alpha Voice Assistant Code Explanation formatted as a beginner-friendly.

🧠 Alpha Voice Assistant – Full Code Explanation

This document breaks down the entire Python code for Alpha Voice Assistant into clear, easy-to-understand sections. Whether you're a beginner or a curious developer, this guide will help you grasp how Alpha works and how each part contributes to its functionality.

🔊 Text-to-Speech: speak(audio)

- Uses pyttsx3 to convert text into speech.
- Initializes the engine and selects a voice named "Ravi" for a personalized Indian accent.
- Sets the speaking rate to 150 words per minute.
- Speaks the given text aloud.

🎙 Voice Command Recognition: commands()

- Uses speech_recognition to listen through the microphone.
- Adjusts for ambient noise and listens for voice input.
- Converts speech to text using Google’s speech API.
- Handles errors gracefully and prompts the user to repeat if needed.

🌅 Time-Based Greetings: greets()

- Uses datetime to get the current hour.
- Greets the user with a personalized message based on the time of day:
  - Morning (0–12)
  - Afternoon (12–16)

    - Evening (16–20)
    - Night (20–24)

🔐 Secure Startup: alphastatus() and Password Check

- Displays "Alpha is offline..." at launch.
- Prompts the user to enter a password ("ALPHA") to activate the assistant.
- Ensures secure and exclusive access.

📅 Time, Date, and Day Queries

- Responds to voice commands like "time", "date", or "day".
- Uses datetime to format and speak the current time, date, or weekday.

🌐 Browser Control

- Opens Microsoft Edge using os.startfile.
- Closes Edge using os.system("taskkill /f /im msedge.exe").
- Confirms each action with voice feedback.

📺 YouTube Playback

- Detects commands like "play [video name]".
- Uses pywhatkit.playonyt() to play the video on YouTube.
- Speaks confirmation before launching.

📖 Wikipedia Summaries

- Detects "wikipedia" in the command.
- Uses wikipedia.summary() to fetch a 3-sentence summary.

- Reads the summary aloud or informs if no results are found.

🧠 System Monitoring

- Uses psutil to check:
    - CPU usage
    - RAM usage
    - Battery percentage
- Speaks the system status in a clear format.

📝 Notepad Dictation

- Opens Notepad using os.system("start notepad.exe").
- Listens for dictated text and types it using pyautogui.
- Supports:
    - Creating a new page (ctrl + n)
    - Saving the note (ctrl + s)
    - Naming the file via voice
    - Closing Notepad with or without saving

📸 Screenshot Capture

- Takes a screenshot using pyautogui.screenshot().
- Previews the screenshot and closes the preview window.
- Handles errors and provides voice feedback.

❌ YouTube Tab Closure

- Scans running processes using psutil.
- Identifies Edge tabs with "youtube.com" in the command line.

- Closes only the YouTube tab and confirms success or failure.

🛑 Graceful Exit

- Detects commands like "exit" or "alpha exit".
- Speaks a goodbye message and stops the assistant loop.

⚠️ Error Handling and Feedback

- Uses try-except blocks to catch and report errors.
- Always provides voice feedback to keep the user informed.

🧩 Modular Design for Expansion

- Each feature is wrapped in a function for clarity and scalability.
- Easy to add future enhancements like:
  - Emotion detection
  - Context memory
  - Smart device integration
  - LLM-powered conversations
