**Project Report: Nebula Virtual Assistant**

**Introduction:**

The Nebula Virtual Assistant is an AI-powered system. It is designed so that the user can give input that will be received or otherwise responded to through speech-activated voice recognition systems designed to help with common needs such as finding a good restaurant, sending a scheduled email, or shopping. The system makes efficient and effective use of a myriad of techniques in natural language processing, including those required for text-to-speech and speech recognition techniques.

This is a Python-based project aimed at incorporating a few external libraries that have allowed speech interaction and also task automation. This virtual assistant can interpret voice commands and do web searches, emails, and even give response answers to queries.

**Features and Capabilities:**

* Speech Recognition

Nebula uses the speech recognition library, where it translates voice commands uttered by the user into text. The assistant listens to the user's voice and processes it into commands the system can understand.

Library Used: speech recognition

Functionality: Listen for user input and recognizes commands such as "send an email," "search Wikipedia," or "tell me a joke."

* Text-to-Speech (TTS)

It uses audio feedback to the user using the pyttsx3 library, thereby enabling it to speak out and provide a response to questions raised by the user.

Library Used: pyttsx3

Voice Configuration: Male-female voices can be interchanged at the user's choice.

Functionality: Can be used to voice output greeting messages, accepting the tasks, and retrieving results for search queries.

* Sending Emails

Nebula can send emails using the smtplib library. The assistant asks for the recipient's address and the contents of the email using voice input and sends it accordingly.

Library Used: smtplib

Functionality: Connects with the Gmail SMTP server securely and sends emails by pulling credentials from environment variables.

Nebula interfaces with Wikipedia and web browser modules to search the web for the information and present to the users.

Library Used: Wikipedia, web browser

Functionality: Answers the question by fetching the suitable data from Wikipedia or opening pages in the default browser by opening the link of that page.

* Telling Jokes

To make the application interesting, Nebula interfaces with pyjokes library for joke-telling based on the command of the user.

Library Used: pyjokes.

Functionality: This provides the user with what seems to be a friendly response when requesting a joke.

**Key Functional Components:**

**Voice Command Listener**

This function uses a speech recognition module that captures and transforms spoken input into text that the assistant can process.

* **Function Wish Me**

This is a greeting functionality that says good morning, good afternoon, or good evening; it will check the clock of the system and produce the output accordingly.

* **Sent Emails**

A healthy feature of emailing is included with smtplib using SMTP authentication. Credentials are kept safely in environment variables so that they can easily be accessed during execution.

* **Task Automation**

While Nebula can simply automate most tasks with voice commands, she can do anything from:

Opening sites Matching Information Searching Email Composition Comics for fun.

**External Libraries and Dependencies:**

The following are the Python libraries needed for the project. They must be installed before running the assistant:

pyttsx3: For text-to-speech conversion.

speech\_recognition is for voice command interpretation.

Wikipedia: For pulling information from Wikipedia.

pyjokes: To entertain users with jokes. smtplib for sending mails via SMTP,

**Conclusion:**

The Nebula Virtual Assistant has the potential for developing task automation and human-computer interaction through artificial intelligence. It can even do necessary tasks like send an email, find information for me, or entertain in exchange for voice commands. Development in this would allow the accompaniment of additional tasks, and the more use the product has toward assisting its users in everyday tasks, the better it becomes.

By team: ACT   
Tanmay (2320040093)

Chaitanya (2320040003)

Akshay (2320040128)