

Jarvis – AI Voice Assistant

Introduction

Jarvis is a Python-powered AI voice assistant inspired by Alexa and Google Assistant. It allows natural language interactions through voice commands to perform various tasks such as launching apps, browsing websites, fetching real-time data (news/weather), and leveraging large language models for intelligent conversations.

Key Features

- Wake Word Activation — Responds to the keyword "Jarvis"
- Speech Interface — Converts speech to text and vice versa (Google API + pytsx3)
- App & Web Control — Launch/close applications or open websites with voice
- Smart AI Responses — Integrated with Groq (LLaMA & Compound Beta models)
- YouTube Playback — Play specific videos via voice
- Weather Updates — Fetches real-time weather reports
- News Headlines — Summarizes trending news
- App Termination — Close apps or browser tabs using voice commands

Tech Stack

Voice Input: speech_recognition (Google API)

Voice Output: pytsx3

LLM Integration: Groq API

Web Automation: webbrowser, requests, pywhatkit

Custom Command Logic: apps.py, browser.py

OS Interaction: os, winsound (Windows only)

Project Structure

Jarvis/

- main.py # Core execution logic
- apps.py # App path configuration
- browser.py # Website URL mapping
- requirements.txt # Dependency definitions
- beep2.wav # Wake-up sound file

Getting Started

1. Clone the Repository:

```
git clone https://github.com/yourusername/jarvis-assistant.git  
cd jarvis-assistant
```

2. Install Required Packages:

```
pip install -r requirements.txt
```

3. Configure API Keys:

- newsapi = ""
- Set api_key in the Groq(...) constructor

4. Launch Jarvis:

```
python main.py
```

Sample Commands

“Jarvis” — Activates assistant

“Open YouTube” — Launches YouTube in browser

“Start Notepad” — Opens Notepad application

“Play Despacito” — Plays video on YouTube

“What’s the weather?” — Retrieves and reads current weather

“News” — Reads top news headlines

“Close WhatsApp” — Terminates WhatsApp

Customization Options

- Application Paths: Modify apps.py to suit your installed programs
- Web Links: Edit browser.py with preferred website URLs
- TTS Speed: Adjust speaking rate using engine.setProperty("rate", 170)

Dependencies

pytsx3

speechrecognition

requests

pywhatkit

groq

Known Limitations

- Windows-only compatibility (os.system, winsound)
- Requires internet for speech recognition & AI responses
- Wake word detection lacks fallback logic
- beep2.wav file must be present in the project root

Author

Jitesh Nepal

Passionate about AI, automation, and Python development.

Created Jarvis to streamline digital interaction with voice-first control.