

# PROJECT REPORT: AI VOICE ASSISTANT ("MUMMY")

## 1. Project Overview

**Project Name:** Mummy AI Assistant

**Developer:** Softcapphyjas

**Language:** Python 3.10+

**Purpose:** To develop a highly responsive, bilingual (English/Hindi) personal assistant that streamlines web navigation, fetches real-time information, and automates media playback through voice commands.

---

## 2. System Architecture

The system is designed using a **modular approach**, separating the core logic from the data libraries. This allows for easy updates to the website and music databases without modifying the main execution engine.

### The Processing Pipeline:

1. **Speech Recognition:** Captures audio via `speech_recognition` and converts it to text using the Google Web Speech API.
  2. **Intent Recognition:** Analyzes the text for keywords like "open," "play," "news," or "who is."
  3. **Library Lookup:** Matches keywords against `Soical_media_Library.py` or `MusicLibrary.py`.
  4. **Action Execution:** Uses the web browser and requests modules to perform the task.
  5. **Voice Feedback:** Delivers a response via the `pyttsx3` text-to-speech engine.
- 

## 3. Functional Modules

### 3.1 Bilingual News Aggregator

The assistant connects to the **NewsData.io API** to fetch the latest headlines. It is programmed to distinguish between English and Hindi requests.

- **API Integration:** Uses `requests.get()` to fetch JSON data.
- **Multilingual Output:** If "Hindi" is detected in the command, the assistant switches its voice profile to a Hindi-compatible ID.

### 3.2 Intelligent Information Retrieval

The assistant utilizes the **Wikipedia API** for educational queries.

- **Summarization:** It extracts exactly 5 sentences to keep the response concise.
- **Translation:** Integrated with googletrans, it can translate English Wikipedia summaries into Hindi on the fly for better accessibility.

### 3.3 Secure Web & Utility Navigation

Instead of relying solely on search engine results, which may contain ads or phishing links, the assistant uses a **Hard-Coded Mapping System**.

- **SML (Social Media Library):** Contains over 100 links including Govt. Portals (Aadhar, PAN), Banking (SBI, HDFC), and AI Tools (ChatGPT, Gemini).
- **Accuracy:** Ensures the user lands on the official domain every time.

---

## 4. Technical Stack & Dependencies

Category	Library/API	Purpose
Speech Processing	SpeechRecognition	Converts vocal input to string data.
Synthesis	pyttsx3	Offline text-to-speech conversion.
Network	requests	Facilitates API calls for news and data.
Logic/Search	wikipedia, pywhatkit	Information scraping and advanced searching.
Audio Alerts	winsound	Provides UI sound effects (1.wav, 3.wav).
Translation	googletrans	Handles English-to-Hindi text conversion.

---

## 5. Library Analysis

### 5.1 Music Repository (120+ Tracks)

The assistant features a curated MusicLibrary that maps song titles to direct YouTube URLs.

- **Features:** Includes Old Hindi Classics, Modern Pop, and Instrumental tracks.
- **Efficiency:** Playing a song from the library is **30% faster** than performing a live search.

### 5.2 Utility & Services (100+ Sites)

The website dictionary acts as a "Voice-Activated Bookmark Manager."

- **Categories:** PDF Modifiers, Indian Government Portals, Banking, Shopping, and Food Delivery.
- 

## 6. Error Handling & Optimization

- **Ambient Noise Adjustment:** The system uses `r.adjust_for_ambient_noise` for 0.75 seconds before listening to ensure accuracy in noisy rooms.
  - **Exception Catching:** The main loop is wrapped in a try-except block to prevent the program from crashing if audio is unclear or the internet disconnects.
  - **Wake Word Mechanism:** To save resources, the full command processor only activates after the wake word "**Mummy**" is recognized.
- 

## 7. Future Enhancements

1. **gTTS Integration:** Replacing the current engine with Google Cloud TTS for more natural, human-like inflection.
  2. **System Control:** Adding capabilities to shut down the PC, adjust volume, or open local applications (Excel, Word).
  3. **GPT-4 Integration:** Moving from keyword-based logic to a Large Language Model (LLM) for more conversational "Mummy" interactions.
- 

## 8. Conclusion

The "Mummy" AI Voice Assistant is a successful implementation of Python-based automation. By combining local library mapping with global API data, it provides a fast, safe, and personalized user experience. It serves as a robust foundation for further development into a fully autonomous smart-home or productivity tool.