Guru Gobind Singh College of Engineering and Research Centre, Nashik.

Department of Computer Engineering

Project Synopsys

Academic Year: **2025-26**

Project Title	**CallGPT An AI-Powered Voice Assistant for Customer Care**
Team Members	1. Chanchal Sandip Aher
	2. Darade Akanksha Balasaheb
	3. Parth Gajanan Patil
	4. Roshan Jankiram Pawara
Internal Guide (Name and Sign)	Mr.P.R.Kulkarni
Project Coordinator (Name and Sign)	Mr.A.R.Jain

Guru Gobind Singh College of Engineering and Research Centre, Nashik
Department of Computer Engineering

Academic Year 2025-26

Final Year Project Synopsys Format

Title: CallGPT An AI-Powered Voice Assistant for Customer Care

Objective and Scope:

The main objective of this project is to develop **CallGPT**, an intelligent voice-based customer support assistant that uses **Retrieval-Augmented Generation (RAG)** to deliver accurate, real-time, and context-aware responses to user queries over a phone call.

CallGPT combines **speech-to-text (STT)**, **language model (LLM)** reasoning, **retrieval from organization-specific knowledge**, and **text-to-speech (TTS)** to enable seamless and automated customer care experiences. The system reduces human intervention and provides consistent support 24/7.

Process Description:

The proposed system, **CallGPT**, is an AI-powered voice assistant designed to handle customer service queries through voice calls using advanced AI technologies like **Speech-to-Text (STT)**, **Retrieval-Augmented Generation (RAG)**, and **Text-to-Speech (TTS)**.

The overall working process involves:

Use Case Diagram:

1. Incoming Call Handling: The user initiates a phone call which is routed through a telephony platform (Vapi).

2. Speech-to-Text Conversion (STT): The caller's voice is recorded and converted into text using STT engines like OpenAI Whisper or Amazon Transcribe.

3. Retrieval-Augmented Generation (RAG):
   - The transcribed query is passed to the RAG engine.
   - The RAG module uses semantic search over a vector database (Pinecone/Weaviate/pgvector) to fetch relevant documents from internal data sources (FAQs, manuals, CRM notes).
   - These documents are given to a Large Language Model (LLM) like GPT-4 or LLaMA to generate a customized and accurate response.

4. Text-to-Speech (TTS):
   The AI-generated response is then converted into natural speech using a TTS service like Amazon Polly or Coqui TTS.

5. Response Delivery: The spoken reply is streamed back to the caller via the telephony platform.

6. Call Flow :
   The system also handles:
   - Logging the call transcript for analytics.
   - Escalating the call to a human agent when AI is unsure.
   - Storing interactions for future training/improvements.

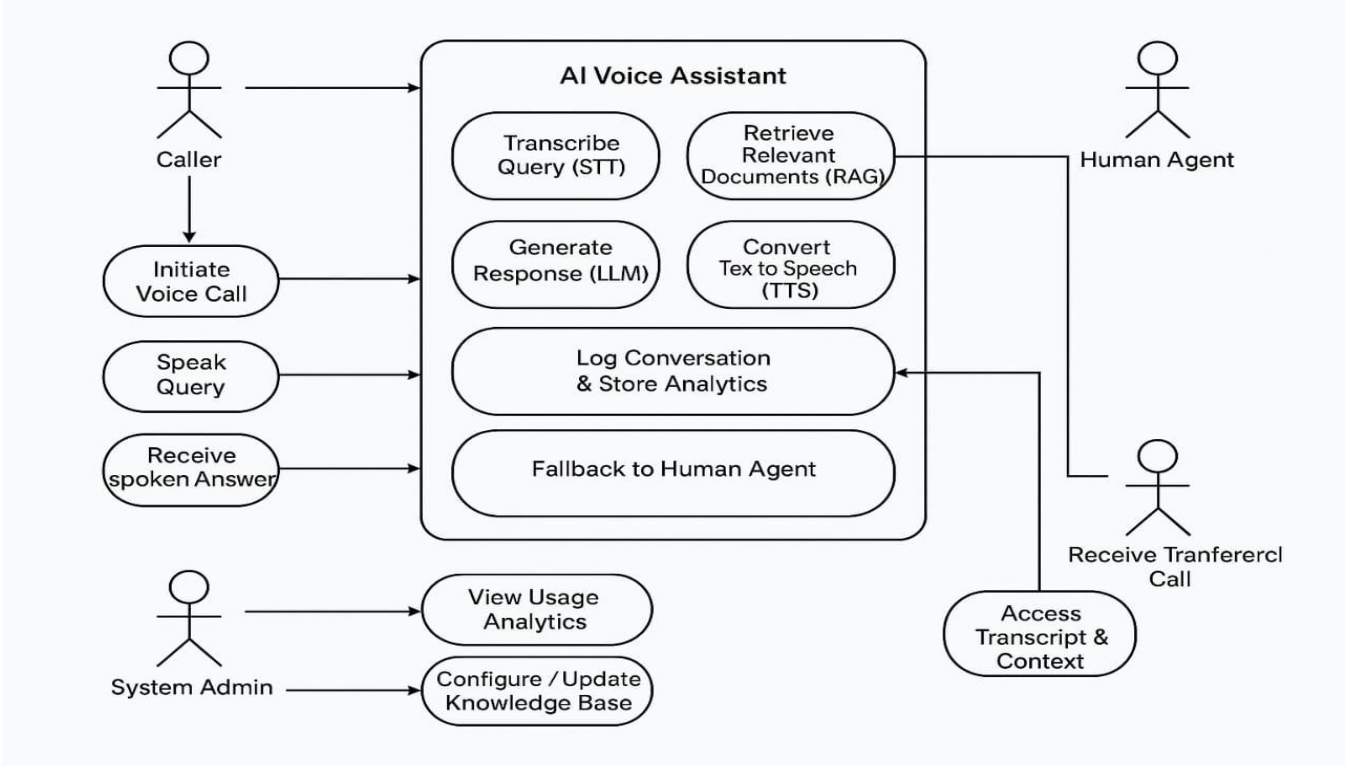

Fig: Use Case

- **Retrieval-Augmented Generation (RAG):**

  The query text is used to search relevant documents using RAG:

  - A document loader ingests multiple content formats (PDFs, JSON, URLs, etc.).
  - Documents are split into manageable chunks.
  - Each chunk is embedded into vector format using an embedding model.
  - These vectors are stored in a vector database (Pinecone, Weaviate, pgvector).
  - During runtime, relevant chunks are retrieved based on the user query.

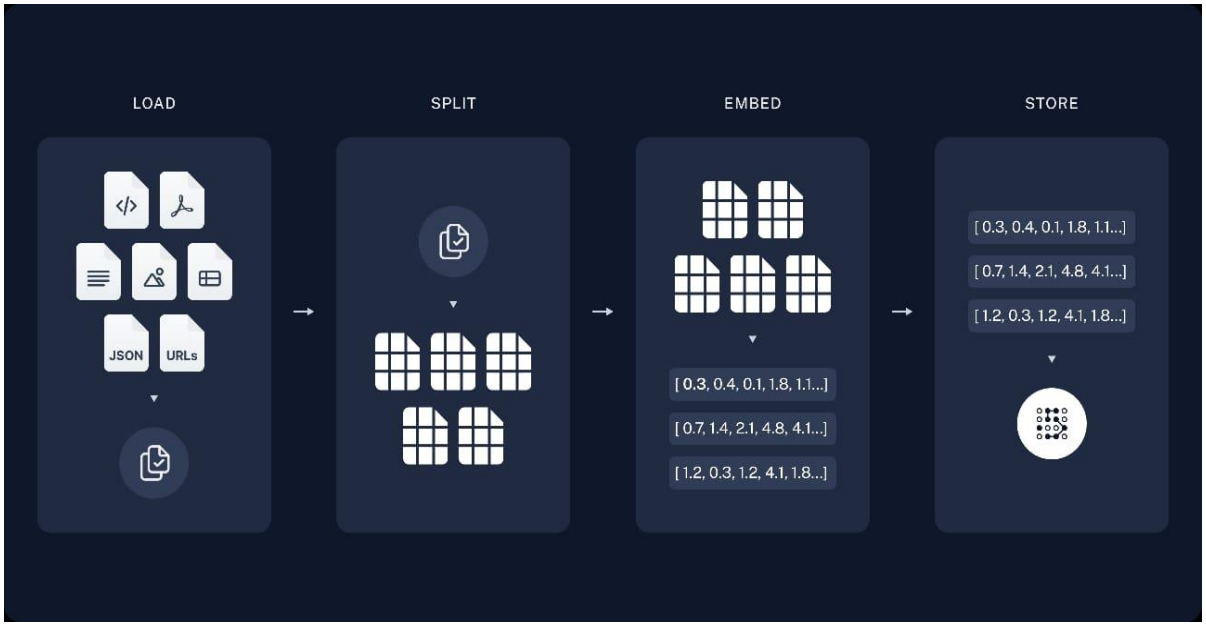

Fig : System workflow

Resources:

- **Software Requirements:**
  - Python 3.9+ : Primary development language
  - Vapi SDK/API : Call streaming and voice input/output integration
  - OpenAI Whisper (or API) : Converts speech to text
  - FAISS / ScaNN : For document retrieval (RAG)
  - OpenAI GPT / Local LLM (e.g., Mistral, LLaMA) : For generating context-aware replies
  - FastAPI / Flask : Backend framework for local API hosting
  - Text-to-Speech (TTS) Engines : pyttsx3 / gTTS / Coqui for converting responses to speech- SQLite /MongoDB : Optional, for logging and session tracking

- **Supporting Tools:**
  - Jupyter Notebooks : For testing and experimentation
  - Git + GitHub : Source code version control
  - Ngrok / LocalTunnel : For exposing local server to Vapi for webhook calls

- **Hardware Requirements :**
  - CPU: 4-core processor;
  - RAM: Minimum 8 GB;
  - Storage: Minimum 20 GB;.
  - Operating System: Windows

References and bibliography:

1) DataMiner Streamlit App – A tool for document processing and RAG-ready chunking Available at: https://dataminer.streamlit.app/ Accessed on: 22 July 2025.

2) Google AI – Gemini Speech Generation API Documentation for generating speech using Gemini API. Available at: https://ai.google.dev/gemini-api/docs/speech-generation .

3) S. Khan and M. Iqbal, "AI-Powered Customer Service: Does it Optimize Customer Experience?," *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2020.

4) Veturi, S., Vaichal, S., Jagadheesh, R. L., Tripto, N. I., & Yan, N. (2024). *RAG based Question-Answering for Contextual Response Prediction System*. 1st Workshop on GenAI and RAG Systems for Enterprise (CIKM 2024), Boise, Idaho, USA. arXiv:2409.03708. https://doi.org/10.48550/arXiv.2409.03708/