<p style="text-align:center">**Case Study**</p>

<p style="text-align:center">Samantha – A Voice AI Agent</p>



<p style="text-align:center">**Developer: Ian Baumeister**</p>

<p style="text-align:center">**Repository:** https://github.com/ibxibx/samantha-voice-ai-agent</p>

The year 2025 had been nicknamed as "the year of AI agents" and that's an exciting new direction the tech is taking, meanwhile the AI agents with their rich and effective functional power and usefulness is hard to overestimate. Big thanks to my friend and mentor Aemal Sayer for his methodical guidance and help grasp the principles, workflow, structure and help build a working Voice AI Agent.

Samantha Voice AI Agent is your virtual assistant that talks to you each time you call her, has access to your Google Drive, can brainstorm and check facts for you, create and save your ideas and thoughts in the cloud.

The entire AI Agent was built without writing any code, and the process contained a series of steps targeted to assemble all parts together, to test and verify the full functioning of all parts that constitute a working AI agent.

The AI agent can be called while cooking, driving, jogging, reading, or even having your bath – it is always by your side wherever you take your phone with you, your hands are busy, and you want to call Samantha. Additionally I could also enhance it and integrate in it for example my Google calendar, search the web, read news etc. The telephone number of the voice ai agent can be saved under a conventional name, in my case Samantha, and be called through voice command as for example with "OK, Google" etc: for example "Siri please call Samantha".

## Project Objectives

The Samantha Voice AI Agent project was developed with several key objectives:

**1. Create a fully functional voice assistant without writing code**

 - Demonstrate the power of no-code platforms for AI development
- Build a complete solution using only visual workflows and integrations

**2. Provide hands-free access to AI capabilities**
- Enable users to access AI assistance during activities where typing is impractical
- Create a natural voice interface that feels conversational and helpful

**3. Implement practical functionality**
- Allow users to brainstorm ideas and capture thoughts on the go
- Enable fact-checking and information retrieval through voice
- Create a system for automatically saving conversation content to Google Drive

4. Learn and demonstrate API integration skills

- Connect multiple third-party services into a cohesive solution

- Understand authentication flows and data passing between services

The project aimed to create not just a technical demonstration but a genuinely useful tool that showcases how modern AI capabilities can be harnessed by developers of all levels of seniority.

## Tech Stack and Specs

**n8n** - workflow automation platform that connects all components

**Vapi.ai** - voice AI platform for natural speech interface

**OpenAI API** - powers the intelligence with GPT-4o-mini

**Google Drive** - stores conversation summaries in the cloud

## Supported Platforms
**is activated through phone calls from any type of phone or internet**

## User Stories

⇒ **As a user, I want to be able to quickly start conversations with my voice agent while my hands are busy (for example while I'm driving, cooking, having a shower, cleaning the flat wtc.).**

⇒ **As a user, I want the ability to brainstorm, talk, check facts over a phone call with an agent who can give me quick and useful advice.**

⇒ **As a user, I want to be able to get fast consultation over the phone – also be able to save notes or the entire conversation in my Google drive, in order to be able to use this information later.**

⇒ **As a user, in situation of doubt, lack on information or ambiguity I want to be able to make a quick call and get a reliable and actual information / answer.**

## The structure of an AI Agent

The basic structure of an AI agent (pic.1) contains 3 parts: the Input, the Processing and the Output parts. In this particular diagram you can see that the Input is the trigger of the AI agent action that can be a text message or voice message, email etc.



**Pic.1**

The Processing part (is the heart of the AI agent) that consists of several parts that can be integrated into it:

- a chat model that contains an LLM behind it which could be any LLM API as for ex: Claude, ChatGPT,
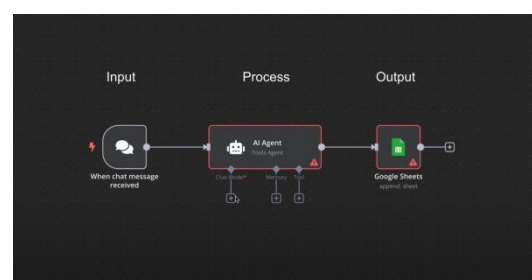
DeepSeek, Gemini, etc) that processes our natural language input that comes from the Input

- a memory – for remembering the conversation: for example if you send it a long text or message and ask it to summarize this text, it should be able to analyse it, keep in memory and execute the task over the information in its memory

- a tool – which has access to your tools and can execute tasks on your behalf: as for example write a Microsoft Word / or Excel document, access SalesForce, Microsoft 365 etc.

## Building Steps of the Workflow:

1. I've opened an n8n account, and then added a "new workflow", with option "personal" (pic.2).
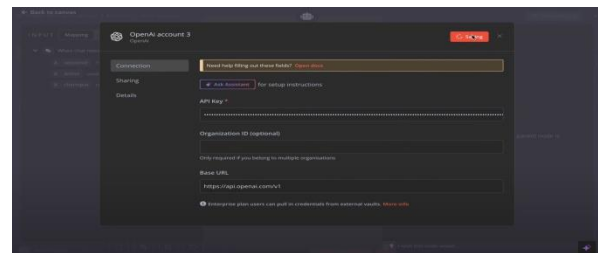The saved workflows on n8n that my Samantha AI Agent is,
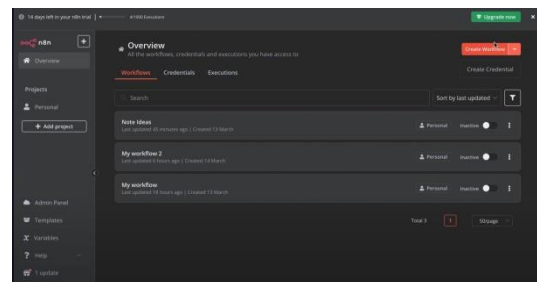


**Pic.2**

can be downloaded in the json format for personal backup, that can be added later to your other account.

2. Added the first node: the manual trigger: the Input which is a webhook with the "POST" http method.
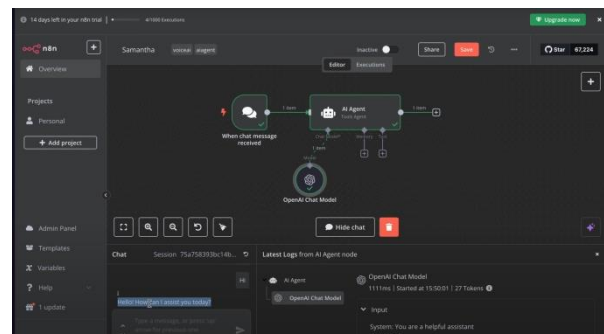Next to it I added the second functional node: the AI



**Pic.3**

Agent.

To the AI Agent in the "Chat Model" port I attached the OpenAi API key – as it's functional brain.

First I went and got my ChatGPT API secret key from my ChatGPT dashboard, copied and pasted it into the "Chat Model" port – in "create new credentials" and saved it. (pic.3)

3. Next, I tested the ChatGPT port if it is working right in the general diagram by clicking on it and writing "Hi" – and seeing how ChatGPT responds instantly in the same chat window via text "Hello, how can I assist you today" (pic.4)

4. The next step was going to Vapi.ai (a Silicon Valley based company building orchestration layers for any voice company) – created an "Assistant" and configured it with all the necessary specifications: voice of sarah, AI model: gpt-4o-mini, the first message "Hi, I'm Samantha, your virtual voice assistant, how are you today?" – and



**Pic.4**
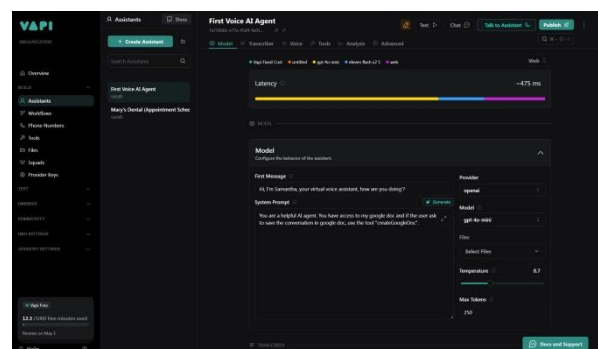


**Pic.5**

published it. (pic.5)

5. Next, in the Vapi.ai I went and configured a free Phone Number for testing purposes, I got a free US telephone number that I saved with my Assistant in it. It took over 2 minutes to activate the number before I could call it and talk with the "sarah" voice to test it.
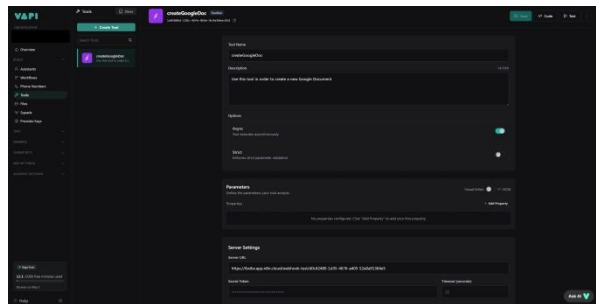
(pic.6)


**Pic.6**

6. In the next step I went and interfaced Vapi with n8n. For that I copied the URL from my first node Webhook, went to Vapi and in the Tools I created a function tool "CreateGoogleDoc" and pasted the Webhook Url into the Server URL and saved it.
How it works? This function will be called in "Tools", if I ask my voice assistant to go and save my content onto a Google Doc, it will go and do it for me by using this Tool. (pic.7)
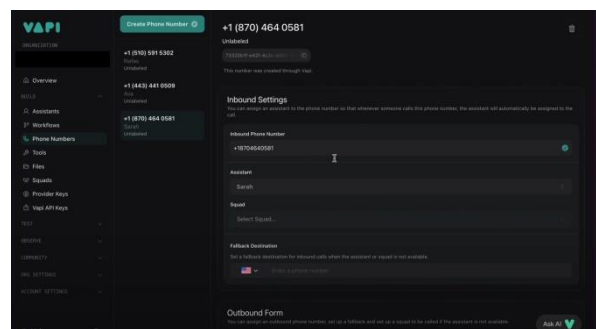

**Pic.7**

7. Then I went back to "Assistants" tab on Vapi, found "Sarah", and in the functions, tools, selected the "createGoogleDoc" and clicked publish. Deleted all the function widgets besides the "function-call" and published that. (pic.8)


**Pic.8**

8. The next step I went into my "AI Agent" node, opened it and went into its Parameters section, in the "Prompt field" added a note "Summarize this conversation between the AI agent and the human", clicked the option of "Expression" of this window and dragged from the Input area on the left into it the widget "messages". Then I edited the converted widget into the following code {`{{ JSON.stringify($json.body.message.artifact.messages) }}`} thus  directing the messages in form of strings towards ChatGPT. (pic.9)
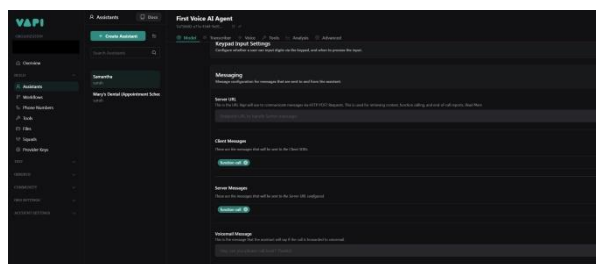

**Pic.9**

9. In the next step I had to create the function to take the generated output by the AI Agent node and save it onto a Google doc. For that I clicked on the "+" to add the "Google Docs – Create" node – and chose "Create a Document" option for its type, (pic.10) which will create Google documents for me, then in its Parameters I created new credentials.
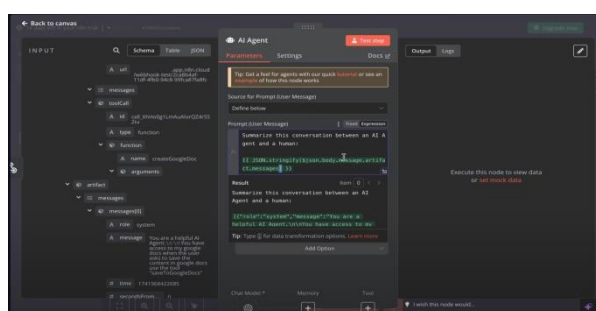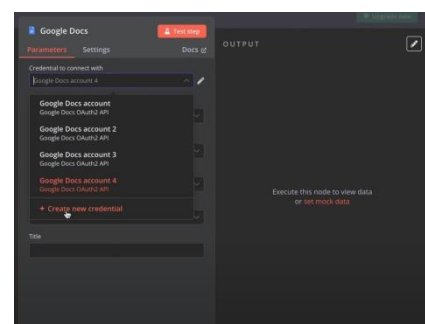

**Pic.10**

4

10. Then I went to the GCP Console and created a new Project "AIVoiceAgent" and filled all the API & Service fields which assure the integration of n8n with Google cloud (pic.11). Filled all the fields in this Project in API and Services and filled the Overview section with the needed data. In the "Clients"- created a Client, category web application, and put in the OAuth Redirect URL from n8n the Google Docs 4th node, inside the "Authorized redirect URIs. After that I got from the Google Console the Client

ID and Client secret codes, I took the Client ID and Client Secret codes and went into the n8n 3rd Doc node and pasted them in the respective fields. Then I went into the Audience dialog on the GCP and in the "Test Users" I added my Google user account email address and saved.
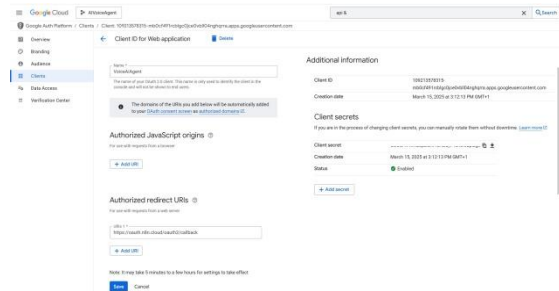
11. Then I went and logged in with Google from within the node 3 "Google Doc" in the Connection menu (pic.12) and got n8 connected to the Google cloud.
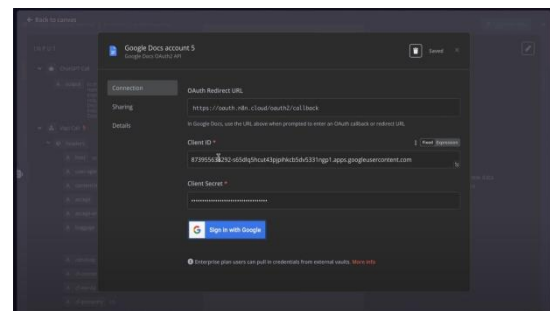


Pic.12

12. Then I went to the Google Docs API and clicked on the tab "Enable". The same concerns the Google Drive API too – clicked "Enable".

13. The I went onto my 3rd node and its menu, filled the fields with my Google drive account and folder address information where I want the created documents to be saved, gave the test document a name and clicked "Test Step"(pic.13) – and got the document created in my Google drive.
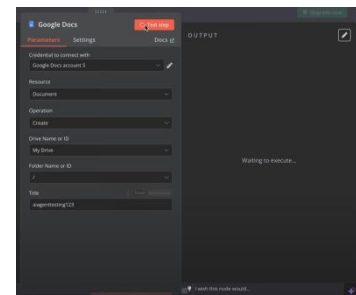


Pic.13

14. For the last step of the workflow creation I've created the 4th node "Google Docs - Update" – which updates my Google documents – I went into tis properties and dragged from the left column "input" towards its parameters the following: the "id" widget into the "Doc ID or URL" field, as well as the "output" into the "Text" field (pic.14). Then when I clicked to test the step, it updated the document.

Then I switched the toggle button above the canvas or diagram from Inactive to active, and I got the workflow activated, meaning in production mode.
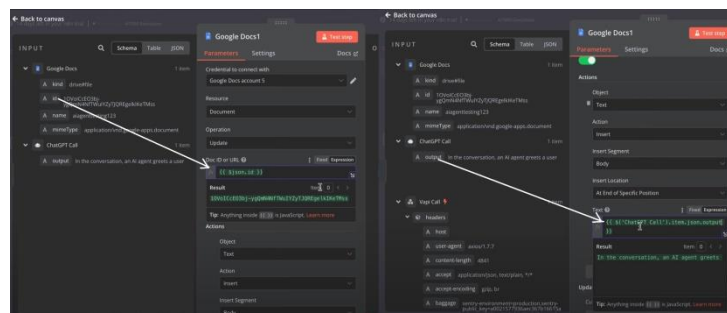


Pic.14

## Technical Challenges & Solutions

During the development of Samantha, I encountered several technical challenges that required creative problem-solving:

**Challenge 1**: Webhook Configuration

When setting up the webhook to receive data from Vapi.ai, I initially struggled with data formats and payload structures. The JSON data coming from the voice service didn't perfectly match what n8n expected.

**Solution:** I spent time analyzing the webhook payload structure using test calls and the n8n debugger. By examining the actual data structure, I was able to craft the correct expression {{ JSON.stringify($json.body.message.artifact.messages) }} to properly extract the conversation content.

**Challenge 2**: Google API Authentication

Setting up the OAuth flow for Google Drive integration proved complicated, with several configuration steps across different platforms.

**Solution:** I created a systematic approach, starting with project creation in GCP, enabling specific APIs, configuring OAuth consent screens, and carefully mapping redirect URLs. The step-by-step process documented in my case study ensured I didn't miss any critical authentication components.

**Challenge 3:** Voice Recognition Quality

Initial tests showed that Samantha sometimes misunderstood certain words or phrases, particularly in noisy environments.

**Solution:** I experimented with different voice models in Vapi.ai and found that the "Sarah" voice provided the best recognition accuracy. I also added prompt engineering in the assistant configuration to better handle ambiguous inputs and request clarification when uncertain.

**Challenge 4:** Workflow Reliability

Early versions of the workflow occasionally failed to complete all steps, particularly when saving to Google Drive.

**Solution:** I implemented better error handling in the n8n workflow and added conditional checks to verify that each step completed successfully before proceeding. This significantly improved the reliability of the end-to-end process.

These challenges highlighted the importance of systematic troubleshooting and the value of understanding the underlying APIs, even when working with no-code platforms.

## Skills Acquired

This project significantly expanded my technical skillset across multiple domains:

**API Integration:**

- Gained experience connecting multiple third-party APIs into a cohesive system

- Learned OAuth authentication flows and security best practices

- Developed skills in webhook configuration and payload handling AI & Natural Language

**Processing:**

- Learned prompt engineering techniques for conversational AI

- Developed understanding of LLM capabilities and limitations

- Gained experience in conversation summarization and content extraction

**No-Code Development:**

- Mastered n8n workflow creation and management

- Learned visual programming concepts and dataflow architecture

- Developed skills in expression-based data transformation

**Voice Interface Design:**

- Learned principles of conversation design for voice assistants

- Gained experience in voice model selection and configuration

- Developed understanding of speech recognition challenges and solutions

**Cloud Service Integration:**

- Gained hands-on experience with Google Cloud Platform

- Learned document creation and management via API

- Developed skills in cloud service authentication and authorization

**Project Documentation:**

- Improved technical writing and documentation skills

- Learned to create comprehensive case studies

- Developed skills in architectural diagram creation

These skills have significantly enhanced my capabilities as a developer and prepared me for more complex projects involving AI, automation, and integration work.

# Final thoughts and reflections

Building Samantha, the Voice AI Agent, was a rewarding journey that provided valuable insights into the world of AI agents and no-code development. Here are my key takeaways from this project:

The power of no-code platforms like n8n has truly democratized the development of sophisticated AI applications. Creating a functional voice agent without writing a single line of code demonstrates how accessible AI development has become in 2025.

Integrating various services (OpenAI, Vapi.ai, Google Drive) taught me the importance of understanding APIs and authentication flows. Each integration presented unique challenges that expanded my technical knowledge.

Voice AI represents a significant shift in human-computer interaction. Creating Samantha reinforced my belief that voice interfaces offer unprecedented accessibility and convenience, especially in situations where traditional interfaces are impractical.

The project revealed several areas for future enhancement. Adding context awareness, more sophisticated memory handling, and additional integrations with services like calendars and email would make Samantha even more useful as a personal assistant.

Throughout development, I gained a deeper appreciation for the AI Agents architecture, the node structure, the properties of nodes, the functioning of an AI agent and how all components interact with one another. Crafting appropriate responses, handling errors gracefully, and ensuring the conversation flows naturally required careful consideration.

This project has equipped me with valuable skills in AI agent architecture, webhook handling, and cloud service integration that I'll carry forward into future development work. I'm excited to continue refining Samantha and exploring new possibilities in the AI agent ecosystem.