ANANT Voice Assistant – Detailed Project Report

Name: Vaibhav Tiwari
Registration Number: 25BCE10251

1. Introduction

This project explores the integration of voice recognition, desktop automation, and conversational AI through a custom-built assistant named ANANT.
The solution demonstrates real-world application of Python libraries, system interaction, and modern LLM APIs to build a smart, voice-controlled AI assistant.

2. Problem Statement

Most existing voice assistants are platform-locked or require heavy frameworks.
ANANT solves the need for a lightweight, Python-based assistant capable of both desktop control and natural AI conversation using the Groq LLM API.

3. Project Objectives

1. Implement continuous speech recognition using microphone input.
2. Allow control of desktop applications like browser, file explorer, games, and music.
3. Provide conversational AI mode using Groq's GPT-based model.
4. Deliver responses through system TTS (SAPI).

4. Functional Requirements

1. Speech recognition via Google SR API.
2. Desktop Mode:
- Open apps (Valorant, Aimlabs, Hollow Knight)
- Open websites
- Open File Explorer & Browser
- Check time
- Play music (local or Spotify)
3. AI Mode:
- Natural conversation using Groq API
- Audio output using SAPI
4. Exit loop safety and fallback responses.

5. Non-Functional Requirements

- Performance: Low-latency recognition and API response.
- Usability: Simple voice-triggered workflow.
- Reliability: Handles recognition errors gracefully.
- Maintainability: Modular command structure allowing easy addition of new commands.
- Scalability: AI mode extendable with more tools or memory features.

6. System Architecture

The system consists of:
1. Speech Recognition Layer
2. Command Processing Layer
3. Desktop Automation Layer
4. AI Response Layer (Groq API)
5. Text-to-Speech Output Layer

7. Workflow Diagram Description

1. User speaks →
2. Speech captured →
3. Speech converted to text →
4. Text passed to command parser →
5. If desktop mode: execute system commands →
6. If AI mode: send text to Groq LLM →
7. Output spoken using SAPI

8. Implementation Details

Main Python modules used:

- speech_recognition
- win32com.client
- webbrowser
- os
- groq

The code uses a continuous loop with keyword detection ("Desktop Mode", "AI Mode").

9. Testing Approach

Manual testing through:

- Testing various speech inputs
- Validating command accuracy (opening apps/websites)
- Stress-testing AI mode with long queries
- Handling unrecognized inputs

10. Challenges Faced

- Background noise affecting recognition
- Maintaining continuous loop without crashes

- Ensuring fast API responses in AI mode

11. Future Enhancements

- Add wake word detection
- Add GUI dashboard
- Add offline recognition using Vosk
- Add home automation features

12. Conclusion

The ANANT assistant successfully demonstrates speech recognition, desktop automation, and AI-powered interaction in a single Python-based system.
It serves as a strong demonstration of applied programming and system integration skills.