**CS23501**

**OPERATING SYSTEMS**

**PROJECT REPORT**

AI-Powered-Shell

****

**AJAY M S** (2023503053)

**VIJAY ATHITHYAA G V** (2023503304)

**B.E CSE** (5/8)

**Department of Computer Technology,**

**MIT Campus, Anna University**

1. **INTRODUCTION**
   1. **Purpose**

* The purpose of this Software Requirements Specification (SRS) document is to define the objectives, scope, and functionality of the AI-Powered-Shell project. This project offers a voice-controlled Linux terminal that allows users to execute shell commands using Tamil or English speech.
* The system captures the user’s voice, transcribes it with Sarvam AI, interprets it with Gemini AI, and securely runs the corresponding command in Windows Subsystem for Linux (WSL).
* This document acts as a guide for developers, testers, and stakeholders to understand the software’s features, limitations, and intended behavior.
  1. **Scope**
* The **AI-Powered-Shell** aims to simplify Linux command execution by integrating speech recognition, translation, and automation in a unified interface.

It provides:

* Hands-free terminal interaction using natural voice commands.
* Real-time translation from Tamil to English.
* Automatic command generation using Gemini AI.
* Safe and controlled command execution in WSL.
* Visual feedback through a Rich-based Text User Interface (TUI).
* The project is especially useful for:
* Developers or system users seeking hands-free terminal operation.
* Tamil-speaking users who prefer using their native language for commands.
* Accessibility-focused users who rely on voice interaction.
  1. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **WSL** | Windows Subsystem for Linux |
| **TUI** | Text-based User Interface |
| **STT** | Speech-To-Text |
| **TTS** | Text-To-Speech |
| **Sarvam AI** | An AI based platform used for speech transcription. |

1. **OVERALL DESCRIPTION**

**2.1 Product Description**

* The **AI-Powered-Shell** is an intelligent extension built on top of the traditional Linux terminal. It combines voice recognition, natural language understanding, and secure command execution to deliver a **hands-free terminal experience**.  
  The system records the user’s voice input, transcribes it into text using **Sarvam AI**, translates it into English with **Gemini AI**, and generates a corresponding **non-interactive Linux command**.  
  This command is safely executed in **Windows Subsystem for Linux (WSL)**.  
  A text-based graphical interface (TUI) developed with the **Rich library** displays all interactions — including transcript, generated command, output, and status.
* The system consists of two main Python modules:
* **voice\_shell\_tunglish.py:** Core logic handling speech recognition, translation, command generation, and WSL execution.
* **terminal\_gui.py:** Provides the Rich-based TUI to visualize the entire workflow and user interaction in real time.

**2.2 Product Functions**

* The major functional components of **AI-Powered-Shell** are as follows:

1. **Voice Recording:** Continuously listens to the user’s speech and stops automatically after detecting silence.
2. **Speech Transcription:** Converts Tamil audio into text using Sarvam AI (or Whisper as fallback).
3. **Translation & Command Generation:** Translates the transcribed text into English and generates a single, safe Linux command using Gemini AI.
4. **Command Execution:** Executes the generated command securely within WSL, blocking unsafe or interactive commands.
5. **Output Display & Feedback:** Presents command results and status updates through the TUI, and provides spoken feedback using text-to-speech.
6. **Error Handling:** Manages exceptions such as network errors, command failures, and unsafe command detection gracefully.

**2.3 User Characteristics**

* The primary users of the system include:
* **Developers and Students** who wish to interact with the Linux terminal using their voice.
* **Tamil-speaking users** who prefer issuing commands in their native or mixed Tamil-English language.
* **Accessibility-focused users** who benefit from a voice-driven interface instead of keyboard-based interaction.

**2.4 Constraints**

* The **AI-Powered-Shell** operates within the following constraints:
* Requires **stable internet connectivity** for Sarvam AI and Gemini API.
* Depends on **microphone input** and **audio drivers** functioning correctly.
* Must have **WSL** and **Python 3.10+** installed with all dependencies (rich, sounddevice, pyttsx3, etc.).
* Cannot execute **interactive commands** like nano, vi, top, or man.
* Execution timeout is set to **10 seconds** to prevent hanging processes.
* The system is designed primarily for **Windows 10/11** environments with WSL support.

1. **SYSTEM FEATURES**

**3.1 Voice Command Recognition**

* The **Voice Command Recognition** feature enables users to interact with the shell hands-free by speaking commands in **Tamil.**  
  The system uses the **sounddevice** library to record audio at 16 kHz and automatically stops recording after detecting silence.  
  A small buffer (pre-roll and tail frames) ensures that no part of the speech is missed.  
  The recorded voice is stored as input.wav, which becomes the input for the transcription module.
* **Functional Requirements:**
* The system must start recording when the user begins speaking.
* The recording must automatically stop after silence is detected.
* Audio should be stored locally in WAV format with minimal latency.

**3.2 Speech Transcription**

* Once audio is recorded, it is converted into text using a **Speech-to-Text (STT)** engine.  
  By default, **Sarvam AI’s “saarika:v2.5” model** is used for Tamil speech recognition.  
  As a fallback, **Whisper** can be employed for offline transcription if network access is unavailable.
* **Functional Requirements:**
* The system must transcribe the user’s voice into text with reasonable accuracy.
* The transcription should handle Tamil and mixed Tamil-English sentences.
* Errors due to connectivity or API failure must be caught and reported gracefully.