PROJECT NAME: RUSTY- Responsive Unified System for Tasks & Yielding

GitHub Repository: RUSTY , Full repository private — available upon request

1. Project Overview

Rusty is a context-aware, voice-driven AI assistant designed to support users with ADHD and executive dysfunction. It blends multi-threaded response handling, voice interaction, adaptive memory, and strategic modeling inspired by AI principles to provide helpful and non-intrusive support during daily tasks.

---

2. Core Features

- **Real-time voice interaction (Whisper + Edge-TTS)**
- **Short-term, long-term, and episodic memory modules**
- **Multi-threaded response system with priority queuing**
- **Google Calendar integration (WIP)**
- **Custom intent recognition and learning**
- **QGT-inspired behavioral strategy modeling (planned)**

---

3. Project motivation

Rusty was born out of frustration with traditional assistants that aren't designed for neurodiverse users. The project prioritizes empathy, user context, privacy, and adaptability — not just command-response logic. This makes Rusty particularly useful for individuals who struggle with task initiation, context-switching, or executive overload.

4. Architecture / Modules

```
rusty_core/
├── app_manager
          ├── app_control.py
          ├── app_scanner.py
          ├── installed_apps.json
├── main.py
├── voice_engine.py - Handles STT + TTS with buffer and silence detection
├── intent_engine- Routes inputs using Gemini NLP or hand-mapped logic
          ├── detect_intent.py
          ├── gemini_intent.py
          ├── intent_map.py
          ├── intent_router.py
├── config.py
├── gpt_conversation.py
├── style_learning.py - Personalized behavior adaptation based on
interaction style
├── memory - Stores and manages memory modules (short-term, long-term,
episodic)
          ├── memory.py
          ├── episodic_memory.json
          ├── long_term_memory.json
          ├── rusty_context.json
          ├── user_command_map.json
├── system_control.py
├── spotify_control.py
├── helper_calendar.py
```

5. Tech Stack

- Python 3.10+
- faster-whisper, edge-tts, webrtcvad, sounddevice
- Google Calendar API
- Gemini 1.5 API (for fallback NLP)
- llama-cpp-python (for local models like TinyLlama)

---

6. Installations / Usage

pip install -r requirements.txt

python main.py

---

7. Vision / Future Roadmap

- **Better memory context chaining**
- **Behavioral feedback loops**
- **Emotion-sensitive response handling**
- **Mobile wrapper / GUI overlay**
- **Local LLM integration beyond fallback mode**
- **ADHD-specific strategy profiles**

---

8. License / Credits / Notes

MIT License

Built with curiosity, trial-and-error, and an ongoing desire to make AI feel more like a friend than a tool.