Elder Care Monitoring Rover with Voice Assistant - Project Plan

# 1. Project Summary

This project plan outlines the detailed steps to build the Elder Care Monitoring Rover with Voice Assistant. The team consists of a single developer with 2 days/week access to the physical robot. Some tasks will be built and tested locally on a laptop before integration.

# 2. Project Structure

elder\_care\_rover/

│ ├── main.py # Main application entry point

│ ├── face\_recognition/ # Facial recognition module

│ │ ├── capture.py

│ │ ├── recognize.py

│ │ └── train\_model.py

│ ├── voice\_assistant/ # Voice assistant module

│ │ ├── speech\_recognition.py

│ │ ├── tts.py

│ │ └── wake\_word.py

│ ├── reminders/ # Reminder and scheduler module

│ │ ├── medication\_reminder.py

│ │ ├── sleep\_alarm.py

│ │ └── custom\_reminder.py

│ ├── emergency\_call.py # Emergency call module

│ ├── notifications.py # Discord/SMS notification module

│ └── config.py # Configuration file

# 3. Project Timeline

|  |  |  |
| --- | --- | --- |
| Task | Description | Time |
| Project Setup & Local Development Environment | Set up Python environment, install necessary libraries (dlib, OpenCV, Vosk, etc.) on local machine. | Week 1 |
| Facial Recognition Module (Local) | Develop and test facial recognition on laptop using webcam feed; build dataset and train model. | Week 2-3 |
| Speech Recognition Module (Local) | Develop Vosk-based speech recognition module with test commands. | Week 4 |
| Wake Word Detection (Local) | Implement Picovoice Porcupine for wake word activation. | Week 5 |
| Text-to-Speech (Local) | Implement eSpeak for offline voice output. | Week 6 |
| Emergency Call Module (Local) | Write Python module to trigger calls via connected GSM module or VOIP (simulated locally). | Week 7 |
| Medication & Custom Reminders (Local) | Develop reminder modules for medication and custom reminders with 'got that' confirmation. | Week 8 |
| Sleep Alarm System (Local) | Develop sleep alarm feature with voice prompt. | Week 9 |
| Hardware Integration & Testing | Deploy local modules to Raspberry Pi, test with physical robot during lab access. | Week 10-11 |
| Full System Integration & Testing | Combine all modules and test end-to-end functionality. | Week 12 |
| Documentation & Report | Finalize project documentation and demonstration. | Week 13 |