Requirements Document

REQ-1: ref: refs/heads/main

REQ-2: [core]  
 repositoryformatversion = 0  
 filemode = true  
 bare = false  
 logallrefupdates = true  
[remote "origin"]  
 url = https://github.com/deploymentqorix/RT\_SLAM\_qorix\_incubDeployment.git  
 fetch =...

REQ-3: Unnamed repository; edit this file 'description' to name the repository.

REQ-4: DIRC<h¢ñ"<qzh¢ñ"<qzN¤èè×øa2È42êû»ÜüØg>pÁÇ.github/workflows/npm-gulp.ymlh¢ñ"Gnh¢ñ"GnN¤èè0ÅÀjYiHÜjÑ±}ë&tEâ'LICENSEh¢ñ"Gnh¢ñ"GnN¤èèë°"]5)ÅôWõÅ¤¤ëý3aõMakefileh¢ñ"Gnh¢ñ"GnN¤èè.G!f[rHÍqò...

REQ-5: git ls-files --others --exclude-from=.git/info/exclude  
Lines that start with '#' are comments.  
For a project mostly in C, the following would be a good set of  
exclude patterns (uncomment them if you...

REQ-6: 0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

REQ-7: 0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

REQ-8: 0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

REQ-9: pack-refs with: peeled fully-peeled sorted

REQ-10: 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab

REQ-11: ref: refs/remotes/origin/main

REQ-12: MIT License  
  
Copyright (c) 2025 deploymentqorix  
  
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to...

REQ-13: The main Makefile for the entire project

REQ-14: from fastapi import FastAPI  
from fastapi.middleware.cors import CORSMiddleware  
  
app = FastAPI()  
  
app.add\_middleware(  
 CORSMiddleware,  
 allow\_origins=["\*"],  
 allow\_credentials=True,  
 ...

REQ-15: import rclpy  
from rclpy.node import Node  
from nav\_msgs.msg import Odometry  
from nav\_msgs.msg import OccupancyGrid  
import asyncio  
import websockets  
import json  
  
CONNECTED\_CLIENTS = set()  
  
async def...

REQ-16: import rclpy  
from rclpy.node import Node  
from geometry\_msgs.msg import PoseStamped  
  
class PoseSubscriber(Node):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_('pose\_subscriber\_for\_streamer')  
 ...

REQ-17: File: backend/dev\_scripts/mock\_api.py  
Description: A standalone FastAPI server to provide mock data for frontend development.  
To run: python3 mock\_api.py

REQ-18: Copyright 2016-2019 Dirk Thomas  
Licensed under the Apache License, Version 2.0

REQ-19: Copyright 2016-2019 Dirk Thomas  
Licensed under the Apache License, Version 2.0

REQ-20: Python packages for the backend services

REQ-21: cmake\_minimum\_required(VERSION 3.8)  
project(slam\_node)  
  
# Find all dependencies  
find\_package(ament\_cmake REQUIRED)  
find\_package(rclcpp REQUIRED)  
find\_package(geometry\_msgs...

REQ-22: include <Eigen/Dense>  
include <vector>  
include "rclcpp/rclcpp.hpp"  
include "geometry\_msgs/msg/pose\_stamped.hpp"  
include "sensor\_msgs/msg/imu.hpp"  
include "sensor\_msgs/msg/laser\_scan.hpp"  
include...

REQ-23: REST API Endpoints  
# Get System Health

REQ-24: Message for Live Pose Updates

REQ-25: Stage 1: Build the React application

REQ-26: Getting Started with Create React App

REQ-27: https://www.robotstxt.org/robotstxt.html

REQ-28: import React, { useState, useEffect } from 'react';  
import { connectWebSocket, disconnectWebSocket } from './services/websocketService';  
import MapView from './components/MapView';  
import...

REQ-29: import { render, screen } from '@testing-library/react';  
import App from './App';  
  
test('renders learn react link', () => {  
 render(<App />);  
 const linkElement = screen.getByText(/learn react/i);  
...

REQ-30: import React, { useRef, useEffect } from 'react';  
  
const MapView = ({ pose, trail, mapData }) => {  
 const canvasRef = useRef(null);  
  
 useEffect(() => {  
 const canvas = canvasRef.current;  
 ...

REQ-31: import React, { useState, useEffect } from 'react';  
import { fetchSystemHealth } from '../services/apiService';  
  
const SystemHealthPanel = () => {  
 const [health, setHealth] = useState({  
 ...

REQ-32: import React from 'react';  
import ReactDOM from 'react-dom/client';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
const root =...

REQ-33: const reportWebVitals = onPerfEntry => {  
 if (onPerfEntry && onPerfEntry instanceof Function) {  
 import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {  
 ...

REQ-34: const API\_URL = "http://localhost:8000";  
const MOCK\_MODE = true; // Uses mock data from websocketService  
  
export const fetchSystemHealth = async () => {  
 if (MOCK\_MODE) {  
 // --- MOCK MODE:...

REQ-35: const MOCK\_MODE = false;  
const WEBSOCKET\_URL = "ws://localhost:3000/ws";  
  
let socket = null;  
let mockInterval = null;  
  
export const connectWebSocket = (onMessageCallback) => {  
 if (MOCK\_MODE) {  
 ...

REQ-36: jest-dom adds custom jest matchers for asserting on DOM nodes.  
allows you to do things like:  
expect(element).toHaveTextContent(/react/i)  
learn more: https://github.com/testing-library/jest-dom

REQ-37: REST API Endpoints  
# Get System Health

REQ-38: Message for Live Pose Updates