Overview

Design and implement a robust, server-intensive, and highly optimized database-driven control system for an autonomous wall-finishing robot. The system must handle intensive computations for intelligent path planning, real-time communication through message brokers, detailed logging and monitoring, and sophisticated visualizations.

Task 1: BE Intern Assignment

Objectives:

- **Coverage Planning**:
    - Implement basic coverage planning logic for rectangular walls.
    - Allow users to input custom dimensions and rectangular obstacles.
- **Backend Data Management**:
    - Set up a simple API using Python (FastAPI).
    - Use SQLite database for trajectory storage with basic indexing. Add as much optimization for real world use case for this.
    - Implement basic query APIs to retrieve stored trajectory data.
    - Include logging for request handling and response timing.
    - We don’t mind you going overkill, as long as you can understand the overkill.
- **Frontend Visualization**:
    - Provide a basic web-based 2D visualization without using Matplotlib.
    - Make sure it intelligently explains the path planning
    - Implement simple trajectory playback.
- **Testing**:
    - Write basic API tests using tools like pytest or FastAPI's built-in TestClient.
    - Test API endpoints for basic CRUD operations and validate response times.

Technical Requirements:

- Programming Language: Python (FastAPI)
- Database: SQLite (with basic indexing)

Sample Case:

- Wall dimensions and obstacles should be user-input parameters.
- Example provided:
    - Wall: 5m x 5m
    - Obstacle: Window (25cm x 25cm)

Note: Use of AI tools to complete the assignment is discouraged, vibe coding of the entire assignment is frowned upon, intelligent use of AI tools to get ideas and solve the same is encouraged.

Submission : Make sure it is a private github repository , add tanay@10xconstruction.ai and tushar@10xconstruction.ai as collaborators. Attach a small video walkthrough of you assignment explaining the same , and your approach in the github readme or as a drive link.