WaterLevel Requirements Document

# 1. Problem & Context

## Pain:

Operators of angle-controlled robots currently lack a unified, real-time, visual dashboard to track the robot’s actual vs. ideal path, predict future arrival points, and adjust movement to stay within tolerance. Without such a tool, adjustments are reactive and inefficient, leading to off-path arrivals and wasted operation time.

## Who:

Industrial operators, field engineers, or technicians responsible for guiding robots along precise paths between two points, with defined X/Y tolerance limits.

## Why now:

The target customer is piloting robots that navigate based on continuous X/Y angle inputs and requires a reliable, mobile-friendly tool for monitoring and control before scaling deployment.

## Assumptions:

* UART connection is reliable and follows a fixed ASCII format with known delimiters.
* Operators have at least intermittent network access to view the dashboard.
* Initial use case involves ≤ 100 concurrent users with a few admins, oprators and mostly read-only viewers.
* No sensitive data is transmitted, but encryption is still required for integrity and anti-tampering.

# 2. Goals & Success Metrics

## Primary Goals:

1. Role-based access control
2. Allow certain users to configure UART parameters and job settings.
3. Demonstrate accurate, real-time plotting of robot’s actual vs. ideal path.
4. Provide projected arrival point visualization based on current movement angles.

## Guardrails:

1. End-to-end latency from UART input to updated graph ≤ 1s in POC environment.
2. UI with large, touch-friendly control buttons.
3. Web access to the application.

# 3. Users & JTBD (Job to be Done)

## Primary User: Robot operator monitoring and adjusting movement trajectory.

JTBD: “When guiding the robot along a path, I want to see its actual position, deviation from the ideal path, and projected arrival point, so I can make timely angle adjustments to stay within tolerance.”

## Won’t-serve:

1. Users requiring full manual robot control via the web app.
2. Third-party developers needing raw UART data access from frontend.

# 4. Scope (This Release — POC)

## Must:

1. Role-based access (admin/viewer/operator).
2. Backend UART handling (ASCII, comma-delimited, LF-terminated messages).
3. Frontend view of raw UART data for troubleshooting/verification.
4. No direct communication with UART from front-end, only through the backend.
5. Audit log for user actions and received UART data.
6. Real-time graph plotting actual vs. ideal path with projected arrival point and tolerance visualization.

## Should:

1. Printable audit reports.
2. Basic menu & image-based button interface.
3. Send commands to UART (like start/stop/pause/resume): TBD.

## Won’t/Non-goals:

1. Direct robot control from frontend.
2. Multi-language support.
3. Offline mode.