IoT Telemetry Simulation Prototype

Objective:

To demonstrate how lightweight IoT devices can generate and transmit real-time telemetry to a central server. The prototype focused on streaming simulated location data (lat/long) to illustrate distributed communication, data storage, and visualization.

Architecture Overview:

- **Device Side (Rust):**
  - Simulated IoT nodes generate pseudo-random telemetry (e.g., latitude/longitude, timestamp).
  - Each node publishes messages via MQTT.
- **Server Side (Bun + Elysia):**
  - MQTT broker receives messages from devices.
  - Server parses incoming telemetry and stores it in PostgreSQL.
  - Exposes APIs for querying historical and live data.
- **Frontend (VueJs):**
  - Real-time dashboard subscribing to updates.
  - Map-style visualization of device movement.

Key Features:

- Event-driven MQTT communication between Rust devices and server.
- Simulated mobility data without dependency on physical GPS hardware.
- Web-based visualization showing telemetry flow from devices to server.

Outcome:

A working end-to-end prototype showcasing IoT telemetry pipelines. The project highlighted how resource-constrained clients (in Rust) can efficiently publish to a modern backend stack, and how such telemetry can be stored, processed, and visualized in real time.

Applications (Conceptual):

- Smart transport monitoring
- Sensor-driven infrastructure
- Real-time IoT dashboards