IoT Meeting Room Occupancy and Booking Dashboard

ESP32 room nodes with PIR and door sensing feed a booking dashboard that shows live occupancy, detects no-shows, and drives e-paper door signs.

Title

IoT Meeting Room Occupancy and Booking Dashboard — sensor-verified room booking with utilization analytics.

Introduction / Background

Meeting rooms are booked but sit empty: someone reserves for an hour, the meeting ends in twenty minutes, and nobody releases the room. No-show bookings waste the scarcest office resource while teams hunt for space. The root problem is that booking systems track reservations, not reality. This project adds ground truth — an ESP32 node per room with a PIR motion sensor and a door reed switch — and builds the booking layer on top: a web dashboard showing live FREE/BOOKED/IN-USE status, automatic release of no-show bookings after a grace period, e-paper door signs that update themselves, and weekly utilization analytics.

It is an applied IoT + web build: embedded sensing, MQTT messaging, a booking server and a real dashboard.

Problem Statement

Without occupancy sensing, booking data is fiction: utilization reports lie, no-shows block rooms for hours, and door signs go stale. A useful system must sense actual presence, reconcile it with bookings in real time, free rooms nobody is using, keep door signage truthful without human effort, and prove the savings with utilization numbers.

Objectives

- Sense per-room occupancy with PIR motion + door reed switch on an ESP32 node.
- Maintain live room state (FREE / BOOKED / IN-USE) reconciled against bookings.
- Auto-release no-show bookings after a 15-minute grace period and log the event.
- Drive e-paper door signs showing current status and next booking.
- Serve a dashboard with room cards, booked-vs-occupied analytics and an event log; allow web booking.

Proposed Methodology

Each room node runs an occupancy state machine: PIR motion events and door open/close transitions feed a presence decision with timeouts (a room with recent motion is IN-USE; a booked room with no motion past the grace period is a NO-SHOW). Nodes publish state over MQTT/WiFi to a booking server that holds the reservation table. The server reconciles sensed state with bookings every few seconds: no-shows are released and logged, early departures free the remaining slot, walk-ins are recorded. E-paper displays poll the server and redraw only on change (their key power advantage). The dashboard renders room cards, a booked-vs-occupied weekly chart with utilization percentages, and the event log; a booking form writes directly to the

reservation table with conflict checks.

System Architecture / Working

Per-room node: ESP32 + HC-SR501 PIR + reed switch + e-paper display. Messaging: MQTT broker (room/telemetry, room/command topics). Server: booking logic, reconciliation engine, web dashboard + booking UI, REST/MQTT bridge. Dashboard: room status cards, analytics chart, no-show/release log, booking form. Demo runs 3 rooms on one broker.

Hardware / Software Requirements

- Hardware: ESP32 dev boards (one per room; single-node demo acceptable), HC-SR501 PIR sensors, reed switch + magnet per door, e-paper display module, USB power per node.
- Software: Arduino IDE (nodes), MQTT broker (Mosquitto), booking server in Python/Node, web dashboard (HTML/CSS/JS).

Expected Outcomes

A working 3-room demo: live occupancy status, no-show auto-release at design-target within 1 minute after the 15-minute grace, e-paper signs updating on state change, weekly booked-vs-occupied chart with utilization percentages, and web booking with conflict detection. Timing targets are firmware/server design values verified by the buyer during testing.

Applications

- Office meeting-room management.
- College seminar-hall and lab booking.
- Library study-room allocation.
- Co-working space utilization analytics.

Future Scope

- Calendar sync with Google/Outlook.
- QR check-in at the door.
- People counting with time-of-flight sensors.
- HVAC/lighting integration driven by occupancy.