PROJECT ABSTRACT

# IoT Parking Slot Occupancy Monitor with Live Availability Dashboard

*An ESP32-based parking monitor with an IR sensor per bay, MQTT telemetry, a live availability dashboard and an entry guidance display, demonstrated on a 24-bay model. Delivered built-to-order with prototype, firmware, dashboard, report, PPT and viva kit.*

## 01. Title

IoT Parking Slot Occupancy Monitor with Live Availability Dashboard — a hybrid IoT project combining a 24-bay sensor instrumented parking model with a real-time web dashboard and entry display.

## 02. Introduction / Background

Drivers in malls, campuses and office basements lose time circling for a free slot while parking attendants can only guess at availability. Automated parking guidance exists in large commercial installations, but the sensing-to-display pipeline is rarely built end-to-end by students. This project implements the complete loop at demonstration scale: an IR reflective sensor on each of 24 bays across two levels, ESP32 nodes publishing bay states over Wi-Fi via MQTT, a web dashboard rendering the live availability map with occupancy analytics, and a 20x4 LCD at the model entrance guiding drivers to free bays.

## 03. Problem Statement

The project addresses the lack of real-time bay-level visibility in parking facilities. Without per-bay sensing, guidance displays cannot be trusted and occupancy analytics are impossible. The build demonstrates reliable per-bay presence detection with debounce, sub-5-second state propagation to the dashboard, and an entry display that agrees with the dashboard — the three properties a real deployment needs.

## 04. Objectives

- Detect vehicle presence on each of 24 bays with adjustable IR reflective sensors and firmware debouncing.
- Publish bay-state changes from ESP32 nodes over Wi-Fi via MQTT with a design-target latency under 5 seconds.
- Render a live bay-availability map, occupancy percentage and hourly trends on a web dashboard.
- Drive a 20x4 LCD entry display showing live free-bay counts per level.
- Compute occupancy analytics: peak hours, average dwell time and daily utilization.
- Provide a per-sensor threshold calibration procedure stored in EEPROM.

## 05. Proposed Methodology

Each bay's IR sensor is aimed at the parking surface and its threshold calibrated with a vehicle present and absent. Firmware samples the sensors, applies a debounce window to reject transient shadows, and publishes state changes as JSON over MQTT. A backend stores timestamped

events; the dashboard re-renders the bay map and aggregates hourly trends, peak occupancy and dwell time. The entry LCD polls the same state store on a 5-second cycle. Alerts fire when a level crosses 90% occupancy or a sensor goes silent.

## 06. System Architecture / Working

Two ESP32 nodes cover the 24 bays (12 each), publishing to an MQTT broker. The dashboard backend subscribes, persists events and serves the live map, trends and analytics views. The entry LCD is driven by a third lightweight poll of the state store. Power is 5 V regulated distribution across the model; wiring runs through trunking on the two-level bay model for a realistic installation look.

## 07. Hardware / Software Requirements

- Hardware: 2x ESP32 development boards, 24x adjustable IR reflective sensor modules, 20x4 LCD with I2C backpack, 5 V regulated power distribution, scaled two-level parking bay model with wiring trunking, buzzer and status LEDs.
- Software: Arduino IDE (C/C++ firmware), MQTT broker, web dashboard (HTML/CSS/JS), EEPROM threshold storage.

## 08. Expected Outcomes

- A working 24-bay parking model with live per-bay occupancy sensing.
- A web dashboard showing the real-time bay map, occupancy trends and analytics.
- An entry LCD displaying live free-bay counts per level.
- Sub-5-second design-target latency from bay change to dashboard update.
- Complete documentation: wiring diagrams, calibration procedure, report, PPT and viva Q&A.

## 09. Applications

Parking guidance for malls, campuses and office basements; the sensing and MQTT telemetry pattern transfers to any per-slot occupancy problem such as warehouse dock monitoring or EV charging-bay availability.

## 10. Future Scope

Number-plate recognition per bay, reservation of bays from a mobile app, payment integration, scaling the node architecture to multi-level real facilities, and LoRa backhaul for outdoor lots without Wi-Fi.