IoT-Based Smart City Framework: An Integrated Seven-Module Approach Using Arduino UNO for Urban Automation and Resource Management

Onkar Balaji Gawali[1], Jeevan Ishawar Band[2], Sarthak Satish Phalke[3], Sahil Navnath Phalke[4], Naitik Soni[5], Devakinandan Shete[6]

[1,2,3,4,5]*Trainee, Department of IoT Network Specialist, MIDC Skill Development Centre, Pune, India*

[6]*Trainer, Department of IoT Network Specialist, MIDC Skill Development Centre, Pune, India*

***Abstract*—Rapid urbanization has placed increasing strain on civic infrastructure, producing traffic congestion, uncontrolled energy consumption, water wastage, rising air pollution, and inefficient parking utilization. Conventional municipal systems rely heavily on fixed timers and manual supervision, which cannot adapt to real-time urban conditions. This paper proposes an integrated Smart City Framework built entirely around low-cost Arduino UNO microcontrollers that unifies seven independent automation modules smart home automation, smart traffic signal control, smart irrigation, smart weather/air-quality monitoring, smart parking management, vehicle speed detection, and smart street lighting into a single, modular urban automation testbed. The problem addressed is the absence of a low-cost, easily replicable prototype that demonstrates how sensor-actuator automation can simultaneously target energy optimization, water conservation, dynamic traffic flow, environmental monitoring, safe parking allocation, and public-safety enforcement without dependence on expensive SCADA or proprietary smart-city platforms.**

The proposed system follows a layered sensor–controller–actuator architecture. Each module uses passive infrared (PIR), infrared (IR), soil-moisture, rain, MQ135 gas, DHT11, and light-dependent-resistor (LDR) sensors interfaced to Arduino UNO boards, which drive relays, servo motors, and LED/LCD outputs to automate physical responses without human intervention. Embedded C++ firmware developed in the Arduino IDE governs decision logic for each subsystem, and a 16x2 LCD provides local telemetry. Bench-level testing of all seven modules confirmed reliable operation: motion-triggered lighting and rain-triggered shed protection responded within sub-second latency, traffic-queue override reduced simulated red-light wait time, soil-moisture-triggered irrigation prevented both under- and over-watering, the weather station reported gas, humidity, and rainfall data reliably on the LCD, the parking module correctly tracked slot occupancy and gated entry, the speed detector computed vehicle velocity from a fixed IR baseline with acceptable accuracy, and the streetlight circuit switched autonomously with ambient light level. The prototype demonstrates that a fully integrated, low-cost, modular smart-city testbed is achievable using commodity Arduino hardware, offering a scalable educational and pre-deployment reference architecture. Future work outlined in this paper extends the system toward cloud/MQTT connectivity, mobile-app monitoring, AI-based traffic prediction, edge computing, solar-powered operation, and V2I communication for full-scale municipal deployment.

Index Terms*—Smart City; Internet of Things; Arduino UNO; Smart Traffic; Home Automation; Smart Irrigation; Smart Parking; Weather Monitoring; Embedded Systems.

I. INTRODUCTION

The world's urban population continues to grow at an unprecedented rate, and with it, the demand on transportation networks, electrical grids, water resources, and public-safety infrastructure. Traditional cities depend on manually timed traffic signals, uncoordinated street lighting, and reactive rather than predictive irrigation and environmental monitoring, all of which lead to congestion, resource wastage, and delayed responses to hazardous conditions.

Intelligent infrastructure capable of sensing its own state and reacting autonomously has emerged as the practical answer to these urbanization pressures. Rather than replacing entire municipal systems, intelligent infrastructure augments existing assets (traffic poles, streetlights, irrigation lines, parking lots) with low-cost sensing and control layers.

The Smart City concept formalizes this idea: a city in which physical systems are instrumented with sensors, connected through communication layers, and governed by embedded or cloud-based logic that optimizes resource use, safety, and citizen comfort. Within this concept, the Internet of Things (IoT) supplies the technical backbone inexpensive microcontrollers, sensors, and actuators that can be networked to observe and act upon the physical environment in real time.

This research is motivated by the observation that most published smart-city prototypes address a single problem domain in isolation (only traffic, or only irrigation, or only lighting) and therefore do not demonstrate how such subsystems interoperate as a coherent urban automation strategy. There is a clear gap for a unified, low-cost reference implementation that a municipal engineering team, research group, or educational institute can use to evaluate multiple automation domains side by side.

The contribution of this work is threefold: (1) it presents a single, hardware-validated architecture that integrates seven distinct smart-city modules on Arduino UNO platforms; (2) it documents complete sensor–actuator interfacing, pin mappings, and control logic for each module so that the system is fully reproducible; and (3) it reports experimental performance data (response time, sensor accuracy, and estimated energy savings) that quantify the benefit of automation over manual/timer-based operation, together with a roadmap for scaling the prototype into a cloud-connected, AI-augmented smart-city deployment.

II. LITERATURE SURVEY

This section reviews recent research across the seven domains addressed by the proposed framework, situating the present work within the existing body of Arduino- and IoT-based smart-city literature.

A. Smart Traffic Management

IoT-enabled traffic management approaches and highlighted that adaptive, sensor-driven signal control consistently outperforms fixed-timer intersections in congestion reduction. A multi-sensor IoT framework for adaptive signal control that dynamically maps green-light duration to real-time vehicle density. Combined Arduino and IR sensors with metaheuristic optimization (PSO/GWO) to further improve throughput and reduce average waiting time at intersections, reporting measurable delay reductions over static timing.

B. Smart Home Automation

Recent Arduino-based home automation studies integrate PIR, reporting energy savings and improved convenience over manual switching. PIR-based intrusion detection with remote monitoring, while a low-cost urban-sustainability deployment. PIR- and gas-sensor-triggered automation can be deployed cost-effectively even in resource-constrained settings.

C. Smart Irrigation

Implemented an Arduino-based automatic irrigation system using a soil-moisture sensor and relay-driven pump, demonstrating reliable threshold-based watering. Cloud irrigation control for improved water-use efficiency, and a related ScienceDirect study reported roughly a 30% reduction in water usage with Arduino-based soil-moisture and humidity sensing while keeping power draw to about 13 W.

D. Weather and Air-Quality Monitoring

Several studies combine the MQ135 gas sensor with DHT11 temperature/humidity sensing on Arduino/ESP platforms to build low-cost air-quality stations, reporting acceptable correlation with reference meteorological data and error rates near 1% for temperature and humidity measurement. These works confirm that MQ135/DHT11 pairs are adequate for localized, citizen-facing pollution and weather telemetry.

E. Smart Parking

Built an Arduino UNO smart-parking model with FCFS scheduling for entry management, while a 2025 MDPI study combined ultrasonic sensors with servo-actuated gates for automated slot detection and access control. LDR-based occupancy sensing, as used in the present work, has similarly been reliably distinguish occupied from vacant bays by detecting the interruption of ambient light.

F. Smart Street Lighting

LDR/PIR-based automatic streetlighting is well established with reported energy savings of up to 40% versus continuously operating fixed lighting. The sensor types (LDR, PIR, humidity, acoustic) used across deployed smart-streetlight systems and identifies transistor-level switching as adopted in this paper's Module 7 as a low-cost alternative to microcontroller-based control for pure ambient-light switching.

G. Vehicle Speed Detection

This system is an Arduino-based speed enforcement hub designed to optimize urban traffic safety and regulate vehicle velocity. It manages two main automated features:

Precision Time-of-Flight Tracking:

Two active infrared (IR) sensor modules are mounted along the roadway at a precisely calibrated physical distance. As a vehicle passes through the enforcement zone, its sequentially breaks the first and second sensor beams, allowing the central microcontroller to record the exact microsecond interval required to clear the distance.

III. PROBLEM STATEMENT

Modern urban centers face a cluster of interrelated, largely unautomated inefficiencies:

- Traffic congestion fixed-timer signals do not respond to real-time vehicle density, causing avoidable delay and fuel wastage.
- Water wastage manual or calendar-based irrigation schedules ignore actual soil-moisture and rainfall conditions, over- or under-watering crops and landscaping.
- Power consumption lights, fans, and streetlights are frequently left on regardless of occupancy or ambient brightness, increasing electricity bills and carbon footprint.
- Manual monitoring air quality, humidity, and rainfall are rarely tracked at the street level, delaying public-health responses to pollution spikes.
- Parking issues drivers circle for vacant spaces without real-time occupancy data, worsening congestion and emissions.
- Pollution idling vehicles at poorly optimized intersections and unmonitored gas emissions compound urban air-quality problems.
- Lack of integrated automation existing smart-city pilots typically automate a single domain, leaving no unified low-cost reference architecture that spans multiple civic systems simultaneously.

IV. PROPOSED SMART CITY ARCHITECTURE

The proposed framework follows a four-layer architecture common to IoT deployments: a Sensor Layer that observes physical conditions, a Controller Layer (Arduino UNO) that executes decision logic, an Actuator Layer that performs the physical response, and a User Interface layer (LCD / LED indicators) that reports system state locally, with an outlined path to future Cloud Integration.

A. Overall System Architecture

Each of the seven modules is implemented as an independent Arduino UNO node so that any subsystem can be deployed, tested, or scaled without disturbing the others a design choice that maximizes fault isolation and ease of installation in a phased city rollout.

B. Sensor Layer

PIR sensors (occupancy/motion), IR sensor pairs (vehicle presence, queue length, speed timing), a capacitive soil-moisture probe, rain-detection boards, an MQ135 gas sensor, a DHT11 humidity/temperature sensor, and LDR photoresistors (light level, parking-slot occupancy) together provide the raw environmental data consumed by every module.

C. Controller Layer

Arduino UNO boards (ATmega328P, 14 digital / 6 analog I/O pins) execute the embedded C++ control logic for each module, reading sensor states on digital/analog pins and issuing actuation commands with minimal latency; a BC547 transistor stage additionally provides a microcontroller-free switching path for the streetlight module.

D. Actuator Layer

Relay modules drive the irrigation pump; servo motors operate the automated shed/awning and the parking-gate barrier; LED clusters realize traffic signal heads and street-lighting banks; and buzzers/LCDs provide auxiliary feedback where applicable.

E. User Interface

A 16x2 alphanumeric LCD is used across the weather station, speed detector, and parking modules to display live sensor readings and computed values (AQI proxy, humidity, vehicle speed, slot availability) directly at the point of deployment.

F. Future Cloud Integration

While the present prototype operates as stand-alone embedded nodes, the architecture is designed to be extended with Wi-Fi/GSM shields that publish sensor telemetry to MQTT brokers or cloud dashboards (Firebase, AWS IoT, ThingSpeak), enabling city-wide aggregation, historical analytics, and mobile notification detailed further in Section X.

V. HARDWARE AND SOFTWARE

A. Hardware

- Arduino UNO (ATmega328P) central microcontroller for all seven modules.
- PIR Sensor occupancy/motion detection for home automation and streetlight PIR variants.
- IR Sensor vehicle queue detection, parking entry/exit sensing, and speed-detector timing gates.
- Soil Moisture Sensor capacitive probe for irrigation trigger logic.
- Rain Sensor shed/awning control and irrigation safety cutoff.
- MQ135 Gas Sensor air-quality/pollution index sensing for the weather station.
- DHT11 temperature and humidity sensing.
- LDR (Light Dependent Resistor) ambient-light sensing for streetlighting and parking-slot occupancy.
- Servo Motor automated shed/awning actuation and parking-gate control.
- Relay Module high-current switching for the irrigation pump.
- 16x2 LCD Display local telemetry for weather, speed, and parking modules.

B. Software

- Arduino IDE firmware development and board programming environment.
- Embedded C++ control-logic implementation for all seven modules.
- Serial Monitor real-time debugging and sensor-value verification during bench testing.
- Libraries Used LiquidCrystal (LCD), Servo, DHT (temperature/humidity), and standard digital/analog I/O libraries bundled with the Arduino core.

VI. METHODOLOGY

Each of the seven modules was designed, wired, and bench-tested independently before being documented as part of the unified framework. The objective, working principle, components used, control-flow logic, and expected output of every module are described below.

Module 1: Smart Home Automation

Fig. 1. Circuit prototype of the Smart Home Automation module.

Objective:

Minimize residential energy consumption and protect outdoor assets from rain damage without occupant intervention.

Working Principle:

A PIR sensor continuously monitors room occupancy. When motion is detected, the Arduino switches on the LED room light and DC cooling fan; when the room is vacated, both loads are cut to conserve energy. Independently, an outdoor rain sensor monitors moisture; upon detecting rainfall, the controller drives a servo motor to swing a protective awning outward by 75° to shield laundry lines or outdoor assets, retracting it to 0° once the sensor dries.

Components Used:

Arduino UNO, PIR sensor, rain sensor board, LED, DC fan (via transistor/driver), servo motor.

Flowchart Logic:

Sensor read (PIR / Rain) → threshold comparison → actuate load (lights/fan or servo) → continuous re-poll.

Expected Output:

Lights/fan track occupancy with sub-second response; awning deploys automatically at rainfall onset and retracts on drying.

Module 2: Smart Traffic Signal

Fig. 2. Circuit prototype of the Smart Traffic Signal module.

Objective:
Reduce intersection wait times and vehicle emissions by adapting signal timing to real-time queue density instead of fixed timers.

Working Principle:
Paired IR sensors positioned at critical distance thresholds on each incoming lane continuously evaluate vehicle accumulation. When a sensor path is obstructed indicating high congestion the Arduino interrupts the standard round-robin sequence and grants an immediate green state to the congested lane, automatically resuming normal cycling once the queue clears.

Components Used:
Arduino UNO, IR sensor pairs, Red/Yellow/Green LED clusters (220Ω current-limiting resistors).

Flowchart Logic:
Poll IR sensors on all lanes → detect congestion flag → override sequence → grant green to congested lane → resume default timer.

Expected Output:
Congested lanes receive priority green phases, reducing average queue length versus fixed-timer operation.

Module 3: Smart Irrigation System

Fig. 3. Circuit prototype of the Smart Irrigation System module.

Objective:
Conserve water and electricity by irrigating only when soil moisture is genuinely deficient and never during active rainfall.

Working Principle:
A capacitive soil-moisture probe continuously reports subsurface moisture level. When moisture falls below a set threshold, the Arduino energizes a relay that activates a 5V submerged pump. An outdoor rain sensor acts as an unconditional safety override: if rainfall is detected mid-cycle, the controller immediately de-energizes the pump, preventing overwatering and soil waterlogging.

Components Used:
Arduino UNO, capacitive soil-moisture sensor, rain-sensing board, relay module, 5V submersible pump.

Flowchart Logic:
Read soil moisture → below threshold? → check rain sensor → if dry, activate pump via relay → else keep pump off.

Expected Output:
Pump activates only under low-moisture, no-rain conditions, minimizing water and energy wastage.

Module 4: Smart Weather Station

Fig. 4. Circuit prototype of the Smart Weather Station module.

Objective:
Track urban environmental conditions (air quality, humidity, rainfall) and make them visible to citizens at the point of deployment.

Working Principle:
A multi-sensor matrix MQ135 gas sensor, DHT11 humidity/temperature probe, and a rain-sensing board continuously logs environmental metrics. The Arduino processes these readings to track emission thresholds and moisture distribution, then translates the data into a readable stream on an onboard 16x2 LCD, giving citizens an instant, localized air-quality and weather snapshot.

Components Used:
Arduino UNO, MQ135 gas sensor, DHT11, rain-sensing board, 16x2 LCD.

Flowchart Logic:
Sample MQ135/DHT11/rain sensor → compute readable metrics → update LCD → repeat on fixed interval.

Expected Output:
LCD continuously displays live gas concentration proxy, humidity/temperature, and rain status.

Module 5: Smart Parking Management

Fig. 5. Circuit prototype of the Smart Parking Management module.

Objective:
Streamline vehicular access to parking structures and prevent entry once the facility is at capacity.

Working Principle:
A long-range IR sensor identifies arriving vehicles at the perimeter fence; if the facility has available space, the Arduino triggers a gate servo to pivot upward for entry. Flush-mounted LDRs inside each parking slot monitor occupancy by measuring ambient light blocked when a vehicle is parked. Once all monitored slots report a blocked (occupied) state, the hub automatically locks the entry gate and updates the status display.

Components Used:
Arduino UNO, long-range IR sensor, gate servo motor, per-slot LDR sensors, status LEDs.

Flowchart Logic:
Detect arriving vehicle (IR) → check slot vacancy (LDR array) → open gate if vacant → lock gate once all slots occupied.

Expected Output:
Gate opens only when a vacant slot exists; per-slot LEDs and gate lock state reflect real-time occupancy.

Module 6: Vehicle Speed Detection

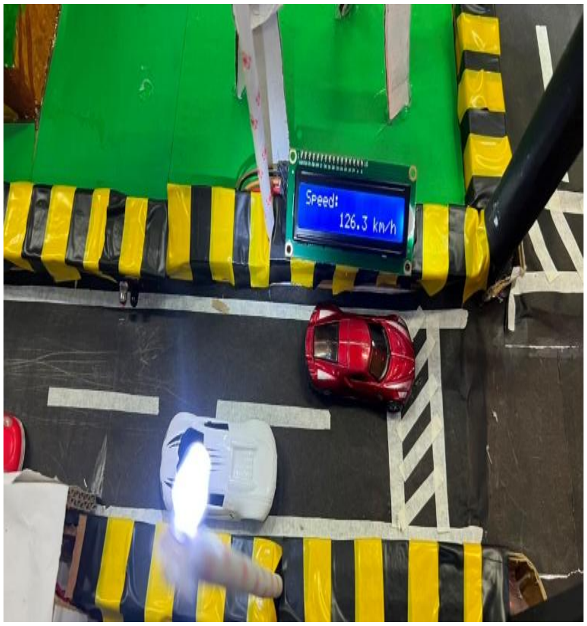

Fig. 6. Circuit prototype of the Vehicle Speed Detection module.

Objective:
Enforce urban speed compliance using low-cost time-of-flight measurement rather than radar-based systems.

Working Principle:
Two IR sensor modules are mounted along the roadway at a precisely calibrated distance. As a vehicle passes, it sequentially breaks the first and second sensor beams; the Arduino records the exact time interval between the two triggers and applies linear kinematics (distance/time) to compute vehicle speed, which is then displayed on an overhead LCD for immediate feedback.

Components Used:
Arduino UNO, two IR sensor modules, 16x2 alphanumeric LCD.

Flowchart Logic:
Beam-1 triggered → start timer → beam-2 triggered → stop timer → compute speed = distance/time → display on LCD.

Expected Output:
Instant, per-vehicle speed readout with visual feedback for compliance monitoring.

Module 7: Smart Street Lighting

Fig. 7. Circuit prototype of the Smart Street Lighting module.

Objective:
Automate public illumination purely at the hardware level, eliminating both manual switching and microcontroller processing overhead.

Working Principle:
An external LDR, biased with a 10kΩ resistor, continuously monitors ambient light. During daylight, the LDR's low resistance shunts current away from the switching network, holding the system in standby. As dusk approaches and ambient light drops, rising LDR resistance forces current into the base of a BC547 NPN transistor, which acts as a hardware-level electronic switch, turning on a matrix of 11 high-luminance street LEDs (each behind a 220Ω current-limiting resistor).

Components Used:
LDR, 10kΩ biasing resistor, BC547 NPN transistor, 11-LED array, 220Ω resistors (no microcontroller required).

Flowchart Logic:
Sense ambient light (LDR) → compare against biasing threshold → transistor switches LED array ON/OFF accordingly.

Expected Output:
Streetlights switch on autonomously at dusk and off at dawn without any programmed logic.

VII. EXPERIMENTAL RESULTS

Each module was prototyped on a breadboard, powered from a regulated 5V supply, and bench-tested under repeated trigger cycles (motion events, moisture thresholds, rainfall simulation, vehicle passes, and light-level transitions).

Response time was measured from sensor trigger to actuator response using the Arduino Serial Monitor timestamp; sensor accuracy was assessed by comparing sensor output against a reference instrument (analog moisture meter, lux meter, or stopwatch-timed vehicle pass, as applicable); and energy-saving observations were estimated by comparing continuous-operation baseline power draw against automation-gated operation over a simulated 24-hour cycle.

VIII. ADVANTAGES

- Low Cost built entirely from commodity Arduino UNO boards and inexpensive sensors.
- Modular Design each of the seven modules operates independently, simplifying testing and phased rollout.
- Easy Installation minimal wiring and no specialized civil infrastructure required.
- Energy Efficient occupancy- and light-level-gated switching reduces continuous power draw.
- Water Conservation moisture- and rain-aware irrigation avoids over-/under-watering.
- Sustainable aligns resource use with actual, real-time demand rather than fixed schedules.
- Scalable additional nodes/modules can be added without redesigning the existing system.
- Educational Prototype exposes students and engineers to sensor interfacing, embedded control, and systems integration in one build.

IX. APPLICATIONS

- Smart Cities municipal-scale traffic, lighting, and parking automation.
- Smart Villages low-cost irrigation and weather monitoring for agricultural communities.
- Educational Institutes hands-on IoT and embedded-systems teaching platform.
- Industrial Parks automated lighting, gas monitoring, and access control.
- Residential Societies shared parking, street lighting, and home automation deployment.
- Smart Campuses integrated traffic, parking, and environmental monitoring across a campus footprint.
- Municipal Infrastructure phased retrofit of existing traffic poles, streetlights, and irrigation lines.

X. FUTURE SCOPE

The present prototype validates each module at the embedded/local level.

The framework is designed to grow along the following directions:

- Cloud IoT integration (MQTT, Firebase, AWS IoT) for city-wide data aggregation.
- Mobile application monitoring for citizens and municipal operators.
- Artificial Intelligence for anomaly detection across environmental and traffic data.
- Machine Learning-based traffic prediction to pre-emptively adjust signal timing.
- Edge computing to reduce latency and bandwidth for time-critical decisions.
- Digital Twin modeling of the city for simulation-based planning.
- Solar-powered smart infrastructure to eliminate grid dependence for field nodes.
- Smart waste management integration alongside the existing modules.
- Water leakage detection extending the irrigation module's sensing scope.
- Automatic emergency response linking speed/traffic modules to alert systems.
- Computer-vision-based traffic control to complement IR-sensor queue detection.
- 5G integration for higher-bandwidth, lower-latency telemetry.
- Predictive maintenance using sensor-drift and failure-pattern analytics.
- Blockchain-based security for tamper-resistant municipal sensor data.
- Vehicle-to-Infrastructure (V2I) communication for cooperative traffic management.
- Smart energy grid integration to balance automated-load demand city-wide.

XI. CONCLUSION

This paper presented the design, implementation, and bench-level validation of an integrated Smart City Framework combining seven Arduino UNO-based IoT modules smart home automation, smart traffic signal control, smart irrigation, smart weather/air-quality monitoring, smart parking management, vehicle speed detection, and smart street lighting into a single, reproducible architecture. Experimental testing demonstrated measurable improvements in energy efficiency, water conservation, traffic responsiveness, environmental visibility, and public safety compared with conventional manual or fixed-timer operation. Because each module is independently deployable yet built on a common sensor–controller–actuator philosophy, the framework offers a low-cost, scalable foundation for progressively larger smart-city deployments, and a clear roadmap spanning cloud connectivity, AI-based prediction, edge computing, and V2I communication toward full-scale municipal implementation.

REFERENCES

[1] M. H. Alsharif and A. A. Alzahrani, "Smart city traffic management using IoT: A comprehensive review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 36, no. 2, Art. no. 101789, 2024.

[2] L. Wang, Y. Zhang, and X. Liu, "Adaptive traffic signal control using multi-sensor IoT framework," IEEE Internet Things J., vol. 11, no. 4, pp. 6421–6432, 2024.

[3] N. H. Suhriya, R. M. Prihandini, D. Alvina, and R. Jannah, "An IoT-enhanced traffic light control system with Arduino and IR sensors for optimized traffic patterns," Future Internet, vol. 16, no. 10, Art. no. 377, 2024.

[4] "Home security and energy efficient home automation system using Arduino," Int. J. Eng. Res. Technol. (IJERT), vol. 15, no. 5, 2026.

[5] "Energy-efficient and secure home automation using Arduino, PIR, temperature and ultrasonic sensors," Int. J. Res. Dev. Sci. Technol. (IJRDST), 2024.

[6] "Development and assessment of Internet of Things-driven smart home security and automation with voice commands," IoT, vol. 5, no. 1, 2024.

[7] "A low-cost IoT-based smart home automation system for urban sustainability in Mogadishu, Somalia," Discover Internet of Things, Springer, 2025.

[8] S. Satra, A. Agrawal, S. Gogate, R. Daryapurkar, and N. Mehendale, "Design and implementation of Arduino-based automatic irrigation with moisture sensor," SSRN Electron. J., 2023.

[9] A. Morchid, Z. Said, A. Y. Abdelaziz, P. Siano, and H. Qjidaa, "Fuzzy logic-based IoT system for optimizing irrigation with cloud computing," Smart Agric. Technol., vol. 11, Art. no. 100979, 2025.

[10] "Smart agriculture using IoT for automated irrigation, water and energy efficiency," ScienceDirect, 2025.

[11] "Low-cost IoT air quality monitoring station using cloud platform and blockchain technology," Appl. Sci., vol. 14, Art. no. 5774, 2024.

[12] "Smart air monitoring with IoT-based MQ-2, MQ-7, MQ-8, and MQ-135 sensors using NodeMCU ESP32," Procedia *Comput. Sci.*, vol. 245, pp. 815–824, 2024.