IoT Based Smart Campus System for Classroom Automation and Intelligent Parking

Mrs. Ch. Swetha[1], Gubba Nithish[2], G. Jyothi[3], Embari Rohith[4], Karkala Esther Rani[5]

[1]Assistant Professor, Department of Information Technology, Vidya Jyothi Institute of Technology (A), Hyderabad, Telangana, India,

[2,3,4,5]UG Students, Department of Information Technology, Vidya Jyothi Institute of Technology (A), Hyderabad, Telangana, India

---***---

Abstract–*Most college campuses still run their day-to-day infrastructure the old-fashioned way. Lights are left on in empty rooms, drivers circle the parking lot looking for a free spot, dustbins overflow before anyone notices, and garden plants get watered on a fixed schedule whether they need it or not. This paper describes an IoT-based Smart Campus System built to address these problems using two low-cost Wi-Fi microcontrollers, an ESP32 and an ESP8266, connected through the Blynk cloud platform. On the ESP32 side, relay modules switch classroom lights and a fan, and each relay can be operated either from the existing wall switch or from the Blynk mobile app, with the two always kept in sync. The same board reads a soil-moisture sensor and turns a water pump on only when the soil needs it, and it reads an ultrasonic sensor to keep track of how full a dustbin is getting. On the ESP8266 side, four IR sensors, three at the parking slots and one at the entry gate, report slot availability and open a servo-driven gate automatically when a vehicle arrives, with a manual override available from the app if required. All of this comes together on a single Blynk dashboard, so the whole system can be checked and controlled from a phone. A working prototype of all four modules was built and tested, and the results were encouraging: the system responded quickly and reliably, and water usage dropped noticeably compared to manual watering. Overall, the work shows that a fairly simple, low-cost combination of relays, sensors and a cloud dashboard is enough to make everyday campus operations a good deal more convenient and less wasteful.*

Key Words: Internet of Things (IoT), Smart Campus, ESP8266, ESP32, Classroom Automation, Intelligent Parking, Smart Irrigation, Blynk Cloud Platform.

1. INTRODUCTION

Over the last decade or so, the Internet of Things has moved from being a buzzword to something that quietly runs in the background of everyday life. Objects that used to be purely physical (a light switch, a water valve, a parking barrier) can now sense their surroundings and act on that information without a person having to step in every time. This shift started in factories and warehouses but has since spread into homes, cities and, increasingly, college campuses. A campus turns out to be a good candidate for this kind of automation, since it brings together so many different types of infrastructure (classrooms, parking areas, gardens, waste bins) under one roof, and small inefficiencies in any one of them add up quickly.

In practice, though, most institutions still manage these facilities by hand. Lights and fans get left running in classrooms nobody is using. Drivers waste time hunting for an empty parking spot that may not even exist. Dustbins are emptied on a fixed schedule rather than when they are actually full, and garden beds get watered the same way regardless of whether the soil is already wet. None of this is disastrous on its own, but together it drives up electricity and water bills and makes day-to-day campus life a little more frustrating than it needs to be.

What makes this worth solving now is that the hardware has become genuinely cheap. Boards like the ESP8266 and ESP32 pack Wi-Fi, decent processing power and a handful of GPIO pins onto a board that costs only a little more than a basic Arduino, and matching sensors are just as inexpensive. Combined with free-tier cloud platforms such as Blynk, it is now realistic for a student project, let alone an institution, to put together a working automation system without anything close to the budget of a commercial building-management solution. That is the gap this work tries to fill: rather than building four separate, one-off gadgets, the aim was to bring classroom switching, parking, garbage monitoring and irrigation together under one architecture and one dashboard.

The rest of the paper is laid out as follows. Section 2 looks at related work on smart campuses, parking, irrigation and waste monitoring. Section 3 covers the system architecture and the hardware and software used. Section 4 walks through how each module was put together. Section 5 reports what happened during testing and discusses what it means. Section 6 lists the advantages of the approach, Section 7 talks about where the project could go next, and Section 8 wraps up.

1.1 Objectives of the Study

The design of the system was guided by five main goals:

- Detect vehicles at each parking slot and at the entry gate using IR sensors, show live slot availability on the mobile dashboard, and let a servo motor handle the gate automatically.
- Let classroom lights and the fan be switched either from the wall or from the Blynk app, with both always showing the same, correct state.
- Track dustbin fill level with an ultrasonic sensor so staff do not have to guess when a bin needs emptying.
- Water the garden only when the soil sensor says it is dry, instead of on a fixed timer.
- Send sensor and switch status to the cloud continuously so the whole system can be checked and controlled from a phone.

1.2 Scope of the Work

This work is scoped as a working prototype rather than a full campus rollout. It covers one classroom's worth of switching, a small three-slot parking bay, one dustbin and one garden bed. That is enough to prove the idea without needing a large budget. The control logic is simple, threshold- and state-based, rather than anything learning-based, and Blynk is used for cloud connectivity instead of a custom server, again to keep things quick to build and easy to maintain. Scaling this up to a full campus, and adding the kind of predictive intelligence discussed in Section 7, is left as future work.

These four functions were picked because they are the ones students and staff actually notice on a daily basis. Everyone has walked past an empty, fully lit classroom, or circled a parking area twice before finding a spot, and these are small annoyances that add up over a semester. Garbage and irrigation were added for much the same reason: they are easy to overlook until a bin overflows or a plant dies, at which point the fix is usually more effort than the monitoring would have been in the first place. None of the four modules is technically difficult on its own, which was rather the point: the contribution here is less about any single sensor and more about getting all four to sit comfortably under one controller setup and one dashboard.

2. LITERATURE REVIEW

Smart campus systems built on IoT have received a fair amount of research attention in the last few years, ranging from single-function prototypes to broader architectural surveys. This section walks through the work most relevant to the four functions covered in this paper, drawing mainly on papers published between 2021 and 2024.

Most of this literature describes IoT systems using roughly the same three-layer picture: a perception layer of sensors and actuators, a network layer that moves the data around, and an application layer that presents it and lets a user act on it [10]. It is a useful way to compare systems that use very different sensors and cloud platforms but end up moving data through a similar pipeline, and it is essentially the structure followed in Section 3 for the system proposed here.

A couple of papers focus specifically on the classroom side of things. One study proposed a smart campus monitoring system that gathers real-time data from campus devices so instructors can adjust their teaching based on it [8]. Another built an IoT-enabled smart classroom that tied together sensors and software for attendance, lighting, temperature control and security, and reported that the overall classroom experience improved as a result [9]. A broader review looked at how the usual perception-network-application layering shows up across different smart-campus deployments and pointed to room for extending automation to more campus services [10]. What these papers have in common is that connecting classroom electrical loads to a remotely accessible control layer clearly helps cut down on wasted energy, which is really what motivated the relay-based classroom module used here, controllable from both a wall switch and the app, as a simpler and cheaper alternative to full occupancy-sensor automation.

For parking, ultrasonic- and IR-based slot detection keeps coming up as a cheap alternative to camera- or RFID-based systems. One system built around ultrasonic sensors and a low-cost microcontroller reported high detection accuracy at a fraction of the cost of comparable commercial systems [1]. A follow-up study added a mobile app on top of an ultrasonic-sensor network so drivers could check slot availability before even arriving [2]. Another used an ESP32 to detect both approaching and already-parked vehicles with ultrasonic sensors, which simplified how slots were allocated [12]. Taken together, this work supports the choice made here: IR sensing paired with a servo-actuated gate, with slot availability shown directly on a phone rather than on a physical display.

Smart irrigation research has settled fairly consistently on soil-moisture threshold control as the practical way to save water. One ESP8266-based system switched a pump on and off purely based on live moisture readings, cutting down on manual watering [5]. A closely related paper, published in this same journal, did much the same thing and reported clear water savings as a result [3]. Another combined a moisture sensor with an ESP8266 NodeMCU to build a small, Wi-Fi connected controller aimed at gardens and small plots [4], and a cloud-based version of the same idea added remote logging of irrigation cycles

[13]. All of this backs up the threshold-based approach used in the present system.

Waste management research follows a similar pattern, with ultrasonic bin-level monitoring used to move away from fixed collection schedules toward collecting only when needed. One prototype used ultrasonic and moisture sensors with wireless connectivity to classify and report waste levels [6], and a related system used ultrasonic sensors to track fill levels and send Wi-Fi alerts before a bin overflowed [7]. At city scale, linking ultrasonic bin sensors to cloud analytics has been shown to make collection routes noticeably more efficient [14]. The garbage-monitoring approach used here follows the same basic idea, just scaled down to a single bin.

Finally, Blynk itself has been used in a number of smart-home and monitoring projects as a quick way to expose sensor data and device control through a phone app without building a custom backend [11]. That track record is largely why it was chosen here as the dashboard for all four modules.

Table -1 pulls together the focus, method and main limitation of the papers discussed above. Looking across them, a pattern stands out: almost every one of these systems tackles a single campus function on its own (parking, irrigation, waste, or classroom control) rather than combining them. Very few attempt to bring these functions together under one controller-and-dashboard setup. That gap is really the starting point for this paper, which puts all four functions on a shared ESP32/ESP8266-and-Blynk platform.

It is worth adding that most of the systems reviewed above were also evaluated as standalone prototypes in a lab or a single test location, much like the one described in this paper, rather than as part of a live campus deployment. That is a fair and common limitation at this stage of research, and it applies here too: the results in Section 5 come from a small, controlled prototype rather than months of use in an actual building. Even so, the consistency with which threshold-based sensing and Blynk-style dashboards show up across this body of work is a reasonable indication that the same approach should hold up reasonably well outside the lab, provided the thresholds are recalibrated for the specific site.

Table -1: Comparative Summary of Related Work

Ref.	Focus Area	Technique Used	Key Limitation
[1]	Intelligent Parking	Ultrasonic sensor based slot detection with low-cost microcontroller	Limited scalability for large multi-level parking areas
[3]	Smart Irrigation	Soil-moisture threshold based automatic watering	No integration with weather forecast data
[6]	Smart Waste Management	Ultrasonic and moisture sensors with wireless bin-level reporting	Susceptible to false triggers from environmental noise
[8]	Smart Campus Monitoring	IoT devices collecting real-time campus and student data	Focused on data collection with limited automated actuation
[9]	Smart Classroom (S3TH)	Multi-sensor classroom control for lighting, temperature and security	Higher hardware and deployment cost per classroom
[11]	IoT Dashboard Platform	Blynk-based mobile monitoring and device control	Depends on continuous, stable internet connectivity
[14]	City-scale Waste Management	Cloud-linked ultrasonic bin sensors with analytics	Requires wide-area network infrastructure not always available on campus

3. PROPOSED SYSTEM AND METHODOLOGY

3.1 System Architecture

The system proposed here runs on two Wi-Fi microcontrollers that share a common Blynk application layer, rather than trying to force everything onto one board. An ESP32 handles the classroom-switching, irrigation and garbage-monitoring side: it reads three wall switches, a soil-moisture sensor and an ultrasonic sensor, and in turn drives three relay-controlled classroom loads, a relay-driven pump and a buzzer that can be triggered from the app. A second board, an ESP8266, takes care of parking: it reads four IR sensors (three at the parking slots and one at the gate) and drives a servo that opens and closes the gate. Both boards push their readings and states to Blynk over Wi-Fi, and the app layer brings everything together into one dashboard where every actuator can also be overridden manually. Figure 1 shows this arrangement.

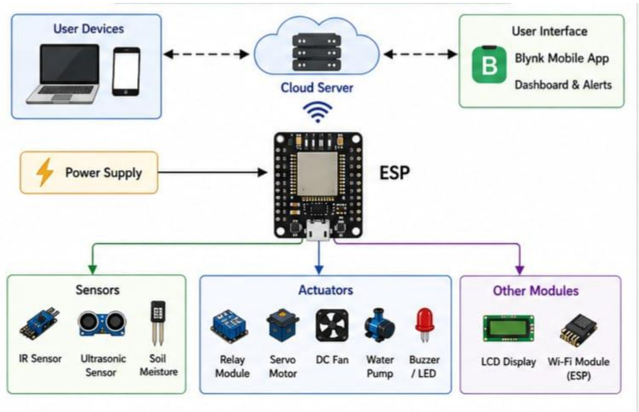

Fig -1: System Architecture of the Proposed IoT Based Smart Campus System

3.2 Hardware Components

The hardware is built entirely from parts that are cheap and easy to source. The list below summarises what is used and what each part does:

- ESP32 – the main controller, handling classroom switching, irrigation and garbage monitoring.
- ESP8266, a second, smaller controller dedicated purely to the parking module.
- Three physical toggle switches, wired so classroom lights and the fan can still be switched locally, exactly as before, alongside app control.
- Relay modules that isolate and switch the classroom loads and the water pump.
- Four IR sensors: one per parking slot, plus one at the entry gate.
- A servo motor to open and close the parking gate.
- An HC-SR04 ultrasonic sensor, which measures how full the dustbin is.
- A soil-moisture sensor that tells the controller when the garden needs water.
- A small water pump for irrigation.
- A buzzer that gives an audible alert, switchable from the app.

3.3 Software Platform

Both controllers were programmed in C/C++ through the Arduino IDE. Each one polls its own sensors and switches, checks a small set of rules to decide what the actuators should do, and writes the result to the matching virtual pins on Blynk over Wi-Fi. On top of that, a single Blynk dashboard was built to show parking availability, the classroom relay/switch states, dustbin fill level and soil-moisture percentage, and to let an authorised user step in and control any of these manually if needed.

3.4 Working Principle

Rather than reach for anything machine-learning based, the system sticks to simple, rule-based logic, which is fast, predictable and well within what an ESP8266 or ESP32 can handle comfortably. Broadly, the control loop runs in four stages:

- First, read the sensors: pull in the latest switch states, IR readings, ultrasonic distance and moisture level.
- Then clean up the readings, filtering out noise so a stray value does not trigger a false action.
- Check the rules: was a switch flipped, is a vehicle at a slot or the gate, is the bin nearly full, is the soil dry.
- Finally, act on it: switch the relevant actuator and push the new state up to the Blynk dashboard.

3.5 Control Algorithm

The pseudocode below is a simplified version of what each controller runs every cycle:

Step 1: Read physical switch states, IR sensor states, ultrasonic distance and soil-moisture sensor values.
Step 2: IF wall switch OR app button state changes THEN set the corresponding relay to match, and keep the switch and app state synchronised.
Step 3: IF vehicle detected in slot THEN mark slot Occupied and update dashboard availability percentage, ELSE mark slot Free.
Step 4: IF vehicle detected at entry gate THEN open Servo Gate and close it after a fixed duration, ELSE keep Gate closed (manual override available from the app).
Step 5: Compute bin fill level from ultrasonic distance and publish it to the dashboard.
Step 6: IF automatic mode enabled AND soil moisture < dry threshold THEN turn Pump ON, ELSE IF soil moisture >= target level THEN turn Pump OFF.
Step 7: Publish all updated sensor and actuator states to the Blynk cloud dashboard and repeat from Step 1.

4. IMPLEMENTATION

With the architecture settled, the next step was to build and test each module. Every module was wired up and checked on its own before being brought together under the two controllers and the shared dashboard.

4.1 Classroom Power Switching Module

The classroom module runs on the ESP32, wired to three relays (two lights and a fan) and three physical toggle switches. Flip the wall switch and the relay responds, with the Blynk app button updating to match; tap the app button instead and the relay responds the same way, with the physical switch position mirrored in software. Either way, the two stay in sync, so a user gets the convenience of controlling classroom devices from a phone without losing the switch that was already there.

4.2 Intelligent Parking Module

Parking runs on its own ESP8266, with an IR sensor at each of the three slots and a fourth at the gate. The board keeps checking the three slot sensors and turns that into a live availability percentage on the dashboard. If the gate sensor picks up an approaching vehicle, the servo opens the gate on its own and closes it again after a short delay; there is also a button in the app to open or close the gate by hand if that automatic behaviour ever needs to be overridden.

4.3 Garbage Detection Module

An ultrasonic sensor sits above the dustbin, wired to the same ESP32 board, and checks the distance down to the waste every few seconds. That distance is converted into a fill percentage and sent straight to the dashboard, so cleaning staff can see which bins are getting full without walking over to check.

4.4 Smart Irrigation Module

The soil-moisture sensor, also on the ESP32, keeps reading the moisture level and reporting it as a percentage on the dashboard. With automatic mode switched on in the app, the pump comes on once the reading drops below the dry threshold and turns off again once it climbs back above the wet threshold. Automatic mode can also be switched off from the same app control, which is handy if it has just rained and the pump should stay off regardless of what the sensor says.

4.5 System Integration and Prototype

The two boards do not talk to each other directly, but both report into the same Blynk template, so everything (classroom switches, parking, the bin and the irrigation) ends up on one dashboard. Figure 2 shows the two prototypes built for testing: the classroom/irrigation/garbage board on one side and the parking board on the other.

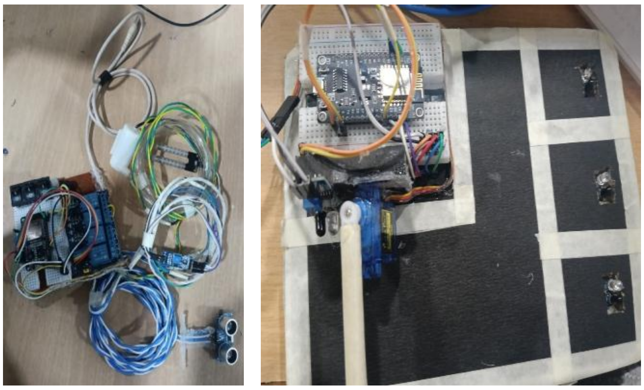

Fig -2: Hardware Prototype – (a) Classroom Automation Module (b) Intelligent Parking Module

5. RESULTS AND DISCUSSION

Once assembled, each module was tested on its own first, checking that each sensor produced the expected actuator response, before the whole setup was checked together through the dashboard. Table -2 lists what was tested and what happened.

Testing itself was fairly hands-on: the wall switches were flipped by hand while watching both the relay and the Blynk app to confirm the two agreed; a small object was moved in and out of the IR sensors' range to simulate a car entering or leaving a slot and to trigger the gate; water was added to and withheld from the soil around the moisture sensor to push it across the dry and wet thresholds; and the ultrasonic sensor was tested by placing an object at varying distances above it to imitate different bin-fill levels. Each of these was repeated a few times to check that the behaviour was consistent rather than a one-off coincidence.

Table -2: Functional Testing Summary

Module	Test Condition	Observed Behaviour	Result
Classroom Switching	Physical wall switch toggled ON	Relay activated; Blynk app button updated to reflect ON state	Pass
Classroom Switching	Blynk app button toggled OFF	Relay deactivated; state synchronised back to the app	Pass
Intelligent	Vehicle enters a	Slot marked	Pass

Parking	monitored slot	occupied; dashboard availability percentage updated	
Intelligent Parking	Vehicle detected at entry gate	Servo gate opened automatically and closed after a fixed duration	Pass
Garbage Detection	Bin filled to a higher level	Fill percentage on dashboard increased accordingly	Pass
Smart Irrigation	Automatic mode ON, soil moisture below dry threshold	Water pump switched ON automatically	Pass
Smart Irrigation	Soil moisture reaches wet threshold	Water pump switched OFF automatically	Pass
Dashboard Sync	Any sensor/actuator state change on either controller	Blynk mobile dashboard updated in real time	Pass

All eight tests behaved as expected, which is a reasonable sign that the rule-based logic is doing its job correctly across all four modules. Dashboard updates lagged behind the actual hardware state by a small, fairly consistent amount, which is typical of ESP8266/ESP32 Wi-Fi communication and not really a concern for functions like these that are not safety-critical. Figure 3 shows the dashboard used to keep an eye on parking, the classroom switches, bin level and soil moisture from a single screen.

Fig -3: Mobile Application Dashboard (Blynk)

What this suggests, overall, is that a fairly ordinary combination of relays and sensors is enough to cover these four campus functions without needing anything like a full building-management system. That said, testing also surfaced a few real limitations. Everything depends on a stable Wi-Fi connection to update the dashboard, so a dropped connection delays that sync, though the physical switches, the automatic gate and the moisture-triggered pump all keep working locally regardless, since none of them need the cloud to function. The dry/wet moisture thresholds needed some manual tuning to match the soil at the test site, and pushing the parking or classroom modules to cover many more slots or rooms would mean adding more controllers or some kind of mesh network rather than just the two boards used here. It is also worth being upfront that the classroom module is switch-based rather than occupancy-based, so any energy savings still depend on someone remembering to turn things off, whether at the wall or through the app.

Cost-wise, everything in this prototype (the two micro controllers, the relays, the switches, the IR sensors, the ultrasonic sensor, the moisture sensor, the servo, the pump and the buzzer) is an off-the-shelf part that is easy to find and inexpensive. That, combined with Blynk's free tier, makes the whole approach realistic even for institutions without much of a budget, and it would not cost much more, in relative terms, to extend the same setup to a few more classrooms or parking bays.

6. ADVANTAGES OF THE PROPOSED SYSTEM

- Classroom lights and the fan can be switched from either the wall or a phone, so it is much easier to remember to turn things off.
- Water use drops because the garden is watered only when the soil sensor says it is needed, not on a fixed schedule.
- Bins are less likely to overflow, since staff can see fill levels before it becomes a problem.
- Drivers save time by checking parking availability before they even reach the lot.
- Four previously separate mini-projects sit together under one cheap platform and one dashboard.
- Every part used is inexpensive and easy to source, so building and maintaining the system stays affordable.
- The modular design means it can be extended to more classrooms, parking bays or garden beds without redesigning anything.

7. FUTURE SCOPE

The current prototype covers all four functions, but there is plenty of room to take it further. A web portal alongside the mobile app, for instance, could give administrators a single consolidated view of energy use, parking trends, bin-collection history and irrigation cycles across the whole campus instead of one board at a time.

Adding real occupancy sensing, such as a PIR sensor or an ambient-light sensor, would be a natural next step, letting the classroom module switch lights and the fan automatically instead of relying purely on the wall switch and the app. On top of that, machine-learning based prediction could be layered in to anticipate classroom usage patterns and adjust lighting ahead of time, or to combine soil-moisture history with weather forecasts so the pump stays off if rain is already on the way.

The parking module could support slot reservations directly from the app, and the garbage module could add load-cell weight sensors alongside the existing ultrasonic sensor to help plan collection routes when several bins are filling up at once.

It would also make sense to power the outdoor nodes (parking and irrigation) from small solar panels with a battery backup, moving the system a step closer to running fully off-grid. Beyond that, adding temperature, humidity or air-quality sensors would extend the system past its current switching, parking, waste and irrigation scope into broader environmental monitoring.

A short, informal trial with a handful of classmates using the current Blynk dashboard also turned up a few smaller software improvements worth pursuing: separate tabs for each module rather than one crowded screen, push notifications when a bin or a parking area crosses a set level instead of requiring someone to open the app and check, and a simple login system so that only authorised staff can override the automatic irrigation and gate behaviour. None of these change the underlying architecture, but they would make the system noticeably more pleasant to use day to day.

8. CONCLUSION

This paper set out to build and test an IoT-based Smart Campus System that brings classroom power switching, intelligent parking, garbage-level monitoring and smart irrigation together on ESP32 and ESP8266 controllers, all tied to one Blynk dashboard. Testing across all four modules showed that the relay-based classroom switches respond correctly whether triggered from the wall or the app, the parking module tracks and reports slot availability while handling gate access on its own, the garbage module reports bin level accurately, and the irrigation module waters the garden automatically based on actual soil conditions. Taken together, the combination of easy local/remote control, threshold-driven irrigation and continuous remote monitoring for parking and waste cuts down on unnecessary water use, keeps bins from overflowing, and gives administrators a live view of the campus from their phone. Because the hardware is cheap and the design is modular, extending it to more classrooms, parking areas or garden beds should not be difficult, and the improvements outlined in Section 7 (adding real occupancy sensing being the most obvious one) point toward a system that is more automatic, more centralised and, eventually, closer to running itself.

ACKNOWLEDGEMENT

The authors would like to thank Mrs. Ch. Swetha, Assistant Professor, Department of Information Technology, for her guidance and support throughout this project. Thanks are also due to Dr. A. Obulesu, Professor and Head of the Department of Information Technology, and to the management of Vidya Jyothi Institute of Technology (An Autonomous Institution), Hyderabad, for the infrastructure and support that made this work possible.

REFERENCES

[1] S. Y. Choong Hong, C. C. Kang, J. D. Tan, and M. Ariannejad, "Smart Parking System Using IoT Sensors," Journal of Engineering Technology and Applied Physics, vol. 5, no. 1, pp. 63–69, Mar. 2023.

[2] F. A. A. Aziz, S. Z. M. Muji, M. H. A. Wahab, Z. Tukiran, C. Uttraphan, and N. Sudin, "Smart Parking System Mobile Application Using Ultrasonic Detector," International Journal of Integrated Engineering, vol. 14, no. 3, 2022.

[3] C. K. Gomathy, A. Vamsikumar, B. Karthik, and A. Purushothamreddy, "The Smart Irrigation System Using IoT," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 10, 2021.

[4] S. Shobana, B. S. Pandey, R. Padmashri, and U. Triveni, "IoT Based Smart Irrigation System Using Soil Moisture Sensor and ESP8266 NodeMCU," International Journal of Computer Science and Information Technology Research, vol. 9, no. 2, pp. 52–58, 2021.

[5] P. K. Sharma and M. I. Shukla, "IoT Based Smart Irrigation Monitoring System Using ESP8266 and Soil Moisture Sensor," in Proc. 7th Int. Conf. on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 670–675.

[6] "Smart Bin Using IoT," International Journal for Research in Applied Science and Engineering Technology (IJRASET), 2023, doi: 10.22214/ijraset.2023.52204.

[7] "IoT Based Intelligent Garbage Monitoring System," International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 2022.

[8] "An IoT-Based Smart Campus Monitoring System," IEEE Conference Publication, IEEE Xplore, 2022.

[9] "Developing an IoT Enabled Smart Classroom S3TH System," IEEE Conference Publication, IEEE Xplore, 2023.

[10] "Internet of Things Based Smart Campus – A Review," IET Conference Publication, IEEE Xplore, 2022.

[11] "Smart Home Using Blynk App Based on IoT," International Journal of Creative Research Thoughts (IJCRT), vol. 10, no. 5, May 2022.

[12] "IoT-Based Smart Parking Management System Using ESP32 Microcontroller," ResearchGate, 2022.

[13] R. Kumar, A. Singh, and M. Jain, "Cloud Based Smart Irrigation System Using IoT," International Journal of Advanced Research in Electronics and Communication Engineering, vol. 10, no. 5, pp. 45–50, 2022.

[14] "Waste Management 2.0: Leveraging Internet of Things for an Efficient and Eco-Friendly Smart City Solution," PMC, 2024.