



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Iot-Based Framework For Intelligent Indoor Environment Control And Occupancy Analysis In Smart Buildings

Prof. Farajana Nadaf
Asst. Professor
Computer Science and Engineering
KLS VEDIT, Haliyal, India

Kalpesh Pavaskar
Computer Science and Engineering
KLS VEDIT
Haliyal, India

Mushtaq Ahmed Jamali
Computer Science and Engineering
KLS VEDIT
Haliyal, India

Nagendra Borekar
Computer Science and Engineering
KLS VEDIT
Haliyal, India

Santosh Turamari
Computer Science and Engineering
KLS VEDIT
Haliyal, India

Abstract— Nowadays, application of Internet of Things (IoT) technologies in smart buildings for optimizing indoor environmental quality and enhancing energy efficiency has rapidly increased. This paper describes an intelligent control system for managing the indoor environment of a building and analyzing occupancy using IoT technology with Raspberry Pi 3B+, DHT11, VEML7700 Sensor, and a webcam. The system developed aims to automatically control and monitor environmental factors such as temperature, light intensity, and human presence through real-time image processing. The arrangement allows for environmental responsiveness; thus a fan can be turned on when the temperature is noted to exceed a set threshold, or an occupant is detected. An LED light can be activated when the level of luminosity is low or when an occupant is present. This system not only enhances comfort of the occupants but also ensures energy optimization. The sensor data is sent in real time to the ThingSpeak cloud platform, thus allowing for live monitoring and decision making based on set thresholds. Additionally, this data is harmonized with a local XAMPP server and kept in a MySQL database for later analysis. A custom web dashboard enables historical and live data trend analysis, providing stakeholders with valuable and prompt insights into room conditions and occupancy patterns. The paper outlines the financial feasibility and scalability of the framework, and considers possibilities for its use in larger smart building systems. Results from prototype implementation show the system's reliable monitoring of environmental changes and human presence detection, accurately automating responses with minimal error. This work, therefore, advances the less affordably IoT frameworks for intelligent building automation.

Keywords— Internet of Things(IoT), Smart Building Automation, Indoor Environment Control, Occupant

Detection, Raspberry Pi 3B+, Sensor based Automation, Real Time Monitoring, Energy Efficiency, ThingSpeak Integration, XAMPP Server.

I. INTRODUCTION

The Director's Welcome smart building cape town 21st century saw the rant of motion smart technological gadgets as buildings with automated systems that supervise and control numerous framework functions. Casey-2014 an indispensable smart building function is to sustain an indoor environment for its occupants while energising processes, at the same time enhancing a building's energy efficiency. Efficiency in 21st century modern systems usually arises due to properties set constraints/schedules, manual operations, increased energy wastes, and energy drains. IoT integration onto building control structures allows for systems to be constructed in a way that enables real-time monitoring, analysis, responding to stimuli, and intelligent automated reaction to changing environmental conditions and human presence. Building occupants experiencing air temperature, humidity, lighting, and air quality to be technologically referred to as Indoor environment quality (IEQ) have a direct essence for health, productivity, and comfort. Systems' proper occupancy detection can become crucial toward energy saving, operating solely when rooms are occupied. Such energy efficiency considered where algorithms adjust technology systems reactive to accurate detection of occupants active within structured managed spaces, despite numerous acquiring available restrictions deemed system solutions too complicated economical, and cumbersome to implement at scale famous commercial smart solutions for small and medium sized buildings.

The purpose of this paper is to develop IoT model that allows for monitoring of indoor environmental settings. It does so through a pragmatic and resourceful approach with regard to costs and expandability by enabling occupancy detection. Occupants can be monitored and counted using a USB webcam attached to the model. The webcam is also used

for real-time monitoring. Other components of the model include: a DHT11 sensor to monitor temperature and a VEML7700 sensor to measure light intensity. Our intelligent automation control model is constructed upon a multi-sensor framework.

If the temperature set by user is surpassed, the model turns ON a fan to prevent overheating. The system does this when an occupant is present. The same goes for dim light situations where an occupant is present within the room. The economical and environmental principles are best exemplified in this scenario since appliances that consume a great deal of energy are kept off until they are needed. In assisting remote access to data, the readings taken by the sensors are sent directly to ThingSpeak. This cloud-based IoT analytics platform allows real-time monitoring of temperature, humidity and luminosity through visual representation.

Furthermore, the information can also be stored locally on a MySQL database via a XAMPP server allowing integration with ThingSpeak to save data. A web-based dashboard has been developed to visualize this data, providing easy access to environmental trends and occupancy analytics over time.

II. LITERATURE REVIEW

The rise of Smart Buildings has evolved with the advancement of the Internet of Things (IoT) technologies - allowing monitoring and control of the buildings' indoor environments in real-time. There are numerous studies on the value IoT brings to the environment data collection processes like temperature, humidity, CO₂ concentration and light intensity to enhance energy conservation and promote comfort and health for the occupants. Accurate prediction of occupancy is one of the key challenges in smart building systems because it highly influences adaptive control over HVAC, lighting systems, security, and air quality needs of a building. Within their research, Floris et al. (2021) developed a complete IoT-based smart building system that merges environmental conditioning with machine learning to predict not only indoor conditions, but also the number of occupants in the room. Their proposed system is based on Raspberry Pi, with an array of inexpensive sensors, and a camera module for use in people detection.

Data uploaded to the cloud are managed with accompanying software that serves as the supervisor of resources responsible for data storage and processing. The implemented solutions provided lightweight machine learning approaches where an artificial neural network was adopted for classification and regression tasks. With these techniques, participants estimated the number of occupants in a room with a 99.5% accuracy and total volatile organic compounds (TVOCs) estimating Pearson correlation of 0.939. This research is unique because it not only addresses environmental monitoring but also includes real-time detection of occupants in order to optimize the energy consumption of the building. Previous work has generally centered on one element-either the collection of environmental data or detection of occupants with the result being limited conclusions for comprehensive building control. Hence, Floris et al. overcome an important deficiency with current systems: non-interoperability and modularity. Large numbers of most conventional building control systems are vertical, that is, operating stand-alone without compatibility for new technology integrations. Their horizontal platform accommodates multiple hardware and software elements with little configuration and offers scalability and flexibility for future enhancements. As another key contribution, they utilized cloud computing for processing data. A lot of previous solutions depended on edge-only computation, which made them ineffective for massive analytics and model training. Cloud-based platforms, however, offer strong computational resources, scalability,

and convenient access for developers and building administrators. Here, the virtual super entities employed in the cloud layer illustrate how sophisticated analytics can be broken down into modular components and run without relying on the strength of local hardware.

The COVID-19 pandemic has further highlighted the significance of indoor air quality and occupancy control. Floris et al.'s system aids this by monitoring air quality parameters in real-time and adapting ventilation strategies accordingly, thereby helping to create a healthier indoor environment. Their solution finds a middle ground between efficiency in energy consumption and health protection-a dilemma from which numerous contemporary building systems have been struggling today. Inspired by this philosophy, our system is based upon a comparable design but adapted toward smaller-scale cost-effective applications such as classrooms, laboratories, or small offices. We use Raspberry Pi 3B+, DHT11, and VEML7700 sensors for environmental monitoring and a simple yet efficient object detection model based on OpenCV and a camera for occupancy counting. In addition, we use ThingSpeak for real-time visualization and XAMPP for local database storage, ensuring accessibility and data security. In contrast to more sophisticated commercial systems, our solution focuses on affordability, modularity, and ease of implementation—making it ideal for educational and semi-professional settings. Our mission is to enable resource-constrained institutions to implement smart building concepts and maximize their indoor environments with easily accessible technologies

III. METHODOLOGY

This project utilizes a Raspberry Pi 3B+ microcontroller to design an efficient and smart indoor environmental monitoring system by connecting two primary sensors: DHT11 (temperature and humidity sensor), and VEML7700 (ambient light intensity sensor). The approach can be classified into several well-structured phases addressing design, implementation, data management, and output visualization.

System Design and Component Selection:

Component selection was made with consideration of cost effectiveness, ease of integration, and availability of the sensor:

Raspberry Pi 3B+: Was selected for its support of GPIO, processing power, and integration of Wi-Fi and Ethernet for network access.

DHT11 Sensor: Implemented for the measurement of temperature and humidity. It is adequate for indoor environments, although less accurate compared to more expensive options such as DHT22.

VEML7700 Ambient Light Sensor: Provides high dynamic range and sensitivity, making it best suited for the detection of small indoor light changes.

The sensors are powered directly via the 5V or 3.3V GPIO pins, with suitable resistors in place to provide stable readings.

Sensor Integration and Code Development:

Sensor modules were interfaced with the GPIO pins of the Raspberry Pi. Python was chosen because of its simplicity and strong support for libraries. The libraries that were utilized were:

Adafruit_DHT to retrieve temperature and humidity.

adafruit-circuitpython-veml7700 to detect ambient light intensity.

Python programs were written for:

- Initializing communication for sensors.
 - Retrieving sensor values after certain intervals of time (default: 10 seconds).
 - Displaying the data to the terminal.
 - Log the data locally and to external systems.
- Mechanisms for error handling were introduced via try-except blocks to make the system still run when a sensor temporarily fails.

C. Data Acquisition and Logging:

Both sensors' data are acquired every 10 seconds — a rate that strikes a balance between real-time and system load. Each reading includes:

- Temperature in Celsius (°C).
- Humidity as a percentage (%).
- Ambient light in lux.
- Timestamp of reading.

These are added into a structured form (JSON, CSV, or SQL ready strings) to local and cloud logs.

D. Local Data Storage using PHP & MySQL:

A local XAMPP server was implemented to save sensor data into a MySQL database utilizing the following building blocks:

PHP Script:

- Listens for POST requests.
- Extracts sensor data.
- Inserts the data into a pre-structured table.
- MySQL Database:
 - Includes a table with fields: id, temperature, humidity, light, timestamp.
 - Indexed on timestamp to allow fast querying for trends.

Advantages of local storage:

- Accessibility of data in the event of network failure.
- Protection through secure backups.
- Rapid testing and debugging.

E. Remote Monitoring with ThingSpeak:

Sensor data is also pushed to ThingSpeak via its REST API using unique API keys. Data is divided into different fields:

- Field 1: Temperature
- Field 2: Humidity
- Field 3: Light Intensity

The platform also has inbuilt visual analytics:

- Live plots
- Time range customizable filters
- Built-in MATLAB integration for more advanced analysis.

F. User Interface & Dashboard:

To make the system user-friendly:

- A simple web dashboard was built using HTML, CSS, Bootstrap and PHP.
- Displays current values, latest timestamp, and min/max Readings.
- Links to ThingSpeak for remote visual access.
- Can be accessed from any device on the local network.

G. Occupant's Prediction:

In order to carefully analyze and forecast room occupancy, we made a targeted investigation within our laboratory setting. For 15 continuous days, we gathered data systematically pertaining to different environmental and contextual factors affecting human occupancy of a closed space. The main goal of this data gathering initiative was to build a predictive model that could accurately estimate the number of people in the room at a particular point in time. This model allows us to assess if the lab is in use and, if it is, give an estimated figure of occupants in view of past trends.

The output of this forecast system has been implemented within an easily accessible web application in the form of a special section called "Occupants Prediction." This is where users may engage with the system by choosing a particular day and time through an easy-to-use interface. Once the input is submitted, the application runs it through and responds with an approximate number of occupants for the chosen duration, using the learned prediction model.

To further augment usability and accessibility, we have added a chatbot to the same section. The chatbot enables users to get occupancy forecasts via natural language interaction without manually choosing date and time inputs. For example, the user can ask, "How many will be in the lab at 3 PM on Wednesday?" And the chatbot will give the corresponding forecast.

This dialogue interface is intended to enhance user interaction, particularly for users who prefer the more natural, conversational interface. The combination of both manual and conversational input channels not only enhances the usability of the system but also demonstrates the synergy of uniting sensor-based analysis of data with cutting-edge AI-powered user interfaces. The method can be especially beneficial in smart building scenarios, where adaptive occupancy forecasting is an important driver of optimizing energy consumption, space allocation, and overall operational performance.

IV. DISCUSSION

This study showcases a hands-on approach to creating an intelligent indoor environment management system. It brings together hardware components like the Raspberry Pi 3B+, DHT11, and VEMLE7700 sensors, along with a computer vision-based occupant detection feature. By integrating these technologies within an IoT framework, we can achieve smooth monitoring, data collection, and responsive actions, resulting in a lively and adaptable indoor ecosystem. The design focuses on being modular and scalable, making it suitable for various indoor environments, from cozy homes to small office spaces. With a threshold-based control logic, the system smartly turns on fans and lights based on real-time data and occupancy, which not only saves energy but also boosts user comfort by ensuring that energy-hungry devices run only when needed.

To enhance usability, we implemented ThingSpeak as a real-time cloud dashboard for remote monitoring, while also using XAMPP for local storage to keep historical data for offline analysis. This combination of storage solutions ensures the system is robust and gives users flexibility in accessing their data. We also created a custom dashboard that presents real-time and historical sensor data, providing users with clear insights.

However, the system does have its limitations. The DHT11 sensor, while budget-friendly and easy to work with, lacks precision and has a slower response time. Additionally, the image processing algorithm for occupant detection may struggle in low light or when multiple people are partially hidden. These issues could be tackled in the future by incorporating machine learning-based image classification or opting for more accurate sensors.

What sets our system apart from similar studies is that we validated it in real-world conditions, rather than relying solely on simulations or static datasets, highlighting its practical application.

V. RESULTS



Fig 1: Geographical Visualization of Building Location System Dashboard

Take a look at Figure 1, which gives us as geographical view of the "Network for Energy and Internet of Things" dashboard. The main map, shown from a satellite perspective, highlights the "Building Position" with a bright red marker, providing a clear spatial reference. There's also a "Switch to Specific Map" button that invites users to dive deeper into the building's layout. The "KLS VDIT" logo marks the system provider, while the banner for "Network for Energy and Internet of Things" emphasizes the dashboard's focus on energy management and IoT integration. On the left sidebar, you'll find options for "Dashboard," "Building," "Camera," and "Analytics," showcasing a wider range of functionalities. At the bottom, the "Room Condition" section features date/time controls and a filter, suggesting that internal environmental data is readily available. All in all, this image highlights the system's ability to pinpoint the building's location within its network, while also hinting at additional features for detailed building insights, camera feeds, data analysis, and monitoring of indoor conditions.

Room	Temperature	Luminosity	Humidity	Head Count	Date & Time
Living Room	33.0 °C	53 Lux	55%	0	2025-04-24 22:08:45
Living Room	33.0 °C	53 Lux	55%	0	2025-04-24 18:30:53
Living Room	33.0 °C	53 Lux	55%	0	2025-04-24 18:28:54
Living Room	33.0 °C	53 Lux	55%	0	2025-04-24 00:12:21
Living Room	33.0 °C	52 Lux	55%	0	2025-04-23 23:47:36
Living Room	33.0 °C	53 Lux	55%	0	2025-04-23 23:46:39
Living Room	33.0 °C	53 Lux	55%	0	2025-04-23 23:46:16

Fig 2: Live Data Feed for Intelligent Indoor

Management

Figure 2 showcases a "Live Data Feed for Intelligent Indoor Management" dashboard, which offers real-time sensor readings for a specific area labeled as the "Living Room." The table features columns for "Room," "Temperature," "Luminosity," "Humidity," "Head Count," and "Date & Time,"

displaying a series of recent measurements. For the "Living Room," the "Temperature" consistently shows "30.0 °C" across the recorded timestamps. Likewise, the "Luminosity" remains steady at "5.0 Lux," and the "Humidity" holds at "50%." The "Head Count" column indicates "0" for all instances, suggesting that no one was detected in the living room during this timeframe. The "Date & Time" column provides a chronological log of these sensor readings, with updates occurring at short intervals, from 23:09:45 to 23:16:48 on "2025-04-24." This quick succession of updates emphasizes the "live" aspect of the data feed, which is essential for real-time monitoring and effective intelligent indoor management. Above the data table, the heading "Room Condition" clearly identifies this section of the dashboard. To the right, users can find date and time selection tools along with a "Filter" button, enabling them to choose a specific timeframe or apply filters to examine historical data or focus on particular conditions. In summary, Figure 2 provides a detailed snapshot of the current and very recent environmental conditions (temperature, luminosity, humidity) and occupancy (head count) of the designated "Living Room." The consistent readings over this brief period suggest a stable indoor environment, with no occupants present. The inclusion of filtering and time selection tools highlights the ability to analyze past data and identify trends for a more thorough approach to intelligent indoor management.

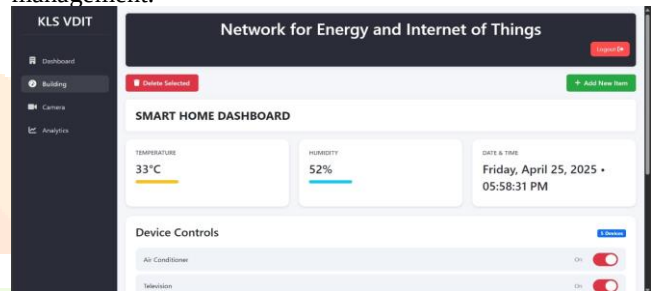


Fig 3: Integrated Smart Home Dashboard Interface

Figure 3 showcases an "Integrated Smart Home Dashboard Interface" as part of a "Network for Energy and Internet of Things" system, highlighted by the consistent branding of "KLS VDIT" and the top banner. On the left sidebar, the "Building" tab is currently selected. The dashboard's title, "SMART HOME DASHBOARD," makes its purpose crystal clear. You can see prominent tiles that display real-time (or a snapshot of) environmental conditions: "Temperature" reads "33°C" (with a yellow indicator) and "Humidity" shows "52%" (with a blue indicator). The "DATE & TIME" section indicates "Friday, April 25, 2025, 05:58:31 PM." Just below that, the "Device Controls" area allows you to easily manage connected devices with toggle switches for "Air Conditioner" (Off), "Television" (On), and "Lights" (On). There are also buttons for "Delete Selected" and "+ Add New Item," hinting at device management features. This interface brings together all the essential smart home information and controls into one convenient view, making it easy for users to keep an eye on environmental conditions and manage their connected appliances within the larger IoT and energy management network.

8. References

1. Floris, A., et al. (2021). An IoT-Based Smart Building Solution for Indoor Environment Management and Occupants Prediction. *Energies*, 14(10), 2959.
2. Atzori, L., et al. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805.
3. Yang, Z., et al. (2017). Smart Home System Design Using IoT and Cloud. *IEEE Conference*.
4. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767*.
5. DHT11 Sensor Datasheet. (n.d.).
6. VEML7700 Datasheet. (n.d.). Vishay Intertechnology.
7. Raspberry Pi Foundation. (n.d.). Raspberry Pi 3 Model B+ Specs. [<https://www.raspberrypi.com>]
(<https://www.raspberrypi.com>)
8. ThingSpeak Documentation. (n.d.). [<https://thingspeak.com>]
(<https://thingspeak.com>)
9. Adafruit. (n.d.). DHT11 & VEML7700 Setup Guides. [<https://learn.adafruit.com>] (<https://learn.adafruit.com>)
10. Mittal, A., et al. (2020). Smart Home Automation using Raspberry Pi. *IJCSIT*, 11(2), 245-249.
11. Kumar, A., & Bansal, A. (2019). IoT Based Real Time Smart Room Automation. *IJETT*, 67(9).
12. Roy, S., & Singh, P. (2018). Machine Learning Approaches for Occupancy Detection. *Sensors*, 18(11), 3711.
13. Salih, A., et al. (2020). Comparative Analysis of Humidity Sensors. *Journal of Measurement Science*, 45(2).
14. Sahu, S., et al. (2021). Smart Energy Systems with IoT and AI. Elsevier, *Energy Reports*.
15. Raj, P., & Buyya, R. (2019). Cloud IoT Paradigm. Springer.
16. Google Firebase Docs. (n.d.). IoT Applications. [<https://firebase.google.com>] (<https://firebase.google.com>)
17. AWS IoT Core Documentation. (n.d.). [<https://docs.aws.amazon.com>] (<https://docs.aws.amazon.com>)
18. Arduino & Raspberry Pi Integration. (2020). *Make: Magazine*.
19. Gupta, M., et al. (2021). Energy Efficient Smart Homes using IoT. Springer, *Smart Cities*.
20. Zhang, Y., et al. (2022). Deep Learning for Smart Buildings. *IEEE Access*.
21. Elhoseny, M., & Ramírez-Gallego, S. (2020). Edge Computing in Smart Homes. Springer.
22. Chatterjee, S. (2019). Computer Vision with OpenCV on Raspberry Pi. Packt Publishing.
23. Alam, M., & Reaz, M. (2021). IoT Smart Home: A Review of System Architecture and Implementation. Elsevier.
24. Feroz, A., et al. (2018). Data Logging in Smart Systems Using XAMPP and PHP. *Journal of Embedded Systems*.
25. Sharma, A., & Jain, R. (2020). Real-Time Environmental Monitoring. *IJERT*.
26. Bhatia, N., & Singh, R. (2021). Image Processing Techniques for Object Detection. IGI Global.
27. Rao, T., & Vasudevan, H. (2022). Python for IoT Projects. BPB Publications.
28. IEEE Smart Grid Initiative. (n.d.). [<https://smartgrid.ieee.org>] (<https://smartgrid.ieee.org>)
29. IBM IoT Platform. (n.d.). Cognitive IoT Systems. [<https://www.ibm.com/internet-of-things>]
(<https://www.ibm.com/internet-of-things>)
30. Raza, S., et al. (2020). Security Challenges in Smart Building Systems. *MDPI Electronics*, 9(7).

