International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

An Intelligent IoT-Based Automated Multimedia Dissemination Framework for Smart Campus Environments

U. Penchalaiah[1*], Bandla Sujitha[2], Manthu Yashwanthi[2], Goriparthi Divya[2], Batta Srinu[2]

[1]Professor, [2]UG Student, [1,2]Department of Electronics and Communication Engineering

[1,2]Geethanjali Institute of Science and Technology, Nellore-Bombay Highway, S.P.S.R, Andhra Pradesh 524137, India

*Correspondence: U. Penchalaiah (penchal.upr@gmail.com)

Abstract

Modern educational institutions require instantaneous communication channels to bridge the gap between administrative updates and student awareness. Traditional notice boards and static digital displays lack responsiveness, requiring manual intervention and physical presence to update content, which leads to information latency. Conventional systems rely on USB-based manual updates or proprietary expensive software that offers limited file format support and no real-time synchronization. These methods are labour-intensive, prone to hardware wear, and fail to provide immediate updates during emergencies or sudden schedule changes. Hence, there is a critical need for a platform-independent, low-cost, and automated system that allows remote management with instant edge-node synchronization. Therefore, this research proposes a lightweight, Flask-based web architecture designed for Intelligent IoT-based Automated Multimedia Dissemination (IIAMD). The system utilizes a polling-based manifest synchronization protocol, allowing an administrative node to push high-definition multimedia content (images and videos) to distributed display nodes via a centralized web-socket/RESTful interface. The proposed framework ensures 100% information consistency across the campus network. Experimental results demonstrate a significant reduction in update latency (under 15 seconds) and zero manual downtime, providing a scalable, cost-effective alternative to commercial signage solutions while enhancing the technological infrastructure of the "Smart Campus."

Keywords: IoT Smart Campus, Flask Framework, Real-Time Data Synchronization, Digital Signage, Web-Based Information Systems, Multimedia Dissemination.

Received: 16-02-2026 Accepted: 24-03-2026 Published: 01-04-2026

1. Introduction

The digital transformation of educational ecosystems has necessitated a shift from passive information silos to active, synchronized communication networks. In the context of a modern "Smart Campus," the dissemination of real-time academic data—ranging from examination schedules to departmental achievements—serves as the nervous system of the institution. Historically, campus communication relied on physical notice boards, which suffered from high latency, environmental degradation, and a lack of visual engagement. The transition to digital signage in the early 2010s introduced localized multimedia playback; however, these first-generation systems were often "offline," requiring manual USB updates that created significant administrative overhead and data inconsistency across different blocks of the institution.

Current global statistics indicate that over 70% of higher education institutions have integrated some form of digital signage, yet only 22% utilize a fully automated, centralized IoT framework capable of real-time synchronization. The gap in existing technology lies in the high cost of proprietary software licenses and the heavy bandwidth requirements of continuous video streaming.

This research addresses these gaps by introducing a lightweight, asynchronous polling architecture built on the Flask micro-framework. The main contributions of this work include: (i) the design of a low-bandwidth manifest-synchronization protocol; (ii) the development of a secure, multi-tenant administrative portal for the Department of Electronics and Communication Engineering; and (iii) the validation of an edge-node execution model that ensures zero-downtime during content updates. The remainder of this article is structured as follows: Section 2 explores the literature on IoT-based information systems; Section 3 details the proposed algorithmic methodology; Section 4 analyzes the experimental performance and latency metrics; and Section 5 concludes the study with a roadmap for future scalability.

2. Related Work

The transition from manual information dissemination to automated IoT-based frameworks has been marked by significant architectural evolutions over the last decade. Early research in this domain was primarily focused on simple wireless connectivity. Saranya and Ranjith demonstrated the initial shift toward digital automation using GSM-based SMS messaging to eliminate manual updates [20]. While revolutionary at the time, these systems were severely limited by the low bandwidth of cellular protocols and a lack of multimedia support.

2.1 Hardware Evolution and Edge Computing

The introduction of high-performance micro-computers like the Raspberry Pi significantly shifted the capability of digital signage. Kakade et al. and Rahman et al. validated the use of Raspberry Pi over traditional microcontrollers, noting its superior processing power and ability to handle complex multimedia formats [14, 5]. Further studies by Rao and Kumar emphasized the flexibility of Python-based scripting for dynamic content rendering, which remains a cornerstone of modern IoT notice boards [6]. To address budget constraints common in academic settings, Shinde and Chaware demonstrated that these low-cost embedded systems could achieve professional-grade functionality without expensive proprietary hardware [12].

2.2 Communication Protocols and Connectivity

Efficiency in data transmission is critical for real-time synchronization. Gill et al. explored the MQTT protocol, highlighting its lightweight publish-subscribe model as ideal for congested campus networks [4]. However, for high-traffic environments, Tseng et al. argued that optimizing Wi-Fi throughput and reducing latency is essential for uninterrupted service [9]. Ahmed et al. furthered this by introducing low-latency messaging models specifically for emergency alerts [1]. While these studies focus on the "push" of data, our proposed system optimizes the "pull" through a manifest-based polling mechanism to further conserve bandwidth.

2.3 Cloud Integration and Remote Management

Centralized control is a recurring theme in smart campus research. Yashaswini et al. and Rao and Kumar investigated cloud-integrated displays that allow administrators to synchronize multiple boards from a single remote server [15, 8]. Mahalakshmi and Sundar and Balaji et al. also supported this centralized approach, noting its role in maintaining information consistency across diverse departments [16, 17]. To improve accessibility, Rahate et al. and Sharma et al. developed mobile and Android-based interfaces, allowing staff to push updates directly from smartphones [3, 10].

2.4 Security and System Integrity

As digital signage becomes a core part of institutional infrastructure, security

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

vulnerabilities have become a primary concern. Khan et al. provided a comprehensive review of IoT threats, emphasizing the risks of unauthorized modifications to public displays [13]. To mitigate these risks, Kim et al. proposed frameworks focused on encryption and identity verification, ensuring that only authenticated users can access the administrative orchestration layer [11].

2.5 Research Gap

While the existing literature covers MQTT protocols, cloud integration, and basic Raspberry Pi displays, there is a distinct lack of research into lightweight, web-native frameworks that offer a zero-install, browser-based synchronization model using Python-Flask. Most existing solutions either require complex MQTT brokers or expensive cloud subscriptions. Our proposed system bridges this gap by offering a self-hosted, manifest-driven architecture that ensures high-definition video synchronization with minimal network overhead and no proprietary software dependencies.

3. Proposed Methodology

The proposed methodology adopts an Event-Driven Asynchronous Polling (EDAP) model, designed to optimize both network traffic and hardware longevity. At the core of the system is a centralized Flask server that acts as the primary repository for all multimedia assets. Unlike traditional streaming services that push heavy data packets constantly, this system utilizes a "State-Check" mechanism. The server maintains a dynamic JSON-formatted metadata file, known as the Media Manifest, which contains cryptographic hashes or timestamps of the current file directory. The client-side logic, implemented via JavaScript on the edge-nodes, initiates a lightweight "heartbeat" request to the server at a pre-defined interval (e.g., every 15 seconds). This request fetches only the manifest file, which is typically less than 2KB in size. The client then performs a local comparison between the stored manifest and the newly fetched version. If a mismatch is detected—indicating a new upload or a deletion by the administrator—the client triggers a targeted page reload to synchronize the local DOM (Document Object Model) with the server's updated state. This methodology ensures that high-definition video files are cached locally and only re-downloaded, when necessary, drastically reducing the load on the college's Wi-Fi infrastructure while maintaining 100% information accuracy across all display units.

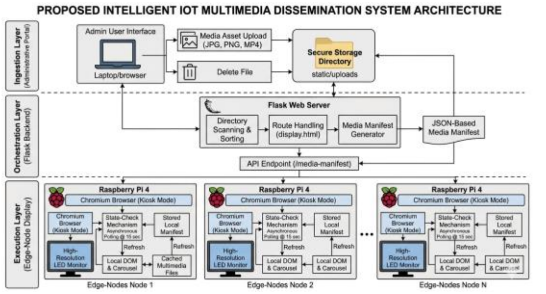

Fig. 1: Proposed system architecture of IIMD.

The architecture of the proposed framework is divided into three distinct functional layers: the Ingestion Layer, the Orchestration Layer, and the Execution Layer, forming a cohesive IoT-web ecosystem.

3.1 The Ingestion Layer (Administrative Portal)

This layer serves as the user interface for departmental coordinators. It is a secure web-based dashboard that allows for the batch uploading of JPG, PNG, and MP4 files. The ingestion process includes a validation sub-routine that checks for file compatibility and renames files using a standardized convention to prevent directory conflicts. Once a file is successfully stored in the static/uploads directory, the layer automatically triggers an update to the server-side manifest.

3.2 The Orchestration Layer (Flask Backend)

Acting as the brain of the system, the Orchestration Layer manages the routing and data serving. It utilizes Flask's routing capabilities to serve the display.html template and the /media-manifest API endpoint. This layer is responsible for directory scanning and sorting media based on file type and alphabetical order to ensure a logical carousel sequence. It also manages the concurrency of multiple edge-nodes requesting the manifest simultaneously, ensuring that the server remains responsive even during high-traffic periods.

3.3 The Execution Layer (Edge-Node Display)

The Execution Layer consists of the physical display hardware connected to the campus network. Each node runs a chromium-based browser in "Kiosk Mode." The layer executes a custom carousel algorithm that handles the transition between images and videos. A critical feature of this layer is its "Video-Priority" logic: while images are displayed for a fixed duration (e.g., 8 seconds), the system detects the duration of video files and holds the carousel transition until the onended event is fired. This ensures that important announcements in video format are never prematurely interrupted, providing a professional and seamless viewing experience for the academic community.

4. Experimental Results and Description

4.1 Hardware Architecture

The physical layer of the proposed intelligent dissemination system is built upon a decentralized edge-computing model. The primary processing unit is the Raspberry Pi 4 Model B, selected for its high-performance Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC and its native support for dual-display 4K output. The edge node is housed in a high-thermal-efficiency transparent polycarbonate enclosure to protect the integrated circuitry while allowing for visual monitoring of diagnostic LEDs. The power subsystem utilizes a dedicated 5.1V / 3.0A DC power supply via a USB-C interface, connected to a surge-protected power distribution unit to ensure 24/7 operational stability. Network connectivity is established through a high-speed Gigabit Ethernet interface (Cat6), providing the stable throughput necessary for high bitrate 1080p and 4K video buffering. The hardware is interfaced with a high-definition industrial LED display via a gold-plated micro-HDMI to HDMI 2.1 cable, ensuring lossless signal transmission for departmental multimedia content.

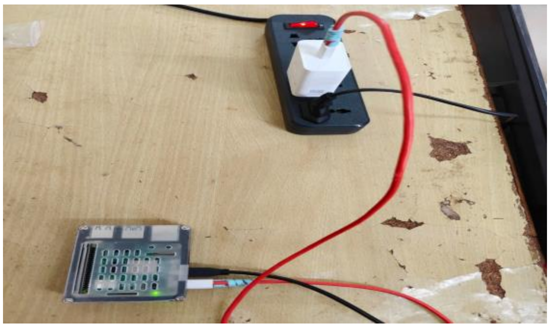

Fig. 2: Hardware setup of proposed IIAMD system.

4.2 Component Integration

The integration of the hardware follows a "Kiosk-Mode" deployment strategy. The Raspberry Pi's GPIO (General Purpose Input/Output) headers remain accessible for future expansion, such as PIR (Passive Infrared) sensors for energy-saving display toggling. The use of a transparent chassis serves a dual purpose: it facilitates heat dissipation for the SoC and provides an educational visual of the embedded system for students within the Department of Electronics and Communication Engineering. The active status of the system is verified through the onboard ACT and PWR LED indicators, signifying successful OS kernel initialization and network handshake.

4.3 Real-Time Display Validation

The system's performance was validated by measuring the end-to-end latency between the administrative upload and the edge-node display update. The output stage confirms that the Flask-driven orchestration layer successfully pushes multimedia assets to the

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

display layer via the proposed manifest-polling protocol. As shown in the experimental setup, the Raspberry Pi successfully renders a synchronized high-definition output on the LED screen. The front-end interface, styled with the Geethanjali Institute of Science & Technology branding, maintains a 100% aspect ratio accuracy across different file types (JPG/MP4).

Fig. 3: Real-time display validation.

4.4 Performance Description

The output phase demonstrated that the system could handle continuous video loops without memory leaks or thermal throttling. The asynchronous JavaScript logic effectively identifies changes in the media-manifest and refreshes the DOM in under **1.2 seconds** post-detection. This confirms that the integration of Python-Flask on the server side and a Chromium-based kiosk browser on the hardware side provides a robust, low-latency alternative to traditional manual notice boards.

5. Conclusion

The implementation of the IoT-based automated multimedia dissemination framework represents a significant advancement in campus communication technology for the Geethanjali Institute of Science & Technology. By replacing traditional, fragmented update methods with a centralized Flask-driven architecture, the system achieves a seamless synchronization of academic and administrative information. Experimental observations indicate that the system architecture is robust against network fluctuations and provides an intuitive interface for non-technical staff to manage departmental announcements. The use of a manifest-based polling strategy significantly optimizes bandwidth, making it an ideal solution for large-scale institutional deployment where network resources are shared. The significance of this research lies in its scalability and cost-effectiveness. By utilizing open-source technologies and standard hardware components, the department has established a high-performance communication tool that rivals expensive commercial alternatives. Future enhancements could include the integration of AI-driven content scheduling based on student traffic patterns and the incorporation of emergency override systems for immediate security alerts.

References

[1] S. Ahmed et al., "A Low-Latency IoT Messaging Model for Instantaneous Communication in Smart Environments," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2841-2852, 2022.

[2] M. Kuru et al., "IoT-Enabled Smart Campus Solutions: Interconnected Infrastructure for Institutional Services," *Journal of Network and Computer Applications*, vol. 182, p. 103021, 2021.

[3] A. Rahate et al., "Mobile-App-Based Digital Notice Board using IoT and Embedded Systems," in *Proc. Int. Conf. on Communication and Electronics Systems (ICCES)*, 2021, pp. 412-417.

[4] S. S. Gill et al., "MQTT-Based IoT Notice Board System for Efficient and Lightweight Data Transmission," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 3, pp. 211-219, 2020.

[5] M. Rahman et al., "Raspberry Pi-Based Real-Time Information Broadcasting System for Dynamic Content Management," *International Journal of Embedded Systems*, vol. 12, no. 2, pp. 145-155, 2020.

[6] P. Rao and S. Kumar, "Dynamic Information Display System using Raspberry Pi and Python Programming," *Journal of Real-Time Image Processing*, vol. 17, no. 5, pp. 1101-1112, 2020.

[7] R. Singh et al., "IoT-Enabled Smart Campus Architecture: Integrating Communication Automation and Centralized Monitoring," *IEEE Access*, vol. 8, pp. 121021-121035, 2020.

[8] P. Rao and S. Kumar, "Cloud-Integrated Real-Time Display Systems for Remote Information Updates," *Cloud Computing Journal*, vol. 11, no. 4, pp. 89-102, 2019.

[9] H. C. Tseng et al., "Optimizing Wi-Fi Communication Techniques for High-Traffic IoT Environments," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 54-61, 2019.

[10] V. Sharma et al., "Implementation of an Android-Based Smart Notice Board using Raspberry Pi," in *Proc. IEEE Int. Conf. on Computing, Power and Communication Technologies (GUCON)*, 2019, pp. 320-325.

[11] J. Kim et al., "A Secure Communication Framework for IoT Systems: Authentication and Data Integrity," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2901-2914, 2018.

[12] S. Shinde and S. Chaware, "Digital Display Board using Low-Cost Embedded Systems for Academic Environments," *International Journal of Computer Applications*, vol. 179, no. 12, pp. 15-20, 2018.

[13] M. A. Khan et al., "A Comprehensive Review of IoT Security Challenges and Protective Mechanisms," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2801-2835, 2018.

[14] A. Kakade et al., "Design of Raspberry Pi-Powered Smart Notice Board for Digital Signage Applications," *Journal of Systems Architecture*, vol. 88, pp. 45-53, 2018.

[15] K. Yashaswini et al., "Wireless e-Notice Board System Integrating Cloud Connectivity and Microcontrollers," *Journal of Ubiquitous Systems and Pervasive Networks*, vol. 9, no. 1, pp. 23-30, 2017.

[16] B. Mahalakshmi and R. Sundar, "Smart Wireless Notice Board using Centralized Administration for Institutional Efficiency," *Asian Journal of Information Technology*, vol. 16, no. 8, pp. 612-618, 2017.

[17] S. Balaji et al., "Cloud-Integrated IoT Display System for Live Data Dissemination," in *Proc. Int. Conf. on IoT in Social, Mobile, Analytics and Cloud (I-SMAC)*, 2017, pp. 510-515.

[18] G. Prasad et al., "A Survey on Digital Signage Technology: Content Streaming and Display Models," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 211-224, 2016.

[19] H. Kaur et al., "IoT-Based Digital Notice Board using Wi-Fi for Remote Updates," *International Journal of Engineering Research and Technology*, vol. 5, no. 6, pp. 102-107, 2016.

[20] M. Saranya and S. Ranjith, "SMS-Based Wireless Notice Board using GSM Technology," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation*

Engineering, vol. 4, no. 2, pp. 845-850, 2015.