IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Powered Smart Notice Board with Chatbot Integration Using Raspberry Pi, Django, And Rasa

[1]Dr.C.V. Subhaskara Reddy, [2]S. Vinay Kumar, [3]C. Surendra, [4]P. Sreenivasulu

[1]Professor of ECE, [2]Student, [3]Student, [4]Student

ECE Department,

S.V.R Engineering College, Nandyal, Andhra Pradesh, India.

Abstract: In modern educational institutions, effective communication is a key pillar of administrative efficiency. Traditional notice boards, often paper-based and manually updated, pose significant limitations in terms of scalability, timeliness, and environmental sustainability. This paper presents the design and implementation of an AI-powered smart notice board system that addresses these challenges through automation, multimedia integration, and conversational AI.

The proposed system is built around a Raspberry Pi 4 platform, functioning as a compact and affordable local server. It hosts a Django-based web application that allows authorized administrators to upload and manage notices in the form of text, images, and videos. These notices are dynamically rendered on a connected HDMI display in a continuous loop. The system is further enhanced by the integration of a Rasa-powered chatbot, embedded within the display interface, which enables real-time interaction with users. The chatbot is trained to handle frequently asked academic queries, including examination schedules, project deadlines, and placement updates, thereby reducing repetitive student–faculty interactions.

Designed to operate fully offline, the system is ideal for deployment in environments with limited network infrastructure. It emphasizes paperless communication, user interactivity, and real-time responsiveness. Extensive testing confirms the system's stability, ease of use, and potential for scalability across departments and institutions. This work contributes to the ongoing digital transformation of educational infrastructure, combining IoT, web technologies, and natural language processing into a cohesive smart campus solution.

***Index Terms* -** Smart Notice Board, Raspberry Pi, Django, Rasa Chatbot, Artificial Intelligence**.**

I. Introduction

In the age of digital transformation, educational institutions are increasingly seeking smart and sustainable alternatives to conventional infrastructure. One of the most overlooked components in this modernization is the traditional notice board — a static, manually updated medium used to disseminate academic, administrative, and event-related information. These physical boards are constrained by inherent limitations, such as delayed updates, dependency on human intervention, and the recurring costs and environmental concerns associated with paper-based communication.

The advent of embedded computing, wireless connectivity, and artificial intelligence presents an opportunity to reimagine notice boards as intelligent, real-time communication systems. This paper introduces a novel approach to information dissemination on academic campuses through the development of an **AI-powered smart notice board integrated with a chatbot assistant**. The proposed solution is designed to automate content delivery, support multimedia formats, and provide conversational query resolution — all hosted locally on a cost-effective and power-efficient Raspberry Pi 4.

The backend of the system is powered by the Django web framework, which provides a secure admin interface for uploading notices. The frontend display dynamically renders these notices — whether text, image, or video — on a full screen HDMI monitor. A JavaScript-based logic controls the automated rotation of notices, ensuring visibility in a looped cycle. The interactivity of the system is enhanced through the integration of a Rasa-based chatbot. This chatbot responds to user queries in real time, handling frequently asked questions related to examination schedules, project deadlines, and placement updates, thus reducing the burden on faculty and administrative staff.

A key feature of the system is its **offline operability**, allowing seamless deployment in institutions with limited or no internet connectivity. The architecture is designed to be modular, scalable, and eco-conscious, making it suitable for use in both urban and rural educational settings. Through the integration of IoT, web development, and natural language processing (NLP), the smart notice board serves not only as a digital announcement platform but also as an intelligent assistant — contributing meaningfully to the concept of smart campuses.

The following sections present a detailed review of related work, the methodology and architecture of the proposed system, implementation specifics, performance evaluation, and future enhancement possibilities.

II. LITERATURE REVIEW

The increasing digitization of educational ecosystems has accelerated the adoption of smart infrastructure, particularly in the domain of communication and information dissemination. Notice boards, though widely used in academic institutions, often remain static, paper-based, and manually updated. This section reviews existing work and technological advancements that inform the development of an AI-integrated smart notice board system. It also identifies key research gaps that the proposed system aims to address.

A. Traditional and Digital Notice Boards

Conventional notice boards rely on physical posting and manual updates, which pose challenges in terms of timeliness, accessibility, and environmental sustainability. These systems are vulnerable to human error, delayed communication, and often lead to the omission or redundancy of important information. Recent studies have proposed electronic notice boards to partially mitigate these issues. For instance, Katapur et al. [1] introduced an IoT-enabled digital notice board using Raspberry Pi 3, allowing messages to be updated remotely through a mobile application. While effective in eliminating paper dependency, their system was limited to text-based content and lacked real-time interactivity or AI integration.

More advanced systems have attempted to incorporate multimedia content, such as images and videos, displayed on LED or LCD panels. However, these systems are often controlled by proprietary platforms, are expensive to deploy at scale, and remain passive in terms of user interaction. As a result, a gap persists in delivering two-way communication within such notice board systems.

B. IoT and Embedded Devices in Campus Communication

The proliferation of embedded devices and low-cost microcomputers like the Raspberry Pi has enabled the implementation of IoT-based solutions in education. These devices support flexible, programmable control of displays and can serve as local web servers, reducing dependency on cloud infrastructure. Research by Arulmurugan et al. [2] leveraged GSM technology to control display units from a mobile interface, showcasing how IoT can improve accessibility and management efficiency.

Despite such advancements, many of these systems rely on limited communication methods such as SMS or Bluetooth, which constrain the range, data transfer capacity, and real-time responsiveness of the solution. Furthermore, most IoT-based notice boards are designed for one-way broadcasting of information and do not allow for intelligent, user-driven interaction.

C. Role of Chatbots and Conversational AI in Education

Recent years have witnessed growing adoption of AI-driven chatbots in educational institutions, primarily for automating administrative responses, FAQs, student counseling, and support services. Chatbots are designed to simulate human-like conversations using predefined intents and natural language understanding (NLU). Research indicates that chatbots reduce workload on faculty and administrative staff while enhancing engagement through 24/7 accessibility [3].

Most chatbot deployments in academia are web-based or mobile-based assistants. However, integration of such systems into physical infrastructures like digital signage remains rare. The present work builds upon this by embedding a Rasa-based chatbot directly into a digital notice board interface, enabling real-time user interaction at the display level, even in offline environments.

D. Raspberry Pi as a Platform for Smart Displays

Raspberry Pi has been widely adopted in educational and research settings due to its affordability, GPIO flexibility, and ability to host lightweight operating systems. Its support for Python and compatibility with open-source tools make it an ideal candidate for rapid prototyping of smart systems. Researchers have implemented Raspberry Pi-based solutions in areas ranging from home automation to industrial displays. Its application in digital signage, however, remains relatively unexplored, particularly in conjunction with full-stack web frameworks and AI models.

The proposed system leverages Raspberry Pi 4's increased RAM and processing capabilities to simultaneously run a Django web server and a Rasa NLP engine, thus integrating content management with conversational AI in a single-device architecture.

E. Django and Rasa: Synergistic Application Stack

Django is a widely used high-level Python web framework designed for rapid development and clean, pragmatic design. It offers built-in support for admin panels, media handling, and database integration — features essential for managing a digital notice system. Rasa, on the other hand, is an open-source conversational AI framework that supports intent classification, entity recognition, and rule-based or machine-learned dialogues [4][5].

While literature documents many systems using Django or Rasa independently, very few have explored their combination in embedded deployments. This research bridges that gap by integrating both frameworks into a unified architecture that operates on a Raspberry Pi, delivering a full-stack, offline-capable smart notice board with embedded chatbot functionality.

F. Identified Gaps and Contributions

Despite significant research into digital notice boards, IoT-based systems, and chatbot technology, several limitations persist in prior implementations:

- Most systems lack multimedia content support (image/video).
- Chatbots are rarely integrated into public digital displays.
- Offline-capable, full-stack deployments on Raspberry Pi remain underexplored.
- Content management is often external, rather than local and admin-controlled.
- Few systems achieve modularity with both AI and web integration.

The proposed system addresses these gaps by offering:

- A hybrid platform combining Django and Rasa, hosted locally on Raspberry Pi.
- Media-rich content rendering and automatic notice rotation.
- Integrated conversational AI to respond to student queries.
- Full offline functionality and local admin control.
- Scalability for multi-display or multi-campus deployments.

III. PROPOSED METHODOLOGY

The methodology adopted in this work integrates both hardware and software components to create a self-contained, interactive smart notice board system. The solution is designed to be modular, cost-effective, and fully operable in offline environments. The system allows administrators to upload and manage multimedia notices, while students can interact with an embedded AI chatbot for real-time query resolution. This section describes the overall architecture, hardware configuration, software stack, and key system functionalities.

A. System Architecture

The proposed system follows a layered architecture composed of the following four key layers:

1. **Hardware Layer**: Consists of Raspberry Pi 4 as the core computing unit, connected to an HDMI monitor for visual output, powered by a regulated 5V/3A adapter.
2. **Backend Layer**: Developed using the Django web framework, it manages user authentication, notice uploads, media storage, and database operations.
3. **Frontend Layer**: Built using HTML5, CSS3, and JavaScript, it renders notices in a rotating display loop and hosts the embedded chatbot widget.
4. **AI Layer**: The Rasa framework is used to power a rule-based conversational AI chatbot, trained to answer domain-specific academic queries.

These layers are deployed on a single Raspberry Pi device, ensuring a compact and offline-operable setup. The architecture supports scalability and modular replacement of components without affecting the core workflow.

B. Hardware Components

The system is implemented using the following hardware:

- **Raspberry Pi 4 Model B (2GB)**: Acts as the server and processing unit. It is capable of running both the Django server and Rasa chatbot simultaneously.
- **HDMI Monitor**: Used to display the notices and chatbot interface. The monitor runs a Chromium browser in kiosk mode.
- **MicroSD Card (32 GB)**: Stores the Raspberry Pi OS, Django application, chatbot models, and uploaded media files.
- **Power Supply (5V/3A)**: Ensures stable operation, especially during simultaneous media rendering and AI response generation.

Peripheral devices such as a keyboard and mouse are used during initial setup but are not required for deployment.

C. Software Stack and Backend Configuration

The backend is powered by Django, a Python-based web framework that provides a built-in administrative panel for authenticated users. Administrators can log into the Django panel and upload notices in one of three formats: plain text, image (JPEG/PNG), or video (MP4). Each notice is categorized and timestamped in a SQLite3 database, and media files are stored in a structured filesystem under the /media/ directory.

Django's model-view-template (MVT) architecture ensures a clean separation between data models, display templates, and application logic. Static and media files are served via Django's built-in static file handler.

D. Frontend Rendering and Display Loop

The frontend interface, accessed through a browser, renders the uploaded notices dynamically using Django templates. A JavaScript script cycles through each notice type:

- **Text notices** are displayed for 10 seconds.
- **Image notices** for 12 seconds.
- **Videos** are played based on their duration with a 2-second buffer.

Each notice is placed inside a styled <div> element, and the JavaScript loop automatically transitions between them. This creates a seamless and automated display without requiring manual input or reloading.

E. Chatbot Integration and NLP Pipeline

A custom chatbot, named *NoticeBot*, is developed using the open-source Rasa framework. The bot is trained on intents specific to academic environments such as:

- ask_exam_schedule
- ask_project_deadline
- ask_placement_updates

Training data is defined in YAML format (nlu.yml), responses are stored in domain.yml, and rules for response logic are in rules.yml. The chatbot runs on localhost:5005 and communicates with the frontend widget via WebSocket.

The chatbot is embedded into the HTML template using a third-party widget that sends and receives messages in real-time. It is initialized with a greeting and remains docked in the corner of the screen, available throughout the display session.

F. Offline Operability and Auto-Boot Configuration

One of the key features of the system is its ability to operate **completely offline**. Once configured, the Raspberry Pi does not require internet access. System services (systemd) are used to automatically launch:

- Django server (python manage.py runserver)
- Rasa server (rasa run --enable-api)
- Chromium browser in kiosk mode (chromium-browser --kiosk)

This ensures the system is resilient to power interruptions. Upon reboot, the notice board resumes full operation without manual intervention.

G. Functional Workflow Summary

1. Admin uploads notice via Django panel.
2. Data is stored in SQLite and /media/.
3. Django renders HTML with all notices.
4. JavaScript handles rotation display.
5. Chatbot widget auto-starts on page load.
6. User enters query; chatbot replies in real-time.
7. System loops endlessly until powered off.

This unified pipeline of hardware and software ensures the system delivers both information and interaction in a single, efficient interface.

VI. IMPLEMENTATION AND RESULTS

This section presents the real-world deployment and evaluation of the proposed AI-powered smart notice board system. The implementation was carried out on a Raspberry Pi 4 with all system modules—web server, chatbot engine, media handling, and user interface—configured to work together in an embedded environment. The objective of this stage was to validate the system's usability, performance, and offline operability under realistic conditions within an academic setting.

A. Hardware Deployment and Setup

The system was assembled using a Raspberry Pi 4 (2GB RAM) as the core computing device. A 32GB microSD card was flashed with Raspberry Pi OS and used to store the Django project, Rasa chatbot files, and the media content database. The Pi was connected to a 21.5-inch HDMI monitor, serving as the primary display for both the rotating notice system and the chatbot widget.

A regulated 5V/3A power supply ensured uninterrupted performance. On initial setup, a USB keyboard and mouse were connected to configure system parameters such as Wi-Fi, SSH, and time synchronization. The Raspberry Pi was then configured to auto-launch all services using systemd, enabling the system to self-recover after any power outage.

Figure 1. Raspberry Pi 4 2GB, Micro SD Card 32 GB

B. Web Backend Usage and Admin Interface

Once deployed, administrators could access the Django backend through a secure login panel on the local network. The admin dashboard provided the capability to create, modify, and delete notices. Three notice types were tested:

- **Text-based notices**, such as announcements and instructions.
- **Image-based notices**, such as posters and infographics.
- **Video notices**, such as short clips introducing events or placements.

Uploaded content was instantly stored in the SQLite3 database and displayed on the frontend interface without requiring system reboot or refresh. The ease of use of the admin panel significantly reduced the time and complexity involved in managing notice content.

C. Frontend Behaviour and Media Display

The frontend was launched in full-screen kiosk mode using Chromium browser on the Raspberry Pi. It dynamically rendered notices using Django's templating engine and handled transitions using JavaScript timers. Each notice was displayed based on its type:

- Text notices remained for 10 seconds.
- Images displayed for 12 seconds.
- Videos were shown for their full duration with a short buffer for transition.

The interface maintained smooth transitions and readability from distances of up to 4 meters, making it suitable for installation in common areas like corridors and department entrances. The continuous, looping behavior ensured that no content was missed during idle periods.

D. Chatbot Response and Interaction Flow

The chatbot widget, powered by Rasa and embedded into the display interface, was initialized with a greeting and remained active throughout the session. It handled queries such as:

- "When are the final exams?"
- "What is the last date for project submission?"
- "Any placement drives this month?"

Each query was matched with a pre-trained intent, and the corresponding response was returned within 1–2 seconds. The system operated on the local Rasa server without internet, ensuring reliability in offline conditions.

Fallback scenarios were also tested. When a user entered an unknown query, the chatbot responded with a predefined fallback message asking the user to rephrase the question. The overall interaction was intuitive and engaging, even for users unfamiliar with chatbots.

E. Observations and Performance Evaluation

The system was continuously tested for over 10 hours without any performance degradation or overheating of the Raspberry Pi. Key observations include:

- **Responsiveness**: The notice display was fluid and updated in real-time upon content changes.
- **Stability**: The system remained functional after simulated power failures, resuming all services upon reboot.
- **Ease of Use**: Non-technical users, including students and faculty, found the admin panel and chatbot easy to use.
- **Offline Functionality**: All features remained operational without internet access.

F. Screenshots and Visual Outputs

Figure 3. Notice Board with different media. Figure 4. Notice Board with AI Chatbot.

V. CONCLUSION

This paper presented the design and implementation of an AI-powered smart notice board system that effectively addresses the limitations of traditional communication mechanisms in academic environments. The proposed system replaces static, manually operated notice boards with a dynamic, multimedia-enabled, and interactive digital solution that integrates web technologies, conversational AI, and embedded computing. By leveraging a Raspberry Pi 4 as the central control unit, the system successfully hosted both a Django-based web server for notice management and a Rasa-based chatbot engine for real-time user interaction. The administrator interface enabled secure and immediate uploading of text, image, and video-based notices, while the frontend provided a seamless user experience through automated notice rotation. The embedded chatbot further enhanced usability by responding to frequently asked academic queries, thereby reducing the administrative burden on faculty and improving information accessibility for students.

The system functioned effectively in offline environments, which is especially beneficial for deployment in rural or resource-constrained institutions. It was also designed to be modular, cost-efficient, and easily scalable. Testing confirmed that the system is stable, user-friendly, and highly effective for real-world applications.

In essence, the project contributes to the growing field of smart campus technologies by merging automation, multimedia communication, and artificial intelligence into a unified, reliable, and eco-friendly platform.

References

[1] M. S. Katapur, E. Hadapada, and M. G. Kattimani, "IoT Based Digital Notice Board using Raspberry Pi," *International Journal of Engineering Research and Management (IJERM)*, vol. 7, no. 8, pp. 19–22, Aug. 2020.

[2] S. Arulmurugan, S. Anitha, A. Priyanga, and S. Sangeethapriya, "Smart Electronic Notice Board Using Wi-Fi," *International Journal of Innovative Science, Engineering & Technology (IJISET)*, vol. 3, no. 3, Mar. 2016.

[3] A. Hussain and M. A. Shaikh, "Chatbot-Based Smart Campus System for Student Support," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 3, pp. 145–158, 2020.

[4] Rasa Technologies Inc., "Rasa Open Source Documentation," 2024. [Online]. Available: https://rasa.com/docs/

[5] Django Software Foundation, "Django Documentation," 2024. [Online]. Available: https://docs.djangoproject.com/

[6] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Datasheet," 2024. [Online]. Available: https://www.raspberrypi.com/

[7] P. M. Repe, A. Hadoltikar, and P. Deshmukh, "Android Controlled Digital Notice Board," *International Journal of Advanced Foundation and Research in Computer*, vol. 3, no. 5, May 2016.

[8] R. K. Gurav and R. Jagtap, "Wireless Digital Notice Board Using GSM Technology," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 9, Dec. 2015.