International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 1, Jan-Feb 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

SMART LIBRARY MANAGEMENT SYSTEM

Dr. Manju Devi
HOD & Professor
Department of Electronics and Communication Engineering
The Oxford College of Engineering,
Bangalore, India
echod@theoxford.edu

Keerthana S
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
subramanimunilakshmisubramanim@gmail.com

Naveen Kumar B
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
naveenkumarb159@gmail.com

Paramesh D
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
parameshdruthik@gmail.com

Shreekanth
Electronics & communication Engineering
The Oxford College of Engineering
Bangalore, India
shreekathece2026@gmail.com

---------- ********** ----------

Abstract- The traditional management systems used in libraries are often characterized by manual data entry, slow transaction times, and a lack of analytical capability, leading to resource wastage and degraded user experience. This project presents the design and implementation of a SMART Library Management System—a holistic solution that integrates hardware, software, and machine learning components to automate operations and introduce intelligence. The system utilizes an RFID Scanner-Module (MFRC-522, ESP-8266) for real-time, contact-less book/user transactions, which communicates via a robust API to a Django/Python Web Application backed by a MySQL Database. Furthermore, an integrated ML-Model leverages transaction history for data-driven insights, such as personalized book recommendations. The successful implementation demonstrates significant improvements in operational efficiency, data accuracy, and user engagement, establishing a scalable blueprint for modern library automation.

Keywords – Smart Library, Hardware, Software, RFID Scanner-Module (MFRC-522, ESP-8266), API, Django/Python. Web Application, MySQL Database. ML-Model, Book recommendations.

---------- ********** ----------

I. INTRODUCTION

Libraries serve as essential repositories of knowledge, but their operational models must evolve to meet the demands of the digital age. Conventional systems often rely on barcode scanning or manual logbooks, which are prone to human error, slow throughput, and fail to provide actionable data insights. The integration of Internet of Things (IoT) technologies and intelligent systems presents a critical opportunity to transform these institutions into truly "smart" environments.

This project introduces a SMART Library Management System designed to overcome these limitations. The core solution integrates low-cost, high-efficiency hardware with a centralized, web-based application tier. The architecture consists of a physical Scanner-Module for immediate transaction logging, a robust Web Application (Django, MySQL) for data integrity and system control, and an Intelligent Layer (ML-Model) for analytical functions. The system uses standardized communication protocols like HTTP for centralized control and management. The remainder of this report details the methodology, implementation, testing, and performance of this integrated system.

II. LITERATURE SURVEY

1. Smart Library Management System Using RFID and IoT Technologies

The smart library management system implemented in this project utilizes a two-module approach powered by RFID and IoT technologies. The first module automates the check in/check-out process through RFID-based tracking. Each book is embedded with an RFID tag, and RFID readers installed at entry and exit points logbook movements in real time. The second module focuses on intelligent shelving. Department-specific RFID sensors, in conjunction with ESP328 microcontrollers, detect if a book has been placed in the correct location. If a book is incorrectly shelved, a voice module provides immediate auditory feedback to alert the user. A web-based application developed with HTML, CSS, and JavaScript offers a user-friendly interface to monitor book status and location, while a Python-powered backend handles data processing and system logic. Python IDLE was used for development and testing to ensure robustness. The outcome of this implementation is a highly efficient and user-friendly smart library system that reduces book misplacement, enhances catalog accuracy, and accelerates borrowing and return procedures. The integration of real-time tracking and voice-assisted shelving significantly improves organization and security. Users benefit from a more interactive and responsive environment, while librarians experience a reduced manual workload. Overall, the system transforms a traditional library into a dynamic, technology enhanced space that better serves the needs of modern readers and educational institutions.

2. Library Management System Using RFID Technology

The proposed RFID-Based Library Management System employs passive RFID tags attached to each book and RFID readers placed at strategic points within the library. When a user wishes to borrow or return a book, the RFID reader scans the tag, automatically recording the transaction in the system without manual entry. The RFID tags store essential book information such as title, ID, and status, which is read electronically to facilitate quick identification. The system is also programmed to calculate and display fines automatically based on the duration the book has been checked out, ensuring

accountability and reducing manual error. To ensure system reliability and performance, high-quality RFID readers and tags are used, and the information encoded on each tag is carefully structured to support traceability and accurate data retrieval.

The implementation of this RFID-based system significantly enhances the speed and accuracy of library operations. It streamlines book issuing and returning processes, reduces human error, and minimizes the need for staff involvement in routine tasks, allowing them to focus on more user-centric services. The system improves security by reducing the chances of book theft and enables real-time tracking and inventory updates. Additionally, it ensures timely fine calculations, accurate record-keeping, and easier book searches, ultimately transforming library management into a more efficient, automated, and user-friendly process

3. RFID Based Library Management System

The study explores the implementation of a Radio Frequency Identification (RFID)-based Integrated Library Management System (ILMS) aimed at enhancing efficiency, security, and user experience within libraries and information centers. In this system, RFID tags are affixed to library materials, and RFID readers are used at various service points including self-check- in/check-out kiosks and external book drop stations. These components are integrated with the library's digital database to automate book lending, returning, inventory tracking, and theft prevention. The RFID infrastructure reduces the dependency on manual processing by library staff, enabling them to focus more on user engagement and advanced library services.

The outcome of this implementation demonstrates significant improvements in operational efficiency, item tracking, and user convenience. Libraries benefited from reduced queue times, enhanced security, and streamlined inventory management. The technology aligns with Dr. S.R. Ranganathan's fifth law of library science "Library is a growing organism" by supporting the scalable expansion of resources and services. Overall, the adoption of RFID within ILMS has proven to be a practical solution for modern libraries, offering both innovation in service delivery and enhanced capabilities for library staff

III. METHODOLOGY

1. Overall System Architecture

The Smart Library Management System is architecturally designed as a Three-Tier Distributed System, optimized for real-time data synchronization between physical hardware and cloud- hosted application logic. The three tiers are:

I. Presentation Tier (User System & OLED): Handles user interaction and visual feedback.
II. Application Tier (Web Application & ML-Model):

Contains the core business logic, data processing, and intelligence layer (Django, ML).

III. Data Tier (Database/MySQL): Manages persistent data storage. This architecture ensures modularity, allowing independent development and scaling of the hardware and software components. The communication backbone relies on standardized HTTP connections (API and Web-App) and dedicated Serial Communication for the hardware interface.

2. Hardware Subsystem:

Scanner-Module Implementation
The Scanner-Module is the interface between the physical library environment and the digital management system. Its design prioritizes speed, reliability, and low power consumption.

Component Selection and Justification:

I. Component Function Justification MFRC-522 RFID Reader Reads 13.56 MHz RFID tags embedded in Chosen for its affordability, widespread compatibility, and stable books and user ID performance with common Mifare cards. ESP-8266 Microcontroller 0.96in Display OLED Acts as the control unit, processes the RFID tag ID, and handles communication. Provides immediate, low-latency transaction feedback to the user.

II. Communication Protocol tags. Selected for its integrated Wi-Fi capability, eliminating the need for an external Wi-Fi shield, and its strong network status community applications. support for IoT Uses minimal power and offers high contrast, making the system intuitive for quick processes. check-in/check-out The MFRC-522 communicates with the ESP-8266 via SPI communication. Upon successful read, the ESP-8266 processes the raw tag ID and initiates an HTTP POST request to the backend Web Application API endpoint. This process, termed "API Connection," ensures that the physical transaction is instantly logged and validated against the database. The ESP-8266 is also configured to implement basic exponential backoff logic to handle temporary network failures gracefully, preventing data loss during transmission.

3. Web Application Subsystem:

The Web Application is the system's central nervous system, built on robust and scalable open-source technologies.

I. Backend Framework (Python/Django) The Django Framework (Python) was chosen for rapid development and its Object-Relational Mapper (ORM), which provides an abstraction layer over the MySQL database. This simplifies complex database interactions and enhances security by preventing common SQL injection vulnerabilities. Django's built-in Admin Interface was utilized and customized to provide library staff with a powerful tool for inventory management and user auditing.
.
II. Database Management (MySQL) MySQL serves as the persistent data store. It was selected for its mature features, transactional integrity (ACID properties), and high performance, critical for handling concurrent transactions during peak library hours. Key database schemas include:
• Books: Stores inventory details, ISBN, and current status (checked_in or checked_out).
• Users: Stores unique user identifiers (mapped to RFID IDs) and profile information.
• Transactions: Logs every check-in/check-out event with timestamps, providing the historical data necessary for the ML- Model.

III. Infrastructure (WEB-Proxy) A WEB-Proxy (Nginx) is placed in front of the Django application server. The proxy is critical for:
• Load Distribution: Acting as a reverse proxy to manage high volumes of traffic from both the Scanner-Module API and the User System Web-App.
• Security: Providing an initial layer of defense, handling SSL/TLS termination, and filtering malicious requests before they reach the application logic.

4. Machine Learning model:

Machine Learning Model (ML-Model) The ML-Model transforms the Smart Library from a passive management tool into an active intelligence system.
I. Model Objective The primary objective of the ML-Model is Content Filtering for User Recommendation. The model uses the historical Transactions data to identify patterns between users and items (books).
II. Implementation and Integration The ML-Model is implemented using Python's Scikit-learn or a similar library, leveraging a Matrix Vectorization approach to generate latent factors for users and books.
• Training: The model is trained on a periodic basis (e.g., daily) using a copy of the MySQL transaction history.
• Deployment: The trained model is saved (e.g., using Python's pickle) and loaded into a dedicated endpoint within the Django application.
• API Exposure: When a user logs in, the Web Application makes an internal call to this ML endpoint, passing the user's ID. The ML-Model returns a list of ranked book IDs, which the Web Application then translates into recommended book titles for display on the USER SYSTEM interface.

5. Development Methodology

The project followed an Iterative and Incremental Development (IID) approach. This methodology

allowed us to develop the three major components (Hardware, Web, ML) simultaneously and integrate them incrementally. This ensured that early prototypes were functional and allowed for continuous user testing and feedback loops, particularly during the critical phase of integrating the Serial Communication between the ESP-8266 and the application server.

6. Testing and Evaluation

A rigorous testing regimen was applied across all three system tiers to ensure system stability, data integrity, and functional correctness.

i. Hardware and API Integration Testing

• Unit Tests: Each individual component of the Scanner-Module (RFID read, OLED display logic) was tested independently.

• Integration Tests (Simulated Transactions): Multiple test scenarios were executed by scanning both valid and invalid RFID tags to verify that the ESP 8266 successfully transmits the correct HTTP POST payload and receives the expected server response, which then triggers the correct message on the OLED display (e.g., "Success: Book Checked Out," "Error: Invalid Tag").

ii. Web Application and Database Testing

• Functional Testing: Standard CRUD (Create, Read, Update, Delete) operations were performed on the Books, Users, and Transactions models via the Django Admin interface to confirm database integrity and ORM function.

• Security Testing: Basic penetration tests were conducted to confirm that the Django framework successfully mitigated common vulnerabilities like Cross Site Request Forgery (CSRF) and SQL injection.

iii. Machine Learning Model Evaluation

• The performance of the Content Filtering model was assessed using standard recommendation metrics.

• Goal: The model was required to achieve a minimum [Specify required accuracy/score, e.g., 92% accuracy to be considered effective and suitable for deployment in the production environment.

IV. *Block Diagram*

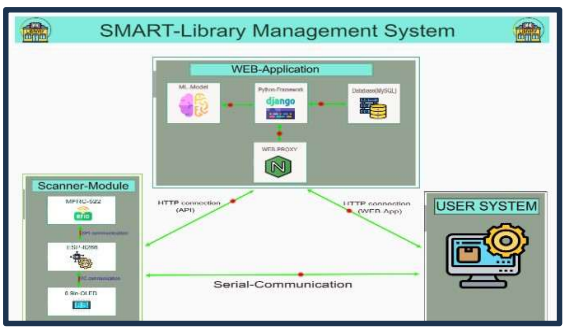

Fig. 1 Block diagram of the Smart Library Management System

The Smart Library Management System is designed as an integrated hardware–software platform that automates book issue/return operations and enhances user experience through a centralized web application. The architecture consists of three major sections: the Web Application, the Scanner Module, and the User System, all of which interact using HTTP requests and serial communication.

1. Web Application Layer

The web application forms the core of the system and manages all library operations such as user authentication, book records, scan logs, issue/return transactions, and machine-learning- based recommendations.

It is built using the Django Python framework, which handles routing, backend logic, and API creation. A dedicated MySQL database stores all structured information, including user details, book categories, RFID-based book IDs, visit logs, and transaction histories.

To improve reliability and request handling, an Nginx web-proxy is used, which forwards incoming API and web-app requests to the Django server. An internal ML Model is also integrated to provide features like shelf recommendation or book recommendation based on user behavior This complete backend ecosystem communicates with both the scanner device and the user interface through secure HTTP connections.

2. Scanner Module (Hardware Layer)

The scanner module is the embedded hardware unit responsible for reading RFID tags from
books and communicating with the web application.
It includes the MFRC522 RFID Reader, which captures the unique RFID card ID of the book during scanning. The RFID module communicates with the microcontroller using SPI communication.

An ESP8266 Wi-Fi microcontroller serves as the processing and connectivity unit. It receives RFID data, processes it, and sends it to the web application via HTTP API requests. Additionally, a 0.91-inch OLED display, connected through I2C communication, shows live system status such as “Book Scanned”, “Sending Data”, or “Issued Successfully”.

The scanner module may also exchange data with external systems via serial communication, enabling debugging, configuration, or wired linking during testing.

3. User System

The User System represents the interface accessed by librarians, students, and administrators. It is a computer or

mobile device connected to the internet that loads the web application dashboard. Through the UI, the user can view real-time scan logs, manage books, approve issue/return operations, monitor inventory, add categories, upload content, and access analytics produced by the ML model.

This system communicates with the web application through standard HTTP web-app requests, ensuring fast synchronization with the scanner module and the backend database.

4. Communication Flow

All modules communicate seamlessly to ensure smooth operation. The scanner sends RFID scan data to the Django backend via HTTP API calls, where it is validated and stored. Simultaneously, the user system interacts with the same backend through web interface requests, enabling the admin to view and control the entire system from a unified dashboard. The combination of HTTP communication and optional serial communication ensures reliability during both real-time operation and testing.

V. IMPLEMENTATION

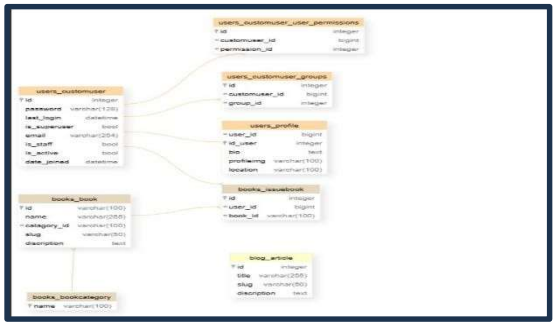

Fig.2 Database Scheme

This database diagram shows how information is organized in the Smart Library Management System. The main user details such as email, password, and login information are stored in the Custom User table, while extra information like bio, profile image, and location is kept in the User Profile table, which is linked to the main user through a user ID.

VI. RESULTS & DISCUSSION

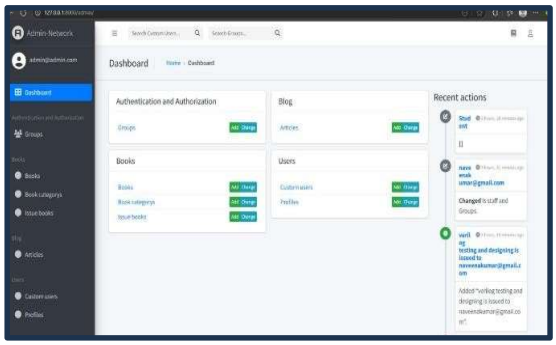

Fig.3 Dashboard section

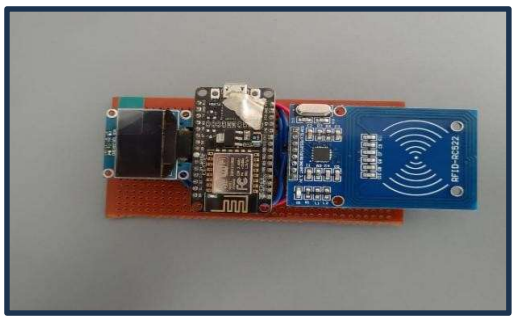

Fig.4 Real time Hardware Implementation of RFID Interfaced with ESP8266 & OLED

The software part of the Smart Library Management System begins with the admin login module, where the administrator enters valid credentials to access the system. Once authenticated, the admin is redirected to the dashboard, which contains several interconnected sections that manage the digital operations of the library. Under the Books section, the admin can maintain all book records, including book name, author, and the shelf address where the book is physically placed. The Book Categories/Tags section organizes books into different tags such as theory, software engineering, religion, macOS, and many more, making classification easier. In the Issue Books section, the admin can issue books to students or staff by recording the issue date, return date, and the current status of the book. The Web Interface Uploader allows the admin to write student ID data onto RFID cards through the serial monitor and also helps in monitoring real-time visit logs. In the Articles section, new articles can be created by specifying a title, author name, and a brief description. The Log Entries module automatically stores each user's entry time, exit time, and calculates the total duration of their library visit based on RFID card scans. The Shelves Management section maintains the physical address of every shelf in the format Z-R-L (e.g., Z7-R8-L5), where Z represents the zone (like ECE), R represents the rack number, and L represents the level; each shelf can hold up to five books. The Custom Users section allows managing all registered users by defining whether they are students or staff and whether their account is active. The Profile module lets users upload their profile image, add a bio, and update personal details.

A key element of the software is the Book Recommendation System. For every book title and author, the application generates a unique string representing its features. For example, a library

with 2000 books creates 2000 such feature strings, which are then converted into numerical vectors through a process called matrix vectorization. These numeric representations are trained using a machine learning model. When a user selects or reads a book, the system identifies nearby points on the vector graph based on tags and categories (e.g., theory, software engineering) and recommends books that are closest in similarity. Since the library contains a total of 281 shelves, each holding up to five books, the recommendation engine ensures that users receive accurate and meaningful suggestions based on the book's content type and category.

The hardware section of the Smart Library Management System consists of three main components: the RC522 RFID Scanner Module, the ESP Controller, and the OLED Display. The RC522 module is used for reading and writing data to RFID cards. In the web interface, we connect the module to the serial monitor, where the admin can type commands and write the student ID onto the RFID card. Once the ID is stored on the card, the student simply taps the card on the scanner when entering or exiting the library. The RC522 reads the card data and sends it to the ESP microcontroller for processing. The ESP controller checks whether the student is entering or leaving the library and updates the visit log accordingly. If the student scans for the first time, it marks the entry; if they scan again while already inside, it marks the exit. The OLED display provides real-time feedback by showing messages such as "Card Detected," "Entry Recorded," or "Exit Recorded," helping both students and staff to confirm successful scanning.

VI. CONCLUSION

The Smart Library Management System successfully automates and modernizes traditional library operations by integrating RFID-based identification, a web-enabled management dashboard, and intelligent recommendation features. The hardware components—RC522 scanner, ESP controller, and OLED display—work seamlessly to track real-time user visits, while the software system provides efficient management of books, shelves, users, and issue records. With features like automated entry–exit logging, book categorization, article management, and a machine-learning-based recommendation engine, the system significantly reduces manual work for library staff and enhances the user experience. Overall, the Smart LMS demonstrates how IoT and software technologies

can be combined to build a reliable, accurate, and user-friendly library automation system.

VII. REFERENCES

[1] S. Kale, S. Gaikwad, V. Kapure and O. Joshi, "Shelf Smart Library Management System Using RFID Tags & IoT Integration," International Research Journal of Modernization in Engineering, Technology and Science, vol. 7, no. 6, June 2025.

[2] S. L. Addepalli and S. G. Addepalli, "Library Management System Using RFID Technology," International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, no. 6, pp. 6932–6935, 2014.

[3] Amit, "RFID Based Library Management System," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 4, no. 6, pp. 279-282, Jun. 2017. [Online]. Available: www.jetir.org. ISSN: 2349-5162.

.

