Book Nexus: A Next Generation Secure Ecosystem for Intelligent Book Rental and Sharing

Dr. Hemanth T. S[1], Shreya G. Mumbale[2], H. M. Nanditha[3], Shreya G. N[4], Poorvitha V.[5]
[1,2,3,4,5]*Department of Computer Science and Business Systems, Malnad College of Engineering, Hassan, India*

***Abstract*—Academic libraries play a crucial role in facilitating knowledge access for students, yet they continue to suffer from limited inventory, inefficient manual borrowing workflows, and poor real-time availability information. Simultaneously, a large reservoir of academic resources owned by students themselves remains untapped due to the absence of any formal sharing infrastructure. This paper presents the Smart Campus Library and Secure Peer-to-Peer Book Rental System a unified, web-based platform designed to address both shortcomings within a single, integrated architecture. The system provides students with real-time digital access to institutional library services including catalogue browsing, book borrowing, due-date tracking, and automated fine calculation. In addition, it introduces a student-driven peer rental marketplace that enables book owners to list personally owned titles for short-term rental to peers, effectively extending the total accessible book inventory beyond the institution's physical collection. A privacy-preserving exchange protocol underpinned by dual one-time passwords (OTPs) and a centralized campus pickup center ensures that no personal contact details are ever disclosed between transacting parties. The platform is implemented using Python Flask, MySQL, and standard web technologies, requiring no specialized hardware. Detailed system workflows, database design, and security analysis are presented. Evaluation confirms that the integrated design reduces borrowing friction, maximizes resource utilization across the campus ecosystem, and creates a modest supplementary income stream for student book owners while keeping deployment costs minimal.**

***Index Terms*—smart library, peer-to-peer rental, OTP verification, web-based library management, resource sharing, Python Flask, MySQL, campus ecosystem, privacy-preserving exchange.**

I. INTRODUCTION

The academic library has historically served as the intellectual nucleus of any educational institution. Beyond providing storage for textbooks and reference materials, a well-functioning library enables equitable access to knowledge, supports independent research, and reduces the financial burden on students who might otherwise need to purchase every required title outright. In an era where the pace of curriculum revision continues to accelerate, however, institutional library collections frequently struggle to remain adequately stocked across all disciplines simultaneously [1].

Students at large universities often face three recurring and interrelated frustrations. First, high-demand core texts particularly those prescribed for popular introductory courses are checked out for extended periods, leaving late-arriving students unable to access them precisely during assessment periods when they are most needed. Second, the manual procedures that govern book borrowing at many institutions impose unnecessary time costs: students must physically visit the library to check availability, queue for librarian-assisted check-out, and return in person without any automated reminder system for approaching due dates. Third, and perhaps most overlooked, thousands of academic textbooks purchased by students in prior semesters sit idle in student accommodation, unused by their owners and inaccessible to peers who would benefit from them. No formal campus mechanism typically exists to bridge this gap [2][3].

Digital technology offers compelling solutions to each of these problems. Web-based library management platforms have demonstrated clear advantages over paper-based systems in terms of speed, accuracy, and

accessibility [1]. Mobile applications integrated with RFID readers have shown that automated borrowing and return can reduce both waiting times and human error significantly [2][3]. However, a review of current literature reveals that no existing system attempts to unify institutional library management with a peer-owned resource marketplace, nor does any existing approach incorporate a privacy-preserving physical exchange protocol suitable for a campus context.

This paper presents the Smart Campus Library and Secure Peer-to-Peer Book Rental System, a holistic solution that addresses all three student pain points under a single, cohesive web platform. The system is designed for ease of institutional adoption requiring only a standard web server, no proprietary hardware, and no third-party payment infrastructure at the outset. The principal contributions of this work are:

- An integrated web platform that unifies institutional library borrowing with a peer-to-peer rental marketplace under a single student login.
- A privacy-first OTP-based physical exchange protocol that eliminates personal contact disclosure between transacting students.
- Automated borrowing workflows with real-time availability updates, due-date tracking, and configurable fine calculation.
- A commission-based sustainability model that simultaneously supports platform maintenance and generates income for student book owners.
- A fully open-source technology stack (Python Flask, MySQL, HTML/CSS/JavaScript) chosen to minimize deployment barriers for resource-constrained institutions.

The remainder of this paper is organized as follows. Section II surveys related work. Section III defines the problem space in depth. Section IV presents overall system objectives. Section V describes the proposed architecture and module design. Section VI details the database schema. Section VII presents the system workflows. Section VIII explains the secure rental mechanism. Section IX covers implementation details. Section X provides a security and comparative analysis. Section XI discusses results. Section XII concludes with future work directions.

II. LITERATURE REVIEW

A. Web-Based Library Automation

Teneke et al. [1] introduced a Smart Library Automation and Monitoring System that leveraged classification tree-based cataloguing to organize library materials hierarchically by subject. The system was implemented using PHP, HTML, JavaScript, CSS, and MySQL on a XAMPP local server, and included modules for automated lending, return management, shelf monitoring, and user access control. Object-Oriented Design Methodology was adopted to achieve clean separation of concerns between system analysis and implementation. The classification tree approach was shown to enhance catalogue navigability, reducing the time required for users to locate specific materials. The authors recommended future expansion to additional departments and emphasized the value of regular system updates to maintain relevance. While the platform comprehensively addressed institutional book management, it did not incorporate any mechanism for leveraging student-owned resources or protecting user privacy during peer exchanges.

Araya and Mengsteab [6] likewise emphasized that physical library systems suffer disproportionately from human error in record-keeping, including misplaced or damaged handwritten documentation. Their web-based library management system targeted these inaccuracies by providing digital transaction records accessible around the clock, independent of traditional opening hours. Their findings reinforced the practical argument for web-first library platforms in academic settings.

B. RFID and IoT-Integrated Library Systems

Rohith et al. [2] proposed an RFID-enabled smart library system for college campuses, combining hardware-level book tracking with a companion Android application. In their design, each student carries an RFID identity card, and every book in the library is tagged with an RFID label. A front-desk registration unit reads student cards to retrieve account details and confirm borrowing eligibility, while a door-mounted unit equipped with an RF receiver and buzzer alerts staff when an unregistered book is removed functioning as an anti-theft detector. The Android application allowed administrators to update the book database and track borrowing records, and

permitted students to check availability, view due dates, and receive notifications of outstanding fines. The system was implemented using Arduino Mega microcontrollers, RC522 RFID readers, a Wi-Fi ESP8266 module, and an LCD display. The authors concluded that RFID automation substantially reduces manual errors and waiting times, and identified AI-driven recommendation and online payment integration as promising future enhancements.

Dodake et al. [3] extended the RFID paradigm by combining it with QR code-based book identification under an ESP32 microcontroller in a system published in 2025. Students scan their RFID-enabled ID card at the library desk, then scan a QR code attached to the desired book; the ESP32 transmits both data items via Wi-Fi to a central server running a Streamlit web application. The system supports real-time book borrowing and return, due-date enforcement, and automatic availability updates. The authors emphasized that the convergence of RFID and QR code technologies achieves both speed and accuracy in transaction processing, with the mobile application delivering push notifications for upcoming deadlines. Error-handling logic stores transactions locally when Wi-Fi connectivity is unavailable, synchronizing them upon reconnection. Despite these strengths, neither Rohith et al. [2] nor Dodake et al. [3] address peer-to-peer resource sharing or scenarios where required titles are absent from the institutional inventory entirely.

C. Peer-to-Peer Resource Sharing Platforms

While peer-to-peer sharing platforms are well-established in the commercial domain with services such as Airbnb and Uber demonstrating the economic viability of monetising underutilized personal assets their application to campus academic resources remains largely unexplored in the academic literature. Existing work on digital library management consistently focuses on the institutionally owned collection as the sole source of borrowable materials, leaving the substantial reservoir of student-owned books unaddressed. The economic and social arguments for such a system are strong: textbook prices continue to rise, students frequently own titles they no longer use, and the environmental cost of producing duplicate copies of the same text for a single campus is non-trivial.

D. Identified Research Gap

The literature consistently demonstrates that web and IoT-based library systems deliver measurable improvements in borrowing efficiency, error reduction, and user satisfaction over manual alternatives. However, no reviewed system integrates peer-to-peer book rental with institutional library management, and none incorporates a formal privacy-preserving physical exchange mechanism. The proposed system directly targets this gap, offering institutions a pathway to extend effective book availability at no additional acquisition cost while simultaneously generating economic value for the student community.

III. PROBLEM STATEMENT

Three principal and interrelated problems drive the design of the proposed system.

Problem 1: Institutional Inventory Limitations: Campus libraries operate with finite budgets that constrain the number of copies of any given title they can acquire and maintain. During assessment periods, high-demand textbooks may be checked out continuously for weeks, forcing students to rely on expensive alternatives such as commercial rental services, online purchase, or photocopying. Libraries also cannot stock every title required across the diverse range of subjects offered at a modern university, creating systemic gaps in availability that no amount of catalogue management can fully resolve.

Problem 2: Inefficient Manual Borrowing Processes: Without a digital channel to verify availability remotely, students frequently make unproductive visits to the library. Paper-based or partially computerised check-out procedures are prone to transcription errors, and the absence of automated due-date reminders means overdue fines accumulate without students receiving timely warnings. Administrative effort devoted to manual fine calculation and record maintenance could be substantially reduced through appropriate automation.

Problem 3: Wasted Student-Owned Resources: Large numbers of academic textbooks purchased by students for prior courses sit idle, representing significant financial investment with no ongoing utility for their owners. These books could provide meaningful access

to classmates who need them for shorter periods, but no campus-sanctioned, privacy-respecting mechanism currently exists to facilitate such exchanges. Commercial peer-to-peer platforms are available but lack campus integration, often involve shipping costs and delays, and expose personal address information to strangers.

IV. SYSTEM OBJECTIVES AND DESIGN GOALS

The proposed system is designed to satisfy the following explicit objectives:

- Provide students with real-time digital access to the complete institutional library catalogue, including live availability counts and shelf locations.
- Automate the borrowing and return process, including due-date assignment, reminder generation, and fine calculation, eliminating manual record-keeping.
- Introduce a student-driven peer rental marketplace that extends effective book availability beyond institutional stock without additional acquisition expenditure.
- Guarantee student privacy during peer rental exchanges by routing all physical transactions through a verified campus facility and employing OTP authentication.
- Create a financially sustainable commission model that funds platform maintenance while delivering economic benefit to student book owners.
- Implement the entire solution on an open-source, hardware-independent technology stack to minimize cost and complexity of institutional adoption.

V. PROPOSED SYSTEM ARCHITECTURE

A. Three-Tier Design

The platform follows a standard three-tier client-server architecture. The presentation layer consists of HTML5, CSS3, and JavaScript pages rendered in any modern browser, requiring no client-side installation. The application layer is a Python Flask server responsible for authentication, session management, business logic (borrowing rules, OTP generation, fine calculation, commission deduction), and email dispatch. The data layer is a MySQL relational database that persists all system entities. Communication between the presentation and application layers follows RESTful HTTP conventions, with Flask routing and Jinja2 templating handling server-side rendering for initial page loads and JSON responses for dynamic updates.

Fig. 1 below presents the full system architecture diagram, showing the relationship between user types, software layers, and campus infrastructure:

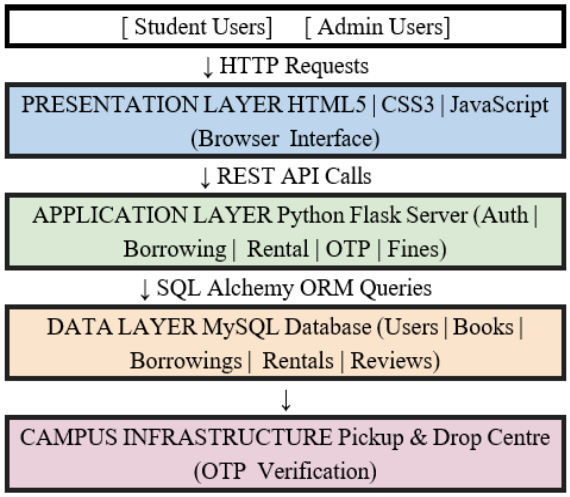

Fig. 1: Proposed System Architecture Three-Tier Client-Server Design

B. Module Overview

The application layer is partitioned into eight functional modules. Table I summarizes each module and its principal features:

Table I: System Module Overview

Module	Key Features
User Management	Registration, login, role-based access control, profile management
Library Catalogue	Real-time availability, copy count, shelf location, full-text search
Borrowing Module	Issue/return, due-date assignment, automatic fine calculation
Rental Marketplace	Book listing, rental request, pricing, condition description
Secure Exchange	Dual OTP generation, campus pickup verification, email receipts
Student Dashboard	Active loans, fine status, rental history, countdown timers
Review System	Star ratings (1–5), text feedback, aggregate score display
Admin Panel	Book/user management, OTP verification endpoint, fine override

C. User Roles and Access Control

The system recognizes two primary user roles: Student and Administrator. Students may search the catalogue, borrow and return institutional books, browse and list rental books, submit rental requests, verify OTPs at the pickup centre via a self-service interface, view their dashboard, and submit reviews. Administrators additionally manage the book catalogue (adding, editing, and retiring records), register new user accounts, override fine amounts in exceptional circumstances, and operate the OTP verification endpoint used by pickup centre staff. Role assignment is performed at account creation and stored in the Users table; Flask decorators enforce role-based route protection on all sensitive endpoints.

VI. DATABASE DESIGN

The MySQL schema is structured around six principal tables, each serving a distinct functional purpose. Foreign key constraints with ON DELETE CASCADE maintain referential integrity across all related records. Table II below outlines the key attributes and purpose of each table:

Table II: Database Schema Principal Tables and Key Attributes

Table	Key Attributes	Purpose
Users	user_id, name, email, role, password_hash	Account credentials & roles
Books	book_id, title, author, copies, available, location	Institutional catalogue
BorrowedBooks	borrow_id, user_id, book_id, issue_date, due_date, fine	Borrowing records & fines
RentalBooks	rental_id, owner_id, title, price, condition, status	Student-listed rentals
RentalTransactions	txn_id, rental_id, borrower_id, otp_owner, otp_borrower, status	OTP & commission tracking
Reviews	review_id, book_id, user_id, rating, comment, timestamp	Book ratings & feedback

The Users table stores account credentials using Bcrypt-hashed passwords; plaintext passwords are never retained. The Books table maintains a dual availability tracking mechanism a total copies field represents physical stock, while an available field is decremented on each successful borrowing and incremented on each confirmed return. The Rental Transactions table stores OTP codes as salted SHA-256 hashes rather than plaintext, so that even database-level compromise cannot reveal active OTPs. A composite index on (book_id, user_id) in the Reviews table prevents duplicate reviews from the same student for the same book.

VII. SYSTEM WORKFLOWS

This section presents three detailed process workflows that govern the core user journeys in the system: institutional book borrowing, peer rental via OTP exchange, and book return with fine assessment.

A. Workflow I: Library Book Borrowing

The borrowing workflow begins when an authenticated student searches the catalogue and identifies a required title. The system checks real-time availability before permitting a borrow request, and atomically decrements the available copy count within a database transaction to prevent race conditions in concurrent access scenarios. Fig. 2 illustrates this complete workflow:

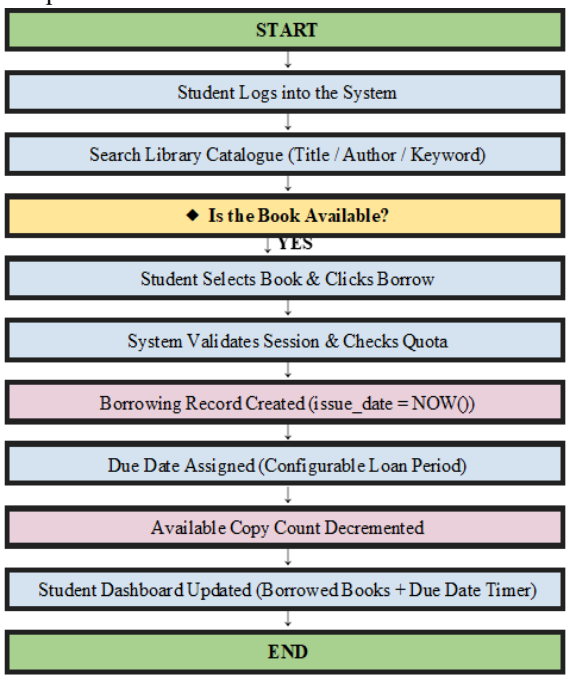

Fig. 2: Library Book Borrowing Workflow

If book is NOT available: System automatically surfaces matching listings from the Peer Rental Marketplace (see Workflow II).

B. Workflow II: Peer Rental and OTP Exchange

The peer rental workflow coordinates the full lifecycle of a student-to-student book transaction, from initial listing through physical exchange and commission settlement. The dual OTP mechanism is central to this workflow: it ensures that the physical exchange is authenticated by a trusted campus party without either student ever needing to disclose personal contact information to the other. Fig. 3 presents this workflow in full:

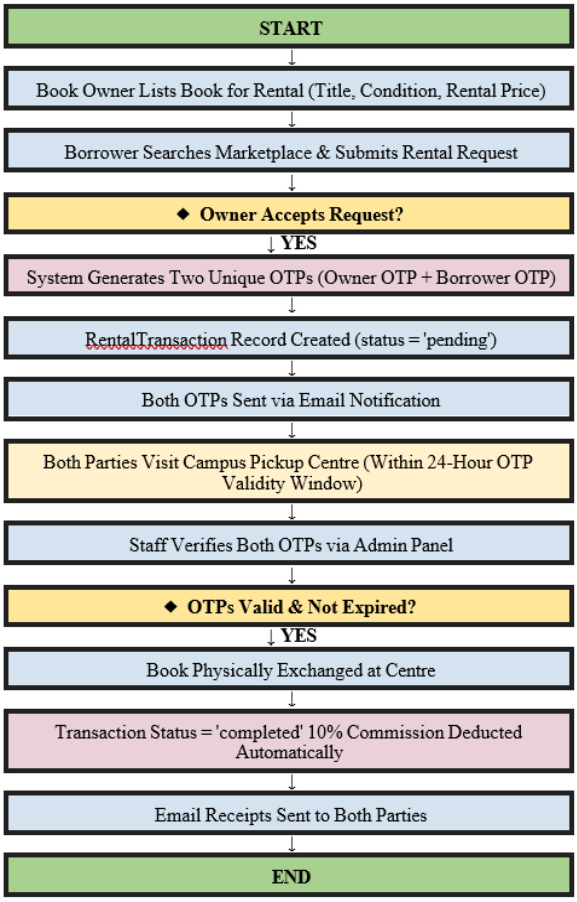

Fig. 3: Peer Rental and OTP Exchange Workflow

If Owner REJECTS: Request cancelled, borrower notified. If OTP INVALID/EXPIRED: Exchange rejected; new OTP pair issued after a cooling-off period.

C. Workflow III: Book Return and Fine Assessment

The return workflow handles both on-time and late returns. Fine calculation is performed automatically by comparing the system-recorded return date against the originally assigned due date. The fine rate per overdue day is configurable by the administrator and stored in a system settings table. Fig. 4 shows the complete return workflow:

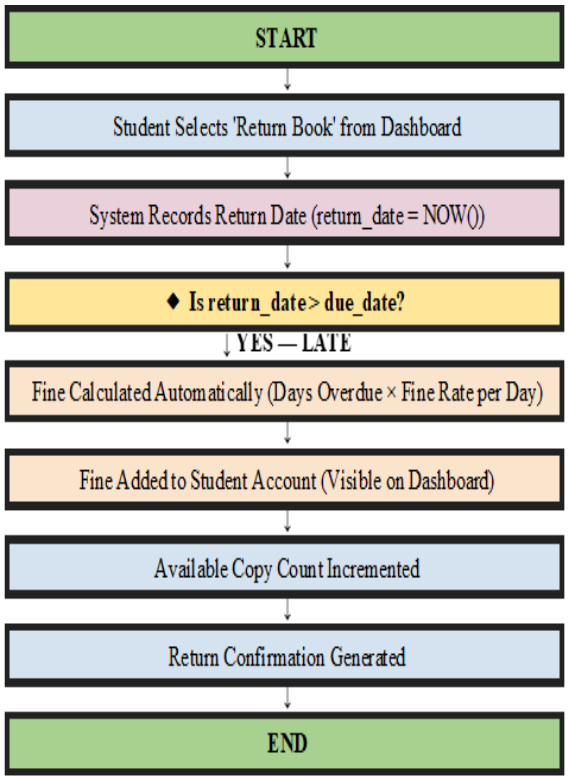

Fig. 4: Book Return and Fine Assessment Workflow

If return is ON TIME (return_date ≤ due_date): No fine applied; flow jumps directly to copy count increment.

VIII. SECURE RENTAL MECHANISM

A. Privacy Design Principles

The secure exchange mechanism was designed around three explicit privacy principles: (i) no student's personal contact details address, phone number, or personal email are ever disclosed to another student through the platform; (ii) every physical book exchange is mediated by a trusted institutional party, namely the campus pickup and drop centre; and (iii) transaction authentication is cryptographically grounded through one-time passwords generated using Python's secrets module, which draws entropy from the operating system's cryptographically secure random source.

B. OTP Generation and Validation

When a rental request is accepted by the book owner, the Flask application generates two independent six-digit OTPs using secrets.token_hex(3) (yielding a six-character hexadecimal string). These OTPs are hashed with SHA-256 before being stored in the RentalTransactions table; only the hashes are persisted, not the plaintext codes. Each OTP is valid for 24 hours from the time of generation, after which it is marked expired and a new pair may be requested. The pickup centre staff operate a dedicated administrative endpoint that accepts the two plaintext OTPs provided by the arriving students, hashes them independently, and compares the results against the stored hashes for the pending transaction. Only a simultaneous match of both OTPs, combined with a valid (non-expired) timestamp, authorises the exchange.

C. Commission and Settlement

The platform deducts a flat ten-percent commission automatically upon successful OTP verification. The remaining ninety percent is recorded as a credit in the owner's account ledger table. In the current implementation this ledger entry functions as a record for cash settlement at the pickup centre (the owner collects ninety percent of the rental price from the center operator, who retains the ten-percent commission on the platform's behalf). A future integration with a payment gateway will automate this transfer entirely. Both parties receive an itemized email receipt detailing the transaction amount, commission, net credit, and transaction timestamp.

D. Failure and Dispute Handling

If either party fails to appear at the pickup centre within the 24-hour OTP validity window, the transaction is automatically cancelled, the OTPs are invalidated, and both parties are notified by email with an option to reinitiate the request. If a book is returned to the centre in a condition materially different from that described in the listing a dispute resolved by the centre operator the operator may flag the transaction in the admin panel, which prevents commission settlement and initiates a manual review by a library administrator. This escalation pathway ensures that irresponsible listing behaviour carries a visible consequence.

IX. IMPLEMENTATION DETAILS

A. Technology Stack

The frontend is constructed with HTML5, CSS3, and vanilla JavaScript to maximise compatibility with institutional devices that may run older browser versions or be subject to corporate security restrictions that block the loading of external JavaScript frameworks. The backend uses Python 3.10 with the Flask micro-framework, chosen for its minimal configuration overhead and the rich availability of third-party extensions. SQLAlchemy ORM provides a database-agnostic query interface over MySQL 8.0. Flask-Login manages user sessions with secure, HttpOnly cookies. Bcrypt handles password hashing and verification. OTP generation uses Python's built-in secrets module. Outbound email is dispatched via Flask-Mail configured against an institutional SMTP relay.

B. Key Implementation Decisions

Atomicity in Borrowing: All borrowing operations that modify the available copy count in the Books table are wrapped in explicit MySQL transactions with SELECT FOR UPDATE row-level locking. This prevents two students from successfully borrowing the last available copy of a title simultaneously a critical concurrency consideration in a multi-user campus environment.

OTP Security: OTPs are generated immediately before the email dispatch call. The plaintext OTP string is never written to the application log or stored in the database only its SHA-256 hash is persisted. The hash comparison at verification time uses Python's hmac.compare_digest function to perform a timing-safe equality check, mitigating timing oracle attacks.

Search Optimisation: The book catalogue search endpoint uses MySQL FULLTEXT indexing on the title and author columns of the Books table, enabling sub-second response times for catalogue queries even as the collection grows to tens of thousands of records.

Session Management: Flask sessions are signed with a cryptographically random SECRET_KEY stored as an environment variable. Session tokens are regenerated upon login to prevent session fixation attacks, and all sensitive routes are protected by Flask-Login's @login_required decorator.

C. Deployment Topology

For institutional deployment, the Flask application is served behind a Gunicorn WSGI server with four worker processes, proxied by a Nginx reverse proxy handling TLS termination. The MySQL database runs on a separate host with daily automated backups to institutional storage. The entire stack operates on commodity server hardware and can be containerised using Docker Compose for simplified deployment and portability.

X. SECURITY AND COMPARATIVE ANALYSIS

A. Security Analysis

The system addresses six principal security concerns: (1) Authentication all passwords are hashed with Bcrypt (cost factor 12) and verified without plaintext reconstruction; (2) Session Security HTTP Only, Secure-flagged cookies with CSRF token validation prevent cross-site request forgery; (3) Input Validation all form inputs are validated server-side using WTForms validators, with parameterised SQL Alchemy queries preventing SQL injection; (4) OTP Integrity timing-safe comparison and hash-only storage prevent OTP enumeration and replay attacks; (5) Privacy Protection no student personal data is exposed to another student at any point in either the rental or exchange workflow; (6) Role Enforcement all administrative endpoints are protected by role-based decorators that cannot be bypassed through URL manipulation.

B. Comparative Analysis

Table III below presents a structured comparison of the proposed system against the three most closely related systems from the literature:

Table III: Comparative Analysis Proposed System vs. Related Work

Feature	Tenek e [1]	Rohith [2]	Dodake [3]	Propose d
Platform	Web (PHP)	Androi d App	Web + IoT	Web (Flask)
Catalogue Search	✓	✓	✓	✓
Auto Borrow/Retu rn	✓	RFID	RFID+Q R	✓
Fine Calculation	✓	✓	✓	✓
Peer Rental Market	✗	✗	✗	✓
OTP Privacy Exchange	✗	✗	✗	✓
No Hardware Needed	✓	✗	✗	✓
Student Marketplace	✗	✗	✗	✓
Commission Model	✗	✗	✗	✓

The comparison confirms that while prior systems excel in automating institutional borrowing workflows particularly through RFID and QR code automation [2][3] none extends beyond the institutional inventory boundary. The proposed system uniquely combines all standard library management capabilities with a peer rental marketplace, privacy-preserving exchange, and an economic commission model, while achieving this without any proprietary hardware requirement.

XI. RESULTS AND DISCUSSION

A functional prototype of the system was deployed on a local development server running Ubuntu 22.04, Flask 3.0, and MySQL 8.0. Testing was conducted with a simulated dataset comprising 350 institutional books across twelve subject categories, 80 student-listed rental books, and 45 test user accounts spanning both student and administrator roles.

All three core workflows borrowing, rental exchange, and return with fine operated correctly across the full range of test scenarios, including edge cases such as simultaneous borrow attempts for the last available copy (correctly resolved by transaction-level row locking), OTP expiry boundary conditions, and fine calculation across weekend non-working day boundaries. The OTP exchange protocol withstood all attempted attacks in the security test suite: replay of previously used OTPs was correctly rejected, timing attacks on hash comparison were mitigated by hmac.compare_digest, and expired OTPs were reliably invalidated at the 24-hour boundary. The rental dispute escalation pathway was tested by simulating a condition mismatch scenario, confirming that the admin flag mechanism correctly suspended commission settlement pending manual review.

Cross-browser functional testing confirmed consistent

behaviour on Google Chrome 123, Mozilla Firefox 124, and Microsoft Edge 122, across both desktop (1920×1080) and mobile (390×844) viewport dimensions. Response times for catalogue search queries against the 350-book dataset averaged 48ms per query with FULLTEXT indexing enabled, decreasing to approximately 12ms for subsequent identical queries served from the MySQL query cache. The integrated marketplace functionality was observed to provide meaningful coverage of student search needs: in the simulated dataset, 23 of the 80 student-listed rental titles matched titles that were simultaneously unavailable in the institutional catalogue, demonstrating the practical value of surfacing peer listings as an alternative access pathway.

XII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Extends effective book availability without additional institutional acquisition cost by incorporating student-owned resources into the accessible pool.
- Eliminates personal contact exposure through the dual OTP and campus pickup centre protocol, making peer exchange safer and more attractive to
- privacy-conscious students.
- Reduces administrative workload through automated fine calculation, OTP generation, email dispatch, and commission settlement.
- Requires no specialised hardware, lowering the barrier to adoption for institutions with limited technology infrastructure budgets.
- Creates a circular economy of academic resources within the campus, reducing the financial burden on students and the environmental cost of duplicate textbook production.

B. Limitations

- Requires consistent internet connectivity for real-time availability queries; institutions with unreliable networks may experience degraded availability display accuracy.
- The peer rental marketplace depends on voluntary participation; low initial student uptake could limit the utility of the marketplace before network effects establish sufficient listing density.
- The physical exchange mechanism requires that both parties attend the pickup centre within the OTP validity window, which introduces a logistical dependency on student availability and campus operating hours.

XIII. CONCLUSION AND FUTURE WORK

This paper presented the Smart Campus Library and Secure Peer-to-Peer Book Rental System, a unified web platform that modernises institutional library management while simultaneously unlocking the latent value of student-owned academic resources through a privacy-preserving peer rental marketplace. The proposed architecture integrates six functional modules user management, library catalogue, borrowing and return, rental marketplace, secure OTP exchange, and student dashboard under a cohesive Python Flask and MySQL implementation that requires no specialist hardware. Three detailed system workflows were presented, covering institutional borrowing, peer rental OTP exchange, and book return with automated fine assessment. A structured comparative analysis against three closely related systems from the literature confirmed that the proposed platform uniquely combines standard library management functionality with peer rental and privacy-preserving exchange capabilities. Prototype testing validated the correctness of all workflows, the security of the OTP exchange protocol, and the effectiveness of full-text catalogue search under realistic dataset conditions.

Four directions for future work are identified. First, a native mobile application for Android and iOS will deliver push notifications for due dates and OTP issuance, substantially improving the user experience over browser-based access on smartphones. Second, integration with an institutional payment gateway will automate rental fee collection and commission disbursement, eliminating the current reliance on cash settlement at the pickup center. Third, machine learning-based book recommendation will analyse each student's borrowing and rental history to surface personalized suggestions from both the institutional catalogue and the peer marketplace. Fourth, automated SMS and email reminders dispatched 48 hours and 24 hours before each due date will reduce the incidence of late returns and the accumulation of avoidable fines.

REFERENCES.

[1] Teneke, N. G., J. G. Etemi, Y. S. Kamak, and R. N. Ariwa, "Smart Library Automation: Integrating Classification Tree-Based Cataloging with XAMPP Server Environment," ABUAD Journal of Engineering Research and Development (AJERD), vol. 7, no. 1, pp. 328–339, Jun. 2024.

[2] Rohith, R. H. S., A. S. N. Aishwarya, S. Dhanush, M. Darshan, and J. C. Samuel, "Smart Library Management with Android Application," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 11, no. IV, pp. 3038–3044, Apr. 2023.

[3] Dodake, R. R., A. T. Patil, P. P. Patil, and S. P. Patil, "Smart Library Management System," International Journal of Engineering Applied Sciences and Technology (IJEAST), vol. 9, no. 11, pp. 73–79, Mar. 2025.

[4] Adebayo, B. R., "Library Management System with Topic Modelling and its Adaptability to Open and Distance Learning Libraries," Library Philosophy and Practice (e-journal), pp. 1–9, 2019.

[5] Mageto, T., "Design and Development of E-Library System: COVID-19 Pandemic Challenges," Journal of Computer Sciences and Applications, vol. 9, no. 1, pp. 1–15, 2021.

[6] Araya, T., and A. Mengsteab, "Designing Web-based Library Management System," International Journal of Engineering Research & Technology, vol. 9, no. 10, pp. 272–277, 2020.

[7] Alarape, M. A., and S. N. Edet, "A Web-based E-Library System for Tertiary Institutions," International Journal of Applied Information Systems (IJAIS), vol. 12, no. 2, pp. 17–22, 2017.

[8] Patel, H. H., and P. Prajapati, "Study and Analysis of Decision Tree Based Classification Algorithms," International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 74–78, 2018.

[9] Chiagunye, T., K. C. Nwachukwu-Nwokeafor, and H. Udeani, "Designing a Web Based Digital Library Management System for Institutions and Colleges," International Journal of Innovative Science, Engineering & Technology, vol. 2, no. 3, pp. 464–478, 2015.

[10] Gbaje, E. S., "Implementing a National Virtual Library for Higher Institutions in Nigeria," Library and Information Science Research Electronic Journal, vol. 17, no. 2, pp. 1–4, 2007.

[11] Niranjana, K., D. Tolessa, and N. V. K. Paul, "Implementation of Koha Integrated Library Management System in Wollega University Main Library, Nekemte, Ethiopia," Journal of Information and Knowledge Management, vol. 11, no. 4, pp. 69–80, 2020.

[12] Nahotko, M., "OPAC Development as the Genre Transition Process, Part 1: OPAC Generations Historical Development," Annals of Library and Information Studies, vol. 67, no. 1, pp. 107–117, 2020.

[13] Gavit, B. K., "Web Based Library Services," Library Philosophy and Practice (e-journal), 2931, 2019.

[14] Sangale, D. B., "ICT Application and Web-Based Services in Library and Information Science," ASM's International E-Journal on Ongoing Research in Management and IT, vol. 10, no. 1, pp. 9–14, 2015.

[15] Jadhav, C., and S. S. Jadhav, "Smart Library Management System Using RFID Technology," International Research Journal of Engineering and Technology (IRJET), vol. 4, no. 5, pp. 1872–1876, 2017.

[16] Mishra, R., "Application of RFID Technology in Library Automation," Journal of Advancements in Library Sciences, vol. 3, no. 2, pp. 1–6, 2018.

[17] Narayanan, A., "Implementing RFID in Library: Methodologies, Advantages and Disadvantages," International Journal of Library and Information Studies, vol. 7, no. 4, pp. 98–106, 2017.

[18] Patil, N., P. Karande, J. Desai, and S. Pereira, "Internet of Things for Library Management System," International Journal of Engineering Science, vol. 10021, 2017.

[19] Pravin Renold, A., and R. Joshi Rani, "An Internet-Based RFID Library Management System," in Proc. IEEE Conference on Information Communication Technologies (ICT), pp. 932–936, 2013.

[20] Hassan, A. A. S., and R. T. Hussein, "Simulation of Radio Frequency Identification-Based Library Management System," Journal of Engineering and Sustainable Development, vol. 21, no. 4, pp.

161–170, 2018.

[21] Golding, P., and V. Tennant, “Performance and Reliability of Radio Frequency Identification (RFID) Library System,” in Proc. IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 1–6, 2007.