

BookTrack

**A PROJECT REPORT
for
Mini Project-II (ID201B) Session
(2024-25)**

Submitted by

Kartik Agarwal(202410116100096)

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATIONS

**Under the Supervision of
Ms. Shruti Aggarwal
Assistant Professor
Associate Professor
Mr. Rabi. N Panda**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh - 201206 (MAY-2025)**

DECLARATION

We, the undersigned, hereby declare that the Mini Project titled "**BookTrack**" is an original work completed by us as part of the curriculum requirement for the course under the Master of Computer Applications (MCA) program at **KIET Group of Institutions**.

We affirm that we have undertaken this during the academic year 2024-25 under the guidance of **Ms. Shruti Aggarwal**. All the content and ideas presented in this report are the result of our own efforts, except where explicitly stated otherwise. Proper citations have been provided wherever references to external sources have been made.

We further declare that this project has not been submitted, either in part or in full, to any other university or institution for any degree or diploma.

Kartik Agarwal

CERTIFICATE

Certified that Kartik Agarwal (202410116100096) have carried out the project work having “BookTrack” (Mini Project-II, ID201B) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself, and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Shruti Aggarwal
Assistant professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean,
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

BookTrack is a web-based platform designed to provide students and library staff with seamless access to library services and digital book records. The system allows users to search, reserve, borrow, and return books, while also offering access to personalized reading history and academic reference materials. The admin end manages book inventory, user activity, and system operations by generating secure access controls and enhancing the security of stored data. This project aims to replace traditional, inefficient library management methods with a secure, scalable, and user-friendly digital solution.

Key features of the system include:

- A user-friendly interface built using React.js.
- Secure storage and retrieval of book records and user data via cloud services.
- Role-based access control to manage privileges for students, librarians, and administrators.
- A digital repository of reading resources and borrowing history to support student learning and research.
- Scalable architecture using Java, Spring Boot, and MongoDB for efficient and real-time performance.
- Integration with Google Drive to allow secure backup of reports and digital book excerpts.
- Students can review their borrowing history, receive due date reminders, and get reading suggestions based on their interests.

Keywords: Library Management, BookTrack, Web Application, React.js, Java, Spring Boot, MongoDB, Cloud Services

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. I extend my deepest gratitude to my project supervisor, Ms. Shruti Agrawal for her invaluable guidance, encouragement, and insightful feedback throughout the development of this project. Her expertise and suggestions played a crucial role in shaping this work.

I am also immensely grateful to Dr. Akash Rajak, Dean, Department of Computer Applications, for his continuous support, insightful comments, and administrative assistance, which have significantly contributed to the successful completion of this project.

Finally, I extend my heartfelt thanks to my family for their unwavering encouragement, moral support, and constant belief in my abilities. Their love and motivation have been the driving force behind my dedication and perseverance. Without their support, the successful completion of this project would not have been possible.

This project is a result of collective efforts, and I am truly grateful to everyone who has contributed to its success.

Name – Kartik Agarwal
Roll -20202410116100096

TABLE OF CONTENTS

1. Certificate	3
2. Abstract	4
3. Acknowledgments	5
4. Table of Contents.....	6
5. Introduction	7
○ 5.1 Overview	7
6. Literature Review.....	8
○ 6.1 Evolution in Digital Certification	8-9
○ 6.2 Role of Modern Web Technologies	9-11
○ 6.3 Existing System.....	11-12
7. Project Flow	13
○ 7.1 Use Case Diagram.....	13
○ 7.2 ER Diagram.....	14-15
○ 7.3 Data Flow Diagram	15-17
○ 7.4 UI Dashboard	17-19
○ 7.5 Database Schema.....	20-22
○ 7.6 System Architecture	23-25
8. Implementation.....	26
○ 8.1 Technology Stack.....	26-27
○ 8.2 Features	27
○ 8.3 Challenges	27-29
9. Project Outcome.....	30-31
10. Project Output	32
11. Technical Output	33
12. Output.....	34-36
13. Conclusion.....	37
14. Future Scope.....	38-39
15. References	40

5.INTRODUCTION

5.1 Overview

In today's fast-paced digital landscape, educational institutions are increasingly transitioning from traditional administrative methods to technologically advanced solutions. One critical area demanding innovation is the management of library services and book circulation. Libraries play a crucial role in supporting academic success, research, and lifelong learning. However, conventional library systems that rely on manual record-keeping and paper-based processes are often inefficient, error-prone, and lack the accessibility and responsiveness expected in the digital age. The need for a reliable, scalable, and secure library management system has become more pressing than ever.

BookTrack addresses this demand by offering a comprehensive, cloud-based digital library management platform. It is a web-based solution that enables students and librarians to search, borrow, reserve, and manage books in a structured and efficient manner. In addition to handling core library functions, BookTrack also provides features such as personalized reading history, overdue reminders, and access to digital academic resources, thereby transforming the library into a dynamic learning hub.

The system is built using modern web technologies to ensure high performance, security, and a user-centered experience. The frontend of BookTrack is developed using React.js, providing a responsive and interactive interface that functions smoothly across different devices. The backend, powered by Java and Spring Boot, manages core operations, secure APIs, and user authentication. MongoDB is used as the NoSQL database to handle flexible data storage, including book inventories, borrowing logs, and user profiles. To facilitate secure digital record keeping, the system integrates the Google Drive API, enabling librarians to back up reports and scanned resources through secure, sharable links.

BookTrack's architecture is designed with scalability and modularity in mind, ensuring easy updates and system maintenance. Security is embedded in its core, with JWT (JSON Web Tokens) used for safe session management and role-based access control restricting user permissions based on their role—student, librarian, or admin. All data transmissions are protected through HTTPS encryption, and user data is isolated to maintain privacy and security. From a usability perspective, BookTrack streamlines the library experience for both students and staff. Students can check book availability, manage their borrowing activities, access digital materials, and receive due-date alerts through a clean and intuitive dashboard. Administrators and librarians can oversee book inventories, issue and return records, generate usage reports, and monitor system activity via a centralized control panel.

The platform also supports institutional sustainability goals by reducing the need for paper-based systems and physical storage. It aligns with the broader digitization initiatives of smart campuses and fosters a culture of efficient, technology-driven academic support.

In summary, BookTrack is more than just a library management tool. It is a modern, holistic solution tailored to the evolving needs of academic institutions—improving operational efficiency, enhancing the user experience, and advancing digital transformation goals. With its secure, scalable architecture and user-friendly features, BookTrack is set to redefine how libraries operate in the digital era.

6.Literature Review

6.1 Evolution in Digital Certification

The evolution of digital library systems is closely tied to the rapid advancements in information technology and the growing demand for efficient, secure, and scalable library resource management. For decades, educational institutions relied on paper-based cataloging and manual record-keeping to manage books, track borrowing activity, and maintain user records. These traditional systems, while foundational, were prone to issues such as human error, data loss, physical wear and tear, and limited accessibility.

As academic institutions expanded and student populations grew, the limitations of manual library management became increasingly evident. Paper-based systems were time-consuming, labor-intensive, and offered little to no support for real-time access or user interaction. Tasks such as searching for books, checking availability, and maintaining issue-return records were inefficient and often resulted in long queues, misplaced records, and poor user satisfaction.

The shift toward digital library systems emerged as a solution to these long-standing issues. A digital library system allows users to interact with library services through a centralized platform that facilitates book searches, reservations, borrowing, and returns in a fully automated manner. These systems are hosted on secure servers or cloud environments, allowing for uninterrupted access, instant updates, and accurate record maintenance.

Cloud computing has played a pivotal role in enabling this transformation. Cloud-based library platforms offer scalable infrastructure, allowing institutions to store large volumes of book records, transaction logs, and user data without physical constraints. Real-time access to these systems means students can search for books, reserve titles, and monitor their borrowing status anytime, from any location.

Another key advancement in digital library systems is the integration of smart technologies such as QR-based book tagging, barcode scanning, and real-time tracking. These innovations reduce the manual effort required by library staff, increase inventory accuracy, and streamline the issue-return workflow. Students can use web or mobile interfaces to scan codes, receive due-date alerts, and even access e-books and academic materials directly through the system. Platforms like Koha, Evergreen, and Alma are at the forefront of modern digital library management, offering features such as OPAC (Online Public Access Catalog), user analytics, and role-based access control. These platforms provide academic institutions with robust solutions for cataloging, user engagement, and inventory management—moving beyond just lending services to becoming comprehensive academic support hubs.

Security and data integrity have also seen major improvements. Digital library systems now implement secure authentication mechanisms such as role-based login, OTP verification, and JWT tokens to ensure that only authorized users can access or modify library records. All user activity is logged, encrypted, and backed up on secure servers, offering high reliability and data protection.

The COVID-19 pandemic further highlighted the importance of digital access to academic resources. As physical libraries closed or operated under restrictions, students and faculty increasingly turned to online libraries and e-resources. Institutions had to digitize their collections rapidly and implement platforms that allowed uninterrupted academic support remotely.

Additionally, digital library systems enable advanced analytics. Institutions can monitor book

usage patterns, identify high-demand resources, and make data-driven decisions on acquisitions and budgeting. Personalized recommendations based on reading history help increase student engagement and promote a culture of continuous learning.

In conclusion, the evolution from manual to digital library management marks a significant shift in how educational resources are accessed, organized, and utilized. Technologies such as cloud infrastructure, smart tagging, and secure digital platforms offer unmatched advantages in efficiency, scalability, and user satisfaction. For systems like BookTrack, embracing these innovations is essential in building a future-ready, student-focused library management solution.

6.2 Role of Modern Web Technologies

Modern web technologies have revolutionized the management and distribution of digital certificates, enabling educational institutions to build scalable, secure, and efficient systems that meet the evolving needs of students and administrators. These technologies not only enhance the user experience but also provide robust infrastructures that ensure data integrity, system reliability, and long-term scalability.

At the core of this digital shift are cloud-based platforms, which serve as the backbone for storing and accessing student records. Cloud storage eliminates the limitations of physical infrastructure, offering institutions the flexibility to scale their storage capacity as needed while maintaining data accessibility around the clock. Leading providers such as Google Cloud Platform (GCP), Amazon Web Services (AWS), and Microsoft Azure offer reliable solutions with advanced data redundancy, real-time availability, and integrated security protocols. In the case of Campus Vault, the Google Drive API is integrated to securely store student-uploaded certificates, allowing authorized users to access documents remotely without compromising data privacy.

Frontend technologies have also seen significant evolution, with frameworks like React.js setting new standards in user interface development. React.js enables developers to build dynamic, responsive web applications that deliver a seamless user experience across desktops, tablets, and mobile devices. Features like component-based architecture, real-time rendering, and virtual DOM updates contribute to the efficiency of frontend development. In Campus Vault, React.js is used to build intuitive interfaces that allow students to upload certificates,

review placement questions, and navigate through their dashboards with ease. Admin users, on the other hand, can interact with data-rich dashboards to manage documents and placement content efficiently.

On the backend, Java remains a popular and powerful language due to its stability, scalability, and community support. Combined with the Spring Boot framework, Java becomes even more powerful for web-based system development. Spring Boot simplifies application development by providing built-in support for creating RESTful APIs, handling authentication, integrating with databases, and configuring application services. It allows Campus Vault to process large volumes of requests securely and reliably while maintaining high performance even under peak load.

One of the most critical aspects of modern web platforms is security, particularly when dealing with academic records and personal information. JWT (JSON Web Tokens) are used in Campus Vault for stateless authentication, ensuring secure user sessions without storing session data on the server. Every request from a user carries a token that is verified at the backend before access is granted. This approach improves security and performance, especially in cloud-native environments.

Further, the integration of OAuth 2.0 protocols is essential for secure third-party authentication, especially when integrating services like Google Drive. OAuth 2.0 ensures that only authorized applications and users have access to sensitive APIs, reducing the risk of data breaches and unauthorized access. SSL (Secure Socket Layer) encryption is enforced across all HTTP communications in the Campus Vault system to protect data in transit from being intercepted or tampered with by malicious actors.

In addition to traditional authentication and encryption, role-based access control (RBAC) is implemented to manage user privileges. This ensures that students only access their own certificates while administrators have additional capabilities, such as document verification and placement content management. Such layered security approaches, powered by modern web technologies, form a comprehensive defense mechanism against data theft and unauthorized operations.

The synergy between these frontend, backend, cloud, and security technologies results in a robust platform that not only functions reliably but also scales gracefully with institutional growth. Moreover, the modular architecture enabled by these technologies ensures that features can be added or modified without disrupting the core system—making Campus Vault adaptable to future innovations such as blockchain credentialing or AI-powered analytics.

In conclusion, modern web technologies are the foundation of digital transformation in education. They enable systems like Campus Vault to deliver a secure, accessible, and user-friendly experience while upholding the integrity and confidentiality of academic records. Their continued evolution will further empower institutions to offer smarter, more efficient services to their students and staff.

6.3 Existing System

Despite the widespread advancement in digital infrastructure, many educational institutions still rely on outdated, manual methods for managing their library operations. These traditional systems involve physical book catalogues, handwritten issue registers, and in-person borrowing and return processes—procedures that are slow, inefficient, and prone to human error. Students are often required to visit the library physically to search for books, check availability, and complete borrowing formalities, resulting in long queues and limited accessibility.

This conventional library system presents several operational challenges. Book records are typically maintained on paper or basic spreadsheets, making them susceptible to damage, misplacement, and inconsistencies. When students need to locate a specific book or check their borrowing history, the lack of a centralized searchable database leads to delays and confusion. Moreover, there is no real-time inventory tracking, which causes difficulties in monitoring book availability or overdue returns.

Attempts at partial digitization—such as storing data in Excel or manually updating shared documents—often fall short of meeting modern expectations. These systems may digitize records superficially but still rely heavily on manual entry and lack essential features such as role-based access, user authentication, or integration with secure cloud storage. As a result, both students and staff continue to experience inefficiencies, and the system does not scale well with growing student populations or expanding inventories.

Another significant drawback of traditional systems is the lack of transparency and accountability. Without audit trails or activity logs, tracking book transactions and user activity becomes nearly impossible. This creates room for disputes over lost or overdue items and makes it difficult to evaluate system performance or improve services.

BookTrack is specifically designed to overcome these limitations by offering a fully digitized, centralized, and secure library management solution. The platform automates book inventory management, streamlines borrowing and returning workflows, and provides real-time access to book availability and user history through a web interface. Students can search, reserve, and manage books online, eliminating the need for frequent physical visits to the library.

By integrating with Google Drive, BookTrack ensures that important digital records—such as usage reports and book excerpts—are securely stored and easily accessible to authorized users. The system also includes features such as due date reminders, reading suggestions, and borrowing history analysis, all delivered through an intuitive user dashboard.

On the administrative side, the platform includes a robust admin panel where librarians and staff can add or remove books, track user activity, and monitor system usage. Automated processes replace manual tasks such as record updates and overdue tracking, significantly reducing administrative workload and human error.

In addition, BookTrack enhances security and accountability by using role-based access control (RBAC), JSON Web Tokens (JWT) for secure session management, and HTTPS encryption for all data transmission. Every transaction and update is logged, enabling accurate tracking and fostering transparency.

In summary, while existing library systems continue to limit operational efficiency and user convenience, BookTrack presents a transformative alternative. It digitizes the full lifecycle of library services—from book management to user engagement—providing a secure, scalable,

and user-friendly platform that aligns with the digital vision of modern academic institutions. In addition, BookTrack enhances security and accountability by using role-based access control (RBAC), JSON Web Tokens (JWT) for secure session management, and HTTPS encryption for all data transmission. Every transaction and update is logged, enabling accurate tracking, ensuring transparency, and building institutional trust. Only authorized personnel can modify or manage critical data, and all users are authenticated before performing any sensitive actions.

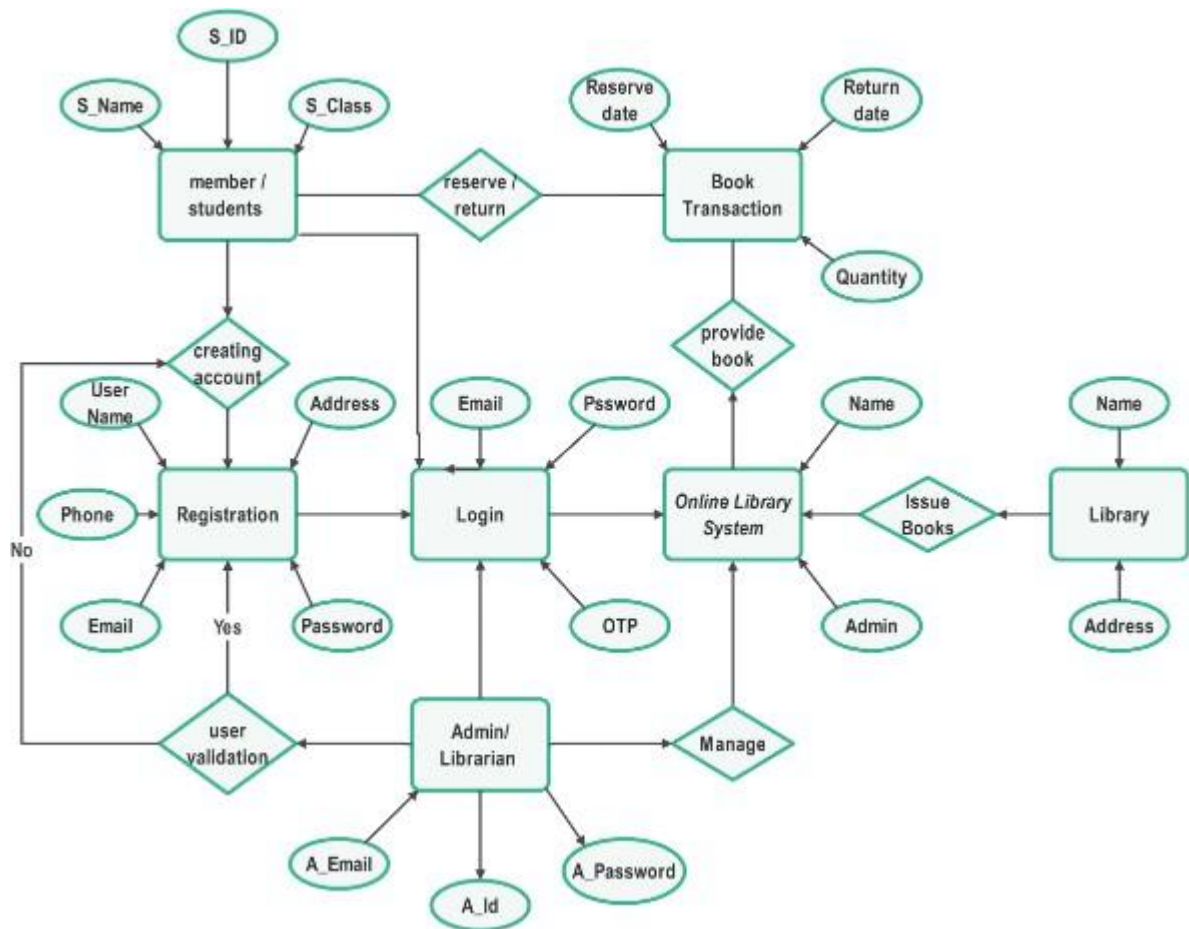
In summary, while existing library systems continue to limit operational efficiency, scalability, and user convenience, **BookTrack** presents a transformative alternative. It digitizes the full lifecycle of library services—from book management to user engagement—providing a secure, scalable, and user-friendly platform. Aligned with the vision of smart campuses and digital transformation in education, BookTrack ensures that institutions remain future-ready, efficient, and student-focused in their library operations.

7. Project Flow

7.1 Use Case Diagram



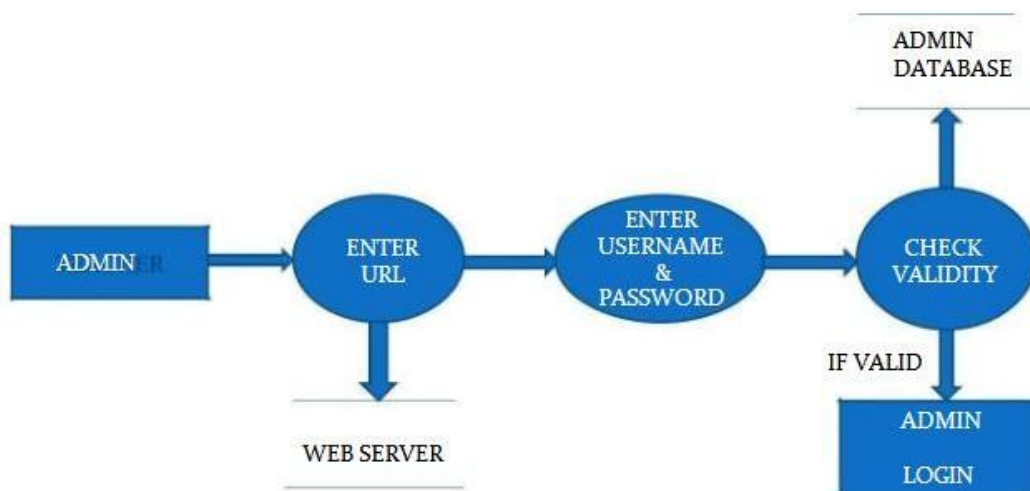
7.2 ER Diagram



7.3 Data Flow Diagram

7.3.1 Level 0 (Context Diagram):

1. Student (External Entity)
 - Interacts with the system to upload certificates, view/download certificates, review placement questions, and receive notifications.
 - Accesses the system via the student dashboard.
2. Admin (External Entity)
 - Manages and verifies certificates, maintains the placement question repository, generates Google Drive upload links, and accesses logs and student records.
 - Uses the admin dashboard to interact with the system.
3. Campus Vault (System – Central Processing Unit)
 - Serves as the centralized platform that processes user authentication, certificate storage/retrieval, placement content management, and secure data handling.
 - Integrates frontend (React.js), backend (Java with Spring Boot), authentication (JWT and OAuth 2.0), and cloud storage (Google Drive API).
 - Maintains security using SSL and role-based access control.



7.3.2 Level 1(Context Diagram):

Entities in Level 1 DFD:

1. Student End

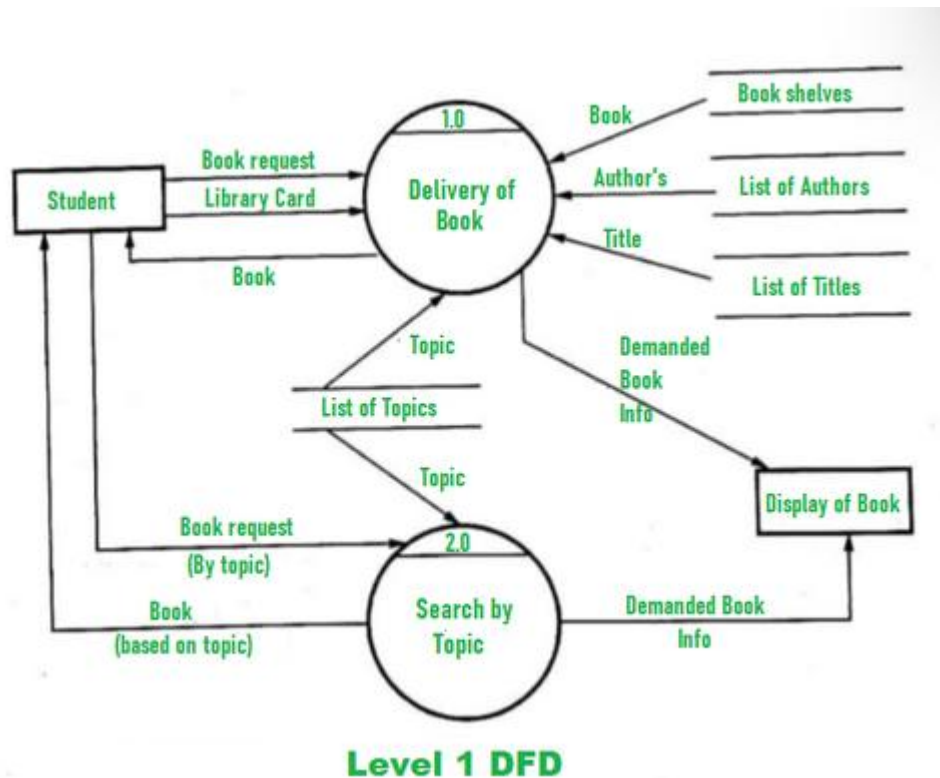
- View Certificates: Students can view their uploaded or received certificates.
- Upload Certificates: Students upload certificates using secure Google Drive links provided by the Admin.
- Review Certificates: Students verify the details of their uploaded certificates.
- Review Placement Questions: Students access placement preparation questions from the repository.

2. Management System

- Stores Certificates: Safely stores all uploaded certificates.
- Manages Placement Questions: Maintains and manages the placement question bank.
- Provides Secure Access: Handles user authentication, authorization, and ensures data security.

3. Admin End

- Access Certificates: Admins review, verify, approve, or reject student-submitted certificates.
- Manage Placement Questions: Admins add, edit, or delete questions in the placement question repository.



7.4 UI Dashboard

Homepage (Login Page)

1. User Management Module

- Purpose: Manages user registration, login, and profile details.
- Features:
 - o User registration (email, phone, library ID).
 - o Login and authentication (username/password).
 - o Role-based access (Student, Librarian, Admin).
- Technologies: PHP (backend), MySQL (database), HTML, CSS, JavaScript (frontend).

2. Book Catalog Management Module

- Purpose: Handles adding, updating, and searching books.
- Features:
 - o Admin can add/edit/delete books (title, author, ISBN, category, etc.).
 - o Users can search and filter books by title, author, or category.

- o Display book availability status (Available/Issued).

- Technologies: PHP, MySQL, JavaScript (AJAX for dynamic search).

3. Issue and Return Module

- Purpose: Manages book lending and return processes.

- Features:

- o Track book issues (user ID, book ID, issue date, due date).

- o Update the status when a book is returned.

- o Overdue tracking with fine calculation.

- Technologies: PHP, MySQL, JavaScript.

4. Overdue Notification Module

- Purpose: Notifies users about due or overdue books.

- Features:

- o Automatic reminders via email or SMS.

- o Overdue fine calculation and notification.

- o Integration with notification APIs (e.g., Twilio, SendGrid).

- Technologies: PHP, MySQL, API Integration (Email/SMS).

5. Reports and Analytics Module

- Purpose: Generates reports and tracks library usage statistics.

- Features:

- o Generate reports (e.g., most borrowed books, overdue fines collected).

- o View analytics like user activity, book borrowing trends.

- o Export reports to PDF or Excel.

- Technologies: PHP, MySQL, JavaScript.

6. Admin Dashboard Module

- Purpose: Central hub for library management.

- Features:

- o View and manage users, books, and transactions.
- o Add/edit/delete categories and authors.
- o Monitor system analytics (e.g., book availability, overdue fines).
- Technologies: PHP, MySQL, HTML, CSS, JavaScript (charts and graphs).

7. Search and Recommendation Module

- Purpose: Enhances user experience with efficient search and recommendations.
- Features:
 - o Advanced search options (filters by author, category, year).
 - o Recommend books based on borrowing history or popular trends.
- Technologies: PHP, MySQL, JavaScript..

7.5 Database Schema

1.Users Table

Stores user information.

Column Name	Data Type	Constraints
User id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL
role	ENUM	('student', 'faculty', 'admin') NOT NULL

2.Documents Table

Stores uploaded documents.

Column Name	Data Type	Constraints
Document id	INT	PRIMARY KEY, AUTO_INCREMENT
User id	INT	FOREIGN KEY REFERENCES Users(user_id) ON DELETE CASCADE
title	VARCHAR(255)	NOT NULL
File path	VARCHAR(255)	NOT NULL
Uploaded at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

3.Permissions Table

Manages document access.

Column Name	Data Type	Constraints
Permission id	INT	PRIMARY KEY, AUTO_INCREMENT
Document id	INT	FOREIGN KEY REFERENCES Documents(document id) ON DELETE CASCADE
User id	INT	FOREIGN KEY REFERENCES Users(user

		id) ON DELETE CASCADE
Access type	ENUM	('view', 'edit', 'download') NOT NULL

4.Categories Table

Stores document categories.

Column Name	Data Type	Constraints
Category id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	UNIQUE, NOT NULL

5.Document Categories Table

Associates documents with categories.

Column Name	Data Type	Constraints
Document id	INT	FOREIGN KEY REFERENCES Documents(document id) ON DELETE CASCADE
Category id	INT	FOREIGN KEY REFERENCES Categories(category id) ON DELETE

6.Notifications Table

Stores notifications for users.

Column Name	Data Type	Constraints
Notification id	INT	PRIMARY KEY, AUTO_INCREMENT
User id	INT	FOREIGN KEY REFERENCES Users(user id) ON DELETE CASCADE
message	TEXT	NOT NULL

7.6 System Architecture

The system architecture of Campus Vault is built on a modern full-stack framework that emphasizes scalability, security, and user-centric design. The architecture follows a client-server model where students and administrators interact through a responsive web interface, while the backend handles data processing, certificate management, and secure communication.

7.6.1 Frontend Layer (Client-Side)

- Developed using React.js, the frontend provides a responsive and dynamic interface for both students and administrators.
- Enables user functionalities such as certificate upload, viewing previous placement questions, and dashboard interaction.
- Includes secure forms and validations for user registration, login, and data submission.
- Implements routing and state management for seamless navigation across features.
- Designed with accessibility and responsiveness in mind to ensure usability across all devices.

7.6.2 Backend Layer (Server-Side)

- Built using Java with Spring Boot framework, ensuring modular development and high performance.
- Handles business logic, API routing, authentication, user roles, and document access control.
- Utilizes Spring Security and JWT (JSON Web Token) for secure role-based access control between admin and student roles.
- Manages the orchestration of certificate uploads to Google Drive and generation of secure access links.
- Includes error handling and response validation for API endpoints to improve system reliability.
-

7.6.3 Database Layer

- MongoDB serves as the primary database, offering a flexible and scalable NoSQL solution for storing user data, certificates, placement questions, and review metadata.
- Collections include user profiles, documents, access permissions, and notification logs.
- Indexed data ensures quick search and retrieval, supporting efficient operations for large-scale datasets.

7.6.4 Cloud Storage Integration

- Google Drive API is integrated to provide secure cloud-based storage for student certificates.
- Each certificate upload is authenticated using Google OAuth tokens to ensure secure access.
- The system automatically generates access-controlled links for both student uploads and admin verification purposes.
- Supports file metadata tagging and folder-wise organization for structured storage.
-

7.6.5 Communication & Security Layer

- RESTful APIs facilitate interaction between the frontend and backend, ensuring modular and stateless communication.
- JWT tokens are used for maintaining session security and identifying authenticated users.
- HTTPS protocols are enforced to encrypt data in transit and protect against eavesdropping and data tampering.

7.6.6 System Workflow

- User Authentication
 - New users register with their email and credentials.
 - Upon login, JWT tokens are issued and used for session handling and secure navigation.
- Certificate Upload and Review
 - Students upload certificates using a Google Drive link provided by the admin panel.
 - Uploaded certificates are tagged and stored securely in the cloud.
 - Students can review uploaded certificates and view placement questions for preparation.
- Admin Panel Functions
 - Admins review certificate submissions, manage placement questions, and control access permissions.
 - The admin dashboard supports analytics and summary views for quick management.
- Data Access and Security
 - Role-based access control ensures that sensitive operations like certificate approval and deletion are restricted to admin users.
 - Students are restricted to view/download operations within their own profiles.
- This modular and secure architecture allows Campus Vault to efficiently manage digital certifications, providing seamless interaction, robust security, and real-time performance for educational institutions.

Implementation

8.1 Technology Stack

Phase 1: Frontend Development

- **User Interface Design & Routing**

Developed using React.js to ensure a fast, interactive, and responsive user experience. Tailwind CSS was used for designing modern, mobile-friendly UI components. Integrated React Router for smooth navigation between modules like login, dashboard, upload, and certificate review.

- **Authentication Interface**

Designed forms for student and admin login, signup, and password reset functionalities. Implemented error handling and user feedback for common input errors.

Developed a dynamic role-based dashboard that changes based on the logged-in user's role (student or admin).

Phase 2: Backend Development

- **Secure REST API Development**

Built using Java and Spring Boot to handle user registration, login, certificate uploads, and access management.

JWT (JSON Web Token) was implemented to enable secure, stateless user authentication.

Endpoints were protected using Spring Security and role-based access control.

- **Certificate & User Management**

Developed APIs to allow admins to upload certificates, generate secure Google Drive links, and manage student access.

Created endpoints for students to fetch, preview, and download their certificates securely.

Enabled admins to manage placement questions in bulk using structured APIs.

Phase 3: Cloud Storage & Integration

- **Google Drive API Integration**

Configured OAuth 2.0 for authenticating and authorizing access to Google Drive.

Programmatically generated Google Drive folder links for each student to upload their certificates.

Allowed secure read-only access to students for reviewing their stored documents. Added fallback mechanisms in case of API call failure or expired tokens.

- **MongoDB Integration**

MongoDB was used to store user profiles, document metadata, access permissions, and

placement content.

Implemented document indexing for fast data retrieval.

Ensured data consistency and rollback support in case of partial failures.

Phase 4: Testing & Deployment

- **Unit & Integration Testing**

Backend APIs were tested using Postman and integrated with JUnit test cases.

Edge case handling was ensured by simulating invalid token access, unauthorized document retrieval, and upload errors.

MongoDB operations were tested with mock data to validate schema and access patterns.

- **Deployment & Hosting**

Hosted on AWS using an EC2 instance and S3-compatible storage.

Environment variables were managed using .env configurations for security.

Implemented CI/CD pipelines using GitHub Actions to automate build and deployment.

HTTPS was enabled using SSL certificates for secure communication.

8.2 Features

Library Management System provides the following key features:

- **Member Features:**

- **View & Borrow Books:** Members can search, view details, and borrow books anytime.

- **Return Books:** Members can return borrowed books through the system.

- **Review Borrowing History:** Members can check their past borrowed books and due dates.

- **Access Digital Resources:** Members can access e-books, journals, and other digital materials.

- **Admin Features:**

- **Manage Books:** Add, update, and delete book records securely.

- **Manage Member Accounts:** Maintain member registrations, renewals, and permissions.

- **Issue & Return Management:** Oversee book issuing and returns with due date tracking.

- **Provide Notifications:** Send alerts for due dates, overdue books, and new arrivals.

8.3 Challenges

During the development of the Library Management System project, the team faced several challenges that required careful planning, innovative problem-solving, and continuous improvement. These challenges included technical implementation complexities, ensuring a seamless user experience, enforcing strong security measures, and integrating with cloud-based resources.

1. Secure Storage & Retrieval

One of the primary concerns of the project was to ensure that library records—such as book details and user borrowing history—were stored and retrieved securely from the cloud.

- Ensuring confidentiality involved encrypting data during transmission using HTTPS and integrating access-controlled cloud storage links.
- Integrity checks were implemented to confirm that records were not altered or tampered with post-upload.
- Designing a system that met institutional data privacy standards without compromising usability for library users required balancing security with performance

2. Role-Based Access Control (RBAC)

Implementing secure yet flexible access control mechanisms for both library users and administrators posed significant challenges.

- The system needed to differentiate privileges between admins (who manage all library content) and users (who only manage their own borrowing records and requests).
- JWT (JSON Web Tokens) were used to manage sessions and enforce role-based permissions securely.
- Managing token expiration, validation, and revocation without affecting user experience required thorough backend logic and testing.
- Ensuring that no unauthorized access occurred through bypassing endpoints or manipulating requests was a critical testing point..

3. Data Consistency & Performance

As the system is expected to scale and handle large amounts of library data, ensuring consistent and high-performance database operations became crucial.

- MongoDB, while flexible, required carefully designed schemas and indexing strategies to avoid slow queries.
- Data models had to support nested structures (like user borrowing records, permissions, and notifications) while still maintaining quick read/write capabilities.
- Ensuring atomic operations during document upload and metadata update was important to avoid partial or corrupted entries.
- Bulk operations, such as admin-side certificate reviews or question repository updates, had to be optimized for performance and minimal system load.

4. User Experience & Interface Design

Designing an intuitive and accessible interface for both library users and administrators came with its own complexities.

- The platform needed to cater to a diverse user base, including students unfamiliar with cloud platforms.
- Components were developed using React.js and Tailwind CSS to ensure responsiveness and

accessibility across devices.

- Designing a simple upload process while integrating Google Drive required visual cues, error handling messages, and real-time progress feedback.
- Admin dashboards needed to present a comprehensive view of certificate requests and placement content while remaining lightweight and uncluttered.

5. Integration with Google Drive

A key innovation in the library system was the integration of Google Drive for secure, scalable document storage—but this introduced numerous technical hurdles.

- Setting up OAuth 2.0 for Google Drive API authentication involved extensive configuration and secure token handling.
- Managing API quotas, permissions, and real-time failures (e.g., expired tokens or interrupted uploads) was necessary to provide a seamless experience.
- Ensuring that each user's uploaded document was routed to the correct folder and did not overwrite existing data required implementing a folder structure with unique identifiers.
- Admins needed controlled access to all documents while ensuring that no user could access another's document, making permission logic complex and multi-layered.

This comprehensive approach to addressing challenges during development has strengthened the overall reliability, security, and scalability of the library platform. Overcoming these hurdles has not only enhanced the current functionality but has also laid the foundation for future feature enhancements and system expansions.

9. Project Outcome

• i. Centralized Certificate Repository

Campus Vault serves as a digital vault where students can securely upload, access, and retrieve their academic and extracurricular certificates.

- Ensures organized storage with categorized folders per student.
- Reduces dependency on physical documentation and manual record-keeping.

• ii. Secure Access and Role-Based Permissions

The system includes user authentication and differentiated access for students and administrators.

- JWT-based login ensures session integrity and secure access.
- Admins can manage certificates and placement questions, while students can only view and upload their documents.

• iii. Placement Preparation Module

A dedicated module allows students to review previous placement drive questions.

- Helps students prepare effectively for interviews.
- Admins can update and manage the question bank regularly.

• iv. Google Drive Integration

Certificates are stored securely via Google Drive API, ensuring scalability and reliability.

- Students upload certificates through unique, access-controlled links.
- Admins review submissions and validate authenticity using the Drive interface.

• v. Intuitive and Responsive UI

Developed using React.js and Tailwind CSS, the frontend is optimized for user experience.

- Clean dashboard for managing certificates and placement resources.
- Accessible on both desktop and mobile devices.

• Testing Phase for BookTrack

The system was tested extensively to ensure performance, security, and user experience meet the required standards. Multiple testing strategies were employed.

• Unit Testing

- Controllers, services, and utility functions were tested individually.
- Frameworks: JUnit for Java-based backend components.

• Integration Testing

- Validated API flows between frontend and backend.
- Tools: Postman (for API testing), Spring Boot's testing framework for service-layer validation.

- **Functional Testing**

- Tested all use-case scenarios including certificate uploads, login authentication, and role-based access.
- Approach: Manual testing of dashboard and student workflows.

- **Performance Testing**

- Simulated multiple concurrent users uploading and accessing certificates.
- Ensured MongoDB and Google Drive handled peak load efficiently.

- **Security Testing**

- Verified protection of sensitive data using JWT and HTTPS.
- Checked for unauthorized access attempts and ensured token validation.

- **Bug Fixing & Debugging**

- Identified UI inconsistencies, certificate upload failures, and access control bugs.
- Debugged backend using Spring Boot logs and frontend using browser dev tools.

- **Deployment Readiness**

- Verified compatibility across different browsers and devices.
 - Final version deployed with SSL and environment configurations.
-

10. Project Output

Functional Output:

User Registration and Authentication

- Students and admins can securely register and log in using JWT-based authentication.
- Tokens are used to access protected API routes, ensuring secure navigation.

Dashboard and Certificate Access

- Students access a personalized dashboard showing uploaded certificates and placement material.
- Admins have enhanced access to manage users, verify documents, and upload question sets.

Certificate Upload and Retrieval

- Students upload documents via Google Drive link.
- Admins verify and manage the submissions through their own dashboard panel.

Placement Questions Review

- Students browse curated lists of previous placement questions to support their preparation.

11. Technical Output:

Frontend Output

- Built with React.js and Tailwind CSS, the interface is clean, modular, and responsive.
- Implements routing, protected components, and state management for seamless interaction.

Backend Output

- Developed with Spring Boot, the backend handles all authentication, user management, and business logic.
- MongoDB stores certificate metadata, user roles, and placement content.

Google Drive Output

- Google Drive API integration provides real-time upload and access capabilities.
- Folders and files are automatically organized based on user ID.

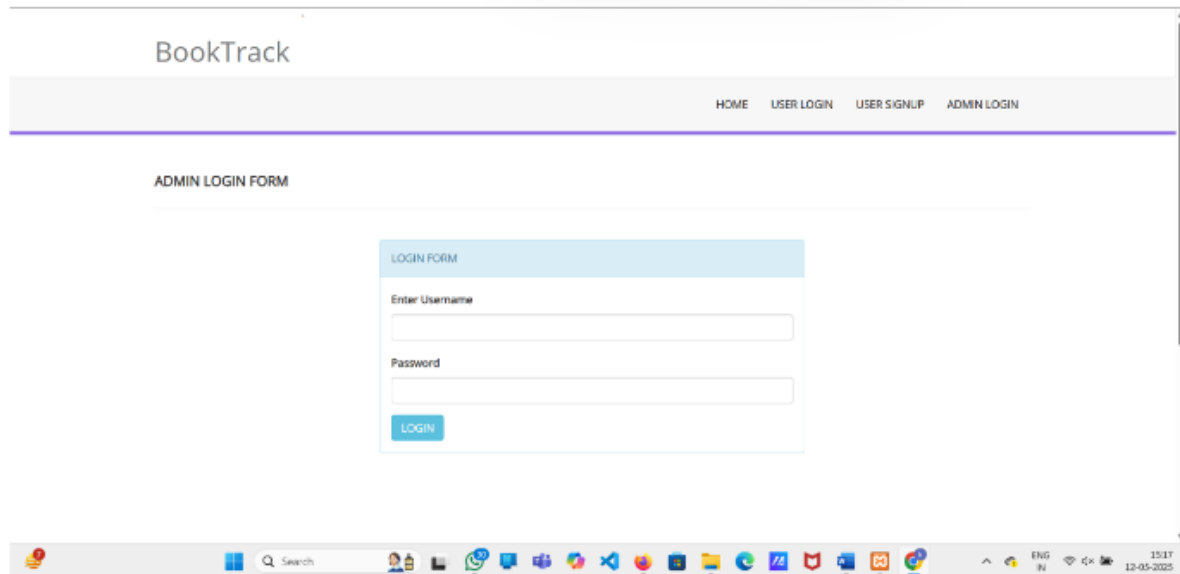
Educational and Practical Impact:

Campus Vault enables students to take control of their academic documentation in a secure, digital environment.

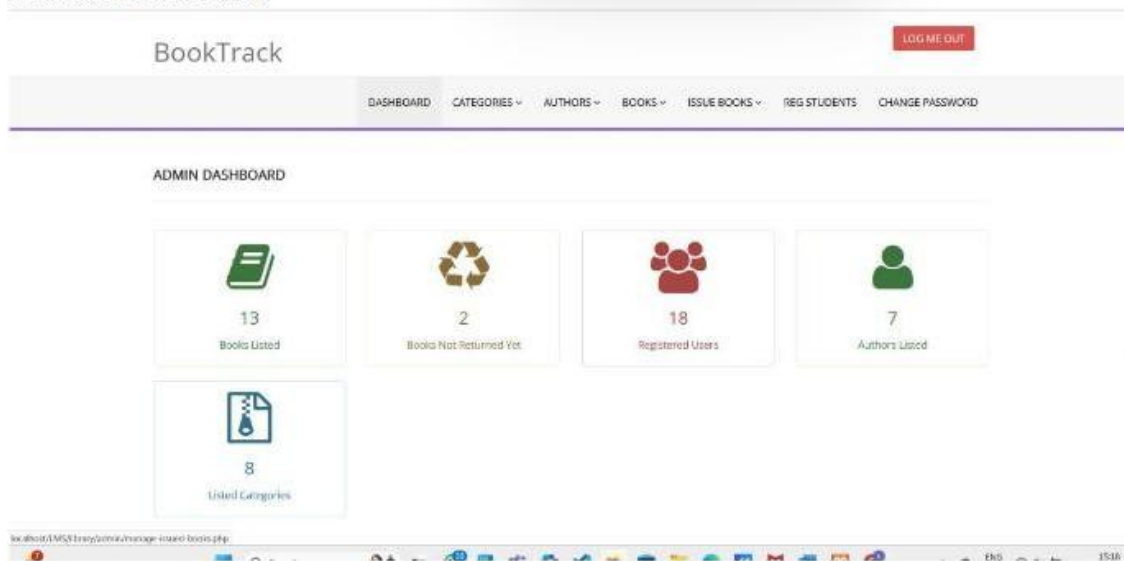
- Promotes tech-literacy in digital records and cloud usage.
 - Facilitates smooth documentation processes for institutional admin staff.
 - Acts as a preparation hub through integrated placement material access.
-

12. OUTPUT

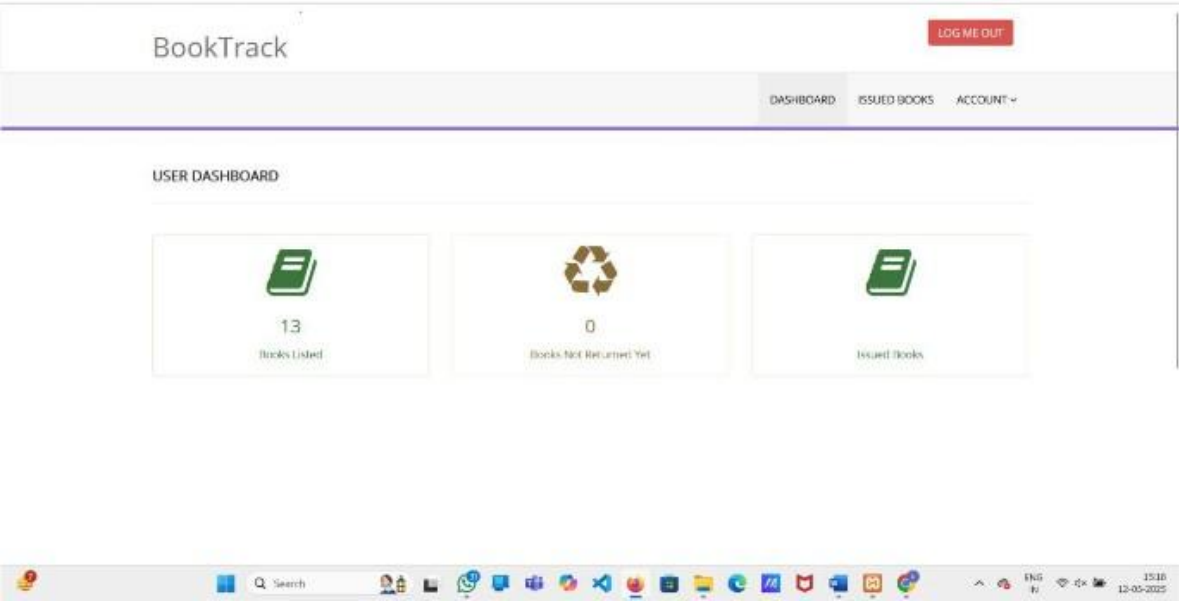
Home Page



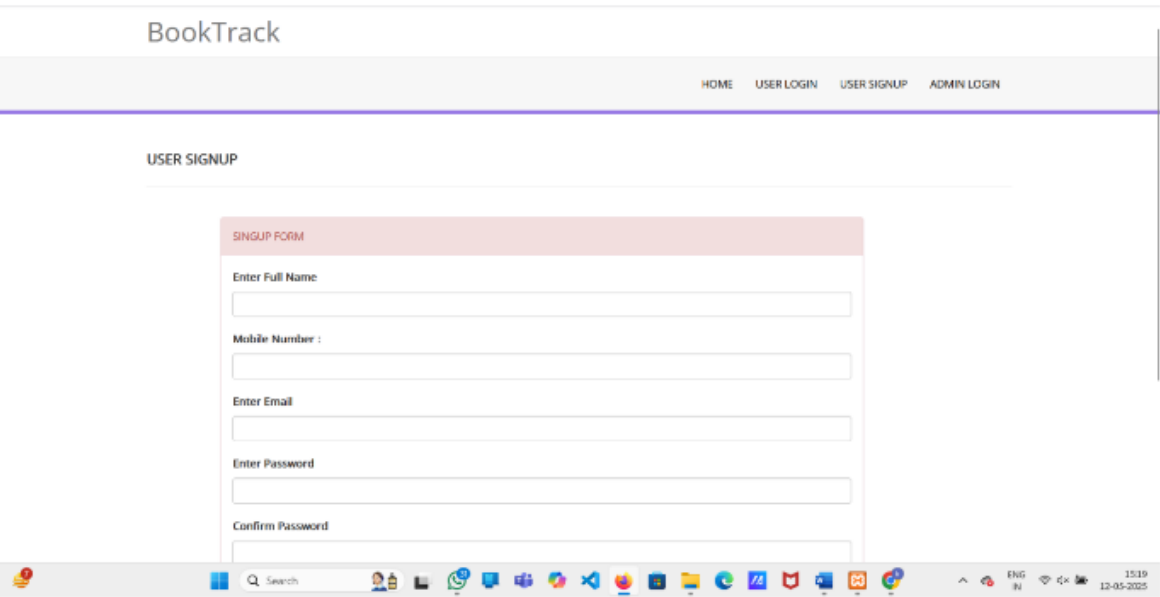
Admin Dashboard



User Dashboard



User Signup



MANAGE AUTHORS

Authors Listing

10 records per page

Search:

#	Author	Creation Date	Update Date	Action
1	Prashanta Kumar Patra	2024-12-10 11:28:36		Edit Delete
2	Asim Kumar Sen	2024-12-10 11:46:07		Edit Delete
3	Rajiv Chopra	2024-12-17 11:16:50		Edit Delete
4	Er. Rishabh Anand	2024-12-17 11:19:21		Edit Delete

Showing 1 to 4 of 4 entries

Previous1Next

MANAGE CATEGORIES

Categories Listing


10 records per page

Search:

#	Category	Status	Creation Date	Update Date	Action
1	Romantic	Active	2024-01-31 12:53:03	2024-02-04 12:03:43	Edit Delete
2	Technology	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
3	Science	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
4	Management	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
5	General	Active	2024-01-31 12:53:03	2024-02-04 12:03:51	Edit Delete
6	programming	Active	2024-01-31 12:53:03	2024-12-09 13:46:53	Edit Delete
7	fiction	Inactive	2024-12-09 13:35:18	2024-12-10 11:03:39	Edit Delete
8	agriculture	Active	2024-12-09 13:03:45	2024-12-10 11:03:29	Edit Delete

Showing 1 to 8 of 8 entries

Previous1Next



Online Library
Management System

LOG ME OUT




DASHBOARDCATEGORIES▼AUTHORS▼BOOKS▼ISSUE BOOKS▼REG STUDENTSCHANGE PASSWORD

MANAGE BOOKS

Books Listing

10 records per page

Search:

#	Book Name	Category	Author	ISBN	Price	Action
1	 Operating System	Technology	Prashanta Kumar Petra	123456	299.00	Edit Delete
2	 DBMS	Technology	Asim Kumar Sen	1234	199.00	Edit Delete
3	 Computer Networking	Technology	Er. Rahabih Anand	234	999.00	Edit Delete

Showing 1 to 3 of 3 entries

[Previous](#) [1](#) [Next](#)

13. Conclusion

The Library Management System project represents a significant advancement in the way libraries operate by transitioning from traditional manual processes to a fully automated and efficient digital platform. This system effectively streamlines critical library functions such as cataloging, issuing, returning, and tracking books, while managing member information and book inventories with accuracy and speed. Leveraging modern technologies ensures that all transactions are recorded in real-time, which minimizes errors, prevents data loss, and greatly enhances overall user experience for both librarians and library members. The system's intuitive interface simplifies access to library resources, enabling users to quickly search for, reserve, or renew books without the need for physical visits, thus promoting greater engagement and accessibility.

Furthermore, the role-based access control embedded in the system ensures that different user roles—such as librarians, members, and administrators—have appropriate levels of access, maintaining security and integrity of the library's operations. Automated alerts and notifications improve communication by reminding users about due dates or overdue returns, reducing fines and improving resource availability. The system also supports detailed reporting and analytics, empowering library staff to make informed decisions on resource management, budgeting, and acquisitions.

In addition to operational benefits, the Library Management System contributes significantly to knowledge sharing and educational support by enabling timely access to information and resources. This fosters an environment conducive to learning and research, aligning with the mission of modern libraries to serve as vital academic and community hubs. Moreover, by digitizing library services, the system promotes sustainability by reducing paper use and manual paperwork, aligning with eco-friendly institutional policies.

Overall, this project lays a strong foundation for future enhancements such as integrating mobile applications, supporting digital content like e-books, implementing RFID-based tracking, or even connecting multiple libraries in a networked consortium. The Library Management System not only modernizes library operations but also enhances user satisfaction, resource utilization, and institutional efficiency, making it an indispensable tool for today's academic and public libraries. It stands as a critical step toward embracing digital transformation in library services, ensuring libraries remain relevant and valuable in the digital age.

14. Future Scope

The Library Management System lays a strong foundation for efficient resource handling and user-friendly services. Future enhancements could include AI-powered search and personalized recommendations, mobile app support for easy access, and real-time inventory updates. Adding multi-language support and managing digital resources like e-books will broaden accessibility. Security can be improved with biometric authentication and blockchain for transaction integrity. Integration with institutional ERP and library networks can streamline operations and resource sharing. These improvements will make the library smarter, more secure, and better suited to meet evolving user needs in the digital age.

Integration with Blockchain Technology

Incorporating blockchain technology into the Library Management System can enable secure, tamper-proof tracking of book transactions, membership records, and digital resource access. This ensures data integrity and transparency, reducing risks of fraud or unauthorized changes. Blockchain's decentralized verification allows instant, trustworthy audits of lending history and user credentials, boosting trust among library staff and users while enhancing accountability and security.

Artificial Intelligence (AI) Integration

Integrating AI and machine learning into the Library Management System can provide personalized book recommendations and resource suggestions based on user reading habits and preferences. AI can analyze borrowing patterns to optimize inventory and predict demand for specific titles. Additionally, AI-powered analytics can help librarians identify gaps in collection diversity and improve user engagement through tailored services and smarter resource management.

Development of a Mobile Application

Creating a dedicated mobile app for the Library Management System will enhance accessibility, allowing users to search, borrow, and return books conveniently from their smartphones. Features like push notifications for due dates, QR code scanning for quick book checkouts, offline access to reading lists, and integration with calendar apps will improve user engagement and streamline library operations. This mobile accessibility will encourage greater usage and make library resources more reachable anytime, anywhere.

Automated Digital Transcript Generation

Integrating automated transcript generation within the Library Management System can streamline record-keeping and reporting. Based on borrowing history, fines, and membership details, the system can automatically compile digital summaries of user activity, book borrowings, and reading achievements. This automation reduces manual paperwork, minimizes errors, and accelerates processes such as membership renewals, library audits, and usage reporting, enhancing administrative efficiency and user convenience.

Internationalization and Multilingual Support

To serve a diverse and global user base, the Library Management System can be enhanced to support multiple languages and regional formats. This includes localized user interfaces, date and currency formats, and compatibility with international library classification standards. Such improvements will make the system accessible to users from different linguistic and cultural backgrounds, facilitating easier resource access and promoting inclusivity. Additionally, supporting international standards will enable collaboration with global libraries and institutions, expanding resource sharing and user engagement worldwide.

E-Signature and Digital Authorization Capabilities

Integrating secure e-signature functionality within the Library Management System will enable authorized librarians or administrators to digitally approve and authenticate library records, membership documents, and loan agreements. This eliminates reliance on physical signatures, accelerates processing times, and enhances document security by preventing unauthorized changes. Combined with audit trails and timestamping, digital signatures will improve legal compliance, ensure accountability, and streamline administrative workflows.

15. Refereneces

- **React.js Documentation**
Official documentation and guides for React.js, a popular JavaScript library for building user interfaces.
Available at: <https://react.dev>
- **Spring Boot Documentation**
Comprehensive resources for Spring Boot, a framework used to create stand-alone, production-grade Spring-based applications with minimal configuration.
Available at: <https://spring.io/projects/spring-boot>
- **MongoDB Documentation**
Official documentation for MongoDB, a NoSQL database known for its scalability and flexibility, used for storing JSON-like documents.
Available at: <https://www.mongodb.com>
- **Google Cloud Platform Documentation**
Documentation and developer resources for Google Cloud services, including APIs, storage solutions, and cloud computing infrastructure.
Available at: <https://cloud.google.com>