INTRODUCTION

In today’s world, technology plays a major role in how organizations manage information, and libraries are no exception. Many libraries still rely on manual processes such as paper records, handwritten logs, and physical book tracking. These methods are slow, stressful, and often lead to mistakes. As a result, modern libraries now require digital systems that can store information accurately, streamline work for staff, and provide better experience for users.

This project focuses on designing and building a Smart Library Management System using PostgreSQL. The system is designed to handle essential library operations, including adding books, managing authors, creating user accounts, tracking loans, and controlling access to various functions based on user roles. The database is designed using proper relational techniques, including ER diagrams, normalization, and foreign keys to maintain data accuracy and prevent duplication.

The Smart Library is built to be secure, efficient, and easy to use. It uses PostgreSQL features like schemas, constraints, indexing, and role-based permissions to ensure that data is well organized and that users only have access to the parts of the system that match their responsibilities. By automating key tasks such as borrowing, returning, and updating available copies, the system helps reduce manual work and improves the overall management of library resources.

REQUIREMENTS SUMMARY

Functional Requirements

1. User login with authentication & role validation.
2. Dashboard showing statistics:
   - Total books
   - Authors
   - Members
   - Active loans
   - Book clubs
   - Available copies
3. Add, edit, delete operations:
   - Books
   - Authors
   - Members
   - Book Clubs
4. Borrow and return books.
5. Search books by title, author, or ID.
6. Automatic update of book availability.
7. Admin-only settings (deleting, system management).
8. Librarian permission to create and loan books.
9. Member read-only access.

Non-Functional Requirements

1. Secure password storage using PBKDF2 hashing (Passlib).
2. User-friendly GUI (PyQt5).
3. Real-time updates on dashboard.
4. Normalized relational database (3NF).
5. OS compatibility: Windows, macOS, Linux.

DATABASE DESIGN

Entities

- Users (user_id, username, password_hash, role_id)
- Roles (role_id, name)
- Books (book_id, title, genre, total_copies, available_copies)
- Authors (author_id, first_name, last_name)
- Book_Author (book_id, author_id)
- Members (member_id, first_name, last_name, phone)
- Loans (loan_id, book_id, member_id, loan_date, return_date, status)
- BookClubs (club_id, club_name, description)
- ClubMembers (club_id, member_id)

GUI WORKFLOW & USER EXPERIENCE

1. Login Window

    - Username + Password
    - Validates using PostgreSQL
    - Hash verification using passlib
    - Loads DashboardWindow with role settings

2. Dashboard Window

    - Displays real-time statistics
    - Buttons:
        - Books
        - Authors
        - Members
        - Loans
        - Book Clubs
        - Logout

3. Book Management

    - View all books
    - Search books
    - Add new book
    - Edit book
    - Delete book
    - Real-time update of available copies

4. Members Management

    - Add new member
    - View all members
    - Edit
    - Delete

5. Loan Management

- Borrow book
- Return book
- Availability updates immediately

6. Book Clubs

- Add club
- View clubs
- Add members to club

7. Borrow & Return Workflow

Borrow

- Select book
- Insert into loans table
- available_copies → available_copies - 1

Return

- Update loan status
- available_copies → available_copies + 1

Integrity is protected with constraints preventing negative copies.

8. Role-Based Access

Administrator

- View everything
- Delete users, books, members
- Change settings
- Cannot borrow books

Librarian

- Add books, members, authors
- Borrow/return books
- Manage clubs
- Cannot delete system-level data

Member

- View only
- No adding
- No editing
- No deleting

9. System Security

- Passwords are secured using pbkdf2_sha256 hashing
- SQL injections prevented via parameterized queries
- User roles enforced at GUI level and database level

CONCLUSION

The Smart Library Management System represents a complete and modern solution for managing essential library operations in an efficient and organized way. Using PostgreSQL, structured database design, and clearly defined user roles, the system ensures that information is stored accurately, tasks are automated, and users interact with the library more smoothly and transparently.

The project successfully demonstrates how database technologies, GUI tools, and software engineering principles can be combined to solve real-world problems. By implementing features such as book management, user authentication, borrowing and returning workflows, and dashboard monitoring, the system reduces manual work and improves the reliability of library processes.

The use of ER diagrams, normalization, and constraints helped create a strong database foundation, while the GUI design provided a user-friendly interface for librarians, members, and administrators. Each user role is given the appropriate level of access, showing the importance of security and permission control in information systems.

Overall, this project not only addresses the operational needs of a modern library but also provides valuable hands-on experience in database systems, system analysis, and software development. It highlights how technology can support better decision-making, strengthen data management, and create a smarter and more effective library environment.