

TECH DIVAS

PROJECT REPORT : LIBRARY MANAGEMENT SYSTEM



JAVA PROGRAMMING

E1UA307C

*TEAM LEADER : PRERNA GUPTA*

1.HARSHITA SINGH 23SCSE1010572

2. PRERNA GUPTA 23SCSE1011480

3.SOMYA PANDEY 23SCSE1011315

4.SWASTIKA CHATTERJEE 23SCSE1010910

**Project Report: Library Management System**

**Introduction**

The Library Management System (LMS) is a software solution designed to automate and streamline various operations within a library. This report details the development process, functionality, and technological implementation of the LMS, utilizing Java Swing for the front-end interface and MySQL as the back-end database. The system aims to enhance operational efficiency, improve user experience, and provide a robust framework for managing library resources.

**Objectives**

* **Automation**: Automate routine library tasks to reduce manual workload and errors.
* **User Management**: Implement a secure user authentication system for admins and staff.
* **Inventory Control**: Maintain an accurate inventory of books and staff records.
* **User-Friendly Interface**: Develop an intuitive interface for seamless navigation and interaction.

**Technology Stack**

* **Front End**: Java Swing
  + **IDE**: NetBeans
  + Java Swing provides a rich set of GUI components that facilitate the development of desktop applications with an attractive user interface.
* **Back End**: MySQL
  + MySQL is a reliable relational database management system that offers robust data handling capabilities, ensuring data integrity and security.

**System Functions**

**Front End Functions**

1. **Login Page**
   * **User Authentication**: The system includes a secure login page where admins and staff enter their credentials. This is essential for controlling access to sensitive library operations.
   * **Error Handling**: The login process includes mechanisms for handling incorrect username or password entries, providing user feedback to enhance security.
2. **Books Available**
   * **Display Functionality**: The system displays a comprehensive list of books currently available in the library, including titles, authors, and categories.
   * **Search and Filter Options**: Users can search for specific titles or filter the list by categories, making it easier to locate desired materials. The interface supports pagination for larger datasets.
3. **Staff Details**
   * **Management View**: Staff details can be viewed and managed from a dedicated section, displaying roles and responsibilities clearly.
   * **Editing Capability**: Admins can edit staff information, ensuring that records remain current and accurate.
4. **Add Books**
   * **Form for New Entries**: A user-friendly form allows for the easy addition of new books, including necessary details such as title, author, ISBN, and category.
   * **Validation Checks**: The system validates input fields to prevent errors, ensuring that all necessary information is provided before submission.
5. **Remove Books**
   * **Deletion Functionality**: Admins can remove books from the system when they are no longer part of the inventory.
   * **Confirmation Prompts**: A confirmation dialog appears to prevent accidental deletions, enhancing data integrity.
6. **Add Staff**
   * **Staff Registration**: New staff members can be added to the library system through a designated form.
   * **Role Assignment**: Admins can assign roles during the addition process, facilitating better management of responsibilities.
7. **Remove Staff**
   * **Staff Removal Process**: The system allows for the removal of staff members as needed, maintaining accurate staff records.
   * **Confirmation Process**: Similar to book removal, a confirmation prompt helps to avoid unintended deletions.

**Back End Structure**

The back end of the LMS is structured around three primary tables in the MySQL database:

1. **Admin Table**
   * **Purpose**: Stores admin credentials for system access.
   * **Fields**:
     + admin\_id: Unique identifier for each admin.
     + username: Login name for the admin.
     + password: Hashed password for security, ensuring that sensitive data is protected.
2. **Books Table**
   * **Purpose**: Maintains detailed information about the library's book inventory.
   * **Fields**:
     + book\_id: Unique identifier for each book.
     + title: Title of the book.
     + author: Author(s) of the book.
     + ISBN: International Standard Book Number, providing a unique identifier for each book.
     + category: Genre or classification of the book.
     + availability: Boolean value indicating whether the book is available for borrowing.
3. **Staff Table**
   * **Purpose**: Holds detailed records of the library staff.
   * **Fields**:
     + staff\_id: Unique identifier for each staff member.
     + name: Full name of the staff member.
     + role: Designated position within the library.
     + contact\_info: Phone number or email address for communication.

**Implementation Steps**

**1. Setting Up the Database**

* **Database Creation**: A MySQL database named LibraryDB was created to house the necessary tables.
* **Table Configuration**: The tables were structured with appropriate fields, primary keys, and relationships to maintain data integrity and facilitate easy data retrieval.

sql

CREATE DATABASE LibraryDB;

CREATE TABLE Admin (

admin\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

ISBN VARCHAR(20) UNIQUE NOT NULL,

category VARCHAR(100) NOT NULL,

availability BOOLEAN DEFAULT TRUE

);

CREATE TABLE Staff (

staff\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

role VARCHAR(50) NOT NULL,

contact\_info VARCHAR(100) NOT NULL

);

**2. Developing the Front End**

* **User Interface Design**: The user interface was designed using Java Swing components such as JFrame, JPanel, JTable, and JButton. The design emphasizes usability and accessibility.
* **Event Handling Implementation**: Event listeners were implemented to handle user interactions, providing instant feedback and improving user experience.

**3. Integrating Front End with Back End**

* **Database Connectivity**: JDBC (Java Database Connectivity) was employed to connect the Java application to the MySQL database.
* **Executing SQL Queries**: Prepared statements were used to execute SQL queries, enhancing security and preventing SQL injection attacks.

java

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/LibraryDB", "username", "password");

PreparedStatement pstmt = connection.prepareStatement("SELECT \* FROM Books WHERE title = ?");

pstmt.setString(1, searchTitle);

ResultSet rs = pstmt.executeQuery();

**4. Testing the Application**

* **Unit Testing**: Each function was tested individually to ensure correct functionality and performance.
* **Integration Testing**: The complete system was tested to ensure seamless interaction between the front end and back end.
* **Data Integrity Checks**: Validation checks were performed to ensure that the database maintained accurate and reliable data.

**5. Deployment**

* **Packaging**: The application was packaged as a JAR file for easy distribution and installation on different systems.
* **User Manual Creation**: A comprehensive user manual was developed to assist users in navigating the system and utilizing its features effectively.

**Areas for Improvement**

* **Borrowing Tracking**: Users suggested implementing a feature to track borrowed books and their due dates.
* **Enhanced Search Options**: Feedback indicated a desire for more advanced search capabilities, such as filtering by multiple criteria (e.g., genre, publication date).
* **Reporting Features**: Users expressed interest in generating reports on book circulation, popular titles, and staff performance metrics.

**Conclusion**

The Library Management System successfully fulfills the essential needs of library administration by providing a robust, user-friendly interface and efficient functionalities for managing library resources. The combination of Java Swing and MySQL ensures reliable performance and data integrity.

**Future Work**

1. **User Accounts for Patrons**: Extend the system to allow patrons to create accounts for managing their borrowed items and preferences.
2. **Borrowing and Returning Module**: Implement a module for tracking borrowed books, including due date notifications and fines for late returns.
3. **Advanced Reporting Features**: Develop a reporting module that generates insights into usage statistics, helping administrators understand trends in book borrowing and staff efficiency.
4. **Mobile Access**: Explore the possibility of developing a mobile version of the LMS to provide users with access on-the-go.
5. **Integration with Library Services**: Investigate integration with external library services, such as inter-library loans and online catalogs, to broaden resource availability.

This project illustrates the effective application of Java and MySQL in creating a functional and scalable Library Management System, laying a strong foundation for future enhancements and adaptations