



**M.Kumarasamy**  
**College of Engineering**

**NAAC Accredited Autonomous Institution**

Approved by AICTE & Affiliated to Anna University

ISO 9001:2015 Certified Institution

Thalavapalayam, Karur - 639 113, TAMILNADU.



A Project Report

on

## **LIBRARY MANAGEMENT SYSTEM**

Submitted in partial fulfillment of requirements for the award of the course

of

### **EGB1221-DATABASE MANAGEMENT SYSTEMS**

Under the guidance of

**Dr. T.L. KAYATHRI**

**Assistant Professor / ECE**

Submitted By

**ELAKKIYA K**

**(927623BEC052)**

**ELAMATHI S**

**(927623BEC053)**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION  
ENGINEERING**

**M.KUMARASAMY COLLEGE OF ENGINEERING**  
(Autonomous)

**KARUR – 639 113**

**MAY 2025**

**M. KUMARASAMY COLLEGE OF ENGINEERING**

**(Autonomous Institution affiliated to Anna University, Chennai)**

**KARUR – 639 113**

**BONAFIDE CERTIFICATE**

This is to certify that this project report on “**LIBRARY MANAGEMENT SYSTEM**” is the bonafide work of **ELAKKIYA K (927623BEC052)** **ELAMATHI S (927623BEC053)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature

**Dr. T.L. KAYATHRI M.E.,Ph.D.,**

**SUPERVISOR,**

Department of Electronics and Communication  
Engineering,

M.Kumarasamy College of Engineering,

Thalavapalayam, Karur -639 113.

Signature

**Dr. A. KAVITHA M.E.,Ph.D.,**

**HEAD OF THE DEPARTMENT,**

Department of Electronics and Communication  
Engineering,

M.Kumarasamy College of Engineering,

Thalavapalayam, Karur -639 113.

## **DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

### **VISION OF THE INSTITUTION**

To emerge as a leader among the top institutions in the field of technical education

### **MISSION OF THE INSTITUTION**

- Produce smart technocrats with empirical knowledge who can surmount the global challenges
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
- Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

## **DEPARTMENT VISION, MISSION, PEO, PO AND PSO**

### **Vision**

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

### **Mission**

**M1:** Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.

**M2:** Inculcate the students in problem solving and lifelong learning ability.

**M3:** Provide entrepreneurial skills and leadership qualities.

**M4:** Render the technical knowledge and skills of faculty members.

### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

- **PEO1:** Graduates will have successful Career in software industries and R&D divisions through continuous learning
- **PEO2:** Graduates will provide effective solutions for real world problems in the key domain of computer science and engineering and engage in lifelong learning.
- **PEO3:** Graduates will excel in their profession by being ethically and socially responsible

### **PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

#### **PROGRAM SPECIFIC OUTCOMES (PSOs)**

- **PSO1: Professional Skills:** Ability to apply the knowledge of computing techniques to design and develop computerized solutions for the problems.
- **PSO2: Successful career:** Ability to utilize the computing skills and ethical values in creating a successful career.

## ABSTRACT

Library Management System (LMS) is a software application designed to automate and simplify the process of managing a library's day-to-day operations. This project aims to develop a user-friendly, efficient, and secure system that handles various tasks such as issuing and returning books, managing book inventories, maintaining records of registered users (students/staff), and tracking overdue books.

Traditional library systems are often paper-based, time-consuming, and prone to human error. With the advancement of technology, the need for a digital solution has become essential to streamline the process and reduce the workload on librarians. The proposed Library Management System solves these challenges by providing a structured and systematic way to manage books and user records.

The system is developed using a DBMS (Database Management System) as the backend to store all book and user data in an organized manner, ensuring data integrity, security, and easy retrieval. The frontend offers a clean interface that allows users to search for books, check availability, and view issue history. Librarians can add or delete books, register new users, generate reports, and track fines for overdue returns.

The LMS enhances operational efficiency by:

- Minimizing manual work and paperwork
- Reducing errors in book issuing/returning processes
- Allowing quick access to user and book information
- Supporting accurate record-keeping and report generation

This system benefits schools, colleges, and public libraries, with future enhancements like barcode scanning, online reservations, and SMS/email notifications to improve user experience.

## ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	COs MAPPED	POs MAPPED	PSOs MAPPED
The Library Management System is a software application designed to simplify and automate library operations such as issuing and returning books, managing book and user records, and tracking overdue items. It replaces manual processes with a digital system, improving accuracy and saving time. The system uses a database to store information securely, allowing librarians to manage books and users efficiently, while users can search for books and view their borrowing history. Suitable for schools, colleges, and public libraries, the system reduces paperwork and enhances productivity. Future upgrades may include barcode scanning, online reservations, and notification features.	CO1	PO1	PSO1
	CO2	PO2	PSO2
	CO3	PO3	
	CO4	PO5	
	CO5	PO6	
		PO7	
		PO8	
		PO9	
		PO10	
		PO11	
		PO12	

Note: 1- Low, 2-Medium, 3- HighPO2

**SUPERVISOR**

**HEAD OF THE DEPARTMENT**

## TABLE OF CONTENTS

<b>CHAPTER No.</b>	<b>TITLE</b>	<b>PAGE No.</b>
	<b>ABSTRACT</b>	<b>vii</b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 DATABASE MANAGEMENT SYSTEMS' CONCEPTS	2
<b>2</b>	<b>PROJECT METHODOLOGY</b>	<b>4</b>
	2.1 PROPOSED WORK	4
	2.2 ARCHITECTURE	5
	2.3 E-R DIAGRAM	5
<b>3</b>	<b>MODULE DESCRIPTION</b>	<b>6</b>
	3.1 USER LOGIN MODULE	6
	3.2 CHOOSE MODULE	6
	3.3 PERFORM ACTION MODULE	6
	3.4 DATABASE UPDATE MODULE	7
	3.5 LOGOUT MODULE	7
<b>4</b>	<b>RESULTS AND DISCUSSION</b>	<b>8</b>
<b>5</b>	<b>CONCLUSION</b>	<b>13</b>
	<b>REFERENCES</b>	<b>14</b>
	<b>APPENDIX</b>	<b>15</b>

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OBJECTIVE**

The objective of the Library Management System is to automate and streamline library operations such as book issuing, returning, and record management. The system aims to reduce manual effort, minimize errors, and ensure efficient handling of library resources through a secure and user-friendly interface. It allows librarians to manage books and users effectively while enabling quick access to information. The project promotes digital record-keeping, supports sustainable practices, and lays the foundation for future enhancements like online access and barcode integration.

### **1.2 OVERVIEW**

The Library Management System is a software application designed to automate library operations such as issuing, returning, and tracking books. It replaces manual record-keeping with a digital system, ensuring accuracy and saving time. The system securely stores book and user data in a database, allowing easy access and efficient management. With a user-friendly interface, it helps librarians handle resources effectively and provides users with quick book searches and borrowing history. Suitable for educational and public libraries, the system supports paperless operations and can be enhanced with features like barcode scanning and online reservations.

## 1.3 DATABASE MANAGEMENT SYSTEMS' CONCEPTS

### 1. Entity-Relationship Model (ER Model)

- Models real-world entities: Books, Students, Staff, Transactions
- Relationships: Issue, Return, Manages
- Designed using ER Diagrams with symbols, keys, and cardinality

### 2. Relational Database Design

- Data stored in relational tables:

Example Tables: Books, Students, IssueRecords

- Keys:
  1. Primary Key: Unique identifier (e.g., BookID)
  2. Foreign Key: References other tables (e.g., MemberID in IssueRecords)

### 3. Normalization

- Organizes data to reduce redundancy and dependency
- Forms used:
  1. 1NF: Atomic values
  2. 2NF: Full functional dependency
  3. 3NF: No transitive dependency
  4. BCNF: Stronger form of 3NF

### 4. SQL – Data Definition Language (DDL)

- Used to define and manage table structure
- Commands:
  - CREATE, ALTER, DROP, UPDATE, RENAME

### 5. SQL – Data Manipulation and Queries

- DML operations for handling records:
  - SELECT, INSERT, UPDATE, DELETE
- Use of JOINS to combine data from multiple tables

## 6. Transaction Management

- Ensures ACID properties:

1. Atomicity: All or nothing
2. Consistency: Valid data only
3. Isolation: Transactions don't interfere
4. Durability: Data is saved permanently

## 7. Concurrency Control and Deadlock

- Prevents conflicts in multi-user access

- Uses:
  1. Locking protocols (e.g., two-phase locking)
  2. Deadlock handling (prevention, detection, recovery)

## 8. Integrity Constraints and Views

- Constraints maintain data accuracy:
  - NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK
- Views provide custom data displays for students, staff, and admin

## **CHAPTER 2**

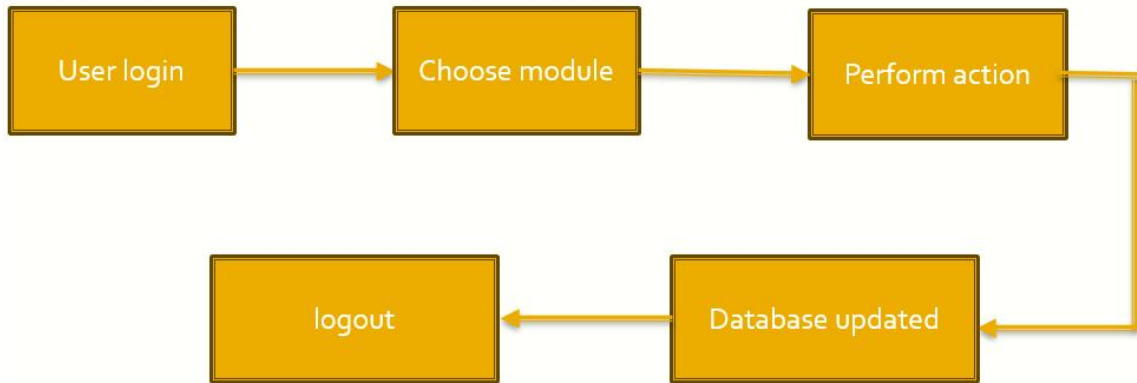
### **PROJECT METHODOLOGY**

#### **2.1.PROPOSED WORK**

The proposed work focuses on developing an efficient and automated Library Management System (LMS) to streamline the key functions of a library, such as book issuance, returns, and user management. The system will utilize a relational database to securely store and manage all essential data, including book details, user information, and transaction records. This will ensure accurate and quick access to information, reducing the reliance on manual processes and minimizing the possibility of errors. The LMS will be designed with a user-friendly interface to cater to both librarians and users. Librarians will have the ability to manage books, track overdue items, register users, and generate reports, while users will be able to search for books, check availability, and view their borrowing history. The system will enable seamless integration between these functionalities to improve operational efficiency and user experience.

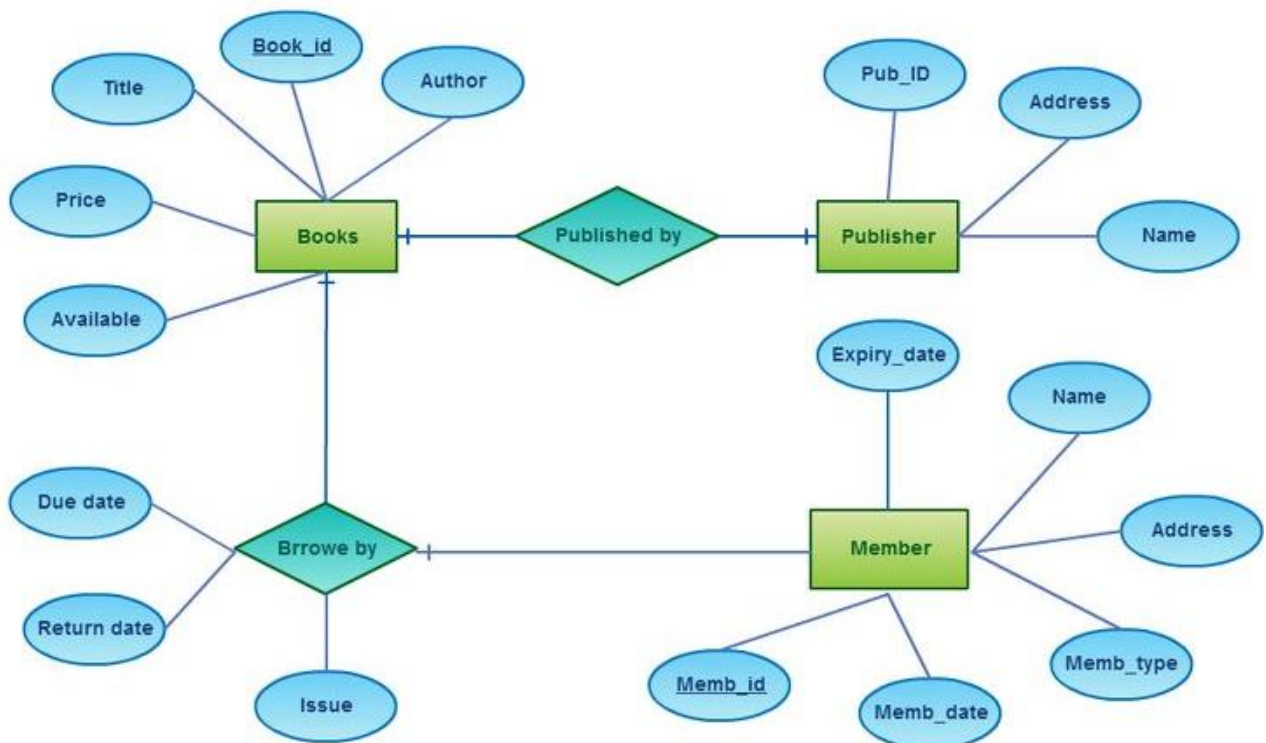
Future enhancements will focus on integrating additional features such as barcode scanning for quicker book identification, online reservations for users to reserve books remotely, and SMS/email notifications to inform users about overdue books or updates to their reservations. These improvements will further enhance the system's usability and functionality, making it a modern and reliable solution for libraries of all sizes, especially in educational and public library settings. The ultimate goal of this project is to improve library management, reduce administrative workload, and promote paperless, digital operations.

## 2.2 ARCHITECTURE



**Fig.2.1.Architecture**

## 2.3 E-R DIAGRAM



**Fig.2.2.ER Diagram**

## **CHAPTER 3**

### **MODULE DESCRIPTION**

#### **3.1 User Login Module**

- Purpose: To authenticate the user (Admin, Librarian, or Student).
- Functionality:
  - Accepts username and password.
  - Verifies credentials with the database.
  - Grants access based on user role.

#### **3.2 Choose Module**

- Purpose: To let users navigate to specific sections based on their role.
- Options:
  - Member: Search books, view issued books, due dates.
  - Admin: Issue/return books, manage inventory, add/delete users or books.

#### **3.3 Perform Action**

- Purpose: Executes the selected operation.
- Examples:
  - Issuing or returning books.
  - Adding new books or students.
  - Searching book records.

### 3.4 Database Updated

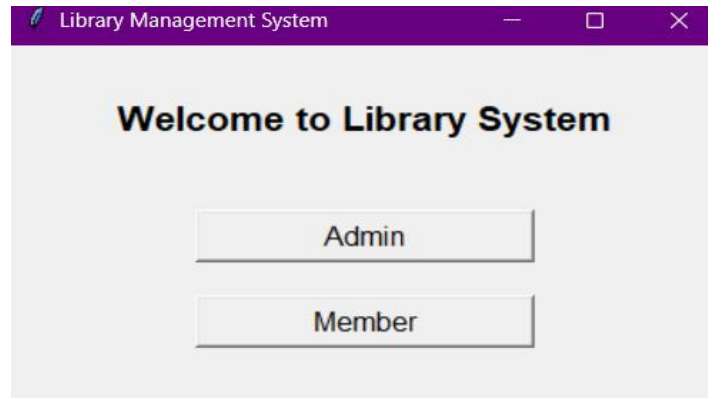
- Purpose: Automatically updates the database with performed actions.
- Details:
  - Updates IssueRecords, Books, or User tables.
  - Ensures data integrity with constraints (like stock decrement or return status).

### 3.5 Logout

- Purpose: Ends the user session securely.
- Functionality:
  - Closes the session.
  - Redirects to the login screen.
  - Ensures no action is performed without re-authentication.

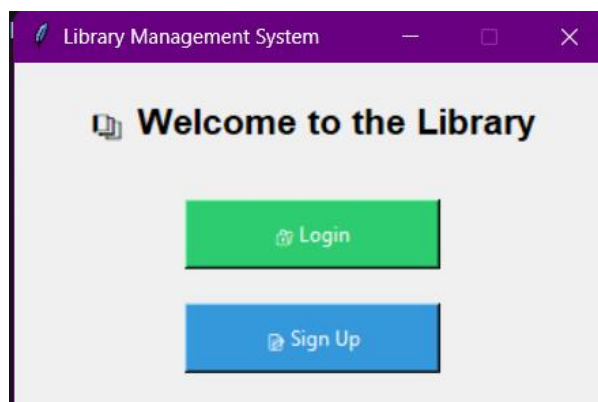
## CHAPTER 4

### RESULTS AND DISCUSSION

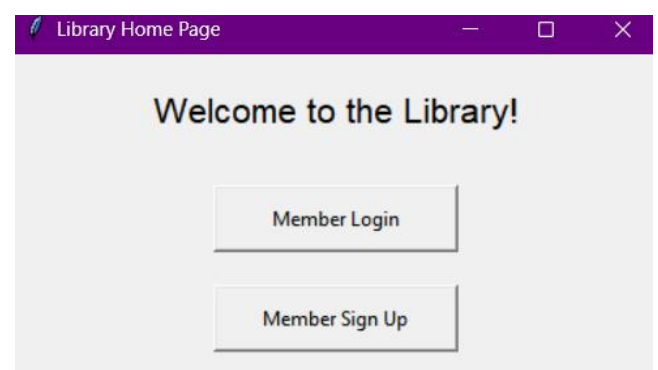


**Fig.4.1. Welcome Page**

The welcome page (**Fig.4.1**) of the Library Management System provides two options: **Admin** and **Member**. Users select their role to proceed to their respective login pages. This interface ensures role-based access and streamlined user experience.

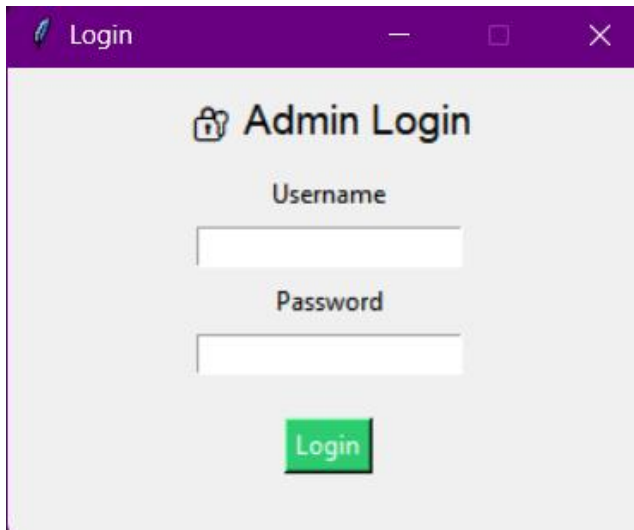


**Fig.4.2. Admin Signup/Login Page**

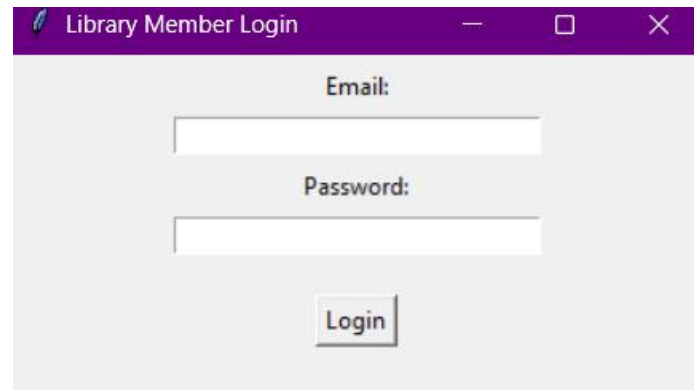


**Fig.4.3. Member Signup/Login Page**

The login/signup page allows users to either log in with existing credentials or create a new account. Admins and members can securely access the system, with separate paths for authentication and registration to ensure authorized usage and easy onboarding.

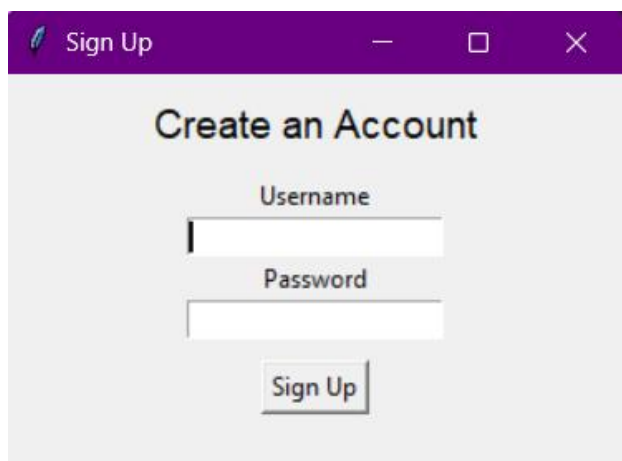
A screenshot of a web browser window titled "Login". The page has a light gray background and a purple header bar. The main heading is "Admin Login" with a lock icon. Below it are two input fields: "Username" and "Password". At the bottom is a green "Login" button.

**Fig.4.4. Admin Login Page**

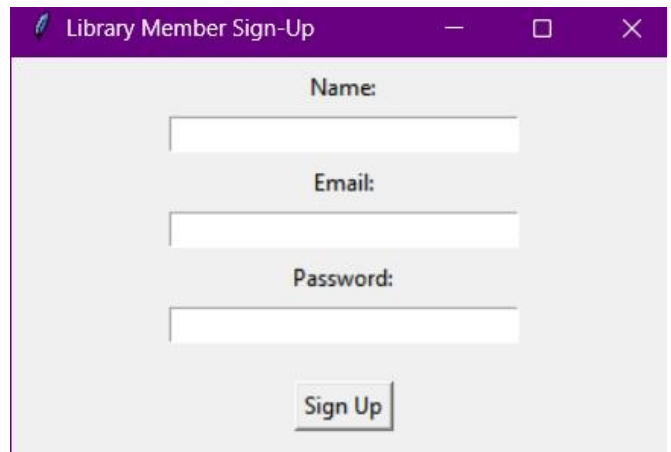
A screenshot of a web browser window titled "Library Member Login". The page has a light gray background and a purple header bar. The main heading is "Library Member Login". Below it are two input fields: "Email:" and "Password:". At the bottom is a "Login" button.

**Fig.4.5.Member Login Page**

The separate login page for admin and members provides secure access to the Library Management System for users like admins and members. It verifies credentials and prevents unauthorized access through a simple, user-friendly interface.

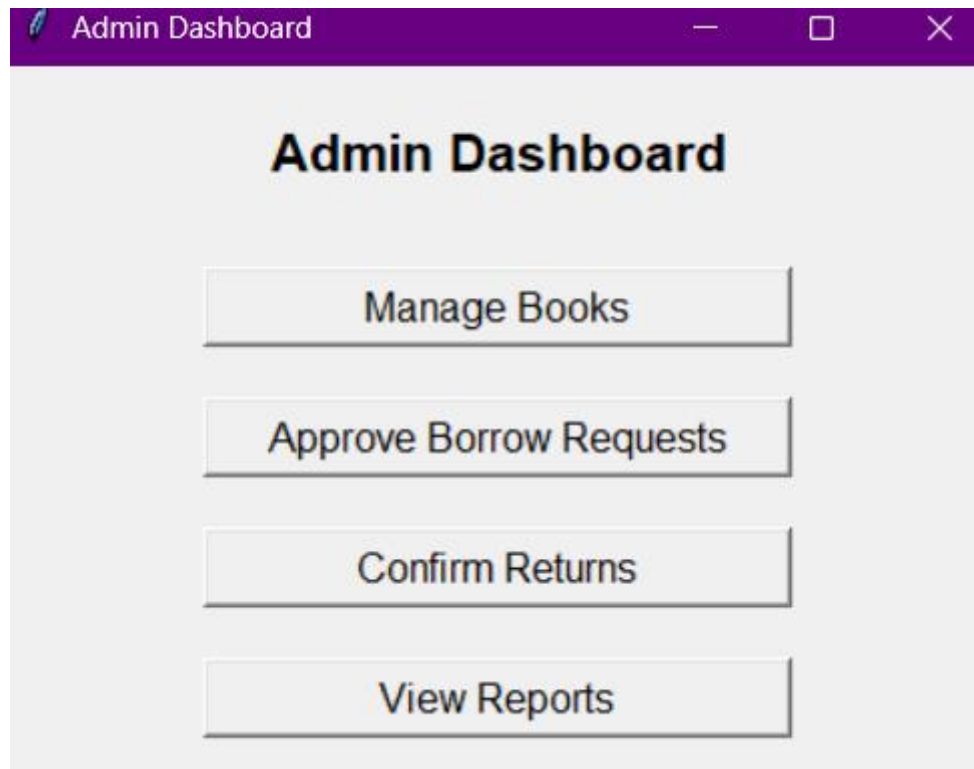
A screenshot of a web browser window titled "Sign Up". The page has a light gray background and a purple header bar. The main heading is "Create an Account". Below it are two input fields: "Username" and "Password". At the bottom is a "Sign Up" button.

**Fig.4.6.Admin Signup Page**

A screenshot of a web browser window titled "Library Member Sign-Up". The page has a light gray background and a purple header bar. The main heading is "Library Member Sign-Up". Below it are three input fields: "Name:", "Email:", and "Password:". At the bottom is a "Sign Up" button.

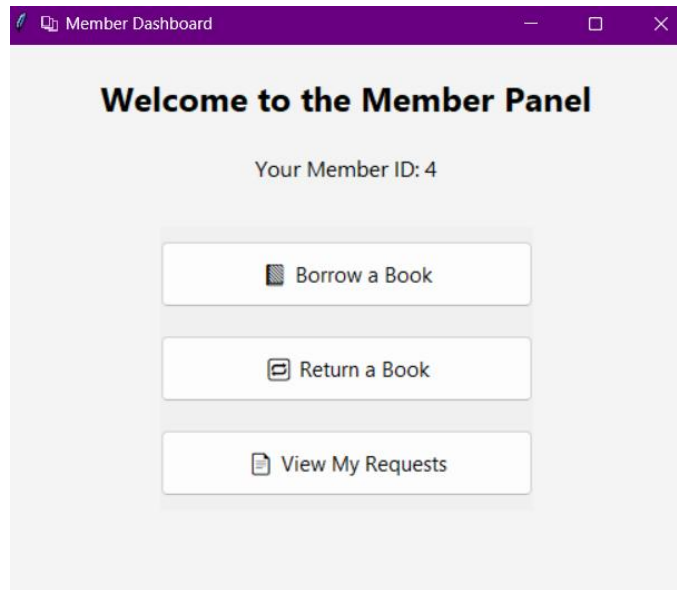
**Fig.4.7.Member Signup Page**

The sign up page allows new users to register for the Library Management System by providing details such as name, email, user ID, and password. It ensures secure account creation and access based on user roles like Admin or Member.



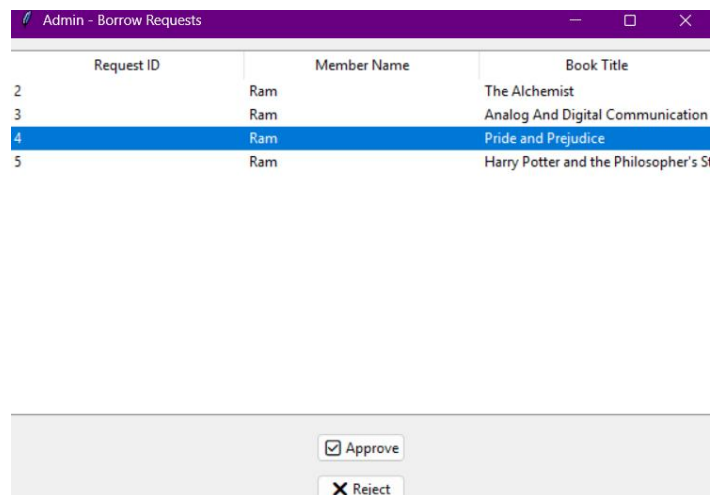
**Fig.4.8 Admin Dashboard**

The admin dashboard provides full control over the Library Management System. Admins can manage books, view and update member records, issue or return books, and monitor system activity through an organized and user-friendly interface.



**Fig.4.9 Member Panel**

The member panel welcomes users with their Member ID and provides three main options: **Borrow a Book**, **Return a Book**, and **View My Requests**. It offers a simple and interactive interface for managing book transactions efficiently within the Library Management System.



**Fig.4.10 Borrow Request Approval Panel**



Library Management System

### Library Management System

Title:

Author:

Year:

Genre:

ID	Title	Author	Year	Genre
1	Harry Potter and t	J.K. Rowling	1997	Fantasy
2	The Hobbit	J.R.R. Tolkien	1937	Fantasy
3	Analog And Digit	T.L.Singal	2025	Electronics and co
4	To Kill a Mocking	Harper Lee	1960	Fiction / Historic
5	Sapiens: A Brief H	Yuval Noah Harari	2011	Non-fiction / His
6	1984	George Orwell	1949	Dystopian / Scien
7	The Alchemist	Paulo Coelho	1988	Fiction / Fantasy ,
8	Atomic Habits	James Clear	2018	Self-help / Psych
9	The Hobbit	J.R.R. Tolkien	1937	Fantasy / Advent
10	Pride and Prejudi	Jane Austen	1813	Romance

**Fig.4.11.Book Management Panel**

## **CHAPTER 5**

### **CONCLUSION**

The Library Management System (LMS) is a robust and efficient solution designed to automate and streamline the daily operations of a library. By integrating core database management concepts such as ER modeling, normalization, SQL operations, and transaction control, the system ensures accuracy, consistency, and data integrity.

This project allows users—students, librarians, and administrators—to seamlessly interact with the system through modules such as login, book search, issue/return transactions, and logout. Every user action triggers appropriate database updates, maintaining real-time records and enabling smooth library workflows. Additionally, the system handles multi-user access through proper concurrency control, preventing data conflicts and ensuring secure transactions.

In conclusion, the LMS not only simplifies the manual processes of library management but also provides a reliable and scalable platform that supports the academic ecosystem effectively. This project demonstrates the practical application of DBMS concepts in real-life scenarios, enhancing both user experience and operational efficiency.

## REFERENCES:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan,  
Database System Concepts, 7th Edition, McGraw-Hill, 2021.
2. Raghu Ramakrishnan and Johannes Gehrke,  
Database Management Systems, 3rd Edition, McGraw-Hill, 2003.
3. Elmasri, Ramez and Navathe, Shamkant B.,  
Fundamentals of Database Systems, 7th Edition, Pearson Education,  
2016.
4. [www.db-book.com](http://www.db-book.com) –  
Official website for Database System Concepts with slides and  
examples.
5. [www.w3schools.com/sql](http://www.w3schools.com/sql) –  
For SQL command references and practical syntax.
6. <https://www.tutorialspoint.com/dbms/index.htm> –  
A learning platform for DBMS theory and SQL queries.
7. MySQL Documentation,  
<https://dev.mysql.com/doc/> –  
For understanding MySQL syntax and functions used in backend  
operations.

## APPENDIX

### (Coding)

#### **Main\_menu.py**

```
import tkinter as tk
import os
from tkinter import messagebox

def open_admin_home():
    try:
        os.system('python home.py')
    except Exception as e:
        messagebox.showerror("Error", f'Could not open Admin Panel: {e}')

def open_member_home():
    try:
        os.system('python member_home.py')
    except Exception as e:
        messagebox.showerror("Error", f'Could not open Member Panel: {e}')

# GUI Setup
window = tk.Tk()
window.title("Library Management System")
window.geometry("400x300")

tk.Label(window, text="Welcome to Library System", font=("Arial", 16,
"bold")).pack(pady=30)

tk.Button(window, text="Admin", font=("Arial", 12), width=20,
```

```
command=open_admin_home).pack(pady=10)

tk.Button(window,      text="Member",      font=("Arial",      12),      width=20,
command=open_member_home).pack(pady=10)
```

```
window.mainloop()
```

### **Admin\_borrow\_requests.py**

```
import tkinter as tk

from tkinter import ttk, messagebox

import mysql.connector

def load_requests():
    try:
        db = mysql.connector.connect(
            host="localhost",
            user="root",
            password="Elamathi@2005",
            database="library_management"
        )
        cursor = db.cursor()
        cursor.execute("""
            SELECT br.request_id, m.name, b.title
            FROM borrow_requests br
            JOIN members m ON br.member_id = m.member_id
            JOIN books b ON br.book_id = b.book_id
            WHERE br.processed = FALSE
        """)
        rows = cursor.fetchall()
        db.close()
```

```
for row in tree.get_children():
    tree.delete(row)
for row in rows:
    tree.insert("", 'end', values=row)

except Exception as e:
    messagebox.showerror("Error", str(e))

def approve_request():
    selected = tree.selection()
    if not selected:
        messagebox.showwarning("Select", "Please select a request to approve.")
        return

    request_id = tree.item(selected[0])['values'][0]

    try:
        db = mysql.connector.connect(
            host="localhost",
            user="root",
            password="Elamathi@2005",
            database="library_management"
        )
        cursor = db.cursor()

        # Approve request
        cursor.execute("UPDATE borrow_requests SET approved = TRUE, processed
```

```
= TRUE WHERE request_id = %s", (request_id,))
    db.commit()

# Insert into borrowed_books table
cursor.execute("""
    INSERT INTO borrowed_books (member_id, book_id)
    SELECT member_id, book_id FROM borrow_requests WHERE request_id
= %s
    """, (request_id,))
    db.commit()

db.close()
load_requests()
messagebox.showinfo("Approved", "Request approved successfully.")
except Exception as e:
    messagebox.showerror("Error", str(e))

def reject_request():
    selected = tree.selection()
    if not selected:
        messagebox.showwarning("Select", "Please select a request to reject.")
        return

    request_id = tree.item(selected[0])['values'][0]

    try:
        db = mysql.connector.connect(
            host="localhost",
```

```
        user="root",
        password="Elamathi@2005",
        database="library_management"
    )
    cursor = db.cursor()
    cursor.execute("UPDATE borrow_requests SET approved = FALSE,
processed = TRUE WHERE request_id = %s", (request_id,))
    db.commit()
    db.close()

    load_requests()
    messagebox.showinfo("Rejected", "Request rejected.")
except Exception as e:
    messagebox.showerror("Error", str(e))

# GUI
window = tk.Tk()
window.title("Admin - Borrow Requests")
window.geometry("600x400")

tree = ttk.Treeview(window, columns=('ID', 'Member', 'Book'), show='headings')
tree.heading('ID', text='Request ID')
tree.heading('Member', text='Member Name')
tree.heading('Book', text='Book Title')
tree.pack(pady=10, fill=tk.BOTH, expand=True)

tk.Button(window, text="✓ Approve", command=approve_request).pack(pady=5)
tk.Button(window, text="✗ Reject", command=reject_request).pack(pady=5)
```

```
load_requests()
```

```
window.mainloop()
```

**Admin\_dashboard.py:**

```
import tkinter as tk
```

```
from tkinter import ttk
```

```
import mysql.connector
```

```
def fetch_reports():
```

```
    db = mysql.connector.connect(
```

```
        host="localhost",
```

```
        user="root",
```

```
        password="Elamathi@2005",
```

```
        database="library_management"
```

```
    )
```

```
    cursor = db.cursor()
```

```
    cursor.execute("SELECT COUNT(*) FROM books")
```

```
    total_books = cursor.fetchone()[0]
```

```
    cursor.execute("SELECT COUNT(*) FROM members")
```

```
    total_members = cursor.fetchone()[0]
```

```
    cursor.execute("SELECT COUNT(*) FROM borrow_requests")
```

```
    total_requests = cursor.fetchone()[0]
```

```
    cursor.execute("SELECT COUNT(*) FROM borrowed_books WHERE returned  
= TRUE")
```



```
total_returns = cursor.fetchone()[0]
```

```
cursor.execute("""
```

```
    SELECT b.title, COUNT(*) as borrow_count
```

```
    FROM borrowed_books bb
```

```
    JOIN books b ON bb.book_id = b.book_id
```

```
    GROUP BY bb.book_id
```

```
    ORDER BY borrow_count DESC
```

```
    LIMIT 5
```

```
""")
```

```
top_books = cursor.fetchall()
```

```
cursor.execute("""
```

```
    SELECT m.name, COUNT(*) as total
```

```
    FROM borrowed_books bb
```

```
    JOIN members m ON bb.member_id = m.member_id
```

```
    GROUP BY bb.member_id
```

```
    ORDER BY total DESC
```

```
    LIMIT 5
```

```
""")
```

```
top_members = cursor.fetchall()
```

```
db.close()
```

```
    return total_books, total_members, total_requests, total_returns, top_books,  
    top_members
```

```
# GUI
```

```
window = tk.Tk()

window.title("    Admin Reports Panel")

window.geometry("650x600")


total_books, total_members, total_requests, total_returns, top_books, top_members
= fetch_reports()


frame = tk.Frame(window)
frame.pack(pady=20)


tk.Label(frame, text=f"    Total    Books:    {total_books}", font=("Arial",
12)).pack(anchor='w')
tk.Label(frame, text=f"    Total    Members:    {total_members}", font=("Arial",
12)).pack(anchor='w')
tk.Label(frame, text=f"    Borrow    Requests:    {total_requests}", font=("Arial",
12)).pack(anchor='w')
tk.Label(frame, text=f"✓✓    Returned    Books:    {total_returns}", font=("Arial",
12)).pack(anchor='w')


# Top Books

tk.Label(window, text="    Top 5 Most Borrowed Books", font=("Arial", 14,
'bold')).pack(pady=10)

book_tree = ttk.Treeview(window, columns=("Title", "Times"), show="headings")
book_tree.heading("Title", text="Book Title")
book_tree.heading("Times", text="Times Borrowed")
book_tree.pack(pady=5)

for title, count in top_books:
    book_tree.insert("", 'end', values=(title, count))
```

# Top Members

```
tk.Label(window, text="    Top 5 Active Members", font=("Arial", 14,
'bold')).pack(pady=10)
member_tree = ttk.Treeview(window, columns=("Name", "Total"),
show="headings")
member_tree.heading("Name", text="Member Name")
member_tree.heading("Total", text="Books Borrowed")
member_tree.pack(pady=5)
for name, total in top_members:
    member_tree.insert("", 'end', values=(name, total))

window.mainloop()
```