# Software Requirements Specification (SRS) Document

# Library Management System

# 1. Brief Description of Project

The Library Management System is a console-based mini-project developed in C that helps manage a small-scale digital library. It supports adding books, viewing them by category, borrowing, returning with fine calculation, and even opening the associated book files using the system’s default viewer.

## 2. Purpose / Goal

The goal of this system is to simplify the management of books for educational or institutional libraries. It facilitates borrowing, returning, and organizing books by category, ensuring ease of access and reducing manual workload.

## 3. Usefulness / Benefit

For Students/Users: Browse books by category, borrow and return books easily, read digital content.

For Library Admins: Add new books, view current book status, maintain availability status.

For Institutions: Reduce dependency on physical registers, enhance the user experience, and promote digital access to books.

## 4. Hardware / Software Involved

Hardware Requirements:  
- Standard PC/Laptop  
- Minimum 2GB RAM  
- 100MB Disk Space

Software Requirements:  
- Programming Language: C  
- Compiler: GCC   
- Operating System: Windows / Linux / macOS

## 5. Detailed Feature List

General Module:  
- Console-based Menu Navigation  
- File access using system's default viewer

User Features:  
- View books by category (Fiction, Autobiography, Sports)  
- Open books in file viewer

Admin Features:  
- Add new books with category and file path  
- Borrow and return book functionality  
- Fine calculation for late returns  
- Display list of all books and their status

## 6. Test / Demonstration Plan

Unit Testing:  
- Test individual functions like adding, borrowing, returning, and listing books.

Integration Testing:  
- Ensure proper linkage between borrowing and returning (e.g., availability toggle, fine display).

System Testing:  
- Full interaction with all menu options and workflows.

User Acceptance Testing:  
- Run the system with mock users to verify ease of use and correctness of features.

## 7. Expected Interaction Interface and Sample Use Cases

Interaction Interface:  
- Console-based UI  
- Text-based input prompts and categorized output displays

Sample Use Cases:  
  
User Borrows Book:  
- User selects a book ID → Enters today's date → Receives borrowing confirmation and return deadline  
  
Admin Adds Book:  
- Admin enters book title, author, category, and file path → Book is added to the library  
  
User Returns Book:  
- User provides book ID and return date → System checks delay and shows fine if applicable  
  
User Opens Book:  
- User selects book ID → File is opened using default text reader

## Individual Member Contribution:

**PRATEEK YV**: Worked on borrowing and return logic with fine system.

**AKSHIT NAIDU**: Developed category-wise display and book listing.

**VIKAS R**: Implemented file open and preload functionality.

**VISHNU PREYAN**: Designed and implemented the menu system and testing framework.

**TEAM MEMBERS NAME, SRN**

YELLAPANTULA VENKATA PRATEEK, PES2UG24CS617

VIKAS R, PES2UG24CS584

AKSHIT NAIDU, PES2UG24CS614

VISHNU PREYAN, PES2UG24AM228

TITLE: Library Management System

TEAM NO: 4