**Software Requirements Specification (SRS)  
Borrow Book Buddy**

# 1. Introduction

• Target Audience: Teachers, Developers and stakeholders.  
• Project Name: Borrow Book Buddy  
• Author: Suraj Pathak  
• Version: 1.0

# 2.1 Purpose

This document specifies the requirements for the "Borrow Book Buddy" web application, a platform designed to facilitate book sharing within a college community. The goal of this application is to promote resource sharing, reduce costs for students, and encourage collaboration within the college ecosystem.

## 2. Scope

Borrow Book Buddy is a web application that allows college students to:  
  
• List books they are willing to lend.  
• Browse and borrow books from peers within the same community.  
• Manage borrowing and lending transactions through an intuitive interface.  
  
The platform aims to streamline the book borrowing and lending process, ensuring transparency and ease of use for all participants.

# 3. Requirements

## 3.1 Functional Requirements

**3.1.1 User Management:**  
• Users can register using their Gmail account  
• Secure login and logout functionality.  
• Ability to create and manage personal profiles, including:  
 - Name  
 - Contact details (email, optional phone number)  
 - Profile picture (optional)

**3.1.2 Book Management:**  
• Users can add books to their personal collection, specifying details such as:  
 - Title  
 - Author  
 - ISBN  
 - Condition (e.g., "New," "Good," "Worn")  
• Users can edit or remove books from their collection.  
• Books can be marked as "Available" or "Unavailable" for lending.

**3.1.3 Book Discovery:**  
• Users can search for books using:  
 - Title  
 - Author  
 - ISBN  
• Filters can be applied to narrow search results by:  
 - Availability  
 - Condition  
• Users can view detailed information about a book, including its current availability and lender details.

**3.1.4 Borrowing System:**  
• Users can send borrowing requests to book owners.  
• Owners can approve or decline borrowing requests.  
• Once a request is approved:  
 - Basic contact information is shared between the lender and borrower for transaction coordination.  
 - The book is marked as "Reserved."  
 - Owners can mark the book as "Available" again once the transaction is completed.

## 3.2 Non-Functional Requirements

• Performance:  
 - The application should ensure page load times under 3 seconds for most user actions.  
 - The platform must support 100-200 concurrent users without significant performance degradation.  
• Security:  
 - User data in transit should be encrypted using HTTP/HTTPS.   
 - Input validation mechanisms must be implemented to prevent SQL injection, XSS, and other vulnerabilities.  
• Usability:  
 - The application should feature a responsive design for compatibility across desktops, tablets, and mobile devices.  
 - The user interface should be intuitive and easy to navigate.  
 - Accessibility standards (e.g., WCAG) should be adhered to, ensuring usability for all users, including those with disabilities.

# 4. Design Constraints

• The system should be compatible with modern web browsers.  
• Development should use the familiar tech stack to streamline backend and frontend operations.

# 5. Feasibility Study

• Technical Feasibility: The suggested tech stack ensures ease of implementation. Resources for development, testing, and deployment are readily available.  
• Economic Feasibility: Development costs are minimal due to the use of open-source technologies. The project aligns with institutional goals of cost reduction for students.  
• Risks and Mitigation:  
 - Risk: Low adoption by users.  
 Mitigation: Promotion on all college events   
 - Risk: Initial bugs during deployment.  
 Mitigation: Conduct thorough testing before launch.

# 6. Appendices

**6.1 Glossary of Terms:**  
• ISBN: International Standard Book Number, a unique identifier for books.   
• WCAG: Web Content Accessibility Guidelines, a set of guidelines for making web content more accessible.

**6.2 Database Schema:**  
• Users Table:  
 - user\_id (Primary Key)  
 - name  
 - email  
 - profile\_picture  
• Books Table:  
 - book\_id (Primary Key)  
 - user\_id (Foreign Key)  
 - title  
 - author  
 - isbn  
 - condition  
 - status  
• Borrow Requests Table:  
 - request\_id (Primary Key)  
 - book\_id (Foreign Key)  
 - borrower\_id (Foreign Key)  
 - status (Pending, Approved, Declined)