LIBRAREASE

Mini Project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Dr K. Kamakshaiah
Assosiate Professor

By

B VARSHITH REDDY	**(21R11A05B0)**
KADA LEKHYA MANASWI	**(21R11A05C6)**
NEHA MUNUGANTI	**(21R11A05D7)**

Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

August-2024

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Accredited by NBA

CERTIFICATE

This is to certify that the B.Tech Mini Project report entitled **"LIBRAREASE"** is a bonafide work done by **Baddam Varshith Reddy(21R11A05B0), Kada Lekhya Manaswi(21R11A05C6), Neha Munuganti(21R11A05D7)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2023-2024.

Internal Guide	**HOD – CSE**
Dr K. Kamakshaiah	**Dr A SreeLakshmi**
Assosiate Professor	Professor

External Examiner

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA

DECLARATION BY THE CANDIDATE

We, **Baddam Varshith Reddy, Kada Lekhya Manaswi, Neha Munuganti**, bearing Roll Nos. **21R11A05B0, 21R11A05C6, 21R11A05D7**, hereby declare that the project report entitled **"LIBRAREASE"** is done under the guidance of **Mr. Dr K. Kamakshaiah**, **Assosiate Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by me/us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Baddam Varshith Reddy(21R11A05B0)
Kada Lekhya Manaswi(21R11A05C6)
Neha Munuganti(21R11A05D7)

Department of CSE,
Geethanjali College of Engineering and Technology,
Cheeryal.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our **Secretary Mr.G.R. Ravinder Reddy** for providing the necessary infrastructure to complete our project. We are also thankful to our **Principal Dr. S. Udaya Kumar** for providing an interdisciplinary & progressive environment.

We would like to express our sincere thanks to **Dr. A. Sree Lakshmi**, **Professor, Head of Department** of Computer Science, Geethanjali College of Engineering and Technology, Cheeryal, whose motivation in the field of software development has made us to overcome all hardships during the course of study and successful completion of project.

We would like to express our deep-felt gratitude and sincere thanks to our guide **Dr. K. Kamakshaiah, Assosiate Professor**, Department of Computer Science, Geethanjali College of Engineering and Technology, Cheeryal, for his skillful guidance, timely suggestions and encouragement in completing this project sucessfully. We would like to express our profound sense of gratitude to all for having helped us in completing this dissertation.

Finally, we would like to express our heartfelt thanks to our parents who were very supportive both financially and mentally and for their encouragement to achieve our set goals.

Baddam Varshith Reddy(21R11A05B0)
Kada Lekhya Manaswi(21R11A05C6)
Neha Munuganti(21R11A05D7)

ABSTRACT

Librarease is a user-friendly web-based library management system designed to simplify book management for both students and librarians. Students can effortlessly search for books, borrow them, and view availability status, while librarians can efficiently manage inventory, update book details, and track borrower records. In this project, a comprehensive approach is undertaken to address the challenges of traditional library management systems, where students often encounter difficulties in locating and accessing relevant books. Through the utilization of modern web development technologies, database management, and user interface design principles, the proposed system aims to streamline the book searching process while providing accurate information on book availability. The methodology involves the design and implementation of a user interface that is intuitive and responsive, allowing students to seamlessly navigate and search the library's inventory. The system integrates with a database that maintains up-to-date records of book quantities and availability statuses and also the borrowed list of students. Now, students can borrow books online and pick them up at the library later.

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.no	Acronym	Abbreviation
1	XAMPP	Cross platform, Apache,MySQL,PHP,Pearl
2	PHP	Hypertext Preprocessor
3	HTML	Hypertext Markup Language
4	CSS	Cascading Style Sheet
5	JS	JavaScript
6	UML	Unified Modeling Language

TABLE OF CONTENTS

1. INTRODUCTION

1.1 ABOUT THE PROJECT

Libraries are evolving to meet the needs of today's society. They've been around for a while and help people learn and share information. However, with the advent of computers and the internet, libraries are changing even more rapidly.

The project we're discussing is a technology-based solution for improving libraries. Consider a library where you can use a computer to quickly locate the book you want. This project is similar to that. It is about creating a computer system that allows students to easily find books in the library. Not only that, but it also informs them whether the book they want is currently available and allows them to reserve it in their name.

The project's goal is to create a simple and user-friendly system. When students use it, they can log in, search for the books they need, and borrow them online. The system will inform them whether the books are present or not. This makes it much easier for students to locate and use the books they require.

This is significant because in today's world, we are accustomed to using technology to complete tasks quickly. This project applies that concept to libraries. It's like having a knowledgeable and helpful friend in the library to help students.

The project's goal is to improve the library experience for students.It uses technology to make it easier for students to find and borrow books, and the student does not have to worry about the submission deadline because he will be notified of the return date. This is an excellent example of how traditional things, such as libraries, can be improved by incorporating new ideas and technology.

1.2 OBJECTIVES

- **User-friendly system:** Develop a digital library management system that simplifies operations.
- **Dynamic login interface:** Create a login system that allows students to access their accounts easily.
- **Efficient book search:** Enable students to search for books quickly using the system.
- **Online borrowing:** Allow students to borrow books online for easier resource access.
- **Real-time availability:** Provide real-time updates on book availability in the library.
- **Streamlined access:** Make the process of finding and accessing library resources smoother.
- **Personalized profiles:** Use student profiles to personalize their library experience with tailored book suggestions.
- **Automated book tracking:** Reduce manual effort by automating book tracking and management.
- **Enhanced library experience:** Improve the overall experience for students through digital solutions.
- **Innovative services:** Contribute to the enhancement of library services through technology-driven solutions.
- **Bridge tradition and technology:** Modernize traditional library practices using contemporary tools and technology.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In traditional libraries, the management of books and records is often done manually, with librarians maintaining physical registers or simple digital spreadsheets to log book issues, returns, and inventory. Users often search for books by browsing shelves or consulting the librarian directly. The manual approach to tracking transactions, maintaining book stock, and managing user data is time-consuming and prone to errors. Communication between users and library staff, like book suggestions or reservation requests, also relies heavily on face-to-face interactions or basic email systems.

Drawbacks of existing system:

- **Manual record-keeping:** Library transactions such as book issues, returns, and inventory updates are often managed manually, leading to inefficiencies and increased chances of mistakes, such as data duplication or loss.

- **Limited searchability:** Users typically search for books by physically browsing through shelves or consulting librarians. This process is time-consuming and inefficient, especially when there is no proper cataloging system.

- **Error-prone transactions:** The manual nature of operations can lead to errors in tracking book availability, due dates, and return status, which can result in misplaced books or incorrect fines imposed on users.

- **Inefficient communication:** Users must physically visit the library or communicate via email to inquire about book availability, make reservations, or submit suggestions. This method lacks convenience and real-time updates.

2.2 PROPOSED SYSTEM

The LIBRAREASE system is a web-based library management platform designed to streamline library operations. It provides a user-friendly interface for both students and administrators, allowing users to easily search for books, manage borrowings, and track inventory in real-time. The system also enables library staff to update records, add new books, and respond to user suggestions. With a focus on automation and efficiency, LIBRAREASE ensures that all library transactions are seamlessly recorded and managed.

Benefits:

- **Automated transactions:** The system automates book borrowing, returning, and record management, reducing the chances of human error and improving overall efficiency.
- **Enhanced search functionality:** Users can search for books by title, author, or category through an intuitive search interface, making it easier to locate available resources quickly.
- **Real-time updates:** The system provides real-time information on book availability, due dates, and user accounts, ensuring that both users and administrators are always up to date.
- **Improved communication:** Users can easily submit suggestions or queries online, eliminating the need for physical visits and improving communication between users and library staff.
- **Detailed record management:** Administrators can track book inventory, borrower details, and suggestion box inputs, ensuring accurate record-keeping and better library management.
- **Reduced workload for staff:** By automating repetitive tasks, the system frees up staff time, allowing them to focus on more critical library operations.

2.3 FEASIBILITY STUDY

2.3.1 Details

The feasibility study for LIBRAREASE examines whether the system can be developed and implemented effectively. It considers technical feasibility (availability of technology and expertise), operational feasibility (how well the system meets the needs of students and librarians), and economic feasibility (cost-effectiveness and return on investment). The study ensures that the system can be developed within the available resources and time frame.

2.3.2 Impact on Environment

The environmental impact of LIBRAREASE is minimal, as it reduces the need for paper-based records and physical cataloging. By digitizing library management, the system decreases paper waste and lowers the carbon footprint associated with traditional library operations. However, the study also considers the energy consumption of the system's servers and the environmental impact of electronic waste.

2.3.3 Safety

LIBRAREASE is designed with safety in mind, ensuring that users' data is protected through robust security measures. The system includes encryption for sensitive information, secure login procedures, and regular security updates to prevent unauthorized access and data breaches. Safety protocols ensure that the system operates without risking users' personal information.

2.3.4 Ethics

The ethical implications of LIBRAREASE focus on user privacy and data security. The system adheres to ethical standards by protecting user data, ensuring transparency in data handling, and providing users with control over their personal information. It avoids practices that could lead to misuse of data or unfair treatment of any users.

2.3.5 Cost

The cost analysis for LIBRAREASE includes the initial development expenses, ongoing maintenance, and potential upgrades. The system is designed to be cost-effective by

minimizing operational costs and reducing the need for physical resources like paper. The feasibility study ensures that the benefits of the system justify the investment.

2.3.6 Type

LIBRAREASE is a web-based library management system, categorized as an information management solution. It integrates various functionalities such as book search, availability checks, borrowing requests, and administrative management within a single platform. The system is designed to be scalable and adaptable to different library sizes.

2.4 SCOPE OF THE PROJECT

The scope of the LIBRAREASE project includes the development, implementation, and maintenance of the web-based library management system. The system will support core functions such as book searches, availability checks, borrowing requests, and administrative tasks like adding and managing books and borrower records. The project will cover all aspects of system design, from the user interface to the backend database, ensuring seamless integration and user satisfaction.

2.5 SYSTEM CONFIGURATION

- **Database Server:**MySQL (provided by XAMPP)
- **Programming Languages:**PHP (version 7.4 or later)
- **Frontend Technologies:**
    - HTML5
    - CSS3 (for styling)
    - JavaScript (for client-side scripting)
- **Development Environment:**
    - XAMPP (for local development with integrated Apache, MySQL, and PHP)
    - Visual Studio Code (as the code editor)

3. LITERATURE OVERVIEW

3.1 LITERATURE REVIEW

1. Manual Library System

- **Description**: Traditional method where all library records were maintained on paper.
- **Disadvantages**:
    - Time-consuming and prone to human error.
    - Difficult to track book loans and inventory.
    - Inefficient search and retrieval of records.

2. Standalone Digital Library System

- **Description**: Early software systems used on a single computer within the library.
- **Disadvantages**:
    - Limited access—only available on one computer.
    - No online access for users.
    - Difficult to scale or share data across multiple libraries.

3. Client-Server Library System

- **Description**: Library management software used in a networked environment with a central server.
- **Disadvantages**:
    - High setup and maintenance costs.
    - Requires skilled IT staff for server management.
    - Limited remote access for users.

4. Web-Based Library System

- **Description**: Web-based software accessible from any internet-enabled device, allowing real-time access to library data.
- **Disadvantages**:
    - Dependent on internet connectivity.
    - Data security concerns due to online access.
    - Subscription or hosting costs can be high for small libraries.

5. Cloud-Based Library System

- **Description**: The most modern system where all data is stored in the cloud, allowing remote access and real-time synchronization across multiple locations.
- **Disadvantages**:
    - Dependent on stable internet access.
    - Data privacy concerns with third-party hosting.
    - High costs for advanced features and scaling.

These systems have evolved to improve efficiency, but each type has had its drawbacks, leading to the development of more advanced solutions over time.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

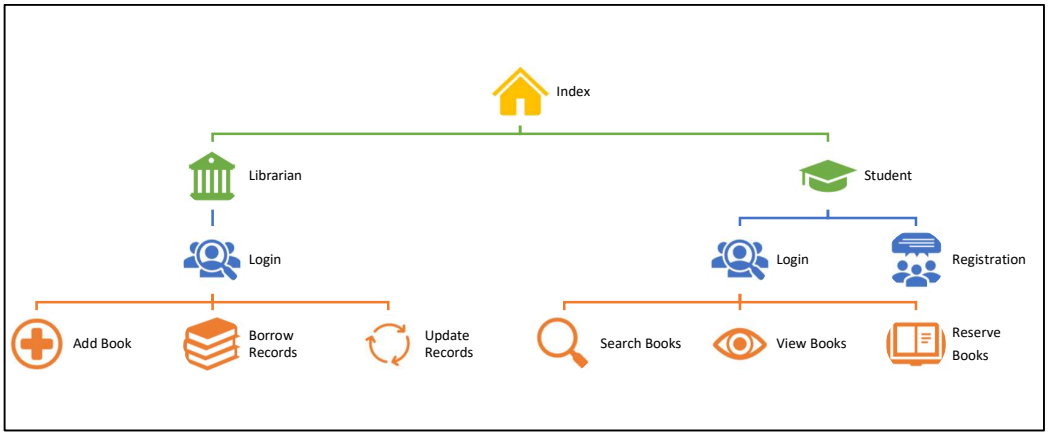

Fig 4.1 System Architecture

Student Portal: A segment of the system where students can perform various actions. It includes:

- **Login/Register**: Students can either log in to their existing account or create a new account to access the system.
- **Search Books**: Once logged in, students can search for books in the library by various criteria such as title, author, or genre.
- **Reserve Books**: Students can reserve available books for borrowing.
- **View Books**: Students can view the list of books they have reserved and their current status.

Librarian Portal: A segment of the system dedicated to librarians, enabling them to manage library resources effectively. It includes:

- **Login**: Librarians must log in to access the administrative features of the system.
- **Add Books**: Librarians can add new books to the library inventory, including details such as title, author, and ISBN.

- **Update Books**: Librarians can update the information of existing books in the library catalog.
- **Manage Book Transactions**: Librarians can track and manage books that have been borrowed and returned, ensuring accurate inventory and records.

4.2 UML DIAGRAMS

Use Case Diagram

The use case diagram for LIBRAREASE illustrates the interactions between the system and its users, specifically librarians and students. In this diagram, the primary actors include Librarians and Students. For Students, use cases include logging in, registering, searching for books, reserving books, and viewing borrowed books. For Librarians, use cases include logging in, adding new books, updating book information, and managing borrowed and returned books. This diagram provides a high-level view of the system's functionality and helps in understanding the roles and interactions within LIBRAREASE.

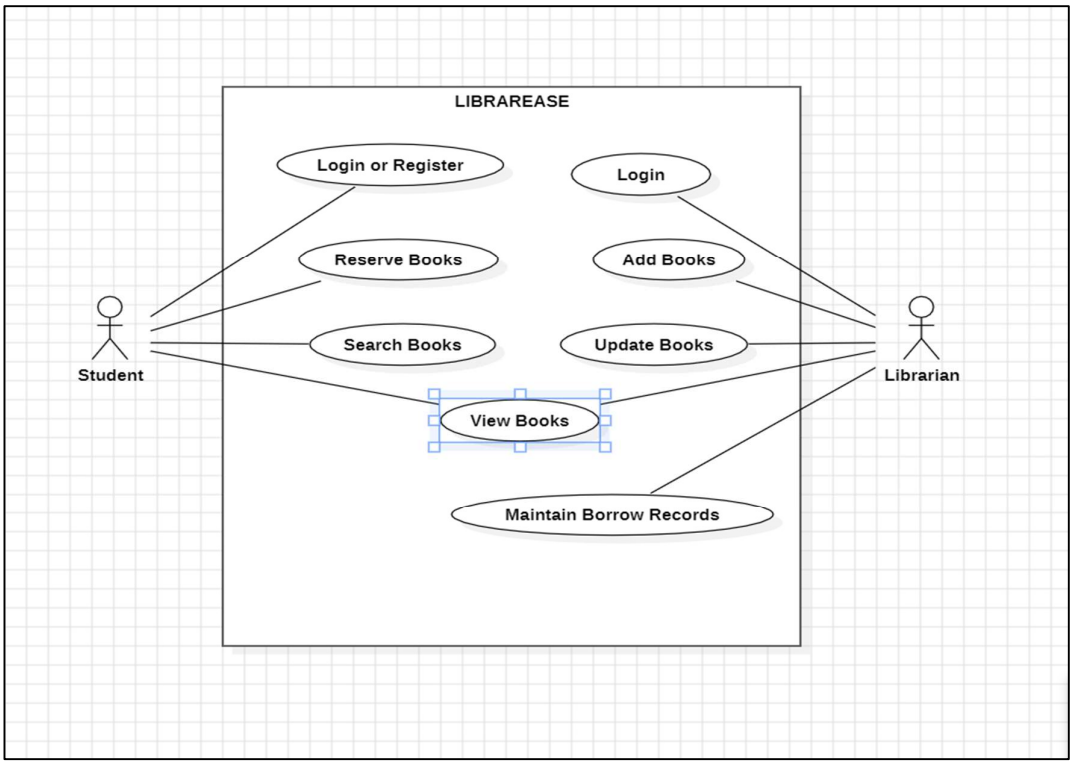

Fig 4.2.1 Usecase Diagram

Sequence Diagram

The sequence diagram for LIBRAREASE outlines the sequence of interactions between different objects or components during specific processes. For instance, during a book search operation, the sequence starts with the Student initiating a search request. The request is sent to the system, which processes it by querying the database. The results are then returned to the Student. This diagram details the flow of messages and interactions over time, providing clarity on how different parts of the system work together to fulfill a particular use case.

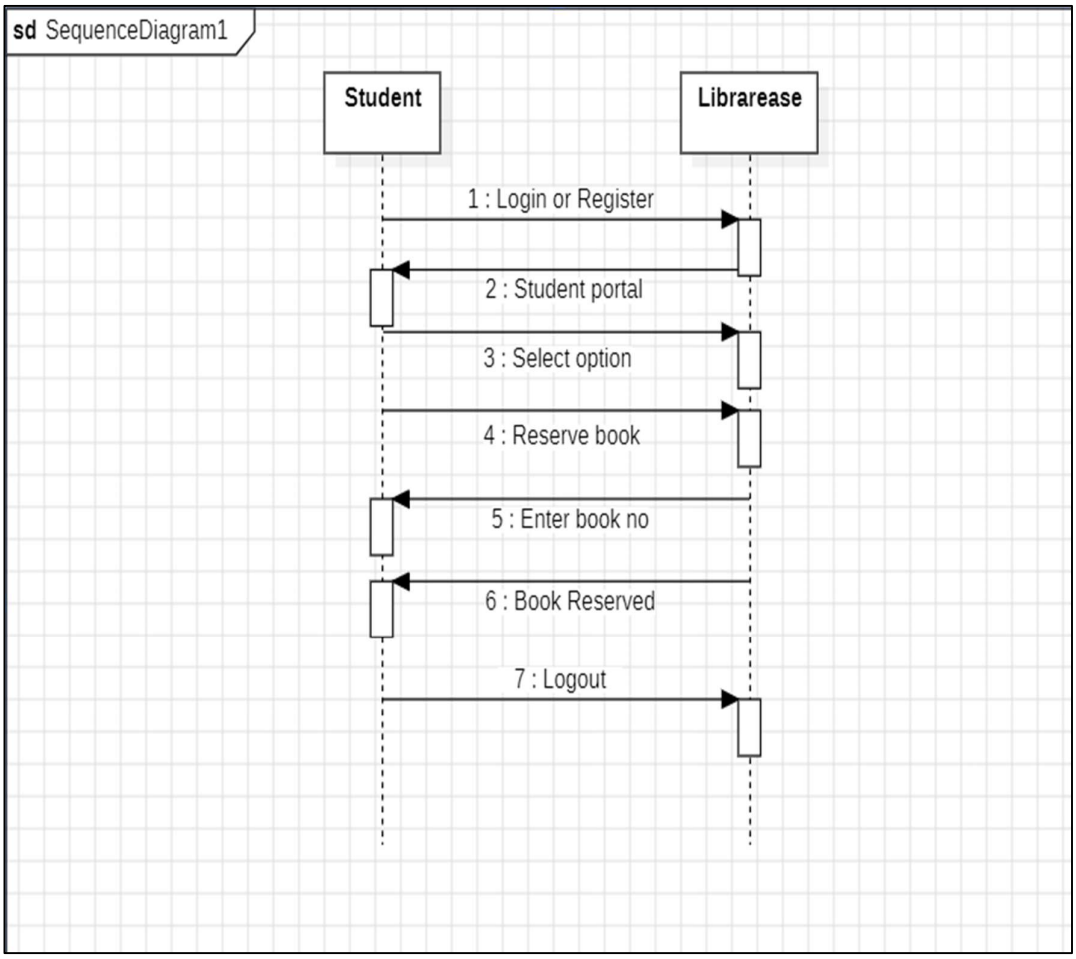

Fig 4.2.2 Sequence Diagram

Class Diagram

The class diagram for LIBRAREASE provides a structural view of the system by depicting its classes, attributes, methods, and the relationships between them. Key classes might include Student, Librarian, Book, Reservation, and Transaction. For instance, the Student class would have attributes such as studentID and name, and methods like searchBook() and reserveBook(). The Book class might include attributes such as bookID, title, and author, with methods for updating book details. This diagram helps in understanding the system's static structure and the interactions between different classes.

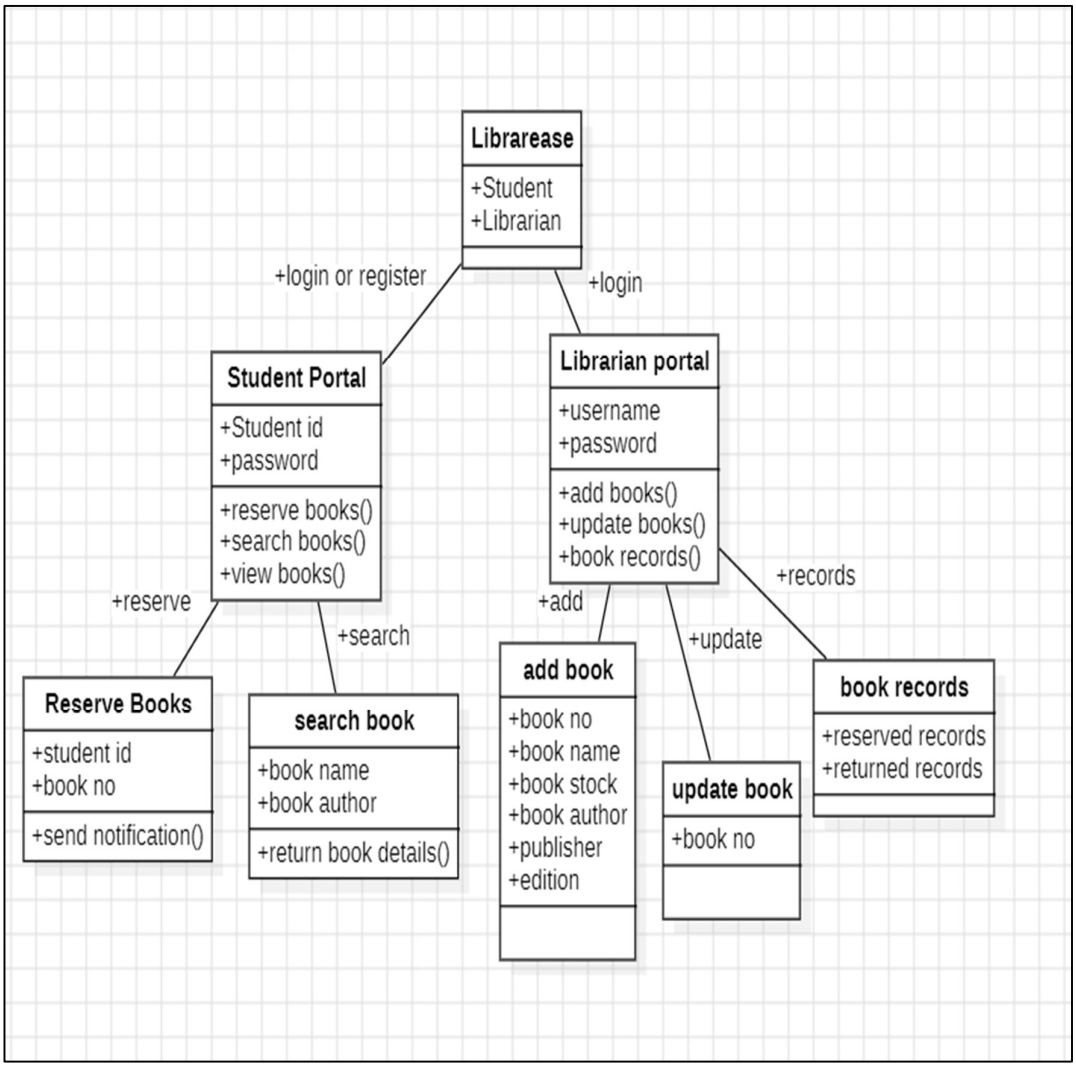

Fig 4.2.3 Class Diagram

Activity Diagram

The activity diagram for LIBRAREASE represents the workflow of specific processes within the system. For example, the process of a student searching for a book is illustrated with various activities such as entering search criteria, submitting the search, processing the request, and displaying search results. The diagram includes decision points and parallel activities, providing a visual representation of the flow of control and the steps involved in each process. This helps in visualizing the dynamic aspects of the system and understanding how different activities are coordinated.

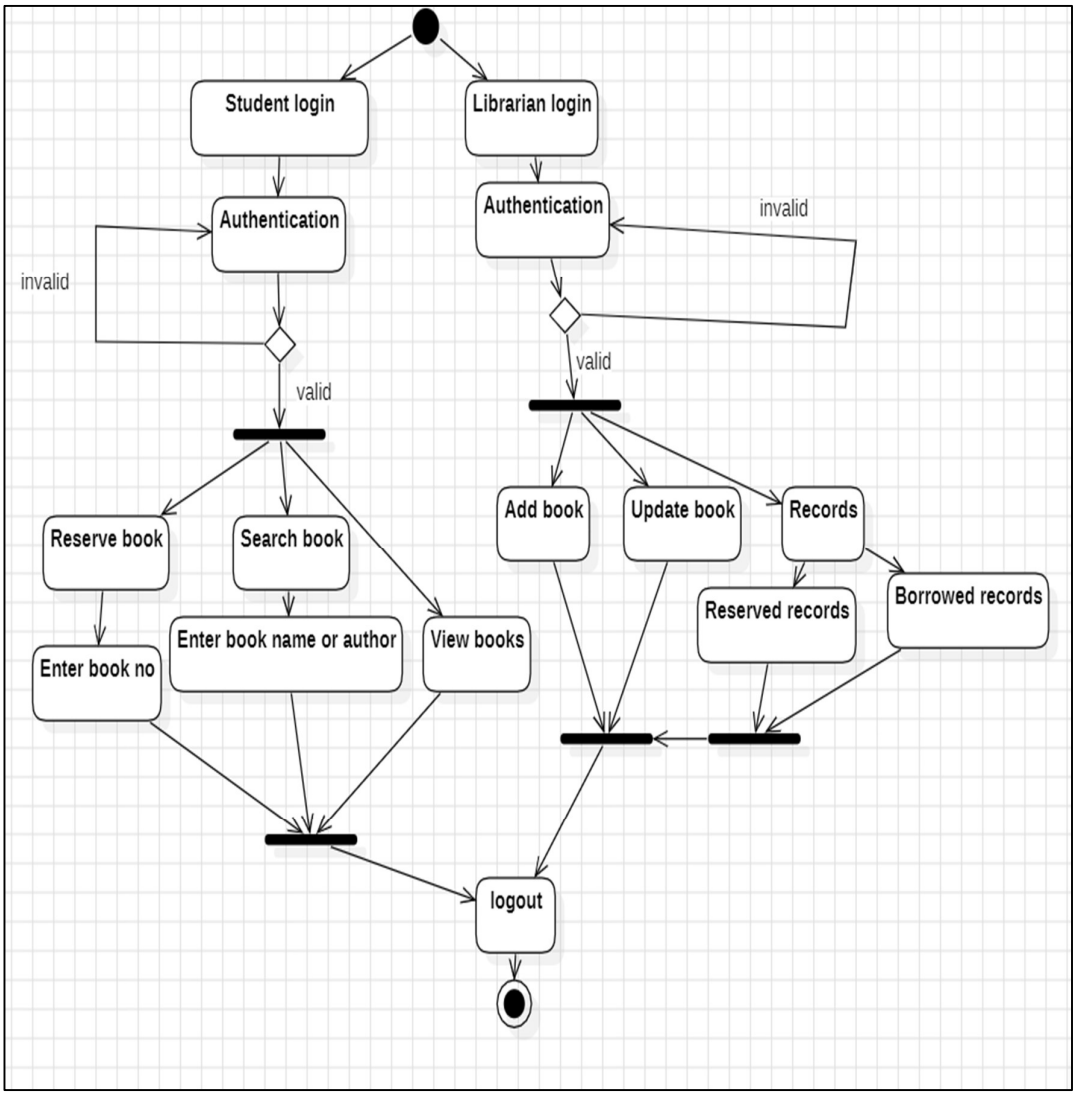

Fig 4.2.4 Activity Diagram

5. IMPLEMENTATION

5.1 IMPLEMENTATION

1. Data Collection: In LIBRAREASE, data collection involves gathering information about the library's books, students, and librarians. The book data includes title, author, ISBN, availability status, and other relevant information. Student data involves personal details like ID, name, and book borrowing history, while librarian data contains details about librarians and their administrative actions.

2. Data Preprocessing: Data preprocessing for LIBRAREASE includes setting up the MySQL database to store book, student, and librarian information. This involves normalizing book and user data, handling cases where data might be missing or incomplete, and ensuring data types are consistent and indexed for faster querying.

3. System Design: The system is designed using PHP for the backend, MySQL for data storage, and HTML/CSS/JavaScript for the user interface. This web-based system is designed to allow easy interactions for both students and librarians. Data is passed between the frontend and backend using forms and processed via PHP scripts.

4. Functional Implementation:

- **Student Operations**: Students can register, log in, search for books, reserve books, and view borrowed and reserved books. The system validates the student's credentials and provides a secure interface for these operations.
- **Librarian Operations**: Librarians can log in, add new books, update existing books, manage borrowed/returned books, and view the list of registered students. These operations involve updating the database with real-time data regarding book availability and student borrowing activity.

5. User Interface: The user interface was implemented using HTML, CSS, and JavaScript to provide a user-friendly experience. The interface for students and librarians is designed differently, with students primarily focused on book operations (borrow, search, reserve), while librarians focus on managing the library inventory and monitoring student activities.

6. Database Management: LIBRAREASE's database management involves creating tables for books, students, librarians, and borrowed books. Queries are used for data retrieval and manipulation, ensuring that each user action (like borrowing or reserving a book) is accurately reflected in the database.

7. Security & Validation: Security features are implemented through session management to ensure that only authenticated users can access the system. Data validation ensures that users cannot input invalid information, such as trying to borrow a book that is unavailable.

8. System Testing: The system is tested by logging in as both students and librarians to ensure all operations like adding books, borrowing, reserving, and viewing are working as expected. Functional testing ensures each operation interacts correctly with the MySQL database.

9. Deployment: LIBRAREASE is deployed using XAMPP as the local server environment. The system is made accessible via a web browser, allowing real-time interactions with the library's database.

10. Maintenance and Updates: The system is designed for easy future updates, such as adding new features or handling more complex book management rules. Continuous monitoring ensures that the system functions efficiently with new data being entered.

5.2 SAMPLE CODE

1. LIBRAREASE home page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Home</title>
</head>
<body>
<?php
    session_start();
    if (isset($_SESSION['id'])) {
        header("Location: studenthome.php");
        exit();
    }
    if (isset($_SESSION['faculty_id'])) {
        header("Location: facultyhome.php");
        exit();
    }
?>
    <header>
        <i class="fas fa-book-open fa-icon"></i>
        LIBRAREASE
        <i class="fas fa-book-open fa-icon"></i>
    </header>
    <div class="button-container">
        <div class="button">
```

```
        <img src="book.png" alt="Librarian Button" class="lib" onclick="window.open('<?php echo isset($_SESSION['faculty_id']) ? 'facultyhome.php' : 'facultylogin.php' ?>', '_blank')">

        <h1>Librarian</h1>

      </div>

      <div class="button">

        <img src="graduated.png" alt="Student Button" class="stu" onclick="window.open('<?php echo isset($_SESSION['id']) ? 'studenthome.php' : 'loginpage.php' ?>', '_blank')">

        <h1>Student</h1>

      </div>

   </div>

</body>

</html>
```

2.Student registration

```
<!DOCTYPE html>
<html>

<head>
   <title>
      REGISTRARTION
   </title>
   <link rel="stylesheet" href="style1.css">
         <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css"
                                                integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJNuGM4XkWuk94WCrrwslk8yWNGmY1EduTA=="
      crossorigin="anonymous" referrerpolicy="no-referrer" />
   <style>
      //css to be added
   </style>
</head>

<body>
```

```
   <header><marquee>Student Registration Page   <img src="https://play-
lh.googleusercontent.com/mtDBaLnbmAi3-
PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8"
class="img"></marquee></header>
   <div class="register">
      <div class="form">


         <h3><font color="black">Student Registration</font></h3>
        <?php
        $conn=mysqli_connect('localhost','root','','library');
        if(mysqli_connect_error()){
           exit("Error connecting to Database");
        }
        if(isset($_POST["submit"])){
          $name=$_POST["name"];
          $id=$_POST["id"];
          $email=$_POST["email"];
          $password=$_POST["password"];
          $dob=$_POST["dob"];
          $rp=$_POST["rp"];
        $errors=array();

        if(strlen($password)<8){
           array_push($errors,"password is too weak");
        }
        if($password!=$rp){
           array_push($errors,"password is not matched");
        }

        if($stmt=$conn->prepare( 'SELECT * FROM student WHERE id = ?')){
           $stmt->bind_param('s',$_POST['id']);
           $stmt->execute();
           $stmt->store_result();

           if($stmt->num_rows()>0){
              array_push($errors,"id already exists!");
           }
        }

        if(count($errors)>0){
           foreach($errors as $error){
```

```
            echo "<div class='danger'>$error</div>";
          }
        }
        else{

 if($stmt=$conn->prepare('INSERT INTO student (username,password,id,email,dob)
VALUES ( ?, ?, ?, ?, ? )')){

            $stmt->bind_param("sssss",$name,$password,$id,$email,$dob);
            $stmt->execute();

      echo '<script type="text/javascript">';
      echo 'alert("successfully registered")';
      echo '</script>';
          }
        else{
            die("Something went wrong");
          }
         }
      }
        ?>
        <form action="register.php" method="post" id="form">
          <div class="name">
            <div class="input">
            <i class="icon fas fa-user"></i>
                <input type="text" placeholder="Enter your Full name" name="name"
id="name" class="data" required>

            </div>
          </div>
          <div class="idno">
            <div class="input">
            <i class="icon fas fa-address-card"></i>
             <input type="text" placeholder="Enter Identity number" name="id" id="id"
class="data" required>

            </div>
          </div>
          <div class="email">
            <div class="input">
            <i class="icon fas fa-envelope" ></i>
```

```
                    <input type="email" placeholder="Enter your Email" name="email"
id="email" class="data" required>

                </div>
            </div>
            <div class="password">
                <div class="input">
                <i class="icon fas fa-lock"></i>
                  <input type="password" placeholder="Enter Password" name="password"
id="password" class="data" required>

                </div>
            </div>
            <div class="check password">
                <div class="input">
                <i class="icon fas fa-lock"></i>
                      <input type="password" placeholder="Re-enter password" name="rp"
id="rp" class="data" required>

                 </div>
            </div>

            <div class="gender">
                <div class="input">
                    <input type="radio" name="gender" id="male"  value="male">
                    <i class="icon fas fa-male "></i>
                    <label for="male" style="color:black;">Male</label>

                    <input type="radio" name="gender" id="female"  value="female">
                    <i class="icon fas fa-female "></i>
                    <label for="female" style="color:black;">Female</label>

                </div>
            </div>
            <div class="flex">
                <div class="dob">
                    <div class="input">
                        <input type="date" name="dob" id="dob" required class="data">
                    </div>
                </div>
            </div>
            <div class="select_btn">
```

```
            <div class="input">
                                <input type="submit" value="Register" class="but" name="submit">    
                <input type="reset" class="but" value="Clear" name="clear">
            </div>
            <div class="message">
              <div class="input">
                    <p class="login">Already registerd then <a href="loginpage.php" target="_self" class="log">Login</a> </p>
              </div>
            </div>

        </form>
      </div>
  </div>

</body>

</html>
```

3.Student login

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Student Login</title>
  <style>
    //css to be added
  </style>
</head>
<body>
      <header><marquee>Welcome to Student Login Page   <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8" class="img"></marquee></header>
  <form action="login.php" class="form" method="post" >
    <h1 align="center"><font color="white">Student Login</font></h1>
    <?php if(isset($_POST['submit'])) {?>
      <p class="error"><?php echo $_GET['error'];?></p>
    <?php } ?>
 <br>
  <center>
```

```
  <div class="mainContainer">
          <p class="data">Username: <input type="text" placeholder="Student-ID" autocomplete="off" name="user" class="name"></p>
     <br><br>
        <p class="data">Password: <input type="password" placeholder="Password" autocomplete="off" name="password" class="pass"></p>
     <br><br>
                        <input type="submit" value="Login" class="user" name="submit">    
                        <input type="button" value="Home" class="user" onclick="window.open('index.php', '_self')">    
     <input  type="reset" value="Clear" class="user">
        <p class="data">New Member ?<a href="register.php"style="color:black;text-decoration:none">Sign-Up</a></p>
     <p class="data"><a href="forgot.php">Forgot Password</a></p>
     <br>
  </div>
  </center>
</form>
</body>
</html>
```

4.Student home

```
<?php
  session_start();
   if(isset($_SESSION['id']) && isset($_SESSION['username']))
   {
  ?>
<!DOCTYPE html>
     <html lang="en">
     <head>
        <title>Student</title>
              <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css">
        <style>
           //css to be added
        </style>
     </head>
     <body>
     <header><marquee>Welcome to Library of Geetanjali College Of Engineering And Technology <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-
```

```
PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8"
class="img"></marquee></header>
        <div class="hello">
          <h1 class="hi">Hello, <?php echo $_SESSION['username']; ?></h1>
        </div>
        <div class="button-container">
          <div class="button">
                  <button class="but" id="search" onclick="window.open('search.php',
'_self')">
            <i class="fas fa-search"></i>Search
            </button>

          </div>
          <div class="button">

             <button class="but" id="search"onclick="window.open('bookborrow.php',
'_self')">
            <i class="fas fa-book"></i>Borrow
            </button>

          </div>

          <div class="button">
            <button class="but" id="search" onclick="window.open('total.php', '_self')">
            <i class="fas fa-eye"></i>View
            </button>

          </div><br><br>
          <div class="button">
                    <button class="but" onclick="window.open('logout.php', '_self')"
id="logout">
            <i class="fas fa-sign-out-alt"></i>Logout
            </button>

          </div>
      </div>
    </body>
    </html>
    <?php
  }
  else
  {
```

```
        header ("Location:loginpage.php");
        exit();
    }
?>
```

5.Search Book

```
<?php
   include "db_con.php";

   if(isset($_POST['search']))
   {
      $book_name = $_POST['book_name'];

      $stmt = $con->prepare("SELECT * FROM books WHERE book_name = ?");
      $stmt->bind_param("s", $book_name);

      $stmt->execute();

      $result = $stmt->get_result();

      if ($result->num_rows > 0) {
         $row = $result->fetch_assoc();

                header("Location: view.php?book_no=" . urlencode($row['book_no']) .
"&book_name=" . urlencode($row['book_name']) . "&book_stock=" .
urlencode($row['book_stock']));
         exit();
      } else {
         echo '<script type="text/javascript">';
         echo 'alert("No record found")';
         echo '</script>';
      }

      $stmt->close();
      $con->close();
   }
?>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
```

```
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Search</title>
         <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
   <style>
      //css to be added
   </style>
</head>
<body>
<header><marquee>Welcome to Search Book Page <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8" class="img"></marquee></header>
<form action="search.php" class="form" method="post" name="f1">
   <h1 align="center">Search Book</h1>
   <br>
   <center>
      <div class="mainContainer">
         <p class="data">Book Name:
               <input type="text" placeholder="Enter Book Name" autocomplete="off" name="book_name" class="name">
         </p>
         <br>
        <button type="submit" class="user" name="search"><i class="fas fa-search"></i> Search</button>
          <button type="button" class="user" onclick="window.open('studenthome.php', '_self')"><i class="fas fa-home"></i> Home</button>
         <button type="reset" class="user"><i class="fas fa-times"></i> Clear</button>
         <br>
      </div>
   </center>
</form>

</body>
</html>
```

6.Reserve Book

```
<?php
   include "db_con.php";
   if(isset($_POST['submit']))
   {
```

```
    $id= $_POST['stu_id'];
    $book_name= $_POST['book_name'];
    $q="select book_no from books where book_name='$book_name'";
    $r=mysqli_query($con,$q);
    if(mysqli_num_rows($r)>0)
    {
       $row=mysqli_fetch_assoc($r);
       $book_no=$row['book_no'];
    }
    else
    {
       echo '<script type="text/javascript">';
       echo 'alert("book name is incorrect please check")';
       echo '</script>';
    }
    $td=date("Y-m-d");
    $rd=date("Y-m-d",strtotime($td."+10days"));
       $sql = "INSERT INTO borrowed (stu_id, book_no, book_name, given_date,
return_date) VALUES ('$id', '$book_no', '$book_name', '$td', '$rd')";
    $result = $con->query($sql);


    if($result == TRUE)
    {
       echo '<script type="text/javascript">';
       echo 'alert("Borrow successful! Return by: ' . $rd . '")';
       echo '</script>';

        $up = "UPDATE books SET book_stock = book_stock - 1 WHERE book_no =
'$book_no'";
       $up_result = $con->query($up);
       if (!$up_result)
       {
          echo "Error updating book stock: " . $con->error;
       }
    }
    else
    {
       echo "Error:".$sql."<br>".$con->error;
    }
    $con->close();
  }
```

```
?>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Borrow Books</title>
   <style>
      //add css here
      }
   </style>
</head>
<body>
<header><marquee>Welcome to Student Borrow Book Page <img src="https://play-
lh.googleusercontent.com/mtDBaLnbmAi3-
PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8"
class="img"></marquee></header>
<form action="bookborrow.php" class="form" method="post" name="f1">
      <h1 align="center">Borrow Book</h1>
  <br>
   <center>
   <div class="mainContainer">
       <p class="data">Student Id:<input type="text" placeholder="Student Id"
autocomplete="off" name="stu_id" class="name"></p><br>
      <p class="data">Book Name:<input type="text" placeholder="Enter Book Name"
autocomplete="off" name="book_name" class="name"></p><br>
                        <input type="submit" value="Borrow" class="user"
name="submit">    
                          <input type="button" value="Home" class="user"
onclick="window.open('studenthome.php', '_self')">    
      <input  type="reset" value="Clear" class="user">
      <br>
   </div>
   </center>
</form>
</body>
</html>
```

7.View Books

```
<?php
session_start();

if (!isset($_SESSION['faculty_id'])) {
   header("Location: facultylogin.php");
   exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
   <title>Faculty</title>
            <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
   <style>
      //css to be added
   </style>
</head>
<body>
<header><marquee>Welcome to Faculty Home Page <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8" class="img"></marquee></header>
<div class="hello">
   <h1>Hello, Admin</h1>
</div>
<div class="button-box">
   <div class="button-container">
      <p>Choose the Operation you want to perform</p>
      <button class="button" id="add" onclick="window.open('add.php', '_self')">
      <i class="fas fa-plus-circle" ></i>Add
      </button>
      <button class="button" id="change" onclick="window.open('update.php', '_self')">
      <i class="fas fa-edit"></i>Update
      </button>
      <button class="button" onclick="window.open('logout.php', '_self')" id="logout">
      <i class="fas fa-sign-out-alt"></i>Logout
      </button>
   </div>
```

```
</div>
</body>
</html>
```

8.Faculty login

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Faculty Login</title>
   <style>
   //css to be added
   </style>
</head>
<body>
       <header><marquee>Welcome to Faculty Login Page   <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8" class="img"></marquee></header>
   <form action="facultycheck.php" class="form" method="post" name="f1">
      <h1 align="center">Faculty Login</h1>
      <?php if(isset($_GET['error'])) {?>
         <p class="error"><?php echo $_GET['error'];?></p>
      <?php } ?>
  <br>
   <center>
   <div class="mainContainer">
       <p class="data">Username: <input type="text" placeholder="Faculty Username" autocomplete="off" name="user" class="name"></p>
      <br><br>
           <p class="data">Password: <input type="password" placeholder="Faculty Password" autocomplete="off" name="password" class="pass"></p>
      <br><br>
      <input  type="submit" value="Login" class="user">    
                        <input type="button" value="Home" class="user" onclick="window.open('index.php', '_self')">    
      <input  type="reset" value="Clear" class="user">
      <br>
   </div>
   </center>
</form>
</body></html>
```

9.Faculty home

```
<?php
session_start();

if (!isset($_SESSION['faculty_id'])) {
   header("Location: facultylogin.php");
   exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
   <title>Faculty</title>
            <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta3/css/all.min.css">
   <style>
     //css to be added
   </style>
</head>
<body>
<header><marquee>Welcome to Faculty Home Page <img src="https://play-
lh.googleusercontent.com/mtDBaLnbmAi3-
PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8"
class="img"></marquee></header>
<div class="hello">
   <h1>Hello, Admin</h1>
</div>
<div class="button-box">
   <div class="button-container">
      <p>Choose the Operation you want to perform</p>
      <button class="button" id="add" onclick="window.open('add.php', '_self')">
      <i class="fas fa-plus-circle" ></i>Add
      </button>
      <button class="button" id="change" onclick="window.open('update.php', '_self')">
      <i class="fas fa-edit"></i>Update
      </button>
      <button class="button" onclick="window.open('logout.php', '_self')" id="logout">
      <i class="fas fa-sign-out-alt"></i>Logout
      </button>
```

```
    </div>
</div>
</body>
</html>
```

10.Add Book

```
<?php
    include "db_con.php";

    if(isset($_POST['submit']))
    {
        $book_no= $_POST['book_no'];
        $book_name= $_POST['book_name'];
        $book_stock= $_POST['book_stock'];

                    $sql="INSERT INTO books (book_no,book_name,book_stock)
VALUES('$book_no','$book_name','$book_stock')";

        $result=$con->query($sql);

        if($result == TRUE)
        {
            echo '<script type="text/javascript">';
            echo 'alert("successfully added")';
            echo '</script>';
        }
        else
        {
            echo "Error:".$sql."<br>".$con->error;
        }
        $con->close();
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        //css to be added
```

```
   </style>
</head>
<body>
<header><marquee>Welcome to Faculty Add Book Page <img src="https://play-lh.googleusercontent.com/mtDBaLnbmAi3-PV8s4wpSeI28LayXVWgFciMurbDrRsmmDLesTVuMCbRkyPJZb65zA8" class="img"></marquee></header>
<form action="add.php" class="form" method="post" name="f1">
      <h1 align="center">Add Book</h1>
  <br>
   <center>
   <div class="mainContainer">
       <p class="data">Book Number:<input type="text" placeholder="Enter Book No" autocomplete="off" name="book_no" class="name"></p>
      <br>
       <p class="data">Book Name:<input type="text" placeholder="Enter Book Name" autocomplete="off" name="book_name" class="name"></p><br>
       <p class="data">Book Stock:<input type="text" placeholder="Enter Book Stock" autocomplete="off" name="book_stock" class="name"></p>
      <br>
                              <input type="submit" value="Add" class="user" name="submit">    
                             <input type="button" value="Home" class="user" onclick="window.open('facultyhome.php', '_self')">    
      <input  type="reset" value="Clear" class="user">
      <br>
   </div>
   </center>
</form>
</body>
</html>
```

11.Reserved Records

```
<?php
$connection = mysqli_connect('localhost', 'root', '', 'library');
if (!$connection) {
   die('Database connection failed: ' . mysqli_connect_error());
}
$query = "SELECT * FROM borrowed";
$result = mysqli_query($connection, $query);
mysqli_close($connection);
```

```
?>
<!DOCTYPE html>
<html lang="en">
<head>
   <title>List of Books</title>
</head>
<style>
       //css should be added here
   </style>
<body>
<div class="container">
   <h1>Borrowed Books</h1>
   <table>
      <thead>
         <tr>
            <th>Student Id</th>
            <th>Book No</th>
            <th>Book Name</th>
            <th>Borrowed date</th>
            <th>Return date</th>
            <th>Status</th>
         </tr>
      </thead>
      <tbody>
         <?php
         while ($row = mysqli_fetch_assoc($result))
         {
            echo '<tr>';
            echo '<td>' . $row['stu_id'] . '</td>';
            echo '<td>' . $row['book_no'] . '</td>';
            echo '<td>' . $row['book_name'] . '</td>';
            echo '<td>' . $row['given_date'] . '</td>';
            echo '<td>' . $row['return_date'] . '</td>';
                echo '<td><a href="return.php?borrowed_id=' . $row['borrowed_id'] . '"
class="return-link"><button class="return-button">Returned</button></a></td>';
            echo '</tr>';
         }
         ?>
      </tbody>
   </table>
   <div class="but">
         <a href="facultyhome.php" target="_self">
```

```
            <button id="but">Home</button>
          </a>
   </div>
</div>
</body>
</html>
```

12.Borrowed Records

```
<?php
$con = mysqli_connect('localhost', 'root', '', 'library');
if (!$con)
{
   die('Database connection failed: ' . mysqli_connect_error());
}

if (isset($_GET['borrowed_id']))
{
   $borrow_id = $_GET['borrowed_id'];

   $query = "SELECT book_no FROM borrowed WHERE borrowed_id = '$borrow_id'";
   $result = mysqli_query($con, $query);

   if (mysqli_num_rows($result) > 0)
   {
      $row = mysqli_fetch_assoc($result);
      $book_no = $row['book_no'];

       $updateQuery = "UPDATE books SET book_stock = book_stock + 1 WHERE
book_no = '$book_no'";
      $updateResult = mysqli_query($con, $updateQuery);

      if (!$updateResult)
      {
         echo "Error updating book stock: " . mysqli_error($con);
      }

      $deleteQuery = "DELETE FROM borrowed WHERE borrowed_id = '$borrow_id'";
      $deleteResult = mysqli_query($con, $deleteQuery);

      if (!$deleteResult)
      {
```

```
            echo "Error deleting row: " . mysqli_error($con);
        }
    }
}

mysqli_close($con);
header("Location: borrow.php");
?>
```

13.db_con.php

```
<?php
    $host="localhost";
    $user="root";
    $password="";
    $db_name="library";
    $con=mysqli_connect($host,$user,$password,$db_name);
    if(!$con)
    {
        echo "Connection Failed";
    }
?>
```

For full code : https://github.com/lekhya04/librarease

6. TESTING

6.1 TESTING

Software testing is a critical component of the software development lifecycle. It involves the process of evaluating software or a system's performance, functionality, and quality to identify and rectify defects, bugs, or errors. The primary purpose of software testing is to ensure that the software or system meets the specified requirements and performs as expected. The types of software testing can be broadly classified into two categories: manual testing and automated testing. Manual testing involves executing test cases manually, while automated testing uses tools to automate the testing process. Some of the most common types of software testing include:

Unit Testing

Unit testing involves testing individual components or modules of LIBRAREASE in isolation to ensure they work correctly. For example:

- Testing the login function to check if the system correctly authenticates students and librarians.
- Testing the book search functionality to ensure it returns the correct books based on the entered search terms.
- Verifying that adding a new book works as expected, ensuring the book is successfully inserted into the database.

Integration Testing

Integration testing ensures that different modules in LIBRAREASE interact correctly when combined. In this phase, individual modules are tested as a group:

- Testing how the student login module interacts with the book borrowing module, ensuring that only authenticated users can borrow books.
- Verifying that the book reservation function properly updates the book availability status in the database.

- Ensuring the librarian's book management interface integrates smoothly with the book database.

Black Box Testing

Black box testing focuses on testing the system's functionality without looking at the internal code structure:

- Testing the student registration form to ensure it properly handles user input (valid or invalid) and displays appropriate error messages.
- Checking whether searching for a book returns the correct results based on the title, author, or ISBN entered by the user.
- Verifying that book borrowing correctly changes the book status to "borrowed" and updates the student's record.

White Box Testing

White box testing focuses on the internal workings of the LIBRAREASE code:

- Testing the PHP code that handles book borrowing to ensure it correctly updates the database and user interface.
- Reviewing the authentication logic to verify it handles all possible cases (e.g., correct and incorrect passwords, session timeouts).
- Checking that the database queries used for book search and retrieval are optimized and return accurate results.

6.2 TEST CASES

S.No	Test Case Description	Expected Result	Test Result
1	Test student login with valid credentials	The system should allow the student to log in and redirect them to their dashboard.	**Success**
2	Test student login with invalid credentials	The system should display an error message indicating incorrect username or password.	**Success**
3	Test librarian login with valid credentials	The system should allow the librarian to log in and redirect them to the librarian dashboard.	**Success**
4	Test librarian login with invalid credentials	The system should display an error message indicating incorrect username or password.	**Success**
7	Test searching a book by title	The system should return a list of books matching the entered title.	**Success**
8	Test searching a book by author	The system should return a list of books by the entered author.	**Success**
9	Test searching for a non-existent book	The system should return a message indicating no results were found.	**Success**
10	Test book reservation by a student	The system should allow the student to reserve a book and update the reservation status in the system.	**Success**
12	Test librarian adding a new book	The system should allow the librarian to add a new book, and it should appear in the book database.	**Success**
13	Test librarian updating book details	The system should allow the librarian to update book details and reflect the changes in the database.	**Success**
14	Test librarian managing borrowed books (mark as returned)	The system should allow the librarian to mark a book as returned and update the book status.	**Success**

7. OUTPUT SCREENS

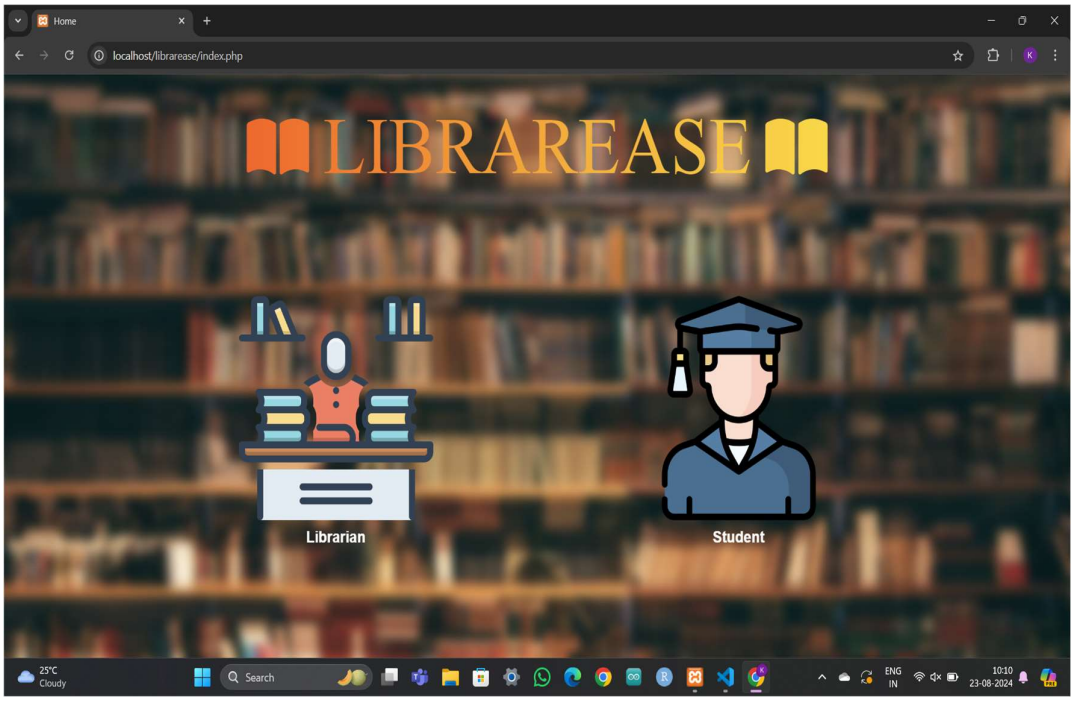

Fig 7.1 LIBRAREASE Home Page

Fig 7.2 Student Login Page

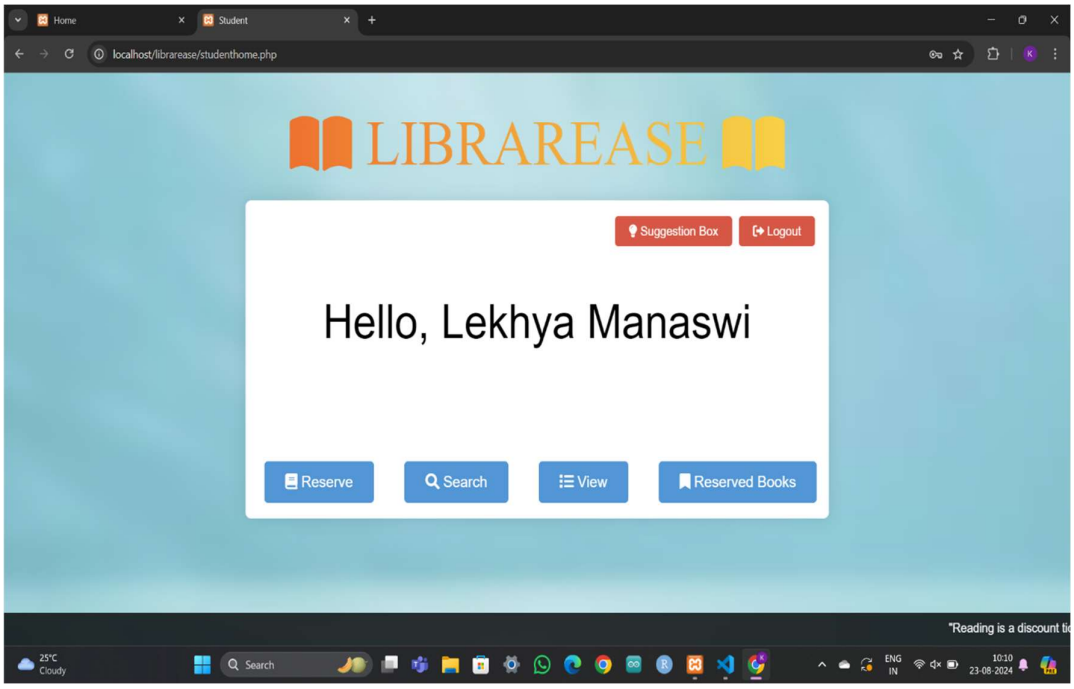

Fig 7.3 Student Home Page

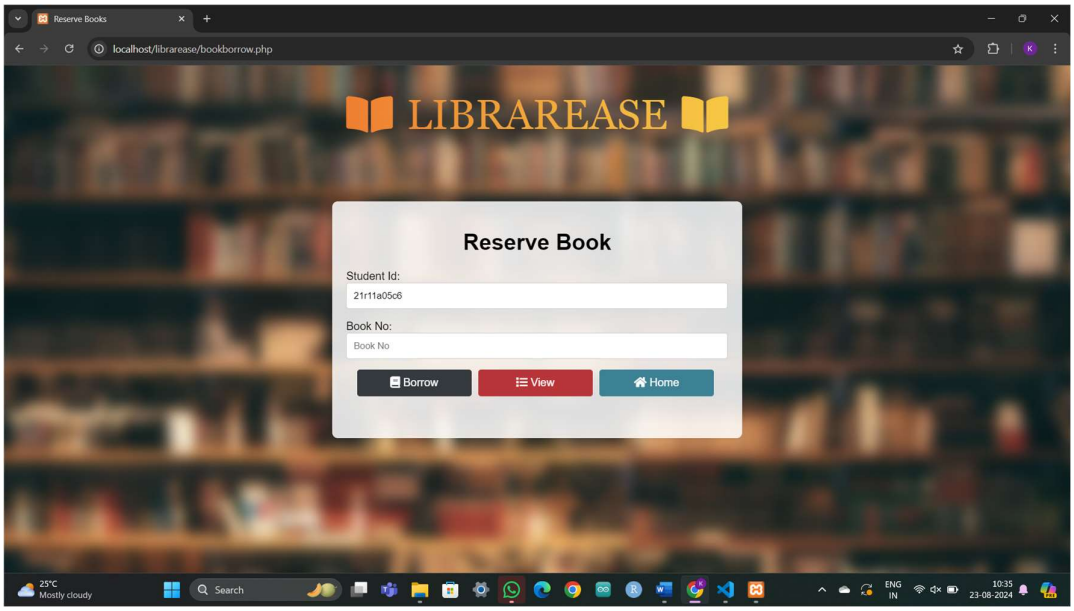

Fig 7.4 Book Reserving Page

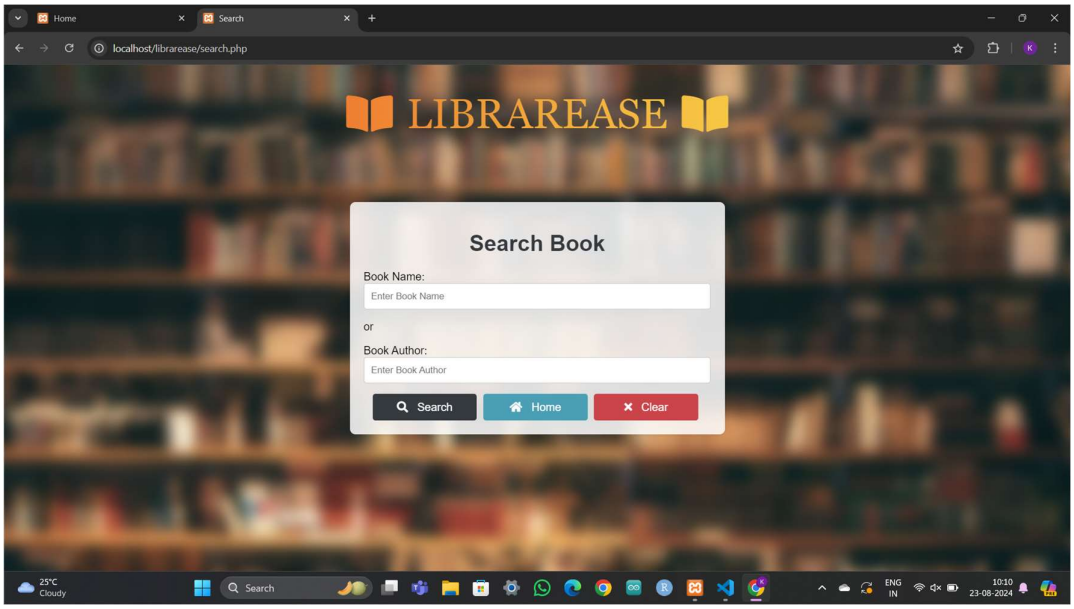

Fig 7.5 Search Book Page

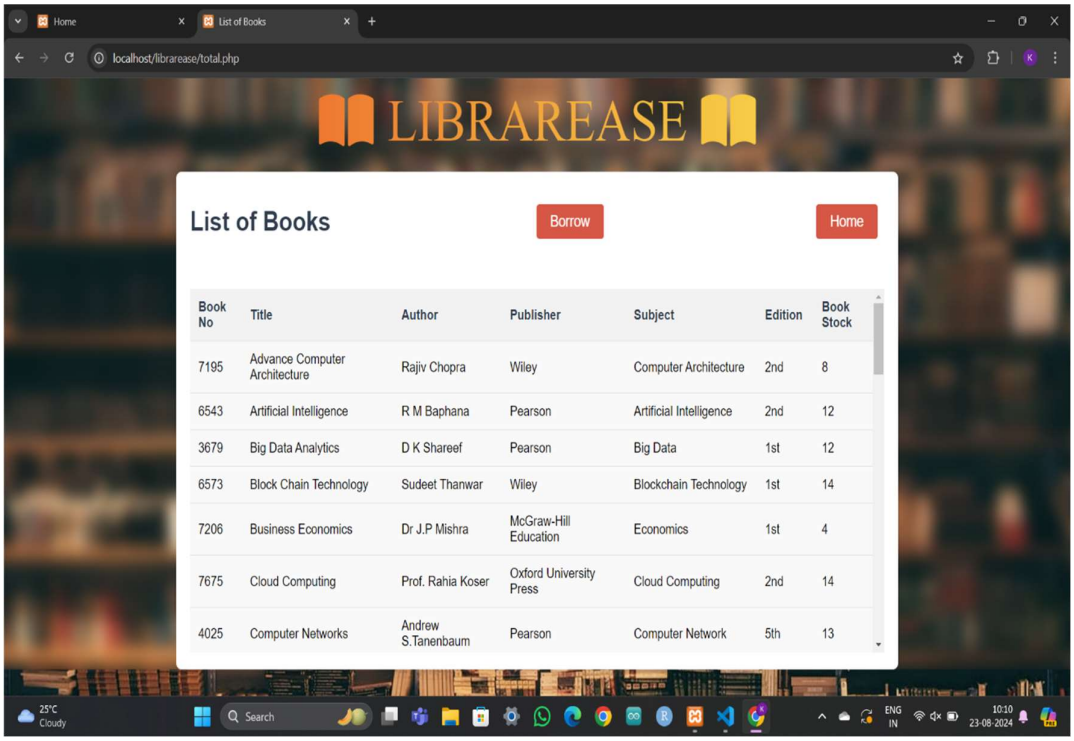

Fig 7.6 View All Books Page

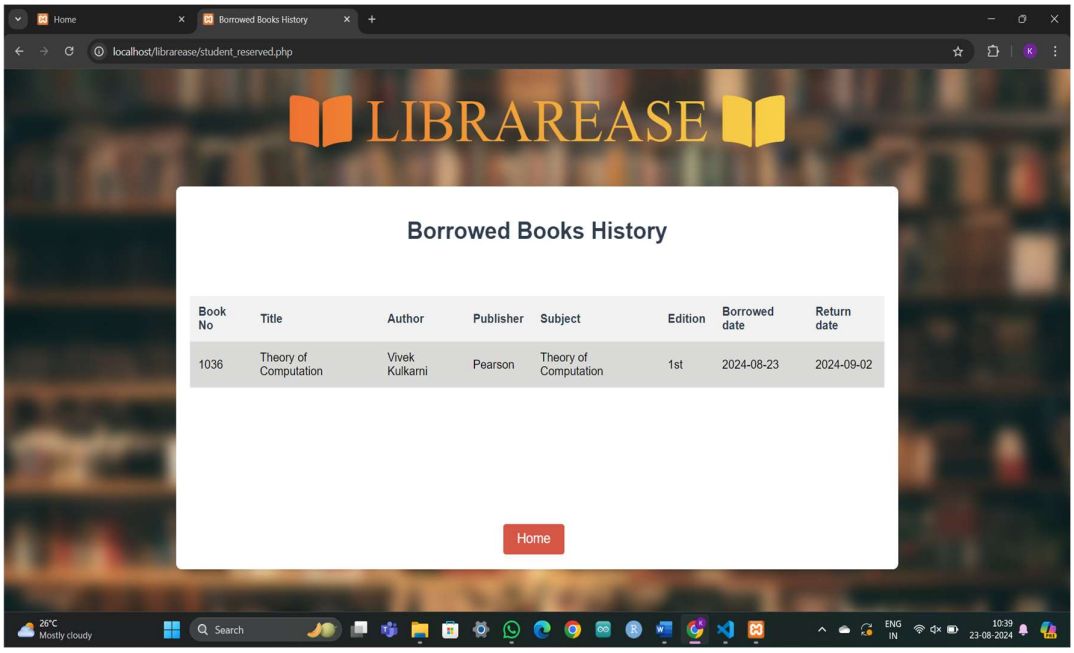

Fig 7.7 History of Borrowed Books Page

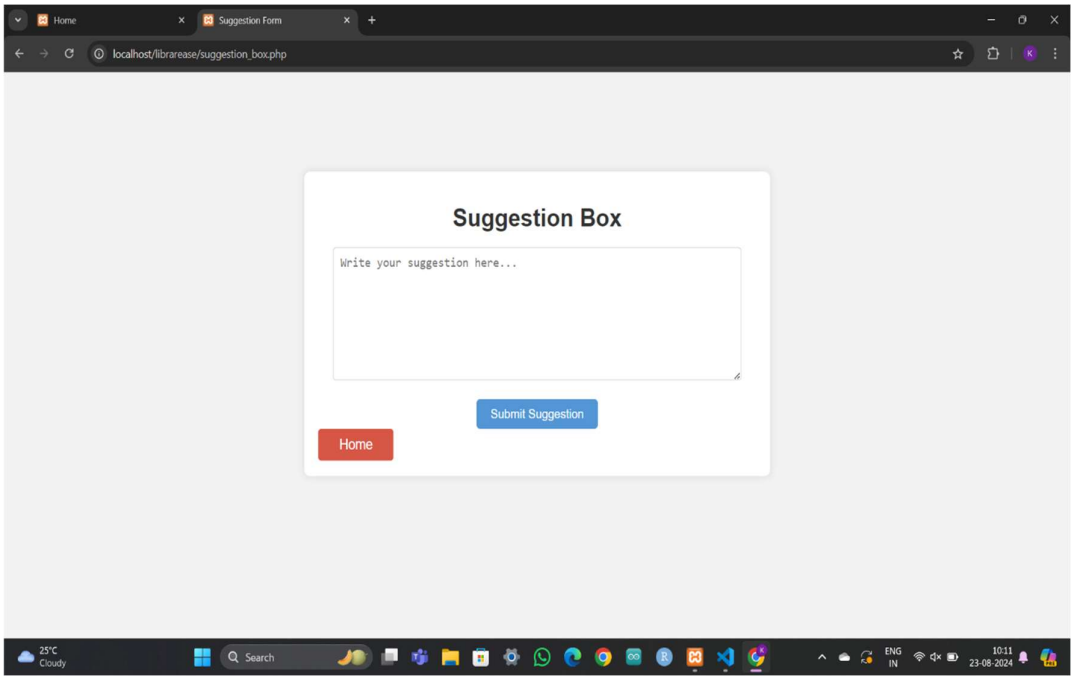

Fig 7.8 Student Suggestion Box Page

Fig 7.9 Admin Login Page

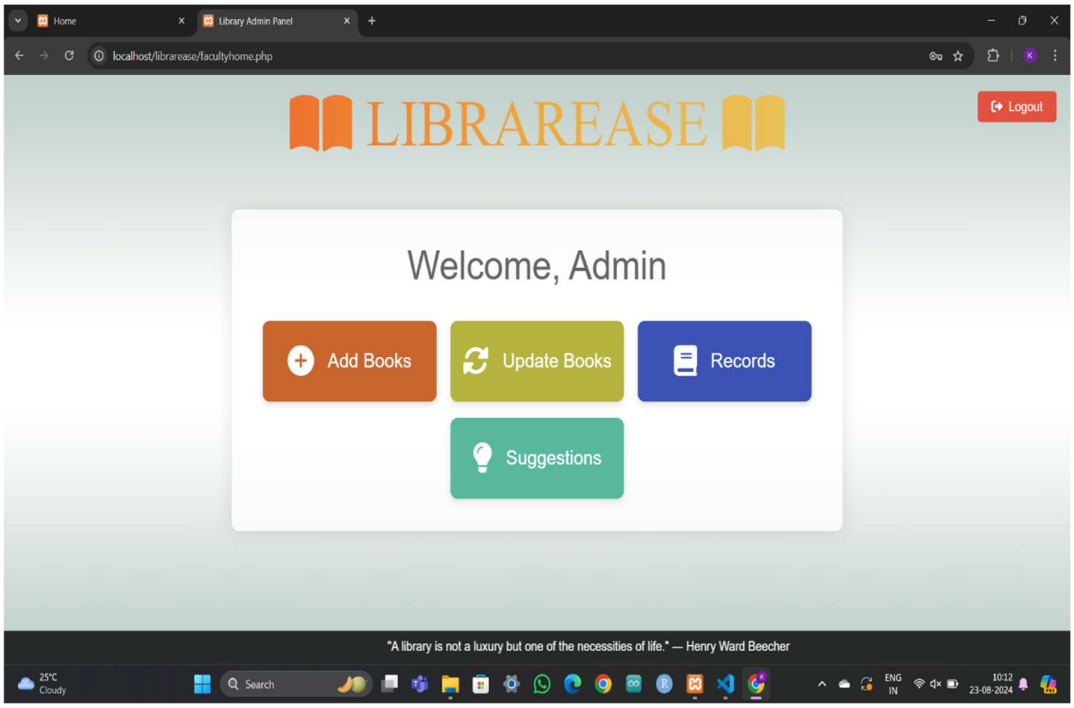

Fig 7.10 Admin Home Page

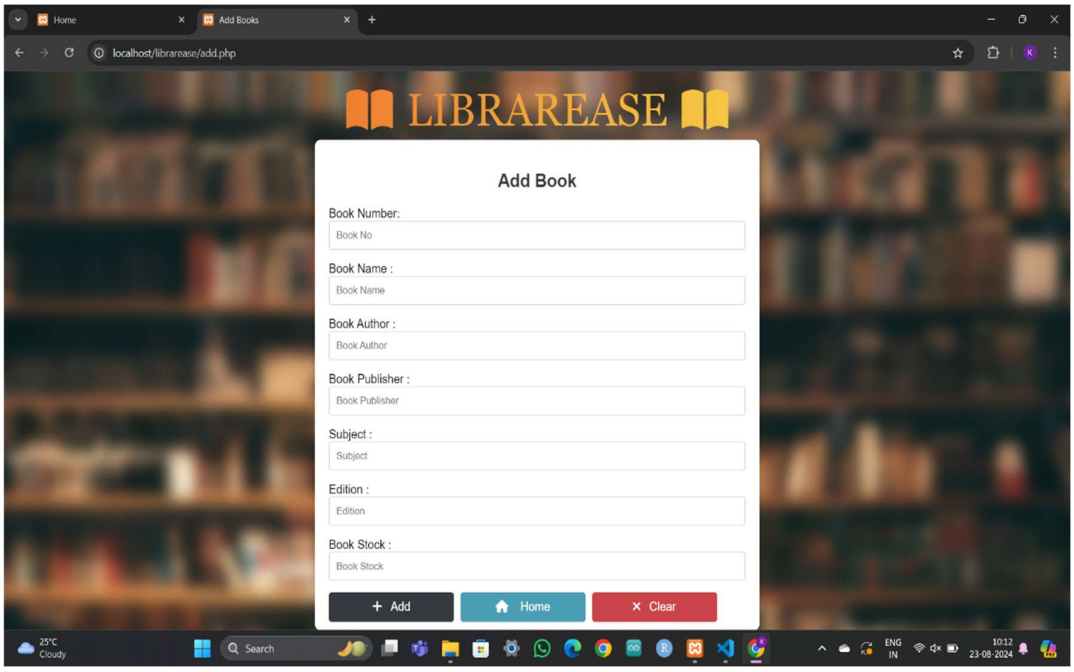

Fig 7.11 Add Book Page

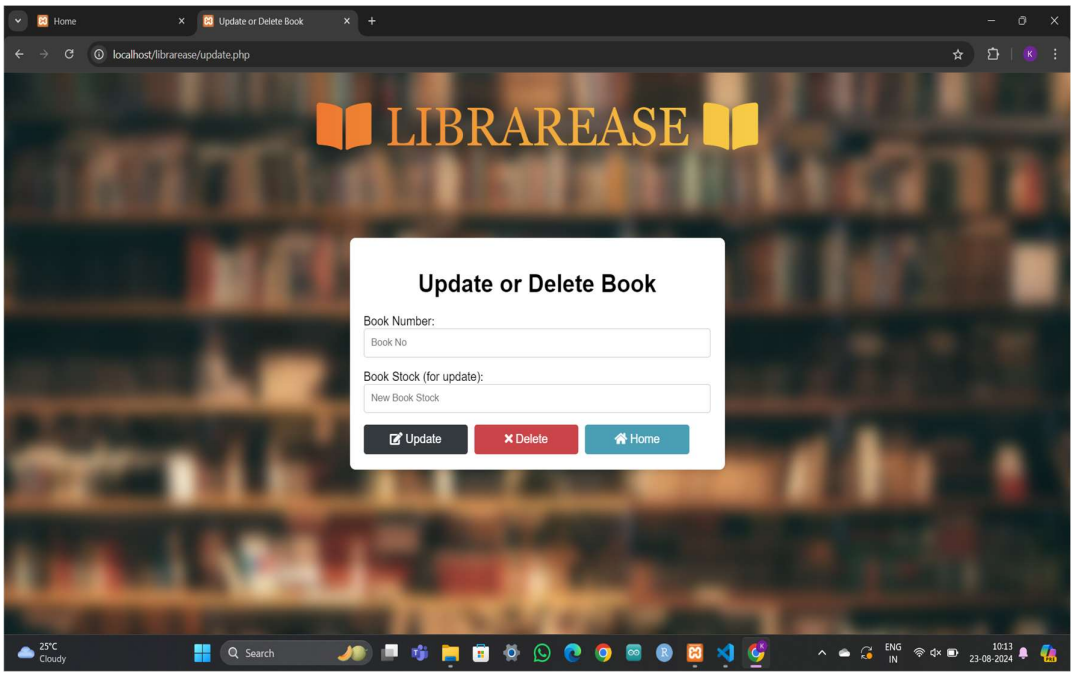

Fig 7.12 Update or Delete Book Page

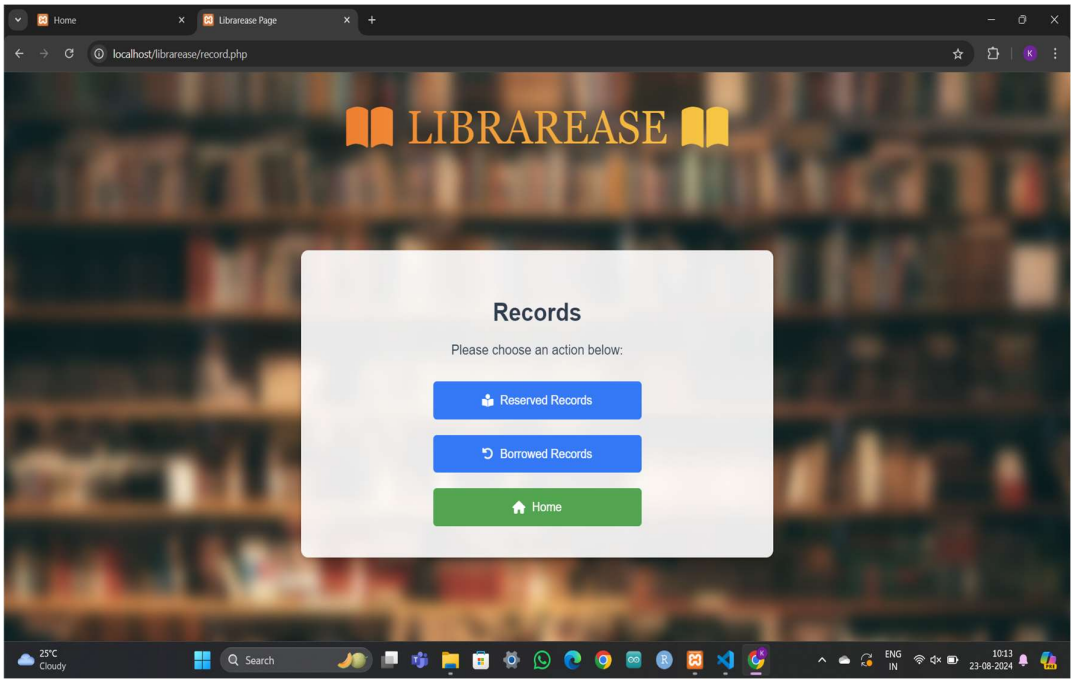

Fig 7.13 Records Page

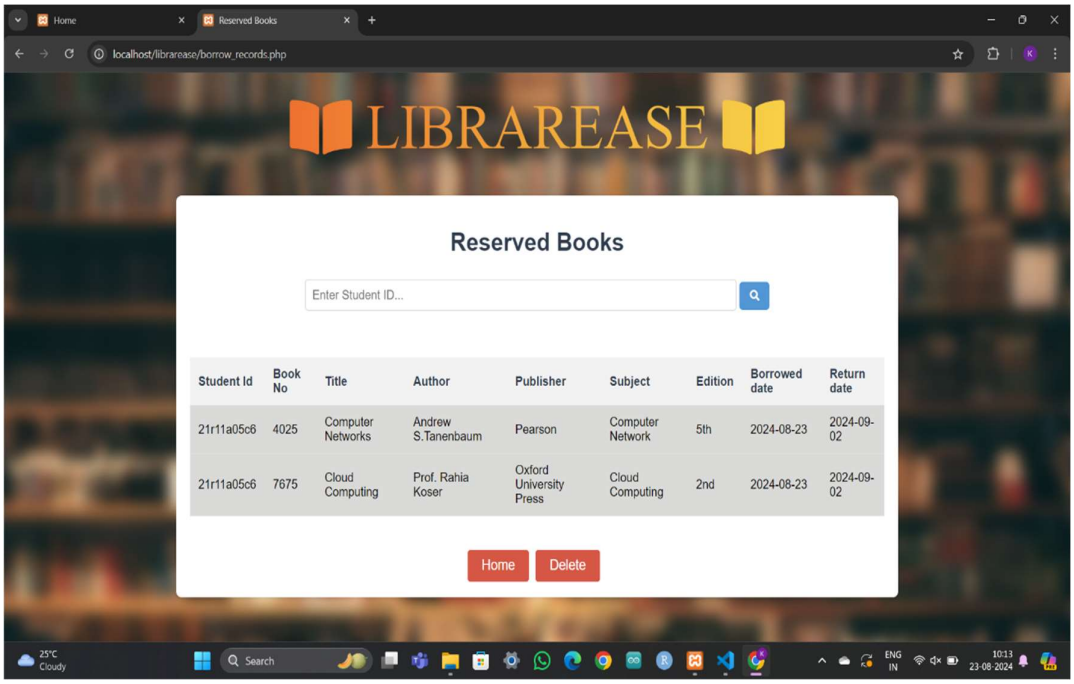

Fig 7.14 Managing Reserved Books Page

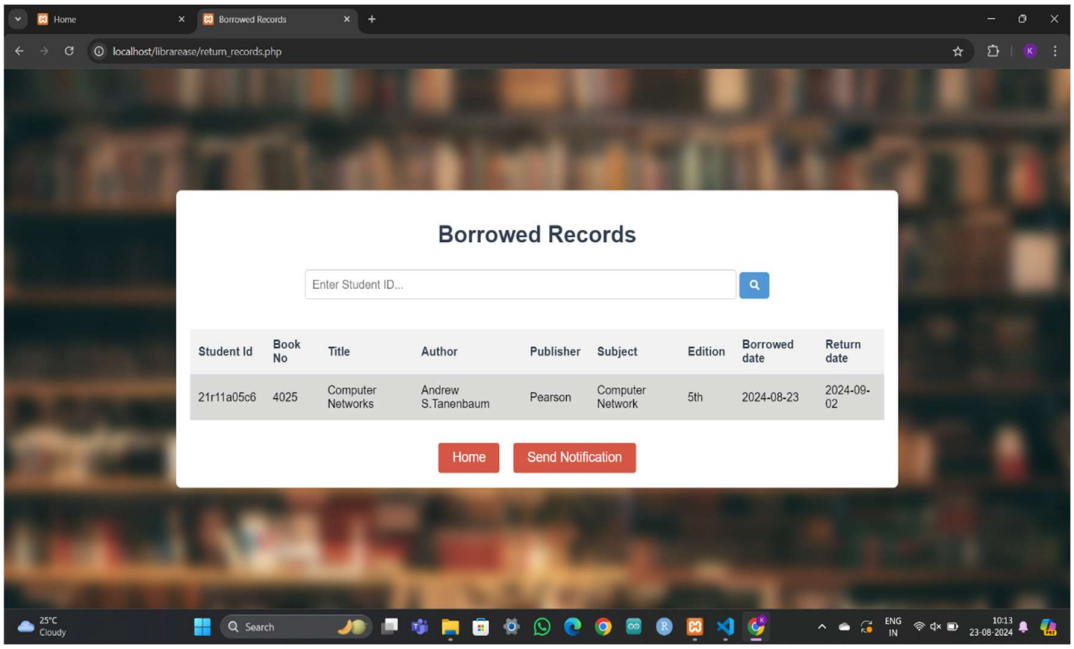

Fig 7.15 Managing Borrowed Records Page

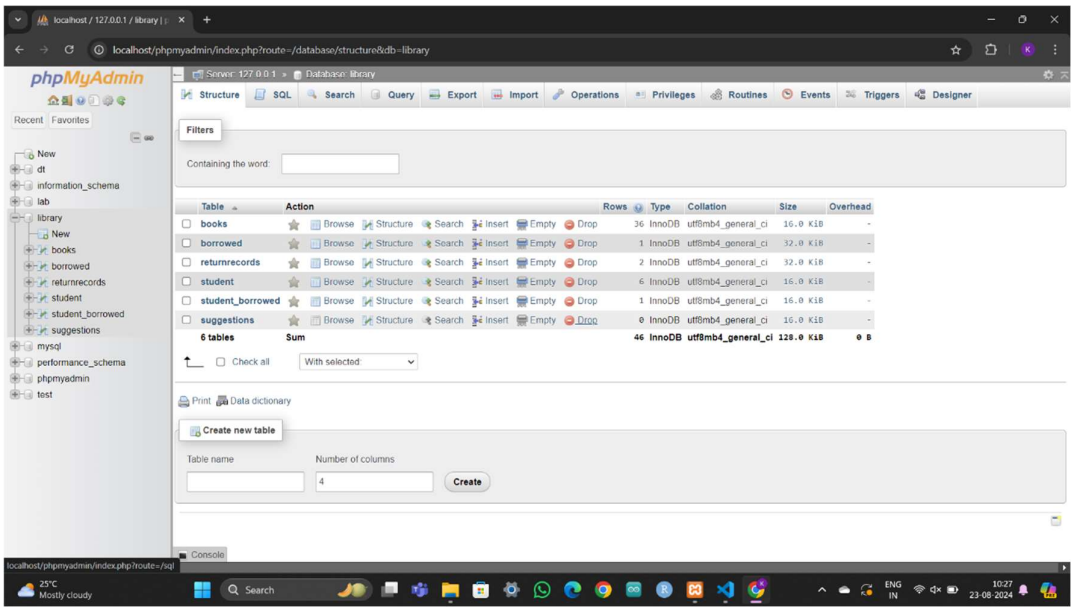

Fig 7.16 LIBRAREASE Database

8. CONCLUSION

8.1 CONCLUSION

LIBRAREASE transforms library management by providing an intuitive, web-based platform that benefits both students and librarians. For students, it offers a seamless experience for searching books, checking availability, and initiating online borrowing, reducing the hassle of traditional library visits. Librarians, on the other hand, gain efficient tools for managing inventory, updating book details, and tracking borrower records. The system's integration of modern web technologies and responsive design addresses common challenges in library management, ensuring accurate, real-time information and minimizing manual errors. By automating key processes and improving accessibility, LIBRAREASE streamlines library operations and enhances user satisfaction, ultimately leading to a more efficient and user-friendly library environment.

8.2 FURTHER ENHANCEMENTS

Here are some potential enhancements for the LIBRAREASE system:

1. Mobile Application Integration: Develop a mobile app version of LIBRAREASE to allow users to access the library system on smartphones and tablets, increasing convenience and accessibility.
2. Personalized Recommendations: Implement an algorithm that suggests books based on users' borrowing history and preferences, enhancing user engagement and discovery of new materials.
3. Reservation System: Add functionality for users to reserve books that are currently checked out, with notifications when the book becomes available.
4. Advanced Search Filters: Introduce more granular search options, such as genre, author, publication year, and user ratings, to help users find books more efficiently.
5. User Reviews and Ratings: Allow users to leave reviews and ratings for books, providing additional insights and helping others make informed decisions.
6. Integration with External Databases: Link with national or international library databases to expand access to a broader range of books and resources.

9. BIBLIOGRAPHY

9.1 REFERENCES

- Bretthauer, D. (2001) gives overview of open source software and describes open source solution for libraries at that time.

- 2002, Bretthauer, D. (2002) presents actual status and updates on open source software for libraries. Catherine, E. (2002) provides an overview of present state of ILS development. Breeding, M.

- (2002) provides the information about Koha, Learning Access ILS, and Avanti Micro LCS Integrated Library system.

- Boss, R. W. (2005), in his article provides criteria and on the basis this criteria he has evaluated 12 open source library management systems,

- Breeding, M. (2007), in his article, provides up-to-date information about Koha Evergreen and learning access ILS, integrated library system. The author gives comprehensive information about latest developments in software since 2002.

- DeVoe, K. (2007) provides a brief overview of nine open source integrated library.

- IAEME 40 Sunil M. V. and Harinarayana, N. S. (2011) has presented the requirement of Indian college libraries in integrated library system and evaluated nine open source library management softwares,

- Salve, A., Lihitkar, S. R., and Lihitkar, R. (2012) provide the information on the general and specific features of content management system and digital library software and 13 integrated library management software.

10. APPENDICES

Appendix A: System Requirements

Software Requirements

- Web Server: XAMPP (version [specific version])
- Database: MySQL (version [specific version])
- Programming Languages: PHP, JavaScript
- IDE/Text Editor: VSCode (version [specific version])

Hardware Requirements

- Computer: Minimum [specify CPU, RAM, and storage requirements]
- Network: [Specify network requirements, if applicable]

Appendix B: Installation Guide

Step-by-Step Installation

1. Download and Install XAMPP
    - [Link to XAMPP download]
    - Installation instructions
2. Set Up the Development Environment
    - How to configure XAMPP and start the Apache and MySQL services
3. Download and Configure LIBRAREASE
    - Steps to clone/download LIBRAREASE from [repository link]
    - Configuration settings (e.g., database connection details)
4. Accessing the Application
    - Instructions on accessing LIBRAREASE via a web browser (e.g., localhost/libraerase)

Appendix C: Database Schema

Tables and Fields

- **Students Table**
    - Columns: student_id, name, email, phone etc.
- **Books Table**
    - Columns: book_id, title, author, publisher, availability_status, etc.
- **Reserved Books Table**
    - Columns: reservation_id, student_id, book_id, reservation_date, status, etc.
- **Return Books Table**
    - Columns: return_id, loan_id, return_date, condition, late_fee, etc.
- **Suggestions Table**
    - Columns: suggestion_id, student_id, book_title, author, suggestion_date, status, etc.

Appendix D: User Manual

User Roles and Permissions

- Student
    - Access rights and functionalities
- Librarian
    - Access rights and functionalities

Common Tasks

- Searching for Books
- Checking Out Books
- Returning Books
- Suggestions

Appendix E: Troubleshooting

Common Issues

- Error: “Database Connection Failed”
    - Solution: Check database configuration in config.php
- Error: “Page Not Found”
    - Solution: Ensure URL is correct and files are in the correct directory

Appendix F: Glossary

- PHP: A server-side scripting language designed for web development.
- MySQL: A relational database management system.
- XAMPP: A free and open-source cross-platform web server solution stack package.