

# LIBRARY MANAGEMENT SYSTEM

Database Systems - CSCI 6622

May 4, 2021



**Team:** INFOSEC

**Professor:** DR. Reza Sadeghi

**Working Group:** Anuj Rane - arane1@unh.newhaven.edu

Kunal Dhanaitkar - kdhan1@unh.newhaven.edu

Ritesh Kumar Parmar - rparm1@unh.newhaven.edu

Udaypratap Prahlad Rathore - urath1@unh.newhaven.edu

# Contents

<b>1 Acknowledgment</b>	<b>4</b>
<b>2 List of Figures</b>	<b>5</b>
<b>3 Introduction</b>	<b>6</b>
3.1 Vision Statement . . . . .	6
3.2 Project Objectives . . . . .	6
3.3 Project Scope . . . . .	7
3.4 Outline of Approach and Methodology Used . . . . .	7
<b>4 Software Development Life Cycle</b>	<b>8</b>
4.1 Information Gathering . . . . .	8
4.2 Planning . . . . .	9
4.2.1 Logical Design . . . . .	9
4.2.2 Physical Design . . . . .	9
4.2.3 Interface Design . . . . .	9
4.3 Architectural Design . . . . .	10
4.3.1 Entity-Relationship Diagram . . . . .	10
4.4 PHASE 1 - Database Development . . . . .	13
4.4.1 Creating Database . . . . .	13
4.4.2 Using the Created Database . . . . .	13
4.4.3 Creating Table for Database . . . . .	14
4.4.4 Inserting data into the Database . . . . .	15
4.5 PHASE 2 - Application Development . . . . .	16
4.5.1 Creating Graphical User Interface . . . . .	16
4.5.2 Connecting GUI with Database . . . . .	18
4.5.3 Creating CRUD operations . . . . .	18
4.6 Testing . . . . .	25
4.6.1 CRUD Operations Test . . . . .	25
4.6.2 Login Test . . . . .	27

<b>5 Deployment</b>	<b>29</b>
<b>6 Future Scope of Project</b>	<b>30</b>
<b>7 Conclusion</b>	<b>31</b>
<b>8 Software Used</b>	<b>32</b>
<b>9 References</b>	<b>33</b>
<b>10 Project Github Link</b>	<b>34</b>

## 1 Acknowledgment

We are grateful to have this opportunity of working on this Database System Project. We would like to express our sincere gratitude to Dr. Reza Sadeghi for his guidance and encouragement in finishing this project and for teaching us in this course.

This project could not have been possible without the effort and cooperation from our group members: Ritesh, Udaypratap, Anuj and Kunal. Finally, we would like to express our gratitude to our friends and respondents for the support and willingness to spend some time with us to fill in the questionnaires.

## 2 List of Figures

1. ER Diagram
2. GUI
3. Add Books
4. Insert Success
5. Update Books
6. Book Update Success
7. Delete Books
8. Books Delete Success
9. Books Search
10. Books Search Success
11. Show All Books
12. Insert Test
13. Login Screen
14. Empty Login fields
15. Invalid Username or Password connection

### 3 Introduction

This report describes the project Library Management System that was developed to manage the daily book transaction, manage the members and the book records efficiently. The Librarian is able to manage the students module, the book maintenance and transaction module. Besides that, Library Management System is designed to keep tract of different books, services and users in a library. It allows users to create account and then handles all their activity by authenticating them using their userID and password. It also allows users to rent or buy books.

Maintenance of all this information manually is a very complex task. It is very difficult to organize manually. Owing to the advancement of technology, organization of the Library Management System has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. This computerization of library helps in many instances of its maintenance. It reduces the workload of management as most of the manual work done is reduced.

#### 3.1 Vision Statement

Our vision is to create a Library Management System, to keep a better track for all the Books, Employees, and the Students in the Library. We are committed to providing a robust and secure database system which will provide help in monitoring and maintaining the data and daily transactions properly and easily.

#### 3.2 Project Objectives

The aim of this project is to develop a system that can handle and manage the activities involved in a library in an efficient and reliable way. Less managing personnel and easy searching availability and user profile managing are major goals in this project. Libraries are essential in a process of giving citizens access to knowledge. In digital times they are needed more than ever before. The project objectives are as follows:

1. Develop a system that can replace the manual library managing system.

2. To eliminate the paper-work in Library.
3. Develop a database which stores user details and book details.
4. To record every transaction in computerized system so that problem such as record file missing does not happen.
5. Facility of Administrator login through which the admin can monitor the whole system.
6. Design a user friendly graphical user interface which suits the user.

### **3.3 Project Scope**

Under the proposed scheme, we believe that each user will obtain an identification login that maybe used to issue a library book, maintain cart, etc. Moreover it also allows the Admin and staff to record various transactions like like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

### **3.4 Outline of Approach and Methodology Used**

The software used to develop the back-end for Library Management System is MySql Work-bench, which provides an environment to develop and store the required database. The software package used to develop the front-end of the project is Python. The graphical user interface is implemented using the Python library package 'tkinter'.

## 4 Software Development Life Cycle

It is important to have an SDLC in place as it helps to transform the idea of a project into a functional and completely operational structure. In addition to covering the technical aspects of system development, SDLC helps with process development, change management and user experience. The SDLC has different phases, and all of them are discussed in detail in the following sections.

### 4.1 Information Gathering

Requirement Gathering is also known as the Discovery phase, which is a process in which we understand and identify the project's technical requirements and proceed with a well-defined plan. During this phase, the group had a few virtual meetings to analyze and understand what the expectation of this project is.

The process of Database design involves the collection of a significant amount of information. Gathering preliminary information for the project – such as deciding what data entities are going to be or the database system that is going to be used to store the data. The following information was gathered to implement this project:

1. We first determined the purpose of our system.
2. Then outline the data entities and attributes related to the data. Determined the relationship between the entities and finally designed an Entity-Relationship Diagram.
3. Selecting the right Database system is quite challenging and after a few comprehensive group meetings, we came up with a decision of using MySQL as our database system for this project. A few of the benefits we found for using MySQL were – Its High Availability, Scalability, Flexibility, and platform independence.
4. Lastly, we implemented a Graphical User Interface (GUI) over the MySQL database. A GUI is an application that is developed using a Programming Language, consists of tools and menus which are adapted for the user. They can make new entries, search for a specific entry in the data and manipulate the data without using the query language itself.

## 4.2 Planning

The Planning Phase for a database project creates a foundation for the final database. The Planning Phase is similar to developing architectural plans for constructing a building and helps ensure that the database project goes smoothly, preventing scope issues and errors while saving time and expense.

### 4.2.1 Logical Design

During the part of Logical Design, a conceptual model is created based on the needs assessment performed in stage one. A conceptual model is typically an entity-relationship (ER) diagram that shows the tables, fields, primary keys, and foreign keys of the database, and how tables are related (linked) to one another.

The tables sketched in the ER diagram are then normalized. The normalization process resolves any problems associated with the database design, so that data can be accessed quickly and efficiently.

### 4.2.2 Physical Design

During this phase, we implement the database design. Here, a DBMS (Database Management System) must be chosen to use. After a thorough discussion between the group, MySQL was considered as the intended database. SQL (Structured Query Language) clauses are written to help in creating the database. Also, the indexes and the integrity constraints (rules) are defined in this phase. And finally, the data is added, and the database can finally be tested.

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications. One of the reasons MySQL is the world's most popular open-source database is that it provides comprehensive support for every application development need.

### 4.2.3 Interface Design

A database management system stores data and response to queries using a query language, such as SQL. A GUI (Graphical User interface) provides a way to query data without having

to use the query language. A graphical user interface is an application that has buttons, windows, and lots of other widgets that the user can use to interact with the application.

Following the discussion in the group, Python was decided to be used for the implementation of the graphical user interface over the MySQL database. The python language is one of the most accessible programming languages available because it has simplified syntax and not complicated, which gives more emphasis on natural language. Due to its ease of learning and usage, python codes can be easily written and executed much faster than other programming languages.

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most used method. Tkinter is the standard GUI library shipped with Python. It provides a powerful object-oriented interface to the Tk GUI toolkit. Python with tkinter is the fastest and easiest way to create the GUI applications. Importing tkinter is same as importing any other module in the Python code. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

## 4.3 Architectural Design

Data Architecture design is important for creating a vision of interactions occurring between data systems and it helps in design, development, implementation, and maintenance of a database. In order to implement this, an Entity-Relationship Diagram was designed. Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design.

### 4.3.1 Entity-Relationship Diagram

Library Management System is designed to keep track of different books based upon their attributes. The ER Diagram shows the workflow of Library Management System consisting of different entities which can be identified by their attributes and their relationship. The ER diagram consists of different entities such as Student, Books, Admin, Staff, Checkout

Cart, Payment, and Authentication system and the description for each of these entities is described below.

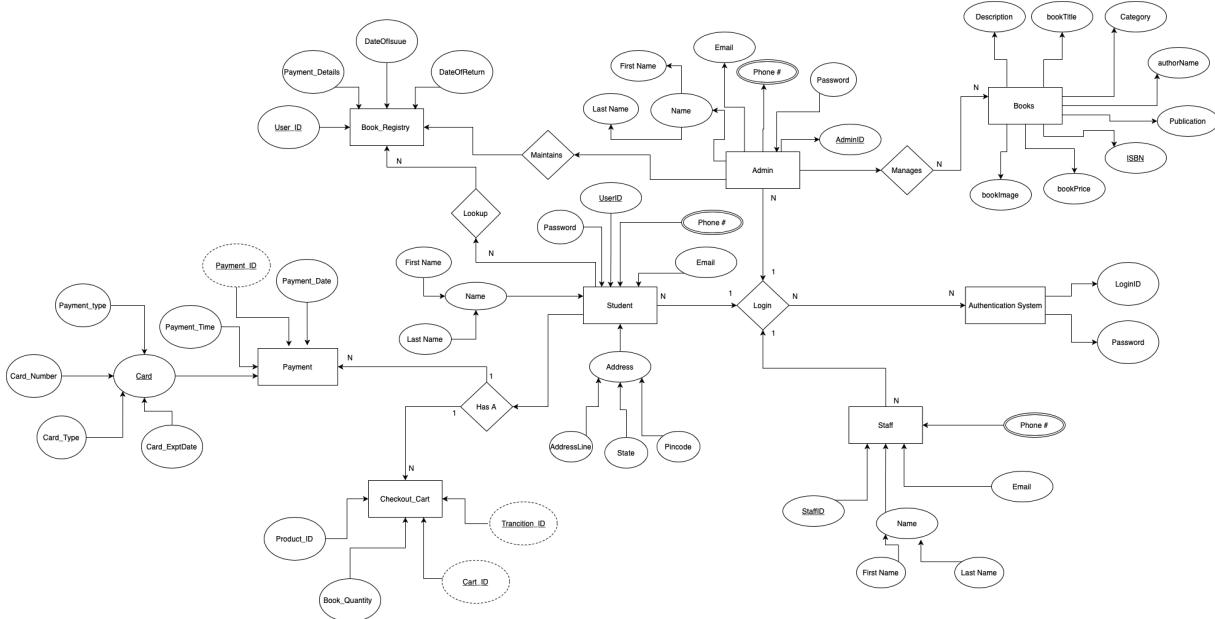


Fig 1: ER Diagram

## Student

Student entity consists of different attributes which is used to identify the student and its relationship with other entities. It consists of attributes such as UserID which is the primary key to uniquely identify the student. The Student entity is further characterized by using different attributes such as First Name, Last Name, Email, Phone, Address, State, Pin code. Student can login using authentication system with valid loginId and password. Students can further lookup for the book registry and can make payment for the selected book.

## Books

Books can be identified by using their ISBN number which is the primary key which helps to uniquely identify each book. Books are characterized by using different attributes such Description, bookTitle, Category, authorName, Publication, bookPrice, bookimage.

## Book Registry

Book Registry is used to keep track of book using different attributes such as DateofIssue which indicates the date issued to the user for a, DateofReturn which indicates the date the

book needs to be returned. Each book is associated with its unique UserId and payment details of that user.

## **Payment**

Payment entity is used to hold payment records. PaymentID is used as primary key to uniquely identify each payment. The Payment entity consists of different attributes such as Payment Date and Time which specifies the date/time on which the payment was made. Payment Type which holds the payment type details. Card Number which holds card number details. CardType which specifies the type of card used for the payment. CardExptDate which hold the date on which the card gets expired.

## **Admin**

Admin entity plays an important role in the overall management and maintenance of the book and book registry. Admin can be uniquely identified by their primary key as AdminID and has been characterized by different attributes such as First Name, Last Name, Email and Phone number. Admin with valid login credentials can be granted access to the system. Admin maintains the book registry as well as manage the books.

## **Checkout Cart**

Checkout Cart is characterized by different attributes such as CartId, TranactionId, ProductId and Book Quantity. Checkout Cart can be Uniquely identified by its CartId and TransactionId and is used to manage the book quantity with its related product Id.

## **Staff**

This entity holds all staff records and can be uniquely identified by the StaffID. The Staff entity is characterized by different attributes such as First Name, Last Name, Email, Phone.

## **Authentication System**

Authentication System is used to validate users as per their login details such that each user will have LoginId and Password. So, upon providing valid LoginId and Password the user gets authenticated and given access to the system.

## 4.4 PHASE 1 - Database Development

After the architectural designed was developed which was based upon our Entity-Relationship diagram, we started working on creation of our database by importing into MYSQL using the MYSQL-workbench. Creation of database involved a four-step process which includes creating database, using the created database, creating tables for the database, and finally pushing data into the database.

### 4.4.1 Creating Database

This is the initial step towards the creation of database was implemented using the keyword “CREATE DATABASE” followed by the name of the desired database name ending with a semicolon. We decided to name our database as “libpos”. Using the keyword “CREATE DATABASE” followed by libpos ending with a semicolon we created our database. The SQL statement which we used to create our database “libpos” is shown below:

“CREATE DATABASE DATABASENAME;”

After executing the statement below, our database was created by the name libpos:

“CREATE database libpos;”

### 4.4.2 Using the Created Database

After creating the database, second step involved in the process was using the created database. The USE statement tells MYSQL to use the database followed by the name of the database. So, we followed the process of using the “USE” keyword followed by “libpos” ending by a semicolon. The SQL statement is shown below to use the created database:

“USE DATABASENAME;”

Executing the statement below, ensures that all the actions will be performed on the selected database:

USE libpos;"

#### 4.4.3 Creating Table for Database

A table for the database can be created using the details such as name of the table, name of the fields, field definitions. The syntax which is used to create a table is using the keyword “CREATE TABLE” which is followed by the name of the table you want to create, followed by parentheses “()” and within the parentheses you can define the column name with their associated data-types. The field attribute “NOT NULL” can be used when you don’t want the particular field to be null and the keyword “PRIMARY KEY” can be used to define a particular column as a primary key. You can use multiple columns which should be separated using “,” between them and using “;” at the end of the parentheses. The SQL statement to create a table for database is shown below:

“CREATE TABLE TABLENAME (COLUMN NAME, COLUMN TYPE);”

We created multiple tables by the name Admin, Student, Staff, Payment, Book, Check-out\_Cart, book\_registry. Below is one of the code snippet which shows how we create a table by the name book with its associated column name and column type.

```
”CREATE TABLE Book (
Description varchar ( 50 ) NOT NULL,
BookTitle varchar ( 25 ) NOT NULL,
Category varchar ( 10 ) NOT NULL,
AuthorName varchar ( 20 ) NOT NULL,
Publication varchar ( 20 ) NOT NULL,
BookPrice int NOT NULL,
ISBN varchar ( 20 ) PRIMARY KEY
) ; ”
```

In the above code snippet, we create a table by the name book which has multiple columns having column names such as Description, BookTitle, Category, AuthorName, Publication, BookPrice, ISBN and their column type as varchar where ISBN was used as a primary key and rest of the columns as “NOT NULL”.

#### 4.4.4 Inserting data into the Database

This was the final step of our database creation process where we pushed the data into the tables using keywords “INSERT INTO” followed by the name of the table having two set of parentheses separated by keyword “VALUES”. The first set of parentheses is used to hold the column names whereas the second of parentheses holds the associated data to the column name defined into first set of parentheses that needs to be pushed into the table ending with “;” at the end of the parentheses. The SQL statement to push data into the database is shown below.

```
INSERT INTO TABLENAME (field1, field2, . . . , field N) VALUES (value1, value2, . . . , value N)
```

```
Book table INSERT INTO Book (Description, BookTitle , Category , AuthorName , Publication , BookPrice , ISBN ) VALUES ( ' This book cover basics of calculus ' , ' Calculus for dummies ' , ' Math ' , ' Mark Ryan ' , ' Dummies ' , 12 , 90002202 );
```

In the above code snippet, we inserted multiple column names such as Description, BookTitle, Category, AuthorName, Publication, BookPrice, ISBN with their associated data as ‘This book cover basics of calculus’, ‘Calculus for dummies’, ‘Math’, ‘Mark Ryan’, ‘Dummies’, 12, 90002202

## 4.5 PHASE 2 - Application Development

After the process of creation of database, we started with the application development phase of our project where we implemented a graphical user interface which involved several important operations such as insert, delete, update, search, clear. The entire process of implementing graphical user interface and configuring the curd operation along with connecting the GUI to the database is explained in detailed in 3 different sections.

### 4.5.1 Creating Graphical User Interface

The first section consists of creating a graphical user interface for our library management system using the python library tkinter which involved several steps such as creating label, creating input box, and creating a button.

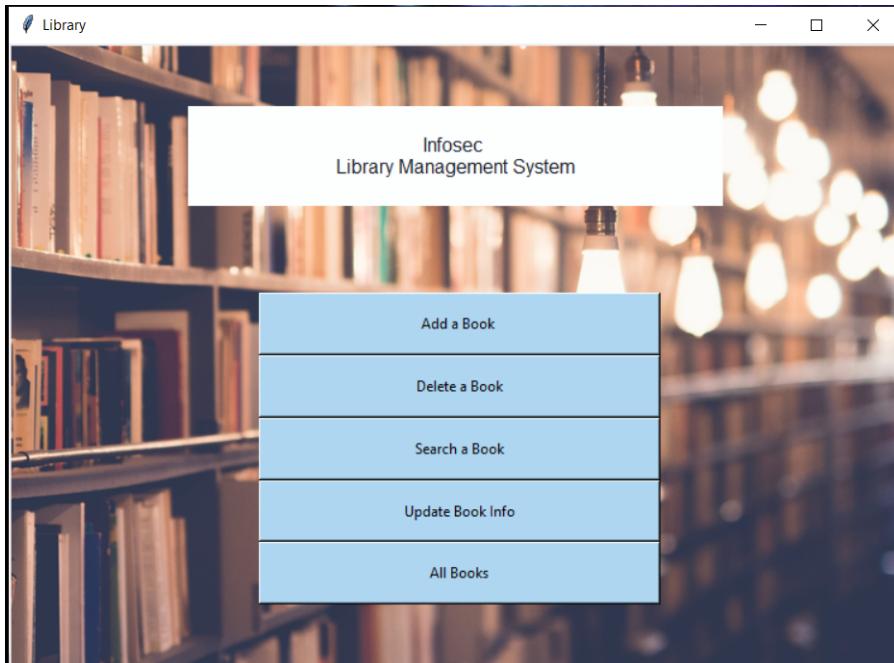


Fig 2: GUI

#### Creating Label

The code snippet shown below shows one of the labels that is implemented as part of our LMS.

```
bookDescription=Label(root, text="Description", width=20, height=2,
```

```
bg="a2cff").grid(row=0, column=0, sticky=E)
```

The book table constitutes of seven attributes such as Description, Title, Category, Author Name, Publication, Price, and ISBN so seven labels were created for the mentioned attributes. The above code snippet was used to design label for description input box. The 'root' defines that it will be using the main window, whereas the 'text' attribute specifies the text that will be displayed on the screen. The attributes 'width' and 'height' are used for the size of the label and the 'bg' attribute specifies the background color for the label. The '.grid()' having rows and column is used to position the label on the screen. The same process was followed for the other labels created for GUI of our LMS project.

### **Creating input box**

The code snippet below shows one of the input boxes that is implemented as the part of our LMS.

```
e1 = Entry(root, width=30, borderwidth=2)  
e1.grid(row=0, column=1)
```

The input box was created using "Entry()" function whereas 'root' attribute was to specify that it will in the main window. The attributes 'width' and 'height' are used for the size of the label. The '.grid()' having rows and column is used to position the label on the screen. So, total of 7 such input box was created for the GUI.

### **Creating Button**

The code snippet below shows one of the buttons that is implemented as the part of our LMS.

```
button1 = Button(root, text="Insert", width=10, height=2,  
command=insertData).grid(row=0, column=3)
```

The above code snippet is used for creating buttons for the curd operations for the GUI. The button was created using the Button () function. The 'root' attribute is used to place

it in the main window, ‘text’ is writing the text in the button. The attributes ‘width’ and ‘height’ are used for the size of the label. The ‘.grid()’ having rows and column is used to position the label on the screen. A total of 6 buttons were created for CURD operations such as insert, update, delete, show record, show all, and clear.

#### **4.5.2 Connecting GUI with Database**

The second section consists of connecting the GUI to our database which was done using a python package named ‘mysql.connector’ which was imported to allow the GUI to connect to our database. Also, function named as ‘mysql.connector.connect()’ was used to connect our GUI to the database by providing it with a ‘host’ name. The authentication process involved ‘username’ and ‘password’ to connect to our database which allowed mysql to authenticated us and to connect our database and GUI.

The code snippet below was used to connect the GUI with database.

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="root",  
    database="libposnew"  
)
```

#### **4.5.3 Creating CRUD operations**

The third section of our application development phase consisted of creating CRUD operations which were configured for the button of our graphical user interface.

##### **Insert Operation**

The code snippet shown below was implemented for the insert operation.

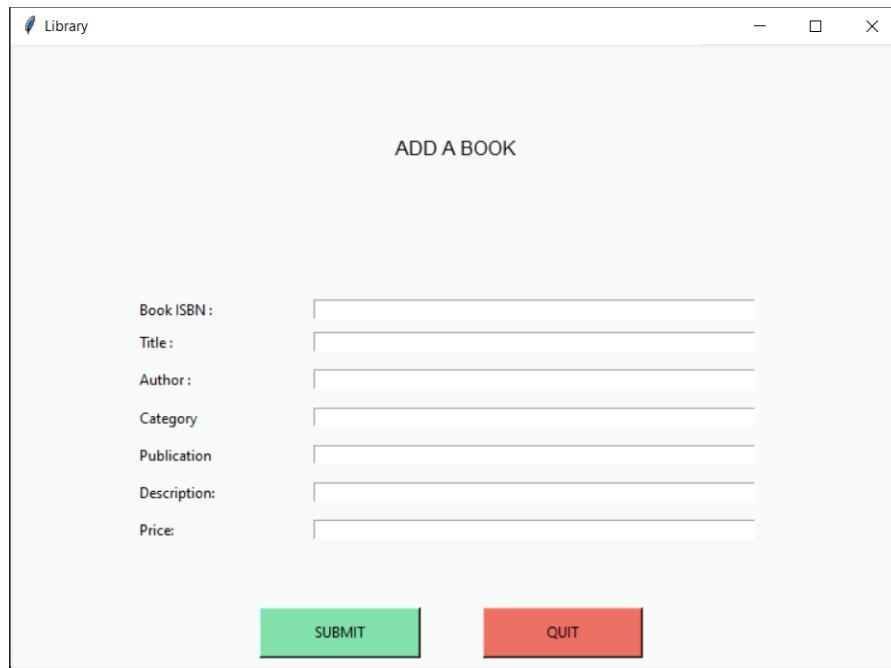


Fig 3: Add Books

The ‘insertData()’ function was created which fetches the data using the input box by the ‘.get()’ function which will pull the data from the front end and stores them in variables once the insert button is clicked. Then the if condition verifies if the input values are not null, else it will throw an error ‘Enter Valid Records’. The insert query pushes all the data into database and ‘Record Insert’ gets displayed as a success message on the screen.

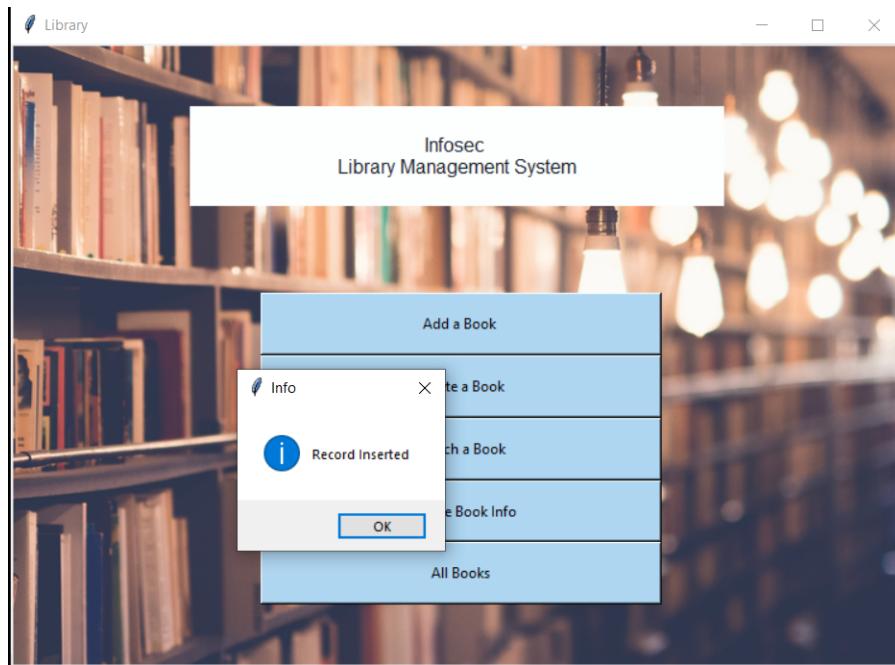


Fig 4: Insert Success

## Update Operation

The code snippet shown below was implemented for the update operation.

A screenshot of a Windows application window titled "Update Books". The window contains a search bar labeled "Search books to update using ISBN" with the value "232323" and a "Search" button. Below the search bar is a table with seven rows, each having a label on the left and an input field on the right. The rows are: Description (containing "testnew"), Title (containing "test"), Category (containing "test"), Author Name (containing "test"), Publication (containing "test"), Price (containing "22"), and ISBN (containing "232323"). At the bottom of the form are two buttons: "Submit" on the left and "CLEAR" and "QUIT" on the right.

Fig 5: Update Books

The 'updateData()' function was created which gets triggered on clicking the updateData

button. It fetches the 'ISBN' of the book entered by the user to update, and then runs a query to pull the entry from the database based on the ISBN provided once the update button is clicked such that it grabs all the updated input entered by the user, and runs an update query and commits all the updated values to the database after which it displays 'Record Updated' and if the ISBN does not match it displays an error 'Record Doesn't Exist'.

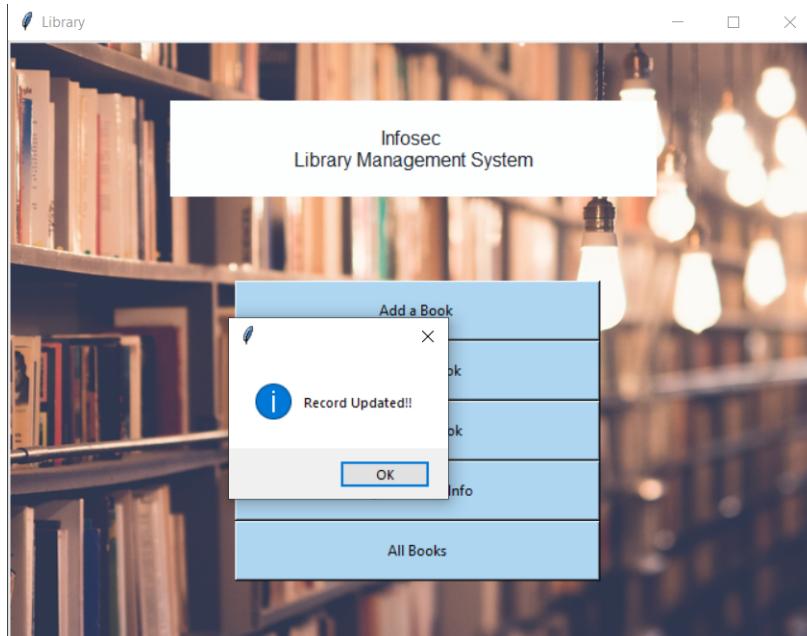


Fig 6: Books Update Success

### Delete Operation

The code snippet shown below was implemented for the delete operation

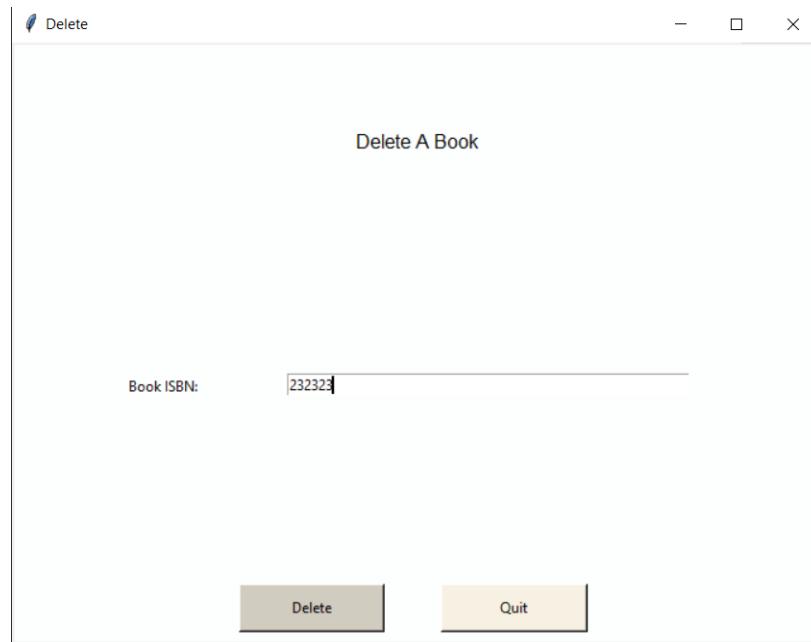


Fig 7: Delete Books

The ‘deleteData()’ function was created which fetches the ISBN from the user, and query for the entry based on the ISBN provided and then runs a delete query to drop the entry from the database and will display ‘Record delete’ message on the screen once the deleteData button is clicked.

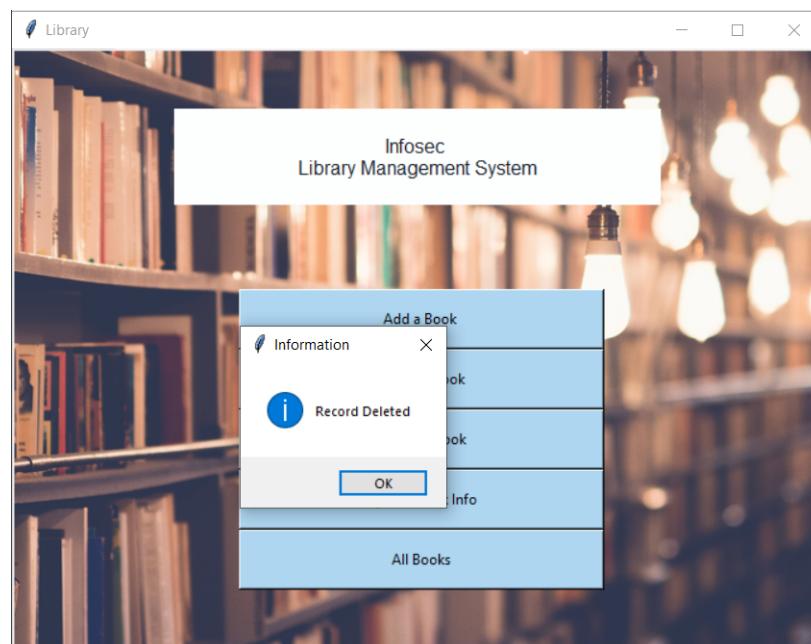


Fig 8: Books Delete Success

## Search Operation

The code snippet shown below was implemented for the search operation.

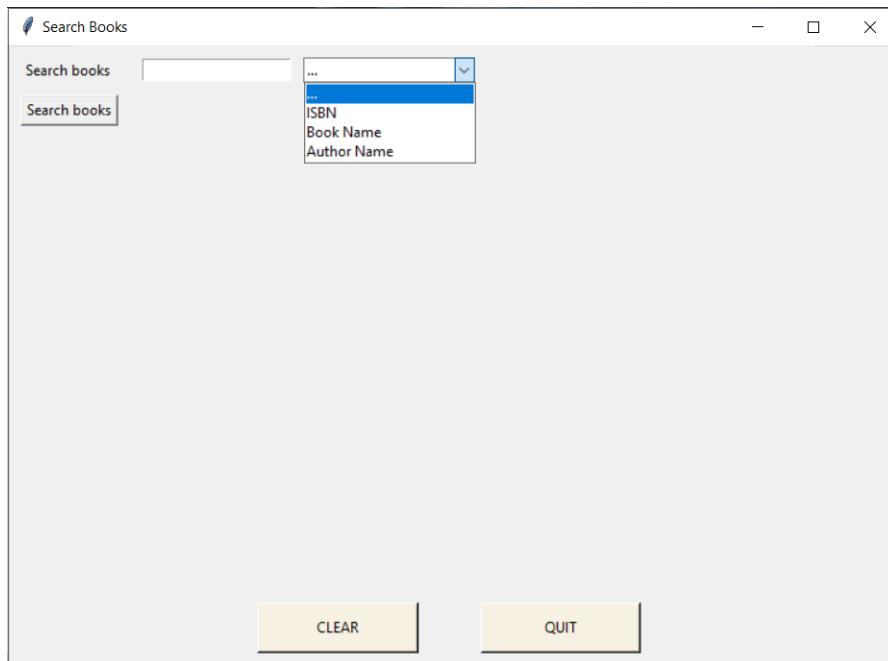


Fig 9: Books Search

The 'showRecord()' function was created which takes ISBN number of the book from the user as an input, and the queries for the data based on the ISBN provided and then displays the result in the input box if the data is found which gets triggered once the showRecord button is clicked.

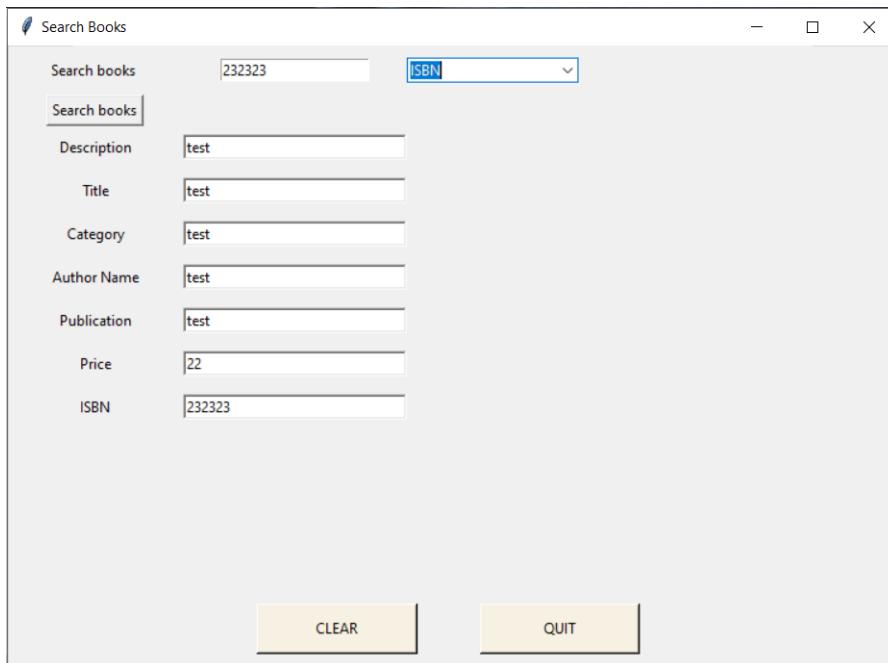


Fig 10: Search books success

## Show All Operation

The code snippet shown below was implemented for the show all operation



Description	Title	Category	AuthorName	Publication	Price	Isbn
testne	test	test	test	test	22	232323
This book covers basics	Calculus for dummies	Math	Mark Ryan	Dummies	12	90002202
This book covers basics	Cryptocurrency	Technology	Kiana Danial	Dummies	23	90003301
This book covers basics	SQL for dummies	Computer	Peter Weverka	Dummies	19	90004402
This book covers basics	PHP for dummies	Computer	Doug Lowe	Dummies	18	90004406
This book covers basics	Javascript for dummies	Computer	Doug Lowe	Dummies	17	90004408
This book covers basics	Java for dummies	Computer	Jack chu	Dummies	16	90004410
This book covers basics	Linux for dummies	Computer	Greg Harvey	Dummies	19	90004412
This book covers basics	Android for dummies	Computer	Sundar Pi	Dummies	18	90004414
This book covers basics	Web-DEV for dummies	Computer	Bhavik Nahar	Dummies	23	90004416

Fig 11: Show All Books

The ‘showAllRecord()’ function was created which queries for all the entry in the database and then displays all them in a tabular form in a new pop-up window once the Show All Record button is clicked

## Clear Data Operation

The ‘clearData()’ function was created which gets triggered on clicking the clear Data button which will clear all the data from the input.

## 4.6 Testing

The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional intervals is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing and module level testing are integrated and carried out. In this phase, we will test the Graphical User Interface for the Library Management System, CRUD Operations, User Dashboard, and any wrong input login.

### 4.6.1 CRUD Operations Test

#### Insert

We will input some records into the input boxes and click on the Insert Button to check whether the data is being pushed in the Database.

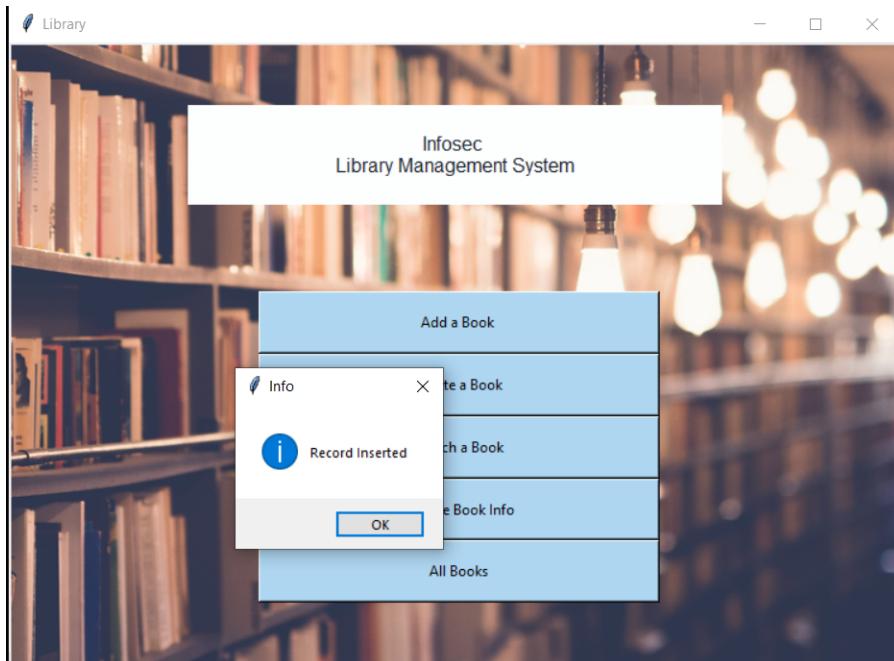


Fig 12: Insert Test

After clicking the Insert Button, the application provides a pop-up message to the user signaling that the data has been successfully recorded in the database.

## **Update**

The Application user interface allows the user to update existing data by clicking on the Update button. Enter the ISBN of the Book with the updated records to update the records for the Book.

## **Delete**

To delete an existing record from the Database, we enter the ISBN of the Book and hit the delete bottom. A message pop-up to indicate that the record has been deleted.

## **Search**

To look up a specific Book in the Library, the Application provides a Show Record Button. Simply type in the ISBN of the Book and hit the Show Record button to look up the Book.

## **Clear**

The Application also provides a Clear Button, which can be used to clear all the input data inside the input inboxes.

## **Show All Record**

To display all the Books available, the user can click the Show All Button and the Application will exhibit all the Books currently available in the Library.

#### 4.6.2 Login Test

In this section we will be testing the login feature for admins and students.

##### User Dashboard

Admin Dashboard: The Application provides a Login for the Admin to administrate all the transactions and records of the Library.

Student Dashboard: The Application also provides the Student a Login to view all the books currently available in the Library and search for a specific book.

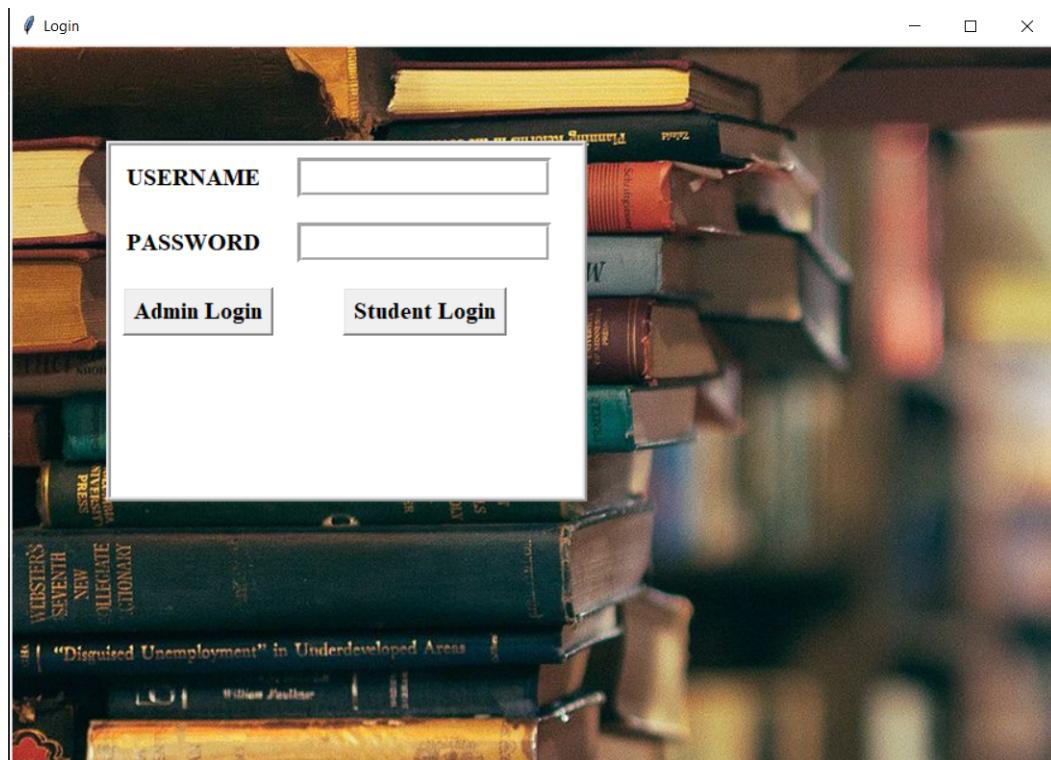


Fig 13: Login Screen

##### Empty Field Validation

The User Application validates empty fields verification if the user has left any field empty during Logging in. A pop-up message is displayed on the screen indicating that all the fields are required.

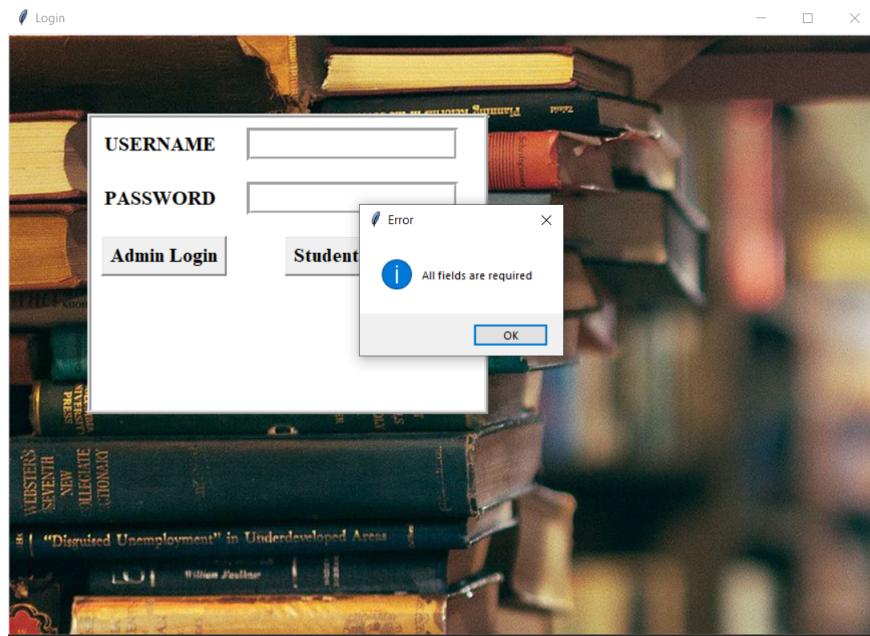


Fig 14: Empty Login Fields

### Login Verification

The user needs to provide a valid Login username and password to login into the Library Management System. If any of the fields are incorrect, it throws away an error.

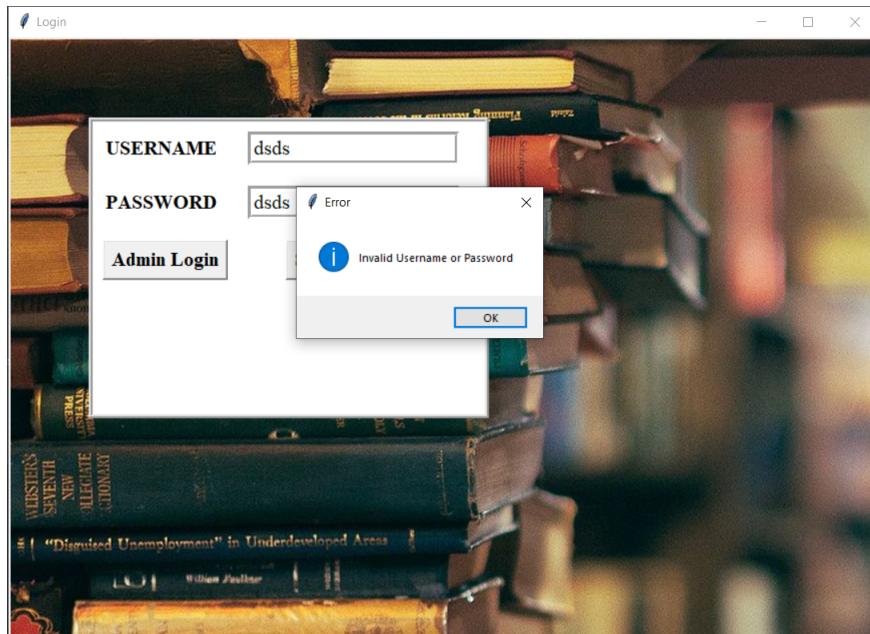


Fig 14: Invalid Username or Password

## 5 Deployment

The project was successfully developed for library management system which consisted of student portal and admin dashboard. The requirement of designing and developing a student portal was to provide student with the options to lookup for different book of their choice. A student can login to the system using login id and password. After successful login attempt student can look up for different books that are available. A student can look up for different types of books based upon different factors such as book-title, category, author name, publication, book-price. Each book has been assigned a unique ISBN number which helps to identify the book the student seeks to find. The requirement to designing and developing admin dashboard was fulfilled to provide admin a platform to manage different books based upon their attributes. The Admin can login to the system using their valid admin id and password. After successfully authenticating itself to the system the admin can keep the track of different books. The project for library management system was successfully designed and deployed by following all the necessary steps to meet the requirements of the project.

## 6 Future Scope of Project

A Library Management System was developed and implemented in this Database Project. We plan to add the following new features in this System:

1. Provide a Recommendation Page to the User Interface such that Students can view and reserve books related to their curriculum.
2. Facility to provide Students with Online Delivery for the Books.
3. Facility to cancel reservation for a book made earlier.
4. An interface to view and edit User Profile.
5. Provide methods for adjusting account setting and a mechanism to reset the password in case user forgets it.
6. Provide an interface for Staff to add or delete Books from the Library.

## 7 Conclusion

This Database Project will provide a computerized version of a library management system that will benefit the students as well as the staff of the library. It makes the entire process online where a student can search books, staff can generate reports and do book transactions. It also has a facility for student login where the student can log in and can see the status of books issued as well as request for a book. Combined with the front desk development tools and platform, it can realize the function of strong design and system implementation. It had been showed that library management information system's reliability and stability had been sure through the use of them, which was developed by using the database structure. It has strong practical significance and application value.

## 8 Software Used

1. Pycharm
2. MySQL Workbench
3. Draw.io
4. Overleaf
5. Microsoft PowerPoint

## 9 References

1. Python : <https://www.python.org/>
2. tkinter : <https://docs.python.org/3/library/tkinter.html>
3. geeksforgeeks : <https://www.geeksforgeeks.org/python-tkinter-tutorial/>
4. Pythonbasics : <https://pythonbasics.org/tkinter/>
5. MySQL workbench : <https://www.mysql.com/products/workbench/>
6. Pycharm : <https://www.jetbrains.com/pycharm/>
7. overleaf : <https://www.overleaf.com/>
8. Draw : <https://app.diagrams.net/>
9. PowerPoint : <https://office.live.com/start/powerpoint.aspx>

## 10 Project Github Link

1. Project GitHub Link : <https://github.com/rioparmar1996/thefirstavenger/tree/main/library-management>