**LIBRARY MANAGEMENT SYSTEM**

A Project Report

Submitted in partial fulfillment of The requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

# BY:

AATIF ZAKIR HUSSAIN PATEL

# ROLL NO : 312 SEAT NO : 1066637

Under the esteemed guidance of

**Prof. SAIMA SHAIKH**



# DEPARTMENT OF INFORMATION TECHNOLOGY

MAHARASHTRA COLLEGE OF ART, SCIENCE & COMMERCE

(affiliated to university of Mumbai)

MUMBAI 400008

MAHARASHTRA

YEAR 2023 - 2024

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

***(Note:All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

PNR **NO**:-…202101640189554 ROLL **NO**:- **312**

1. Name of the Student

PATEL AATIF ZAKIR HUSSAIN

1. Title of the project

LIBRARY MANAGEMENT SYSTEM

1. Name of the guide

Prof. SAIMA SHAIKH

1. Teaching experience of the guide

11 YEARS OF TEACHING EXPERIENCE

1. Is your first submission?

YES NO

Signature of student Signature of Guide

Date: ……………….. Date: ………………….

Signature of coordinator

Date: …………………..

MAHARASHTRA COLLEGE OF ART, SCIENCE & COMMERCE

( Affiliated to University of Mumbai)

MUMBAI MAHARASHTRA 400008 DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that project entitle, **“ LIBRARY MANAGEMENT SYSTEM”,** is bonafide work of **PATEL AATIF ZAKIR HUSSAIN** bearing Seat no **1066637** submitted inpartial fulfillment of the requirements for award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from UNIVESITY of MUMBAI

Internal Guide Coordinator

External Examiner

Date: College seal

# INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Topic** | **Page-No** |
| **I.** | **Preliminary Investigation** | 05 |
| (i) | Description of system | 06 |
| (ii) | Limitation of present system | 06 |
| (iii) | Proposed system and its advantages | 06 |
| (iv) | Scope of the system | 07 |
| (v) | Feasibility Study | 08 |
| (vi) | Stakeholders | 09 |
| (vii) | Gantt Chart | 10 |
| **II.** | **Justifications of the packages** | 11 |
| (i) | About the packages used | 12 |
| **III.** | **Operating Environment** | 16 |
| (i) | Hardware and software used | 17 |
| **IV.** | **System Analysis** | 18 |
| (i) | Fact Finding Techniques | 19 |
| (ii) | Event Tables | 20 |
| (iii) | Use Case Diagram | 21 |
| (iv) | Sequence Diagram | 25 |
| (v) | ERD Diagram | 27 |
| (vi) | Class Diagram | 28 |
| (vii) | Activity Diagram | 29 |
| (viii) | Object Diagram | 32 |
| (ix) | State Diagram | 33 |
| **V.** | **System Design** | 34 |
| (i) | Converting ERD to Tables | 35 |
| (ii) | Component Diagram | 38 |
| (iii) | Package Diagram | 41 |
| (iv) | Deployment Diagram | 42 |
| (v) | System Flow Chart | 43 |
| (vi) | Structure Chart Diagram | 44 |
| **VI.** | **System Coding** | 45 |
| (i) | Menu Tree / Sitemap | 46 |
| (ii) | List of tables with attributes and constraints | 47 |
| (iii) | Program Description with Naming Conventions | 50 |
| (iv) | Validations | 51 |
| (v) | Test Cases, Test Data and Test Results | 52 |
| **VII.** | **Screen Layouts** | 55 |

|  |  |  |
| --- | --- | --- |
| **VIII.** | **Program Code Listing** | 71 |
| **IX.** | **System Implementation / Uploading** | 95 |
| **X.** | **Future Enhancements** | 97 |
| **XI.** | **References and Bibliography** | 98 |

**DECLARATION**

I here by declare that the project entitled, **"Library Management System "**done at Place where the project is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

**ABSTRACT**

The project titled "Library Management System in Python with MySQL Database" is a comprehensive desktop application aimed at enhancing the efficiency of library resource management. Developed using

Python's Tkinter GUI library and integrated with a MySQL. database, the system empowers librarians to seamlessly handle tasks such as book registration, updates, deletions, and member management. The user-friendly interface ensures easy navigation and quick data entry, while the MySQL database ensures data integrity and facilitates easy retrieval.

Looking towards the future, the project has the potential for further expansion. Future scope includes implementing advanced search and filtering options for quicker access to information, extending management capabilities to fines,

returns, and due dates, incorporating reporting features for generating lists of borrowed and available books, and enhancing security through user authentication and role-based access.

The project aims to provide a robust solution to the challenges faced by libraries, offering a user-friendly interface, efficient data management, and a foundation for future enhancements in library resource administration.

## ACKNOWLEDGEMENT

I undersigned, have great pleasure in giving my sincere thanks to those who have contributed their valuable time in helping me to achieve the success in my project work. My heartfelt thanks to the principal of the College

**Prof, SIRAJUDDIN CHOUGLE** and the IT department of College for helping in the project with words of encouragement and has shown full confidence in our abilities.

I would to express my sincere thanks to Prof. Saima Shaikh head of IT and CS Department for her constant encouragement, which made this project a success. I am indebted and thankful to our project Guide, Saima Shaikh to whom I own piece of knowledge for her valuable and timely guidance, co-operation encouragement and time spent for this project work I would also like to thanks our IT staff for providing us sufficient information, which helped us to complete our project successfully. My sincere thanks to library staff for extending their help and giving me all books for reference in very short span of time.

I also thanks my PARENT and all my Family Member for their continues Support, without their support this project would not be possible.

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

# INTRODUCTION

The "Library Management System in Python with MySQL Database" project represents a significant leap forward in the realm of library management. This web-based application is designed to empower librarians and administrators with the tools they need to efficiently oversee library resources. By harnessing the capabilities of Python's Tkinter GUI library and seamlessly integrating with a MySQL database, this system aims to simplify tasks such as book registration, updates, deletions, and member management.

This project's primary objective is to offer a feature-rich, user-friendly solution that transcends the limitations of traditional manual systems. With the ability to add, update, and delete data effortlessly. along with a convenient data reset option, the system not only enhances productivity but also minimizes the risk of errors in library records. Moreover, the project capitalizes on the advantages of digitalization, providing data persistence and saving valuable time by automating labor-intensive processes

**Module:** The Library Management System has four main module

* Database Interaction Module
* Data Retrieval Module
* User Authentication Module
* Bill Management Module

**Objective:**

* Efficient Management: Create a system that streamlines the management of library resources, including books and member information, to enhance the overall efficiency of library operations.
* User-friendly Interface: Design an intuitive and user-friendly graphical userinterface (GUI) using Python's Tkinter library to ensure ease of use for librarians and staff members.
* Data Integrity: Utilize a MySQL database to ensure the integrity of library data, making it easy to store, retrieve, and manage information accurately Less chances of getting confused.
* Time-saving Automation. Automate manual tasks such as book registration, updates, and deletions, reducing the time and effort required for administrative work.

## Scope:

* Book Management. The system allows librarians to add, update, and delete book records efficiently.
* Member Management: Librarians can manage member information including registration and updates.
* User Interface: A user-friendly Tkinter-based GUI facilitates ease of use.
* MySQL Integration: The system integrates with a MySQL database to store and manage library data, ensuring data integrity and easy retrieval.
* Streamlined Operations: Automates manual tasks, such as book registration and updates, to improve operational efficiency.

## Purpose:

* Efficient Library Resource Management: The primary purpose is to provide a tool for efficient management of library resources, including books, journals, magazines, and member information.
* Improved User Experience: To enhance the experience of both librarians and library user.

Librarians can easily perform administrative tasks, while users can access the library's resources more conveniently, leading to increased satisfaction and usage.

* Data Integrity and Accuracy: To maintain data integrity and accuracy, reducing the risk of typographical errors, incomplete information, and inaccuracies that are common in manual record- keeping systems.
* Time and Effort Savings: By automating various processes such as book registration and updates, the system saves time and effort, allowing library staff to focus on more critical tasks.

## Technology Used:

Python: Python is the primary programming language used for the development of the application. It is chosen for its simplicity, readability, and versatility.

Tkinter: Tkinter is a standard Python library for creating graphical user interfaces (GUls).

It is used to design the user interface of the application, providing an intuitive and user-friendly front end for librarians and users.

MySQL. Database: MySQL, is employed as the backend database management system. It is responsible for storing and managing data related to books, members, and other library information, ensuring data integrity and retrieval efficiency.

CRUD Operations: The project involves implementing CRUD (Create, Read, Update, Delete) operations to manage data in the MySQL database. Python's database libraries, such as pymysql or MySQLdb, are used to perform these operations.

SQL: Structured Query Language (SQL) is used to interact with the MySQL database. SQL queries are employed to retrieve, insert, update, and delete data from the database

## EXISTING SYSTEM

System analysis for the "Library Management System in Python with MySQL Database" project is a critical phase in which we identify existing challenges in manual library management. By closely collaborating with library staff and stakeholders, we gather essential information and study current processes.

We aim to pinpoint problems such as data accuracy issues and time-consuming tasks Based on this analysis, we propose a comprehensive solution that streamlines library operations and enhances user experience.

User feedback is integral, and we iterate on the proposal until it aligns perfectly with the library's needs, ensuring a successful and user-friendly system.

## PROPOSED SYSTEM

The proposed system for the "Library Management System in Python with MySQL Database" project

introduces a comprehensive solution that leverages automation and database integration to revolutionize library management. With a user-friendly interface and rich features, this system simplifies the project management process, making it more flexible and accessible. By centralizing data storage and enforcing

user authentication, it ensures data security and integrity. Users can seamlessly upload and download files, share information with others, and generate various reports for informed decision-making. With a focus on efficiency and data management, this proposed system aims to modernize library operations, ultimately

enhancing user experience and streamlining administrative tasks.

## NEED OF THE APPLICATION

The need to develop this project arises because the existing system has fewer features as compared to the proposed system.

In the propose system all data will be updated fast and with understandable manner.

This System will provide Information regarding to Notice which is one of the important features of Institute.

# FEASIBILITY STUDY

All projects are feasible, given unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time.

Different types of feasibilities are as follows:

* + Organizational and cultural feasibility
  + Technical feasibility
  + Economic feasibility
  + Schedule feasibility
  + Operational feasibility
  + Resource feasibility

**Organizational and cultural feasibility:**

Cultural feasibility means how well the end user and management accept the new system. This project is feasible because of following points:

* + As the process is managed by computer literate individuals, there is no phobiarelated to automation.
  + A perceived loss of control on the system is minimal due to less work involved.

#### Economic Feasibility:

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

#### Technical Feasibility:

Technical feasibility centers on the existing computer system (hardware, software, etc.,) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible.

#### Operational Feasibility:

People are inherently resistant to change, and computers have been known to facilitatechange. It is understandable that the introduction of a candidate system requiresspecial effort to educate, sell, and train the staff on new ways of conducting business

#### Resource Feasibility:

It is measure of availability of system resources

* + - Being individually developed there is no requirement for staff.
    - Resources needed for development are also not very expensive.
    - There is no risk of employees or staff getting transferred as these resourcesare not required at all.

# STAKE HOLDERS:

There are two types of stake holders : Internal & External

* 1. **Internal Stakeholders:**
     1. Admin
     2. Shareholders

#### External Stakeholders

* + 1. Local Community
    2. Customer
    3. Society
    4. Government

PRELIMINARY INVESTIGATION

**Software Environment**

**Python:**

Python is a general purpose interpreted, interactive, object oriented, and high level programming language. It was created by Guido van Rossum during 1985-1990 Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on python programming language.

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keyword frequently where as other languages use punctuation. and it has fewer syntactical construction than other languages.

Python is a must for student and working professionals to become a great Software Engineer specially when they are working in a Web Development Domain. I will list down some of the key advantages of learning python:

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms. Including structured object-oriented and functional programming python is often described as a" batteries included" language due to its comprehensive standard library.

**Python is Interpreted:** python is processed at runtime by The interpreter you do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive:** you can actually sit at python prompt and interact with the interpreter directly to write your program.

Python is object-oriented: python support object- oriented style or technique of programming that encapsulate code within objects.

Python is Beginners language: python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**Characteristics of python:**

Following are important characteristics of python programming It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte- code for building large applications.

It provides very high-level dynamic data type and supports dynamic type checking. It supports automatic garbage collection.

### Python GUI:

Python provides various options for developing graphical user Interface (GUls). Most important are listed below.

Tkinter-Thinter is the python interface to the TK GUI toolkit shipped with python, we would look this option in this chapter.

**Tkinter**

**A standard GUI (Graphical User Interface) toolkit in Python.**

1. **GUI Development**: Tkinter is a built-in Python library used to create GUI applications. It provides a set of widgets (such as buttons, labels, text boxes, etc.) that developers can use to build desktop applications with graphical interfaces.
2. **Cross-Platform**: Tkinter is cross-platform, meaning that applications developed using Tkinter can run on any platform that supports Python, including Windows, macOS, and Linux.
3. **Simple and Easy to Use**: Tkinter is known for its simplicity and ease of use. It has a relatively straightforward syntax, making it accessible to beginners who want to create simple GUI applications.
4. **Integration with Python**: Tkinter seamlessly integrates with Python, allowing developers to combine the power of Python with the flexibility of building GUI applications.
5. **Customization**: While Tkinter provides a set of standard widgets, it also allows for extensive customization. Developers can customize the appearance and behavior of widgets to suit their application's needs.
6. **Event-Driven Programming**: Tkinter follows an event-driven programming model, where actions such as button clicks or mouse movements trigger events that can be handled by the application.
7. **Documentation and Resources**: Tkinter has comprehensive documentation and plenty of online resources, including tutorials and examples, making it easier for developers to learn and use.

**My SQL WorkBench**

MySQL Workbench is a visual database design and administration tool provided by MySQL, a popular open-source relational database management system. It offers a wide range of features to help developers and database administrators work with MySQL databases more efficiently.

One of the key features of MySQL Workbench is its ability to design and model databases visually using Entity-Relationship (ER) diagrams. This allows users to create and modify database schemas intuitively, by dragging and dropping tables, defining relationships between them, and specifying attributes.

1. **Querying and SQL Development**: It includes a built-in SQL editor with syntax highlighting, auto-completion, and query execution capabilities. Users can write and execute SQL queries, stored procedures, and scripts directly within the application.
2. **Database Administration**: MySQL Workbench offers various administrative features such as user management, backup and restore functionalities, performance monitoring, and server configuration. It provides a centralized interface for managing MySQL servers and databases.
3. **Data Migration and Import/Export**: Users can easily migrate data between different databases or import/export data from/to various formats (e.g., CSV, JSON, XML) using MySQL Workbench's data migration and import/export wizards.

JUSTIFICATIONS OF PACKAGES

## SOFTWARE AND HARDWARE REQUIREMENTS:

**HARDWARE REQUIREMENTS**

Operating System: A 64-bit distribution capable of running 32-bit applications.

RAM: Minimum 4 GB of RAM, with 16 GB of RAM recommended for optimal performance. Disk Space: At least 2 GB of available disk space is required, although 16 GB is recommended to accommodate potential data growth and system updates.

Screen Resolution. A minimum screen resolution of 1280 x 800 is recommended to ensure a comfortable and visually pleasing user experience.

**SOFTWARE REQUIREMENTS:**

* Python
* Tkinter
* MySQL Workbench

## Development Environment:

(Visual Studio Code, PyCharm)

Operating System Support:

* Windows
* Mac
* Linux

OPERATING ENVIRONMENT

## Project Design

The project design process is not a step by step. But still much of design work depends on knowledge and experience of the design, when we start working on project design, we will face different types of problems. However, following considerations should be kept in mind during the project design phase:

## Design Objective

The primary objective of the design of course, is to deliver the requirement as specified already. In general the following design objectives kept in mind.

## Practically-

The system must be Stable and can be operated by people with average.

## Efficiency-

This involves accuracy, timelines, and comprehensiveness to the system input.

## Costs-

It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all requirement & maintain maintenance.

## Flexibility -

The system should be modified depending on the changing needs if the user. It should also be portable to difference computer system but not other than window OS.

## Security -

This is very important aspect of the design and should cover areas of hardware reliability, fall back procedure, physical security if the data. System design involves first logical design and then physical construction of the system.

## Major Project Design Activities-

Several development activities are carried out during structured designed. They are database design. Implementation planning, and system interface.

## Design process -

The system design process is an exercise of specifying how, the system will work. It is an interactive process which is based on what the system will be do as shown in the report.

## LOGICAL DESIGN:

The logical flow of a system and defines the boundaries Of a system. It includes the following steps:

* 1. Review the current physical system-its data flows, file content, volumes. Frequencies etc.
  2. Prepares output specifications that is, determines the format, content and frequency of reports.
  3. Prepares input specifications-format, content and Frequency of reports

14 Prepare edit, security and control specifications

* 1. Specifies the implementation plan

## PHYSICAL DESIGN:

Physical system produce the working system by defines the design specification that tell the programmers exactly what the candidates system must do. It includes the following steps

* + - Design the physical system
    - Specify input and output media.
    - Design the database and specify backup procedures.
    - Design physical information flow through the system and a physical design
    - Plan system implementation.
    - Prepare a conversion schedule and target date.

## Design/ Specification activities:

1. Concept formulation
2. Problem understanding
3. High level requirements proposal
4. IV Feasibility study.
5. V Requirement's engineering.
6. VI Architectural design.

## ENTITY-RELATIONSHIP DIAGRAM:

An entity-relationship model (ER model for short) describes interrelated things of Interest In a specific domain of knowledge A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

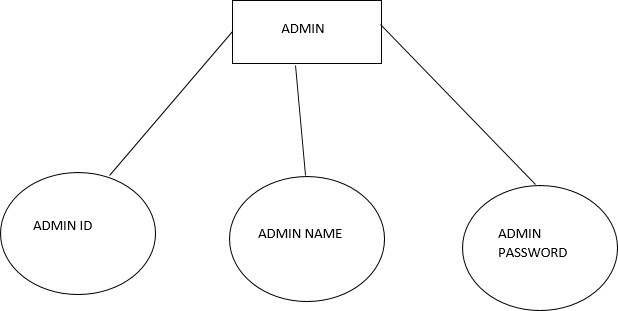
In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

## Notations used in E-R diagram

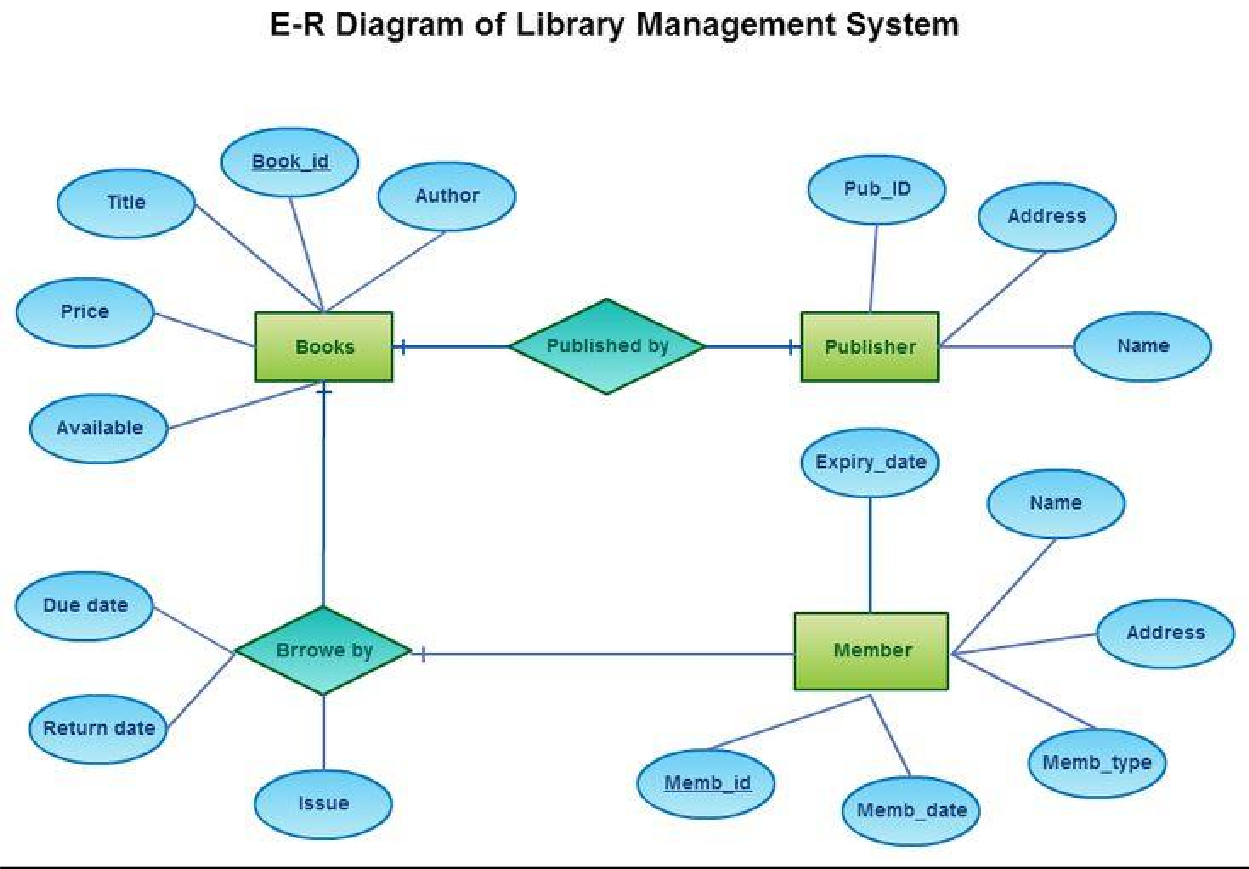
It Represents Entity Set.

It Represents Attributes.

It Represent Relationship.



## COMPLETE E-R DIAGRAM



**Class Diagram**

#### The UML Class diagram is a graphical notation used to construct and visualize object oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

* classes
* Their attributes
* operations (or methods)
* The relationships among objects.

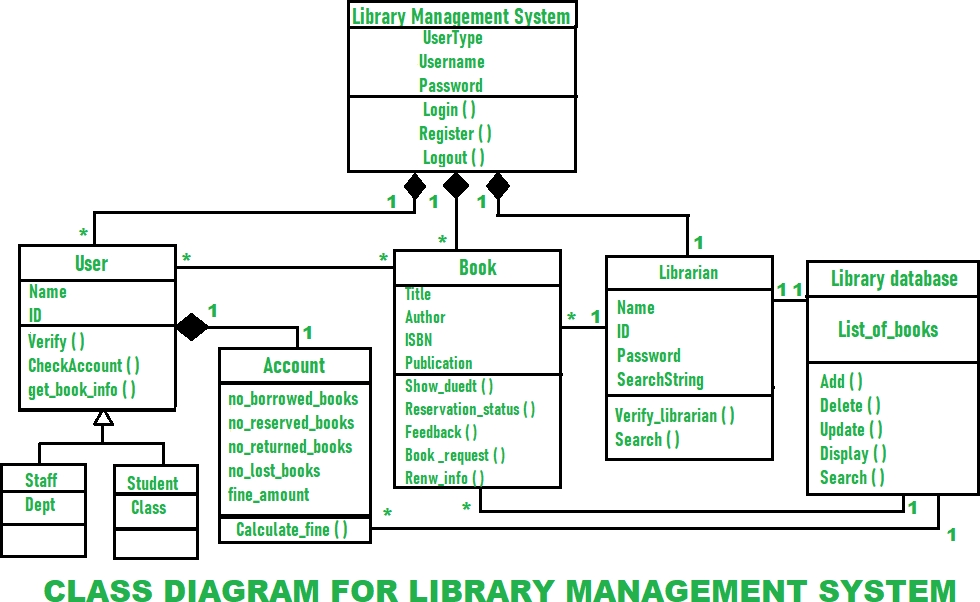
**What is a Class?**

A Class is a blueprint for an object. Objects and classes go hand in hand. We can't talk about one without talking about the other. And the entire point of Object-Oriented Design is not about objects, it's about classes, because we use classes to create objects. So a class describes what an object will be, but it isn't the object itself.

In fact, classes describe the type of objects, while objects are usable instances of classes. Each Object was built from the same set of blueprints and therefore contains the same components (properties and methods). The standard meaning is that an object is an instance of a class and object - Objects have states and behaviors.

**Class Notation**

A class represent a concept which encapsulates state (attributes) and behavior (operations). Each attribute has a type. Each operation has a signature. The class name is the **only mandatory information.**



## USE CASE DIAGRAM

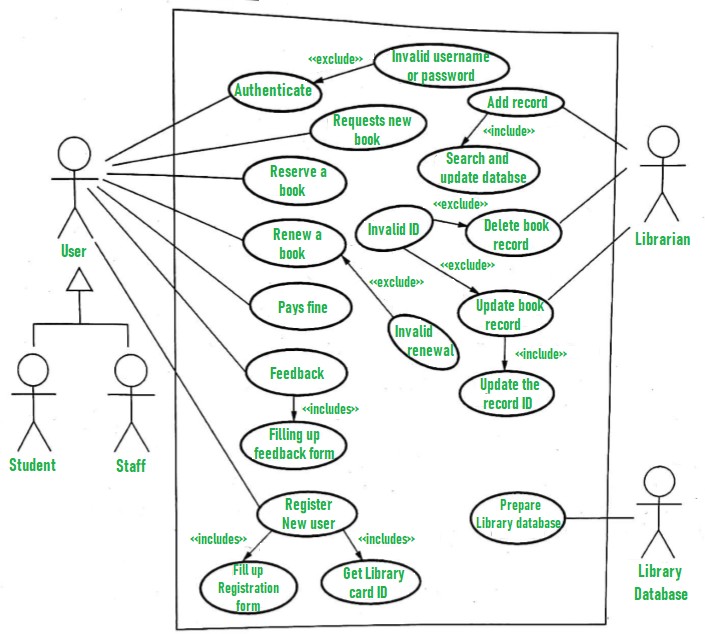
**Actors:**

**Admin:** since the system is used by him

**Student:** since who used the system.

A use case is a set of scenarios that describing an interaction between a Students and the App. A use case diagram displays the relationship among actors and use cases.

The two main components of a use case diagram are use cases and actors. The object-



oriented approach uses the term use case to describe an activity the APP carries out in response model that summaries the information about the actors of cases.

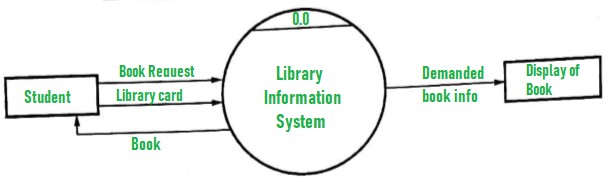
## DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams.

**Dataflow:** data moves in a specific from an origin to a destination.

**Process:** People procedure or device that use or produce data. The physical components not identified.

**Source:** External source or destination of data, which may be people program, organizations or other entities.



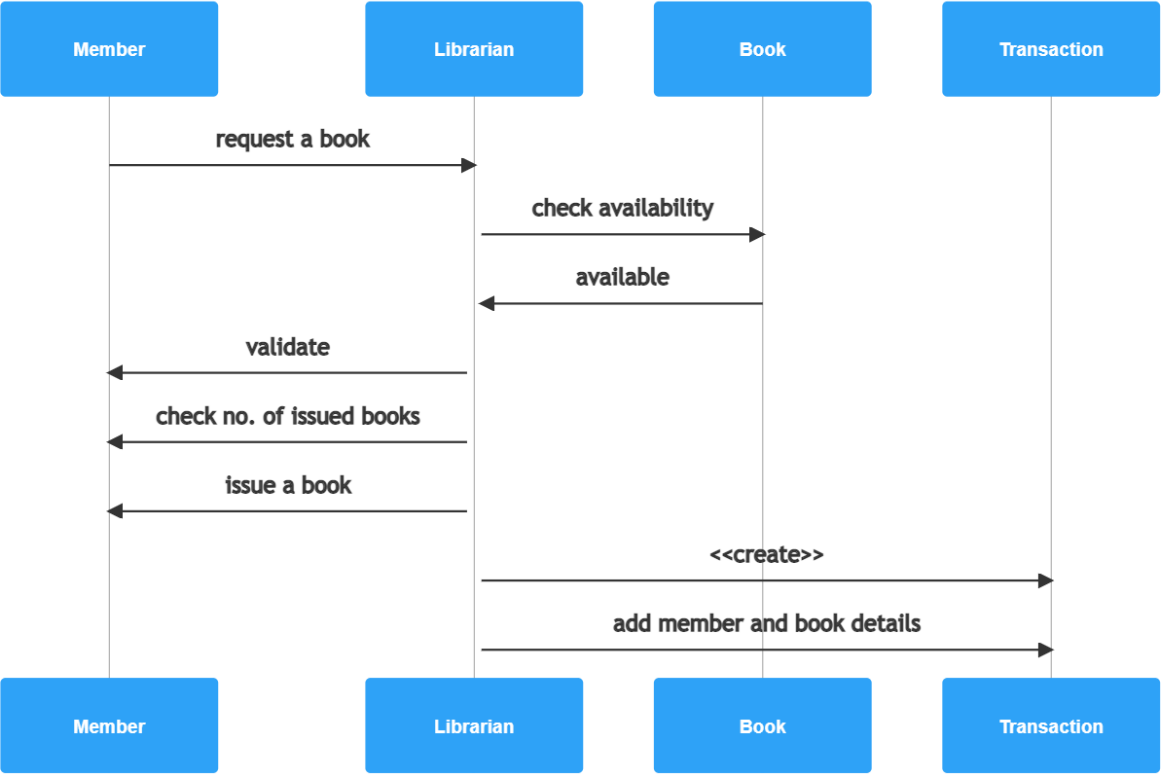
## SEQUENCE DIAGRAM:

Sequence diagram demonstrate the behavior of objects in a use case by describing the object and the messages they pass, the diagram are read left to right and descending. Sequence diagram generally show the sequence of events that occur.

A sequence diagram shows how processed operate with one another. sequence diagram are also sometime called event trace diagram. A sequence diagram shows parallel vertical lines. Different processed or object that live simultaneous and a horizontal arrows, the messages exchange between then, in the order in which they occur.

Sequence diagram emphasizes on time sequence of message on the structural organization of the objects that send and receive messages.

A sequence diagram shows the sequence of the instruction between the object that occurs during the flow of event of a single scenario or use case.



SYSTEM ANAYSIS

SYSTEM DESIGN

Add Data Show Data Update Delete Reset

Exit

# E-R TO TABLES

**iNSERT**

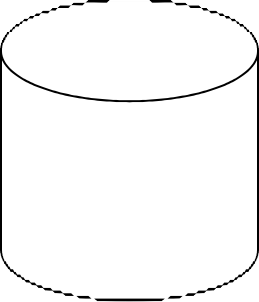
Username

Password

**Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **ALLOW NULL** |
|  | Membertype | varchar | 45 | No |
|  | PRNNo | Varchar | 45 | No |
|  | IDNo | Varchar | 45 | No |
|  | FirstName | Varchar | 45 | No |
|  | LastName | Varchar | 45 | No |
|  | Address1 | Varchar | 45 | No |
|  | Address2 | Varchar | 45 | No |
|  | Mobile | Varchar | 45 | No |
|  | BookId | Varchar | 45 | No |
|  | BookTitle | Varchar | 45 | No |
|  | Author | Varchar | 45 | No |
|  | DateBorrowed | Varchar | 45 | No |
|  | DueDate | Varchar | 45 | No |
|  | DaysOfBook | varchar | 50 | No |
|  | LaterReturnFine | varchar | 50 | No |
|  | DateOverDue | varchar | 50 | No |
|  | ActualPrice | varchar | 50 | No |

**COMPONENT DIAGRAM**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **ALLOW NULL** |
| Primary key | User name | Varchar | 50 | No |
|  | Password | Varchar | 50 | No |

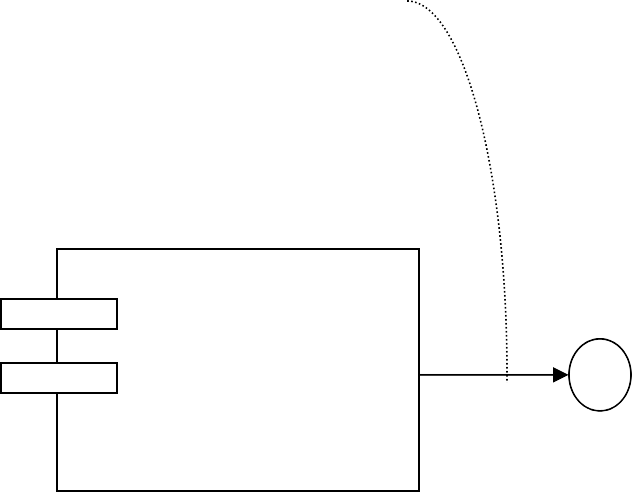
Python ,Tkinter

Library Management System

DATABASE

ADMIN

Book issued



Entry in Db

Admin

Booked Issued

Data

Data in DB

PACKAGE DIAGRAM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Library Management System | | | | | | |
| Member Info |  | BookId |  | Book Detail | | |
| Prnno | BookTitle |  | | |
| IdName | Author Name |  | | |
| Firstname | Date Borrowed |  | | |
| Mobile | Actual Price |  | | |
| AddData | Showdata | Update | Delete | | Reset | Exit |
| DataBase Data | | | | | | |

# STRUCTURE CHART DIAGRAM

Proces

Library Management System

Student comes for book

Book chosen Successfull

Admin Takes

Details

Generation of

Database

Generate

Information

MENU TREE



# PROGRAM DESCRIPTION AND

**NAMING**

# CONVENTION

**PROGRAM DESCRIPTION:**

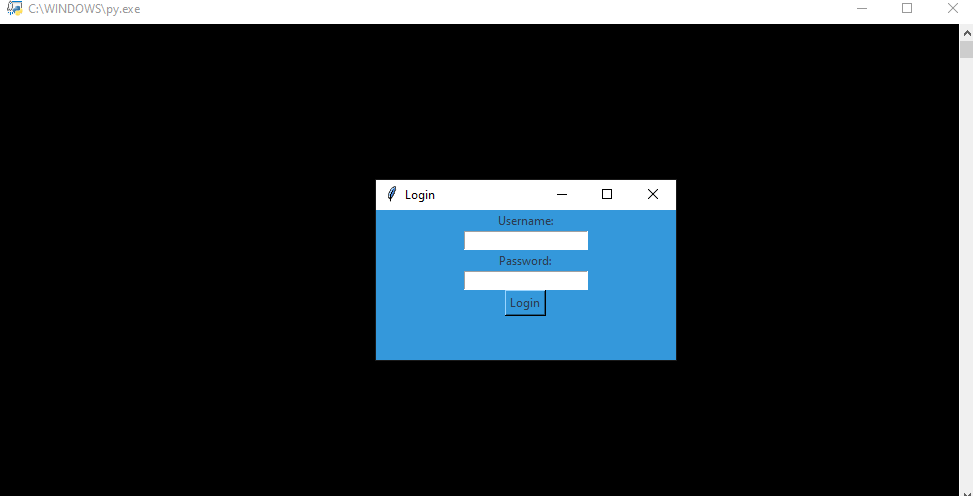
|  |  |  |
| --- | --- | --- |
| **SR.NO** | **PROGRAM** | **DESCRIPTION** |
| 1 | Library membership Info | Helps create a new member info |
| 2 | Book Details | Helps to check any member details by  just entering ckicking show data All required Details are viewed |
| 3 | Button table | Display button |
| 4 | Registration report | It helps to view all the registration details |

# VALIDATION

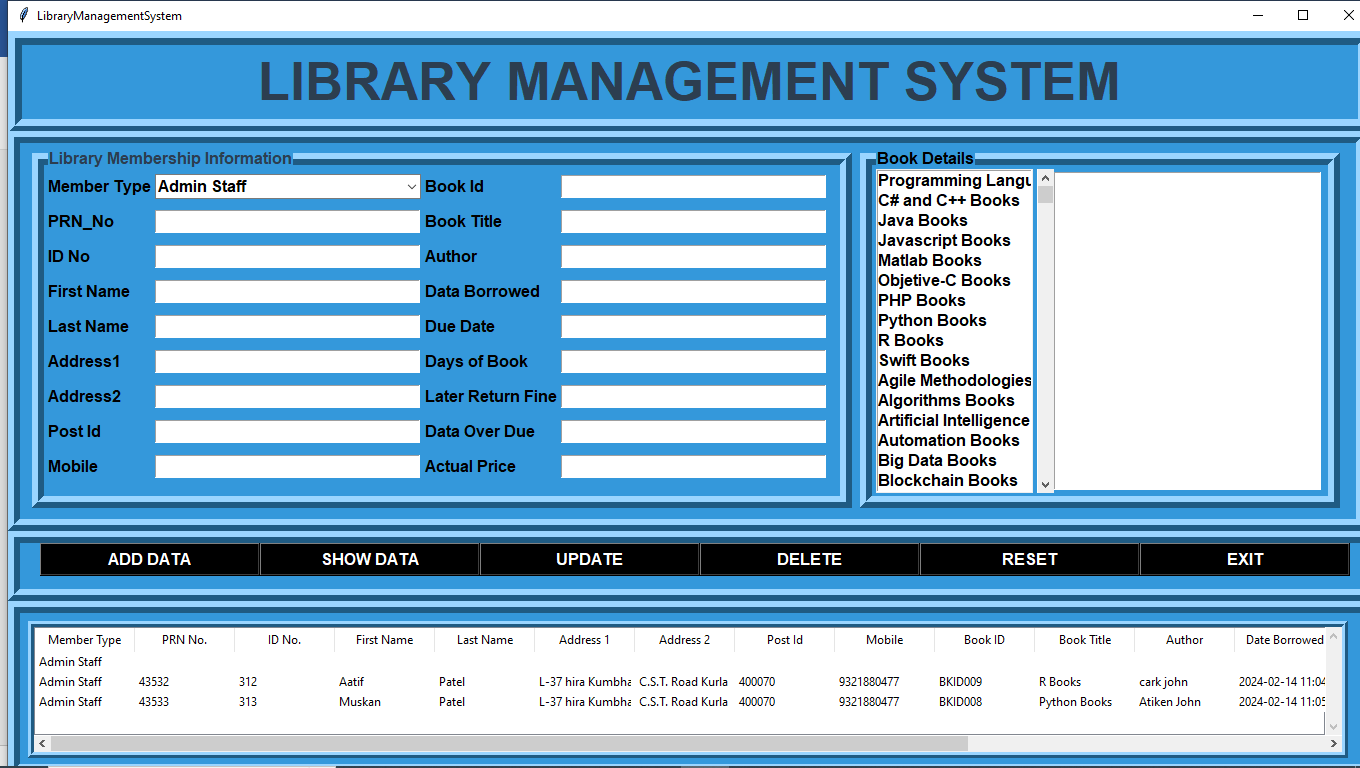
For Validation Purpose In This Application, I have done following:

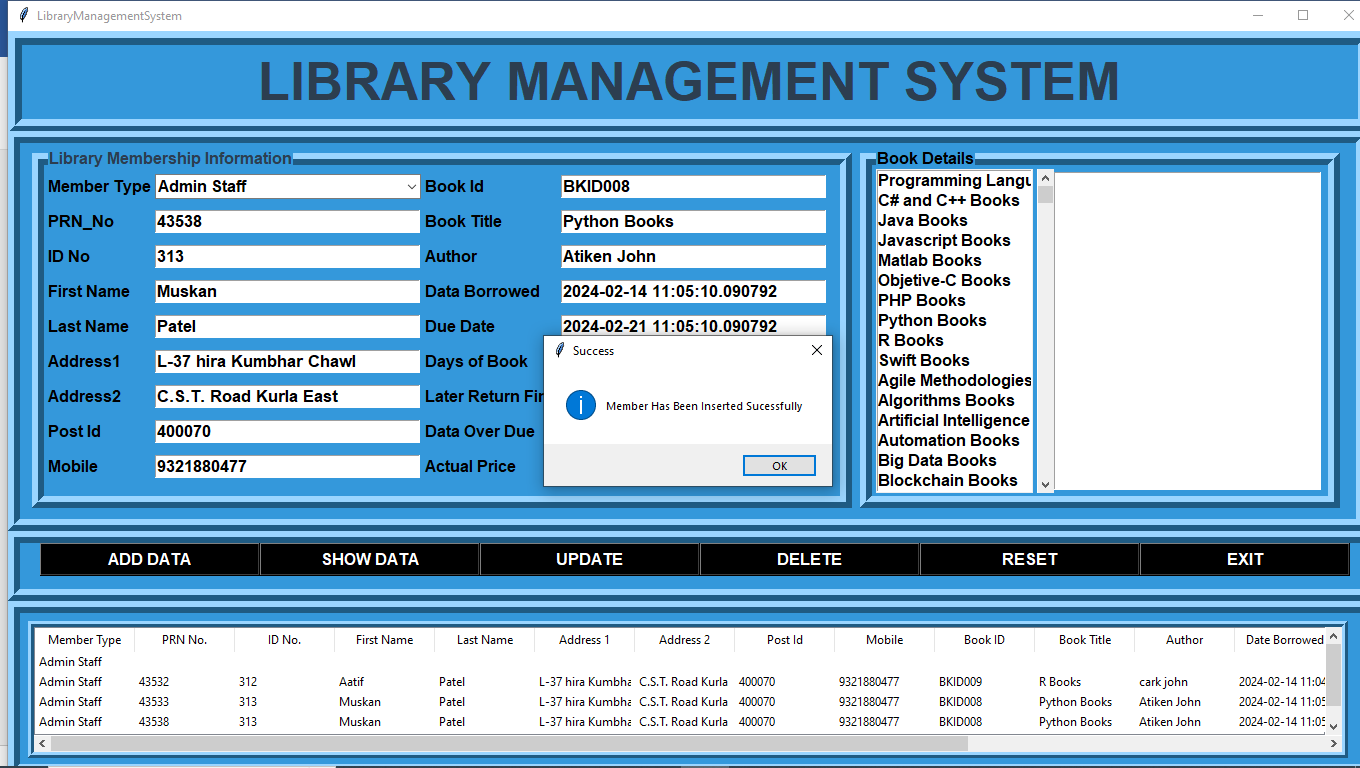
* One subroutine Is Used For validating fields which should accept only character values like name field.
* Other subroutine Is Used For validating fields which should accept only number values like contact number field.
* The text length property of the textbox is used to ensure that no fields are left empty.

SYSTEM CODING

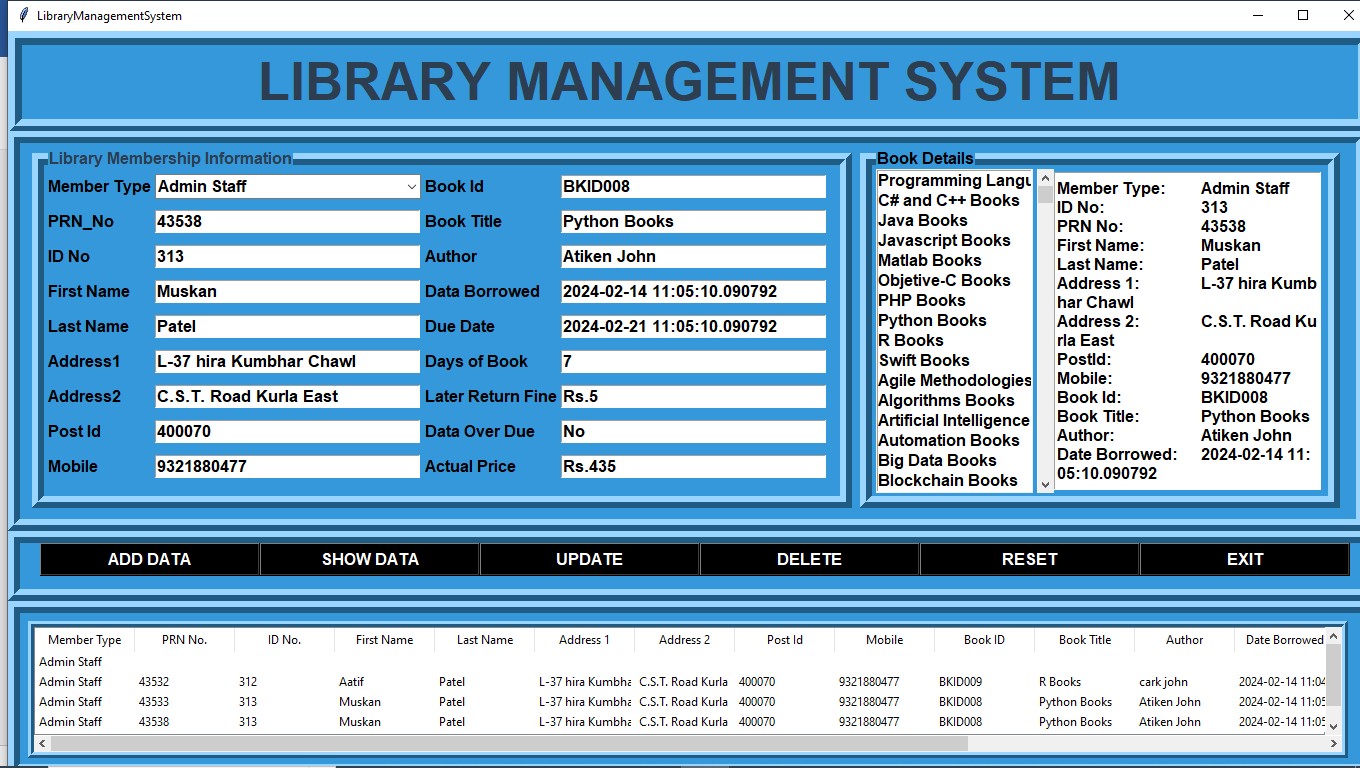


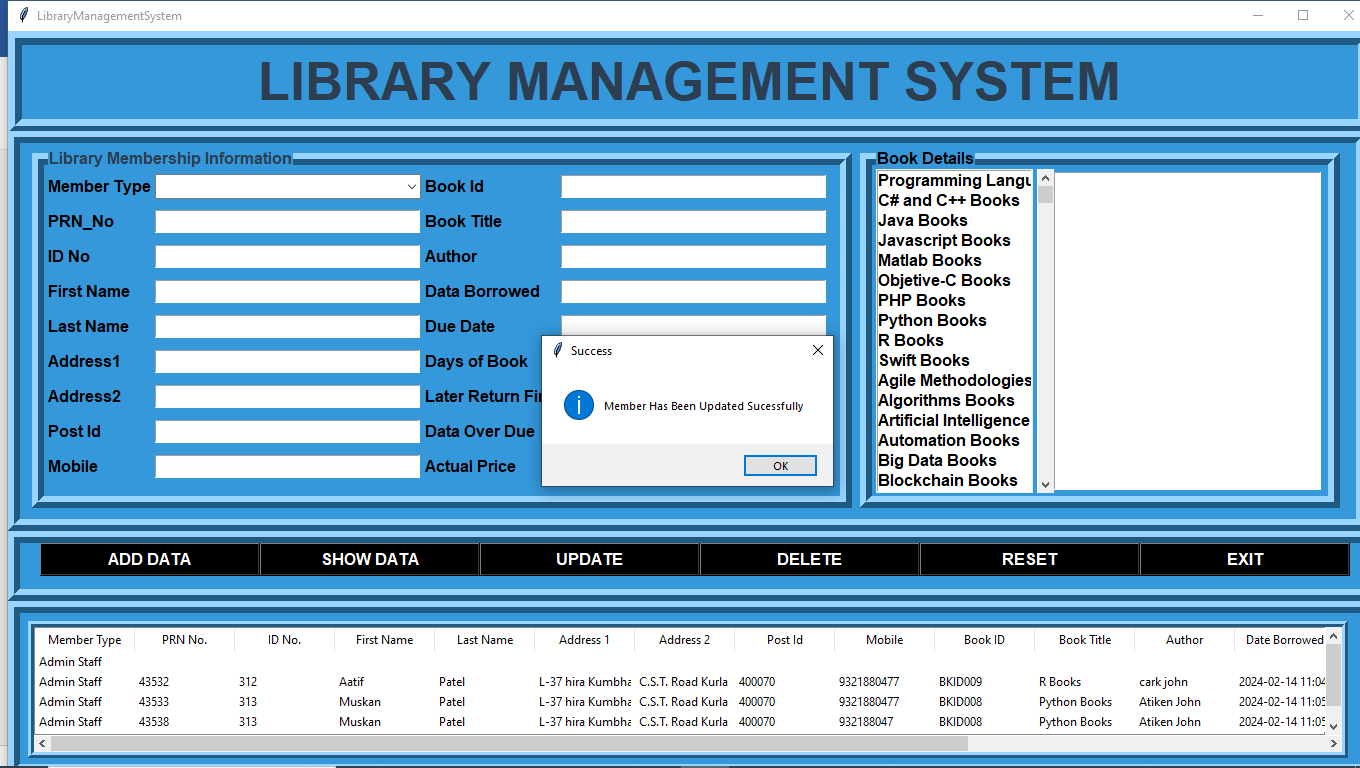
### LOGIN PAGE



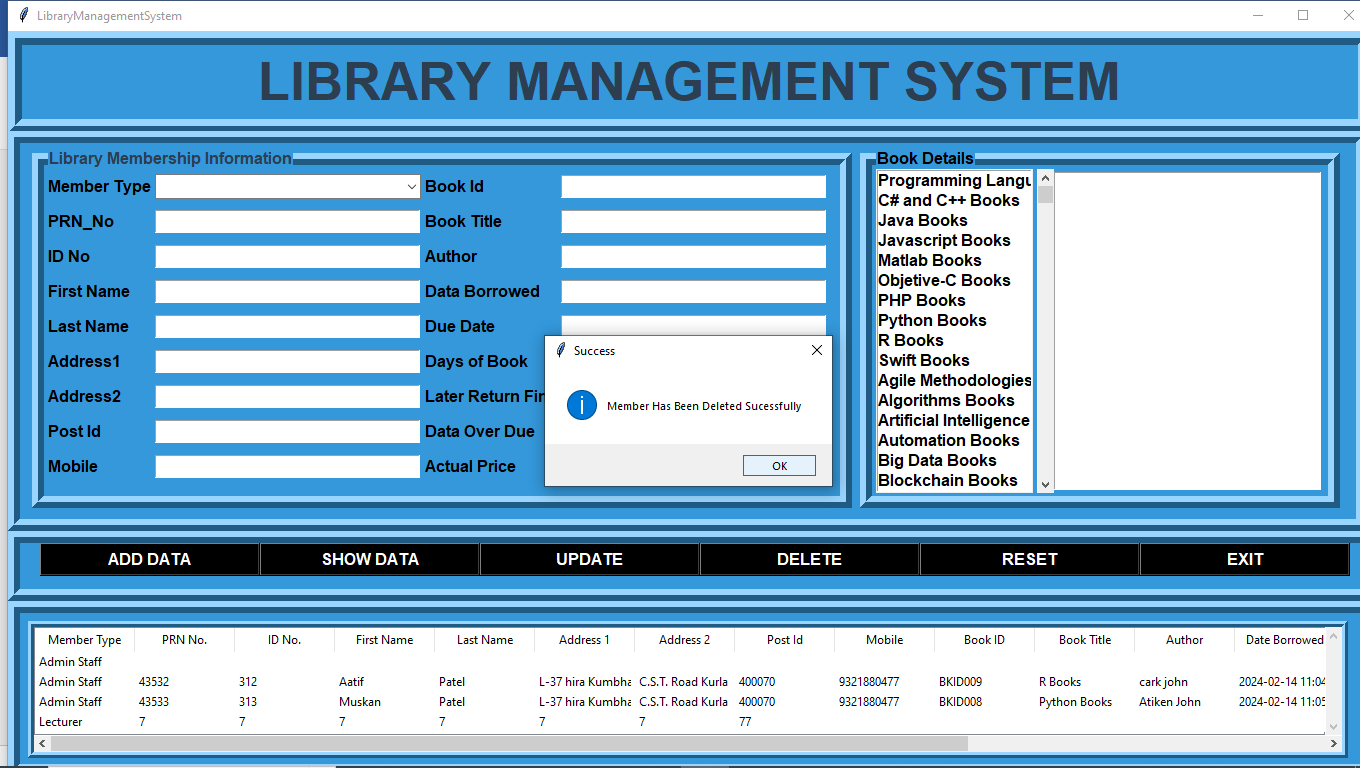


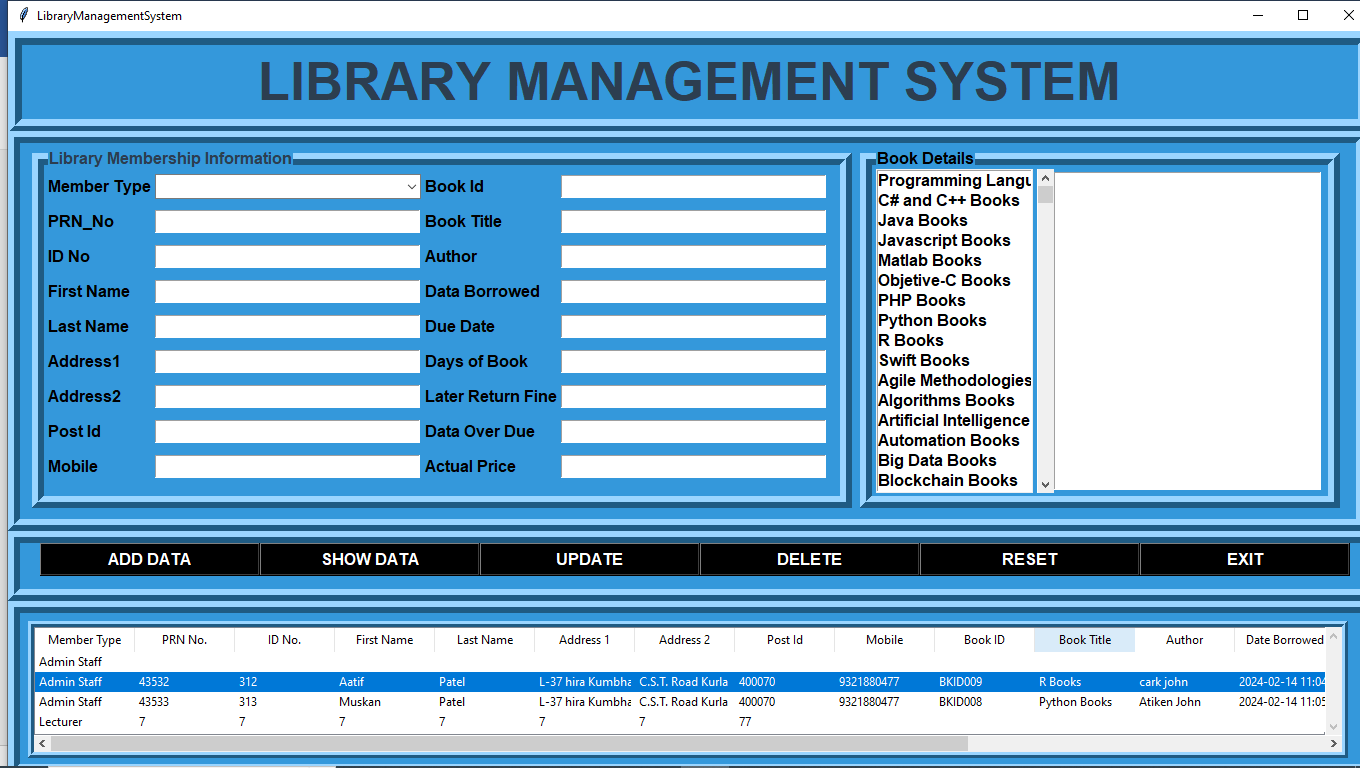
**ADD DATA**



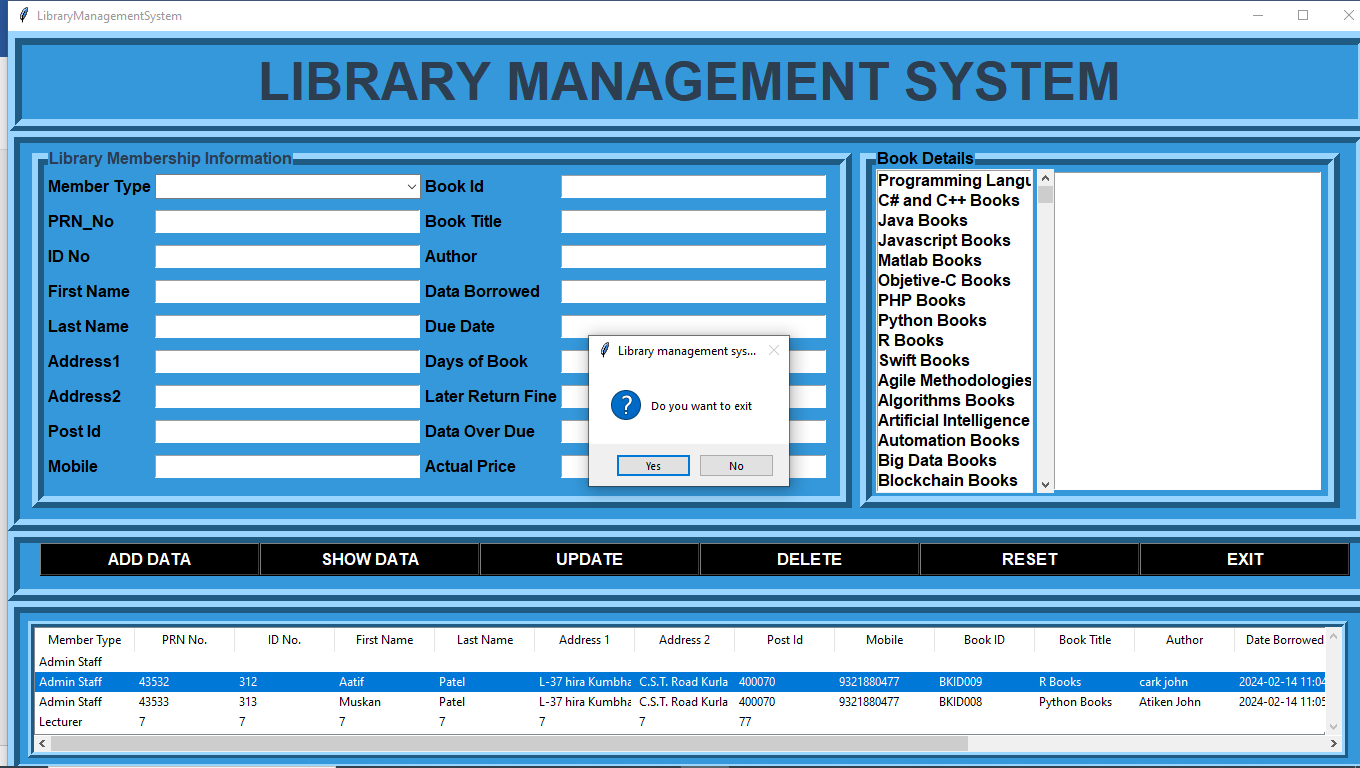


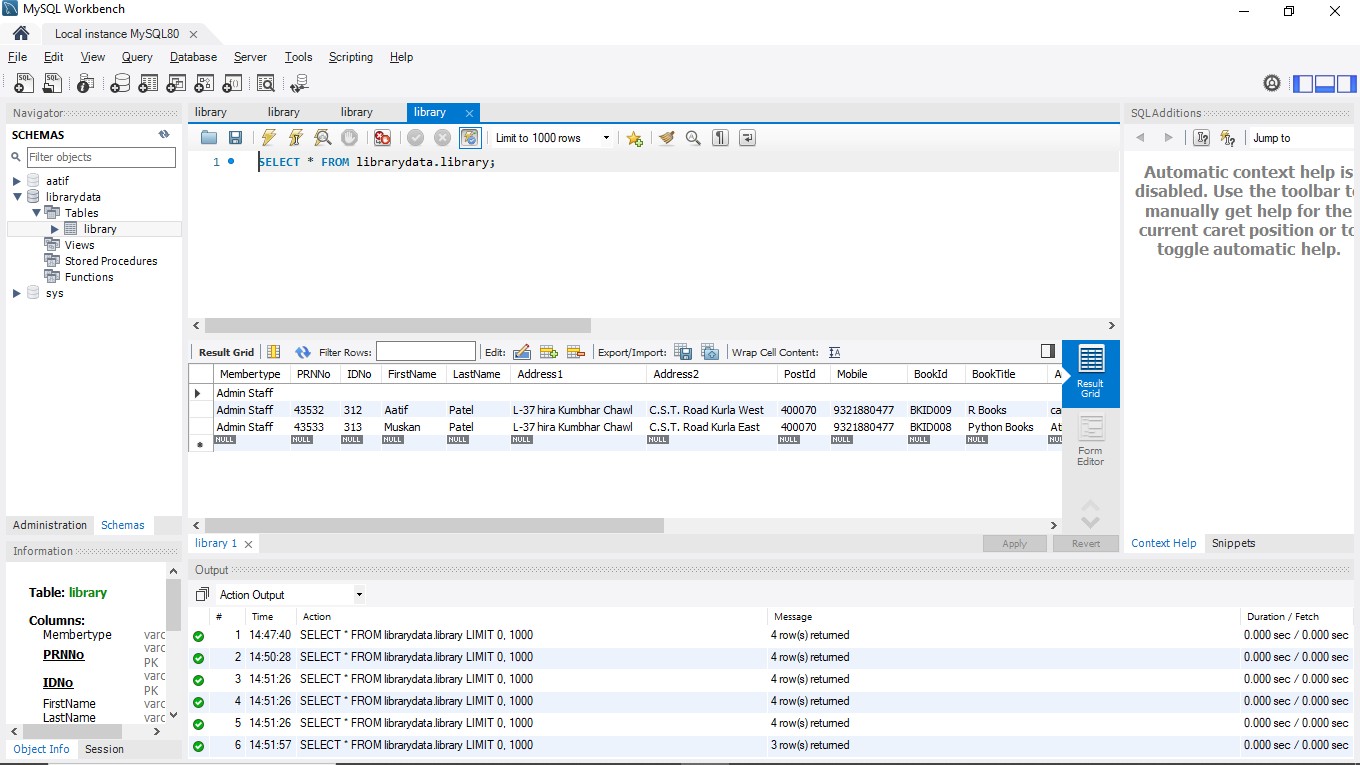
### UPDATE DATA



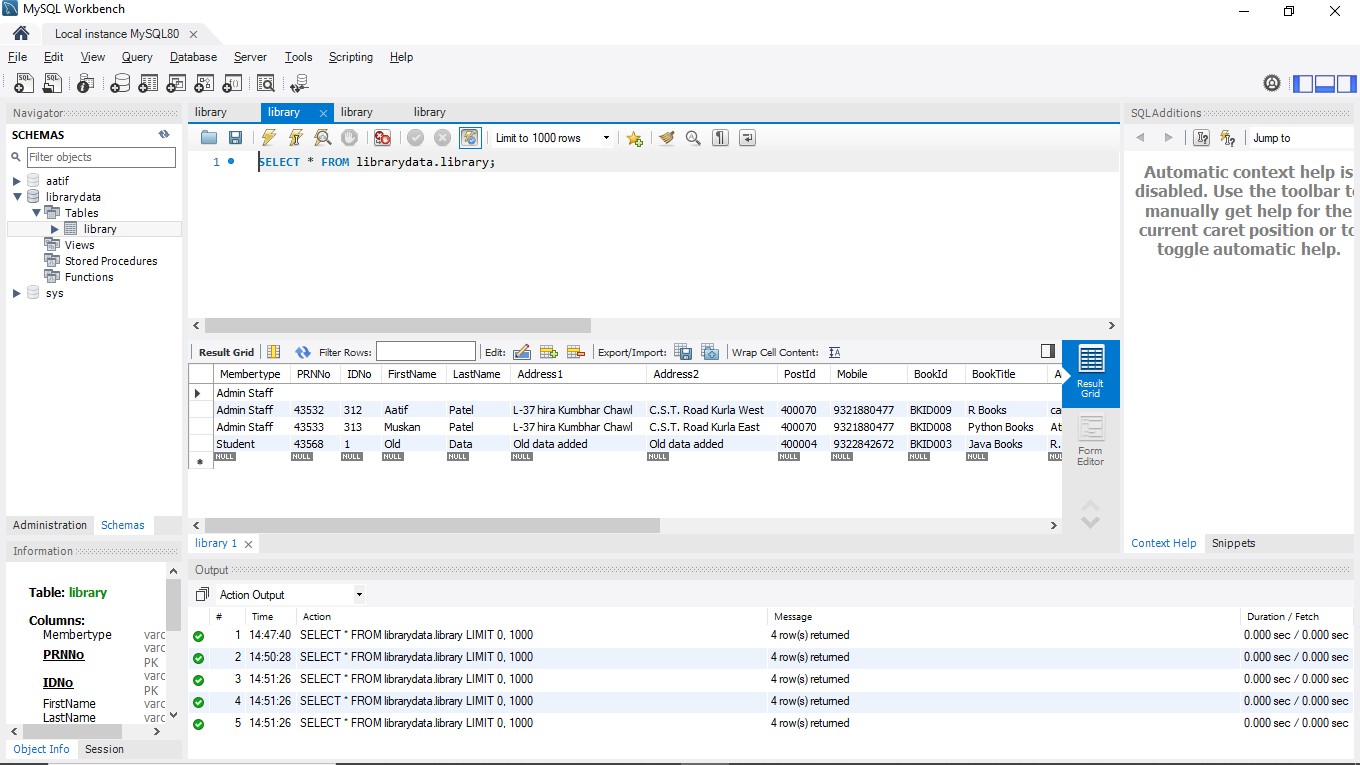
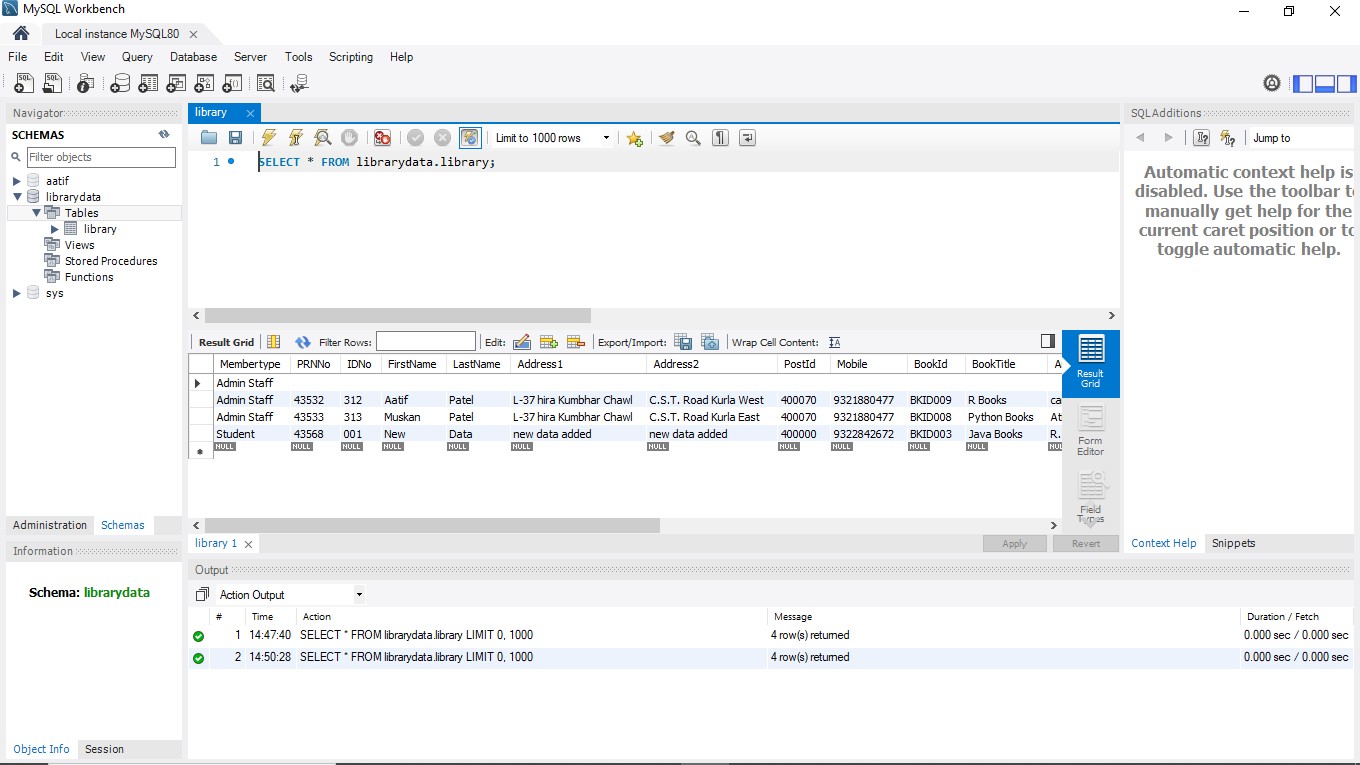


**RESET DATA**

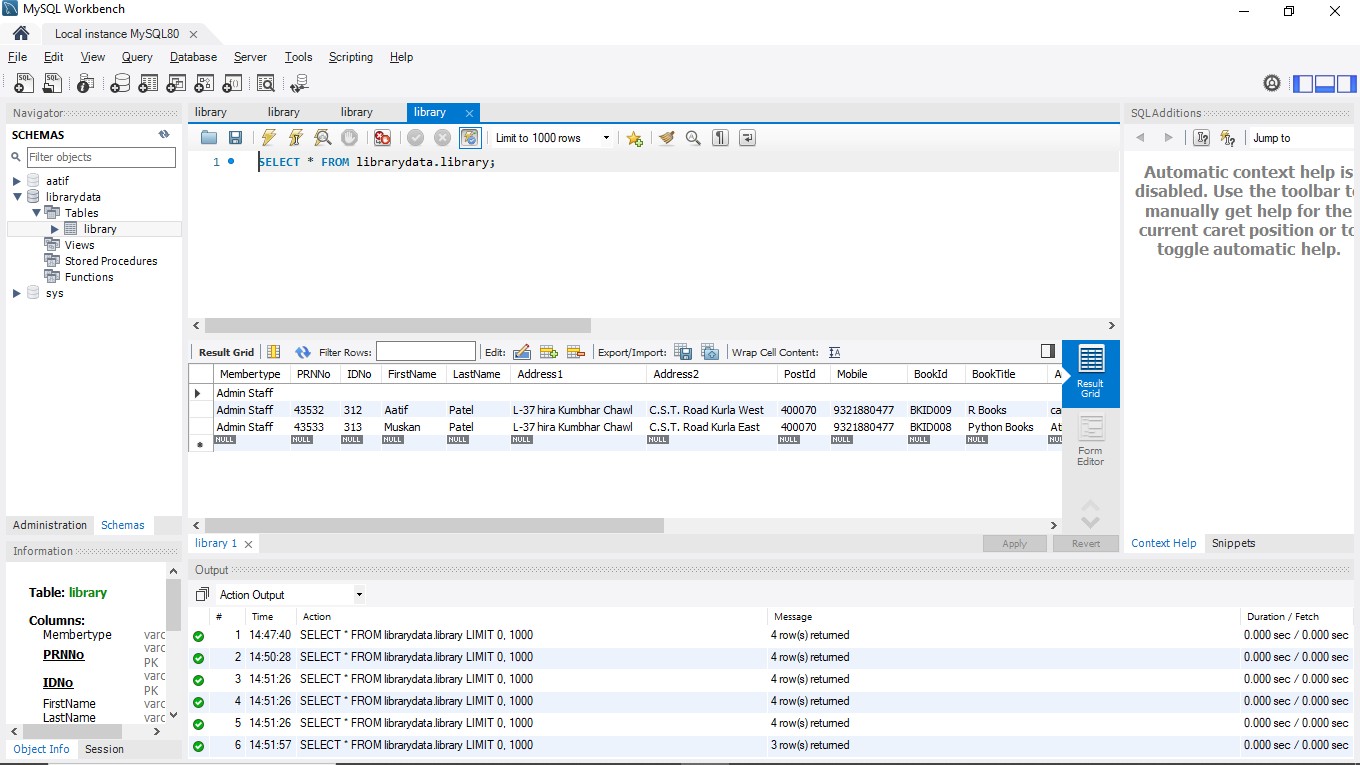




**DATA TABLE**



**DATA UPDATED**



SCREEN LAYOUTS

from tkinter import \* from tkinter import ttk import mysql.connector

from tkinter import messagebox import tkinter

import datetime

class LoginWindow:

def init (self, root):

self.root = root self.root.title("Login") self.root.geometry("300x150")

self.root.configure(bg="#3498db") # Background color

self.lblUsername = Label(self.root, text="Username:", bg="#3498db", fg="#2c3e50") # Background and foreground colors

self.lblUsername.pack() self.entUsername = Entry(self.root) self.entUsername.pack()

self.lblPassword = Label(self.root, text="Password:", bg="#3498db", fg="#2c3e50") # Background and foreground colors

self.lblPassword.pack()

self.entPassword = Entry(self.root, show="\*") self.entPassword.pack()

self.btnLogin = Button(self.root, text="Login", command=self.login, bg="#3498db", fg="#2c3e50") #

Background and foreground colors self.btnLogin.pack()

def login(self):

# Check if username and password match predefined values

username = self.entUsername.get() password = self.entPassword.get()

if username == "admin" and password == "admin": # Change to your actual username and password self.root.destroy() # Close the login window

self.open\_main\_application() else:

messagebox.showerror("Login Failed", "Invalid username or password")

def open\_main\_application(self):

root = Tk()

obj = LibraryManagementSystem(root) root.mainloop()

class LibraryManagementSystem: def init (self, root):

self.root = root self.root.title("LibraryManagementSystem") self.root.geometry("1550x800+0+0")

#=======================variable===================

self.MemberType\_var=StringVar() self.PRNNo\_var=StringVar() self.IDNo\_var=StringVar() self.FirstName\_var=StringVar() self.LastName\_var=StringVar() self.Address1\_var=StringVar() self.Address2\_var=StringVar() self.PostId\_var=StringVar() self.Mobile\_var=StringVar() self.BookId\_var=StringVar() self.BookTitle\_var=StringVar()

self.Author\_var=StringVar() self.DateBorrowed\_var=StringVar() self.DueDate\_var=StringVar() self.DaysOfBook\_var=StringVar() self.LaterReturnFine\_var=StringVar() self.DateOverDue\_var=StringVar() self.ActualPrice\_var=StringVar()

lbltitle = Label(self.root, text="LIBRARY MANAGEMENT SYSTEM", bg="#3498db", fg="#2c3e50", bd=14, relief=RIDGE, font=("new times roman", 40, "bold"), padx=2, pady=6)

lbltitle.pack(side=TOP, fill=X)

frame=Frame(self.root,bd=12, relief=RIDGE,padx=20,bg="#3498db") frame.place(x=0,y=100,width=1360,height=400) #================Data Frame Rate====================

DataFrameLeft=LabelFrame(frame,text="Library Membership Information", bg="#3498db", fg="#2c3e50", bd=12, relief=RIDGE, font=("new times roman", 12, "bold"))

DataFrameLeft.place(x=-8,y=5,width=820,height=360)

lblMember=Label(DataFrameLeft,font=("new times roman", 12, "bold"),text="Member Type",padx=2,pady=6,bg="#3498db")

lblMember.grid(row=0,column=0,sticky=W)

comMember=ttk.Combobox(DataFrameLeft,textvariable=self.MemberType\_var,state="readonly", font=("new times roman", 12, "bold"),width=27)

comMember["value"]=("Admin Staff","Student","Lecturer") comMember.current(0)

comMember.grid(row=0,column=1)

lblPRN\_No=Label(DataFrameLeft,bg="#3498db",text="PRN\_No",font=("new times roman", 12, "bold"),padx=2)

lblPRN\_No.grid(row=1,column=0,sticky=W)

textPRN\_No=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.PRNNo\_var,width=29)

textPRN\_No.grid(row=1,column=1)

lblTitle=Label(DataFrameLeft,bg="#3498db",text="ID No",font=("new times roman", 12, "bold"),padx=2,pady=4)

lblTitle.grid(row=2,column=0,sticky=W)

textTitle=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.IDNo\_var,width=29)

textTitle.grid(row=2,column=1)

lblFirstName=Label(DataFrameLeft,bg="#3498db",text="First Name",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblFirstName.grid(row=3,column=0,sticky=W)

textFirstName=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.FirstName\_var,width=29)

textFirstName.grid(row=3,column=1)

lblLastName=Label(DataFrameLeft,bg="#3498db",text="Last Name",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblLastName.grid(row=4,column=0,sticky=W)

textLastName=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.LastName\_var,width=29)

textLastName.grid(row=4,column=1)

lblAddress1=Label(DataFrameLeft,bg="#3498db",text="Address1",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblAddress1.grid(row=5,column=0,sticky=W)

textAddress1=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.Address1\_var,width=29)

textAddress1.grid(row=5,column=1)

lblAddress2=Label(DataFrameLeft,bg="#3498db",text="Address2",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblAddress2.grid(row=6,column=0,sticky=W)

textAddress2=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.Address2\_var,width=29)

textAddress2.grid(row=6,column=1)

lblPostCode=Label(DataFrameLeft,bg="#3498db",text="Post Id",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblPostCode.grid(row=7,column=0,sticky=W)

textPostcode=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.PostId\_var,width=29)

textPostcode.grid(row=7,column=1)

lblMobile=Label(DataFrameLeft,bg="#3498db",text="Mobile",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblMobile.grid(row=8,column=0,sticky=W)

textMobile=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.Mobile\_var,width=29)

textMobile.grid(row=8,column=1)

lblBookId=Label(DataFrameLeft,bg="#3498db",text="Book Id",font=("new times roman", 12, "bold"),padx=2)

lblBookId.grid(row=0,column=2,sticky=W)

textBookId=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.BookId\_var,width=29)

textBookId.grid(row=0,column=3)

lblBookTitle=Label(DataFrameLeft,bg="#3498db",text="Book Title",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblBookTitle.grid(row=1,column=2,sticky=W)

textBookTitle=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.BookTitle\_var,width=29)

textBookTitle.grid(row=1,column=3)

lblAuthor=Label(DataFrameLeft,bg="#3498db",text="Author",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblAuthor.grid(row=2,column=2,sticky=W)

textAuthor=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.Author\_var,width=29)

textAuthor.grid(row=2,column=3)

lblDataBorrowed=Label(DataFrameLeft,bg="#3498db",text="Data Borrowed",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblDataBorrowed.grid(row=3,column=2,sticky=W)

textDataBorrowed=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.DateBorrowed\_var,width=29)

textDataBorrowed.grid(row=3,column=3)

lblDataDue=Label(DataFrameLeft,bg="#3498db",text="Due Date",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblDataDue.grid(row=4,column=2,sticky=W)

textDataDue=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.DueDate\_var,width=29)

textDataDue.grid(row=4,column=3)

lblDaysOfBook=Label(DataFrameLeft,bg="#3498db",text="Days of Book",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblDaysOfBook.grid(row=5,column=2,sticky=W)

textDaysofBook=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.DaysOfBook\_var,width=29)

textDaysofBook.grid(row=5,column=3)

lblLastReturnFine=Label(DataFrameLeft,bg="#3498db",text="Later Return Fine",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblLastReturnFine.grid(row=6,column=2,sticky=W)

textLastReturnFine=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.LaterReturnFine\_var,width=29)

textLastReturnFine.grid(row=6,column=3)

lblDataOverDue=Label(DataFrameLeft,bg="#3498db",text="Data Over Due",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblDataOverDue.grid(row=7,column=2,sticky=W)

textDataOverDue=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.DateOverDue\_var,width=29)

textDataOverDue.grid(row=7,column=3)

lblActualPrice=Label(DataFrameLeft,bg="#3498db",text="Actual Price",font=("new times roman", 12, "bold"),padx=2,pady=6)

lblActualPrice.grid(row=8,column=2,sticky=W)

textActualPrice=Entry(DataFrameLeft,font=("new times roman", 13, "bold"),textvariable=self.ActualPrice\_var,width=29)

textActualPrice.grid(row=8,column=3)

#===================DataFrame Right==================

DataFrameRight=LabelFrame(frame,text="Book Details", bg="#3498db", bd=12, relief=RIDGE, font=("new times roman", 12, "bold"))

DataFrameRight.place(x=820,y=5,width=480,height=360)

self.txtBox=Text(DataFrameRight,font=("new times roman", 12, "bold"),width=29,height=16,padx=2,pady=6)

self.txtBox.grid(row=0,column=3)

listScrollbar=Scrollbar(DataFrameRight) listScrollbar.grid(row=0, column=1, sticky="ns")

listbooks = ['Programming Languages Books', 'C# and C++ Books', 'Java Books', 'Javascript Books', 'Matlab Books', 'Objetive-C Books', 'PHP Books', 'Python Books', 'R Books', 'Swift Books', 'Agile Methodologies Books', 'Algorithms Books', 'Artificial Intelligence Books', 'Automation

Books',

'Big Data Books', 'Blockchain Books', 'Books about Computing', 'Books about ICT', 'Cloud Computing Books', 'Computer Networks Books', 'Computer Security Books',

'Cryptography Books',

'Database Books', 'E-Commerce Books', 'Excel Books']

def SelectBook(event=""):

value = str(listbox.get(listbox.curselection())) x=value

if x=="Programming Languages Books": self.BookId\_var.set("BKID001")

self.BookTitle\_var.set("Programming Languages Books") self.Author\_var.set("Patel Aatif") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.534")

elif x=="C# and C++ Books": self.BookId\_var.set("BKID002") self.BookTitle\_var.set("C# and C++ Books") self.Author\_var.set("yasmin Khan") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.500")

elif x=="Java Books": self.BookId\_var.set("BKID003") self.BookTitle\_var.set("Java Books") self.Author\_var.set("R. K. Narayan") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.734")

elif x=="Javascript Books": self.BookId\_var.set("BKID004") self.BookTitle\_var.set("Javascript Books") self.Author\_var.set("Tarn Adams") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.574")

elif x=="Matlab Books": self.BookId\_var.set("BKID005") self.BookTitle\_var.set("Matlab Books") self.Author\_var.set("Paul Allen") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7)

self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.349")

elif x=="Objetive-C Books": self.BookId\_var.set("BKID006") self.BookTitle\_var.set("Objetive-C Books") self.Author\_var.set("Bill Atiken") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.645")

elif x=="PHP Books": self.BookId\_var.set("BKID007") self.BookTitle\_var.set("PHP Books") self.Author\_var.set("John Backus") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.409")

elif x=="Python Books": self.BookId\_var.set("BKID008") self.BookTitle\_var.set("Python Books") self.Author\_var.set("Atiken John") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.435")

elif x=="R Books": self.BookId\_var.set("BKID009") self.BookTitle\_var.set("R Books") self.Author\_var.set("cark john") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.458")

elif x=="Swift Books": self.BookId\_var.set("BKID0010") self.BookTitle\_var.set("Swift Books") self.Author\_var.set("cark john")

d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.458")

elif x=="Agile Methodologies Books": self.BookId\_var.set("BKID0011") self.BookTitle\_var.set("Agile Methodologies Books") self.Author\_var.set("Mark black") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.654")

elif x=="Algorithms Books": self.BookId\_var.set("BKID0012") self.BookTitle\_var.set("Algorithms Books") self.Author\_var.set("Lars bak") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1)

self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.435")

elif x=="Artificial Intelligence Books": self.BookId\_var.set("BKID0013") self.BookTitle\_var.set("Artificial Intelligence Books") self.Author\_var.set("carl backhouse") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.438")

elif x=="Automation Books": self.BookId\_var.set("BKID0014") self.BookTitle\_var.set("Automation Books") self.Author\_var.set("Richerd goad") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No")

self.ActualPrice\_var.set("Rs.358")

elif x=="Big Data Books": self.BookId\_var.set("BKID0015") self.BookTitle\_var.set("Big Data Books") self.Author\_var.set("kent bak") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2

self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.899")

elif x=="Blockchain Books": self.BookId\_var.set("BKID0016") self.BookTitle\_var.set("Blockchain Books") self.Author\_var.set("Doung bel") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.358")

elif x=="Books about Computing": self.BookId\_var.set("BKID0017")

self.BookTitle\_var.set("Books about Computing") self.Author\_var.set("Doung bel") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.478")

elif x=="Books about ICT": self.BookId\_var.set("BKID0018") self.BookTitle\_var.set("Books about ICT") self.Author\_var.set("Kim hang") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.675")

elif x=="Cloud Computing Books": self.BookId\_var.set("BKID0019") self.BookTitle\_var.set("Cloud Computing Books") self.Author\_var.set("hang fang")

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.655")

elif x=="Computer Networks Books": self.BookId\_var.set("BKID0020") self.BookTitle\_var.set("Computer Networks Books") self.Author\_var.set("han Tim ") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.255")

elif x=="Computer Security Books": self.BookId\_var.set("BKID0021") self.BookTitle\_var.set("Computer Security Books") self.Author\_var.set("Kim hang")

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.980")

elif x=="Cryptography Books": self.BookId\_var.set("BKID0022") self.BookTitle\_var.set("Cryptography Books") self.Author\_var.set("Kim hang") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.665")

elif x=="Database Books": self.BookId\_var.set("BKID0023") self.BookTitle\_var.set("Database Books") self.Author\_var.set("Tim burner") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7)

self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.312")

elif x=="E-Commerce Books": self.BookId\_var.set("BKID0024") self.BookTitle\_var.set("E-Commerce Books") self.Author\_var.set("ganf dog") d1=datetime.datetime.today() d2=datetime.timedelta(days=7)

d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.312")

elif x=="Excel Books": self.BookId\_var.set("BKID0025") self.BookTitle\_var.set("Excel Books") self.Author\_var.set("Muskan Patel") d1=datetime.datetime.today() d2=datetime.timedelta(days=7) d3=d1+d2 self.DateBorrowed\_var.set(d1) self.DueDate\_var.set(d3) self.DaysOfBook\_var.set(7) self.LaterReturnFine\_var.set("Rs.5") self.DateOverDue\_var.set("No") self.ActualPrice\_var.set("Rs.980")

listbox=Listbox(DataFrameRight,font=("new times roman", 12, "bold"),width=17,height=16) listbox.bind("<<ListboxSelect>>", SelectBook)

listbox.grid(row=0,column=0,padx=4) listScrollbar.config(command=listbox.yview)

for item in listbooks: listbox.insert(END, item)

#===================Button Frame==================

framebutton=Frame(self.root,bd=12, relief=RIDGE,padx=20,bg="#3498db") framebutton.place(x=0,y=500,width=1366,height=70)

btnAddData=Button(framebutton,command=self.add\_data,text="ADD DATA",font=("new times roman", 12, "bold"),width=21,bg="BLACK",fg="WHITE")

btnAddData.grid(row=0,column=0) btnAddData=Button(framebutton,command=self.showData,text="SHOW DATA",font=("new times

roman", 12, "bold"),width=21,bg="BLACK",fg="WHITE")

btnAddData.grid(row=0,column=1) btnAddData=Button(framebutton,command=self.update,text="UPDATE",font=("new times roman",

12, "bold"),width=21,bg="BLACK",fg="WHITE")

btnAddData.grid(row=0,column=2) btnAddData=Button(framebutton,command=self.delete,text="DELETE",font=("new times roman", 12,

"bold"),width=21,bg="BLACK",fg="WHITE") btnAddData.grid(row=0,column=3)

btnAddData=Button(framebutton,command=self.reset,text="RESET",font=("new times roman", 12, "bold"),width=21,bg="BLACK",fg="WHITE")

btnAddData.grid(row=0,column=4)

btnAddData=Button(framebutton,command=self.iExit,text="EXIT",font=("new times roman", 12, "bold"),width=20,bg="BLACK",fg="WHITE")

btnAddData.grid(row=0,column=5)

#===================Information Frame==================

framedetails=Frame(self.root,bd=12, relief=RIDGE,padx=20,bg="#3498db") framedetails.place(x=0,y=570,width=1366,height=175)

table\_frame=Frame(self.root,bd=6, relief=RIDGE,bg="#3498db") table\_frame.place(x=20,y=590,width=1320,height=137)

xscroll=ttk.Scrollbar(table\_frame,orient=HORIZONTAL) yscroll=ttk.Scrollbar(table\_frame,orient=VERTICAL)

self.library\_table = ttk.Treeview(table\_frame, columns=("MemberType", "PRNNo", "IDNo", "FirstName", "LastName", "Address1", "Address2",

"PostId", "Mobile", "BookId", "BookTitle", "Author", "DateBorrowed",

"DueDate",

"Days", "LaterReturnFine", "DateOverDue",

"ActualPrice"),xscrollcommand=xscroll.set,yscrollcommand=yscroll.set)

xscroll.pack(side=BOTTOM,fill=X) yscroll.pack(side=RIGHT,fill=Y)

xscroll.config(command=self.library\_table.xview) yscroll.config(command=self.library\_table.yview)

self.library\_table.heading("MemberType", text="Member Type") self.library\_table.heading("PRNNo",text="PRN No.") self.library\_table.heading("IDNo",text="ID No.") self.library\_table.heading("FirstName",text="First Name") self.library\_table.heading("LastName",text="Last Name") self.library\_table.heading("Address1",text="Address 1")

self.library\_table.heading("Address2",text="Address 2") self.library\_table.heading("PostId",text="Post Id") self.library\_table.heading("Mobile",text="Mobile") self.library\_table.heading("BookId", text="Book ID") self.library\_table.heading("BookTitle", text="Book Title") self.library\_table.heading("Author",text="Author") self.library\_table.heading("DateBorrowed",text="Date Borrowed") self.library\_table.heading("DueDate",text="Due Date") self.library\_table.heading("Days",text="Days") self.library\_table.heading("LaterReturnFine", text="Later Return Fine") self.library\_table.heading("DateOverDue",text="Date Over Due") self.library\_table.heading("ActualPrice",text="Actual Price")

self.library\_table["show"] = "headings" self.library\_table.pack(fill=BOTH,expand=1)

self.library\_table.column("MemberType",width=100) self.library\_table.column("PRNNo",width=100) self.library\_table.column("IDNo",width=100) self.library\_table.column("FirstName",width=100) self.library\_table.column("LastName",width=100) self.library\_table.column("Address1",width=100) self.library\_table.column("Address2",width=100) self.library\_table.column("PostId",width=100) self.library\_table.column("Mobile",width=100) self.library\_table.column("BookId",width=100) self.library\_table.column("BookTitle",width=100) self.library\_table.column("Author",width=100) self.library\_table.column("DateBorrowed",width=100) self.library\_table.column("DueDate",width=100)

self.library\_table.column("Days",width=100) self.library\_table.column("LaterReturnFine",width=100) self.library\_table.column("DateOverDue",width=100) self.library\_table.column("ActualPrice",width=100)

self.fetch\_data()

self.library\_table.bind("<ButtonRelease-1>", self.get\_cursor)

def add\_data(self):

conn = mysql.connector.connect(host="localhost", port=3306, username="root", password="Myfirstproject@123", database="librarydata")

my\_cursor=conn.cursor() my\_cursor.execute("insert into library

values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(

self.MemberType\_var.get(), self.PRNNo\_var.get(), self.IDNo\_var.get(), self.FirstName\_var.get(), self.LastName\_var.get(), self.Address1\_var.get(), self.Address2\_var.get(), self.PostId\_var.get(), self.Mobile\_var.get(), self.BookId\_var.get(), self.BookTitle\_var.get(), self.Author\_var.get(), self.DateBorrowed\_var.get(),

self.DueDate\_var.get(), self.DaysOfBook\_var.get(), self.LaterReturnFine\_var.get(), self.DateOverDue\_var.get(), self.ActualPrice\_var.get(),

))

conn.commit() self.fetch\_data() conn.close()

messagebox.showinfo("Success","Member Has Been Inserted Sucessfully")

def update(self):

conn = mysql.connector.connect(host="localhost", port=3306, username="root", password="Myfirstproject@123", database="librarydata")

my\_cursor=conn.cursor()

my\_cursor.execute("UPDATE library SET Membertype=%s, IDNo=%s, FirstName=%s, LastName=%s, Address1=%s, Address2=%s, PostId=%s, Mobile=%s, BookId=%s, BookTitle=%s, Author=%s, DateBorrowed=%s, DueDate=%s, DaysOfBook=%s, LaterReturnFine=%s, DateOverDue=%s, ActualPrice=%s WHERE PRNNo=%s", (

self.MemberType\_var.get(), self.IDNo\_var.get(), self.FirstName\_var.get(), self.LastName\_var.get(), self.Address1\_var.get(), self.Address2\_var.get(), self.PostId\_var.get(), self.Mobile\_var.get(), self.BookId\_var.get(), self.BookTitle\_var.get(), self.Author\_var.get(),

self.DateBorrowed\_var.get(), self.DueDate\_var.get(), self.DaysOfBook\_var.get(), self.LaterReturnFine\_var.get(), self.DateOverDue\_var.get(), self.ActualPrice\_var.get(), self.PRNNo\_var.get(),

))

conn.commit() self.fetch\_data() self.reset() conn.close()

messagebox.showinfo("Success","Member Has Been Updated Sucessfully")

def fetch\_data(self):

conn = mysql.connector.connect(host="localhost", port=3306, username="root", password="Myfirstproject@123", database="librarydata")

my\_cursor=conn.cursor() my\_cursor.execute("select \* from library ") rows=my\_cursor.fetchall()

if len(rows)!=0: self.library\_table.delete(\*self.library\_table.get\_children()) for i in rows:

self.library\_table.insert("", END, values=i) conn.commit()

conn.close()

def get\_cursor(self,event=""): cursor\_rows=self.library\_table.focus() content=self.library\_table.item(cursor\_rows) row=content['values'] self.MemberType\_var.set(row[0]) self.PRNNo\_var.set(row[1]) self.IDNo\_var.set(row[2]) self.FirstName\_var.set(row[3]) self.LastName\_var.set(row[4]) self.Address1\_var.set(row[5]) self.Address2\_var.set(row[6]) self.PostId\_var.set(row[7]) self.Mobile\_var.set(row[8]) self.BookId\_var.set(row[9]) self.BookTitle\_var.set(row[10]) self.Author\_var.set(row[11]) self.DateBorrowed\_var.set(row[12]) self.DueDate\_var.set(row[13]) self.DaysOfBook\_var.set(row[14]) self.LaterReturnFine\_var.set(row[15]) self.DateOverDue\_var.set(row[16]) self.ActualPrice\_var.set(row[17])

def showData(self):

self.txtBox.insert(END,"Member Type:\t\t"+ self.MemberType\_var.get()+"\n") self.txtBox.insert(END,"ID No:\t\t"+ self.IDNo\_var.get()+"\n") self.txtBox.insert(END,"PRN No:\t\t"+ self.PRNNo\_var.get()+"\n") self.txtBox.insert(END,"First Name:\t\t"+ self.FirstName\_var.get()+"\n")

self.txtBox.insert(END,"Last Name:\t\t"+ self.LastName\_var.get()+"\n") self.txtBox.insert(END,"Address 1:\t\t"+ self.Address1\_var.get()+"\n") self.txtBox.insert(END,"Address 2:\t\t"+ self.Address2\_var.get()+"\n") self.txtBox.insert(END,"PostId:\t\t"+ self.PostId\_var.get()+"\n") self.txtBox.insert(END,"Mobile:\t\t"+ self.Mobile\_var.get()+"\n") self.txtBox.insert(END,"Book Id:\t\t"+ self.BookId\_var.get()+"\n") self.txtBox.insert(END,"Book Title:\t\t"+ self.BookTitle\_var.get()+"\n") self.txtBox.insert(END,"Author:\t\t"+ self.Author\_var.get()+"\n") self.txtBox.insert(END,"Date Borrowed:\t\t"+ self.DateBorrowed\_var.get()+"\n") self.txtBox.insert(END,"Due Date:\t\t"+ self.DueDate\_var.get()+"\n") self.txtBox.insert(END,"Days Of Book\t\t"+ self.DaysOfBook\_var.get()+"\n") self.txtBox.insert(END,"Later Return Fine:\t\t"+ self.LaterReturnFine\_var.get()+"\n") self.txtBox.insert(END,"Date Over Due:\t\t"+ self.DateOverDue\_var.get()+"\n") self.txtBox.insert(END,"Actual Price:\t\t"+ self.ActualPrice\_var.get()+"\n")

def reset(self): self.MemberType\_var.set(""), self.IDNo\_var.set(""),

self.PRNNo\_var.set(""), self.FirstName\_var.set(""), self.LastName\_var.set(""), self.Address1\_var.set(""), self.Address2\_var.set(""), self.PostId\_var.set(""), self.Mobile\_var.set(""), self.BookId\_var.set(""), self.BookTitle\_var.set(""), self.Author\_var.set(""), self.DateBorrowed\_var.set(""), self.DueDate\_var.set(""), self.DaysOfBook\_var.set(""), self.LaterReturnFine\_var.set(""),

self.DateOverDue\_var.set(""), self.ActualPrice\_var.set(""), self.txtBox.delete("1.0",END)

def iExit(self):

iExit=tkinter.messagebox.askyesno("Library management system", "Do you want to exit") if iExit>0:

self.root.destroy() return

def delete(self):

if self.PRNNo\_var.get()=="" or self.IDNo\_var.get()=="": messagebox.showerror("Error","First Select The Member")

else:

conn = mysql.connector.connect(host="localhost", port=3306, username="root", password="Myfirstproject@123", database="librarydata")

my\_cursor=conn.cursor()

query="delete from library where PRNNo=%s" value=(self.PRNNo\_var.get(),) my\_cursor.execute(query,value)

conn.commit() self.fetch\_data() self.reset() conn.close()

messagebox.showinfo("Success","Member Has Been Deleted Sucessfully")

if name == " main ": login\_root = Tk()

login = LoginWindow(login\_root) login\_root.mainloop()

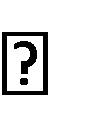
PROGRAM CODE

**Steps:**

* + Open VIRTUAL STUDIO
  + In the main file, click on start on python project and MySQL Workbench.
  + MySQL and Python both should be started
  + Type admin/admin to login.
* **Setup Project**
  + To start the project make a folder with any name.
  + Start adding files in your folder.
  + Start coding.
  + Be sure to give appropriate and correct name and extension to your file.

# FUTURE ENHANCEMENTS

The software developed as part of the project can further be enhanced as follows:

* + This project can be implemented online using some web technology
  + Number of users can be increased with different levels of access.
  + Security can be improved by providing data encryption. Monthly updates will be provided.

**Implementation and Testing**

Testing Approach

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Deless free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Boy and Black Box Testing.

In simple terms, Software Testing means Verification of Application Under Test (AUT).

Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.

Nissan cars have to recall over 1 million cars from the market due to software failure in the airbag sensory detectors. There has been reported two accident due to this software failure.

Starbucks was forced to close about 60 percent of stores in the US and Canada duc to software failure in its POS system. At one point store served coffee for free as they unable to process the transaction.

Some of the Amazon's third party retailers saw their product price is reduced to Ip due to a software glitch. They were left with heavy losses.

Vulnerability in Window 10. This bug enables users to escape from security sandboxes through a flaw in the win32k system.

In 2015 fighter plane F-35 fell victim to a software bug, making it unable to detect targets correctly.

China Airlines Airbus A300 crashed due to a software bug on April 26, 1994, killing 264 innocent live ln 1985, Canada's Therac-25 radiation therapy machine malfunctioned due to software bug and delivered lethal radiation doses to patients, leaving 3 people dead and critically injuring 3 others.

In April of 1999, a software bug caused the failure of a $1.2 billion military satellite launch, the costliest accident in history.

**Types of Software Testing**

Typically Testing is classified into three categories.

**Functional Testing**

In my project Functional testing will be done to ensure that every component are functioning properly. So that student are provided with the correct information

#### Non-Functional Testing or Performance Testing

Non-functional testing is defined as a type of Software testing to check non- functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing

#### Maintenance (Regression and Maintenance)

once the Restaurant Billing System come into existence, its maintenance will seen by the admin, to ensure that all the time customer get correct information.

## Conclusion

The project titled as **" Library Management System"** is desktop based application. This software provides facility for create, update and delete the student and books details. The system is developed with modular approach. All modules in the system have been tasted with the valid data and invalid data and everything work successfully Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This system has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Been made to upgrade the software.

## Bibliography

[www.academia.edu.com](http://www.academia.edu.com/) [www.slideshare.com](http://www.slideshare.com/) [www.geeksforgeeks.org](http://www.geeksforgeeks.org/) [www.quera.com](http://www.quera.com/) [www.](http://www/) wikipedia.com