

A
PROJECT
ON

“Library Management System”

Submitted to
Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)

In the Partial Fulfillment of
B.Com. (Computer Application) Final Year

Submitted by
Pratik Tarachandrao Dhale
Ketan Anilrao Mahalle

Under the Guidance of
Pravin J. Yadao



Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)
2021-2022

Shiksha Mandal's

**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)**

(2021 - 2022)

This is to certify that Mr. Pratik Tarachandrao Dhale & Ketan Anilrao Mahalle has completed their project on the topic of "LIBRARY MANAGEMENT SYSTEM" prescribed by G. S. College of Commerce & Economics, Nagpur (Autonomous) for B.Com. (Computer Application) – Semester-VI.

Date:

Place: Nagpur

Pravin J. Yadao

Project Guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and whole hearted thanks to project guide Prof. Pravin Yadao, Coordinator for his guidance throughout this work.

We are very much thankful to him for his constant encouragement, support and kindness.

We are also grateful to our teachers Prof. Rahul Tiwari, Prof. Sushma Gawande, Prof. Preeti Rangari, Prof. Prajkta Deshpande and Prof. Haresh Naringe for their encouragement, help and support from time to time.

We also wish to express our sincere thanks to Principal Dr. N. Y. Khandait for providing us wide range of opportunities, facilities and inspiration to gather professional knowledge and material without which this project could not have been completed.

1) Pratik Tarachandrao Dhale

2) Ketan Anilrao Mahalle

Date:

Place: Nagpur

DECLARATION

We **Pratik Tarachandrao Dhale, Ketan Anilrao Mahalle** hereby honestly declare that the work entitled “**LIBRARY MANAGEMENT SYSTEM**” submitted by us at G. S. College of Commerce & Economics, Nagpur (Autonomous) in partial fulfillment of requirement for the award of B.Com. (Computer Application) degree by Rashtrasant Tukadoji Maharaj, Nagpur University, Nagpur has not been submitted elsewhere for the award of any degree, during the academic session 2021-2022.

The project has been developed and completed by us independently under the supervision of the subject teacher and project guide.

1) Pratik Tarachandrao Dhale

2) Ketan Anilrao Mahalle

Date:

Place: Nagpur

INDEX

Sr. No.	Particulars	Page No.	Signature	Remarks
1.	Introduction	6 to 8		
2.	Objectives	9 to 11		
3.	Preliminary System Analysis 3.1 Preliminary Investigation 3.2 Present System in Use 3.3 Flaws in Present System 3.4 Need for New System 3.5 Feasibility Study	12 to 17		
4.	Project Category	18 to 21		
5.	Software and Hardware Requirement Specification	22 to 23		
6.	Detailed System Analysis 6.1 Data Flow Diagram 6.2 Numbers of Modules and Process Logic 6.3 Data Structures and Tables 6.4 Entity Relationship Diagram	24 to 29		
7.	System Design 7.1 Form Design 7.2 Source Code 7.3 Input Screen and Output Screen	30 to 69		
8.	Testing and Validation Checks	70 to 75		
9.	System Security Measures	76 to 77		
10.	Implementation, Evaluation and Maintenance	78 to 80		
11.	Future Scope of Project	81 to 82		
12.	Conclusion	83		
13.	Bibliography and References	84		

INTRODUCTION

INTRODUCTION

The "Library Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Library Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Books, Student, Librarian, Address, Member. Every Library Management System has different Student needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times.

The Application includes:

- Login Form
- Admin Page
- Student Information
- Book Information
- Issuing Books
- Borrowing Books

These systems will ultimately allow you to better manage resources.

OBJECTIVES

OBJECTIVES

The main objective of the Project on Library Management System is to manage the details of Student, Books, Issues, Librarian, Member. It manages all the information about Student, Address of Member as well as Student. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build a web-based application program to reduce the manual work for managing the Student, Books, Address, Issues. It tracks all the details about the Issues, Librarian, Member, etc.

1. The objectives of a library management system is to operate a library with efficiency and at reduced costs. The system being entirely automated streamlines all the tasks involved in operations of the library.
2. The activities of book purchasing, cataloging, indexing, circulation recording and stock checking are done by the software. Such software eliminates the need for repetitive manual work and minimizes the chances of errors.
3. The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationery. This leads to lower operational costs.
4. The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books.
5. Adding, removing or editing the database is a simple process. Adding new members or cancelling existing memberships can be done with ease.

- 6.** Stock checking and verification of books in the library can be done within a few hours. The automated system saves a considerable amount of time as opposed to the manual system.
- 7.** The library management system software makes the library a smart one by organizing the books systematically by author, title and subject. This enables users to search for books quickly and effortlessly.
- 8.** Students need access to authentic information. An advanced organized library is an integral part of any educational institution.
- 9.** In this digital age a web-based library management system would be ideal for students who can access the library's database on their smartphones.
- 10.** The main objective of the Project of Library Management System is to manage the details of users as well as books.
- 11.** It also manages all the information about Student, Address of Member as well as Student.

PRELIMINARY
SYSTEM
ANALYSIS

Preliminary Investigation:

The first step in the System Development Life Cycle (SDLC) is the identification of the need. This is the user request to change, improve or enhance an existing system. Because there is likely to be a stream of such requests, standard procedures must be established to deal with them. The initial investigation is one way of this solutions. In this process, the development team visit the customer and studies their system. They investigate the need for possible software automation of the given system by the end of the preliminary investigation. Our project furnish the document that hold the different recommendation of the needs of the user.

Present System In Use:

Present system comprises all the information related to books such as book name, author and edition. Current system also contains information about the students and books. It contains registration option to users. Without registration nobody can use this system.

Flaws In Present System:

Present system has various limitations. Present System is difficult to use. User interface not shown properly. Fast report generations is not possible. Tracing a book is difficult. Security issues occurring sometimes. Information about issue or return of the book books are not properly maintained.

Need For New System:

A current system is not compatible because the current system is a bit complicated to use. The new system is designed to be easy for the user to use. Admin can get all the information about books and students according to the new system. The new system includes student's mobile number, issuing and borrowing date of the books, fine if the student does not return the book within the given period, etc. The aim of new system is to develop a system of improved facilities. The new system can overcome all the limitations of the existing system.

Feasibility Study

Feasibility Study

Feasibility study of a system means whether the system is practically possible to build or not. It also evaluates the benefits of the new system. A feasibility study is an analysis of how a successfully a project can be completed, according factors that affect is such as Economical, behavioral, Technical and Operational. Project Manager use feasibility studies to determine potential positive and negative outcomes of the project before investing a consideration of amount of time and money into it.

The software that has to be developed is analyze in details and the system which is to be developed in technically, operationally and economically feasible or not is taken care of. The feasibility study means not to solve the problems completely but also to acquire the scope and work ability of the problem by giving various solutions to give problem and picking up one of the best solutions.

- **Technical Feasibility:**

Technical Feasibility one of the studies that must be conducted after a project has been identified. Technical Feasibility means to solve the problems related to hardware and software. It refers to the technical resources needed to develop the new application. The analyst must find out whether current technologies are sufficient for proposed system. In “Library Management System” web-based application is developed in Microsoft Visual Studio Code, which can be easily run on any system with the required configuration.

- **Behavioral Feasibility:**

It is natural observation that people are resistant to change and computers have know to facilitate change. When the user system has been developed or when a step is taken to convert a manual system to the computerized system, it is a significant factor to know the reaction of the user staff as they are once who will judge the working of the

new system. It is checked as to what percentage of staff members are against the change and one who support it.

- **Operational Feasibility:**

Operational feasibility ascertains how well the implementation of a project fits in with the current organizational structure. The solutions to a current problem must come as close as possible to a perfect fit with the organizational structure and be able to be applied to solve other arising problems. The opportunities that come along the way during the solution implementation must be able to be harnessed for even easier implementation.

PROJECT CATEGORY

Project Category

A category of project is web-based application named “Library Management System”. Login Form and Main Window are designed by using Python language with Tkinter module. MySQL are used for the database connection with python.

Programming Language and Database Connectivity Used In Project:

Python:-

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehension, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0,

released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

MySQL

MySQL is a Relational Database Management System (RDBMS) developed by Oracle that is based on Structured Query Language (SQL). MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it's clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL. With MySQL, even those new to relational systems can immediately build fast, powerful, and secure data storage systems. MySQL's programmatic syntax and interfaces are also perfect gateways into the wide world of other popular query languages and structured data stores.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to actually implement, manage, and query such a database.

MySQL is integral to many of the most popular software stacks for building and maintaining everything from customer-facing web applications to powerful, data-driven B2B services. Its open-source nature, stability, and rich feature set, paired with ongoing development and support from Oracle, have meant that internet-critical organizations such as Facebook, Twitter, Wikipedia, and YouTube all employ MySQL backends.

SOFTWARE
AND
HARDWARE
REQUIREMENT
SPECIFICATION

Hardware

Hardware is a term that refers to all the physical parts that make up a computer.

The internal hardware devices that make up the computer. Various devices which are essentials to form a hardware is called as components.

Following are the hardware specifications that is required to develop this project is as follows:

1. Computer components like Monitor, Keyboard, Mouse, CPU, Keyboard.
2. Minimum 1 GB ram for smooth working of application.
3. 250 GB Hard Disk or More. CD ROM Drive.

Software

Computer software, or simply software, is a collection of data or computer instructions that tell the computer how to work. This is in contrast to physical hardware, from which the system is build and actually performs the work.

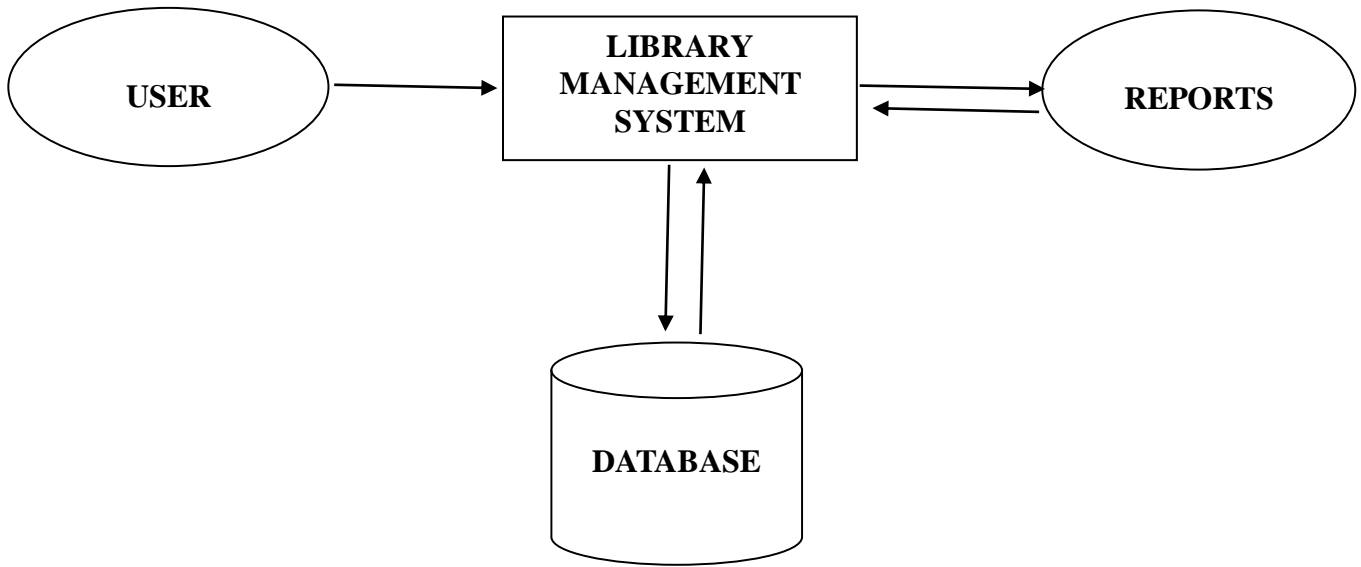
- **Front End-** Python
- **Back End-** MySQL Workbench
- **Text Editors-** VS Code Editor

Operating System

- **Windows10**
- **Ubuntu Linux**

Detailed System Analysis

Data Flow Diagram



Number of Modules And Process Logic

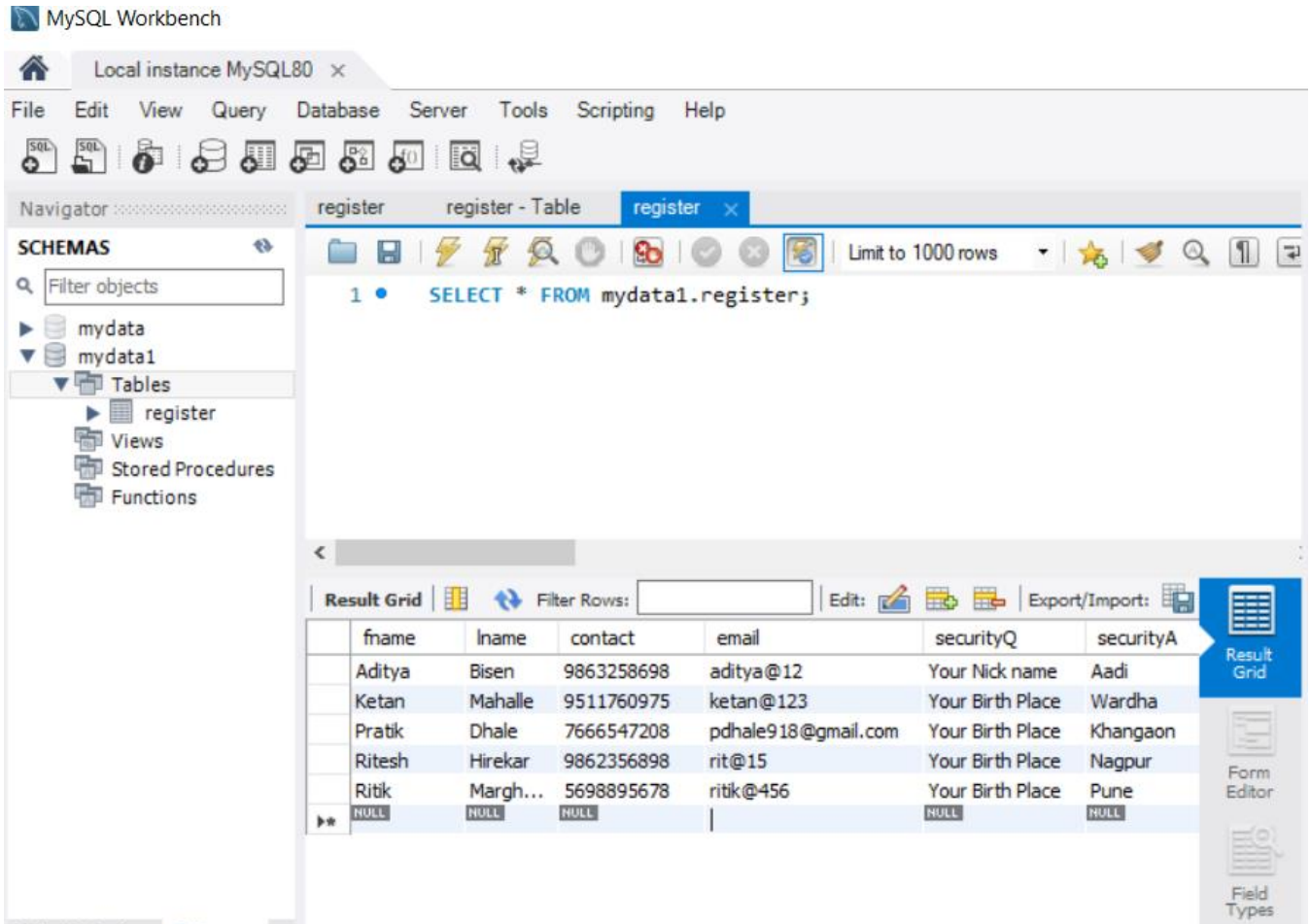
- **Modules used in Project:**

```
from tkinter import*  
import tkinter  
  
from tkinter import ttk  
from PIL import Image,ImageTk  
from tkinter import messagebox  
import random  
import time  
import datetime  
import mysql.connector  
from time import strftime
```

Data Structures and Tables

1) Registration:

This is a MySQL Database in which the Administrator can manage the User Registration details.

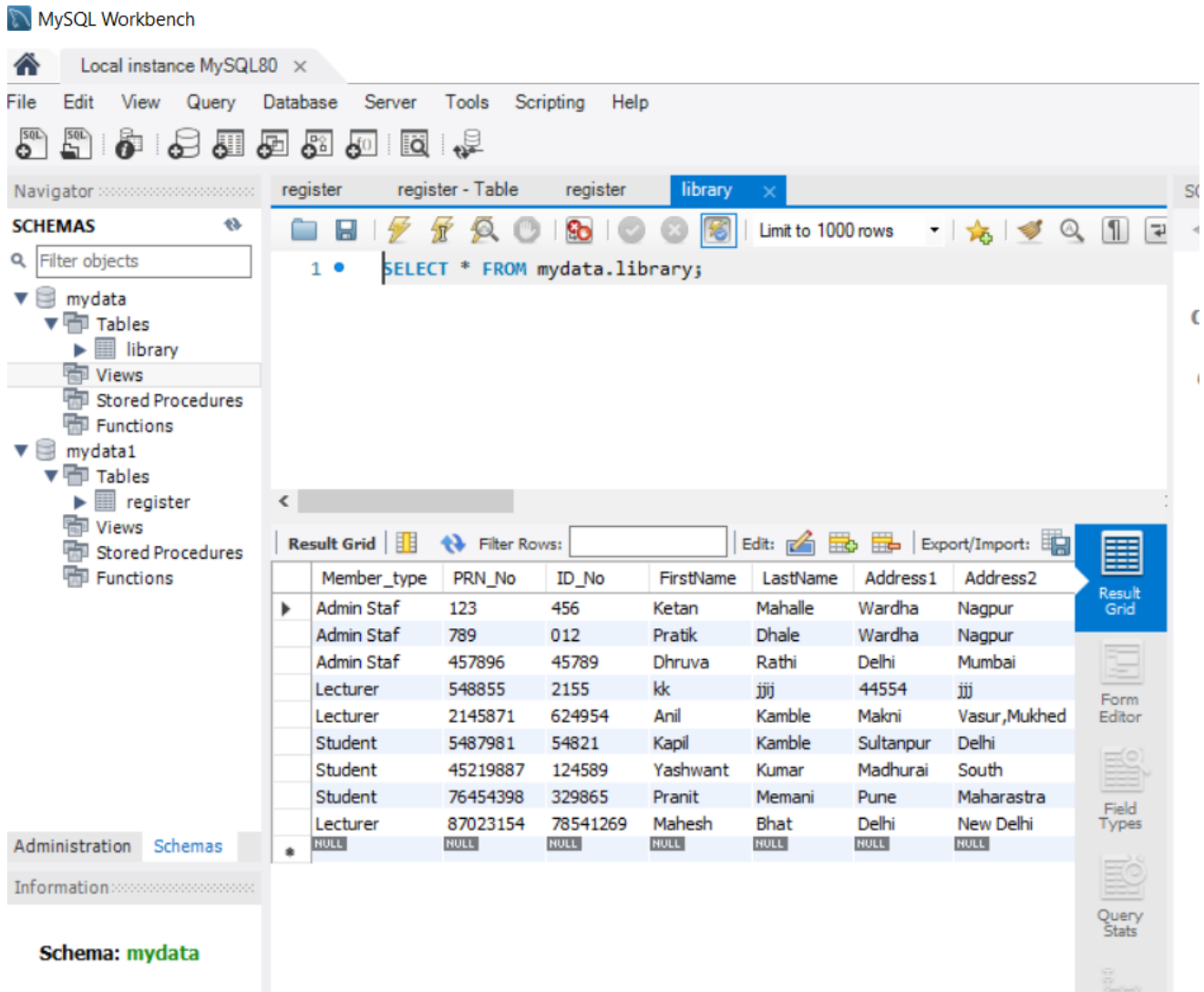


The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mydata1' expanded to show the 'register' table. The main window shows a query: `SELECT * FROM mydata1.register;` The 'Result Grid' at the bottom displays the following data:

fname	lname	contact	email	securityQ	securityA
Aditya	Bisen	9863258698	aditya@12	Your Nick name	Aadi
Ketan	Mahalle	9511760975	ketan@123	Your Birth Place	Wardha
Pratik	Dhale	7666547208	pdhale918@gmail.com	Your Birth Place	Khangaon
Ritesh	Hirekar	9862356898	rit@15	Your Birth Place	Nagpur
Ritik	Margh...	5698895678	ritik@456	Your Birth Place	Pune
NULL	NULL	NULL		NULL	NULL

2) User Details:

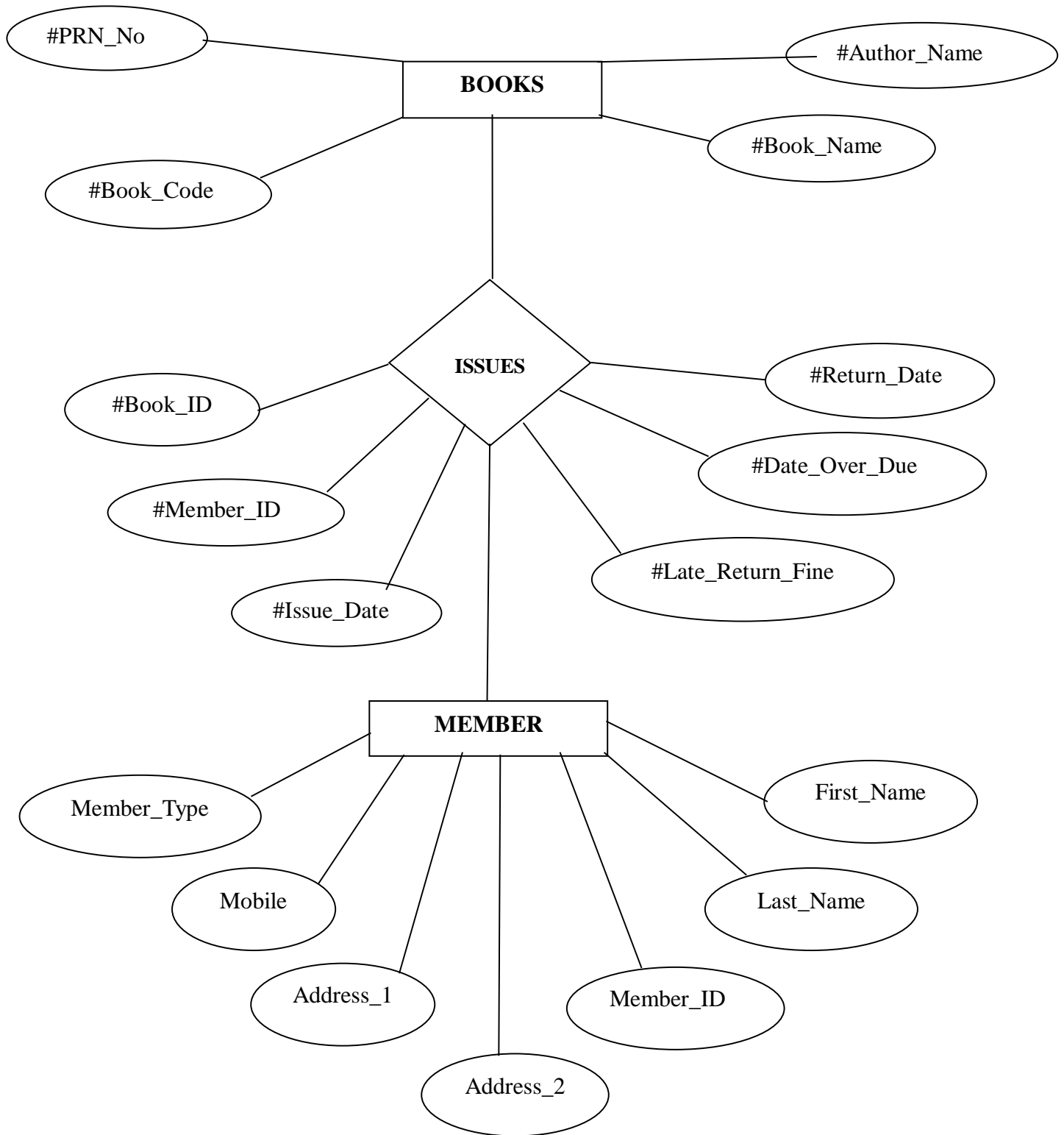
This is a MySQL Database in which the Administrator can manage the User details of Library Management System.



The screenshot displays the MySQL Workbench interface. The left sidebar shows the Schemas tree with 'mydata' selected. The main window shows a query editor with the SQL statement: `SELECT * FROM mydata.library;`. Below the editor, the Result Grid displays the following data:

	Member_type	PRN_No	ID_No	FirstName	LastName	Address1	Address2
▶	Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur
	Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur
	Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai
	Lecturer	548855	2155	kk	jjj	44554	jjj
	Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed
	Student	5487981	54821	Kapil	Kamble	Sultanpur	Delhi
	Student	45219887	124589	Yashwant	Kumar	Madhurai	South
	Student	76454398	329865	Pranit	Memani	Pune	Maharastra
	Lecturer	87023154	78541269	Mahesh	Bhat	Delhi	New Delhi
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Entity Relationship Diagram

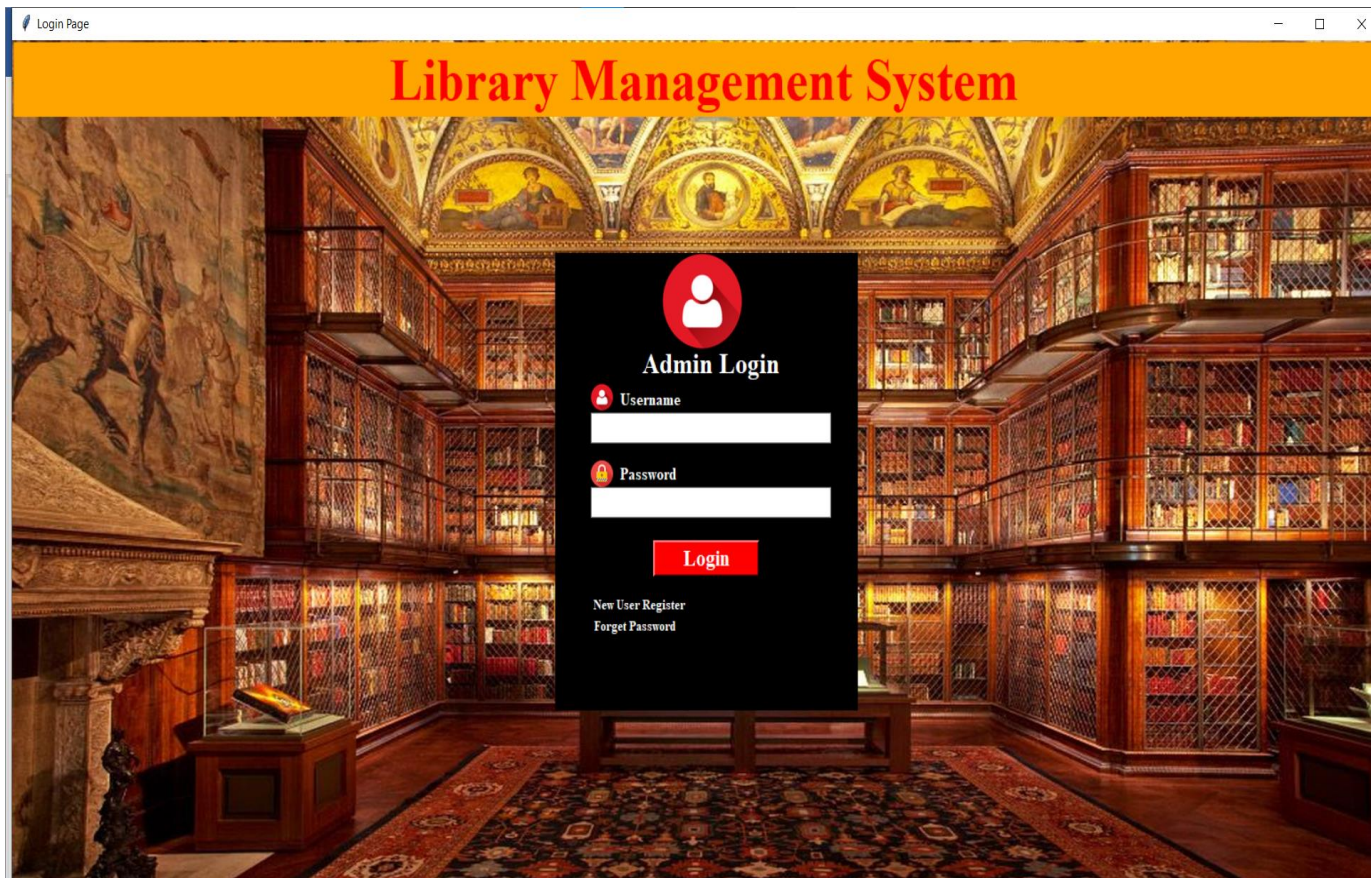


SYSTEM DESIGN

Form Design

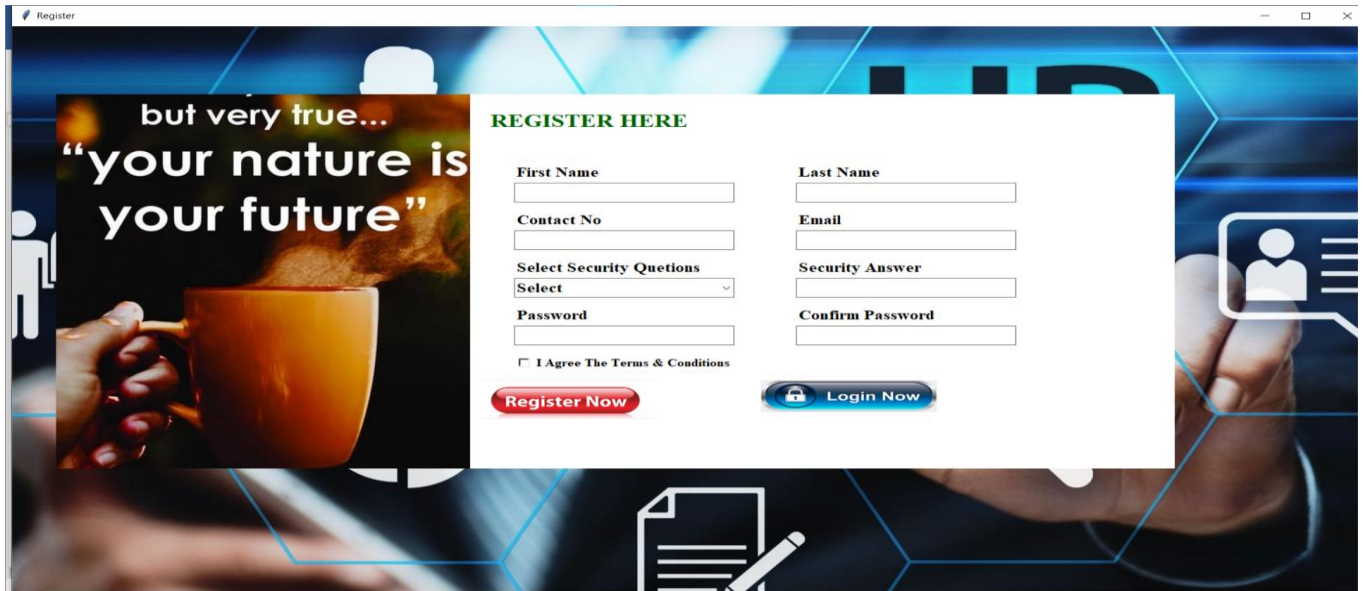
1) Admin:

This is the Library Management System Login Form. You have to enter Username and Password on it. There are also options like New User Registration and Forget Password.



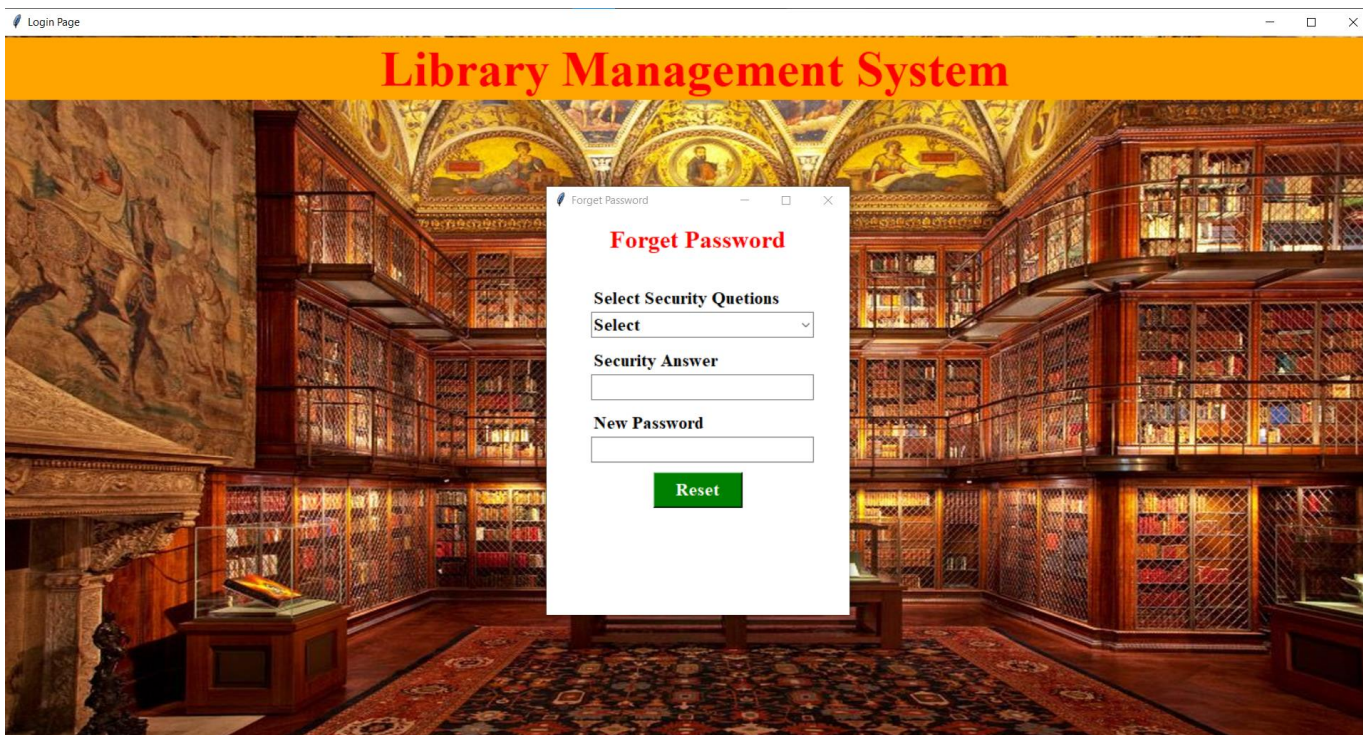
2) Register:

This is the New User Registration Form of the Library Management System. The user needs a Username and Password to login to the system, and to create it, the user has to first register from here.



3) Forget Password:

If a user Forgets his Login Password, then from here he can create his new Password. For that, the user has to enter details like his Username, Security Question and Security Answer.



4) Home:

This is the Main Window of the Library Management System. From here the Admin uses this system.

Library Management System

10:12:56 PM

LIBRARY MANAGEMENT SYSTEM

Library Membership Information

Member Type: Admin Staf

PRN No:

ID No:

FirstName:

Surname:

Address1:

Address2:

Post Code:

Mobile Number:

Book Id:

Book Title:

Auther Name:

Date Borrowed:

Date Due:

Days On Book:

Late Return Fine:

Date Over Due:

Actual Price:

Book Details

- Head Firt Book
- Learn Python The Hard
- Python Programming
- Secrete Rahshy
- Python CookBook
- Into to Machine Learnin
- Fluent Python
- programmng Python
- The Algorithm
- The tecnich Python
- Machine tecno
- My Python
- Joss Elif guru
- Elite Jungle python
- Jungll Python
- Mumbai Python

ADD DATA SHOW DATA UPDATE DELETE RESET EXIT

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guido	2022-04-22	2022-05-07	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjkj	2155488532	BKID1245	Intro to python C	John Zhelle	2021-01-20	2021-02-04	15

5) User Details:

This is the Library Management System User Details Form. This form shows the user details such as First name, Last Name, Contact, Addresses and other details to the Administrator.

10:15:09 PM

LIBRARY MANAGEMENT SYSTEM

Library Membership Information

Member Type: Admin Staf
 PRN No: 789
 ID No: 12
 FirstName: Pratik
 Surname: Dhale
 Address1: Wardha
 Address2: Nagpur
 Post Code: 442001
 Mobile Number: 7666547208

Book Id: BKID8796
 Book Title: Basic Of Pythpn
 Auther Name: ZED A. SHAW
 Date Borrowed: 2022-04-22
 Date Due: 2022-05-07
 Days On Book: 15
 Late Return Fine: Rs.25
 Date Over Due: NO
 Actual Price: Rs.725

Book Details

Head Firt Book
 Learn Python The Hard
 Python Programming
 Secrete Rahshy
 Python CookBook
 Into to Machine Learnin
 Fluent Python
 progmming Python
 The Algorithm
 The tecnich Python
 Machine tecno
 My Python
 Joss Ellif guru
 Elite Jungle python
 Jungli Python
 Mumbai Python

Member Type: Admin Staf
 PRN No: 123
 ID No: 456
 FirstName: Ketan
 LastName: Mahalle
 Address1: Wardha
 Address2: Nagpur
 Post Code: 442001
 Mobile No: 9511760975
 Book ID: BKID8796
 Book Title: Intro to Machine L
 earning
 Auther: Sarah Guido
 DateBorrowed: 2022-04-22
 DateDue: 2022-05-07
 DaysOnBook: 15

ADD DATA

SHOW DATA

UPDATE

DELETE

RESET

EXIT

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guido	2022-04-22	2022-05-07	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjjk	2155488532	BKID1245	Intro to python C	John Zhelle	2021-01-20	2021-02-04	15

25 April 2022
Monday

6) Book Details:

This is Library Management System Book Details Form. This form shows the details of the Books of the Library such as the Title of the Book, Author Name, Price of the Book and other details to the Administrator.

10:18:15 PM

LIBRARY MANAGEMENT SYSTEM

Library Membership Information

Member Type: Admin Staf
 PRN No:
 ID No:
 FirstName:
 Surname:
 Address1:
 Address2:
 Post Code:
 Mobile Number:

Book Id: BKID1245
 Book Title: Intro to python Comp Science
 Auther Name: John Zhelle
 Date Borrowed: 2022-04-25
 Date Due: 2022-05-10
 Days On Book: 15
 Late Return Fine: Rs.25
 Date Over Due: NO
 Actual Price: Rs.500

Book Details

Head Firt Book
 Learn Python The Hard
 Python Programming
 Secrete Rahshy
 Python CookBook
 Into to Machine Learnin
 Fluent Python
 progmming Python
 The Algorithm
 The tecnich Python
 Machine tecno
 My Python
 Joss Ellif guru
 Elite Jungle python
 Jungli Python
 Mumbai Python

Member Type: Admin Staf
 PRN No:
 ID No:
 FirstName:
 LastName:
 Address1:
 Address2:
 Post Code:
 Mobile No:
 Book ID: BKID1245
 Book Title: Intro to python Co
 mp Science
 Auther: John Zhelle
 DateBorrowed: 2022-04-25
 DateDue: 2022-05-10
 DaysOnBook: 15

ADD DATA

SHOW DATA

UPDATE

DELETE

RESET

EXIT

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guido	2022-04-22	2022-05-07	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjjk	2155488532	BKID1245	Intro to python C	John Zhelle	2021-01-20	2021-02-04	15

Source Code

1) **Admin Login and Registration:**

```
from logging import root
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import random
import time
import datetime
from time import strftime
import mysql.connector

def main():
    win=Tk()
    app=Login_Window(win)
    win.mainloop()

class Login_Window:
    def __init__(self,root):
        self.root=root
        self.root.title("Login Page")
        self.root.geometry("1550x800+0+0")
        img1 = Image.open("images/Library_Image.png")
        img1 = img1.resize((1530,800), Image.ANTIALIAS)

        self.photoImg1 = ImageTk.PhotoImage(img1)
        bg_lbl=Label(self.root,image=self.photoImg1)
```

```

bg_lbl.place(x=0,y=0,width=1530,height=800)

title=Label(bg_lbl,text="Library Management System",font=("times new
roman",42,"bold"),bg="orange",fg="red")
title.place(x=0,y=0,width=1550,height=70)

frame=Frame(self.root,bg="black")
frame.place(x=610,y=200,width=340,height=430)

img1=Image.open("images/LoginIconAppl.png")
img1=img1.resize((90,90),Image.ANTIALIAS)
self.photoimage1=ImageTk.PhotoImage(img1)
lblimg1=Label(image=self.photoimage1,bg="black",borderwidth=0)
lblimg1.place(x=730,y=200,width=90,height=90)

get_str=Label(frame,text="Admin Login",font=("times new
roman",20,"bold"),fg="white",bg="black")
get_str.place(x=95,y=85)

self.txtuser=StringVar()
self.txtpass=StringVar()

username=lbl=Label(frame,text="Username",font=("times new
roman",12,"bold"),fg="white",bg="black")
username.place(x=70,y=125)

txtuser=ttk.Entry(frame,textvariable=self.txtuser,font=("times new roman",15,"bold"))
txtuser.place(x=40,y=150,width=270)

```

```
password=lbl=Label(frame,text="Password",font=("times new  
roman",12,"bold"),fg="white",bg="black")  
password.place(x=70,y=195)
```

```
txtpass=ttk.Entry(frame,textvariable=self.txtpass,font=("times new  
roman",15,"bold"),show="*")  
txtpass.place(x=40,y=220,width=270)
```

```
img2=Image.open("images/LoginIconAppl.png")  
img2=img2.resize((25,25),Image.ANTIALIAS)  
self.photoimage2=ImageTk.PhotoImage(img2)  
lblimg1=Label(image=self.photoimage2,bg="black",borderwidth=0)  
lblimg1.place(x=650,y=323,width=25,height=25)
```

```
img3=Image.open("images/lock-512.png")  
img3=img3.resize((25,25),Image.ANTIALIAS)  
self.photoimage3=ImageTk.PhotoImage(img3)  
lblimg1=Label(image=self.photoimage3,bg="black",borderwidth=0)  
lblimg1.place(x=650,y=395,width=25,height=25)
```

```
btn_login=Button(frame,text="Login",borderwidth=3,relief=RAISED,command=self.login,c  
ursor="hand2",font=("times new roman",16,"bold"),fg="white",bg="red"  
,activebackground="#B00857")
```

```
btn_login.place(x=110,y=270,width=120,height=35)
```

```
registerbtn=Button(frame,text="New User  
Register",command=self.register_window,font=("times new
```

```
roman",10,"bold"),borderwidth=0,fg="white",bg="black",activeforeground="white",activebackground="black")
```

```
registerbtn.place(x=15,y=320,width=160)
```

```
forgetbtn=Button(frame,text="Forget Password",command=self.forgot_password_window,font=("times new roman",10,"bold"),borderwidth=0,fg="white",bg="black",activeforeground="white",activebackground="black")
```

```
forgetbtn.place(x=10,y=340,width=160)
```

```
def register_window(self):
```

```
self.new_window=Toplevel(self.root)
```

```
self.app=Register( self.new_window)
```

```
def login(self):
```

```
if self.txtuser.get()=="Pratik" and self.txtpass.get()=="Ketan":
```

```
messagebox.showinfo("Success","Welcome to Library Management System...")
```

```
self.new_window=Toplevel(self.root)
```

```
self.app=LibraryManagementSystem(self.new_window)
```

```
elif self.txtuser.get()=="" or self.txtpass.get()=="":
```

```
messagebox.showerror("Error","all field required")
```

```
else:
```

```
conn=mysql.connector.connect(host="localhost",user="root",password="Ketan@16m",database="mydata")
```

```
my_cursor=conn.cursor()
```

```
my_cursor.execute("select * from register where email=%s and password=%s",(  
self.txtuser.get(),
```

```

        self.txtpass.get()
    ))

row=my_cursor.fetchone()
if row==None:
    messagebox.showerror("Error","Invalid Username & password")
else:
    open_main=messagebox.askyesno("YesNo","Enter Library Management System")
    if open_main>0:
        pass
        self.new_window=Toplevel(self.root)
        self.app=LibraryManagementSystem(self.new_window)
    else:
        if not open_main:
            return
    conn.commit()
    self.clear()
    conn.close()
def clear(self):
    self.txtuser.set("")
    self.txtpass.set("")
def reset_pass(self):
    if self.combo_securiy_Q.get()=="Select" or self.txt_security.get()=="" or
self.txt_newpass=="":
        messagebox.showerror("Error","All fields are required",parent=self.root2)
    else:
        try:

```

```

conn=mysql.connector.connect(host="localhost",user="root",password="Ketan@16m",database="mydata")
    cur=conn.cursor()
    query=("select * from register where email=%s and securityQ=%s and securityA=%s")
    value=(self.txtuser.get(),self.combo_securiy_Q.get(),self.txt_security.get(),)
    cur.execute(query,value)
    row=cur.fetchone()
    if row==None:
        messagebox.showerror("Error","Please select the correct security quetion/Enter answer",parent=self.root2)
    else:
        query=("update register set password=%s where email=%s")
        value=(self.txt_newpass.get(),self.txtuser.get())
        cur.execute(query,value)
        conn.commit()
        conn.close()
        messagebox.showinfo("Success","Your password has been reset,Please login with new password",parent=self.root2)
        self.root2.destroy()
        self.txtuser.focus()
    except Exception as es:
        messagebox.showerror("Error",f"Error Due To:{str(es)}",parent=self.root2)
def forgot_password_window(self):
    if self.txtuser.get()=="":
        messagebox.showerror("Error","Plaese Enter the Email address to reset password")
    else:

```

```

conn=mysql.connector.connect(host="localhost",user="root",password="Ketan@16m",database="mydata")

my_cursor=conn.cursor()
query=("select * from register where email=%s")
value=(self.txtuser.get(),)
my_cursor.execute(query,value)
row=my_cursor.fetchone()
if row==None:
    messagebox.showerror("My Error","Plaese enter the valid user name")
else:
    conn.close()
    self.root2=Toplevel()
    self.root2.title("Forget Password")
    self.root2.geometry("340x450+610+200")
    self.root2.configure(bg="white")
    l=Label(self.root2,text="Forget Password",font=("times new
roman",20,"bold"),fg="red",bg="white")
    l.place(x=0,y=10,relwidth=1)
    security_Q=Label(self.root2,text="Select Security Qutions",font=("times new
roman",15,"bold"),bg="white",fg="black")
    security_Q.place(x=50,y=80)
    self.combo_securiy_Q=ttk.Combobox(self.root2,font=("times new
roman",15,"bold"),state="readonly")
    self.combo_securiy_Q["values"]=("Select","Your Birth Place","Your Nick
name","Your Pet Name")
    self.combo_securiy_Q.place(x=50,y=110,width=250)
    self.combo_securiy_Q.current(0)

```



```

security_A=Label(self.root2,text="Security Answer",font=("times new
roman",15,"bold"),bg="white",fg="black")
security_A.place(x=50,y=150)
self.txt_security=ttk.Entry(self.root2,font=("times new roman",15,"bold"))
self.txt_security.place(x=50,y=180,width=250)
new_password=Label(self.root2,text="New Password",font=("times new
roman",15,"bold"),bg="white",fg="black")
new_password.place(x=50,y=220)
self.txt_newpass=ttk.Entry(self.root2,font=("times new roman",15,"bold"))
self.txt_newpass.place(x=50,y=250,width=250)
btn=Button(self.root2,text="Reset",command=self.reset_pass,font=("times new
roman",15,"bold"),fg="White",bg="green")
btn.place(x=120,y=290,width=100)

```

class Register:

```

def __init__(self,root):
    self.root=root
    self.root.title("Register")
    self.root.geometry("1600x900+0+0")

    self.var_fname=StringVar()
    self.var_lname=StringVar()
    self.var_contact=StringVar()
    self.var_email=StringVar()
    self.var_securityQ=StringVar()
    self.var_SecurityA=StringVar()
    self.var_pass=StringVar()
    self.var_confpass=StringVar()

```

```

self.bg=ImageTk.PhotoImage(file="images/employee_img2.jpg")
bg_lbl=Label(self.root,image=self.bg)
bg_lbl.place(x=0,y=0,relwidth=1,relheight=1)

self.bg1=ImageTk.PhotoImage(file="images/thought-good-morning-messages-
LoveSove.jpg")
left_lbl=Label(self.root,image=self.bg1)
left_lbl.place(x=50,y=100,width=470,height=550)

frame=Frame(self.root,bg="white")
frame.place(x=520,y=100,width=800,height=550)

register_lbl=Label(frame,text="REGISTER HERE",font=("times new
roman",20,"bold"),fg="darkgreen",bg="white")

fname=Label(frame,text="First Name",font=("times new
roman",15,"bold"),bg="white")
fname.place(x=50,y=100)

self.fname_entry=ttk.Entry(frame,textvariable=self.var_fname,font=("times new
roman",15,"bold"))
self.fname_entry.place(x=50,y=130,width=250)

l_name=Label(frame,text="Last Name",font=("times new
roman",15,"bold"),bg="white",fg="black")
l_name.place(x=370,y=100)

self.txt_lname=ttk.Entry(frame,textvariable=self.var_lname,font=("times new
roman",15,"bold"))

```

```

self.txt_lname.place(x=370,y=130,width=250)
contact=Label(frame,text="Contact No",font=("times new
roman",15,"bold"),bg="white",fg="black")
contact.place(x=50,y=170)
self.txt_contact=ttk.Entry(frame,textvariable=self.var_contact,font=("times new
roman",15,"bold"))
self.txt_contact.place(x=50,y=200,width=250)
email=Label(frame,text="Email",font=("times new
roman",15,"bold"),bg="white",fg="black")
email.place(x=370,y=170)
self.txt_email=ttk.Entry(frame,textvariable=self.var_email,font=("times new
roman",15,"bold"))
self.txt_email.place(x=370,y=200,width=250)
security_Q=Label(frame,text="Select Security Qestions",font=("times new
roman",15,"bold"),bg="white",fg="black")
security_Q.place(x=50,y=240)

self.combo_securiy_Q=ttk.Combobox(frame,textvariable=self.var_securityQ,font=("times
new roman",15,"bold"),state="readonly")
self.combo_securiy_Q["values"]=("Select","Your Birth Place","Your Nick name","Your
Blood Group")
self.combo_securiy_Q.place(x=50,y=270,width=250)
self.combo_securiy_Q.current(0)
security_A=Label(frame,text="Security Answer",font=("times new
roman",15,"bold"),bg="white",fg="black")
security_A.place(x=370,y=240)
self.txt_security=ttk.Entry(frame,textvariable=self.var_SecurityA,font=("times new
roman",15,"bold"))
self.txt_security.place(x=370,y=270,width=250)

```

```

pswd=Label(frame,text="Password ",font=("times new
roman",15,"bold"),bg="white",fg="black")
pswd.place(x=50,y=310)
self.txt_pswd=ttk.Entry(frame,textvariable=self.var_pass,font=("times new
roman",15,"bold"))
self.txt_pswd.place(x=50,y=340,width=250)
confirm_pswd=Label(frame,text="Confirm Password",font=("times new
roman",15,"bold"),bg="white",fg="black")
confirm_pswd.place(x=370,y=310)
self.txt_confirm_pswd=ttk.Entry(frame,textvariable=self.var_confpass,font=("times new
roman",15,"bold"))
self.txt_confirm_pswd.place(x=370,y=340,width=250)
self.var_check=IntVar()
self.checkbtn=Checkbutton(frame,variable=self.var_check,text="I Agree The Terms &
Conditions",bg='white',font=("times new roman",12,"bold"),onvalue=1,offvalue=0)
self.checkbtn.place(x=50,y=380)
img=Image.open("images/register-now-button1.jpg")
img=img.resize((200,55),Image.ANTIALIAS)
self.photoimage=ImageTk.PhotoImage(img)

b1=Button(frame,image=self.photoimage,command=self.register_data,borderwidth=0,cursor
="hand2",font=("times new roman",15,"bold"),fg="white")
b1.place(x=10,y=420,width=200)
img1=Image.open("images/loginpng.png")
img1=img1.resize((200,45),Image.ANTIALIAS)
self.photoimage1=ImageTk.PhotoImage(img1)

b1=Button(frame,image=self.photoimage1,command=self.return_login,borderwidth=0,cursor
="hand2",font=("times new roman",15,"bold"),fg="white")

```

```

b1.place(x=330,y=420,width=200)
def register_data(self):
    if self.var_fname.get()=="" or self.var_email.get()=="" or
self.var_securityQ.get()=="Select":
        messagebox.showerror("Error","All fields are required",parent=self.root)
    elif self.var_pass.get()!=self.var_confpass.get():
        messagebox.showerror("Error","password & confirm password must be
same",parent=self.root)
    elif self.var_check.get()==0:
        messagebox.showerror("Error","Plaese agree our terms ane
condition",parent=self.root)
    else:

conn=mysql.connector.connect(host="localhost",user="root",password="Ketan@16m",datab
ase="mydata")
    my_cursor=conn.cursor()
    query=("select * from register where email=%s")
    value=(self.var_email.get(),)
    my_cursor.execute(query,value)
    row=my_cursor.fetchone()
    if row!=None:
        messagebox.showerror("Error","User already exist,plaese try another
email",parent=self.root)
    else:
        my_cursor.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s)",(
                                                                    self.var_fname.get(),
                                                                    self.var_lname.get(),
                                                                    self.var_contact.get(),
                                                                    self.var_email.get(),

```

```
        self.var_securityQ.get(),
        self.var_SecurityA.get(),
        self.var_pass.get()
    ))

    conn.commit()
    conn.close()
    messagebox.showinfo("Success", "Register Successfully")
def return_login(self):
    self.root.destroy()
```

2) Main Homescreen:

```
class LibraryManagementSystem:
    def __init__(self,root):
        self.root=root
        self.root.title("Library Management System")
        self.root.geometry("1550x800+0+0")
#=====Variables=====
    self.member_var=StringVar()
    self.ref_var=StringVar()
    self.title_var=StringVar()
    self.firstname_var=StringVar()
    self.lastname_var=StringVar()
    self.address1_var=StringVar()
    self.address2_var=StringVar()
    self.postcode_var=StringVar()
    self.mobile_var=StringVar()
    self.bookid_var=StringVar()
    self.booktitle_var=StringVar()
    self.auther_var=StringVar()
    self.dateborrowed_var=StringVar()
    self.datedue_var=StringVar()
    self.daysonbook=StringVar()
    self.lateratefine_var=StringVar()
    self.dateoverdue=StringVar()
    self.finallprice=StringVar()
```

```

#-----TitleLabel-----
    lbltitle=Label(self.root,text="LIBRARY MANAGEMENT
SYSTEM",bg="white",fg="crimson",bd=20,relief=RIDGE,font=("times new
roman",50,"bold"),padx=2,pady=6)
    lbltitle.pack(side=TOP,fill=X)
    def time():
        string = strftime('%I:%M:%S %p')
        lbl.config(text = string)
        lbl.after(1000, time)
    lbl = Label(lbltitle, font = ('times new roman',15, 'bold'),background =
'purple',foreground = 'white')
    lbl.place(x=0,y=0,width=150)
    time()
#-----Dataframe-----
    DataFrame=Frame(self.root,bd=20,padx=20,relief=RIDGE)
    DataFrame.place(x=0,y=130,width=1530,height=400)

    DataFrameLeft=LabelFrame(DataFrame,bd=12,padx=20,relief=RIDGE,fg="darkgreen",
        font=("arial",12,"bold"),text="Library Membership
Information")
    DataFrameLeft.place(x=0,y=5,width=900,height=350)

    DataFrameRight=LabelFrame(DataFrame,bd=12,padx=20,relief=RIDGE,fg="darkgreen",
        font=("arial",12,"bold"),text="Book Details")
    DataFrameRight.place(x=910,y=5,width=540,height=350)

#-----Buttonframe-----
    ButtonFrame=Frame(self.root,bd=20,padx=20,relief=RIDGE)

```



```
ButtonFrame.place(x=0,y=530,width=1530,height=70)
```

```
# =====ButtonFrame=====
```

```
btnAddData=Button(ButtonFrame,command=self.add_data,text="ADD  
DATA",font=("arial",12,"bold"),width=23,bg="crimson",fg="white")
```

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

```
btnShowData=Button(ButtonFrame,command=self.showData,text="SHOW  
DATA",font=("arial",12,"bold"),width=23,bg="crimson",fg="white")
```

```
btnShowData.grid(row=0,column=1)
```

```
btnUpdate=Button(ButtonFrame,command=self.update_data,text="UPDATE",font=("arial",12  
,"bold"),width=23,bg="crimson",fg="white")
```

```
btnUpdate.grid(row=0,column=2)
```

```
btnDelete=Button(ButtonFrame,command=self.mDelete,text="DELETE",font=("arial",12,"bol  
d"),width=23,bg="crimson",fg="white")
```

```
btnDelete.grid(row=0,column=3)
```

```
btnReset=Button(ButtonFrame,command=self.reset,text="RESET",font=("arial",12,"bold"),wi  
dth=23,bg="crimson",fg="white")
```

```
btnReset.grid(row=0,column=4)
```

```
btnExit=Button(ButtonFrame,command=self.iExit,text="EXIT",font=("arial",12,"bold"),width  
=23,bg="crimson",fg="white")
```

```
btnExit.grid(row=0,column=5)
```

```
#
```

```
=====Framedetails=====
```

```
=====
```

```
FrameDetails=Frame(self.root,bd=20,padx=20,relief=RIDGE)
```

```
FrameDetails.place(x=0,y=600,width=1530,height=195)
```

```
lblMember=Label(DataFrameLeft,font=("arial",12,"bold"),text="Member  
Type",padx=2,pady=6)
```

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

```
comMember=ttk.Combobox(DataFrameLeft,textvariable=self.member_var,state="readonly",  
font=("arial",12,"bold"),width=27)
```

```
comMember['value']=("Admin Staf","Lecturer","Student")
```

```
comMember.current(0)
```

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

```
lblref=Label(DataFrameLeft,font=("arial",12,"bold"),text="PRN No:",padx=2)
```

```
lblref.grid(row=1,column=0,sticky=W)
```

```
txtref=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.ref_var,width=29)
```

```
txtref.grid(row=1,column=1)
```

```
lblTitle=Label(DataFrameLeft,font=("arial",12,"bold"),text="ID No:",padx=2,pady=4)
```

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

```
txtTitle=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.title_var,width=29)
```

```
txtTitle.grid(row=2,column=1)
```

```
lblFirstName=Label(DataFrameLeft,font=("arial",12,"bold"),text="FirstName:",padx=2,pady=6)
```

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

```
txtFirstName=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.firstname_var,width=29)
```

```
    txtFirstName.grid(row=3,column=1)
```

```
lblLastName=Label(DataFrameLeft,font=("arial",12,"bold"),text="Surname:",padx=2,pady=6)
```

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

```
txtLastName=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.lastname_var,width=29)
```

```
    txtLastName.grid(row=4,column=1)
```

```
lblAddress1=Label(DataFrameLeft,font=("arial",12,"bold"),text="Address1:",padx=2,pady=6)
```

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

```
txtAddress1=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.address1_var,width=29)
```

```
    txtAddress1.grid(row=5,column=1)
```

```
lblAddress2=Label(DataFrameLeft,font=("arial",12,"bold"),text="Address2:",padx=2,pady=6)
```

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

```
txtAddress2=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.address2_var,width=29)
```

```
txtAddress2.grid(row=6,column=1)
```

```
lblPostCode=Label(DataFrameLeft,font=("arial",12,"bold"),text=" Post Code:",padx=2,pady=4)
```

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

```
txtPostCode=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.postcode_var,width=29)
```

```
txtPostCode.grid(row=7,column=1)
```

```
lblMobile=Label(DataFrameLeft,font=("arial",12,"bold"),text="Mobile Number:",padx=2,pady=6)
```

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

```
txtMobile=Entry(DataFrameLeft,font=("arial",13,"bold"),textvariable=self.mobile_var,width=29)
```

```
txtMobile.grid(row=8,column=1)
```

```
lblBookId=Label(DataFrameLeft,font=("arial",12,"bold"),text="Book Id:",padx=2)
```

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

```
txtBookId=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.bookid_var,width=29)
```

```
txtBookId.grid(row=0,column=3)
```

```
lblBookTitle=Label(DataFrameLeft,font=("arial",12,"bold"),text="Book
Title:",padx=2,pady=6)
lblBookTitle.grid(row=1,column=2,sticky=W)

txtBookTitle=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.booktitle_var,wi
dth=29)
txtBookTitle.grid(row=1,column=3)

lblAuther=Label(DataFrameLeft,font=("arial",12,"bold"),text="Auther
Name:",padx=2,pady=6)
lblAuther.grid(row=2,column=2,sticky=W)

txtAuther=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.auther_var,width=2
9)
txtAuther.grid(row=2,column=3)

lblDateBorrowed=Label(DataFrameLeft,font=("arial",12,"bold"),text="Date
Borrowed:",padx=2,pady=6)
lblDateBorrowed.grid(row=3,column=2,sticky=W)

txtDateBorrowed=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.dateborrow
ed_var,width=29)
txtDateBorrowed.grid(row=3,column=3,sticky=W)

lblDateDue=Label(DataFrameLeft,font=("arial",12,"bold"),text="Date
Due:",padx=2,pady=6)
lblDateDue.grid(row=4,column=2,sticky=W)
```

```
txtDateDue=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.datedue_var,width=29)
```

```
txtDateDue.grid(row=4,column=3)
```

```
lblDaysOnBook=Label(DataFrameLeft,font=("arial",12,"bold"),text="Days On Book:",padx=2,pady=6)
```

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

```
txtDaysOnBook=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.daysonbook,width=29)
```

```
txtDaysOnBook.grid(row=5,column=3)
```

```
lblLateReturnFine=Label(DataFrameLeft,font=("arial",12,"bold"),text="Late Return Fine:",padx=2,pady=6)
```

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

```
txtLateReturnFine=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.lateratefine_var,width=29)
```

```
txtLateReturnFine.grid(row=6,column=3)
```

```
lblDateOverdate=Label(DataFrameLeft,font=("arial",12,"bold"),text="Date Over Due:",padx=2,pady=6)
```

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

```
txtDateOverdate=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.dateoverdue,width=29)
```

```
txtDateOverdate.grid(row=7,column=3)
```

```

lblActualPrice=Label(DataFrameLeft,font=("arial",12,"bold"),text="Actual
Price:",padx=2,pady=6)
lblActualPrice.grid(row=8,column=2,sticky=W)

txtActualPrice=Entry(DataFrameLeft,font=("arial",12,"bold"),textvariable=self.finallprice,width=29)
txtActualPrice.grid(row=8,column=3)

#=====DataframeRight=====
#=====textBox=====

self.textBox=Text(DataFrameRight,font=("arial",12,"bold"),width=32,height=16,padx=2,pady=6)
self.textBox.grid(row=0,column=2)

#=====ListBox=====

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

ListOfBooks=['Head Firt Book','Learn Python The Hard Way','Python
Programming',"Secrete Rahshy",'Python CookBook','Into to Machine Learning','Fluent
Python','progrmming Python','The Algorithm','The tecnich Python',
'Machine tecno','My Python','Joss Ellif guru','Elite Jungle
python','Jungli Python','Mumbai Python','Pune Python','Guru Of Python','Yellow Dragan','Red
python',
'Machine python','Advance Python','Inton
Python','RedChilli Python','Ishq Python']

def SelectBook(event=""):
    value=str(bookList.get(bookList.curselection()))

```

```
x=value
if (x=="Head Firt Book"):
    self.bookid_var.set("BKID5487")
    self.booktitle_var.set("Python manual")
    self.auther_var.set("Paull berry")

    d1=datetime.date.today()
    d2=datetime.timedelta(days=15)
    d3=d1+d2
    self.dateborrowed_var.set(d1)
    self.datedue_var.set(d3)
    self.daysonbook.set("15")
    self.lateratefine_var.set("Rs.25")
    self.dateoverdue.set("NO")
    self.finallprice.set("Rs.375")

elif (x=="Learn Python The Hard Way"):
    self.bookid_var.set("BKID8796")
    self.booktitle_var.set("Basic Of Pythpn")
    self.auther_var.set("ZED A.SHAW")

    d1=datetime.date.today()
    d2=datetime.timedelta(days=15)
    d3=d1+d2
    self.dateborrowed_var.set(d1)
    self.datedue_var.set(d3)
    self.daysonbook.set("15")
    self.lateratefine_var.set("Rs.25")
    self.dateoverdue.set("NO")
```



```
self.finallprice.set("Rs.725")
```

```
elif (x=="Python Programming"):
```

```
self.bookid_var.set("BKID1245")
```

```
self.booktitle_var.set("Intro to python Comp Science")
```

```
self.auther_var.set("John Zhelle")
```

```
d1=datetime.date.today()
```

```
d2=datetime.timedelta(days=15)
```

```
d3=d1+d2
```

```
self.dateborrowed_var.set(d1)
```

```
self.datedue_var.set(d3)
```

```
self.daysonbook.set("15")
```

```
self.lateratefine_var.set("Rs.25")
```

```
self.dateoverdue.set("NO")
```

```
self.finallprice.set("Rs.500")
```

```
elif (x=="Secrete Rahshy"):
```

```
self.bookid_var.set("BKID8796")
```

```
self.booktitle_var.set("Basic Of Pythpn")
```

```
self.auther_var.set("Ref.Kapil Kamble")
```

```
d1=datetime.date.today()
```

```
d2=datetime.timedelta(days=15)
```

```
d3=d1+d2
```

```
self.dateborrowed_var.set(d1)
```

```
self.datedue_var.set(d3)
```

```
self.daysonbook.set("15")
self.lateratefine_var.set("Rs.25")
self.dateoverdue.set("NO")
self.finallprice.set("Rs.289")
```

```
elif (x=="Python CookBook"):
```

```
self.bookid_var.set("BKID2546")
self.booktitle_var.set("Python Cookbook")
self.auther_var.set("Brian Jones")
```

```
d1=datetime.date.today()
d2=datetime.timedelta(days=15)
d3=d1+d2
self.dateborrowed_var.set(d1)
self.datedue_var.set(d3)
self.daysonbook.set("15")
self.lateratefine_var.set("Rs.25")
self.dateoverdue.set("NO")
self.finallprice.set("Rs.354")
```

```
elif (x=="Into to Machine Learning"):
```

```
self.bookid_var.set("BKID8796")
self.booktitle_var.set("Intro to Machine Learning")
self.auther_var.set("Sarah Guido")
```

```
d1=datetime.date.today()
d2=datetime.timedelta(days=15)
```

```
d3=d1+d2
self.dateborrowed_var.set(d1)
self.datedue_var.set(d3)
self.daysonbook.set("15")
self.lateratefine_var.set("Rs.25")
self.dateoverdue.set("NO")
self.finallprice.set("Rs.725")
```

```
bookList=Listbox(DataFrameRight,font=("arial",12,"bold"),width=20,height=16)
bookList.bind('<<ListboxSelect>>',SelectBook)
bookList.grid(row=0,column=0,padx=4)
listScrollbar.config(command=bookList.yview)
```

```
for item in ListOfBooks:
    bookList.insert(END,item)
```

```
#=====Scrollbar=====
```

```
Table_frame=Frame(FrameDetails,bd=6,relief=RIDGE)
Table_frame.place(x=0,y=1,width=1460,height=150)

scroll_x=ttk.Scrollbar(Table_frame,orient=HORIZONTAL)
scroll_y=ttk.Scrollbar(Table_frame,orient=VERTICAL)
```

```
self.library_table=ttk.Treeview(Table_frame,column=("member","ref","title","firtname","lastn
ame","adress1",
```

```
"adress2","postid","mobile","bookid","booktitle","auther","dateborrowed",
```

```
        "datedue","days","laterreturnfine","dateoverdue","finalprice")
        ,xscrollcommand=scroll_x.set,yscrollcommand=scroll_y.set)
scroll_x.pack(side=BOTTOM,fill=X)
scroll_y.pack(side=RIGHT,fill=Y)

scroll_x.config(command=self.library_table.xview)
scroll_y.config(command=self.library_table.yview)

self.library_table.heading("member",text="Member Type")
self.library_table.heading("ref",text="Reference No.")
self.library_table.heading("title",text="Title")
self.library_table.heading("firname",text="First Name")
self.library_table.heading("lastname",text="Last Name")
self.library_table.heading("adress1",text="Address1")
self.library_table.heading("adress2",text="Address2")
self.library_table.heading("postid",text="Post ID")
self.library_table.heading("mobile",text="Mobile Number")
self.library_table.heading("bookid",text="Book ID")
self.library_table.heading("booktitle",text="Book Title")
self.library_table.heading("auther",text="Auther")
self.library_table.heading("dateborrowed",text="Date Of Borrowed")
self.library_table.heading("datedue",text="Date Due")
self.library_table.heading("days",text="DaysOnBook")
self.library_table.heading("laterreturnfine",text="LateReturnFine")
self.library_table.heading("dateoverdue",text="DateOverDue")
self.library_table.heading("finalprice",text="Final Price")

self.library_table["show"]="headings"
```

```

self.library_table.column("member",width=100)
self.library_table.column("ref",width=100)
self.library_table.column("title",width=100)
self.library_table.column("firtname",width=100)
self.library_table.column("lastname",width=100)
self.library_table.column("adress1",width=100)
self.library_table.column("adress2",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("auther",width=100)
self.library_table.column("dateborrowed",width=100)
self.library_table.column("datedue",width=100)
self.library_table.column("days",width=100)
self.library_table.column("latereturnfine",width=100)
self.library_table.column("dateoverdue",width=100)
self.library_table.column("finalprice",width=100)
self.library_table.pack(fill=BOTH,expand=1)

self.fatch_data()
self.library_table.bind("<ButtonRelease-1>",self.get_cursor)
self.fatch_data()
# =====Function Declaration=====
def add_data(self):
if self.member_var.get()==" " or self.postcode_var.get()==" ":
    messagebox.showerror("Error","All Fields Are Required",parent=self.root)

else:

```

try:

```
conn=mysql.connector.connect(host="localhost",username="root",password="Ketan@16m",database="mydata")
```

```
    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,%s,%s),(
```

```
    self.member_var.get(),
```

```
    self.ref_var.get(),
```

```
    self.title_var.get(),
```

```
    self.firstname_var.get(),
```

```
    self.lastname_var.get(),
```

```
    self.address1_var.get(),
```

```
    self.address2_var.get(),
```

```
    self.postcode_var.get(),
```

```
    self.mobile_var.get(),
```

```
    self.bookid_var.get(),
```

```
    self.booktitle_var.get(),
```

```
    self.auther_var.get(),
```

```
    self.dateborrowed_var.get(),
```

```
    self.datedue_var.get(),
```

```
    self.daysonbook.get(),
```

```
    self.lateratefine_var.get(),
```

```
    self.dateoverdue.get(),
```

```
    self.finallprice.get()
```

```
))
```

```
conn.commit()
```

```

self.fetch_data()

conn.close()

messagebox.showinfo("Success","Member has been inserted",parent=self.root)

except Exception as es:
    messagebox.showerror("Error","f" Must be enter Integer number,PRN NO&ID NO is
Primery Key :{str(es)}",parent=self.root)

def update_data(self):
    if self.ref_var.get()=="":
        messagebox.showerror("Error","All Fields Are Required",parent=self.root)
    else:

conn=mysql.connector.connect(host='localhost',username='root',password='Ketan@16m',data
base='mydata')
    my_cursor=conn.cursor()
    my_cursor.execute("update library set
Member_type=%s,ID_No=%s,FirstName=%s,LastName=%s,Address1=%s,Address2=%s,Pos
tCode=%s,Mobile=%s,Bookid=%s,Booktitle=%s,Auther=%s,DateBorrowed=%s,DateDue=%
s,DaysOfBook=%s,LateReturnFine=%s,DateOverDue=%s,FinalPrice=%s where
PRN_No=%s",

self.member_var.get(),

self.title_var.get(),

self.firstname_var.get(),

self.lastname_var.get(),

```

```
self.address1_var.get(),  
self.address2_var.get(),  
self.postcode_var.get(),  
self.mobile_var.get(),  
self.bookid_var.get(),  
self.booktitle_var.get(),  
self.auther_var.get(),  
self.dateborrowed_var.get(),  
self.datedue_var.get(),  
self.daysonbook.get(),  
self.lateratefine_var.get(),  
self.dateoverdue.get(),  
self.finallprice.get(),  
self.ref_var.get()
```

```
)
```

```
conn.commit()  
self.fatch_data()  
self.reset()
```



```

    conn.close()

    messagebox.showinfo("UPDATE","Record has been updated
successfully",parent=self.root)

def fetch_data(self):

conn=mysql.connector.connect(host="localhost",username="root",password="Ketan@16m",d
atabase="mydata")

    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_row=self.library_table.focus()
    content=self.library_table.item(cursor_row)
    row=content["values"]

    self.member_var.set(row[0]),
    self.ref_var.set(row[1]),
    self.title_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.postcode_var.set(row[7]),
self.mobile_var.set(row[8]),
self.bookid_var.set(row[9]),
self.booktitle_var.set(row[10]),
self.auther_var.set(row[11]),
self.dateborrowed_var.set(row[12]),
self.datedue_var.set(row[13]),
self.daysonbook.set(row[14]),
self.lateratefine_var.set(row[15]),
self.dateoverdue.set(row[16]),
self.finallprice.set(row[17])
```

```
    def mDelete(self):
if self.ref_var.get()=="":
    messagebox.showinfo("ERROR","First Select the Member!!",parent=self.root)
else:
```

```
conn=mysql.connector.connect(host='localhost',username='root',password='Ketan@16m',data
base='mydata')
```

```
    my_cursor=conn.cursor()
    query="delete from library where PRN_No=%s"
    value=(self.ref_var.get(),)
    my_cursor.execute(query,value)
```

```
    conn.commit()
    conn.close()
    self.fatch_data()
    self.reset()
```

```
messagebox.showinfo("DELETE","Member has been Deleted  
successfully",parent=self.root)
```

```
def iExit(self):  
    iExit=messagebox.askyesno("Library Management System","Confirm if you want to  
exit",parent=self.root)  
    if iExit>0:  
        self.root.destroy()  
        return
```

```
def reset(self):  
    self.member_var.set(""),  
    self.ref_var.set(""),  
    self.title_var.set(""),  
    self.firstname_var.set(""),  
    self.lastname_var.set(""),  
    self.address1_var.set(""),  
    self.address2_var.set(""),  
    self.postcode_var.set(""),  
    self.mobile_var.set(""),  
    self.bookid_var.set(""),  
    self.booktitle_var.set(""),  
    self.auther_var.set(""),  
    self.dateborrowed_var.set(""),  
    self.datedue_var.set(""),  
    self.daysonbook.set(""),  
    self.lateratefine_var.set(""),  
    self.dateoverdue.set(""),  
    self.finallprice.set("")
```

```
self.txtBox.delete("1.0",END)
```

```
def showData(self):
```

```
self.txtBox.insert(END,"Member Type:\t\t"+ self.member_var.get() + "\n")
```

```
self.txtBox.insert(END,"PRN No:\t\t"+ self.ref_var.get() + "\n")
```

```
self.txtBox.insert(END,"ID No:\t\t"+ self.title_var.get() + "\n")
```

```
self.txtBox.insert(END,"FirstName:\t\t"+ self.firstname_var.get() + "\n")
```

```
self.txtBox.insert(END,"LastName:\t\t"+ self.lastname_var.get() + "\n")
```

```
self.txtBox.insert(END,"Address1:\t\t"+ self.address1_var.get() + "\n")
```

```
self.txtBox.insert(END,"Address2:\t\t"+ self.address2_var.get() + "\n")
```

```
self.txtBox.insert(END,"Post Code:\t\t"+ self.postcode_var.get() + "\n")
```

```
self.txtBox.insert(END,"Mobile No:\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,"Auther:\t\t"+ self.auther_var.get() + "\n")
```

```
self.txtBox.insert(END,"DateBorrowed:\t\t"+ self.dateborrowed_var.get() + "\n")
```

```
self.txtBox.insert(END,"DateDue:\t\t"+ self.datedue_var.get() + "\n")
```

```
self.txtBox.insert(END,"DaysOnBook:\t\t"+ self.daysonbook.get() + "\n")
```

```
self.txtBox.insert(END,"LateRateFine:\t\t"+ self.lateratefine_var.get() + "\n")
```

```
self.txtBox.insert(END,"DateOverDue:\t\t"+ self.dateoverdue.get() + "\n")
```

```
self.txtBox.insert(END,"FinallPrice:\t\t"+ self.finallprice.get() + "\n")
```

```
if __name__ == "__main__":
```

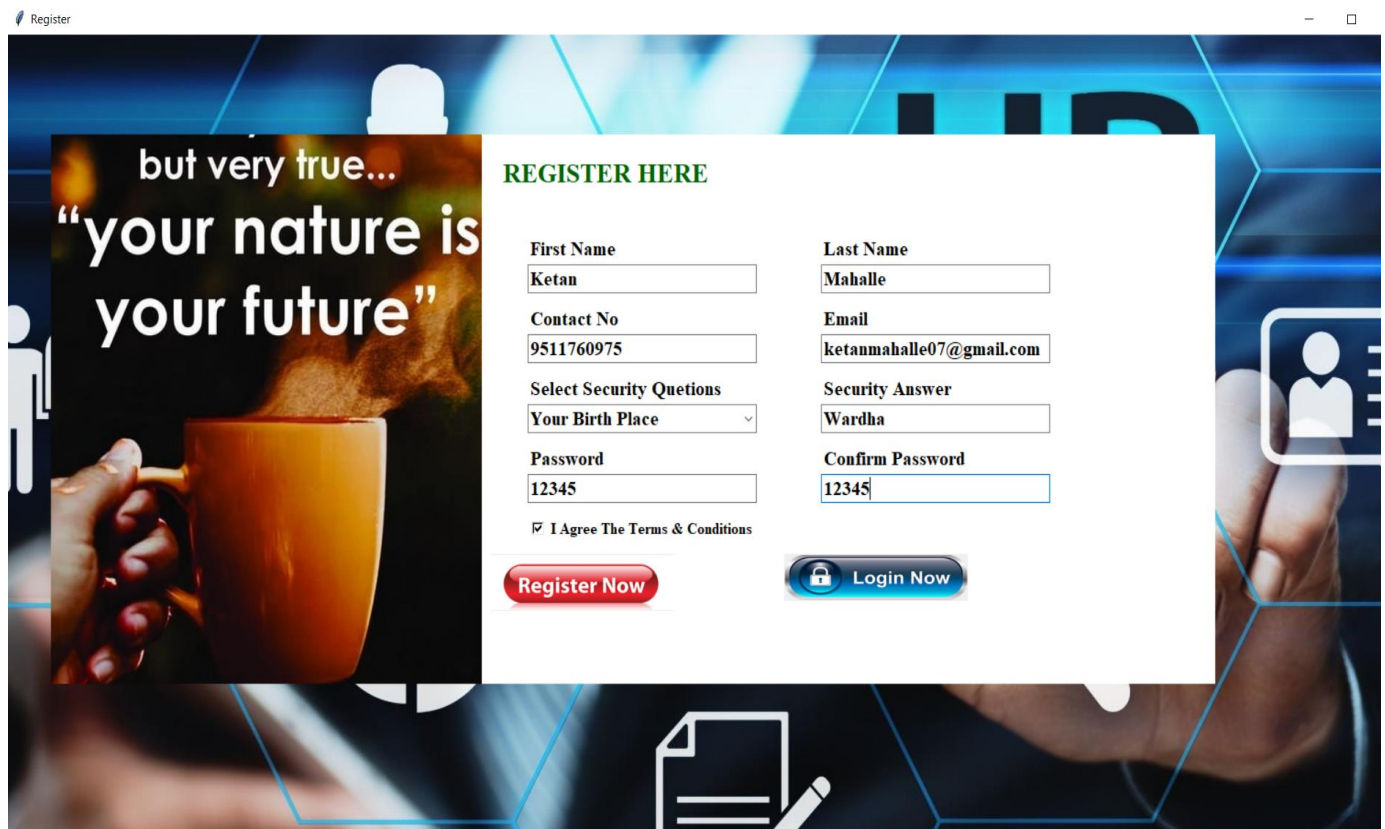
```
main()
```

Input Screen and Output Screen

1) Registration:

Input:

The User or Admin has to first register to use the Library Management System. This is the Registration input screen, in which the user has to enter some details to generate his Username and Password. After entering the details, the user has to tick (✓) 'I Agree the Terms and Conditions' option and then click on the 'Register Now' button.

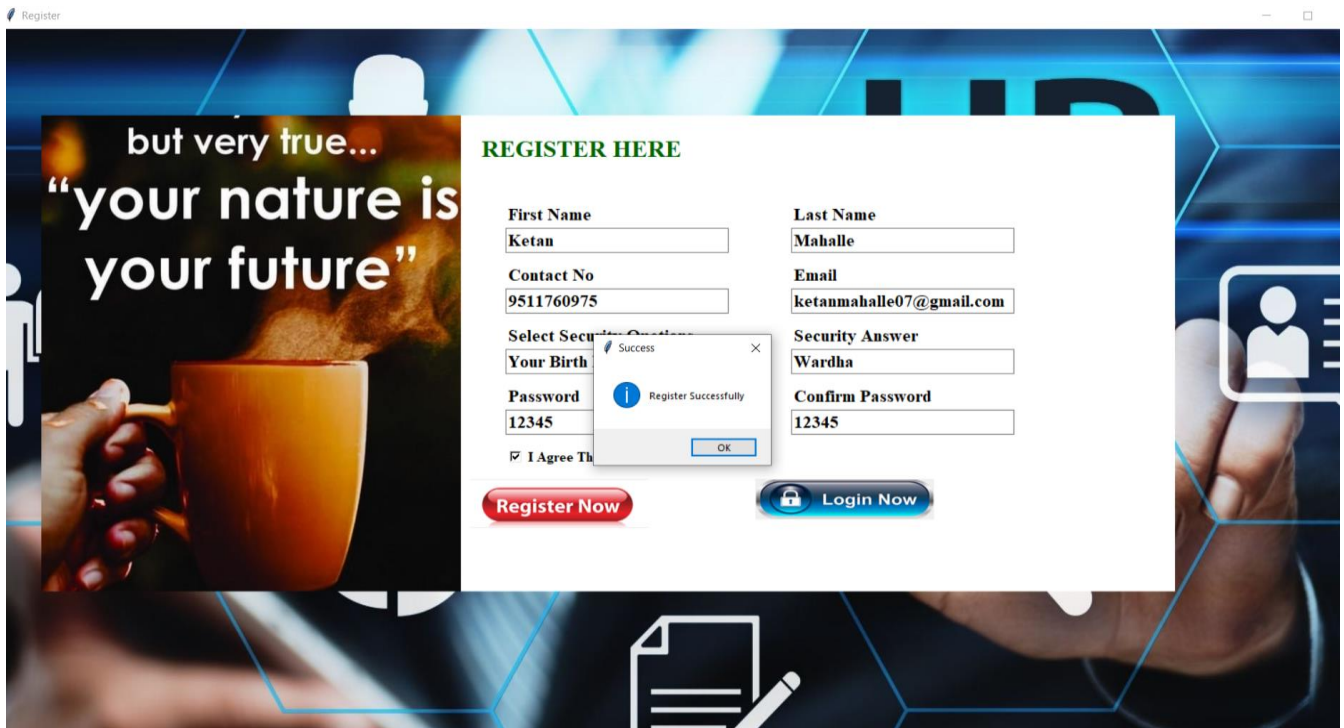


The screenshot shows a web browser window with a registration form. The form is titled "REGISTER HERE" and is overlaid on a background image of a hand holding a mug of coffee. The form contains the following fields and elements:

- First Name:** Input field with the value "Ketan".
- Last Name:** Input field with the value "Mahalle".
- Contact No:** Input field with the value "9511760975".
- Email:** Input field with the value "ketanmahalle07@gmail.com".
- Select Security Questions:** A dropdown menu with the selected option "Your Birth Place".
- Security Answer:** Input field with the value "Wardha".
- Password:** Input field with the value "12345".
- Confirm Password:** Input field with the value "12345".
- I Agree The Terms & Conditions:** A checked checkbox.
- Buttons:** A red "Register Now" button and a blue "Login Now" button.

Output:

When the User or Admin clicks on the Register Now button, he will see the message 'Registration Successful'.

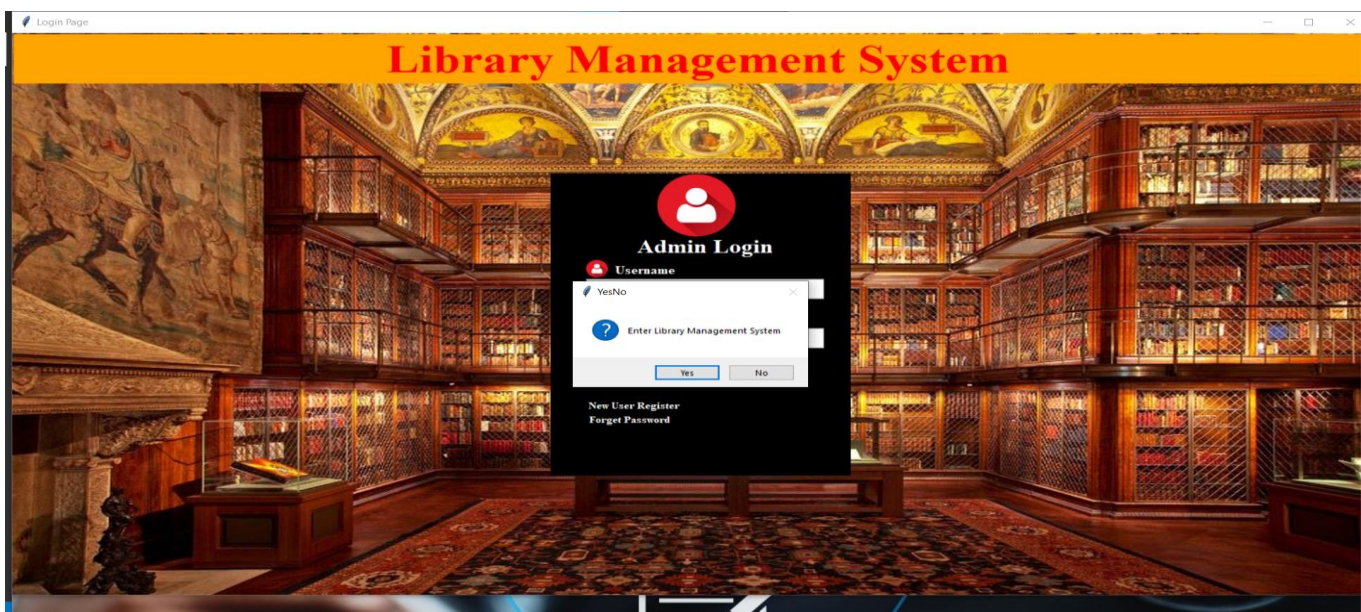


2) Login Page:

Input: This is the Library Management System login form. Here the Admin has to enter his Username and Password. When the Admin registers, the email he enters is his Username.

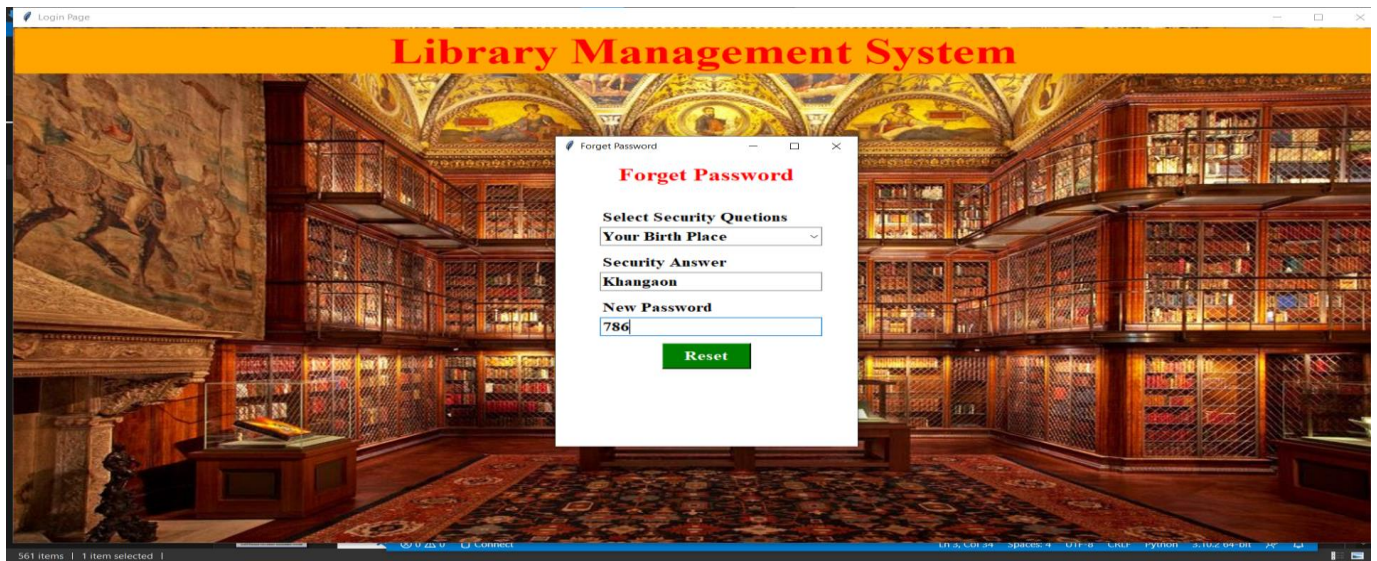


Output: After the Admin has entered his Username and Password, the Admin has to click on the 'Login' button. The Library Management System screen will open only if the Admin has entered the correct Username and Password, otherwise not. 'Enter in Library management system' Application asks user to enter or not.

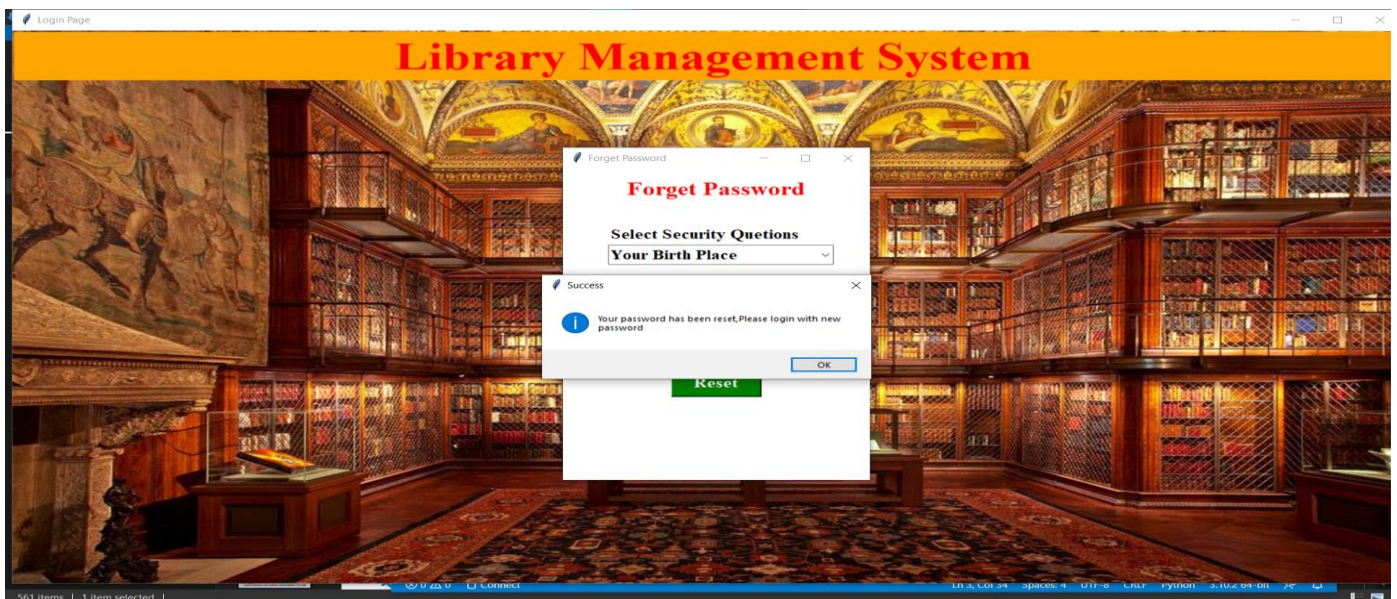


3) Forget Password:

Input: If a Admin or any user forgets his Password, they can re-create it by clicking on the 'Forget Password' button. On the Forget Password Page, Admin has to type the Security Question, Security Answer and New Password that Admin wants to create and then click on the 'Reset' button.



Output: When the Admin clicks on the reset button, his password will be reset and he will see the message 'Your Password has been reset. Please login with new Password'.



4) Add Data:

Input:

Admin can add data of members and Books in Library Management System. To add new or old member's data, the Admin has to enter the Member's First Name, Last Name, Contact, PRN No and other details. The Admin has to click on the book that the member wants to issue and next click on the 'ADD DATA' button.

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjk	2155488532	BKID1245	Intro to pythpn C	John Zhelle	2021-01-20	2021-02-04	15

Output: When the admin clicks on the 'ADD DATA' button, Admin will see the message 'Member has been Inserted' on the screen.

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	Iakhni	nagpur	442500	9875632147	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15

5) Show Data:

Input:

Whenever the Admin wants to see the details of any member, then the Admin has to select that member from the scroll bar and click on the 'SHOW DATA' button.

The screenshot shows the 'LIBRARY MANAGEMENT SYSTEM' interface. The 'Library Membership Information' section contains the following data:

Member Type	Student	Book Id:	BKID5487
PRN No:	786	Book Title:	Python manual
ID No:	258	Author Name:	Paull berry
FirstName:	Adi	Date Borrowed:	2022-04-26
Surname:	Bisen	Date Due:	2022-05-11
Address1:	Jakhni	Days On Book:	15
Address2:	Dharampath	Late Return Fine:	Rs.25
Post Code:	442500	Date Over Due:	NO
Mobile Number:	9875632157	Actual Price:	Rs.375

The 'Book Details' section lists the following books:

- Head Firt Book
- Learn Python The Hard
- Python Programming
- Secrete Rahshy
- Python CookBook
- Into to Machine Learnin
- Fluent Python
- progrrming Python
- The Algorithm
- The tecnich Python
- Machine tecno
- My Python
- Joss Ellif guru
- Elite Jungle python
- Jungli Python
- Mumbai Python

The navigation bar at the bottom contains buttons: ADD DATA, SHOW DATA, UPDATE, DELETE, RESET, EXIT. The 'SHOW DATA' button is highlighted in red.

The table below the navigation bar shows a list of members with columns: Member Type, Reference No., Title, First Name, Last Name, Address1, Address2, Post ID, Mobile Number, Book ID, Book Title, Author, Date Of Borrowec, Date Due, Da.

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Author	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	Jakhni	Dharampath	442500	9875632157	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed	431715	78946561230	BKID5487	Python manual	Paull berry	2020-12-17	2021-01-01	15
Student	5487981	54821	Kapil	Kamble	Sultanpur	Delhi	458721	9876543210	BKID8796	Basic Of Pythpn	ZED A.SHAW	2020-12-17	2021-01-01	15

Output: When the Admin will click on the 'SHOW DATA' button, Admin will see the details of that member on the right side of the book details frame.

The screenshot shows the 'LIBRARY MANAGEMENT SYSTEM' interface. The 'Library Membership Information' section contains the following data:

Member Type	Student	Book Id:	BKID5487
PRN No:	786	Book Title:	Python manual
ID No:	258	Author Name:	Paull berry
FirstName:	Adi	Date Borrowed:	2022-04-26
Surname:	Bisen	Date Due:	2022-05-11
Address1:	Jakhni	Days On Book:	15
Address2:	nagpur	Late Return Fine:	Rs.25
Post Code:	442500	Date Over Due:	NO
Mobile Number:	9875632147	Actual Price:	Rs.375

The 'Book Details' section lists the following books:

- Head Firt Book
- Learn Python The Hard
- Python Programming
- Secrete Rahshy
- Python CookBook
- Into to Machine Learnin
- Fluent Python
- progrrming Python
- The Algorithm
- The tecnich Python
- Machine tecno
- My Python
- Joss Ellif guru
- Elite Jungle python
- Jungli Python
- Mumbai Python

The 'Book Details' section also displays the member's information:

Member Type:	Student
PRN No:	786
ID No:	258
FirstName:	Adi
LastName:	Bisen
Address1:	Jakhni
Address2:	nagpur
Post Code:	442500
Mobile No:	9875632147
Book ID:	BKID5487
Book Title:	Python manual
Author:	Paull berry
DateBorrowed:	2022-04-26
DateDue:	2022-05-11
DaysOnBook:	15
LateRateFine:	Rs.25

The navigation bar at the bottom contains buttons: ADD DATA, SHOW DATA, UPDATE, DELETE, RESET, EXIT. The 'SHOW DATA' button is highlighted in red.

The table below the navigation bar shows a list of members with columns: Member Type, Reference No., Title, First Name, Last Name, Address1, Address2, Post ID, Mobile Number, Book ID, Book Title, Author, Date Of Borrowec, Date Due, Da.

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Author	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	Jakhni	nagpur	442500	9875632147	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythpn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythpn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15

6) Update Data:

Input:

When the Admin needs to update the details of any member, Admin has to select member from the scroll bar and after selecting member, Admin can change any detail of member. After changing, click on the 'UPDATE' button.

LIBRARY MANAGEMENT SYSTEM

Library Membership Information

Member Type: Student
 PRN No: 786
 ID No: 258
 FirstName: Adi
 Surname: Bisen
 Address1: Iakhni
 Address2: Nagpur
 Post Code: 442500
 Mobile Number: 9875632147

Book Id: BKID5487
 Book Title: Python manual
 Auther Name: Paull berry
 Date Borrowed: 2022-04-26
 Date Due: 2022-05-11
 Days On Book: 15
 Late Return Fine: Rs.25
 Date Over Due: NO
 Actual Price: Rs.375

Book Details

Head Firt Book
 Learn Python The Hard
 Python Programming
 Secrete Rahshy
 Python CookBook
 Into to Machine Learnin
 Fluent Python
 progmming Python
 The Algorithm
 The tecnich Python
 Machine tecno
 My Python
 Joss Ellif guru
 Elite Jungle python
 Jungli Python
 Mumbai Python

Member Type: Student
 PRN No: 786
 ID No: 258
 FirstName: Adi
 LastName: Bisen
 Address1: Iakhni
 Address2: Dharampeth
 Post Code: 442500
 Book ID: BKID5487
 Book Title: Python manual
 Auther: Paull berry
 DateBorrowed: 2022-04-26
 DateDue: 2022-05-11
 DaysOnBook: 15
 LateRateFine: Rs.25

ADD DATA SHOW DATA UPDATE DELETE RESET EXIT

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	Iakhni	Dharampeth	442500	9875632157	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Lecturer	548855	2155	kk	jij	44554	jjj	kjkj	2155488532	BKID1245	Intro to pythn C	John Zhelle	2021-01-20	2021-02-04	15
Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed	431715	78946561230	BKID5487	Python manual	Paull berry	2020-12-17	2021-01-01	15

Output: When the Admin clicks on the 'UPDATE' button, Admin will see a 'Record has been Updated Successfully' message on the screen.

LIBRARY MANAGEMENT SYSTEM

Library Membership Information

Member Type:
 PRN No:
 ID No:
 FirstName:
 Surname:
 Address1:
 Address2:
 Post Code:
 Mobile Number:

Book Id:
 Book Title:
 Auther Name:
 Date Borrowed:
 Date Due:
 Days On Book:
 Late Return Fine:
 Date Over Due:
 Actual Price:

Book Details

Head Firt Book
 Learn Python The Hard
 Python Programming
 Secrete Rahshy
 Python CookBook
 Into to Machine Learnin
 Fluent Python
 progmming Python
 The Algorithm
 The tecnich Python
 Machine tecno
 My Python
 Joss Ellif guru
 Elite Jungle python
 Jungli Python
 Mumbai Python

ADD DATA SHOW DATA UPDATE DELETE RESET EXIT

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guida	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	Iakhni	Dharampeth	442500	9875632157	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Admin Staf	789	012	Pratik	Dhale	Wardha	Nagpur	442001	7666547208	BKID8796	Basic Of Pythn	ZED A.SHAW	2022-04-22	2022-05-07	15
Admin Staf	457896	45789	Dhruva	Rathi	Delhi	Mumbai	214578	3214567890	BKID8796	Basic Of Pythn	Ref.Kapil Kamble	2020-12-17	2021-01-01	15

Record has been updated successfully

7) Delete Data:

Input:

When an Administrator wants to delete a member, the Administrator has to select that member from the scroll bar and click on the 'DELETE' button.

The screenshot shows the 'LIBRARY MANAGEMENT SYSTEM' interface at 12:38:37 AM. It features two main sections: 'Library Membership Information' and 'Book Details'. Below these is a navigation bar with buttons: 'ADD DATA', 'SHOW DATA', 'UPDATE', 'DELETE', 'RESET', and 'EXIT'. At the bottom is a table with columns: Member Type, Reference No., Title, First Name, Last Name, Address1, Address2, Post ID, Mobile Number, Book ID, Book Title, Author, Date Of Borrowed, Date Due, and Days On Book. The row for Member Type 'Admin Staf', Reference No. '789', and Title 'Basic Of Pythpn' is selected.

Output: When the Admin clicks on the 'DELETE' button, Admin will see a 'Member has been Deleted Successfully' message on the screen.

The screenshot shows the same 'LIBRARY MANAGEMENT SYSTEM' interface at 12:52:48 AM. A confirmation dialog box is overlaid on the 'DELETE' button, displaying the message: 'Member has been Deleted successfully'. The dialog box has an 'OK' button. The table below still shows the same data as in the previous screenshot.

8) Reset Data:

Input: Admin will click on ‘RESET’ button when Admin has to clear the Library Management System screen.

The screenshot shows the 'LIBRARY MANAGEMENT SYSTEM' interface at 12:40:09 AM. The 'RESET' button in the navigation bar is highlighted in red. The 'Library Membership Information' and 'Book Details' sections are populated with data. Below the navigation bar is a table with the following data:

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Author	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guaido	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	lakhni	Dharampeth	442500	9875632157	BKID5487	Python manual	Pauli berry	2022-04-26	2022-05-11	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjk	2155488532	BKID1245	Intro to python C	John Zhelle	2021-01-20	2021-02-04	15
Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed	431715	78946561230	BKID5487	Python manual	Pauli berry	2020-12-17	2021-01-01	15

Output:

When the admin will click on the ‘RESET’ button, the Library Management System screen will be completely cleared and it will look like it has just been opened.

The screenshot shows the 'LIBRARY MANAGEMENT SYSTEM' interface at 12:55:56 AM. The 'RESET' button in the navigation bar is highlighted in red. The 'Library Membership Information' and 'Book Details' sections are now empty. Below the navigation bar is a table with the following data:

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Author	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guaido	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	lakhni	Dharampeth	442500	9875632157	BKID5487	Python manual	Pauli berry	2022-04-26	2022-05-11	15
Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed	431715	78946561230	BKID5487	Python manual	Pauli berry	2020-12-17	2021-01-01	15
Student	5487981	54821	Kapil	Kamble	Sultanpur	Delhi	458721	9876543210	BKID8796	Basic Of Pythpn	ZED A.SHAW	2020-12-17	2021-01-01	15

9)Exit:

When the Administrator's task is over and Admin wants to close the Library Management System, Admin has to click on the 'EXIT' button. When Admin clicks on the 'EXIT' button Admin will see the message box. It will ask 'Confirm if you want to Exit'. If Admin clicked 'Yes' button the screen will closed and if Admin clicked 'No' button the screen will stay on.

The screenshot displays the Library Management System interface. At the top, the title bar reads "Library Management System" with a timestamp of "12:40:56 AM". The main header is "LIBRARY MANAGEMENT SYSTEM" in large red letters. Below the header, there are two main sections: "Library Membership Information" and "Book Details".

The "Library Membership Information" section contains several input fields for member details, including Member Type, PRN No, ID No, FirstName, Surname, Address1, Address2, Post Code, and Mobile Number. It also includes fields for Book Id, Book Title, Auther Name, Date Borrowed, Date Due, Days On Book, Late Return Fine, Date Over Due, and Actual Price.

The "Book Details" section shows a list of books, with "Intro to Machine Learning" highlighted. The list includes titles like "Head Firt Book", "Learn Python The Hard", "Python Programming", "Secrete Rahshy", "Python CookBook", "Fluent Python", "progrrmming Python", "The Algorithm", "The tecnich Python", "Machine tecno", "My Python", "Joss Ellif guru", "Elite Jungle python", "Jungli Python", and "Mumbai Python".

A confirmation dialog box is open in the center, titled "Library Management System", with the message "Confirm if you want to exit" and "Yes" and "No" buttons.

Below the input fields, there is a navigation bar with buttons for "ADD DATA", "SHOW DATA", "UPDATE", "DELETE", "RESET", and "EXIT".

At the bottom, there is a table displaying the library data:

Member Type	Reference No.	Title	First Name	Last Name	Address1	Address2	Post ID	Mobile Number	Book ID	Book Title	Auther	Date Of Borrowec	Date Due	Da
Admin Staf	123	456	Ketan	Mahalle	Wardha	Nagpur	442001	9511760975	BKID8796	Intro to Machine	Sarah Guaido	2022-04-22	2022-05-07	15
Student	786	258	Adi	Bisen	lakhni	Dharampeth	442500	9875632157	BKID5487	Python manual	Paull berry	2022-04-26	2022-05-11	15
Lecturer	548855	2155	kk	jjj	44554	jjj	kjkj	2155488532	BKID1245	Intro to python C	John Zhelle	2021-01-20	2021-02-04	15
Lecturer	2145871	624954	Anil	Kamble	Makni	Vasur,Mukhed	431715	78946561230	BKID5487	Python manual	Paull berry	2020-12-17	2021-01-01	15

The bottom status bar shows "1 item | 1 item selected 32.3 KB |" and a taskbar with various application icons.

TESTING AND
VALIDATION CHECKS

Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. Our Project went through two levels of testing:

1. Unit testing

2. integration testing

UNIT TESTING:

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module, we need to provide a complete environment i.e., besides the module we would require:

- The procedures belonging to other modules that the module under test calls.
- Non local data structures that module accesses.
- A procedure to call the functions of the module under test with appropriate parameters.

1. Test For the admin module:

- **Testing admin login form-**This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.
- **Student account addition-** In this section the admin can verify student details from student academic info and then only add student details to main library database it contains add and delete buttons if user click add button data will be added to student database and if he clicks delete button the student data will be deleted.
- **Book Addition-** Admin can enter details of book and can add the details to the main book table also he can view the books requests.

2. Test for Student login module:

- **Test for Student login Form-** This form is used for log in of Student. In this we enter the library id, username and password if all these are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for library id, username and password.
- **Test for account creation-** This form is used for new account creation when student does not fill the form completely it asks again to fill the whole form when he fill the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.

3. Test for teacher login module:

- **Test for teacher login form-** This form is used for log in of teacher .In this we enter the username and password if all these are correct teacher login page will

open otherwise, if any of data is wrong it will get redirected back to the login page and again ask for username and password.

INTEGRATION TESTING:

In this type of testing, we test various integration of the project module by providing the input. The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

Validation Checks

The process of evaluating web-based application during the development process or at the end of the development process to determine whether it satisfied information requirement. Validation testing ensures that the product actually meets the user needs. It can also have defined as to demonstrate that the information fulfills its intended use when deployed on appropriate environment.

Validation testing can be best demonstrated. The web-based application under test is evaluated during this type of testing.

- **VALIDATION INPUT TRANSACTION: -**

Validation input data is largely done through website which is the programmer's responsibility but it is important that system analyst must know what a common problem might in validation a transaction. Business committed to quality will include validation checks a part of their routine website.

1. Submitting the wrong data to system.
2. Submitting the data by an unauthorized person.
3. Asking the system to perform an unacceptable function.

- **VALIDATION INPUT DATA: -**

It is essential that the input data themselves along with the transaction requested are valid. Several texts can be incorporated into website to ensure the validity. We consider many possible ways to validate input and they are as follows:

1. Test for missing data.
2. Test for correct field length.
3. Test for range or reasonable.
4. Test for comparison with stored data.

SYSTEM SECURITY

MEASURES

System security measures

Security of a computer system is a crucial task. It is a process of ensuring confidentiality. A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of the various threats and unauthorized access.

Security measures will be taken:

- Strong passwords:

This first measure is taken that users may use special characters in their passwords and password get strong.

- Confidentiality:

If any users are sharing their personal details in login form it will be secure safely as only users can access such information.

- Hidden Password:

Whenever user input the password in login form, then password show in * (Asterisk) format.

IMPLEMENTATION,
EVALUATION AND
MAINTENANCE

Implementation

The design of a management information system may seem to management to be an expensive project, the cost of getting the MIS on line satisfactorily may often be comparable to that of its design, and the implementation has been accomplished when the outputs of the MIS are continuously utilized by decision makers. Once the design has been completed, there are four basic methods for implementing the MIS.

These are following:

1. Install the system in a new operation or organization.

2. Cut off the old system and install the new:

This produces a time gap during which no system is in operation. Practically, installation requires one or two days for small companies or small systems.

3. Cut over by segments:

This method is also referred as” phasing in” the new system. Small parts or subsystems are substituted for the old. In the case of upgrading old systems, this may be a very desirable method.

4. Operate in parallel and cut over:

The new system is installed and operated in parallel with the current system until it has been checked out, then only the current system is cut out. This method is expensive

because of personal and related costs. Its big advantages are that the system is fairly well debugged when it becomes the essential information system.

Evolution

After the MIS has been operating smoothly for a short period of time, an evaluation of each step in the design and of the final system performance should be made. Evaluation should not be delayed beyond the time when the system's analysts have completed most of the debugging. The longer the delay, the more difficult it will be for designer to remember important details. The evaluation should be made by the customer as well as by the designers.

Maintenance

Control and maintenance of the system are the responsibilities of the line managers. Control of the systems means the operation of the system as it was designed to operate. Sometimes, well-intentioned people or operators may make unauthorized changes to improve the system, changes that are not approved or documented. Maintenance is closely related to control. Maintenance is that ongoing activity that keeps the MIS at the highest levels of effectiveness and efficiency within cost constraints. Maintenance is directed towards reducing errors due to design, reducing errors due to environmental changes and improving the system's scope and services.

FUTURE SCOPE
OF PROJECT

FUTURE SCOPE OF PROJECT

Our web-based application “Library Management System” which provides complete information about Users like Student, Admin and Lecturer. We will add more content on them in future. In our web-based application right now, only Books and Users with their information available but in future we will add Online Lectures, Links, etc.

We will also provide more images in GUI related to our web-based application in future. We will try to find out more about this topic and add in future. We will try to make this application more attractive so that visitor cannot get bored while using it. We will provide login id to each and every user so that he can access our website from anywhere through log in id and password. In future we add some major facilities like Reservation of Book.

We will also provide more images in GUI related to our web-based application in future. We will try to find out more about this topic and add in future. We will try to make this application more attractive so that visitor cannot get bored while using it. We will provide login id to each and every user so that he can access our website from anywhere through log in id and password. In future we add some major facilities like Reservation of Book.

CONCLUSION

CONCLUSION :

Library Management System allows the user to store the book details and the customer details. This software package allows storing the details of all the data related to library. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

BIBLIOGRAPHY
&
REFERENCES

BIBLIOGRAPHY & REFERENCES :

- **Websites References:**

<https://www.tutorialspoint.com/index.htm>

<https://www.javatpoint.com>

<https://www.w3schools.com>

<https://html.com>

- **Books References:**

Head First Python: A Brain-Friendly Guide

Programming Python: Powerful Object-Oriented Programming

The Pragmatic Programmer

**A
PROJECT SYNOPSIS
ON**

“Library Management System”

Submitted to

**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS
In the Partial Fulfillment of**

B.Com. (Computer Application) Final Year

Synopsis Submitted by :

Ketan Anilrao Mahalle,
Pratik Tarachand Dhale.

Under the Guidance of

Pravin J. Yadao



**G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
AUTONOMOUS
2021-2022**

1.Introduction: (Write 4 to 5 lines)

Library Management System is Library Management software for monitoring and controlling the transactions in a library .

This project of “LIBRARY MANAGEMENT” of gives us the complete information about the library.

We can enter the record of new books and retrieve the details of books available in the library.

We can issue the books to the students and maintain their records and can also check how many books are issued.

2. Objectives of the project: (Write only 5 points)

This project makes a convenient way for the library to keep the records of books, borrowing and returning of books.

It will help the management to do the work at ease.

It can assist the staff in capturing the effort spent on their respective working areas.

The Library Management System is an application for assisting a librarian in managing a book library in a university.

3. Project Category: Web based application .

4. Tools/ Platform/ Languages to be used: Python and mysql .

5. Scope of future application: (Write 4 to 5 points):

To assist the staff in capturing the effort spent on their respective working areas.

To utilize resources in an efficient manner by increasing their productivity through automation.

The system generates types of information that can be used for various purposes.

The system would provide basic set of features to add/update members, add/update books, and manage check in specifications for the systems based on the client’s statement of need.

Submitted by,

Ketan Anilrao Mahalle.

Pratik Tarachand Dhale.

Approved by,

**Prof. Pravin Yadao
Project Guide**