BMS

(BOOKS MANAGEMENT SYSTEM)

SUBMITTED BY :

- **Yuvraj Rajbhar - 12115849 – RK21QTB42**
- **Saisree Pulavarthi – 12114133 -RK21QTB67**

TABLE OF CONTENTS

INTRODUCTION

BMS:

The Books Management System is Software that is designed to manage all the functions of the library It helps the librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.

This system completely automates all your activities. Implementing a Books Management System Software is the best way to systematically maintain, organize, and handle countless books. A BMS is used to maintain library records, It tracks the records of the number of books in the library, issued, how many books returned or renewed or late fine charges, etc. You can find books in an instant, issue/reissue books quickly, and manage all the data efficiently and orderly using this system. The purpose of BMS is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.

PYTHON:

Python is a widely used general-purpose, high-level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since its relatively easy to learn, Python has been adopted by non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical errors than other languages.

TKINTER:

Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it is simple and easy to work with. You don’t need to worry about the installation of the Tkinter module separately as it comes with Python already. It gives an object-oriented interface to the Tk GUI toolkit.

Widgets in Tkinter are the elements of a GUI application that provide various controls (such as Labels, Buttons, ComboBoxes, CheckBoxes, MenuBars, RadioButtons, and many more) to users interact with the application.

The multi-layered approach taken in designing Tkinter can have some disadvantages as far as execution speed is concerned. While this could constitute a problem with older, slower machines, most modern computers are fast enough to cope with the extra processing in a reasonable time. When speed is critical, proper care must be taken to write code that is as efficient as possible.

➢ OBJECTIVE:

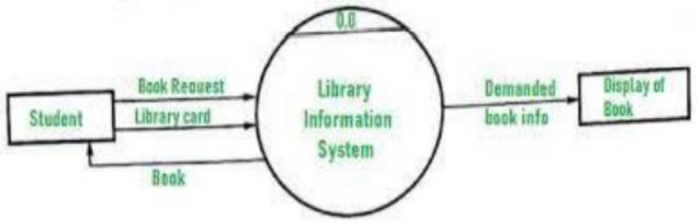
The main objective of a Books Management System is to manage and track the daily work of the library such as the number of books, separation according to the author and alphabet order, issuing books, return books, due calculations, etc.,

This software makes the work of the librarian and user easier to collect and store data and easily get all the necessary information with less effort.

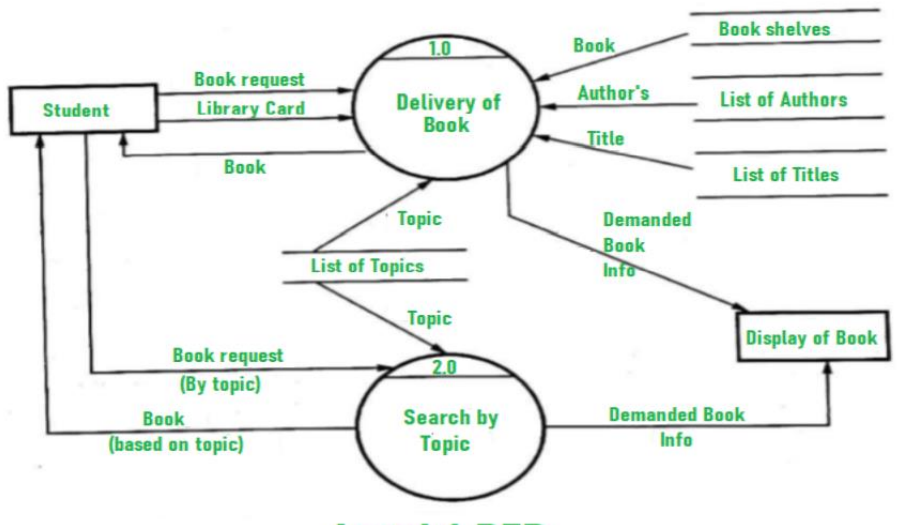
➢ DATA FLOW DIAGRAM:

Data Flow Diagram(DFD), The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have a control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. DFD helps us to visualize the major steps and data involved in the software system process.

LEVEL 0, DFD:



LEVEL 1, DFD:



Level 1 DFD

LEVEL 2, DFD:

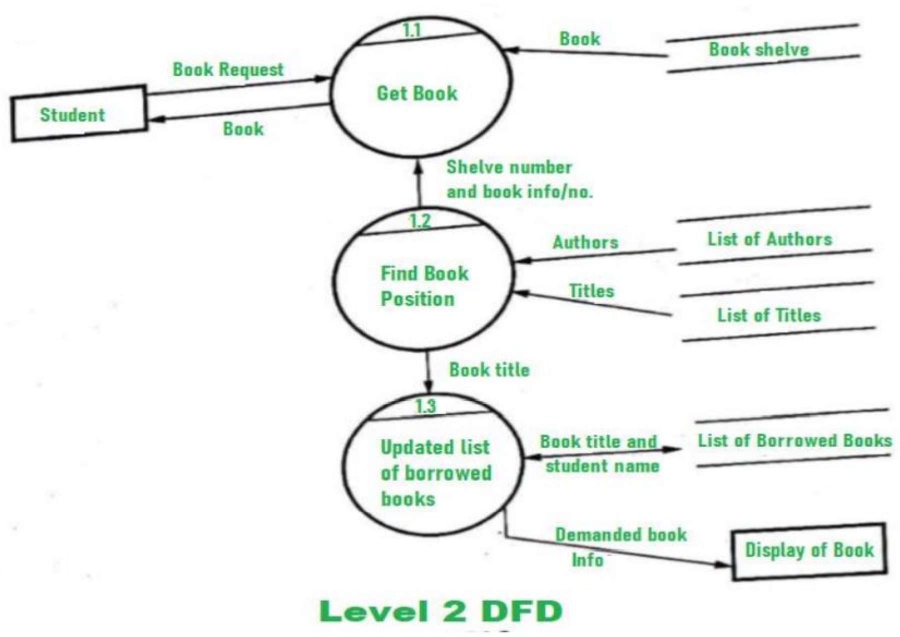


TABLE: Defining a table means, defining the names and datatypes of the attributes as well as the constraints to be applied to those attributes. Both MATLAB and Python use the same syntax to define tables.

TABLE USED :

TABLE – 1 (BOOK)
Author
Book Name
Price
Reference code

TABLE – 2 (STUDENT)
Name
Address
Gender
Phone number
E-Mail I’d
Registration Number
Password

Table1(book): To get the details of book, the user is requested to enter the author name and book name. Following that user will be given the details of book with price, if bought, the reference code is displayed which is used to keep track of the book.

Table2(student): The user is requested to provide the mentioned details(like : name,address,registration number,gender,phone number,email,password) to create profile and keep track of books issued to you. Users have to give accurate login details to get access to their profiles.

MODULE

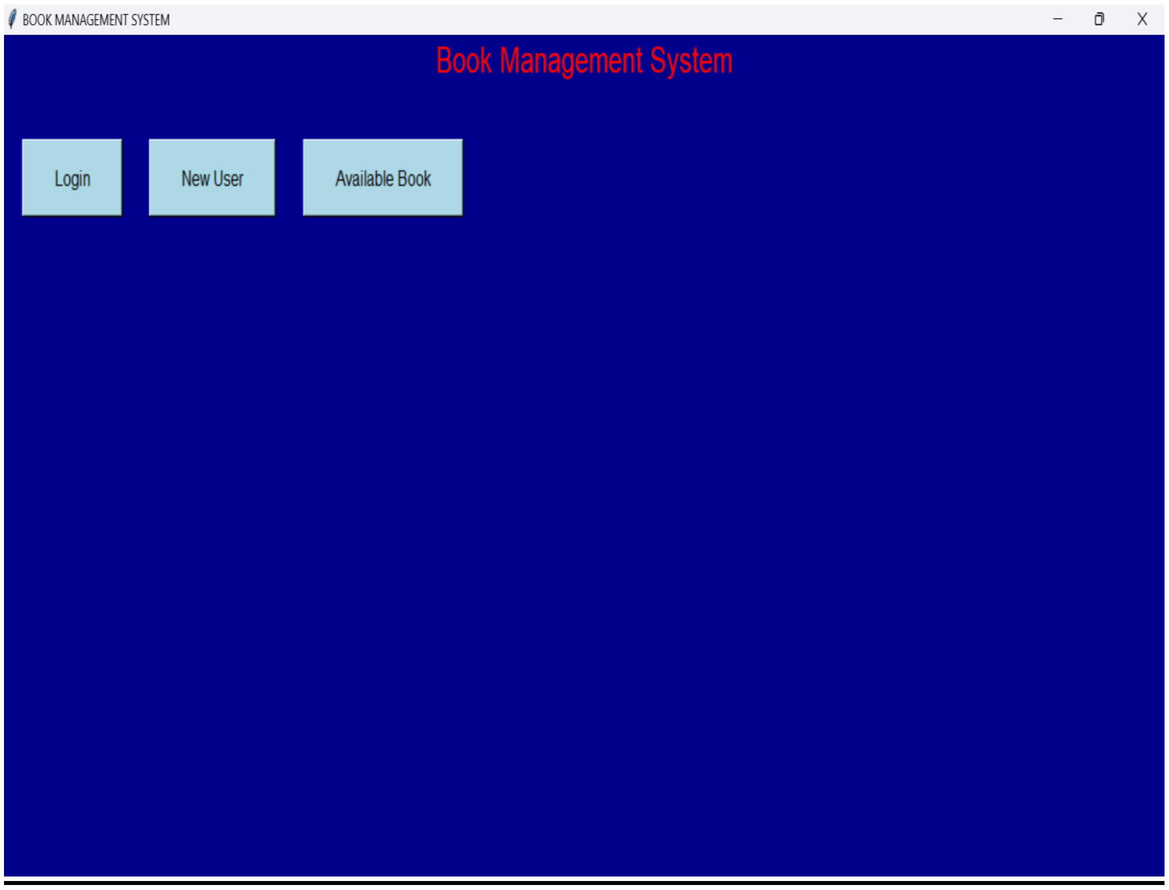
```
import tkinter
from tkinter import *
import tkinter.messagebox as msg
class BMS():
    def __init__(self,root):
        self.root=root
        self.root.geometry("500x200")
        self.root.title("BOOK MANAGEMENT SYSTEM")
        self.root.config(bg= "darkblue")
        title=Label(self.root,text="Book Management
System",bg="darkblue",font=('bold','20'),fg="red")
        title.pack()

login_button=Button(self.root,text="Login",padx=30,pady=15,bg="lightblue",font=('
bold','12'),command=self.login)
        login_button.place(x=20,y=80)
        new_user_button=Button(self.root,text="New
User",padx=30,pady=15,bg="lightblue",font=('bold','12'),command=self.register)
        new_user_button.place(x=160,y=80)
        available_book_button=Button(self.root,text="Available
Book",padx=30,pady=15,bg="lightblue",font=('bold','12'),command=self.login)
        available_book_button.place(x=330,y=80)
    def login(self):
        login_window=Tk()
        login_window.title("Login Page")
        login_window.geometry("500x300")
        login_window.config(bg="darkblue")

login_username=Label(login_window,text="Username",bg="darkblue",font=('bold','20'
),fg="white")
        login_username.place(x=30,y=50)

login_password=Label(login_window,text="Password",bg="darkblue",font=('bold','20'
),fg="white")
        login_password.place(x=30,y=130)
        self.loginusername_entry=Entry(login_window)
        self.loginusername_entry.place(x=200,y=60)
        self.loginpassword_entry=Entry(login_window)
        self.loginpassword_entry.place(x=200,y=140)
        login_submit_button=Button(login_window,text="Login 
Now",bg="lightblue",font=('bold','12'),command=self.request_submit)
        login_submit_button.place(x=160,y=190)
        new_user_button=Button(login_window,text="New 
User",bg="lightblue",padx=10,font=('bold','12'),command=self.register)
        new_user_button.place(x=400,y=190)

    def register(self):
        register_window=Tk()
        register_window.title("Student Registeration")
        register_window.geometry("500x400")
        register_window.config(bg="darkblue")
        register_name=Label(register_window,text="Student 
Details",bg="darkblue",font=('bold','20'),fg="red")
        register_name.pack()

register_name=Label(register_window,text="Name",bg="darkblue",font=('bold','20'),
fg="white")
        register_name.place(x=90,y=50)
        self.name_entry=Entry(register_window)
        self.name_entry.place(x=260,y=60)

register_address=Label(register_window,text="Address",bg="darkblue",font=('bold',
'20'),fg="white")
        register_address.place(x=90,y=100)
        self.address_entry=Entry(register_window)
        self.address_entry.place(x=260,y=110)
        register_gender=Label(register_window,text="Gender 
(M/F)",bg="darkblue",font=('bold','20'),fg="white")
        register_gender.place(x=90,y=150)
        self.gender_entry=Entry(register_window)
        self.gender_entry.place(x=260,y=160)
        register_mobile_no=Label(register_window,text="Mobile 
No",bg="darkblue",font=('bold','20'),fg="white")
        register_mobile_no.place(x=90,y=200)
        self.no_entry=Entry(register_window)
        self.no_entry.place(x=260,y=210)
        register_email=Label(register_window,text="Email 
ID",bg="darkblue",font=('bold','20'),fg="white")
        register_email.place(x=90,y=250)
        self.email_entry=Entry(register_window)
        self.email_entry.place(x=260,y=260)

register_password=Label(register_window,text="Password",bg="darkblue",fg="white",
font=('bold','20'))
```

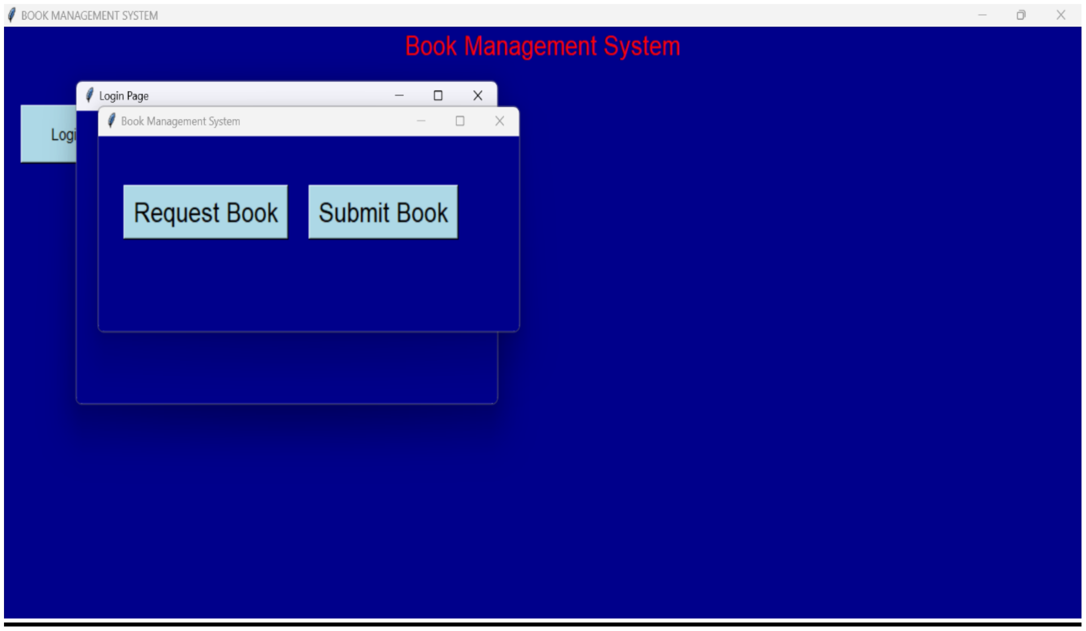
```
        register_password.place(x=90,y=300)
        self.password_entry=Entry(register_window)
        self.password_entry.place(x=260,y=310)

register_submit_button=Button(register_window,text="SUBMIT",bg="lightblue",font=(
'bold','12'),command=self.add_student)
        register_submit_button.place(x=190,y=350)
    def request_submit(self):
        request_window=Tk()
        request_window.geometry("500x200")
        request_window.title("Book Management System")
        request_window.config(bg="darkblue")
        request_button=Button(request_window,text="Request
Book",bg="lightblue",font=('bold','20'),command=self.request_book)
        request_button.place(x=30,y=50)
        submit_button=Button(request_window,text="Submit
Book",bg="lightblue",font=('bold','20'),command=self.submit_book)
        submit_button.place(x=250,y=50)
    def request_book(self):
        rwindow=Tk()
        rwindow.geometry("500x400")
        rwindow.title("Request Book")
        rwindow.config(bg="darkblue")

register_name=Label(rwindow,text="Author",bg="darkblue",font=('bold','20'),fg="wh
ite")
        register_name.place(x=90,y=50)
        self.name_en=Entry(rwindow)
        self.name_en.place(x=260,y=60)
        book_name_request=Label(rwindow,text="Book
Name",bg="darkblue",font=('bold','20'),fg="white")
        book_name_request.place(x=90,y=100)
        self.book_name_entry=Entry(rwindow)
        self.book_name_entry.place(x=260,y=110)

order_book=Button(rwindow,text="ORDER",bg="darkblue",fg="white",font=('bold','20'
))
        order_book.place(x=200,y=250)
    def submit_book(self):
        swindow=Tk()
        swindow.geometry("500x300")
        swindow.title("Submit Book")
        swindow.config(bg="darkblue")

s_author=Label(swindow,text="Author",bg="darkblue",font=('bold','20'),fg="white")
```

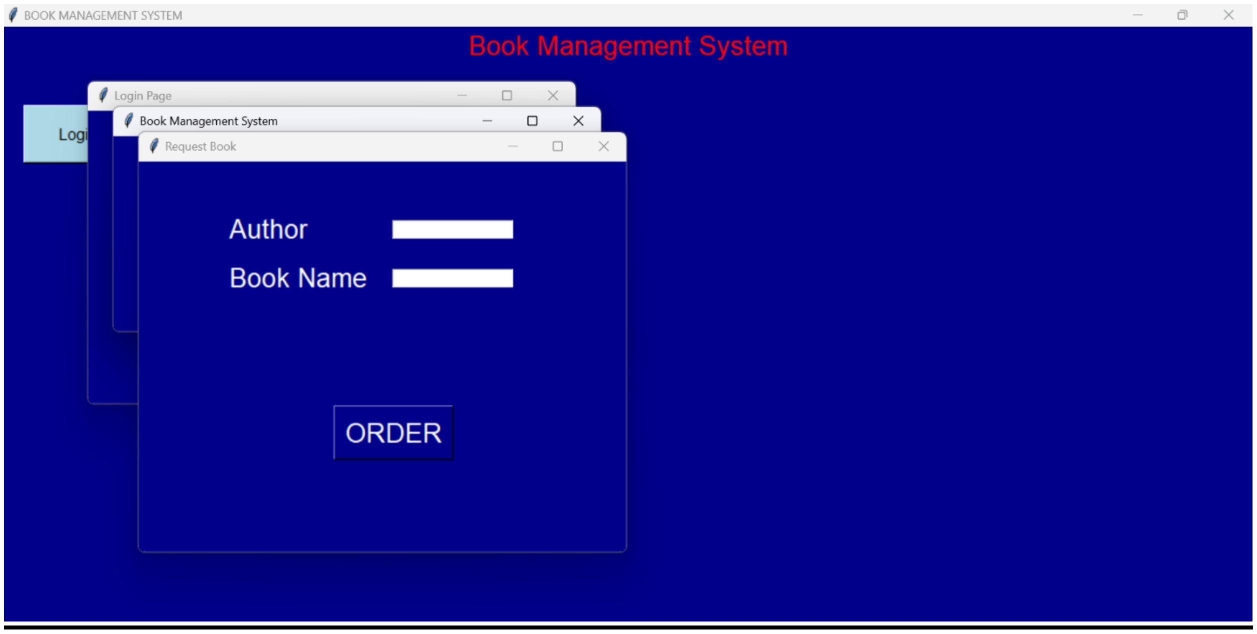
```
        s_author.place(x=60,y=50)
        self.author_entry=Entry(swindow)
        self.author_entry.place(x=260,y=60)
        book_name=Label(swindow,text="Book
Name",bg="darkblue",font=('bold','20'),fg="white")
        book_name.place(x=60,y=100)
        self.book_entry=Entry(swindow)
        self.book_entry.place(x=260,y=110)

s_price=Label(swindow,text="Price",bg="darkblue",font=('bold','20'),fg="white")
        s_price.place(x=60,y=150)
        self.price_entry=Entry(swindow)
        self.price_entry.place(x=260,y=150)

s_submit=Button(swindow,text="Submit",bg="lightblue",font=('bold','14'),command=s
elf.add_book)
        s_submit.place(x=190,y=200)
    def add_book(self):
        import mysql.connector

mydb=mysql.connector.connect(host='localhost',port='3306',username='root',passwor
d='yuvi@1234',database='library_management')
        mycursor=mydb.cursor()
        author=self.author_entry.get()
        book=self.book_entry.get()
        price=self.price_entry.get()
        mycursor.execute("insert into book values(%s,%s,%s)",(author,book,price))
        mydb.commit()
        msg.showinfo("Book","Book added to stock")
    def add_student(self):
        import mysql.connector

mydb=mysql.connector.connect(host='localhost',port='3306',username='root',passwor
d='yuvi@1234',database='library_management')
        mycursor=mydb.cursor()
        name=self.name_entry.get()
        address=self.address_entry.get()
        gender=self.gender_entry.get()
        phone=self.no_entry.get()
        email=self.email_entry.get()
        password=self.password_entry.get()
        mycursor.execute("insert into student
values(%s,%s,%s,%s,%s,%s)",(name,address,gender,phone,email,password))
        mydb.commit()
        msg.showinfo("student database","new student added")
```

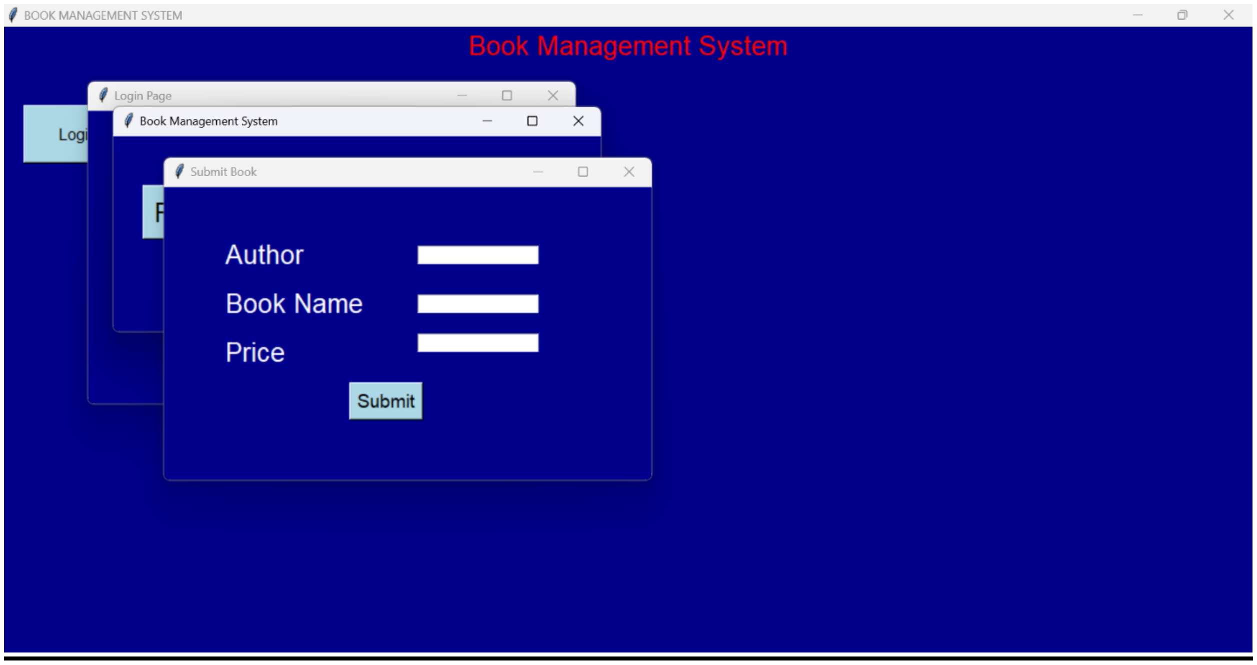
```python
root=Tk()
obj=BMS(root)
root.mainloop()
```

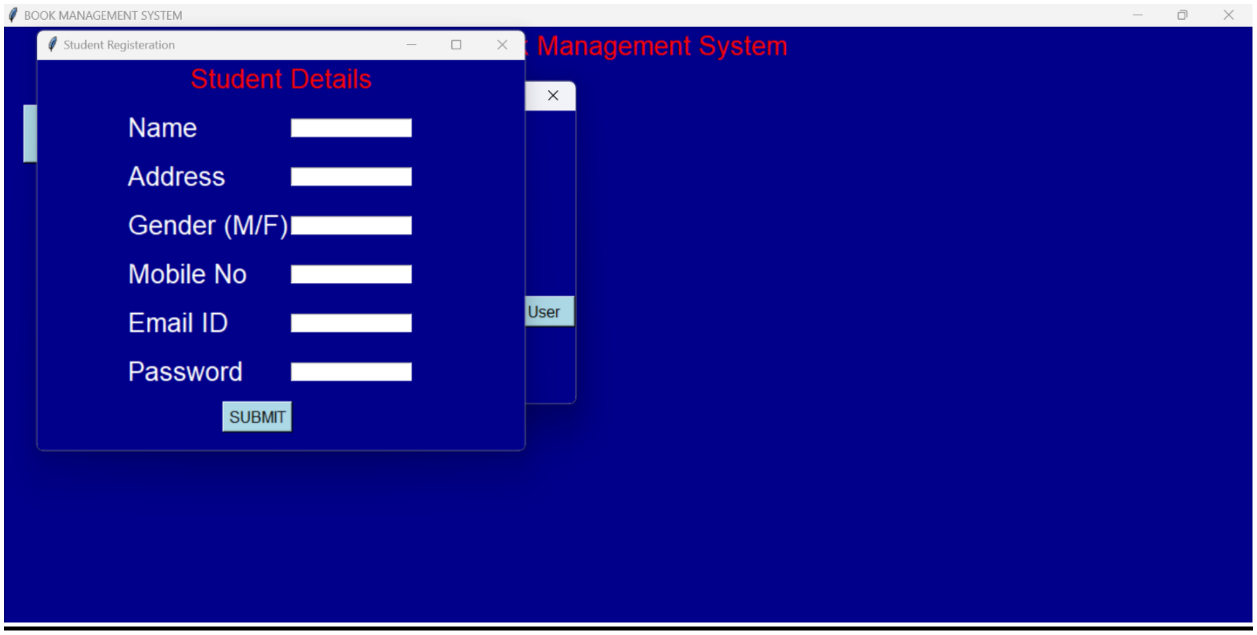
❖ OUTPUTUT/RESULT:




BOOK MANAGEMENT SYSTEM
Book Management System
Login Page
Book Management System
Request Book
Submit Book


BOOK MANAGEMENT SYSTEM
Book Management System
Login Page
Book Management System
Request Book
Author
Book Name
ORDER


BOOK MANAGEMENT SYSTEM
Book Management System
Login Page
Logi
Book Management System
Submit Book
Author
Book Name
Price
Submit


BOOK MANAGEMENT SYSTEM
Student Registeration
Management System
Student Details
Name
Address
Gender (M/F)
Mobile No
Email ID
Password
SUBMIT
User

REFERENCE :

https://www.w3schools.com/python/

https://www.geeksforgeeks.org/python-gui-tkinter/

https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/

❖ CONCLUSION:

The software is built keeping the user in mind so that they do not spend too much of their time learning how to use the software. It is user-friendly and the personal information is highly secured in the database. All the information of user data stored in the data base is assured to be not leaked or misused. The changes can be made if necessary in the form of an update.