Class 12th
Informatics Practices
Project Work
On
"LIBRARY MANAGEMENT SYSTEM USING PYTHON"

Name Of Student: Ronak Kumar Bothra

Name Of School: Modi Public School

Subject: Informatics Practices (CODE 065)

Session: 2024-25

Submitted To: Mr. Vijay Nagar

Date Of Submission: 16th December 2024

INDEX

CERTIFICATE

This is to certify that **Ronak Kumar Bothra** has successfully completed the Informatics Practices project for Class 12 under the CBSE curriculum. The project titled **Library Management System Using Python** was undertaken at **Modi Public School** during the academic year **2024-25**.

This project involved the application of various concepts covered in the Informatics Practices curriculum, including data handling using Pandas Library and programming in Python. The student demonstrated outstanding skills and creativity in executing the project, showcasing their ability to apply theoretical knowledge in practical scenarios.

We commend the student for their dedication and hard work throughout this project, which reflects their commitment to excellence in the field of Informatics.

Vijay Nagar Sir

Examiner

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who supported me throughout the journey of my Informatics Practices project. This endeavor would not have been possible without the encouragement, guidance, and contributions of several individuals.

First and foremost, I would like to extend my sincere thanks to my teacher, Vijay Nagar Sir, whose unwavering support and insightful feedback were instrumental in shaping my project.

Also, I would like to thank my family, whose constant encouragement and understanding provided me with the motivation to pursue my academic goals. Their belief in my abilities and willingness to help me create a conducive study environment were invaluable.

Lastly, I would like to acknowledge the resources and materials provided by Modi Public School, which facilitated my research and project development.

To all who contributed to my project, thank you for your support. Your encouragement has been a beacon of motivation and has inspired me to strive for excellence.

INTRODUCTION

A Library Management System (LMS) is a type of software that assists in automating the organization of resources such as books, library members, and even transactions such as issuing and returning members. Libraries, in the past, have relied upon manual record keeping, which came with its own challenges such as being tedious as well as the chances of human error. Creating such a system will greatly assist the library by improving the overall accuracy and efficiency of operations.

This project is aimed at developing a Library Management System using Python programming language with its various libraries like Pandas for accomplishing many subtasks related to the management of a library. The emphasis is to develop a system that will help library staff in books and members and transactions maintenance with minimum input and maximum effectiveness as well as accuracy.

The system allows users (library staff or administrators) to perform a variety of tasks, including

- Managing Books and Members records.
- Handling Book Issue and Return Transactions.

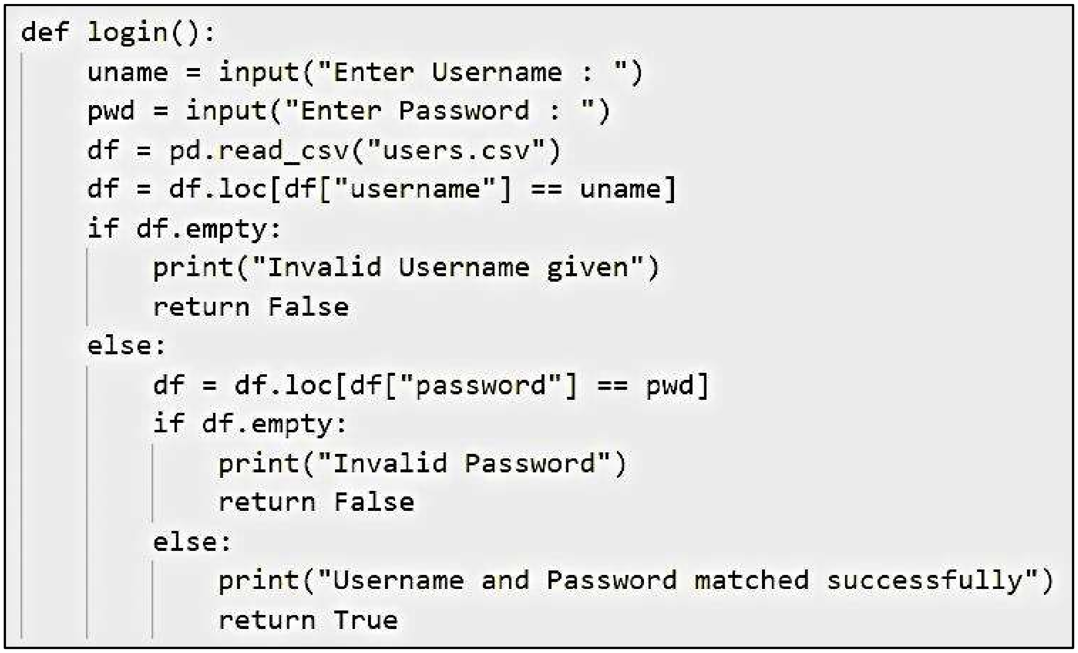
APPROACH

- **Planning**: The planning process began with analyzing the core tasks that needed to be automated in the Library Management System. This included identifying key operations such as adding and removing books from the library's collection, managing library members, and handling the borrowing and return of books. By breaking down these essential tasks, the project could be organized into manageable components, ensuring that each function was efficiently addressed and that the system would streamline the overall library operations.
- **Design**: The system was divided into modules:
  - **Book Management**: Includes adding new books, searching, deleting, and displaying all books.
  - **Member Management**: Includes adding members, searching, deleting, and displaying members.
  - **Issue and Return Management**: Handles issuing and returning books to/from members.
- **Implementation**: Python was used as the primary programming language, with CSV files as the storage medium for books, members, and issued books. Pandas is used for handling these CSV files, making data manipulation more efficient

FEATURES

1. Admin Authentication:

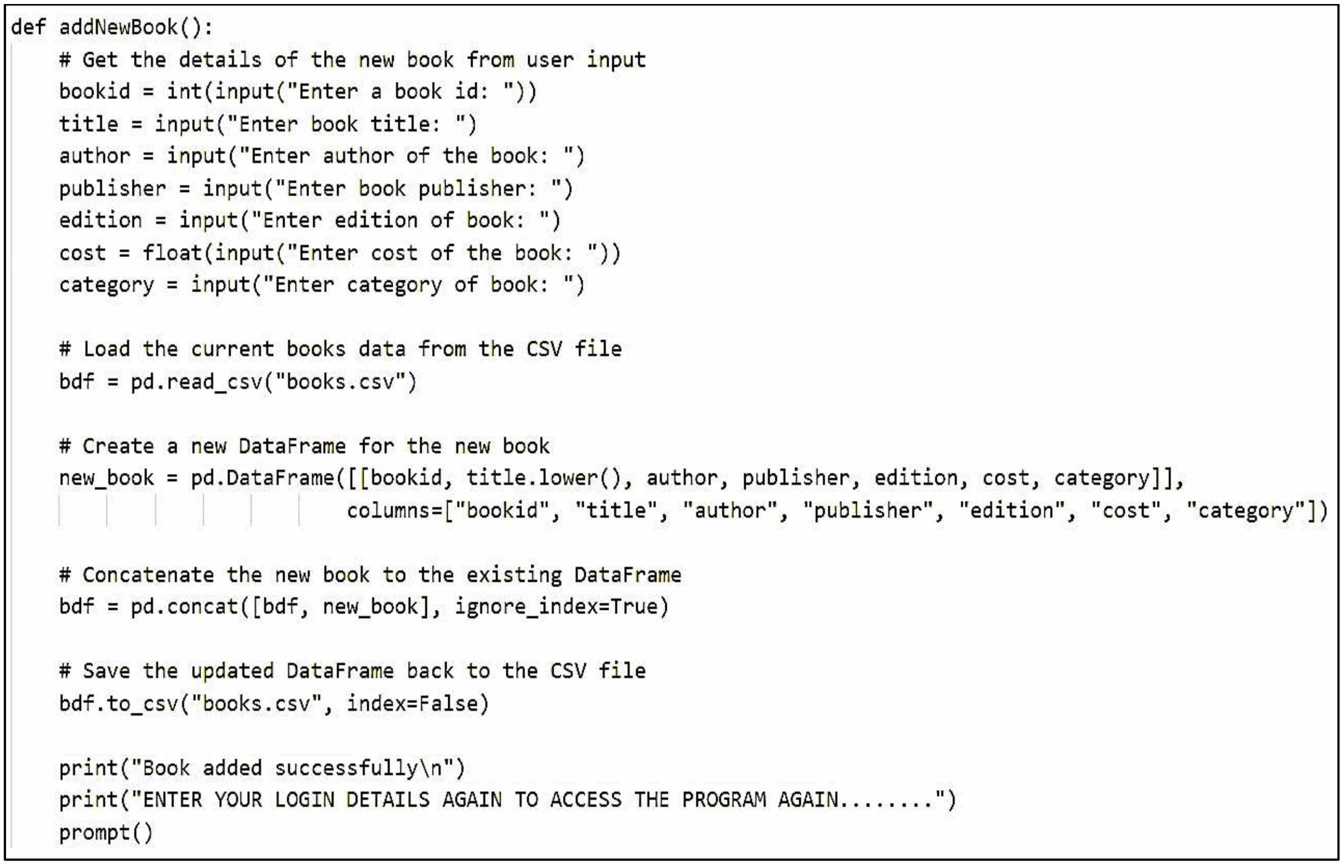
A login system is implemented where users must enter a valid username and password to access the system. This is done by using a function called login() and storing the user login detail on a file named users.csv.



```
def login():
    uname = input("Enter Username : ")
    pwd = input("Enter Password : ")
    df = pd.read_csv("users.csv")
    df = df.loc[df["username"] == uname]
    if df.empty:
        print("Invalid Username given")
        return False
    else:
        df = df.loc[df["password"] == pwd]
        if df.empty:
            print("Invalid Password")
            return False
        else:
            print("Username and Password matched successfully")
            return True
```

2. Adding a New Book:

Users can add new books with attributes like book ID, title, author, publisher, edition, cost, and category. This is implemented using a function named addNewBook().

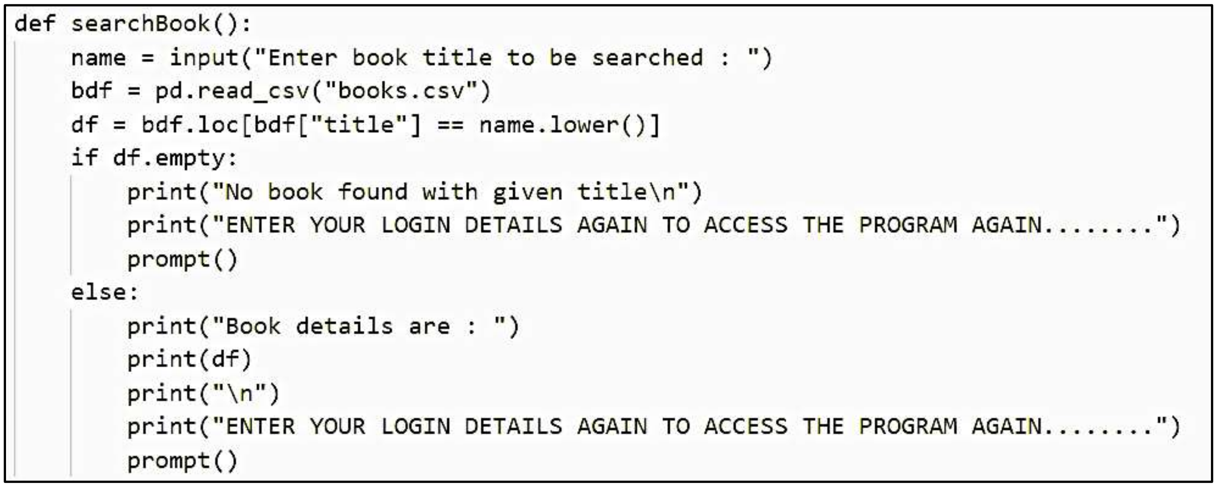


```
def addNewBook():
    # Get the details of the new book from user input
    bookid = int(input("Enter a book id: "))
    title = input("Enter book title: ")
    author = input("Enter author of the book: ")
    publisher = input("Enter book publisher: ")
    edition = input("Enter edition of book: ")
    cost = float(input("Enter cost of the book: "))
    category = input("Enter category of book: ")

    # Load the current books data from the CSV file
    bdf = pd.read_csv("books.csv")

    # Create a new DataFrame for the new book
    new_book = pd.DataFrame([[bookid, title.lower(), author, publisher, edition, cost, category]],
                            columns=["bookid", "title", "author", "publisher", "edition", "cost", "category"])

    # Concatenate the new book to the existing DataFrame
    bdf = pd.concat([bdf, new_book], ignore_index=True)

    # Save the updated DataFrame back to the CSV file
    bdf.to_csv("books.csv", index=False)

    print("Book added successfully\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
    prompt()
```

3. Searching for a Book:

Allows searching for a book by its title. This is implemented by using a function called searchBook()

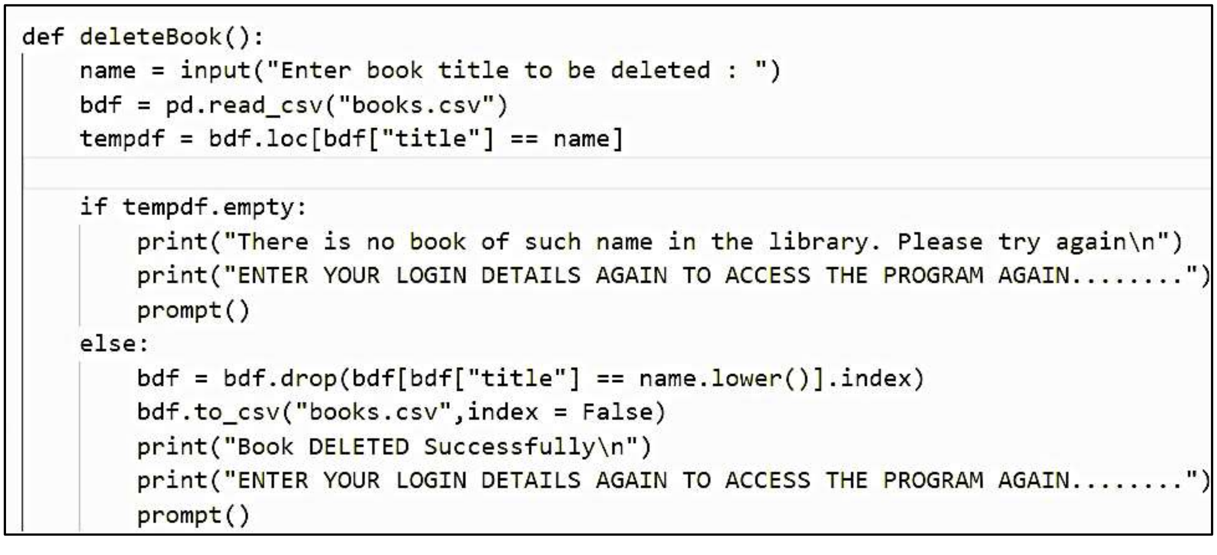
.



```
def searchBook():
    name = input("Enter book title to be searched : ")
    bdf = pd.read_csv("books.csv")
    df = bdf.loc[bdf["title"] == name.lower()]
    if df.empty:
        print("No book found with given title\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
    else:
        print("Book details are : ")
        print(df)
        print("\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
```

4. Delete a Book:

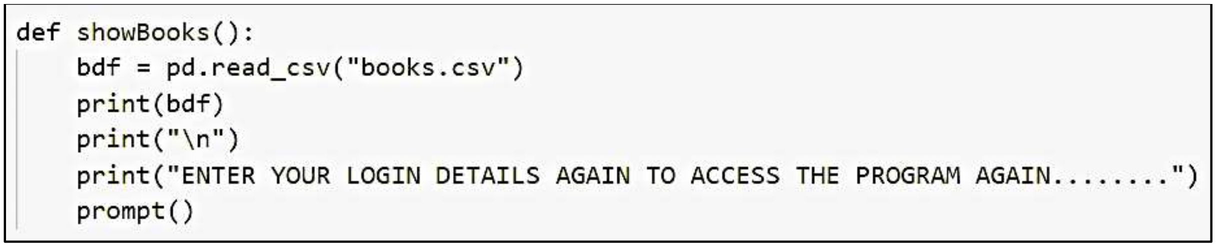
Enables deletion of a book by title. This is implemented by using a function called deleteBook().



```
def deleteBook():
    name = input("Enter book title to be deleted : ")
    bdf = pd.read_csv("books.csv")
    tempdf = bdf.loc[bdf["title"] == name]

    if tempdf.empty:
        print("There is no book of such name in the library. Please try again\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
    else:
        bdf = bdf.drop(bdf[bdf["title"] == name.lower()].index)
        bdf.to_csv("books.csv",index = False)
        print("Book DELETED Successfully\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
```

5. Show all Books:

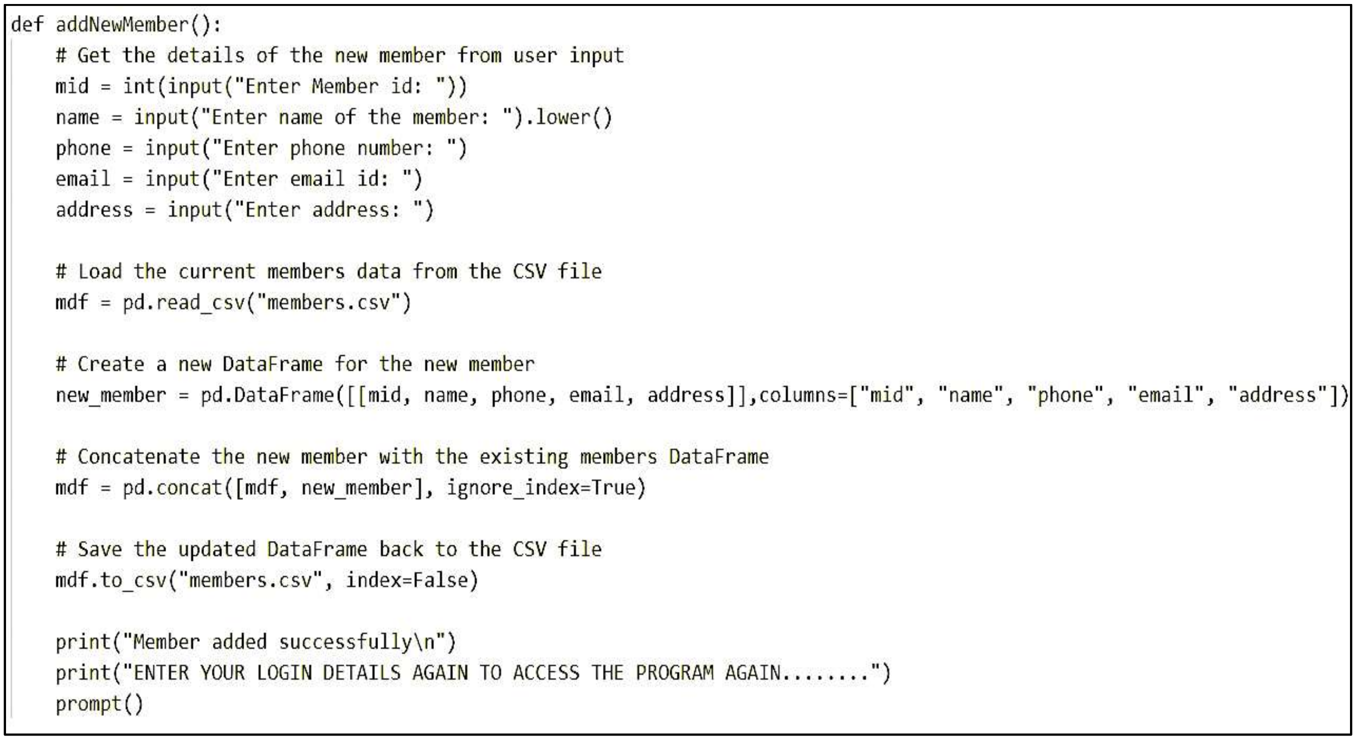
Displays a list of all available books in the library. This is implemented by using a function called showBooks().



```
def showBooks():
    bdf = pd.read_csv("books.csv")
    print(bdf)
    print("\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
    prompt()
```

6. Adding a Member:

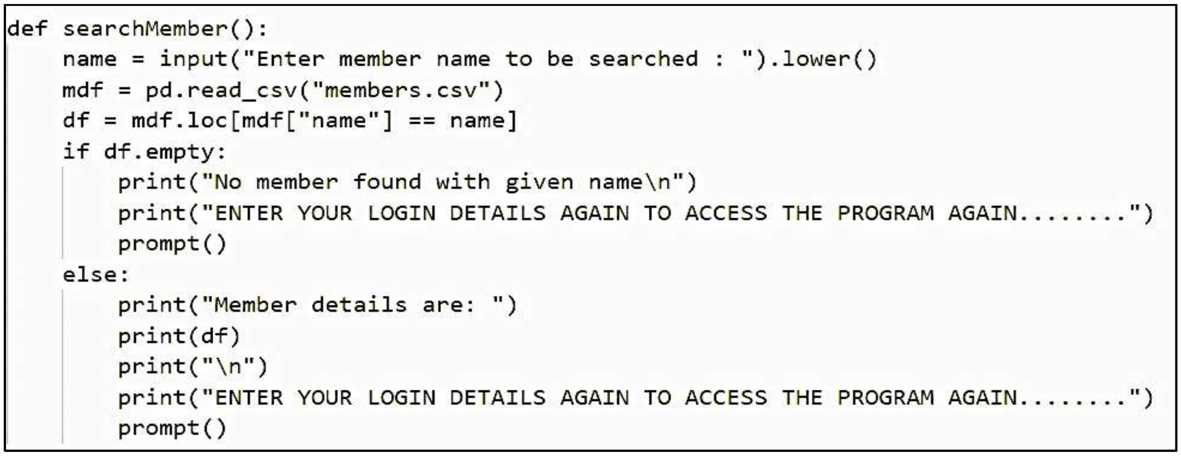
Adds a new library member with details such as name, phone, email, and address. This is implemented by addNewMember() function.



```
def addNewMember():
    # Get the details of the new member from user input
    mid = int(input("Enter Member id: "))
    name = input("Enter name of the member: ").lower()
    phone = input("Enter phone number: ")
    email = input("Enter email id: ")
    address = input("Enter address: ")

    # Load the current members data from the CSV file
    mdf = pd.read_csv("members.csv")

    # Create a new DataFrame for the new member
    new_member = pd.DataFrame([[mid, name, phone, email, address]],columns=["mid", "name", "phone", "email", "address"])

    # Concatenate the new member with the existing members DataFrame
    mdf = pd.concat([mdf, new_member], ignore_index=True)

    # Save the updated DataFrame back to the CSV file
    mdf.to_csv("members.csv", index=False)

    print("Member added successfully\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
    prompt()
```

7. Searching for a Member:

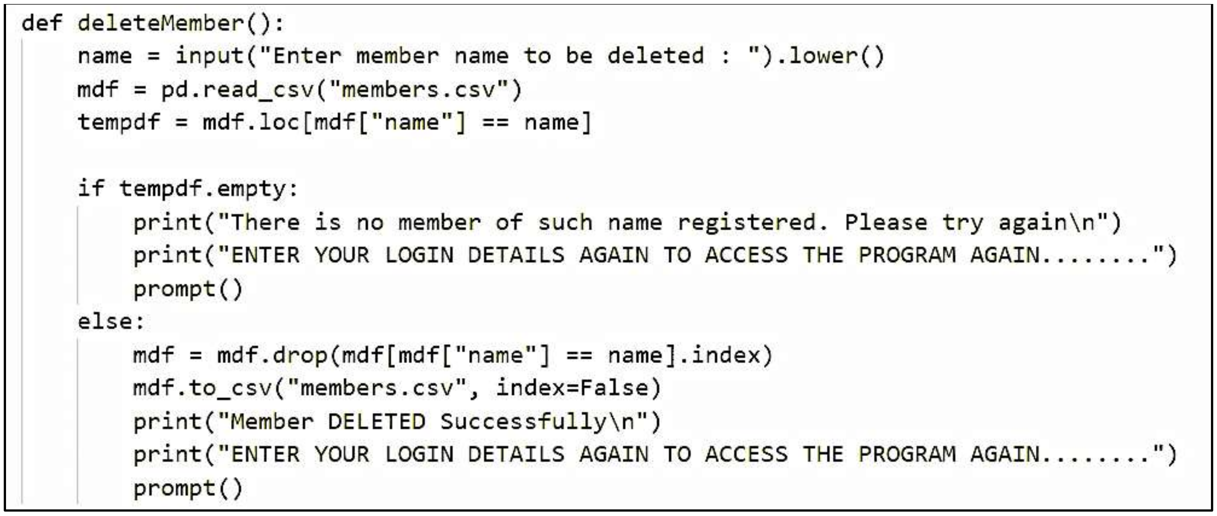
Allows searching for a member by name. This was implemented by the means of a function named searchMember().



```
def searchMember():
    name = input("Enter member name to be searched : ").lower()
    mdf = pd.read_csv("members.csv")
    df = mdf.loc[mdf["name"] == name]
    if df.empty:
        print("No member found with given name\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
    else:
        print("Member details are: ")
        print(df)
        print("\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
```

8. Delete a Member:

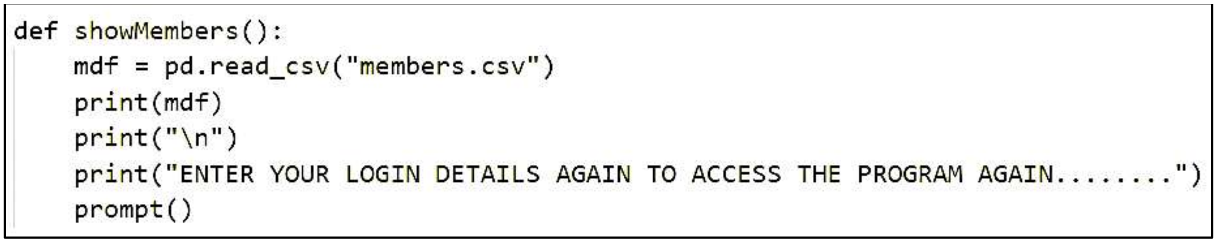
Enables deletion of a member by name. This is done with the incorporation of a function called deleteMember(), It asks for the name of the member to be deleted.



```
def deleteMember():
    name = input("Enter member name to be deleted : ").lower()
    mdf = pd.read_csv("members.csv")
    tempdf = mdf.loc[mdf["name"] == name]

    if tempdf.empty:
        print("There is no member of such name registered. Please try again\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
    else:
        mdf = mdf.drop(mdf[mdf["name"] == name].index)
        mdf.to_csv("members.csv", index=False)
        print("Member DELETED Successfully\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
```

9. Show all Members:

Displays a list of all members in the library. This is implemented by using a function called showMembers()

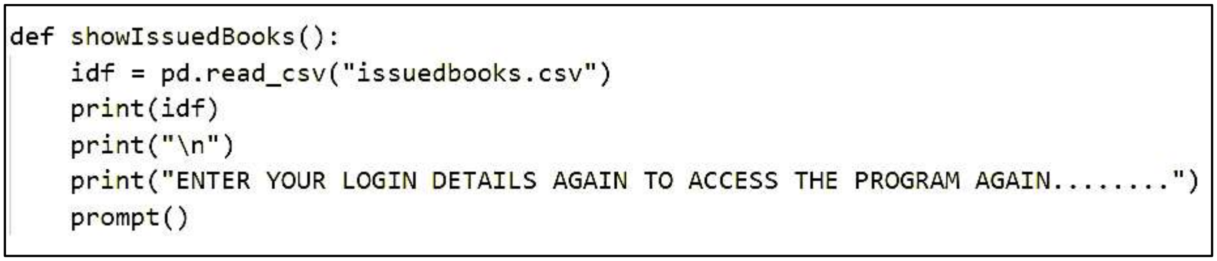


```
def showMembers():
    mdf = pd.read_csv("members.csv")
    print(mdf)
    print("\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN.......")
    prompt()
```

____

10. Showing all Issued Books:

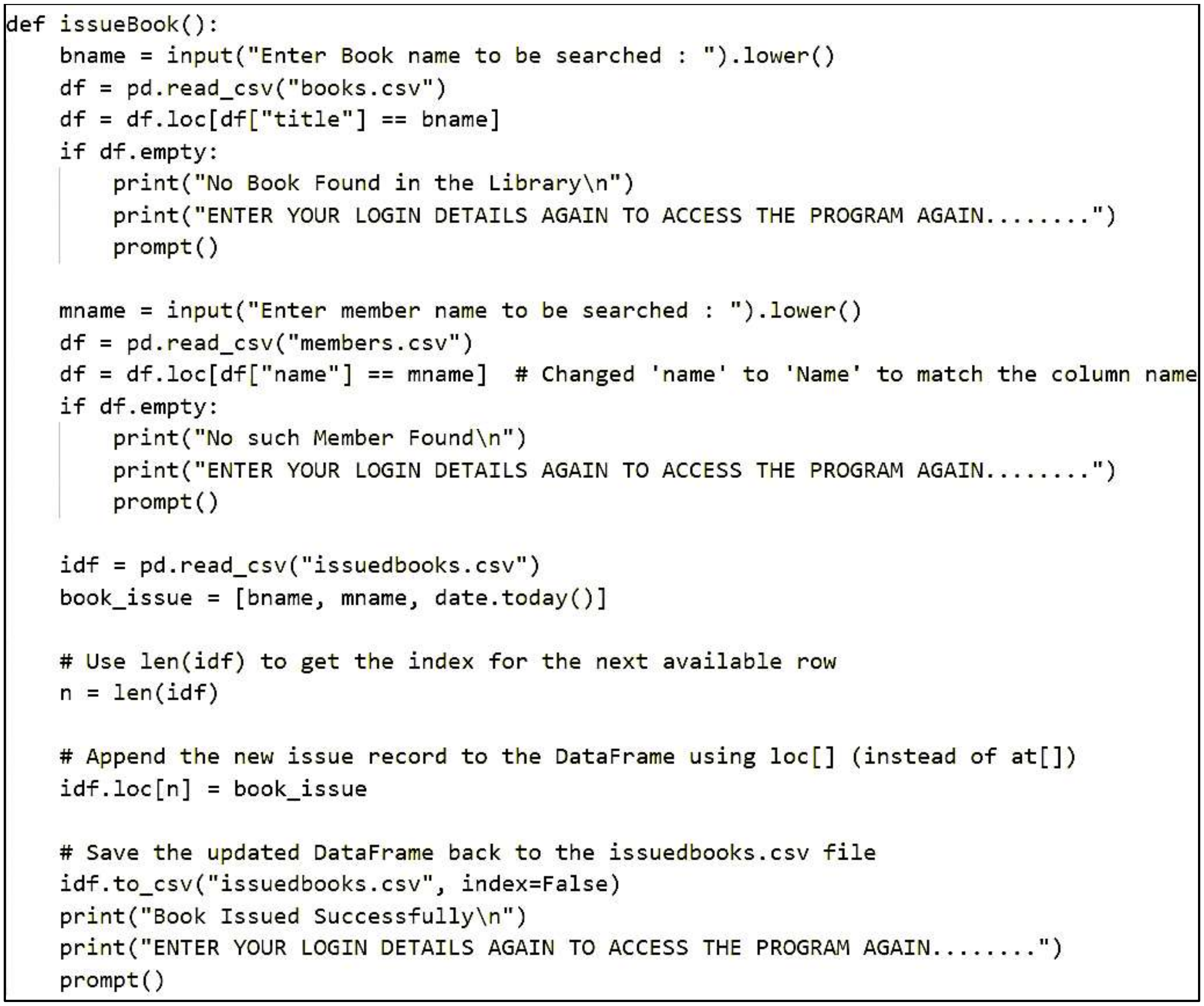
The showIssuedBooks() function does this while also auto entering the date of issue using the datetime library



```
def showIssuedBooks():
    idf = pd.read_csv("issuedbooks.csv")
    print(idf)
    print("\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN.......")
    prompt()
```

11. Issuing a Book:

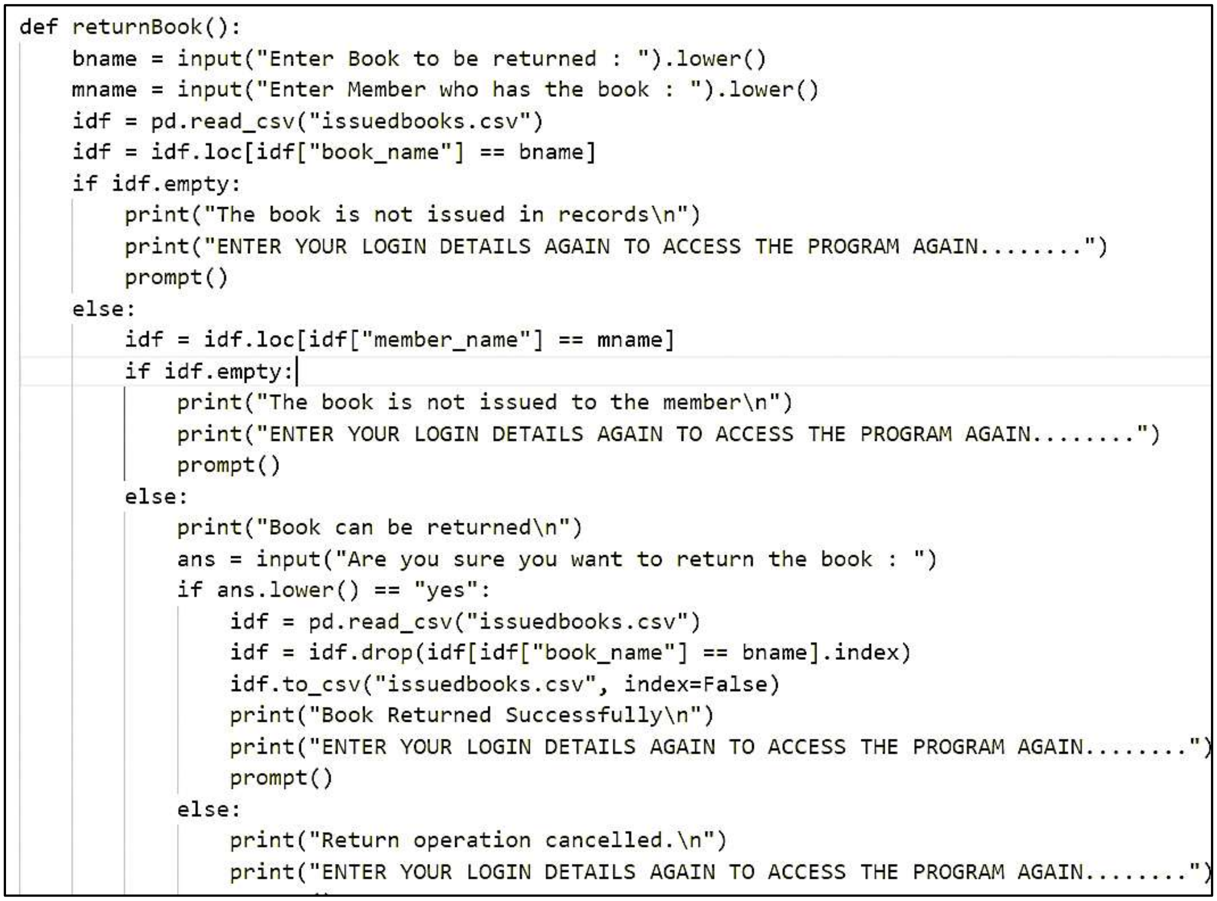
This is the feature of the LMS that deals with the transactions related stuff. It is down using the functions issueBook() along with the file issuedBooks,csv for store the data of the transactions.



```
def issueBook():
    bname = input("Enter Book name to be searched : ").lower()
    df = pd.read_csv("books.csv")
    df = df.loc[df["title"] == bname]
    if df.empty:
        print("No Book Found in the Library\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()

    mname = input("Enter member name to be searched : ").lower()
    df = pd.read_csv("members.csv")
    df = df.loc[df["name"] == mname]  # Changed 'name' to 'Name' to match the column name
    if df.empty:
        print("No such Member Found\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()

    idf = pd.read_csv("issuedbooks.csv")
    book_issue = [bname, mname, date.today()]

    # Use len(idf) to get the index for the next available row
    n = len(idf)

    # Append the new issue record to the DataFrame using loc[] (instead of at[])
    idf.loc[n] = book_issue

    # Save the updated DataFrame back to the issuedbooks.csv file
    idf.to_csv("issuedbooks.csv", index=False)
    print("Book Issued Successfully\n")
    print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
    prompt()
```

12. Returning a Book:

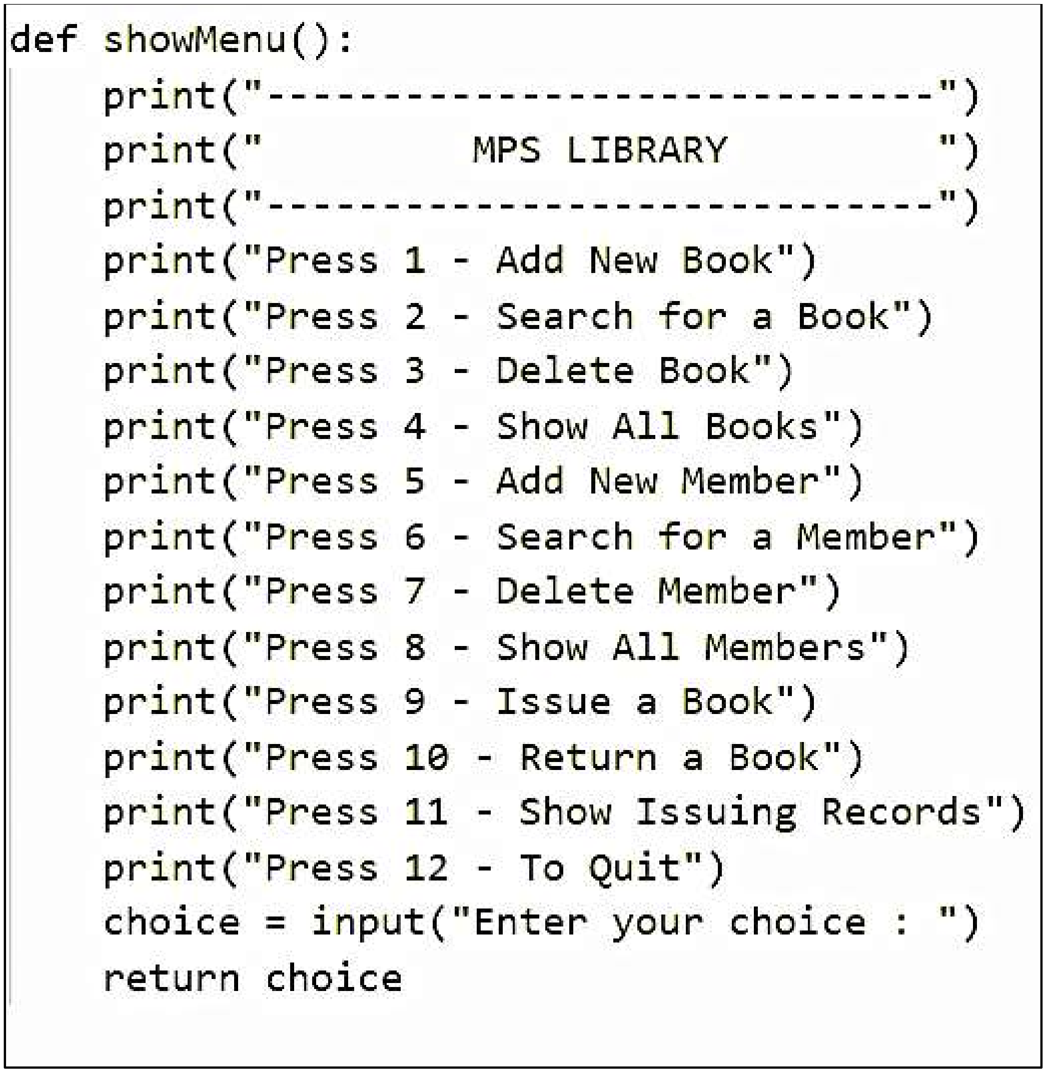
This is the feature of the LMS that also deals with the transactions related stuff. It is down using the functions returnBook().



```
def returnBook():
    bname = input("Enter Book to be returned : ").lower()
    mname = input("Enter Member who has the book : ").lower()
    idf = pd.read_csv("issuedbooks.csv")
    idf = idf.loc[idf["book_name"] == bname]
    if idf.empty:
        print("The book is not issued in records\n")
        print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
        prompt()
    else:
        idf = idf.loc[idf["member_name"] == mname]
        if idf.empty:
            print("The book is not issued to the member\n")
            print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
            prompt()
        else:
            print("Book can be returned\n")
            ans = input("Are you sure you want to return the book : ")
            if ans.lower() == "yes":
                idf = pd.read_csv("issuedbooks.csv")
                idf = idf.drop(idf[idf["book_name"] == bname].index)
                idf.to_csv("issuedbooks.csv", index=False)
                print("Book Returned Successfully\n")
                print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
                prompt()
            else:
                print("Return operation cancelled.\n")
                print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
```

13. Main Menu:

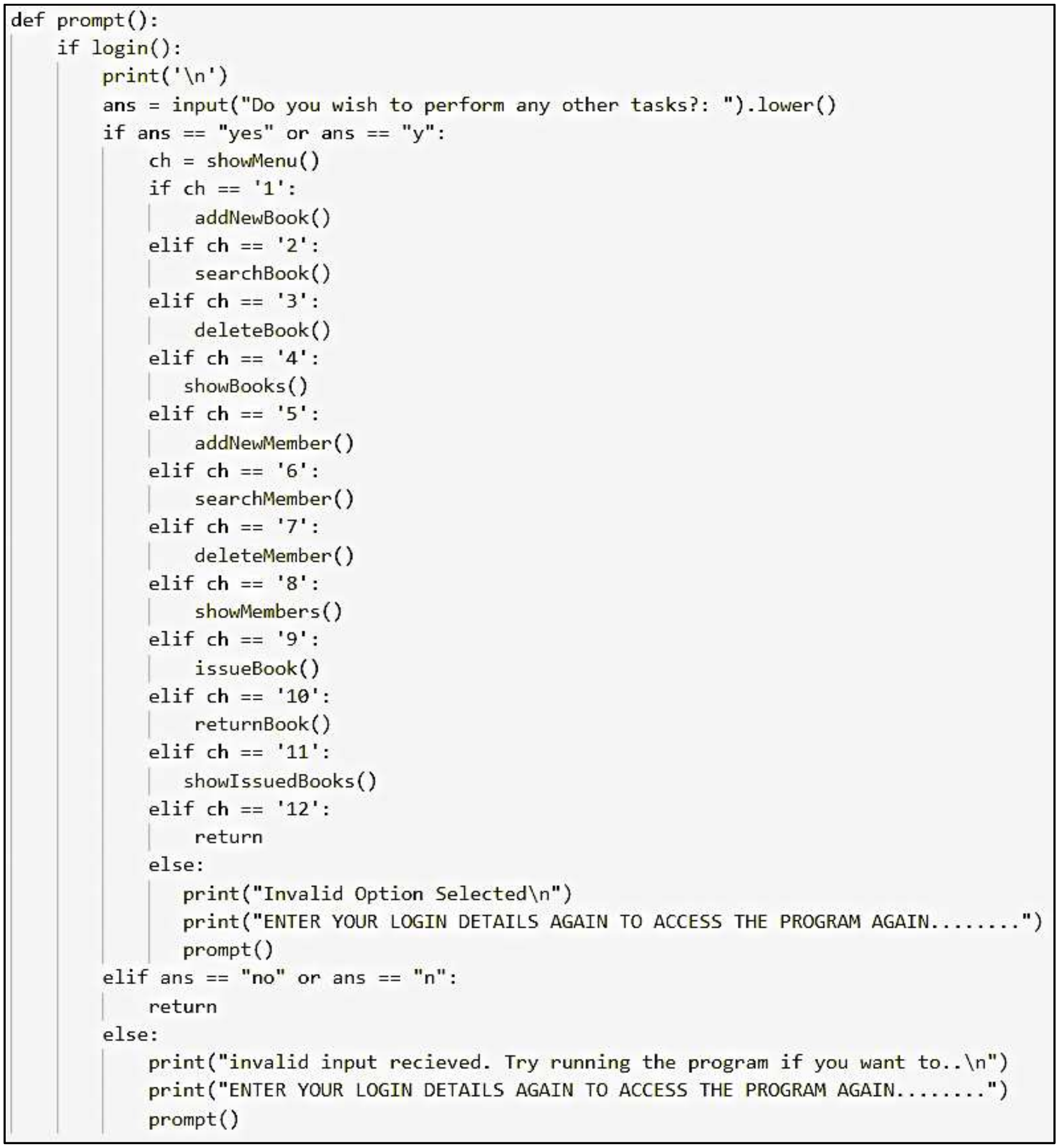
The main menu is presented using the showMenu() function as shown below:



```
def showMenu():
    print("-----------------------------")
    print("         MPS LIBRARY         ")
    print("-----------------------------")
    print("Press 1 - Add New Book")
    print("Press 2 - Search for a Book")
    print("Press 3 - Delete Book")
    print("Press 4 - Show All Books")
    print("Press 5 - Add New Member")
    print("Press 6 - Search for a Member")
    print("Press 7 - Delete Member")
    print("Press 8 - Show All Members")
    print("Press 9 - Issue a Book")
    print("Press 10 - Return a Book")
    print("Press 11 - Show Issuing Records")
    print("Press 12 - To Quit")
    choice = input("Enter your choice : ")
    return choice
```

14. Logic behind the Menu:

This function is basically connecting different functions with the menu.



```python
def prompt():
    if login():
        print('\n')
        ans = input("Do you wish to perform any other tasks?: ").lower()
        if ans == "yes" or ans == "y":
            ch = showMenu()
            if ch == '1':
                addNewBook()
            elif ch == '2':
                searchBook()
            elif ch == '3':
                deleteBook()
            elif ch == '4':
                showBooks()
            elif ch == '5':
                addNewMember()
            elif ch == '6':
                searchMember()
            elif ch == '7':
                deleteMember()
            elif ch == '8':
                showMembers()
            elif ch == '9':
                issueBook()
            elif ch == '10':
                returnBook()
            elif ch == '11':
                showIssuedBooks()
            elif ch == '12':
                return
            else:
                print("Invalid Option Selected\n")
                print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
                prompt()
        elif ans == "no" or ans == "n":
            return
        else:
            print("invalid input recieved. Try running the program if you want to..\n")
            print("ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........")
            prompt()
```

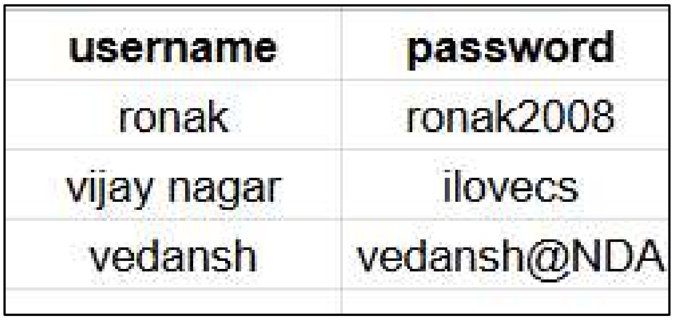
How to Use the Program

1. **Login**: When you run the program, you'll first be prompted to log in with a valid username and password. If authentication is successful, the system will allow access to the menu.
2. **Main Menu**: After logging in, you will see a menu with various options to manage books, members, and issued books. Here's a summary of the options:
    a. Press **1** to add a new book.
    b. Press **2** to search for a book by title.
    c. Press **3** to delete a book by title.
    d. Press **4** to display all books.
    e. Press **5** to add a new member.
    f. Press **6** to search for a member.
    g. Press **7** to delete a member.
    h. Press **8** to display all members.
    i. Press **9** to issue a book to a member.
    j. Press **10** to return a book.
    k. Press **11** to view all issued books.
    l. Press **12** to exit the program.
3. **Perform Operations**: Choose the desired option and follow the on-screen prompts to interact with the system (such as entering book titles, member names, etc.).
4. **Data Storage**: The program uses CSV files (books.csv, members.csv, issuedbooks.csv) to store data persistently. Ensure these files exist in the same directory as the Python script.

DATA STORAGE

1. Admin Username And Passwords:

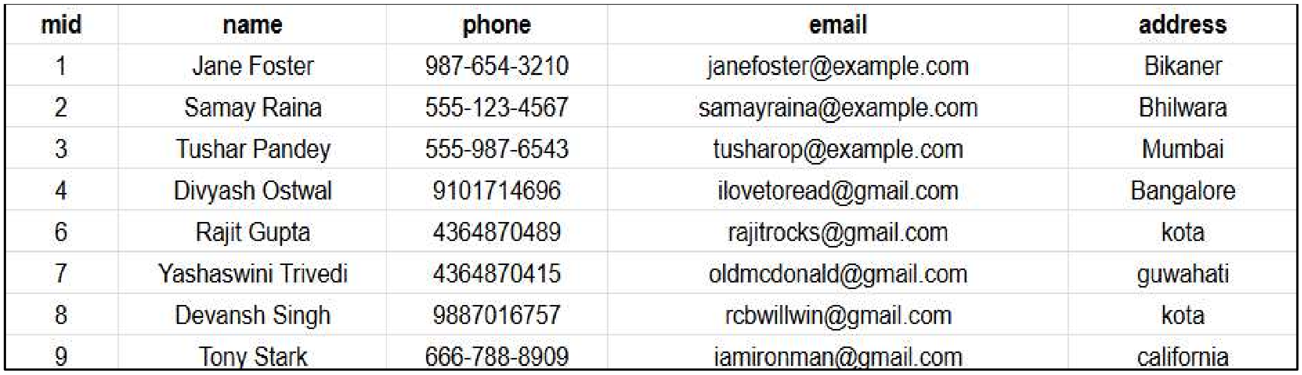
The admins' username and passwords are stored in the file named 'users.csv'.



username	password
ronak	ronak2008
vijay nagar	ilovecs
vedansh	vedansh@NDA

2. Members of the Library:

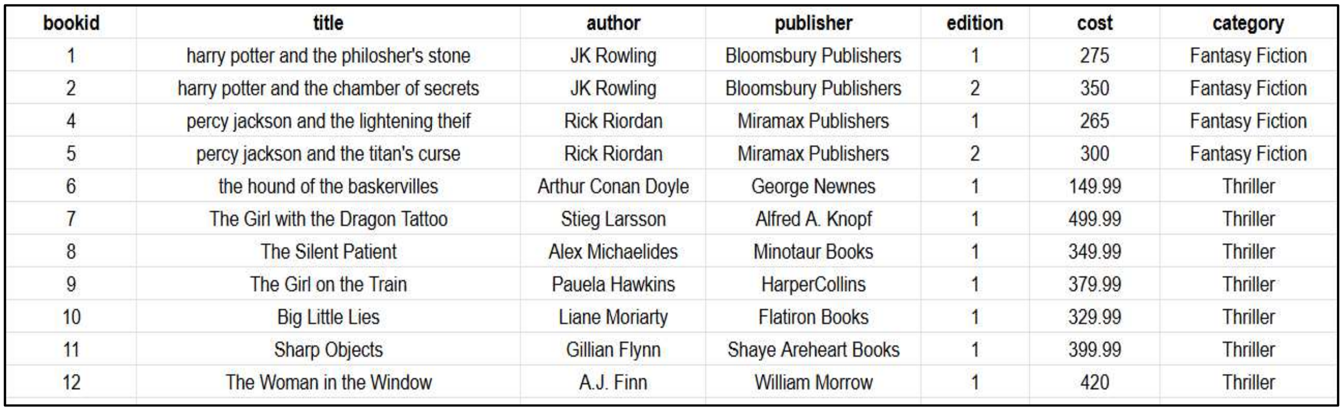
The names of all the members are stored and updated constantly in the file named 'members.csv'. It stores the member ID, name, email, phone no. and address.



mid	name	phone	email	address
1	Jane Foster	987-654-3210	janefoster@example.com	Bikaner
2	Samay Raina	555-123-4567	samayraina@example.com	Bhilwara
3	Tushar Pandey	555-987-6543	tusharop@example.com	Mumbai
4	Divyash Ostwal	9101714696	ilovetoread@gmail.com	Bangalore
6	Rajit Gupta	4364870489	rajitrocks@gmail.com	kota
7	Yashaswini Trivedi	4364870415	oldmcdonald@gmail.com	guwahati
8	Devansh Singh	9887016757	rcbwillwin@gmail.com	kota
9	Tony Stark	666-788-8909	iamironman@gmail.com	california

3. Books available in Library:

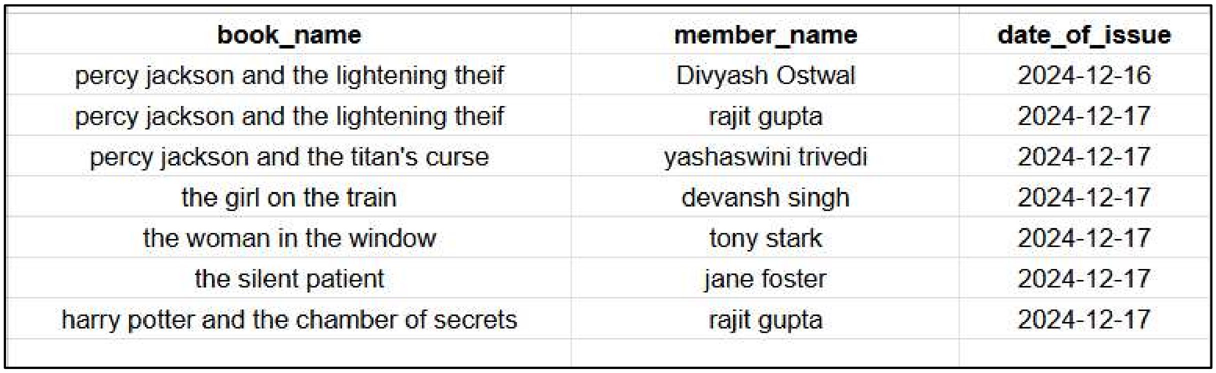
The data of books available in the library is stored in the file named ‘books.csv’. The file stores the book ID, title, author name, publisher name, edition, cost and the genre of the book



bookid	title	author	publisher	edition	cost	category
1	harry potter and the philosher's stone	JK Rowling	Bloomsbury Publishers	1	275	Fantasy Fiction
2	harry potter and the chamber of secrets	JK Rowling	Bloomsbury Publishers	2	350	Fantasy Fiction
4	percy jackson and the lightening theif	Rick Riordan	Miramax Publishers	1	265	Fantasy Fiction
5	percy jackson and the titan's curse	Rick Riordan	Miramax Publishers	2	300	Fantasy Fiction
6	the hound of the baskervilles	Arthur Conan Doyle	George Newnes	1	149.99	Thriller
7	The Girl with the Dragon Tattoo	Stieg Larsson	Alfred A. Knopf	1	499.99	Thriller
8	The Silent Patient	Alex Michaelides	Minotaur Books	1	349.99	Thriller
9	The Girl on the Train	Pauela Hawkins	HarperCollins	1	379.99	Thriller
10	Big Little Lies	Liane Moriarty	Flatiron Books	1	329.99	Thriller
11	Sharp Objects	Gillian Flynn	Shaye Areheart Books	1	399.99	Thriller
12	The Woman in the Window	A.J. Finn	William Morrow	1	420	Thriller

4. Book Issuing Records:

The records of issued books is kept in the file named ‘issuedBooks.csv’. This file contains the name of the member who issued the book, the name of the book issued and date of issuing.

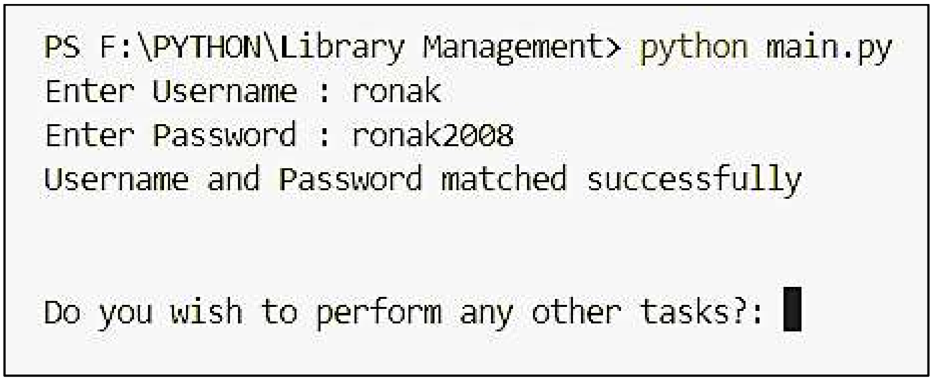


book_name	member_name	date_of_issue
percy jackson and the lightening theif	Divyash Ostwal	2024-12-16
percy jackson and the lightening theif	rajit gupta	2024-12-17
percy jackson and the titan's curse	yashaswini trivedi	2024-12-17
the girl on the train	devansh singh	2024-12-17
the woman in the window	tony stark	2024-12-17
the silent patient	jane foster	2024-12-17
harry potter and the chamber of secrets	rajit gupta	2024-12-17

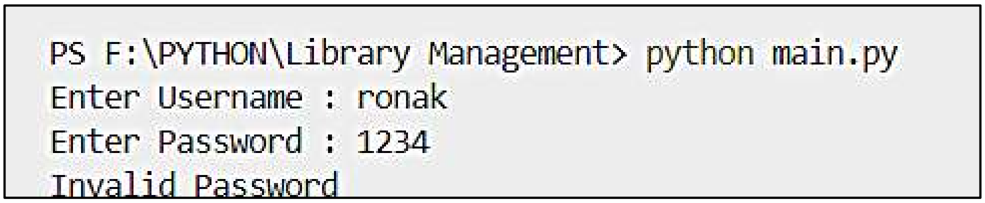
OUTPUTS

1. Login Process:

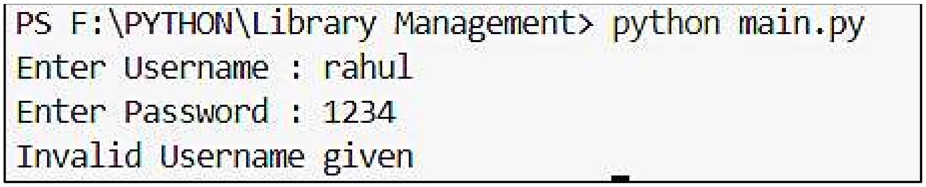
This is the only process where case of the text matters due to security reasons. If the entered password or username is incorrect then the message will be returned. But if the username and password are both correct then the prompt asking a prompt will be returned.



```
PS F:\PYTHON\Library Management> python main.py
Enter Username : ronak
Enter Password : ronak2008
Username and Password matched successfully


Do you wish to perform any other tasks?: 
```



```
PS F:\PYTHON\Library Management> python main.py
Enter Username : ronak
Enter Password : 1234
Invalid Password
```

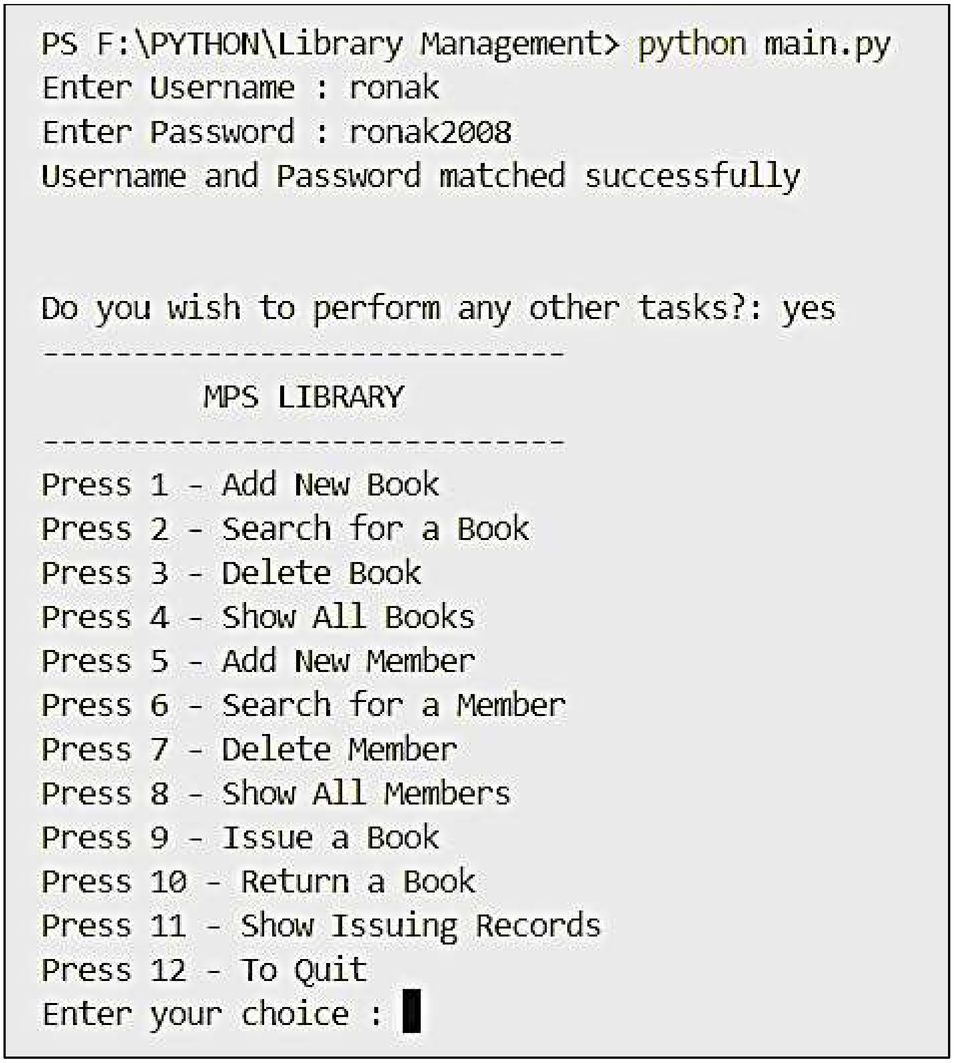


```
PS F:\PYTHON\Library Management> python main.py
Enter Username : rahul
Enter Password : 1234
Invalid Username given
```

2. Task Menu:

After the authentication and the prompt, the main menu will be displayed to select the task we need to perform.

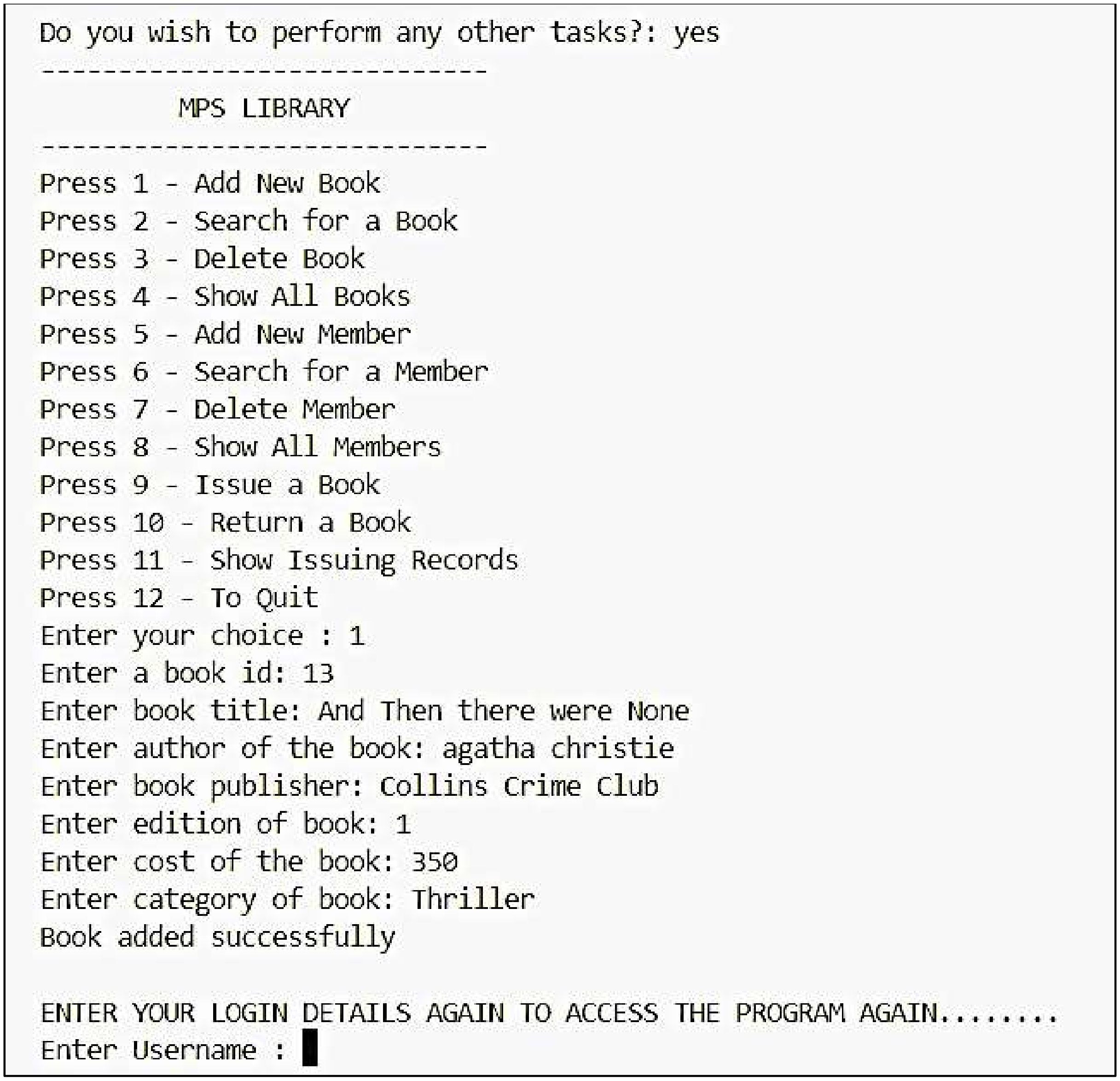
If the user enters any invalid value, the message for the same will be returned.



```
PS F:\PYTHON\Library Management> python main.py
Enter Username : ronak
Enter Password : ronak2008
Username and Password matched successfully


Do you wish to perform any other tasks?: yes
-----------------------------
         MPS LIBRARY
-----------------------------
Press 1 - Add New Book
Press 2 - Search for a Book
Press 3 - Delete Book
Press 4 - Show All Books
Press 5 - Add New Member
Press 6 - Search for a Member
Press 7 - Delete Member
Press 8 - Show All Members
Press 9 - Issue a Book
Press 10 - Return a Book
Press 11 - Show Issuing Records
Press 12 - To Quit
Enter your choice :
```

3. Adding a Book:

A book can be added as displayed. Also, there is no need to worry about case of the title as it is stored in lower case always using lower() function. This makes it easy to use the program as case is never an issue now.

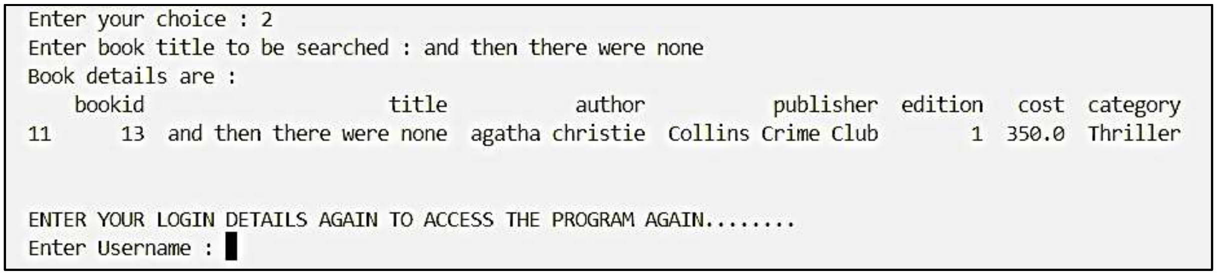


```
Do you wish to perform any other tasks?: yes
-----------------------------
         MPS LIBRARY
-----------------------------
Press 1 - Add New Book
Press 2 - Search for a Book
Press 3 - Delete Book
Press 4 - Show All Books
Press 5 - Add New Member
Press 6 - Search for a Member
Press 7 - Delete Member
Press 8 - Show All Members
Press 9 - Issue a Book
Press 10 - Return a Book
Press 11 - Show Issuing Records
Press 12 - To Quit
Enter your choice : 1
Enter a book id: 13
Enter book title: And Then there were None
Enter author of the book: agatha christie
Enter book publisher: Collins Crime Club
Enter edition of book: 1
Enter cost of the book: 350
Enter category of book: Thriller
Book added successfully

ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........
Enter Username : █
```

.

4. Searching for a Book:

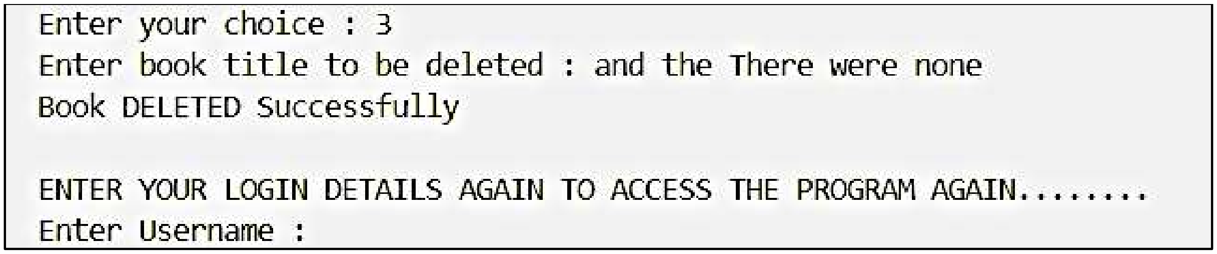
This process is also programmed such that the case of the book name you enter to find the details about the book again doesn’t matter. Just enter the name in any case and get results.



```
Enter your choice : 2
Enter book title to be searched : and then there were none
Book details are :
    bookid                     title          author         publisher  edition   cost  category
11      13  and then there were none  agatha christie  Collins Crime Club        1  350.0  Thriller


ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN.......
Enter Username :
```

5. Deleting a Book:

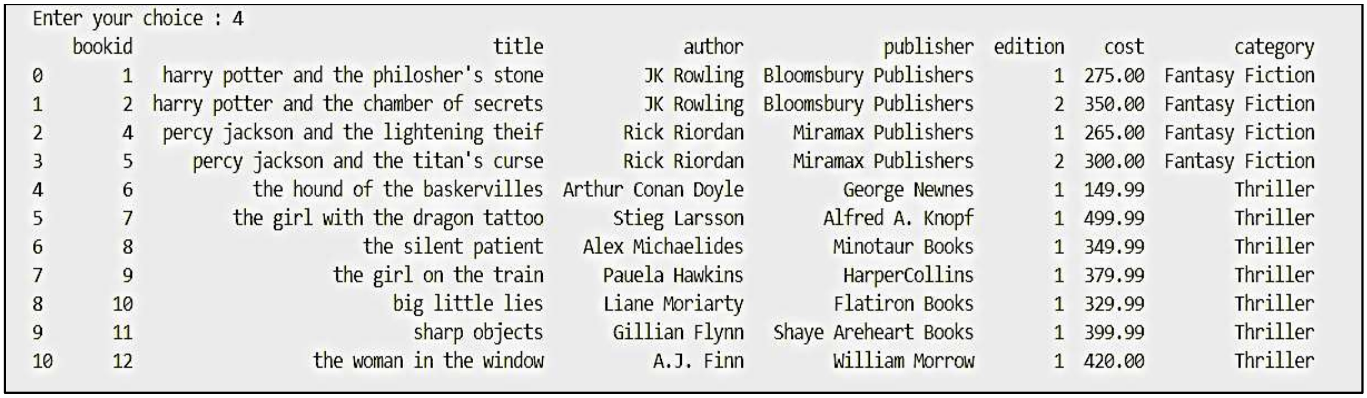
Again, you have to type in the title of the book you want to delete from the library. The case doesn’t matter.



```
Enter your choice : 3
Enter book title to be deleted : and the There were none
Book DELETED Successfully

ENTER YOUR LOGIN DETAILS AGAIN TO ACCESS THE PROGRAM AGAIN........
Enter Username :
```

6. Showing all books in library:

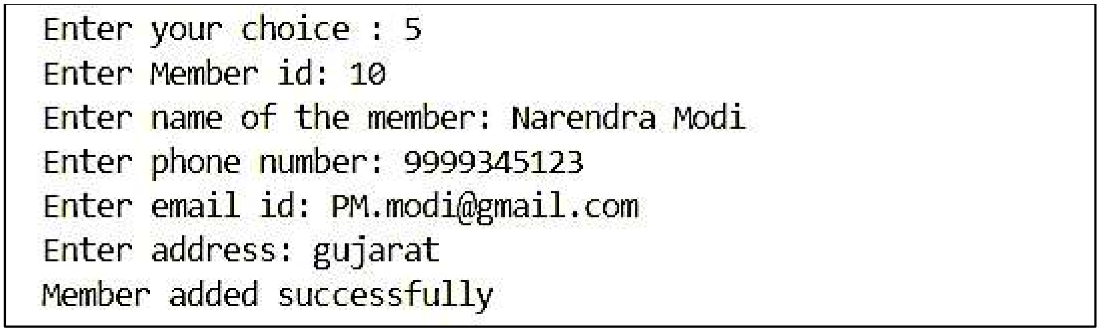
This feature shows all the books present in the library



```
Enter your choice : 4
    bookid                                 title              author              publisher  edition    cost         category
0        1    harry potter and the philosher's stone          JK Rowling  Bloomsbury Publishers        1  275.00  Fantasy Fiction
1        2   harry potter and the chamber of secrets          JK Rowling  Bloomsbury Publishers        2  350.00  Fantasy Fiction
2        4    percy jackson and the lightening theif        Rick Riordan     Miramax Publishers        1  265.00  Fantasy Fiction
3        5       percy jackson and the titan's curse        Rick Riordan     Miramax Publishers        2  300.00  Fantasy Fiction
4        6            the hound of the baskervilles  Arthur Conan Doyle          George Newnes        1  149.99         Thriller
5        7            the girl with the dragon tattoo       Stieg Larsson         Alfred A. Knopf        1  499.99         Thriller
6        8                         the silent patient    Alex Michaelides          Minotaur Books        1  349.99         Thriller
7        9                       the girl on the train      Pauela Hawkins           HarperCollins        1  379.99         Thriller
8       10                            big little lies       Liane Moriarty          Flatiron Books        1  329.99         Thriller
9       11                              sharp objects       Gillian Flynn    Shaye Areheart Books        1  399.99         Thriller
10      12                      the woman in the window           A.J. Finn          William Morrow        1  420.00         Thriller
```

7. Adding a Member:

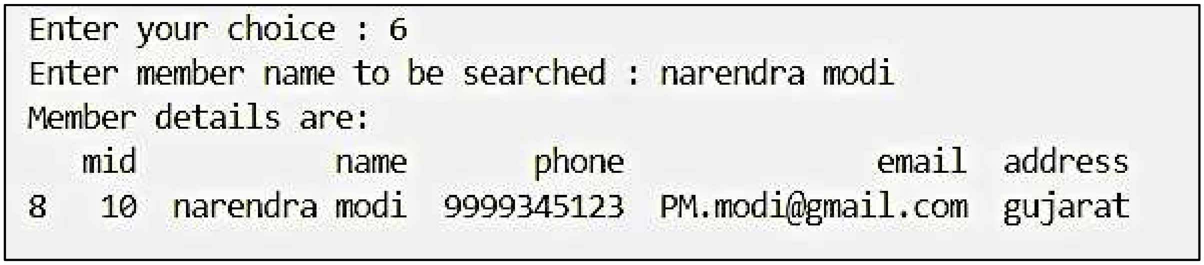
A member can be added as displayed. Also, there is no need to worry about case of the title as it is stored in lower case always using lower() function. This makes it easy to use the program as case is never an issue now.



```
Enter your choice : 5
Enter Member id: 10
Enter name of the member: Narendra Modi
Enter phone number: 9999345123
Enter email id: PM.modi@gmail.com
Enter address: gujarat
Member added successfully
```

8. Searching for a Member:

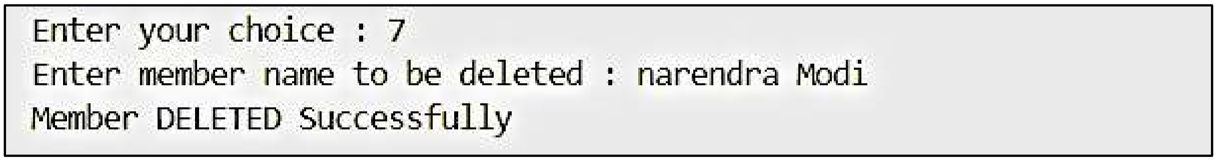
Just enter the name of the member you want to search for in any case



```
Enter your choice : 6
Enter member name to be searched : narendra modi
Member details are:
   mid            name       phone              email  address
8   10  narendra modi  9999345123  PM.modi@gmail.com  gujarat
```

9. Deleting a Member:

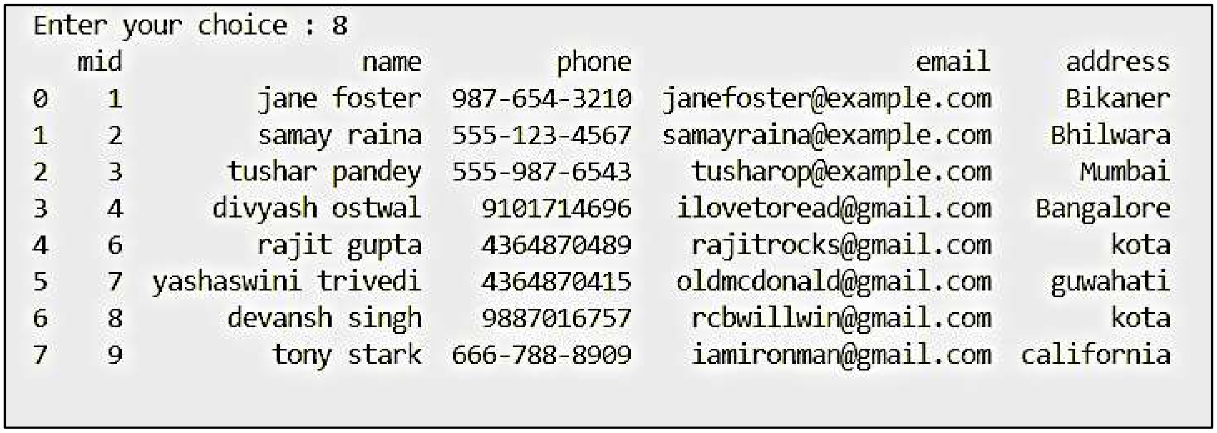
Just enter the name of the member you want to remove.



```
Enter your choice : 7
Enter member name to be deleted : narendra Modi
Member DELETED Successfully
```

10. Showing all Members:

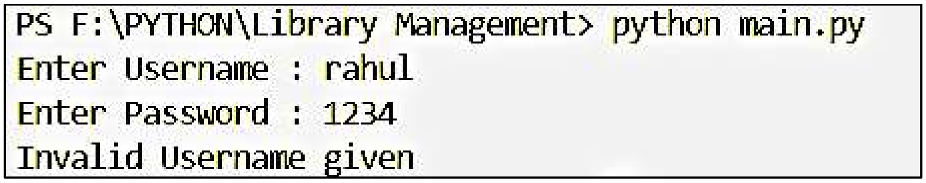
.It is output when we ask the program to show all the members.



```
Enter your choice : 8
   mid                name         phone                    email     address
0    1         jane foster  987-654-3210  janefoster@example.com     Bikaner
1    2         samay raina  555-123-4567  samayraina@example.com    Bhilwara
2    3       tushar pandey  555-987-6543    tusharop@example.com      Mumbai
3    4      divyash ostwal    9101714696   ilovetoread@gmail.com   Bangalore
4    6         rajit gupta    4364870489    rajitrocks@gmail.com        kota
5    7  yashaswini trivedi    4364870415   oldmcdonald@gmail.com    guwahati
6    8        devansh singh    9887016757    rcbwillwin@gmail.com        kota
7    9          tony stark  666-788-8909    iamironman@gmail.com  california
```

11. Issuing a Book:

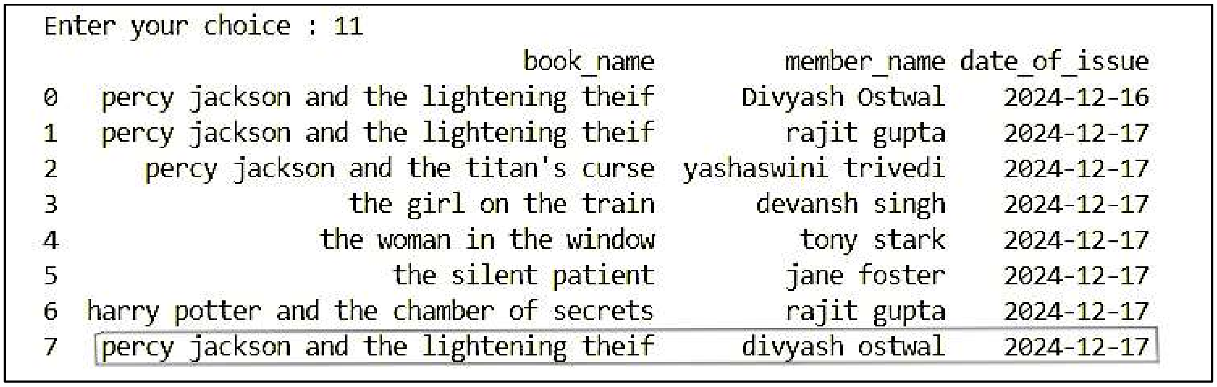
Enter book name, member name and you are good to go. The date will be automatically filled.



```
PS F:\PYTHON\Library Management> python main.py
Enter Username : rahul
Enter Password : 1234
Invalid Username given
```

12. Issue Records:

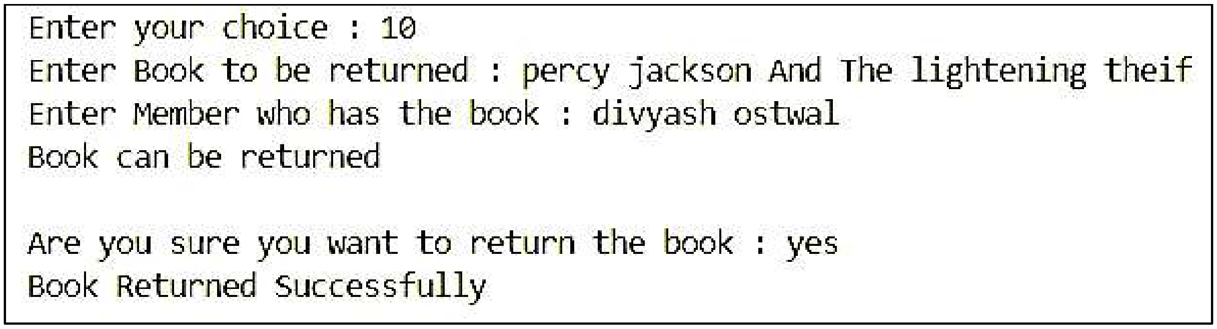
This simply shows records of issued books.



```
Enter your choice : 11
                                   book_name         member_name date_of_issue
0   percy jackson and the lightening theif      Divyash Ostwal    2024-12-16
1   percy jackson and the lightening theif         rajit gupta    2024-12-17
2      percy jackson and the titan's curse  yashaswini trivedi    2024-12-17
3                   the girl on the train        devansh singh    2024-12-17
4                 the woman in the window          tony stark    2024-12-17
5                      the silent patient         jane foster    2024-12-17
6  harry potter and the chamber of secrets         rajit gupta    2024-12-17
7   percy jackson and the lightening theif      divyash ostwal    2024-12-17
```

13. Returning a Book:

After enter the book name and the name of the person who is returning the book, the program will confirm the return.



```
Enter your choice : 10
Enter Book to be returned : percy jackson And The lightening theif
Enter Member who has the book : divyash ostwal
Book can be returned

Are you sure you want to return the book : yes
Book Returned Successfully
```

CONCLUSION

This Library Management System effectively automates several library functions, making it easier to manage books, members, and transactions. Its use of CSV files makes it simple to set up and maintain, even without a complex database system.

The project is an excellent starting point for building more advanced library management systems, where features like automatic overdue book management, detailed reports, and integration with databases (e.g., MySQL or SQLite) can be added to improve scalability and functionality.

By implementing this project, users are provided with a user-friendly and functional tool to enhance the management of a library, saving time and effort for librarians and staff, while offering better access to library resources for members.

BIBLIOGRAPHY

The following resources were used in the development of the Library Management System:

- **Python Documentation:**
  Provided essential information on Python features, functions, and syntax.
  URL: *https://docs.python.org/*
- **Pandas Documentation:**
  Helped with manipulating and processing data using Data Frames for CSV file handling.

  URL: *https://pandas.pydata.org/pandas-docs/stable/*
- **Stack Overflow:**
  Used for solving coding issues and troubleshooting errors.

  URL: *https://stackoverflow.com/*
- **Google.com:**
  Searched for tutorials and resources related to Python and file handling.

  URL: *https://www.google.com/*
- **Class 12 Informatics Practices NCERT Textbook**
  Provided foundational knowledge on Python programming and file handling.