# AIM: Create a class for library and implement its functions

Library Management Systems are used to manage information about contents in a library. They are used to manage information relating to books, their names,whether they have been issued or not. A library management system is used to store and manage all this information. Library management system is a project which aims in developing a computerized system to maintain all the daily work of library.Our library system consist of four tools

a)Display books in the library ,this tool is used to the display the books available at the library .Now the user can make a choice and lend the book he want

b)lend the book,this tool is used while lending the book.When a book is lent from the library ,it is removed from the list of books available at the library

c)Add or return the book is used while returning or adding a new book to the libaray .when thw book is added or returned it is added to list consisting of books available at the library'

d)Exit is used to stop the running of program

## SOFTWARE USED

IDLE Python 3.11.1 is the software used to Create a class for library and implement its functions

IDLE is Python’s Integrated Development and Learning Environment. IDLE has the following features ,coded in 100% pure Python, using the tkinter GUI toolkit cross-platform: works mostly the same on Windows, Unix, and macOS Python shell window (interactive interpreter) with colorizing of code input, output, and error messages

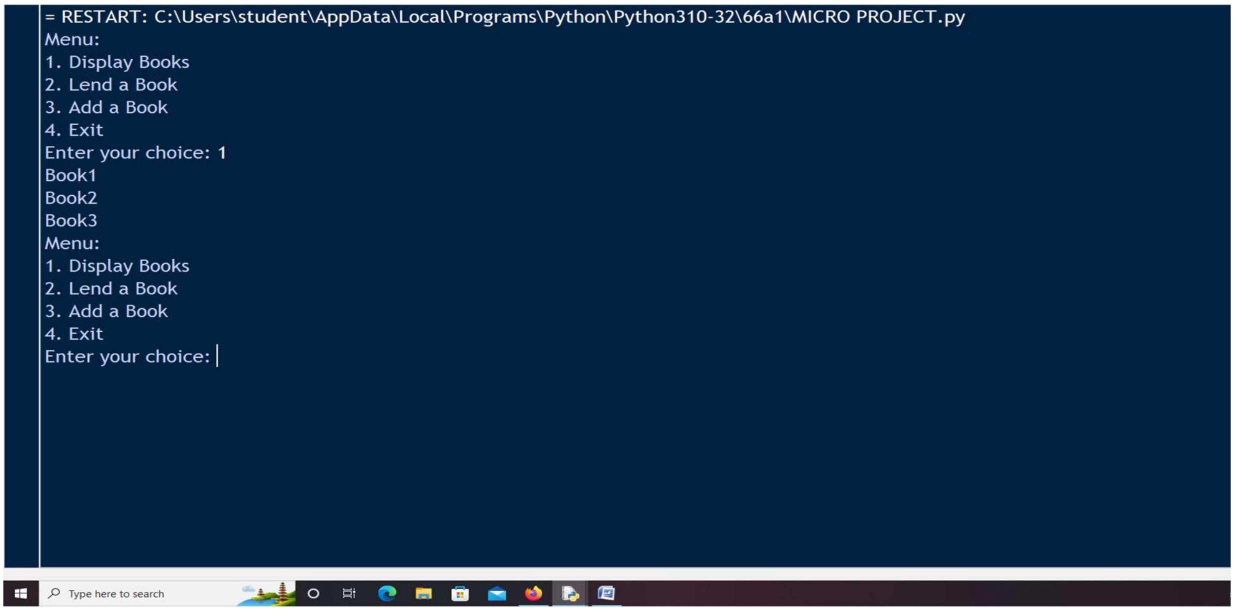
## SOURCE CODE

```
class Library:
  def __init__(self, books):
    self.books = books

def display_books(self):
    for book in self.books:
      print(book)

  def lend_book(self, book):
    if book in self.books:
      self.books.remove(book)
      return print("Book lent successfully")
    else:
      return print("Book not found")

  def add_book(self, book):
    self.books.append(book)
    return print("Book added successfully")
```

```
# create a Library object with some books
my_library = Library(["Book1","Book2","Book3"])
def menu():
  # dictionary of options
  options = {
    1: my_library.display_books,
    2: lambda: my_library.lend_book(input("Enter the name of the book you want to lend: ")),
    3: lambda: my_library.add_book(input("Enter the name of the book you want to add: ")),
    4: exit
    }

  # display menu
  print("Menu:")
  print("1. Display Books ")
  print("2. Lend a Book")
  print("3. Add a Book")
  print("4. Exit")

# read input from the user
  choice = int(input("Enter your choice: "))
  # execute the chosen option
  if choice in options:
    options[choice]()
  else:
    print("Invalid option")

# run the menu
while True:
  menu()
```

## OUTPUT:

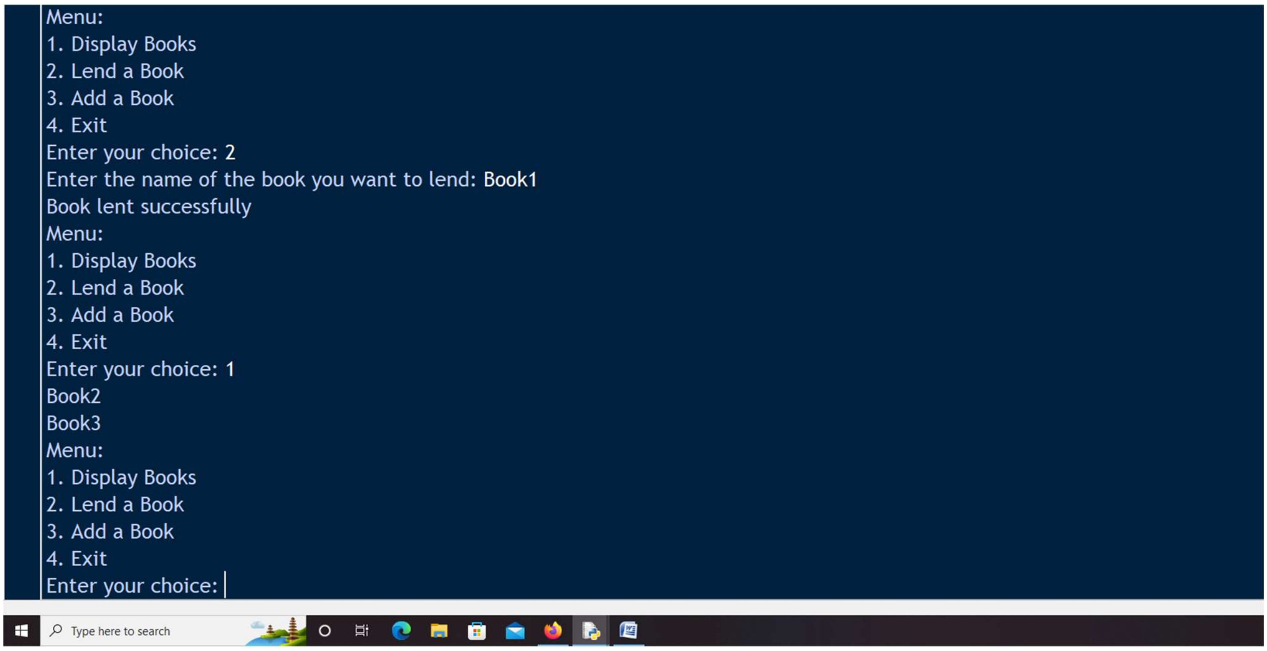
1)Display books



```
= RESTART: C:\Users\student\AppData\Local\Programs\Python\Python310-32\66a1\MICRO PROJECT.py
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice: 1
Book1
Book2
Book3
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice:
```

Here the user wanted to display the books present in the library so he chose option 1.The program returned all the available books at the library

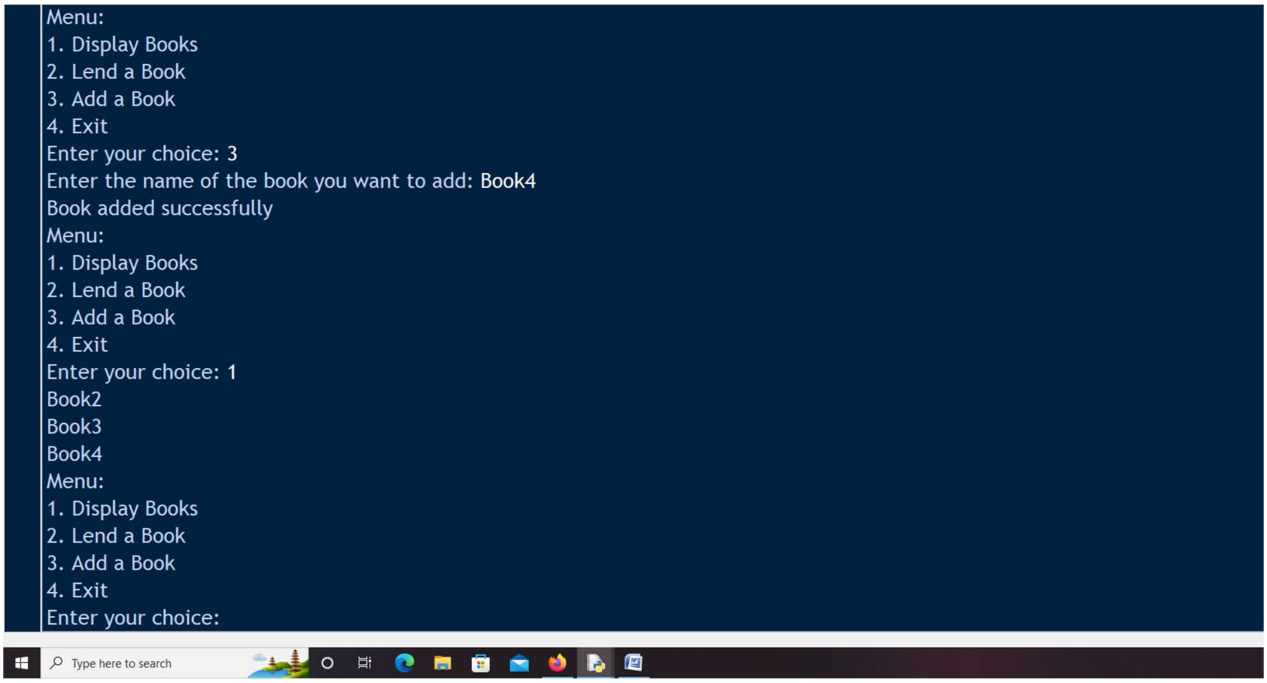
## 2)Lend a Book



```
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice: 2
Enter the name of the book you want to lend: Book1
Book lent successfully
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice: 1
Book2
Book3
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice:
```

To lend a book option 2 should is selected.Then the program asks the user to type the name of the book he want to lend.The program returns “Book lent succesfully” if the book is available else it returns “Book not found”

## 3)Add a Book



```
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice: 3
Enter the name of the book you want to add: Book4
Book added successfully
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice: 1
Book2
Book3
Book4
Menu:
1. Display Books
2. Lend a Book
3. Add a Book
4. Exit
Enter your choice:
```

To return or add a book to the library the user has to select the option 3.After entering the option 3 the program asks the user to enter the name of the book he want to add .After entering the name of the book it returns “Book added successfully”

## Algorithm

This is a simple library management system that allows you to create a Library object with a list of books, display the list of books in the library, lend a book from the library, and add a book to the library. The menu() function provides a simple interface for interacting with the library. You can choose from one of the four options by entering the corresponding number. If you choose option 1, the list of books in the library will be displayed. If you choose option 2, you will be prompted to enter the name of the book you want to lend, and the book will be removed from the library if it is found. If you choose option 3, you will be prompted to enter the name of the book you want to add, and the book will be added to the library. If you choose option 4, the program will exit. The while True loop at the end allows the menu to be displayed repeatedly until you choose to exit.

Here is a breakdown of what is happening:

1. The Library class is defined with a constructor __init__ which initializes the books attribute of the object with the value of the books parameter passed to it.

2. The display_books method prints the list of books in the library.

3. The lend_book method checks if a book is present in the library's list of books. If it is present, it removes the book from the list and prints a message indicating that the book has been lent successfully. If the book is not present, it prints a message indicating that the book was not found.

4. The add_book method adds a book to the library's list of books and prints a message indicating that the book was added successfully.

5. The menu function displays a menu to the user with options to display books, lend a book, add a book, or exit the program.

6. The user is prompted to enter their choice and the corresponding action is taken based on the chosen option.

7. The program runs in a loop, displaying the menu and prompting the user for input until the user chooses to exit the program.