Explanation

1. The ‘ *init* ‘ method creates a window with a title, dimensions, and a background color. It also initializes lists for books and a lend list.
2. Labels and entry boxes for username and password are created, along with Login and Register buttons.
3. The *‘login’* method checks the entered username and password against a list of librarians. If the credentials are correct, the login interface is destroyed and the *‘library_management_screen’* method is called.
4. The *register* method adds the entered username and password to the list of librarians.
5. The *‘library_management_screen’* method creates labels, entry boxes, and buttons for adding, removing, issuing, and viewing books.
6. The *‘add_book’* method adds the entered book to the list of books and displays a success message.
7. The *‘remove_book’* method removes the entered book from the list of books if it exists and displays a success message, or an error message if it doesn’t exist.
8. The *‘issue_book’* method moves the entered book from the list of books to the lend list if it exists and displays a success message, or an error message if it doesn’t exist.
9. The *‘view_books’* method displays a message box with a list of all the books in the library.
10. The if *name == “ main “:* block creates a *‘Tk’* object, initializes an instance of

‘LibraryManagement’, and starts the main event loop to display the window.

Some other information about Project

This is a library management system program. There is not very complex coding It is fully based on Python programming. We import some modules for GUI (Graphical User Interface).

There are some advantages of this program: –

- It can store Unlimited number of books.
- It stores data in memory of computer. (What can we add in it for improve)
- We can borrow books from this program.
- We can also remove books form it.
- We have to create an account for accessing books.
- There is also an option for Login

We can improve this by adding these functions: –

- We can add a database in this program for storing Books.
- We can add an option for uploading Documents, like pdf etc.
- We can improve its Login and Create a new account feature.
- So, we can improve this program by adding these features.

Source Code

```
import tkinter as tk
from tkinter import messagebox

class LibraryManagement:
    def __init__(self, master):
        self.master = master
        self.master.title("Library Management System")
        self.master.geometry("400x400")
        self.master.config(bg='#708090')

        self.books = []
        self.lend_list = []

        # Labels
        self.login_label = tk.Label(self.master, text="Library
Management System", font=("Helvetica", 16), bg='#708090',
fg='white')
        self.login_label.pack()
        self.username_label = tk.Label(self.master, text="Username",
font=("Helvetica", 12), bg='#708090', fg='white')
        self.username_label.pack()
        self.username_entry = tk.Entry(self.master,
font=("Helvetica", 12))
        self.username_entry.pack()
        self.password_label = tk.Label(self.master, text="Password",
font=("Helvetica", 12), bg='#708090', fg='white')
        self.password_label.pack()
        self.password_entry = tk.Entry(self.master,
font=("Helvetica", 12), show="*")
        self.password_entry.pack()

        # Login
        self.login_button = tk.Button(self.master, text="Login",
command=self.login, font=("Helvetica", 12))
```

```
        self.login_button.pack()

        # Register
        self.register_button = tk.Button(self.master,
text="Register", command=self.register, font=("Helvetica", 12))
        self.register_button.pack()

        self.username = ""
        self.password = ""
        self.librarians = []

    def login(self):
        self.username = self.username_entry.get()
        self.password = self.password_entry.get()
        for librarian in self.librarians:
            if self.username == librarian[0] and self.password ==
librarian[1]:
                self.username_entry.delete(0, tk.END)
                self.password_entry.delete(0, tk.END)
                self.login_label.destroy()
                self.username_label.destroy()
                self.username_entry.destroy()
                self.password_label.destroy()
                self.password_entry.destroy()
                self.login_button.destroy()
                self.register_button.destroy()
                self.library_management_screen()
                return
        messagebox.showerror("Error", "Invalid username or
password")

    def register(self):
        self.username = self.username_entry.get()
        self.password = self.password_entry.get()
        self.librarians.append([self.username, self.password])
        self.username_entry.delete(0, tk.END)
        self.password_entry.delete(0, tk.END)
    def library_management_screen(self):
        self.add_book_label = tk.Label(self.master, text="Add Book",
font=("Helvetica", 16), bg='#708090', fg='white')
        self.add_book_label.pack()
        self.add_book_entry = tk.Entry(self.master,
font=("Helvetica", 12))
        self.add_book_entry.pack()
        self.add_book_button = tk.Button(self.master, text="Add
Book", command=self.add_book, font=("Helvetica", 12))
        self.add_book_button.pack()
        self.remove_book_label = tk.Label(self.master, text="Remove
Book", font=("Helvetica", 16), bg='#708090', fg='white')
```

```
        self.remove_book_label.pack()
        self.remove_book_entry = tk.Entry(self.master,
font=("Helvetica", 12))
        self.remove_book_entry.pack()
        self.remove_book_button = tk.Button(self.master,
text="Remove Book", command=self.remove_book, font=("Helvetica",
12))
        self.remove_book_button.pack()
        self.issue_book_label = tk.Label(self.master, text="Issue
Book", font=("Helvetica", 16), bg='#708090', fg='white')
        self.issue_book_label.pack()
        self.issue_book_entry = tk.Entry(self.master,
font=("Helvetica", 12))
        self.issue_book_entry.pack()
        self.issue_book_button = tk.Button(self.master, text="Issue
Book", command=self.issue_book, font=("Helvetica", 12))
        self.issue_book_button.pack()
        self.view_books_button = tk.Button(self.master, text="View
Books", command=self.view_books, font=("Helvetica", 12))
        self.view_books_button.pack()

    def add_book(self):
        book = self.add_book_entry.get()
        self.books.append(book)
        messagebox.showinfo("Success", "Book added successfully")
        self.add_book_entry.delete(0, tk.END)

    def remove_book(self):
        book = self.remove_book_entry.get()
        if book in self.books:
            self.books.remove(book)
            messagebox.showinfo("Success", "Book removed
successfully")
        else:
            messagebox.showerror("Error", "Book not found")
        self.remove_book_entry.delete(0, tk.END)

    def issue_book(self):
        book = self.issue_book_entry.get()
        if book in self.books:
            self.lend_list.append(book)
            self.books.remove(book)
            messagebox.showinfo("Success", "Book issued
successfully")
        else:
            messagebox.showerror("Error", "Book not found")
        self.issue_book_entry.delete(0, tk.END)

    def view_books(self):
```

```
        message = "\n".join(self.books)
        messagebox.showinfo("Books", message)

if __name__ == "__main__":
    root = tk.Tk()
    app = LibraryManagement(root)
    root.mainloop()
```

Output: