UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Graphical User Interface Project Report

Student Name: Anshita Goel **UID: 24MCI10012**
Branch: MCA(AIML) **Section/Group: 1A**
Semester: 1 **Date of Performance: 28-10-2024**
Subject Name: Python Programming Lab **Subject Code: 24CAH-606**
Github Link: https://github.com/anshita2003/mini-python.git

1. **Introduction:** The Sales Analysis Dashboard is a graphical user interface (GUI) application developed using Python's Tkinter library. The system allows users to manage a library's book inventory by adding, updating, deleting, issuing, and returning books. The interface includes input fields for book details and buttons for various actions, along with a list display showing the current state of each book.

2. **Objectives:** The objective of this code is to create a basic Library Management System with the following functionality:

   - **Add New Books**: Allows users to add new books to the library by entering the title, author, and ISBN.
   - **Update Book Details**: Users can update the details (title, author, ISBN) of a selected book.
   - **Delete Books**: Enables users to delete a book from the library's inventory.
   - **Issue and Return Books**: Users can mark books as issued or returned, helping track availability.
   - **Display Inventory**: Displays a list of all books in the library, showing their current status (issued or available).
   - **User-Friendly Interface**: Provides a simple graphical interface using tkinter for easy management of library books.

3. **Technical Overview:**

Tools and Libraries Used -

- **Python**: The programming language used to develop the application.
- **tkinter**: Python's standard GUI library, used to create the graphical user interface (GUI) for the application.
- **messagebox (from tkinter)**: Provides a set of pop-up message boxes for notifications, warnings, and confirmations within the app.

Installation Requirement:

To run the application, the following package should be installed:

- Tkinter (comes with standard Python installations)

Application Structure:

The application consists of the following components:

- Main Window
- Labels and Entry fields
- Buttons
- Book List Display
- Message Box Notifications

4. Code:

```
import tkinter as tk
from tkinter import messagebox

class Book:
  def __init__(self, title, author, isbn):
    self.title = title
```

```
        self.author = author


        self.isbn = isbn
        self.issued = False  # Tracks if the book is issued


class Library:
    def __init__(self, root):
        self.root = root
        self.root.title("Library Management System")
        self.root.geometry("500x600")

        # Book inventory
        self.books = []

        # Create GUI elements
        self.create_widgets()

    def create_widgets(self):
        """Sets up the GUI layout"""
        tk.Label(self.root, text="Library Management System", font=("Helvetica", 16, "bold")).pack(pady=10)

        # Frames for organization
        book_frame = tk.Frame(self.root)
        book_frame.pack(pady=10)

        # Book entry fields
        tk.Label(book_frame, text="Title:", font=("Helvetica", 12)).grid(row=0, column=0, padx=5, pady=5)
        self.title_entry = tk.Entry(book_frame, font=("Helvetica", 12))
```

```
    self.title_entry.grid(row=0, column=1, padx=5, pady=5)

    tk.Label(book_frame, text="Author:", font=("Helvetica", 12)).grid(row=1, column=0, padx=5, pady=5)
    self.author_entry = tk.Entry(book_frame, font=("Helvetica", 12))
    self.author_entry.grid(row=1, column=1, padx=5, pady=5)

    tk.Label(book_frame, text="ISBN:", font=("Helvetica", 12)).grid(row=2, column=0, padx=5, pady=5)
    self.isbn_entry = tk.Entry(book_frame, font=("Helvetica", 12))
    self.isbn_entry.grid(row=2, column=1, padx=5, pady=5)

    # Buttons for book management
    tk.Button(self.root, text="Add Book", command=self.add_book, font=("Helvetica",   12)).pack(pady=5)
    tk.Button(self.root, text="Update Book", command=self.update_book, font=("Helvetica",
    12)).pack(pady=5)
    tk.Button(self.root, text="Delete Book", command=self.delete_book, font=("Helvetica",
    12)).pack(pady=5)
    tk.Button(self.root, text="Issue Book", command=self.issue_book, font=("Helvetica", 12)).pack(pady=5)
    tk.Button(self.root, text="Return Book", command=self.return_book, font=("Helvetica",
    12)).pack(pady=5)

    # Display area
    self.book_list = tk.Listbox(self.root, width=60, height=10, font=("Helvetica", 10))
    self.book_list.pack(pady=10)

def add_book(self):
    """Adds a new book to the library"""
    title = self.title_entry.get()
    author = self.author_entry.get()
     isbn = self.isbn_entry.get()
```

```
    # Validation
    if not title or not author or not isbn:
        messagebox.showwarning("Input Error", "Please fill in all fields")
        return

    # Create a new book and add to inventory
    new_book = Book(title, author, isbn)
    self.books.append(new_book)

    self.update_book_list()
    messagebox.showinfo("Book Added", f"{title} by {author} added to the library.")

    # Clear entries
    self.clear_entries()

def update_book(self):
    """Updates the details of a selected book"""
    selection = self.book_list.curselection()
    if not selection:
        messagebox.showwarning("Selection Error", "Please select a book to update.")
        return

    selected_index = selection[0]
    book = self.books[selected_index]

    # Get updated information
    title = self.title_entry.get()
    author = self.author_entry.get()
```

```
    isbn = self.isbn_entry.get()

    if not title or not author or not isbn:
        messagebox.showwarning("Input Error", "Please fill in all fields")
        return

    # Update the book's details
    book.title = title
    book.author = author
    book.isbn = isbn
    self.update_book_list()

    messagebox.showinfo("Book Updated", f"{book.title} has been updated.")
    self.clear_entries()

def delete_book(self):
    """Deletes a selected book from the library"""
    selection = self.book_list.curselection()
    if not selection:
        messagebox.showwarning("Selection Error", "Please select a book to delete.")
        return

    selected_index = selection[0]
    book = self.books[selected_index]

     # Confirm deletion
    confirm = messagebox.askyesno("Confirm Delete", f"Are you sure you want to delete {book.title}?")
    if confirm:
```

```
        del self.books[selected_index]
        self.update_book_list()
        messagebox.showinfo("Book Deleted", f"{book.title} has been deleted.")
        self.clear_entries()

    def issue_book(self):
        """Issues a book based on the selected item in the list"""
        selection = self.book_list.curselection()
        if not selection:
            messagebox.showwarning("Selection Error", "Please select a book to issue.")
            return

        selected_index = selection[0]
        book = self.books[selected_index]

        if book.issued:
            messagebox.showwarning("Issue Error", "This book is already issued.")
        else:
            book.issued = True
            self.update_book_list()
            messagebox.showinfo("Book Issued", f"{book.title} has been issued.")

    def return_book(self):
        """Returns an issued book back to the library"""
        selection = self.book_list.curselection()
        if not selection:
            messagebox.showwarning("Selection Error", "Please select a book to return.")
            return
```

```
        selected_index = selection[0]
        book = self.books[selected_index]

        if not book.issued:
            messagebox.showwarning("Return Error", "This book is not issued.")
        else:
            book.issued = False
            self.update_book_list()
            messagebox.showinfo("Book Returned", f"{book.title} has been returned.")

    def update_book_list(self):
        """Updates the listbox display with the latest book information"""
        self.book_list.delete(0, tk.END)
        for book in self.books:
            status = "Issued" if book.issued else "Available"
            self.book_list.insert(tk.END, f"{book.title} by {book.author} (ISBN: {book.isbn}) - {status}")

    def clear_entries(self):
        """Clears the entry fields after adding, updating, or deleting a book"""
        self.title_entry.delete(0, tk.END)
        self.author_entry.delete(0, tk.END)
        self.isbn_entry.delete(0, tk.END)

# Run the application
root = tk.Tk()
app = Library(root)
root.mainloop()
```

5. Code Breakdown:

a) Libraries and Imports –

- tkinter: The tkinter library is used to create a graphical user interface (GUI). tk provides window and widget functionalities.
- messagebox: A module within tkinter to create pop-up message dialogs.

b) Book Class - Book class is used to model a book in the library.

c) Library Class - This class handles the main library system and GUI components

d) Methods -

- create_widgets : Defines and arranges the graphical elements for the Library Management System.
- add_book : Collects title, author, and ISBN inputs to create a new book.
- update_book : Allows editing the details of an existing book in the library.
- delete_book : Removes a selected book from the inventory.
- issue_book : Marks a book as issued to track its availability.
- return_book : Marks an issued book as available.
- update_book_list : Refreshes the display in book_list to reflect the current state of self.books.
- clear_entries : Clears the title, author, and ISBN entry fields.

e) Application Initialization and Execution -

Initialization:

- Creates a Tk instance as the main window.
- Initializes Library with root, setting up the interface and starting the main loop.

6. User Instructions:

This Library Management System allows you to add, update, delete, issue, and return books through a graphical interface.

a) Adding a Book:

- Enter the book's Title, Author, and ISBN in the corresponding input fields.
- Click "Add Book" to add the book to the library inventory.
- If all fields are correctly filled, a message will confirm the addition, and the book will appear in the list as "Available."

b) Updating a Book:

- Select the book you want to update from the list.
- Edit the Title, Author, and/or ISBN fields with new information.
- Click "Update Book" to save the changes.
- If successful, a message will confirm the update, and the updated information will display in the list.

c) Deleting a Book:

- Select the book you want to delete from the list.
- Click "Delete Book".
- A confirmation dialog will appear. Click Yes to confirm or No to cancel.
- If confirmed, the book will be removed from the list, and a message will confirm the deletion.

d) Issuing a Book:

- Select the book you want to issue from the list.
- Click "Issue Book".
- If the book is not already issued, it will be marked as "Issued" in the list. A confirmation message will notify you that the book has been issued.
- If the book is already issued, a warning will appear.

e) Returning a Book:

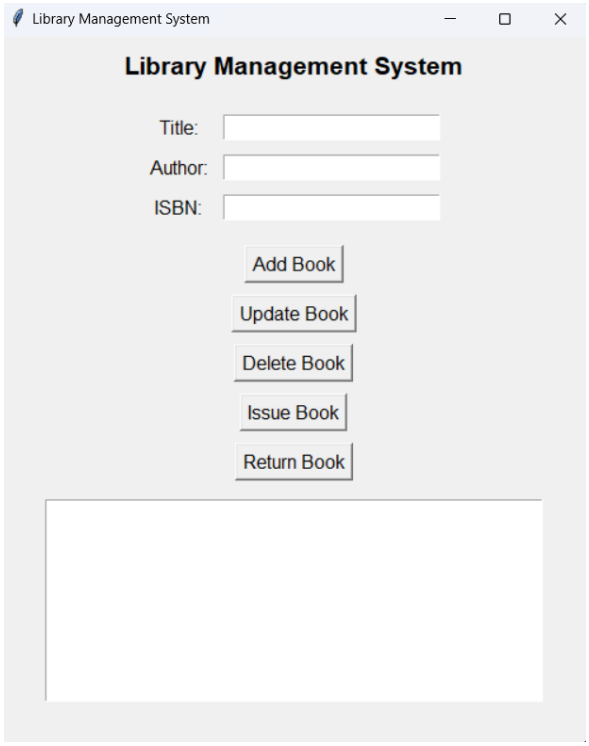
- Select the book you want to return from the list.
- Click "Return Book".
- If the book was issued, it will be marked as "Available" in the list, and a confirmation message will notify you of the return.
- If the book is not issued, a warning will appear.

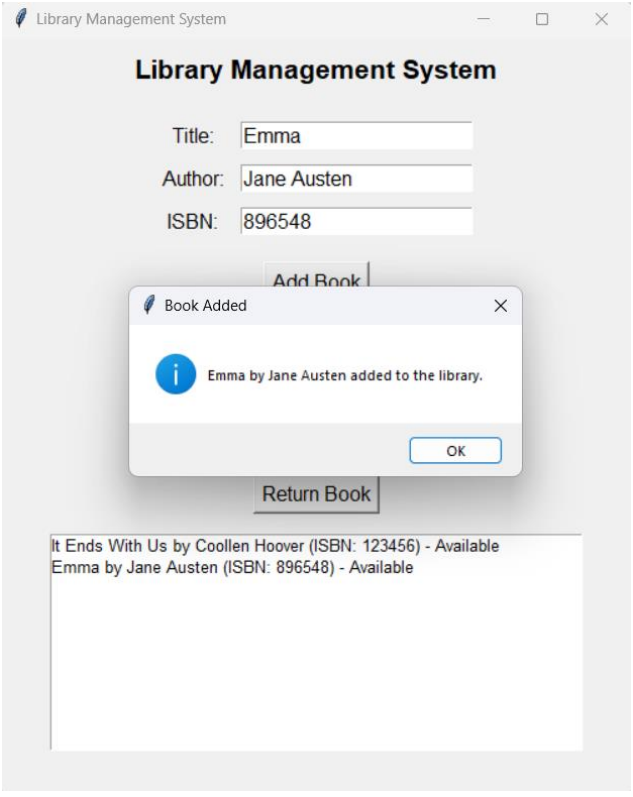
f) Clearing Entries:

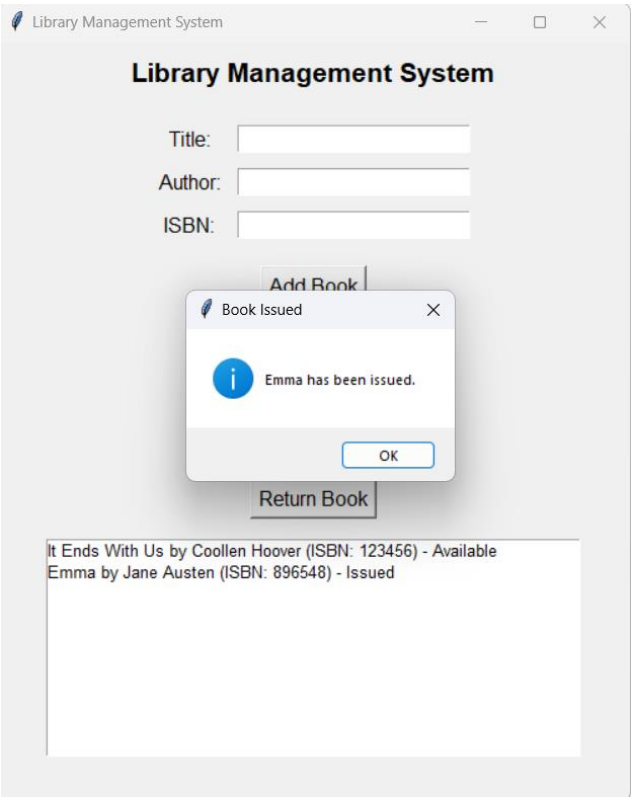
- After any action (Add, Update, or Delete), the input fields are automatically cleared.
- If needed, you can manually clear them by removing text in each input field.

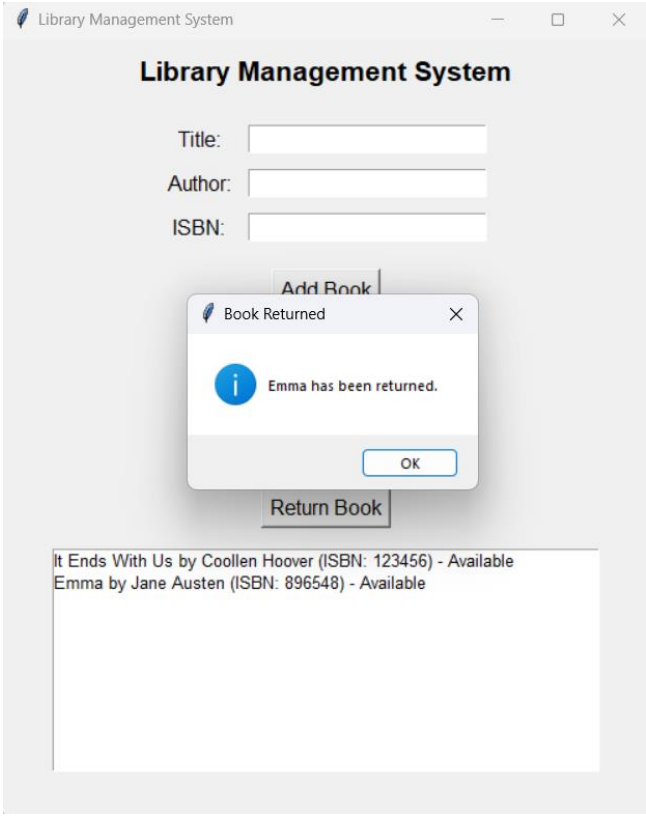
7. Conclusion:

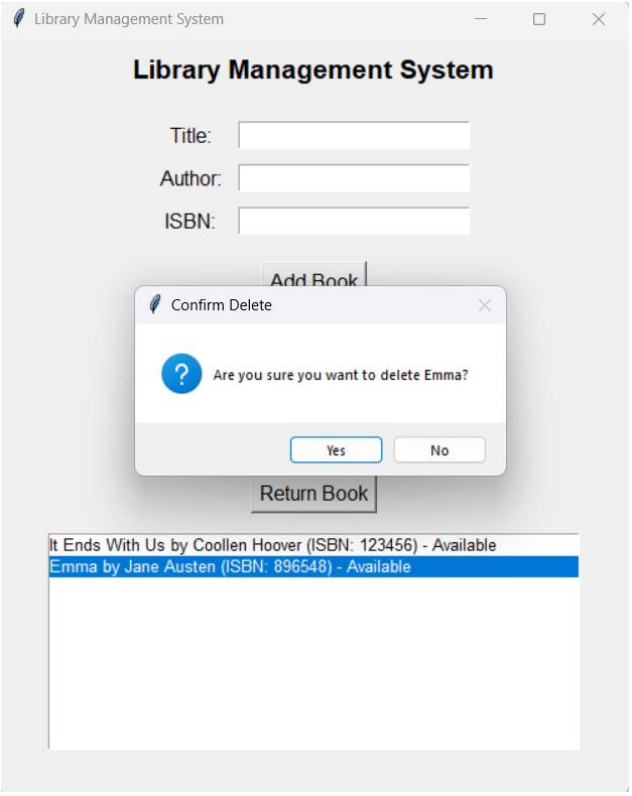
This Library Management System provides a comprehensive and user-friendly interface for managing books within a library. With features to add, update, delete, issue, and return books, it enables users to maintain a detailed and organized inventory. Each book's status (Available or Issued) is displayed in real-time, and clear messages guide users through each operation. The application validates inputs and provides helpful warnings or confirmations to prevent errors. Overall, this code offers a solid foundation for a basic library management system, suitable for small to medium-sized collections.


Library Management System
Library Management System
Title:
Author:
ISBN:
Add Book
Update Book
Delete Book
Issue Book
Return Book


Library Management System
Library Management System
Title: Emma
Author: Jane Austen
ISBN: 896548
Add Book
Book Added
Emma by Jane Austen added to the library.
OK
Return Book
It Ends With Us by Coollen Hoover (ISBN: 123456) - Available
Emma by Jane Austen (ISBN: 896548) - Available


Library Management System
Library Management System
Title:
Author:
ISBN:
Add Book
Book Issued
Emma has been issued.
OK
Return Book
It Ends With Us by Coollen Hoover (ISBN: 123456) - Available
Emma by Jane Austen (ISBN: 896548) - Issued


Library Management System
Library Management System
Title:
Author:
ISBN:
Add Book
Book Returned
Emma has been returned.
OK
Return Book
It Ends With Us by Coollen Hoover (ISBN: 123456) - Available
Emma by Jane Austen (ISBN: 896548) - Available


Library Management System
Library Management System
Title:
Author:
ISBN:
Add Book
Confirm Delete
Are you sure you want to delete Emma?
Yes
No
Return Book
It Ends With Us by Coollen Hoover (ISBN: 123456) - Available
Emma by Jane Austen (ISBN: 896548) - Available