```
import tkinter as tk
from tkinter import ttk, messagebox
from PIL import Image, ImageTk
import csv
import os

# Define the CSV file name
csv_file = "library_data.csv"

# Function to create the CSV file if it doesn't exist
def create_csv_file_if_not_exists():
    if not os.path.exists(csv_file):
        with open(csv_file, mode='w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(["Shelf Number", "Registration Number", "Title", "Author"])

# Function to add a new book
def add_book():
    shelf_number = shelf_entry.get()
    registration_number = registration_entry.get()
    title = title_entry.get()
    author = author_entry.get()

    with open(csv_file, mode='a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([shelf_number, registration_number, title, author])
    messagebox.showinfo("Success", "Book added successfully.")

# Function to delete a book by registration number
def delete_book():
    registration_number = delete_entry.get()
    temp_file = "temp.csv"
    deleted = False

    with open(csv_file, mode='r', newline='') as source, open(temp_file, mode='w',
newline='') as target:
        reader = csv.reader(source)
        writer = csv.writer(target)
        header = next(reader)
        writer.writerow(header)

        for row in reader:
            if row[1] == registration_number:
                deleted = True
            else:
                writer.writerow(row)

    os.replace(temp_file, csv_file)
    if deleted:
        messagebox.showinfo("Success", f"Book with registration number
{registration_number} deleted successfully.")
    else:
        messagebox.showinfo("Error", f"Book with registration number
{registration_number} not found.")
```

```
# Function to fetch book information by title or registration number
def fetch_book_info():
    criteria = fetch_entry.get()
    found_books = []

    with open(csv_file, mode='r', newline='') as file:
        reader = csv.reader(file)
        header = next(reader)

        for row in reader:
            if criteria.lower() in row[2].lower() or criteria == row[1]:
                found_books.append(row)

        if found_books:
            info_text.config(state=tk.NORMAL)
            info_text.delete(1.0, tk.END)
            info_text.insert(tk.END, "Book Information:\n")
            for book in found_books:
                info_text.insert(tk.END, f"Shelf Number: {book[0]}\n")
                info_text.insert(tk.END, f"Registration Number: {book[1]}\n")
                info_text.insert(tk.END, f"Title: {book[2]}\n")
                info_text.insert(tk.END, f"Author: {book[3]}\n\n")
            info_text.config(state=tk.DISABLED)
        else:
            messagebox.showinfo("Info", "No books found with the given criteria.")

# Function to display books on a shelf by shelf number
def display_books_on_shelf():
    shelf_number = shelf_display_entry.get()
    found_books = []

    with open(csv_file, mode='r', newline='') as file:
        reader = csv.reader(file)
        header = next(reader)

        for row in reader:
            if row[0] == shelf_number:
                found_books.append(row)

        if found_books:
            display_text.config(state=tk.NORMAL)
            display_text.delete(1.0, tk.END)
            display_text.insert(tk.END, f"Books on Shelf {shelf_number}:\n")
            for book in found_books:
                display_text.insert(tk.END, f"Registration Number: {book[1]}\n")
                display_text.insert(tk.END, f"Title: {book[2]}\n")
                display_text.insert(tk.END, f"Author: {book[3]}\n\n")
            display_text.config(state=tk.DISABLED)
        else:
            messagebox.showinfo("Info", f"No books found on Shelf {shelf_number}.")

# Create the main window
root = tk.Tk()
root.title("Library Management System")
root.attributes('-fullscreen', True)  # Set to fullscreen
```

```
# Create a canvas for graphics
canvas = tk.Canvas(root, width=root.winfo_screenwidth(),
height=root.winfo_screenheight())
canvas.pack()

# Load and display an image (replace 'background.jpg' with your own image)
bg_image = Image.open("background.jpg")
bg_photo = ImageTk.PhotoImage(bg_image)
bg_label = tk.Label(canvas, image=bg_photo)
bg_label.photo = bg_photo
bg_label.place(x=0, y=0, relwidth=1, relheight=1)

# Create labels and entry widgets for adding books
shelf_label = tk.Label(canvas, text="Shelf Number:", font=("Helvetica", 16))
shelf_entry = tk.Entry(canvas, font=("Helvetica", 16))
registration_label = tk.Label(canvas, text="Registration Number:", font=("Helvetica",
16))
registration_entry = tk.Entry(canvas, font=("Helvetica", 16))
title_label = tk.Label(canvas, text="Title:", font=("Helvetica", 16))
title_entry = tk.Entry(canvas, font=("Helvetica", 16))
author_label = tk.Label(canvas, text="Author:", font=("Helvetica", 16))
author_entry = tk.Entry(canvas, font=("Helvetica", 16))
add_button = tk.Button(canvas, text="Add Book", command=add_book, font=("Helvetica",
16))

# Create labels and entry widgets for deleting books
delete_label = tk.Label(canvas, text="Registration Number:", font=("Helvetica", 16))
delete_entry = tk.Entry(canvas, font=("Helvetica", 16))
delete_button = tk.Button(canvas, text="Delete Book", command=delete_book,
font=("Helvetica", 16))

# Create labels and entry widgets for fetching book information
fetch_label = tk.Label(canvas, text="Search Criteria (Title/Registration Number):",
font=("Helvetica", 16))
fetch_entry = tk.Entry(canvas, font=("Helvetica", 16))
fetch_button = tk.Button(canvas, text="Fetch Book Info", command=fetch_book_info,
font=("Helvetica", 16))
info_text = tk.Text(canvas, height=10, width=50, font=("Helvetica", 14))
info_text.config(state=tk.DISABLED)

# Create labels and entry widgets for displaying books on a shelf
shelf_display_label = tk.Label(canvas, text="Shelf Number:", font=("Helvetica", 16))
shelf_display_entry = tk.Entry(canvas, font=("Helvetica", 16))
display_button = tk.Button(canvas, text="Display Books on Shelf",
command=display_books_on_shelf, font=("Helvetica", 16))
display_text = tk.Text(canvas, height=10, width=50, font=("Helvetica", 14))
display_text.config(state=tk.DISABLED)

# Place widgets in the window
shelf_label.place(x=100, y=100)
shelf_entry.place(x=300, y=100)
registration_label.place(x=100, y=150)
registration_entry.place(x=300, y=150)
title_label.place(x=100, y=200)
```

```
title_entry.place(x=300, y=200)
author_label.place(x=100, y=250)
author_entry.place(x=300, y=250)
add_button.place(x=200, y=300)

delete_label.place(x=100, y=400)
delete_entry.place(x=300, y=400)
delete_button.place(x=200, y=450)

fetch_label.place(x=100, y=550)
fetch_entry.place(x=300, y=550)
fetch_button.place(x=200, y=600)
info_text.place(x=600, y=100)

shelf_display_label.place(x=100, y=750)
shelf_display_entry.place(x=300, y=750)
display_button.place(x=200, y=800)
display_text.place(x=600, y=550)

# Create the CSV file if it doesn't exist
create_csv_file_if_not_exists()

# Start the tkinter main loop
root.mainloop()
```