Assignment 03

Library Inventory Manager - Project Report

Name = Anant Kumar

Roll no. = 2501730189

Course = B-Tech CSE AI/ML

Section = A

1. Introduction

This report presents the design, development, and implementation of the Library Inventory Manager project, created as part of the Programming for Problem Solving using Python course. The project applies Object-Oriented Programming (OOP), file handling using JSON, exception management, and modular package structuring.

2. Objectives

- Implement object-oriented design with classes and methods.
- Manage a library catalog using Book and LibraryInventory classes.
- Ensure data persistence using JSON files.
- Build a menu-driven command-line interface for user interaction.
- Use exception handling and logging for robust execution.
- Package the project with proper structure and documentation.

3. System Design

3.1 Book Class

The Book class handles attributes such as title, author, ISBN, and status. It includes methods to issue and return books, display book information, and convert objects to dictionaries for JSON storage.

3.2 LibraryInventory Class

The LibraryInventory class maintains a list of Book objects and provides functionalities such as adding books, searching by title or ISBN, and viewing all books.

3.3 File Persistence

The system uses a JSON file to store and retrieve book records. The program handles missing or corrupted files using try-except blocks.

3.4 Command-Line Interface (CLI)

A user-friendly CLI provides options to add, issue, return, search for books, or display the entire catalog.

4. Exception Handling & Logging

The project integrates Python's logging module, capturing events such as book additions, file load errors, and successful operations. Exception handling ensures the program remains stable under invalid input or file issues.

5. Conclusion

The Library Inventory Manager successfully demonstrates OOP concepts, modular coding, file-based persistence, and robust programming practices. It forms a scalable base for any future enhancements such as GUI development or online catalog syncing.

6. Code

```
import json
import logging
from pathlib import Path

logging.basicConfig(
    filename="library.log",
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(message)s"
)

class Book:
    def __init__(self, title, author, isbn, status="available"):
        self.title = title
        self.author = author
        self.isbn = isbn
        self.status = status

    def __str__(self):
        return f"{self.title} by {self.author} | ISBN: {self.isbn} | Status: {self.status}"

    def to_dict(self):
        return {
            "title": self.title,
            "author": self.author,
            "isbn": self.isbn,
            "status": self.status
        }
```

```
    def issue(self):
        if self.status == "available":
            self.status = "issued"
            return True
        return False

    def return_book(self):
        if self.status == "issued":
            self.status = "available"
            return True
        return False

class LibraryInventory:
    def __init__(self, file_path="catalog.json"):
        self.file_path = Path(file_path)
        self.books = []
        self.load_data()

    def load_data(self):
        try:
            if self.file_path.exists():
                with open(self.file_path, "r", encoding="utf-8") as f:
                    data = json.load(f)
                    self.books = [Book(**item) for item in data]
            else:
                self.save_data()
```

```
    except Exception as e:
        logging.error(f"Error loading data: {e}")
        self.books = []

def save_data(self):
    try:
        with open(self.file_path, "w", encoding="utf-8") as f:
            json.dump([book.to_dict() for book in self.books], f, indent=4)
    except Exception as e:
        logging.error(f"Error saving data: {e}")

def add_book(self, title, author, isbn):

    if self.search_by_isbn(isbn):
        return False, "A book with this ISBN already exists."
    self.books.append(Book(title, author, isbn))
    logging.info(f"Added book: {title} ({isbn})")
    self.save_data()
    return True, "Book added."

def search_by_isbn(self, isbn):
    for book in self.books:
        if book.isbn == isbn:
            return book
    return None

def search_by_title(self, title):
```

```
        return [book for book in self.books if title.lower() in book.title.lower()]

    def display_all(self):
        return self.books

    def issue_book(self, isbn):
        book = self.search_by_isbn(isbn)
        if book and book.issue():
            self.save_data()
            logging.info(f"Issued: {isbn}")
            return True
        return False

    def return_book(self, isbn):
        book = self.search_by_isbn(isbn)
        if book and book.return_book():
            self.save_data()
            logging.info(f"Returned: {isbn}")
            return True
        return False

def show_menu():
    print("""
-----------------------------------
        LIBRARY MANAGER
-----------------------------------
1. Add Book
```

```
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------
""")

def run_cli():
    inventory = LibraryInventory()

    while True:
        show_menu()
        choice = input("Enter your choice (1-6): ").strip()

        if choice == "1":
            title = input("Enter book title: ").strip()
            author = input("Enter author name: ").strip()
            isbn = input("Enter ISBN: ").strip()
            if not (title and author and isbn):
                print("All fields are required.")
                continue
            ok, msg = inventory.add_book(title, author, isbn)
            print(msg)

        elif choice == "2":
            isbn = input("Enter ISBN to issue: ").strip()
```

```
        if inventory.issue_book(isbn):
            print("Book issued successfully!")
        else:
            print("Cannot issue book. It may already be issued or ISBN is invalid.")

    elif choice == "3":
        isbn = input("Enter ISBN to return: ").strip()
        if inventory.return_book(isbn):
            print("Book returned successfully!")
        else:
            print("Book not found or not currently issued.")

    elif choice == "4":
        books = inventory.display_all()
        if not books:
            print("No books in catalog.")
        else:
            print("\n--- Library Books ---")
            for book in books:
                print(book)

    elif choice == "5":
        query = input("Enter title or ISBN to search: ").strip()
        if not query:
            print("Enter a non-empty search query.")
            continue
        book = inventory.search_by_isbn(query)
```

```
            if book:
                print(book)
            else:
                results = inventory.search_by_title(query)
                if results:
                    for b in results:
                        print(b)
                else:
                    print("No matching books found.")

        elif choice == "6":
            print("Exiting program. Goodbye!")
            break

        else:
            print("Invalid choice! Please select from 1 to 6.")

if __name__ == "__main__":
    run_cli()
```

7. Screenshots

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 1
Enter book title: ABC
Enter author name: DEF
Enter ISBN: GHI
Book added.
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 2
Enter ISBN to issue: GHI
Book issued successfully!
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 3
Enter ISBN to return: qwfg
Book not found or not currently issued.
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 3
Enter ISBN to return: GHI
Book returned successfully!
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 4

--- Library Books ---
ABC by DEF | ISBN: GHI | Status: available
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 5
Enter title or ISBN to search: ABC
ABC by DEF | ISBN: GHI | Status: available
```

```
----------------------------------
        LIBRARY MANAGER
----------------------------------
1. Add Book
2. Issue Book
3. Return Book
4. View All Books
5. Search Book
6. Exit
----------------------------------

Enter your choice (1-6): 6
Exiting program. Goodbye!
PS C:\Users\Anant\Desktop\Python>
```