System Requirements Specification Index

For

Library Book Management System

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

Library Book Management System
System Requirements Specification

1 PROJECT ABSTRACT

Riverside Public Library needs a book management system to efficiently maintain and organize its expanding collection. The system should catalog books, track their availability, and generate reports. It must support key functions such as categorizing books by genre, filtering by availability, transforming book data, and calculating collection statistics. The solution should enable library staff to effectively manage and analyze the collection with ease.

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	1. System needs to store and categorize different genres of books (fiction, non-fiction, reference, children's, biography) 2. System must support filtering books by genre, availability, publication decade, or keyword 3. Console should handle different list comprehension operations like Basic filtering (books by genre, availability), Transforming data (formatting titles, creating citation strings), Nested list comprehension (for multi-criteria filtering), List comprehension with conditionals (if-else structures)

3 CONSTRAINTS

3.1 INPUT REQUIREMENTS

1. Book Records:

- Must be stored as dictionaries with fields for id, title, author, genre, publication_year, available, popularity_score
- Must be stored in list variable `books`
- Example: `{"id": "B001", "title": "Python Fundamentals", "author": "John Smith", "genre": "reference", "publication_year": 2019, "available": True, "popularity_score": 4.5}`

2. Book Genres:
   - Must be one of: "fiction", "non-fiction", "reference", "children", "biography"
   - Must be stored as string in book's "genre" field
   - Example: "fiction"
3. Availability Status:
   - Must be stored as boolean in "available" field
   - Must be True (available) or False (on loan)
   - Example: True
4. Popularity Score:
   - Example: 4.5 - Must be stored as float in "popularity_score" field
   - Must be between 1.0-5.0
   - Example: 4.5
5. Predefined Books:
   - Must use these exact predefined books in the initial book list:
     - `{"id": "B001", "title": "Python Fundamentals", "author": "John Smith", "genre": "reference", "publication_year": 2019, "available": True, "popularity_score": 4.5}`
     - `{"id": "B002", "title": "Mystery at Midnight", "author": "Jane Doe", "genre": "fiction", "publication_year": 2018, "available": False, "popularity_score": 4.2}`
     - `{"id": "B003", "title": "History of Computing", "author": "Alan Turing", "genre": "non-fiction", "publication_year": 2015, "available": True, "popularity_score": 3.8}`
     - `{"id": "B004", "title": "The Dragon's Quest", "author": "Emily Johnson", "genre": "children", "publication_year": 2020, "available": True, "popularity_score": 4.7}`
     - `{"id": "B005", "title": "Life of Einstein", "author": "Robert Brown", "genre": "biography", "publication_year": 2017, "available": False, "popularity_score": 4.1}`

6. New Arrivals:
    - Must use these exact predefined items in the new arrivals list:
        - `{"id": "N001", "title": "Data Science Handbook", "author": "Sarah Miller", "genre": "reference", "publication_year": 2023, "available": True, "popularity_score": 4.9}`
        - `{"id": "N002", "title": "Quantum Physics Simplified", "author": "Richard Feynman", "genre": "non-fiction", "publication_year": 2022, "available": True, "popularity_score": 4.3}`

3.2 Operations Constraints

1. Book List Combination:
    - Must use list comprehension to combine lists with additional transformation
    - Example: `[{**book, "section": "New"} for book in new_arrivals]`
2. Book Filtering by Genre:
    - Must use list comprehension
    - Example: `[book for book in books if book["genre"] == "fiction"]`
3. Available Books Filtering:
    - Must use list comprehension
    - Example: `[book for book in books if book["available"] == True]`transformation
4. Book Title Transformation:
    - Must use list comprehension with string operations
    - Example: `[book["title"].upper() for book in books]`
5. Multi-criteria Filtering:
    - Must use list comprehension with multiple conditions
    - Example: `[book for book in books if book["genre"] == "reference" and book["available"] == True]`
6. Book Publication Decade Filtering:
    - Must use list comprehension with string operations
    - Example: `[book["title"].upper() for book in books]`
7. Book Publication Decade Filtering:
    - Must use list comprehension with formatted strings

- Example: `[f"{book['author']} ({book['publication_year']}). {book['title']}." for book in books]`

8. Conditional Transformation:
   - Must use list comprehension with if-else
   - Example: `[book["title"] if book["available"] else f"{book['title']} (ON LOAN)" for book in books]`
9. Genre Count Calculation:
   - Must use list comprehension with counting operations
   - Example: `len([book for book in books if book["genre"] == "fiction"])`
10. Average Popularity Calcultaion:
    - Must use list comprehension with statistical operations
    - Example: `sum([book["popularity_score"] for book in books]) / len(books)`

3.3 Output Constraints

1. Display Format:
   - Show book ID, title, author, genre, publication year, availability, popularity score
   - Format availability with "Available" or "On Loan" indicator
   - Format popularity with star rating (★)
   - Each book must be displayed on a new line
2. Output Format:
   - Show "===== LIBRARY BOOK MANAGEMENT SYSTEM ====="
   - Show "Total Books: {count}"
   - Show "Available Books: {count}"
   - Show "Current Book Collection:"
   - Show books with format: "{id} | {title} | {author} | {genre} | {year} | {availability} | Rating: {stars}"
   - Show "Filtered Results:" when displaying search results

4. Template code Structure:

1. Data Management Functions:
   - `initialize_data()` - creates the initial book and new arrivals lists
2. List Comprehension Functions:
   - `filter_by_genre(books, genre)` - filters books by genre using list comprehension

- `filter_by_availability(books, available)` - filters books by availability
- `filter_by_decade(books, decade)` - filters books by publication decade
- `filter_by_keyword(books, keyword)` - filters books by keyword in title or author
- `transform_titles(books, case)` - transforms book titles to specified case
- `generate_citations(books)` - generates formatted citations for books
- `get_book_availability(books)` - creates list with availability indicators
- `calculate_genre_counts(books)` - counts books in each genre
- `calculate_average_popularity(books)` - calculates average popularity
- - `integrate_new_arrivals(books, new_arrivals)` - combines book lists with transformation

3. Display Functions:

- `get_formatted_book(book)` - formats a book for display
- `display_data(data, data_type)` - displays books or other data types

4. Program Control:

- `main()` - main program function

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. View the main menu
3. Select operations
4. Perform operations on the book list
5. View results after each operation
6. Exit program when finished