Efficient Library Management System (ELMS)

1. Introduction

The **Efficient Library Management System (ELMS)** is a C++ project designed to demonstrate the integration of **Object-Oriented Programming (OOP) principles** with multiple **data structures** such as **Stack, Queue, Linked List, Hash Table, and Binary Search Tree (BST)**.
It manages library operations like **borrowing, returning, employee search, and book search** efficiently.

This project also showcases concepts of **templates, file handling, and exception handling**, making it an ideal academic and practical implementation of advanced C++ programming.

---

2. Objectives

- Apply **OOP principles** (Abstraction, Encapsulation, Composition).
- Demonstrate the use of **multiple data structures** in a real-world problem.
- Enable efficient **book and employee management** using advanced C++ concepts.
- Showcase **file handling and exception handling** for robust software design.
- Provide a **structured, extensible, and maintainable project** suitable for academic or professional use.

---

3. Features

- OOP concepts: **Composition, Encapsulation, Abstraction**
- Book borrowing & returning handled with Queue
- Overdue and undo/redo functionality with Stack
- Fast book searching via Binary Search Tree (BST)
- Employee record management using Hash Table
- Data persistence via file handling (DataFile.txt)
- Exception handling for safe operations
- Use of templates for generic implementations
- User-friendly console interface with structured output

---

4. Technologies & Concepts Used

- **Language:** C++ (C++11/14 standard compatible)
- **OOP Concepts:** Abstraction, Encapsulation, Composition
- **Data Structures:**
  - Stack
  - Queue
  - Linked List
  - Hash Table
  - Binary Search Tree (BST)
- **Other Concepts:** Templates, File Handling, Exception Handling

---

5. Program Flow

1. Load book & employee data from DataFile.txt.
2. Display all loaded information.
3. Perform **book borrowing operations** (queue-based).
4. Perform **book returning operations** (queue + stack for undo).
5. Undo specific operations using **stack**.
6. Perform **book searching** using BST.
7. Perform **employee searching** using Hash Table.
8. Display program completion message.

---

6. How to Run

6.1 Clone the Repository

```
git clone https://github.com/Muneeb-techpro/efficient-library-management-system.git
cd efficient-library-management-system/src
```

6.2 Compile the Project

```
g++ *.cpp -o LibraryApp
```

6.3 Run the Executable

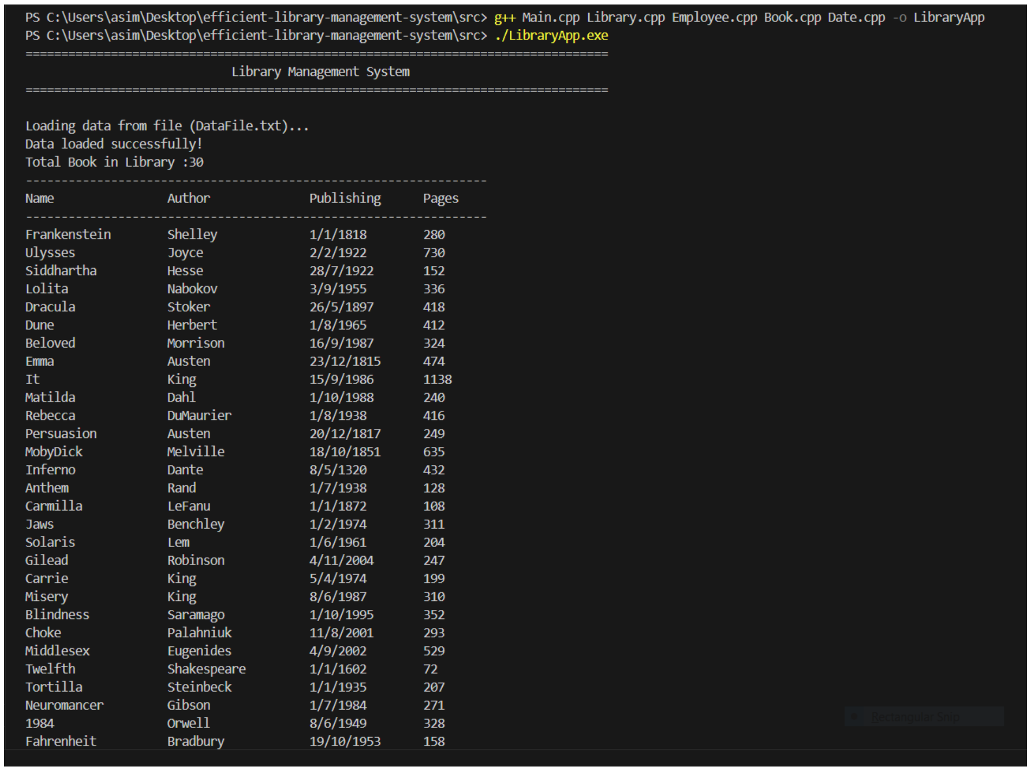
- **On Linux / macOS:**
  ```
  ./LibraryApp
  ```
- **On Windows (PowerShell or CMD):**
  ```
  LibraryApp.exe
  ```

---

7. Output (Screenshots + Logs)

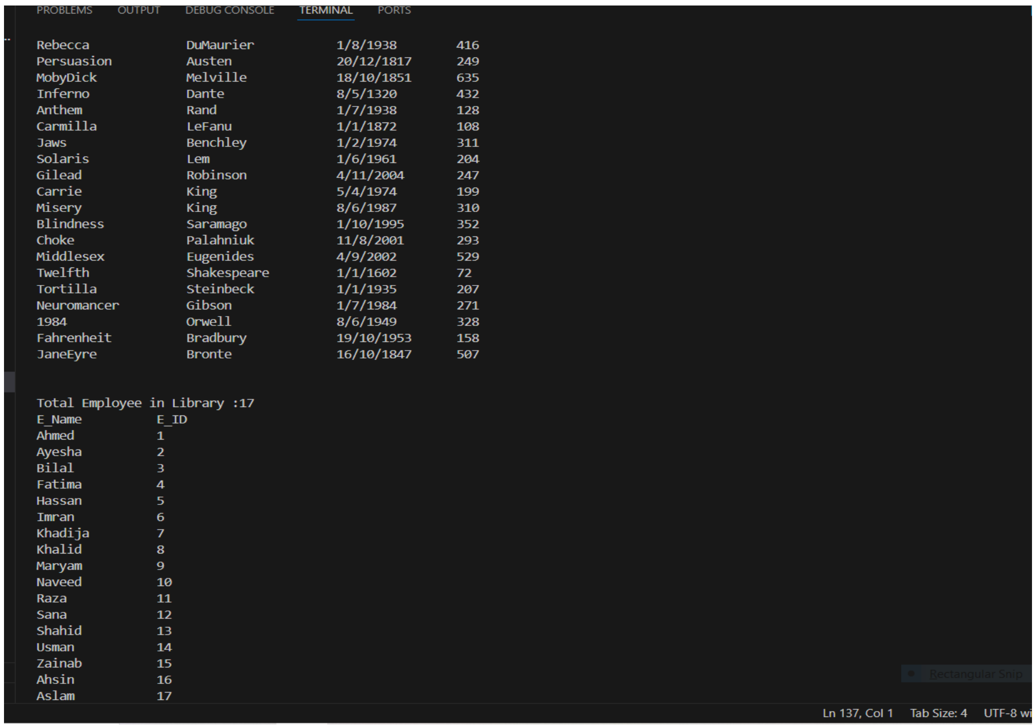
📷 Screenshots

- 01_output.png – Program start screen



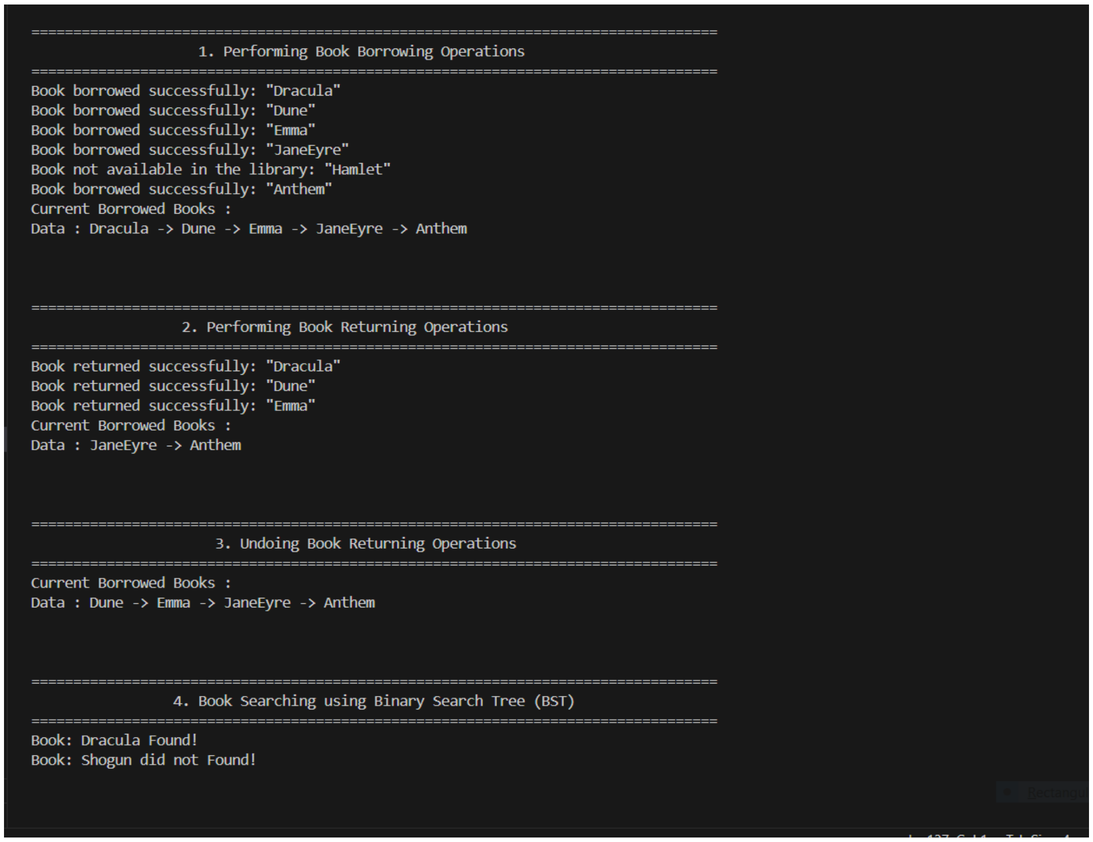
```
PS C:\Users\asim\Desktop\efficient-library-management-system\src> g++ Main.cpp Library.cpp Employee.cpp Book.cpp Date.cpp -o LibraryApp
PS C:\Users\asim\Desktop\efficient-library-management-system\src> ./LibraryApp.exe
==================================================================================
                             Library Management System
==================================================================================

Loading data from file (DataFile.txt)...
Data loaded successfully!
Total Book in Library :30
-----------------------------------------------------------------
Name                Author              Publishing      Pages
-----------------------------------------------------------------
Frankenstein        Shelley             1/1/1818        280
Ulysses             Joyce               2/2/1922        730
Siddhartha          Hesse               28/7/1922       152
Lolita              Nabokov             3/9/1955        336
Dracula             Stoker              26/5/1897       418
Dune                Herbert             1/8/1965        412
Beloved             Morrison            16/9/1987       324
Emma                Austen              23/12/1815      474
It                  King                15/9/1986       1138
Matilda             Dahl                1/10/1988       240
Rebecca             DuMaurier           1/8/1938        416
Persuasion          Austen              20/12/1817      249
MobyDick            Melville            18/10/1851      635
Inferno             Dante               8/5/1320        432
Anthem              Rand                1/7/1938        128
Carmilla            LeFanu              1/1/1872        108
Jaws                Benchley            1/2/1974        311
Solaris             Lem                 1/6/1961        204
Gilead              Robinson            4/11/2004       247
Carrie              King                5/4/1974        199
Misery              King                8/6/1987        310
Blindness           Saramago            1/10/1995       352
Choke               Palahniuk           11/8/2001       293
Middlesex           Eugenides           4/9/2002        529
Twelfth             Shakespeare         1/1/1602        72
Tortilla            Steinbeck           1/1/1935        207
Neuromancer         Gibson              1/7/1984        271
1984                Orwell              8/6/1949        328
Fahrenheit          Bradbury            19/10/1953      158
```

- 02_output.png – Data successfully loaded



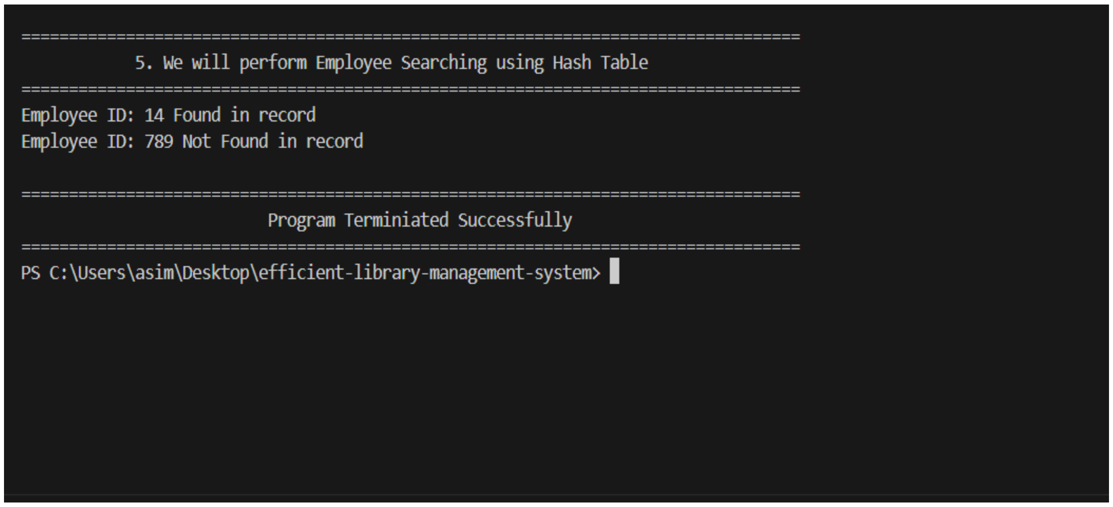
```
Rebecca             DuMaurier           1/8/1938        416
Persuasion          Austen              20/12/1817      249
MobyDick            Melville            18/10/1851      635
Inferno             Dante               8/5/1320        432
Anthem              Rand                1/7/1938        128
Carmilla            LeFanu              1/1/1872        108
Jaws                Benchley            1/2/1974        311
Solaris             Lem                 1/6/1961        204
Gilead              Robinson            4/11/2004       247
Carrie              King                5/4/1974        199
Misery              King                8/6/1987        310
Blindness           Saramago            1/10/1995       352
Choke               Palahniuk           11/8/2001       293
Middlesex           Eugenides           4/9/2002        529
Twelfth             Shakespeare         1/1/1602        72
Tortilla            Steinbeck           1/1/1935        207
Neuromancer         Gibson              1/7/1984        271
1984                Orwell              8/6/1949        328
Fahrenheit          Bradbury            19/10/1953      158
JaneEyre            Bronte              16/10/1847      507


Total Employee in Library :17
E_Name          E_ID
Ahmed           1
Ayesha          2
Bilal           3
Fatima          4
Hassan          5
Imran           6
Khadija         7
Khalid          8
Maryam          9
Naveed          10
Raza            11
Sana            12
Shahid          13
Usman           14
Zainab          15
Ahsin           16
Aslam           17
```

- 03_output.png – Borrowing operations



```
==================================================================================
                    1. Performing Book Borrowing Operations
==================================================================================
Book borrowed successfully: "Dracula"
Book borrowed successfully: "Dune"
Book borrowed successfully: "Emma"
Book borrowed successfully: "JaneEyre"
Book not available in the library: "Hamlet"
Book borrowed successfully: "Anthem"
Current Borrowed Books :
Data : Dracula -> Dune -> Emma -> JaneEyre -> Anthem



==================================================================================
                  2. Performing Book Returning Operations
==================================================================================
Book returned successfully: "Dracula"
Book returned successfully: "Dune"
Book returned successfully: "Emma"
Current Borrowed Books :
Data : JaneEyre -> Anthem



==================================================================================
                      3. Undoing Book Returning Operations
==================================================================================
Current Borrowed Books :
Data : Dune -> Emma -> JaneEyre -> Anthem



==================================================================================
                 4. Book Searching using Binary Search Tree (BST)
==================================================================================
Book: Dracula Found!
Book: Shogun did not Found!
```

- 04_output.png – Employee search results



```
==================================================================================
             5. We will perform Employee Searching using Hash Table
==================================================================================
Employee ID: 14 Found in record
Employee ID: 789 Not Found in record

==================================================================================
                         Program Terminiated Successfully
==================================================================================
PS C:\Users\asim\Desktop\efficient-library-management-system>
```

Complete Output Log

See: program_output.txt

8. Folder Structure

```
efficient-library-management-system/
│
├── data/
│   └── DataFile_backup.txt         # Backup copy of input data
│
├── docs/
│   ├── Library_Management_System_Documentation.docx
│   └── Library_Management_System_Documentation.pdf
│
├── media/
│   ├── 01_Code_Overview.mp4
│   └── 02_Execution_and_Testing.mp4
│
├── output/
│   ├── 01_output.png
│   ├── 02_output.png
│   ├── 03_output.png
│   └── 04_output.png
│
├── src/
│   ├── Book.cpp
│   ├── Book.h
│   ├── BST.h
│   ├── BST.tpp
│   ├── DataFile.txt                # Main working data file
│   ├── Date.cpp
│   ├── Date.h
│   ├── Employee.cpp
│   ├── Employee.h
│   ├── HashTable.h
│   ├── HashTable.tpp
│   ├── Library.cpp
│   ├── Library.h
│   ├── LinkedList.h
│   ├── LinkedList.tpp
│   ├── Main.cpp
│   ├── Queue.h
│   ├── Queue.tpp
│   ├── Stack.h
│   └── Stack.tpp
│
└── README.md
```

9. Exception Handling

The project includes **robust exception handling** for cases such as:

- Missing DataFile.txt
- Invalid book or employee IDs
- Queue underflow/overflow in borrowing/returning operations
- Stack underflow in undo operations

---

10. Future Enhancements

These are some optional future enhancements that may be considered:

- Add a **GUI interface** (Qt/JavaFX/React frontend with backend integration).
- Add **database support** (MySQL/SQLite instead of text file).
- Implement **book recommendation system** using Graph.
- Add **user authentication** and member login system.
- Support for **digital e-books** and issue tracking.

---

11. Conclusion

The **Efficient Library Management System (ELMS)** successfully demonstrates how multiple data structures and OOP principles can be combined to build a robust and efficient C++ application.

It not only serves as a **functional library system** but also as a **learning project** for students to understand **data structures, OOP, templates, and exception handling** in real-world problem solving.