GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTER SCIENCE

TITLE

Library Management System

SUBMITTED BY –

Abhikarsh singh	- 24SCSE1260005
Eshant gupta	- 24SCSE1260002
Mohit Yadav	- 24SCSE1330048
Vishwa raj Saraswat	- 24SCSE1260028

SUBMITTED TO –

Mr. Tarachand Verma

Acknowledgements

We would like to express our special thanks of gratitude to our Guide Mr. Tarachand Verma who helped us a lot in this project, her valuable suggestions helped us to solve tough challenges and without her help this project could not have been completed in time. the topic is "Library Management System", which helped us to gain a significant knowledge in the aforesaid subjects. Secondly, we would like to thank our friends who helped us a lot in finalizing this project within the given time frame.

- Name of Student: Abhikarsh Singh
  Enrollment Number: 24SCSE1260005
- Name of Student: Eshant Gupta
  Enrollment Number: 24SCSE1260002
- Name of Student: Mohit Yadav
  Enrollment Number: 24SCSE1330048
- Name of Student: Vishwa Raj Saraswat
  Enrollment Number: 24SCSE1260028

Library Management System

Creating a Library Management System in Java is a great way to understand object-oriented programming concepts This step-by-step tutorial will guide you through building a simple Library Management System Project in Java

Step 1: Setup Your Project

Step 2: Implement the Book Class

Step 3: Develop the LibraryManager Class

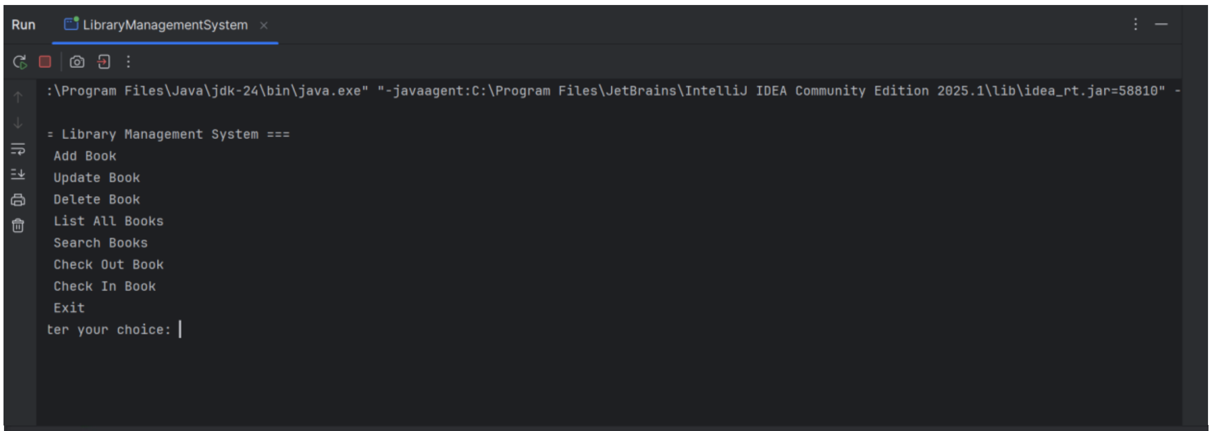
Step 4: Build the Main Class

Step 5: Running the Library Management System

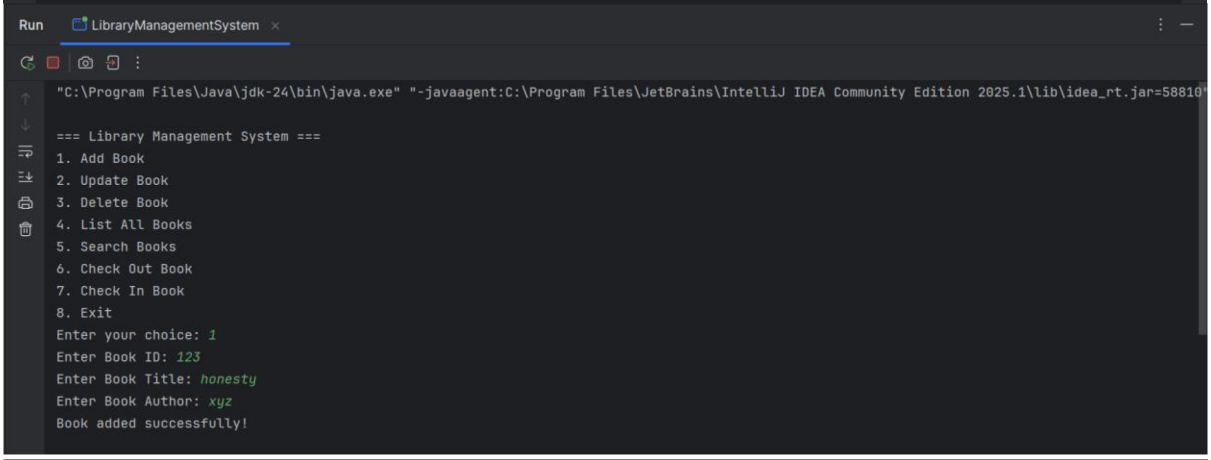
```java
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;

public class LibraryManagementSystem {

    // Book class
    static class Book {
        private int id;
        private String title;
        private String author;
        private boolean isBorrowed;

        public Book(int id, String title, String author) {
            this.id = id;
            this.title = title;
            this.author = author;
            this.isBorrowed = false;
        }

        public int getId() { return id; }
        public void setId(int id) { this.id = id; }
        public String getTitle() { return title; }
        public void setTitle(String title) { this.title = title; }
        public String getAuthor() { return author; }
        public void setAuthor(String author) { this.author = author; }
        public boolean isBorrowed() { return isBorrowed; }
        public void setBorrowed(boolean borrowed) { isBorrowed = borrowed; }

        @Override
        public String toString() {
            return "Book{" +
                    "id=" + id +
                    ", title='" + title + '\'' +
                    ", author='" + author + '\'' +
                    ", isBorrowed=" + isBorrowed +
                    '}';
        }
    }

    // LibraryManager class
    static class LibraryManager {
        private final List<Book> books = new ArrayList<>();

        public void addBook(Book book) {
            books.add(book);
            System.out.println("Book added successfully!");
        }

        public void updateBook(int id, String title, String author) {
            books.stream()
                    .filter(book -> book.getId() == id)
                    .findFirst()
                    .ifPresentOrElse(book -> {
                        book.setTitle(title);
                        book.setAuthor(author);
                        System.out.println("Book updated successfully!");
                    }, () -> System.out.println("Book not found."));
        }

        public void deleteBook(int id) {
            if (books.removeIf(book -> book.getId() == id)) {
                    System.out.println("Book deleted successfully!");
            } else {
                    System.out.println("Book not found!");
            }
        }

        public void listBooks() {
            if (books.isEmpty()) {
                    System.out.println("No books available.");
            } else {
                    books.forEach(System.out::println);
            }
        }

        public void searchBooks(String query) {
            List<Book> foundBooks = books.stream()
                    .filter(book ->
    book.getTitle().toLowerCase().contains(query.toLowerCase())
                            ||
    book.getAuthor().toLowerCase().contains(query.toLowerCase()))
                    .collect(Collectors.toList());
            if (foundBooks.isEmpty()) {
                    System.out.println("No books found matching the query.");
            } else {
                    foundBooks.forEach(System.out::println);
            }
        }

        public void checkOutBook(int id) {
            books.stream()
                    .filter(book -> book.getId() == id && !book.isBorrowed())
                    .findFirst()
                    .ifPresentOrElse(book -> {
                        book.setBorrowed(true);
                        System.out.println("Book checked out successfully!");
                    }, () -> System.out.println("Book is not available or already checked
    out."));
        }

        public void checkInBook(int id) {
            books.stream()
                    .filter(book -> book.getId() == id && book.isBorrowed())
                    .findFirst()
                    .ifPresentOrElse(book -> {
                        book.setBorrowed(false);
                        System.out.println("Book checked in successfully!");
```

```java
        }, () -> System.out.println("Book was not checked out."));
}

private Book inputBookDetails(Scanner scanner) {
    System.out.print("Enter Book ID: ");
    while (!scanner.hasNextInt()) {
        System.out.print("Invalid input. Enter numeric Book ID: ");
        scanner.next();
    }
    int id = scanner.nextInt();
    scanner.nextLine(); // consume newline

    System.out.print("Enter Book Title: ");
    String title = scanner.nextLine();

    System.out.print("Enter Book Author: ");
    String author = scanner.nextLine();

    return new Book(id, title, author);
}

public void start() {
    Scanner scanner = new Scanner(System.in);
    while (true) {
        System.out.println("\n=== Library Management System ===");
        System.out.println("1. Add Book");
        System.out.println("2. Update Book");
        System.out.println("3. Delete Book");
        System.out.println("4. List All Books");
        System.out.println("5. Search Books");
        System.out.println("6. Check Out Book");
        System.out.println("7. Check In Book");
        System.out.println("8. Exit");
        System.out.print("Enter your choice: ");

        int choice;
        while (!scanner.hasNextInt()) {
                System.out.print("Invalid input. Enter a number between 1 and 8: ");
                scanner.next();
            }
            choice = scanner.nextInt();
            scanner.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    addBook(inputBookDetails(scanner));
                    break;
                case 2:
                    System.out.print("Enter Book ID to update: ");
                    int updateId = scanner.nextInt();
                    scanner.nextLine(); // consume newline
                    Book updatedBook = inputBookDetails(scanner);
                    updateBook(updateId, updatedBook.getTitle(), updatedBook.getAuthor());
                    break;
                case 3:
                    System.out.print("Enter Book ID to delete: ");
                    int deleteId = scanner.nextInt();
                    deleteBook(deleteId);
                    break;
                case 4:
                    listBooks();
                    break;
                case 5:
                    System.out.print("Enter search query (title or author): ");
                    String query = scanner.nextLine();
                    searchBooks(query);
                    break;
                case 6:
                    System.out.print("Enter Book ID to check out: ");
                    int checkoutId = scanner.nextInt();
                    checkOutBook(checkoutId);
                    break;
                    case 7:
                        System.out.print("Enter Book ID to check in: ");
                        int checkinId = scanner.nextInt();
                        checkInBook(checkinId);
                        break;
                    case 8:
                        System.out.println("Exiting the system...");
                        return;
                    default:
                        System.out.println("Invalid choice. Please select between 1 and 
8.");
                }
            }
        }
    }

    // Main method
    public static void main(String[] args) {
        LibraryManager manager = new LibraryManager();
        manager.start();
    }
}
```

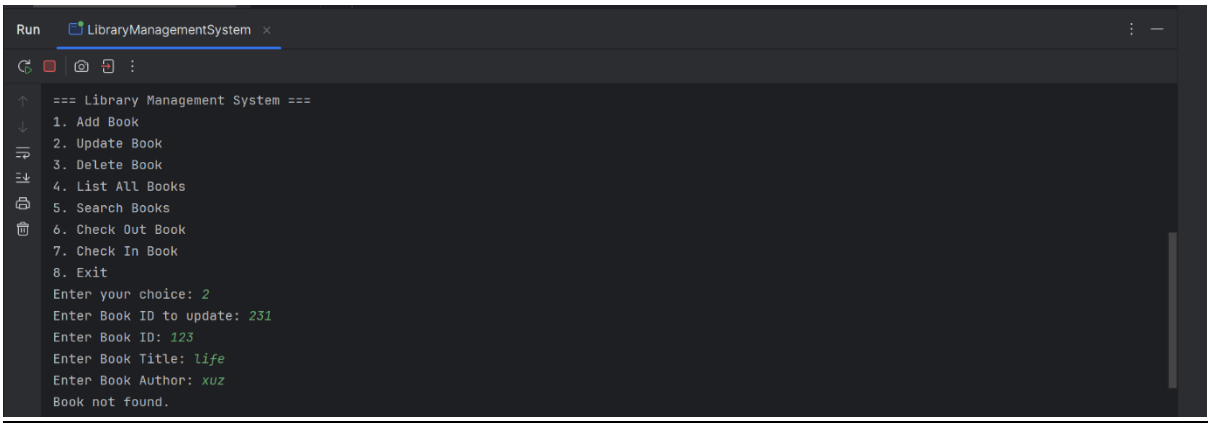
Output -



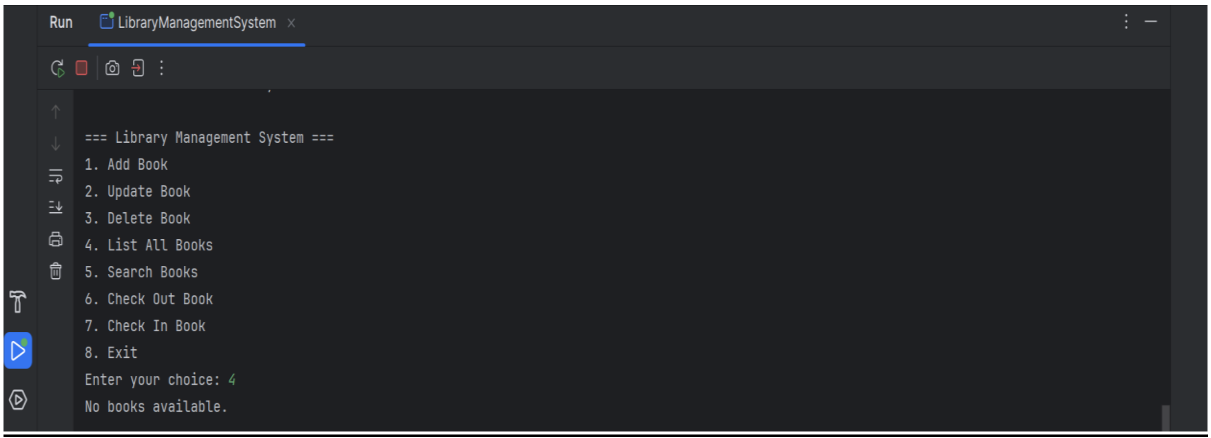
```
:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1\lib\idea_rt.jar=58810" -

= Library Management System ===
 Add Book
 Update Book
 Delete Book
 List All Books
 Search Books
 Check Out Book
 Check In Book
 Exit
ter your choice: 
```



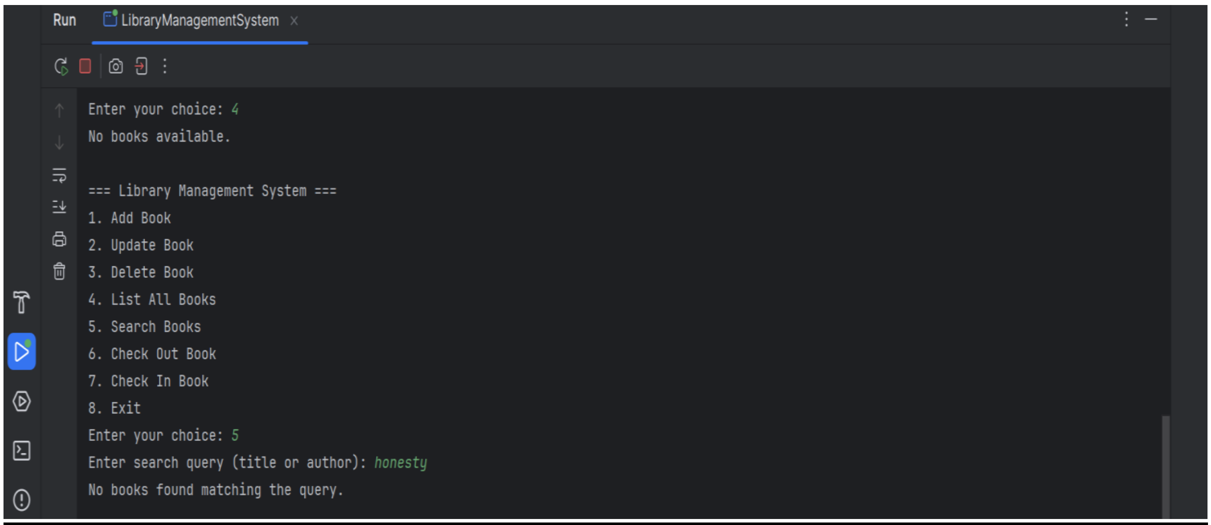
```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1\lib\idea_rt.jar=58810"

=== Library Management System ===
1. Add Book
2. Update Book
3. Delete Book
4. List All Books
5. Search Books
6. Check Out Book
7. Check In Book
8. Exit
Enter your choice: 1
Enter Book ID: 123
Enter Book Title: honesty
Enter Book Author: xyz
Book added successfully!
```



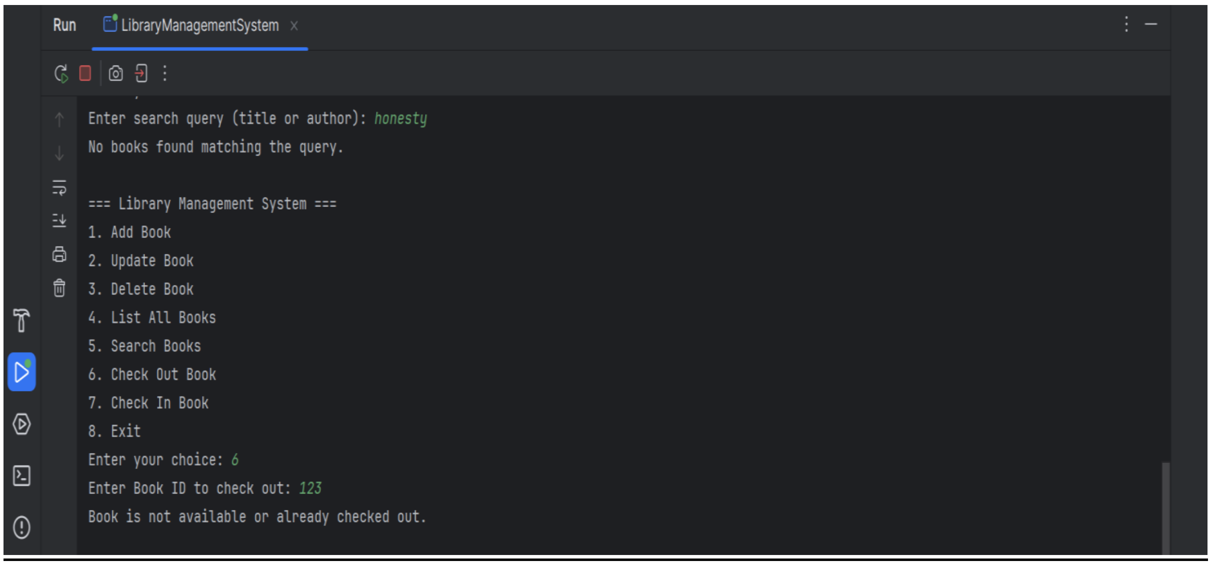
```
=== Library Management System ===
1. Add Book
2. Update Book
3. Delete Book
4. List All Books
5. Search Books
6. Check Out Book
7. Check In Book
8. Exit
Enter your choice: 2
Enter Book ID to update: 231
Enter Book ID: 123
Enter Book Title: life
Enter Book Author: xuz
Book not found.
```



```
=== Library Management System ===
1. Add Book
2. Update Book
3. Delete Book
4. List All Books
5. Search Books
6. Check Out Book
7. Check In Book
8. Exit
Enter your choice: 4
No books available.
```



```
Enter your choice: 4
No books available.

=== Library Management System ===
1. Add Book
2. Update Book
3. Delete Book
4. List All Books
5. Search Books
6. Check Out Book
7. Check In Book
8. Exit
Enter your choice: 5
Enter search query (title or author): honesty
No books found matching the query.
```



```
Enter search query (title or author): honesty
No books found matching the query.

=== Library Management System ===
1. Add Book
2. Update Book
3. Delete Book
4. List All Books
5. Search Books
6. Check Out Book
7. Check In Book
8. Exit
Enter your choice: 6
Enter Book ID to check out: 123
Book is not available or already checked out.
```