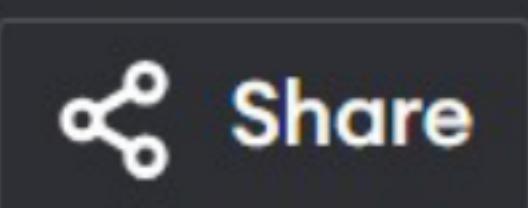


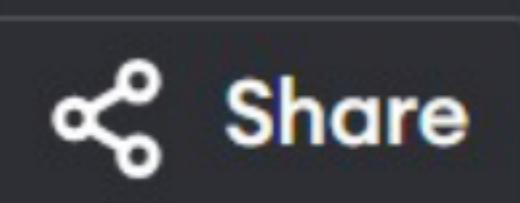
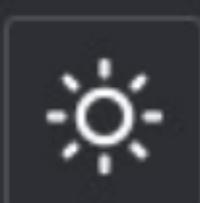
## Main.java



Run

```
3 import models.Book;
4 import models.Patron;
5 import services.LibraryService;
6 import java.util.InputMismatchException;
7 import java.util.Scanner;
8
9 /**
10  * Main application class, providing the Command Line Interface
11  * (CLI).
12  * Handles user interaction and delegates tasks to the
13  * LibraryService.
14  * This class demonstrates clear Input/Output Structure (FR-8 and
15  * NFR-2: Usability).
16  */
17 public class Main {
```

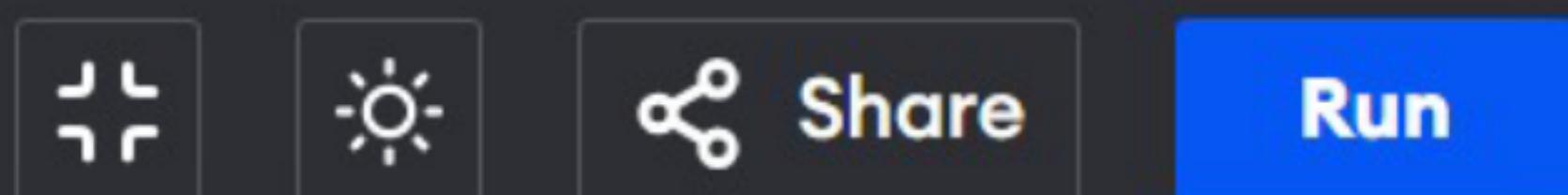
## Main.java



Run

```
15     private static LibraryService service = new LibraryService();
16     private static Scanner scanner = new Scanner(System.in);
17
18     public static void main(String[] args) {
19         System.out.println("--- Library Management System v1.0
---");
20         boolean running = true;
21         while (running) {
22             displayMainMenu();
23             try {
24                 int choice = scanner.nextInt();
25                 scanner.nextLine(); // Consume newline
26                 switch (choice) {
27                     case 1: manageBooks(); break;
28                     case 2: managePatrons(); break;
```

## Main.java



```
28         case 2: managePatrons(); break;
29         case 3: manageCirculation(); break;
30         case 4: generateReports(); break;
31         case 5: running = false; System.out.println
32                   ("Exiting system. Goodbye!"); break;
33     default: System.out.println("Invalid choice.
34                   Please try again.");
35   }
36 } catch (InputMismatchException e) {
37   System.out.println("Error: Please enter a valid
38   number for the menu choice.");
39   scanner.nextLine(); // Clear the invalid input
40 }
```

Main.java



Share

Run

```
41     private static void displayMainMenu() {  
42         System.out.println("\n--- MAIN MENU ---");  
43         System.out.println("1. Book Management (FR-1, FR-3)");  
44         System.out.println("2. Patron Management (FR-4)");  
45         System.out.println("3. Circulation (Borrow/Return) (FR-5,  
46                         FR-6)");  
47         System.out.println("4. Reports (FR-7)");  
48         System.out.println("5. Exit");  
49         System.out.print("Enter choice: ");  
50     }  
51     // --- Module 1: Book Management ---  
52  
53     private static void manageBooks() {  
54         System.out.println("\n--- BOOK MANAGEMENT ---");
```

## Main.java



Share

Run

```
54 System.out.println("\n--- BOOK MANAGEMENT ---");
55 System.out.println("1. Add New Book (FR-1)");
56 System.out.println("2. View All Books");
57 System.out.println("3. Search Book (FR-2)");
58 System.out.println("4. Back to Main Menu");
59 System.out.print("Enter choice: ");
60 try {
61     int choice = scanner.nextInt();
62     scanner.nextLine();
63     switch (choice) {
64         case 1: addBook(); break;
65         case 2: service.getAllBooks().forEach(System.out
66                 ::println); break;
67         case 3: searchBook(); break;
68         case 4: return;
```

## Main.java



Share

Run

```
68         default: System.out.println("Invalid choice.");
69     }
70 } catch (InputMismatchException e) {
71     System.out.println("Error: Invalid input.");
72     scanner.nextLine();
73 }
74
75
76 private static void addBook() {
77     try {
78         System.out.print("Enter ISBN: ");
79         String isbn = scanner.nextLine();
80         System.out.print("Enter Title: ");
81         String title = scanner.nextLine();
82         System.out.print("Enter Author: ");
```

## Main.java

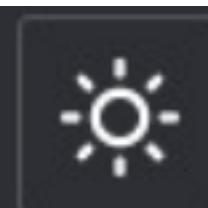
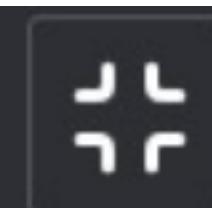


Share

Run

```
82     System.out.print("Enter Author: ");
83     String author = scanner.nextLine();
84     System.out.print("Enter Total Copies: ");
85     int copies = scanner.nextInt();
86     scanner.nextLine();
87
88     Book newBook = new Book(isbn, title, author, copies);
89     service.addBook(newBook);
90     System.out.println("Book added successfully!");
91 } catch (InputMismatchException | IllegalArgumentException
e) {
92     System.out.println("Operation Failed (NFR-4): " + e
         .getMessage());
93     if(scanner.hasNextLine()) scanner.nextLine();
94 }
```

## Main.java

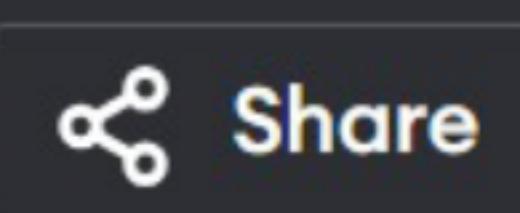


Share

Run

```
95      }
96
97  private static void searchBook() {
98      System.out.print("Enter ISBN, Title, or Author to search:
99          ");
100     String query = scanner.nextLine();
101     Book book = service.findBook(query);
102     if (book != null) {
103         System.out.println("Book Found: " + book);
104     } else {
105         System.out.println("No book found matching the query."
106             );
107     }
}
```

## Main.java



Run

```
110     private static void managePatrons() {  
111         System.out.println("\n--- PATRON MANAGEMENT ---");  
112         System.out.println("1. Register New Patron (FR-4)");  
113         System.out.println("2. Back to Main Menu");  
114         System.out.print("Enter choice: ");  
115         try {  
116             int choice = scanner.nextInt();  
117             scanner.nextLine();  
118             if (choice == 1) {  
119                 registerPatron();  
120             }  
121         } catch (InputMismatchException e) {  
122             System.out.println("Error: Invalid input.");  
123             scanner.nextLine();  
124         }
```

## Main.java



Share

Run

```
125 }  
126  
127 private static void registerPatron() {  
128     try {  
129         System.out.print("Enter Patron ID (e.g., P001): ");  
130         String id = scanner.nextLine();  
131         System.out.print("Enter Name: ");  
132         String name = scanner.nextLine();  
133         System.out.print("Enter Contact Info: ");  
134         String contact = scanner.nextLine();  
135  
136         Patron newPatron = new Patron(id, name, contact);  
137         service.addPatron(newPatron);  
138         System.out.println("Patron registered successfully!");  
139     } catch (IllegalArgumentException e) {
```

## Main.java



Share

Run

```
140         System.out.println("Operation Failed (NFR-4): " + e
                     .getMessage());
141     }
142 }
143
144 private static void manageCirculation() {
145     System.out.println("\n--- CIRCULATION ---");
146     System.out.println("1. Check Out Book (FR-5)");
147     System.out.println("2. Return Book (FR-6)");
148     System.out.println("3. Back to Main Menu");
149     System.out.print("Enter choice: ");
150     try {
151         int choice = scanner.nextInt();
152         scanner.nextLine();
153         switch (choice) {
```

## Main.java



Share

Run

```
154         case 1: checkOutBook(); break;
155         case 2: returnBook(); break;
156         case 3: return;
157         default: System.out.println("Invalid choice.");
158     }
159 } catch (InputMismatchException e) {
160     System.out.println("Error: Invalid input.");
161     scanner.nextLine();
162 }
163
164
165 private static void checkOutBook() {
166     try {
167         System.out.print("Enter Book ISBN to check out: ");
168         String isbn = scanner.nextLine();
```

## Main.java



Share

Run

```
169     System.out.print("Enter Patron ID: ");
170     String patronId = scanner.nextLine();
171     System.out.print("Enter loan duration in days (e.g., 7
172             , 14): ");
173     int loanDays = scanner.nextInt();
174     scanner.nextLine();
175
176     service.checkOutBook(isbn, patronId, loanDays);
177 } catch (Exception e) {
178     System.out.println("Checkout Failed (NFR-4): " + e
179             .getMessage());
180     if(scanner.hasNextLine()) scanner.nextLine();
181 }
```

## Main.java



Share

Run

```
182     private static void returnBook() {
183         try {
184             System.out.print("Enter Book ISBN to return: ");
185             String isbn = scanner.nextLine();
186             System.out.print("Enter Patron ID: ");
187             String patronId = scanner.nextLine();
188
189             double fine = service.returnBook(isbn, patronId);
190             System.out.println("Book returned successfully!");
191             if (fine > 0) {
192                 System.out.printf("OVERDUE: Fine assessed: $%
193                               .2f\n", fine);
194             }
195         } catch (Exception e) {
196             System.out.println("Return Failed (NFR-4): " + e
```

## Main.java

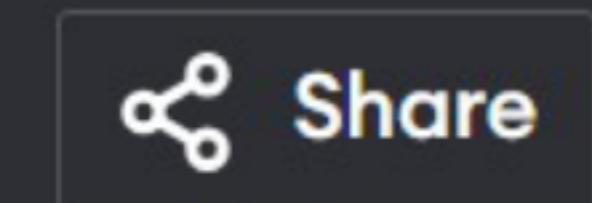


Share

Run

```
195         System.out.println("Return Failed (NFR-4): " + e
196             .getMessage());
197     }
198
199     // --- Module 4: Reports ---
200
201     private static void generateReports() {
202         System.out.println("\n--- REPORTS ---");
203         System.out.println("1. View Patron Borrowing History (FR-7"
204             );
205         System.out.println("2. View All Overdue Loans (FR-7)");
206         System.out.println("3. Back to Main Menu");
207         System.out.print("Enter choice: ");
208         try {
```

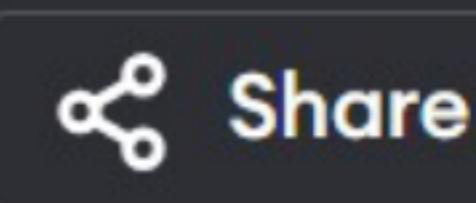
## Main.java



Run

```
208     int choice = scanner.nextInt();
209     scanner.nextLine();
210     switch (choice) {
211         case 1:
212             System.out.print("Enter Patron ID: ");
213             String patronId = scanner.nextLine();
214             service.getPatronHistory(patronId).forEach
215                 (System.out::println);
216             break;
217         case 2:
218             service.getOverdueLoans().forEach(System.out
219                 ::println);
220             break;
221         case 3: return;
222         default: System.out.println("Invalid choice.");
```

## Main.java



Run

```
        (System.out::println);
    break;
case 2:
    service.getOverdueLoans().forEach(System.out
        ::println);
    break;
case 3: return;
default: System.out.println("Invalid choice.");
}
} catch (InputMismatchException e) {
    System.out.println("Error: Invalid input.");
    scanner.nextLine();
}
}
}
```