Library System - How It Works

\This is a simple library system in Java that lets you **add, borrow, and return books**.

1️⃣ How It Stores Books

We have two **HashMaps** to track books:

- `library` → Stores **all books** in the system.
- `borrowedBooks` → Tracks **only books that are currently borrowed**.

Each book is represented by a `Book` object that has:

- An **ID** (auto-incremented)
- **Title & Author**
- **Quantity** (how many copies are available or borrowed)

2️⃣ Borrowing Books

When you borrow a book:

1. It checks if the book exists and has enough copies.
2. If yes, it **reduces the quantity** in `library`.
3. The book is then **added to `borrowedBooks`**, so we know which books are borrowed.
4. If you borrow more of the same book later, it just increases the quantity in `borrowedBooks`.

3️⃣ Returning Books

When returning:

1. It checks if you borrowed that book (from `borrowedBooks`).
2. It **reduces the borrowed quantity** and **increases the quantity** back in `library`.
3. If you return all copies, the book is **removed from `borrowedBooks`**.

4️⃣ Displaying Books Properly

- When borrowing, it shows **all books** from `library`.
- When returning, it only shows **books you've borrowed** from `borrowedBooks`.
- This prevents showing books you never borrowed when returning.

5️⃣ Key Methods

- `displayBooks(Map<String, Book> books)`: Prints books from **either** `library` or `borrowedBooks` depending on the context.
- `findBook(String input, Map<String, Book> books)`: Searches by **ID or title** in **either** `library` or `borrowedBooks`, depending on what we need.

6️⃣ Why Two HashMaps?

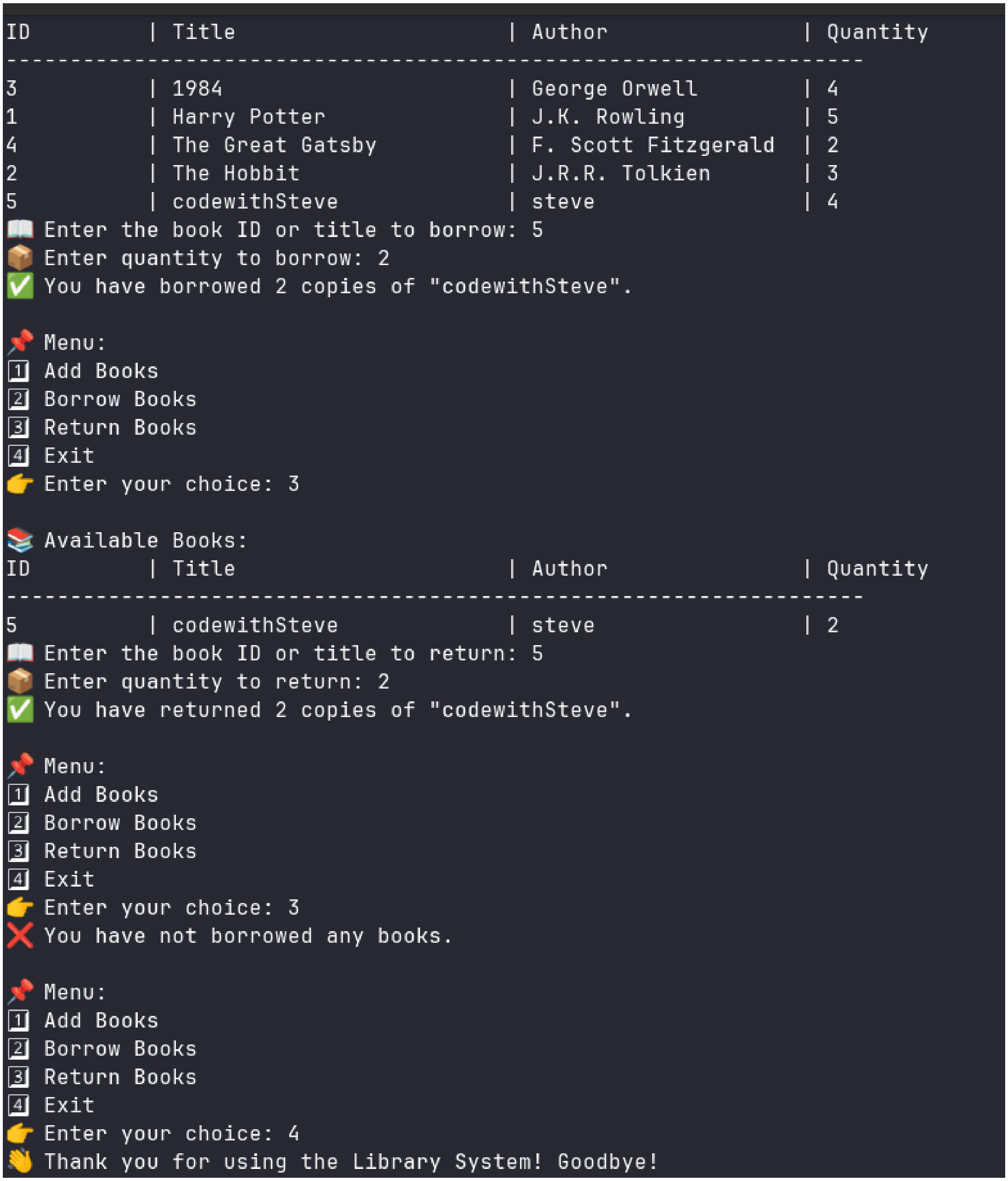
Initially, everything was in **one list**, which caused problems—returning all books, even the ones we never borrowed. Now, **separating borrowed books** makes the system more **accurate and realistic**.

Code

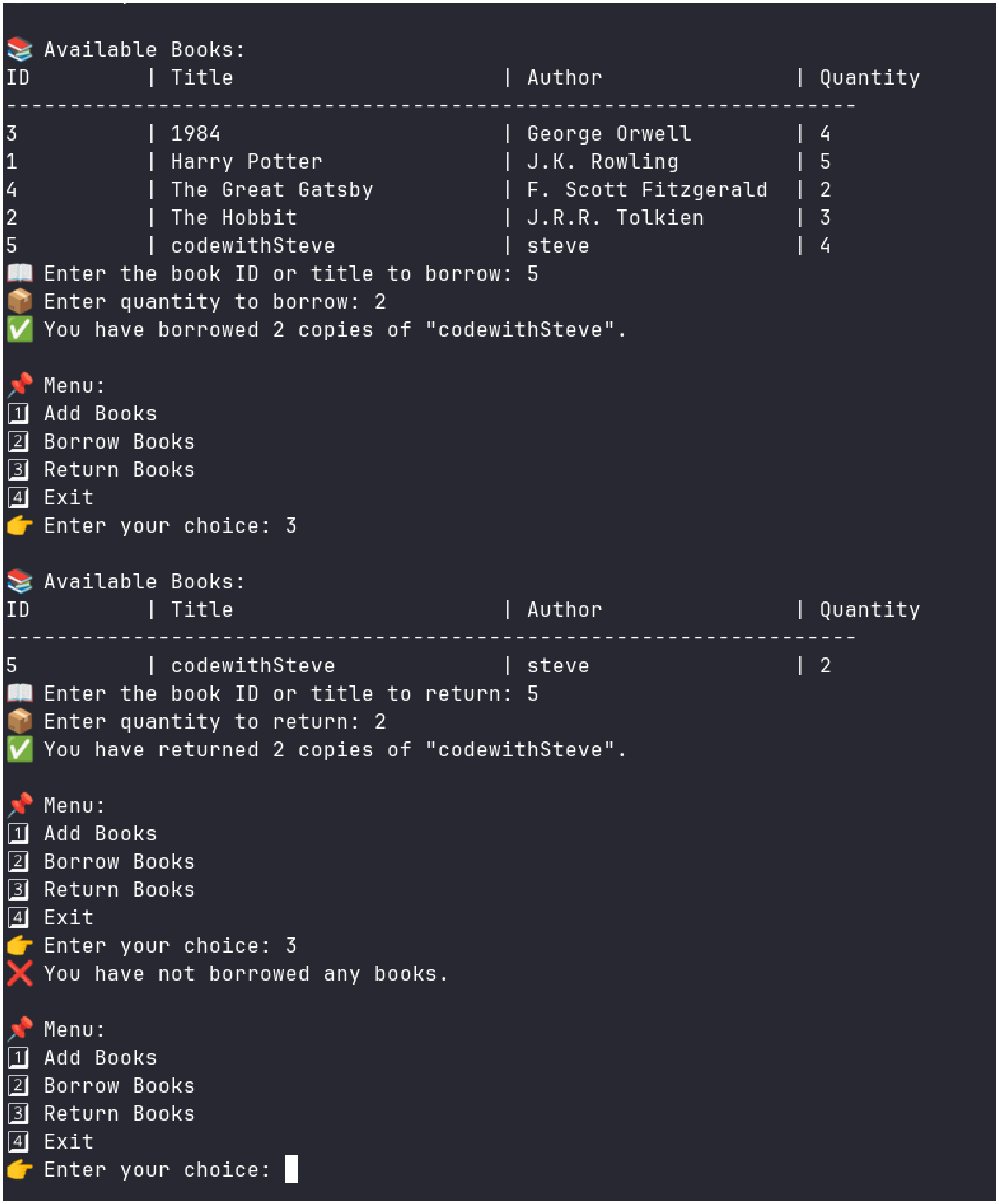
```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class LibrarySystem {
   private static final Map<String, Book> library = new HashMap<>(); // All books
   private static final Map<String, Book> borrowedBooks = new HashMap<>(); // Borrowed books
   private static final Scanner scanner = new Scanner(System.in);
   private static int nextBookId = 1; // Auto-incrementing ID

   public static void main(String[] args) {
      System.out.println("📚 Welcome to the Library System! 📖");

      preloadBooks(); // Load initial books

      while (true) {
         displayMenu();

         int choice = getChoice();

         switch (choice) {
            case 1 -> addBook();
            case 2 -> borrowBook();
            case 3 -> returnBook();
            case 4 -> exitLibrary();
```

```
                default -> System.out.println("❌ Invalid choice. Please try again.");
            }
        }
    }

    // ✅ Preload Books into Library
    private static void preloadBooks() {
        library.put("Harry Potter", new Book(nextBookId++, "Harry Potter", "J.K. Rowling", 5));
        library.put("The Hobbit", new Book(nextBookId++, "The Hobbit", "J.R.R. Tolkien", 3));
        library.put("1984", new Book(nextBookId++, "1984", "George Orwell", 4));
        library.put("The Great Gatsby", new Book(nextBookId++, "The Great Gatsby", "F. Scott
Fitzgerald", 2));
    }

    // ✅ Display Menu
    private static void displayMenu() {
        System.out.println("\n📌 Menu:");
        System.out.println("1️⃣ Add Books");
        System.out.println("2️⃣ Borrow Books");
        System.out.println("3️⃣ Return Books");
        System.out.println("4️⃣ Exit");
    }

    // ✅ Get User Choice with Validation
    private static int getChoice() {
        System.out.print("👉 Enter your choice: ");
        while (!scanner.hasNextInt()) {
            System.out.println("❌ Invalid input. Please enter a number.");
            scanner.next();
        }
        return scanner.nextInt();
    }

    // ✅ Add Book
    private static void addBook() {
        scanner.nextLine(); // Consume newline
        System.out.print("📖 Enter book title: ");
        String title = scanner.nextLine().trim();

        System.out.print("✍ Enter author: ");
        String author = scanner.nextLine().trim();

        System.out.print("📦 Enter quantity: ");
        int quantity = getValidQuantity();
        if (library.containsKey(title)) {
            library.get(title).addQuantity(quantity);
            System.out.println("✅ Book quantity updated successfully!");
        } else {
            int id = nextBookId++; // Auto-generate ID
            library.put(title, new Book(id, title, author, quantity));
            System.out.println("✅ New book added to the library with ID: " + id);
        }
    }

    // ✅ Borrow Book
    private static void borrowBook() {
        displayBooks(library); // Show all books
        scanner.nextLine();
        System.out.print("📖 Enter the book ID or title to borrow: ");
        String input = scanner.nextLine().trim();

        Book book = findBook(input, library);
        if (book != null) {
            System.out.print("📦 Enter quantity to borrow: ");
            int quantityToBorrow = getValidQuantity();

            if (book.getQuantity() >= quantityToBorrow) {
                book.subtractQuantity(quantityToBorrow);

                // Add to borrowed books
                if (borrowedBooks.containsKey(book.getTitle())) {
                    borrowedBooks.get(book.getTitle()).addQuantity(quantityToBorrow);
                } else {
                    borrowedBooks.put(book.getTitle(),
                            new Book(book.getId(), book.getTitle(), book.getAuthor(), quantityToBorrow));
                }

                System.out
                        .println("✅ You have borrowed " + quantityToBorrow + " copies of \"" +
book.getTitle() + "\".");
            } else {
                System.out.println("❌ Not enough copies available.");
            }
        } else {
            System.out.println("❌ Book not found in the library.");
        }
    }
    // ✅ Return Book
    private static void returnBook() {
        if (borrowedBooks.isEmpty()) {
            System.out.println("❌ You have not borrowed any books.");
            return;
        }

        displayBooks(borrowedBooks); // Show only borrowed books
        scanner.nextLine();
        System.out.print("📖 Enter the book ID or title to return: ");
        String input = scanner.nextLine().trim();

        Book book = findBook(input, borrowedBooks);
        if (book != null) {
            System.out.print("📦 Enter quantity to return: ");
            int quantityToReturn = getValidQuantity();

            if (book.getQuantity() >= quantityToReturn) {
                book.subtractQuantity(quantityToReturn);

                // Return to library
                library.get(book.getTitle()).addQuantity(quantityToReturn);

                // Remove from borrowed books if quantity is zero
                if (book.getQuantity() == 0) {
                    borrowedBooks.remove(book.getTitle());
                }

                System.out
                        .println("✅ You have returned " + quantityToReturn + " copies of \"" +
book.getTitle() + "\".");
            } else {
                System.out.println("❌ You cannot return more than you borrowed.");
            }
        } else {
            System.out.println("❌ This book does not belong to your borrowed list.");
        }
    }

    // ✅ Display Books (Generic for both library and borrowed books)
    private static void displayBooks(Map<String, Book> books) {
        System.out.println("\n📚 Available Books:");
        System.out.printf("%-10s | %-25s | %-20s | %-10s\n", "ID", "Title", "Author", "Quantity");
        System.out.println("-------------------------------------------------------------------");
        for (Book book : books.values()) {
            System.out.printf("%-10d | %-25s | %-20s | %-10d\n", book.getId(), book.getTitle(),
book.getAuthor(),
                    book.getQuantity());
        }
    }

    // ✅ Find Book by ID or Title in a given map
    private static Book findBook(String input, Map<String, Book> books) {
        // Check if input is a number (ID)
        try {
            int id = Integer.parseInt(input);
            for (Book book : books.values()) {
                if (book.getId() == id) {
                    return book;
                }
            }
        } catch (NumberFormatException e) {
            // Input is not a number, treat as title
            return books.get(input);
        }
        return null; // Book not found
    }

    // ✅ Get Valid Quantity Input
    private static int getValidQuantity() {
        while (!scanner.hasNextInt()) {
            System.out.println("❌ Invalid input. Please enter a number.");
            scanner.next();
        }
        return scanner.nextInt();
    }

    // ✅ Exit Library
    private static void exitLibrary() {
        System.out.println("👋 Thank you for using the Library System! Goodbye!");
        System.exit(0);
    }

    // 📌 Book Class
    private static class Book {
        private final int id;
        private final String title;
        private final String author;
        private int quantity;

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

        public int getId() {
            return id;
        }

        public String getTitle() {
            return title;
        }

        public String getAuthor() {
            return author;
        }

        public int getQuantity() {
            return quantity;
        }

        public void addQuantity(int quantity) {
            this.quantity += quantity;
        }

        public void subtractQuantity(int quantity) {
            this.quantity -= quantity;
        }
    }
}
```

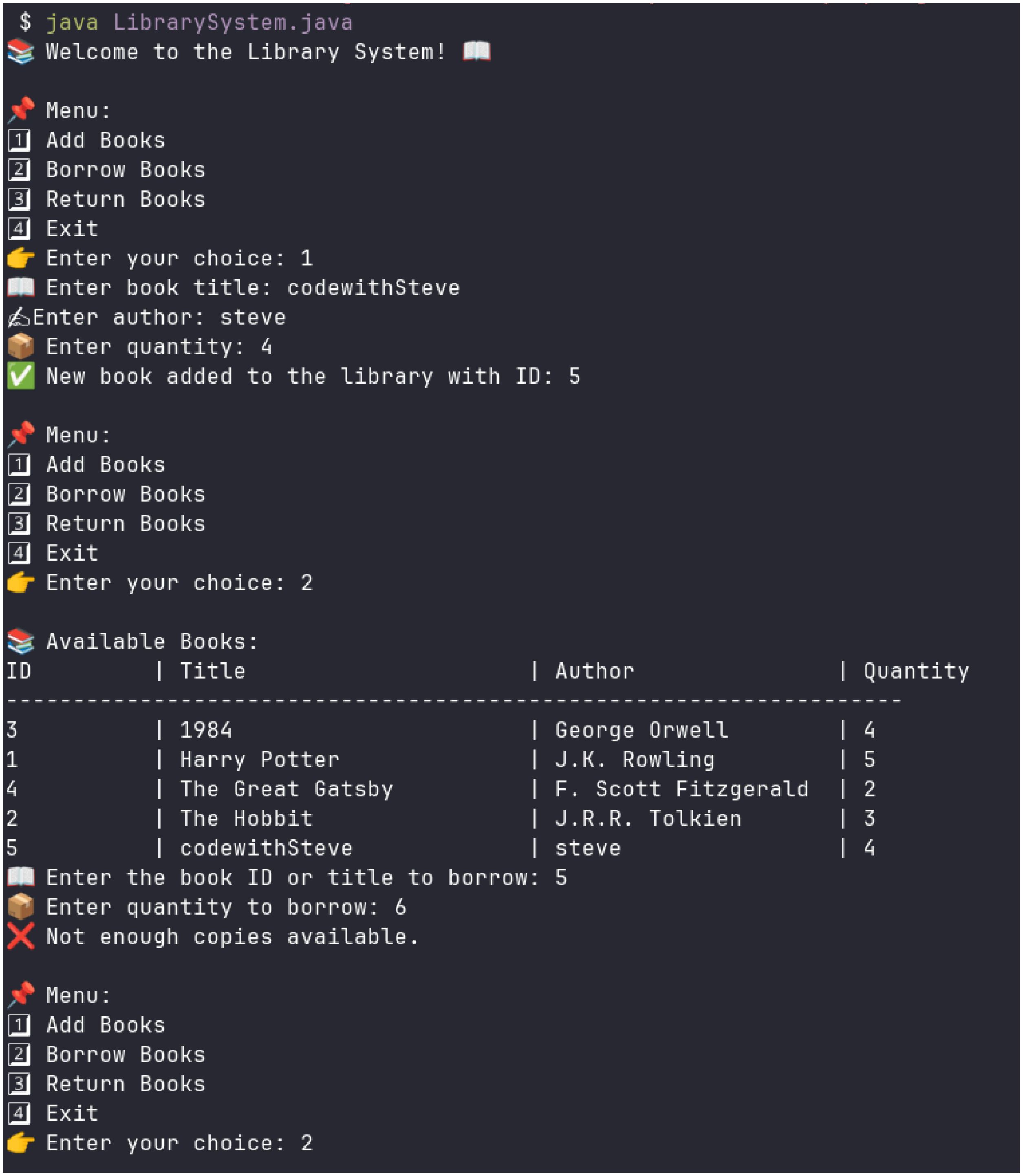
ScreenShots



```
ID         | Title                     | Author               | Quantity
------------------------------------------------------------------
3          | 1984                      | George Orwell        | 4
1          | Harry Potter              | J.K. Rowling         | 5
4          | The Great Gatsby          | F. Scott Fitzgerald  | 2
2          | The Hobbit                | J.R.R. Tolkien       | 3
5          | codewithSteve             | steve                | 4
📖 Enter the book ID or title to borrow: 5
📦 Enter quantity to borrow: 2
✅ You have borrowed 2 copies of "codewithSteve".

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 3

📚 Available Books:
ID         | Title                     | Author               | Quantity
------------------------------------------------------------------
5          | codewithSteve             | steve                | 2
📖 Enter the book ID or title to return: 5
📦 Enter quantity to return: 2
✅ You have returned 2 copies of "codewithSteve".

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 3
❌ You have not borrowed any books.

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 4
👋 Thank you for using the Library System! Goodbye!
```



```
📚 Available Books:
ID         | Title                     | Author               | Quantity
-------------------------------------------------------------------
3          | 1984                      | George Orwell        | 4
1          | Harry Potter              | J.K. Rowling         | 5
4          | The Great Gatsby          | F. Scott Fitzgerald  | 2
2          | The Hobbit                | J.R.R. Tolkien       | 3
5          | codewithSteve             | steve                | 4
📖 Enter the book ID or title to borrow: 5
📦 Enter quantity to borrow: 2
✅ You have borrowed 2 copies of "codewithSteve".

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 3

📚 Available Books:
ID         | Title                     | Author               | Quantity
-------------------------------------------------------------------
5          | codewithSteve             | steve                | 2
📖 Enter the book ID or title to return: 5
📦 Enter quantity to return: 2
✅ You have returned 2 copies of "codewithSteve".

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 3
❌ You have not borrowed any books.

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: █
```



```
$ java LibrarySystem.java
📚 Welcome to the Library System! 📖

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 1
📖 Enter book title: codewithSteve
✍Enter author: steve
📦 Enter quantity: 4
✅ New book added to the library with ID: 5

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 2

📚 Available Books:
ID         | Title                     | Author               | Quantity
-------------------------------------------------------------------
3          | 1984                      | George Orwell        | 4
1          | Harry Potter              | J.K. Rowling         | 5
4          | The Great Gatsby          | F. Scott Fitzgerald  | 2
2          | The Hobbit                | J.R.R. Tolkien       | 3
5          | codewithSteve             | steve                | 4
📖 Enter the book ID or title to borrow: 5
📦 Enter quantity to borrow: 6
❌ Not enough copies available.

📌 Menu:
1️⃣ Add Books
2️⃣ Borrow Books
3️⃣ Return Books
4️⃣ Exit
👉 Enter your choice: 2
```