Exercise: Handling Promise States - Borrowing a Book

Objective: Create TypeScript code to simulate borrowing a book from a library and demonstrate handling different Promise states.

Outline:

1. **Define TypeScript Code:**
    - Create library**.ts** in a code editor and write the TS code for the **borrowBook(bookId:number)** function and the two usages one to fulfill the Promise and one to reject the Promise
2. **Code the borrowBook Function:**
    - Define the **borrowBook** function to simulate borrowing a book from a library.
    - The **borrowBook(bookId:number)** function returns a Promise that simulates borrowing a book from a library. It resolves if the book is available and rejects if the book is not available.
    - Assume that you have array of availableBooks = [1,2,3].
    - Based on the bookId passed, if the bookId exists in the array resolve with a message `Book ${bookId} is successfully borrowed.` otherwise reject with `Book ${bookId} is not available in the library.`.
3. **Handle a Fulfilled Promise Usage 1:**
    - Create usage 1 to demonstrate handling a fulfilled Promise using **.then()**.
    - Borrow a book that is available (e.g., **bookIdToBorrow = 2**) and log the success message in the **.then()** handler.
4. **Handle a Rejected Promise Usage 2:**
    - Create Usage 2 code to demonstrate handling a rejected Promise using **.catch()**.
    - Borrow a book that is not available (e.g., **bookIdToBorrowInvalid = 4**) and log the error message in the **.catch()** handler.