ES6 Exercise: Building a Library Management System With ES6

In this exercise, you will build a Library Management System using JavaScript and ES6 features. The system will allow users to add books, borrow books, return books, and display information about the books in the library. You will use let and const, variable scope, shadowing variables, arrow functions, arrow functions as parameters, using 'this' within arrow functions, template literals, spread operator, ES6 classes, declaring classes, declaring instance methods, accessor methods, static methods, and inheritance with classes.

**Step 1: Create a Book class**

1. Create a Book class using ES6 syntax. The class should have the following properties:
   1. title (string): the title of the book.
   2. author (string): the author of the book.
   3. isBorrowed (boolean): indicates whether the book is currently borrowed or not
2. The class should also have the following methods:
   1. borrowBook(): sets the isBorrowed property to true.
   2. returnBook(): sets the isBorrowed property to false.
   3. displayInfo(): displays the book's title, author, and the borrowing status.

**Step 2: Create a Library class**

1. Create a Library class using ES6 syntax. The class should have the following properties:
   1. name (string): the name of the library.
   2. books (array): an array to store the books in the library.
2. The class should also have the following methods:
   1. addBook(book): adds a book to the library's collection.
   2. borrowBook(title): borrows a book from the library by its title. If the book is available (not already borrowed), set its isBorrowed property to true.
   3. returnBook(title): returns a borrowed book to the library by its title. If the book is borrowed, set its isBorrowed property to false.
   4. displayBooks(): displays information about all the books in the library using the displayInfo() method of each book.

**Step 3: Implement the Library Management System**

1. Create an instance of the Library class. Use the addBook() method to add a few books to the library.
2. Use template literals to display a welcome message, including the library's name.
3. Use arrow functions as parameters to implement the following functionality:
   1. Prompt the user to enter a book title to borrow and call the borrowBook() method.
   2. Prompt the user to enter a book title to return and call the returnBook() method.
4. Use the spread operator to display a list of books in the library after each operation using the displayBooks() method.

**Step 4: Implement Inheritance**

1. Create a new class called ReferenceBook that inherits from the Book class. The ReferenceBook class should have an additional property called category (string) to indicate the category of the reference book.
2. Override the displayInfo() method in the ReferenceBook class to include the category information.
3. Create an instance of the ReferenceBook class and add it to the library using the addBook() method.
4. Display the updated list of books in the library using the displayBooks() method.