JS Advanced Exam

Problem 3. Unit Testing

Your Task

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **library**, which represents an object. You may use the following code as a template:

```
describe("Tests …", function() {
    describe("TODO …", function() {
        it("TODO …", function() {
            // TODO: …
        });
     });
     // TODO: …
});
```

The object that should have the following functionality:

- **`calcPriceOfBook (nameOfBook, year)`** - A function that accepts a string and a number:
    - The function calculates the price of the book depending on the **year** of publication
    - The standard price of the book is 20 BGN
    - If the **year** of publication is **less** than or **equal** to **1980**, there is a **50%** percent discount from the standard price
    - The function calculated price of the book and **return**: **`` `Price of {nameOfBook} is {price}` ``**
    - You need to validate the input, if **`nameOfBook`** is not a string, or the **`year`** is not an **integer** number, **throw** an error: "**`Invalid input`**"

- **`findBook (booksArr, desiredBook)`**- A function that accepts an array and string:
    - The array includes all available **books** in the library ([**"Troy", "Life Style", "Torronto", etc.**])
    - If the length of the **booksArr** array is zero, **throw** an error in the following format: "**`No books currently available`**"
    - The function checks whether the submitted string **desiredBook** is present in the array **`booksArr`**.
    - If present in the array, the function **return**: "**`We found the book you want.`**"
    - Otherwise the function **return**: "**`The book you are looking for is not here!`**"
    - There is no need for validation for the input, you will always be given an array and string

- **arrangeTheBooks (countBooks)** - A function accept a number:
    - You need to validate the input, if the **countBooks** is not an **integer** number, or is a negative number, **throw** an error: "**Invalid input**"
    - The library has 5 **shelves**, each shelf can hold 8 books. Distribute the books on the shelves
    - If all the books are arranged on the shelves, **return**: "**Great job, the books are arranged.**"
    - Otherwise, if no space has been reached, **return**: "**Insufficient space, more shelves need to be purchased.**"

JS Code

To ease you in the process, you are provided with an implementation that meets all of the specification requirements for the **library** object:

library.js

```
const library = {
    calcPriceOfBook(nameOfBook, year) {

        let price = 20;
        if (typeof nameOfBook != "string" || !Number.isInteger(year)) {
            throw new Error("Invalid input");
        } else if (year <= 1980) {
            let total = price - (price * 0.5);
            return `Price of ${nameOfBook} is ${total.toFixed(2)}`;
        }
        return `Price of ${nameOfBook} is ${price.toFixed(2)}`;
    },

    findBook: function(booksArr, desiredBook) {
        if (booksArr.length == 0) {
            throw new Error("No books currently available");
        } else if (booksArr.find(e => e == desiredBook)) {
            return "We found the book you want.";
        } else {
            return "The book you are looking for is not here!";
        }

    },
    arrangeTheBooks(countBooks) {
        const countShelves = 5;
        const availableSpace = countShelves * 8;

        if (!Number.isInteger(countBooks) || countBooks < 0) {
            throw new Error("Invalid input");
        } else if (availableSpace >= countBooks) {
            return "Great job, the books are arranged.";
        } else {
            return "Insufficient space, more shelves need to be purchased.";
        }
    }

};
```

Submission

Submit your tests inside a **`describe()`** statement, as shown above.