# Frontend assignment

## Description

John Doe works in the IT department of a library that uses paper records as a means to keep records of books, authors, and customers. One day, John's manager assigned John to develop a backend application that functions as a record system as well as a rental software.

## Instructions

**Task 1** - Implement component from given figma design

https://www.figma.com/file/EgUOn8iE9xfmFV9TRFjd28/Easy-Frontend-UI?node-id=0%3A1

**Acceptance criteria**
- Should follow all the design in Figma
- Should implement the id and class specified in Figma comment

**Task 2** - call /getTop3ReadBook to retrieve data back, and use the component created above to display it. Refer to Frame 1 & 2, and read the comments in it for details.
Please assume the backend API Url starts with http://localhost:8080

JSON response format for GET /getTop3ReadBook

```
[
    {
        "name": "book_name_value",
        "author": "author_value",
        "borrower": [
            "top_1_borrower_value",
            "top_2_borrower_value",
            "top_3_borrower_value"
        ]
    },
]
```

**Acceptance criteria**
- Should retrieve data from the API and bind to components successfully
- Should implement book-toggle event handling as per comment in Figma

**Task 3** - Create a Get country button to call to GET /getRandomCountry to get random country code when clicked and use the "country_code" as a query parameter to call /getTop3ReadBook?country_code=SG, then update the UI component with the latest data.

The page UI should show error message text if there's no data retrieved from /getTop3ReadBook endpoint. The error message text should be "No data found". Refer to Figma Frame 3 for this.

JSON response format for GET /getRandomCountry

```
{
    "country" : {
        "full_name": "country_name_value",
        "country_code": "country_code_value"
    }
}
```

**Acceptance criteria**
- Button should follow Figma design
- Button should call the 2 API (getRandomCountry and getTop3ReadBook) when clicked
- Upon API call success, render the new data on the page
- Upon API call failure, show the error message on the page