📕

Masai Library

Instructions

- You should use json-server for this.
- Use HTML, CSS and JS DOM to solve this question.
- Read entire question carefully for at-least 15 mins, understand it and code it.
- Use HTML, CSS and JS DOM to solve this question.
- You will have to also deploy both JSON server and app.
- Commit your code every hour with proper commit message to your repository (we will monitor every commit)

`Problem Statement`

- The task is to build a library app, where the admin can ADD, UPDATE and DELETE stocks and the users can borrow the books.
- The app should have two sections,
    - Admin Section
    - User Section
- In the home screen make two buttons for these sections, on clicking on the buttons move to that particular section.
- Details of each section are given below.

`Admin Section`

- Admin should be prompted by a login form.
- Admin need not register. Admin can directly log in with the following credentials.
    - email: `admin@gmail.com`
    - password: `masai`
- After successful login, the admin should be redirected to the admin dashboard.

`Admin Dashboard`

- This page should have a form that takes the following information
- When admin submits the form, post data to JSON server (POST request)
    - Book Cover Image (Use placeholder or dummy images)
    - Name of the book
    - Author of the book
    - Genre - select tag (Science, Fiction, History, Tech, Business)
    - Edition - select tag (2020, 2021, 2022)
    - Publisher
    - Cost

- When admin submits the form, post data to JSON server (POST request)
- Sample data

```
[ { "image_url": "https://m.media-
amazon.com/images/I/5165He67NEL.jpg", "book_name": "Harry Potter and
the Philosopher's Stone", "author": "J. K. Rowling", "genre":
"fiction", "edition": "2021", "publisher": "Bloomsbury", "cost": 1299,
"borrowed": false, } ]
```

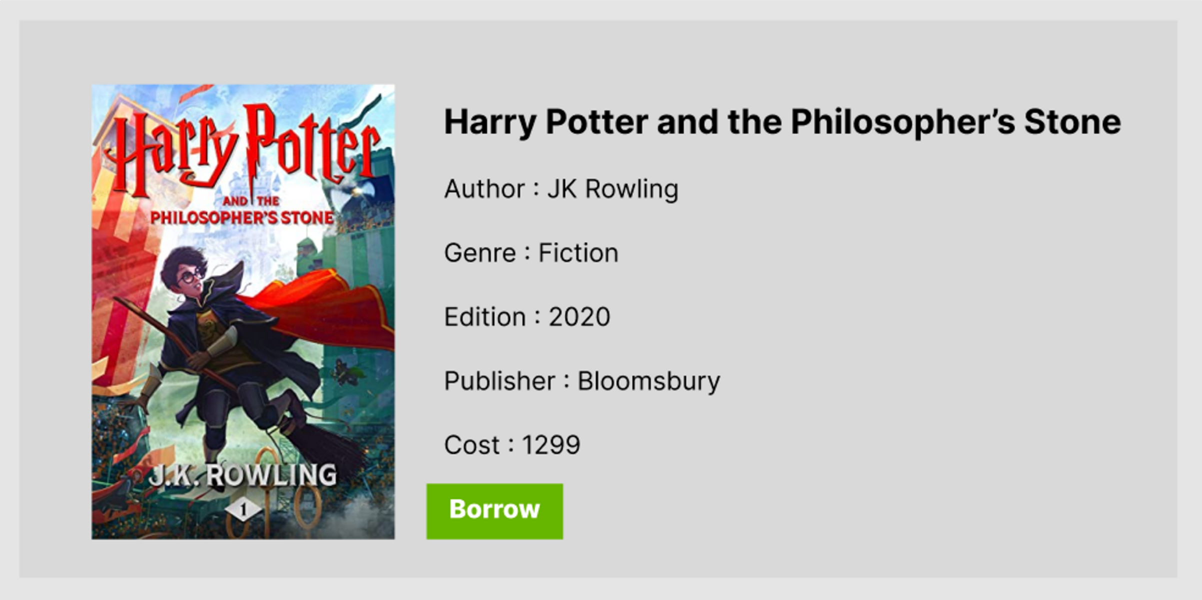
- Use a JSON server to manage the data (using local storage or dummy data will lead to disqualification)Rubrics / Criteria to be judged upon
- Fetch this data from the JSON server and display it in table format below the form.
- The table should contain the following fields as columns
    - Book Cover Image (Use placeholder or dummy images)
    - Name of the book
    - Author of the book
    - Genre - select tag (Science, Fiction, History, Tech, Business)
    - Edition - select tag (2020, 2021, 2022)
    - Publisher
    - Cost
    - Edit
    - Delete
- When admin clicks on edit, admin should be able to edit/change details of the book in JSON data (you can use a modal for this) and display updated data in the table.
- When clicking on Delete, delete that particular entry from JSON data.
- Also, the admin should be able to see all the users who have signed up in form of tabular data in separate route with following
    - Username
    - Email
    - Time stamp.

`User Section`

- The user should be able to signup with the following details
  - Username
  - Email
  - Password
- Once the user submits the input data, store that data in local storage.
- Once they are successfully signed up, the user should be redirected to the login tab.
- When a user fills in credentials, match the entered data with the local storage data you stored while signing up. Display an alert - if login is a success, show a message in the alert box as `login successful`, else show a message as `login failed`
- After successful login, the user should be redirected to the books page.

Books Page

- Fetch all the data of books from the JSON server and display them in form of cards.



Harry Potter and the Philosopher's Stone

Author : JK Rowling

Genre : Fiction

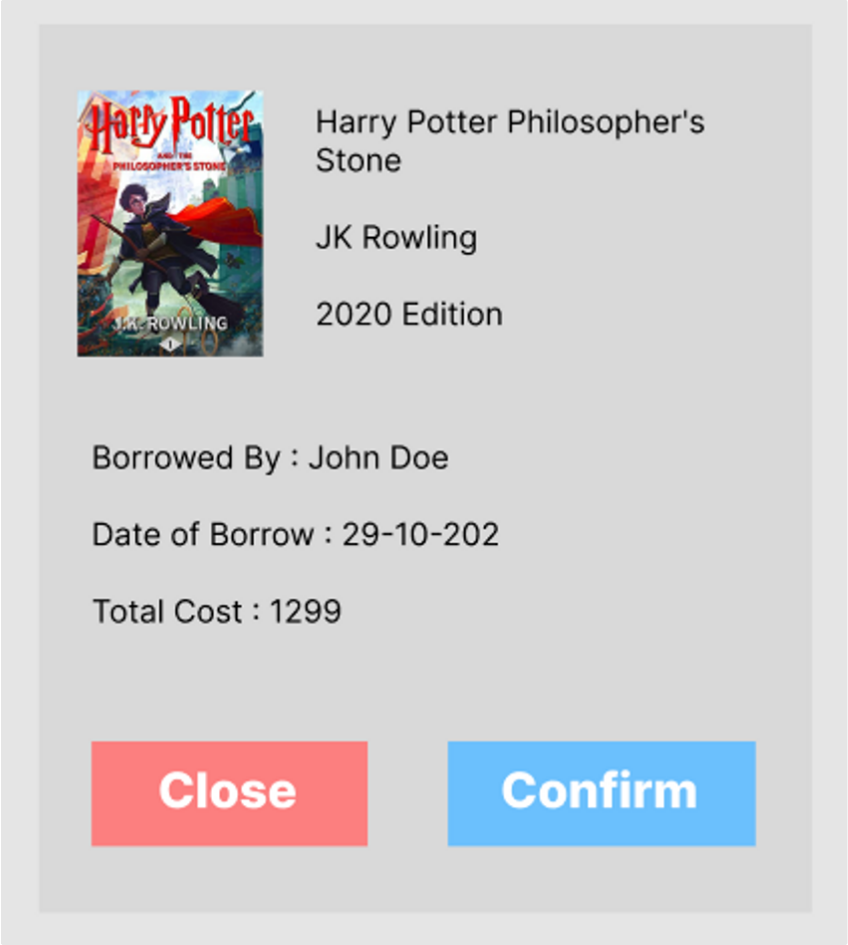
Edition : 2020

Publisher : Bloomsbury

Cost : 1299

Reference Image - 1

- This page should have a filter and sort functionality
    - Filter based on Genre (use the select tag with the onChange event)
        - Science
        - Fiction
        - History
        - Tech
        - Business
    - Sort based on the cost of the books
        - Low to High
        - High to Low
- There should be a borrow button appended to each of the books.
- When clicking on the Borrow button, open up a modal that shows the summary of the bill along with book details.



Harry Potter Philosopher's Stone

JK Rowling

2020 Edition

Borrowed By : John Doe

Date of Borrow : 29-10-202

Total Cost : 1299

Reference Image - 2

- Also after borrowing successfully, the status of the book should be changed to true on the admin panel.
- User should be able to see all borrowed books in separate page in tabular format.
- Please note that this is multi user application, each and every user signups should have separate borrowed books data associated to that account.

Note:

- Maintain flow of app as mentioned.
- Error messages should be shown
- Use loaders.
- Good designs will fetch bonus marks.
- Submitting local host links for mock server will lead to disqualification.

Submission

- Please submit deployed link and Github link of code.
- ***Deploy JSON server too. (submitting localhost link will lead to disqualification)
- Please double check if deployed version works or not (run deployed version on your laptop and then submit it)
- **Push your code into `masai-repo`, don't submit personal repo links (submitting personal repo links will lead to disqualification)**
- Any issues in deployed link will be considered null and void.
- Please verify your submissions.
- Make sure you follow all instructions carefully.
- Submit before deadline.

Rubrics / Criteria to be judged upon

- CRUD Operations***

- JSON server***
- JSON server deployment.
- Fetch
- Promises
- Sorting and Filtering.
- Good UI
- Code cleanliness, folder structure and git hygeine will be looked into.
- Usage of Javascript (using react will lead to disqualification)