NAME: LIBRARY MANAGEMENT SYSTEM

# Features:

A library consists of different category of books like fiction, art, cooking, comic, etc. Each category can contain some extra attributes which can be added later at some point of time. Each book consists of attributes like unique book id, title, author name and price.

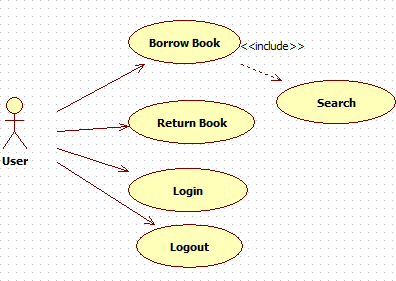
Application has to keep track of book issue date, return date, which user has borrowed the book.

Users of this application can perform the following:

* Login
* Logout
* Borrow a book
  + Select a book
    - User can search a book based on category, and select the book by providing “book Id” from the displayed list.
    - User can directly select a book if “book Id” is known.
  + Once user selects a book, he/she can borrow a book.
* Return borrowed book.

***Note: User can return only books which he/she has borrowed.***

# Diagram:



**Functionalities**:

1. Initially user is given choices
   1. Login
   2. Logout
   3. Exit
2. User should login first
3. After successful login, user should be given choice to either
   1. borrow a book or
   2. return a book
4. If user chooses to borrow a book, then
   1. user should be able to view all categories
   2. User selects a category id
   3. Books belonging to that category should be displayed
   4. User selects a book id, which the user wants to borrow
   5. System should check whether selected book does exist or not. If the book is exists then it should be checked whether the book is available for loan or not. If so, then user can successfully borrow a book and this borrowing information should be recorded. Otherwise this book will not be possible to be borrowed by the user
5. If user chooses to return a book, then
   1. System should check whether it is the same user who borrowed the exact same book or not
   2. If so, then the book should be returned and the borrowing information saved in step 4 should be removed regarding the user and the book should be made free/available for further borrowing
   3. If the book that user claims to have borrowed was not borrowed by the user at all, then the book can’t be returned
6. If user chooses to logout then user should be able to logout
7. If user chooses to exit, then the application should terminate

# Implementation:

Create an application **library-management-system using create-react-app tool.** The application should contain following folders and files:

* All methods to interact with RESTful API application must be created in a service class that should be placed inside “services” folder. Service class must follow singleton pattern.
* All components must be present in the “components” folder and container components should be in separate “containers” folder
* Any images, local data (if any) should be placed inside “assets” folder
* Axios should be used to interact with RESTful API application
* Entire application should be Hook based. In case error handling, if Error Boundary pattern is used, then that component can be a Class component
* The application must be designed using Material UI framework
* Application must use RWD
  + Must be responsive for at least 3 breakpoints (Small, Medium, Large)
  + Preferable Usage of Flex Layout (CSS)
* For form handling Formik used and for validation it can be merged with “Joi” or “Yup” package
* The application must be PWA ready
* Should use “boostrap” CSS for styling if necessary
* Usage of Sass / Scss for styling
* ESLint must be configured for the application
* Minimum 3 components must be tested using “jest/enzyme”
* The application must be built by all the team members using Git
* Should use Jenkins for CI/CD
* The application must follow DDD
* The application must be tested using Lighthouse for the performance and PWA readiness (Lighthouse report >= 80 (SEO & Accessibility))