# CloseReading

IDB Group 13: Sumaya Al-Bedaiwi, Peyton Ausburn, Adrian Sanchez, Peter Hwang, & Niyati Prabhu

## Motivation:

CloseReading is a website that aims to boost people's utilization of local libraries by connecting them with libraries which contain the books they seek. People of all ages and walks of life can benefit from this resource - lending books from libraries is a sustainable, affordable practice that should be encouraged.

Under the books tab, users can discover new books: if a specific book is clicked, the website displays information about the book, such as synopsis, author, page count, publisher, and date published. Furthermore, the user can read reviews, buy books, and locate the book at a particular library. The authors tab displays a view of popular authors - when clicked, they will allow users to read a short background on the author and explore any of the books they've written. Lastly, the library tab populates different libraries that are close to the user - when a library is clicked, it will connect users with more information about the library and availability of a certain book at the library.

## User Stories (personal):

- Create an aesthetically-pleasing splash page on the home screen of the app - display cards that link to the books, authors, and libraries tabs respectively. Display a book-related image as a background above the card navigation below.
- Create a global navigation bar - allows users to select a tab at the top of the screen in order to navigate between books, authors, libraries, about, and home.
- Create a page each for books, authors, and libraries which display three instances of the respective model. The cards should show a preview of at least five attributes that will later be sortable or filterable by.
- Create a detail view page for each instance of each model - clicking on any card should route the user to a new screen that details more information about the instance. It should include media such as a picture, expanded details of the attributes, etc.
- Fill in an About page to display names, photos, and bios for each team member, as well as dynamic tracking of Gitlab commits and issues.

## User Stories (from customer):

- Customer has not yet provided user stories for our team to fulfill.

## RESTful API:

Our API documentation was created on Postman - it defines GET methods to retrieve data for all books, all authors, and all libraries. There are also GET methods defined for retrieving results that filter data for each model by an attribute. The API methods have not yet been implemented, but we plan on implementing it as part of the backend in the next phase of the project.

## Models:

Libraries - This model provides information about a specific library institution.

- Attributes to sort and filter by: location, name, size of collection, library type, rating.
- Rich media: Images of library, google map of library, yelp ratings, nearby libraries

Books - This model provides information about a specific book.

- Attributes to sort and filter by: title, author, genre, page count, publisher, publish year, NYT best-seller status.
- Rich media: Book cover image, book reviews, purchase link, similar books

Authors - This model provides information about a specific author.

- Attributes to sort and filter by: Name, age, ethnicity, gender, number of publications.
- Rich media: Headshot, author bio, average rating, similar authors

## Tools:

- React: JavaScript library for building user interfaces, powers the front-end
- React-Bootstrap: CSS framework built on React, simplifies integration of the frameworks
- React Router: library for routing in React, enable navigation between pages
- NameCheap: domain name registrar, allows purchasing and managing domain names
- AWS Amplify: platform for deploying and hosting web applications, supports continuous integration and continuous deployment
- Visual Studio Code: code editor
- GitLab: git software that allows for managing version control and continuous integration/continuous deployment
- Postman: API platform for documenting, building, and using APIs
- Node Package Manager (NPM): manages installation of node packages
- Microsoft Teams: online communication platform

## Data Sources:

- WorldCat Library Profiles API: to retrieve information about a particular library
- WorldCat New Titles API: retrieve all the new additions to a particular library’s collection
- WorldCat Discovery API: to find all libraries that hold a particular publication

- Google Books API: to retrieve data and reviews on popular and recent books
- The NYT Books API: to retrieve a book’s status as a NYT Best-Seller
- Amazon API: to retrieve a link to purchase a particular book on Amazon
- Wikipedia API: to retrieve popular authors and  information on a particular author
- GitLab API: to retrieve live data on the number of issues and commits per team member
- Google Maps API: to display a Google map view for a particular library
- Yelp API: to retrieve and display reviews of a library

## Hosting:

Our website is hosted on AWS Amplify at the domain closereading.me. The domain name was obtained through Namecheap, and from there we utilized CNAME records to transfer the DNS ownership to AWS Amplify.

## Challenges:

There were several challenges for our team as we began developing this web project:

- Many APIs that we initially brainstormed were not able to be used (e.g. Goodreads)
- There were issues initially setting up the React app with the Gitlab repository and with the AWS Amplify host.
- We were unsure of which AWS host to use for this web app, since there are dozens of options.
- There were issues with transferring DNS ownership to AWS, for which we needed to leverage a TA's help and Namecheap's live chat feature.