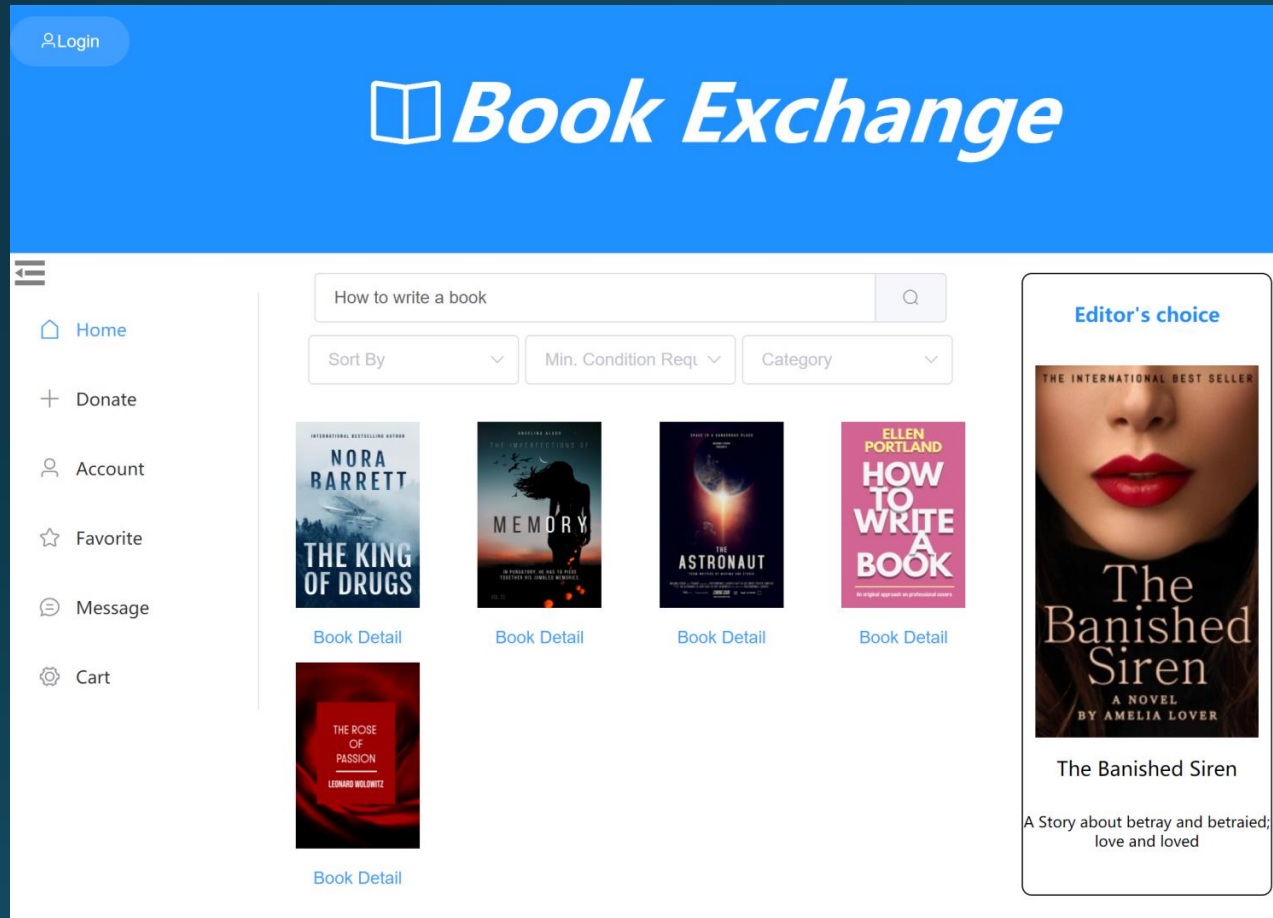


# Book Exchange

*A web service for exchanging second-hand books*

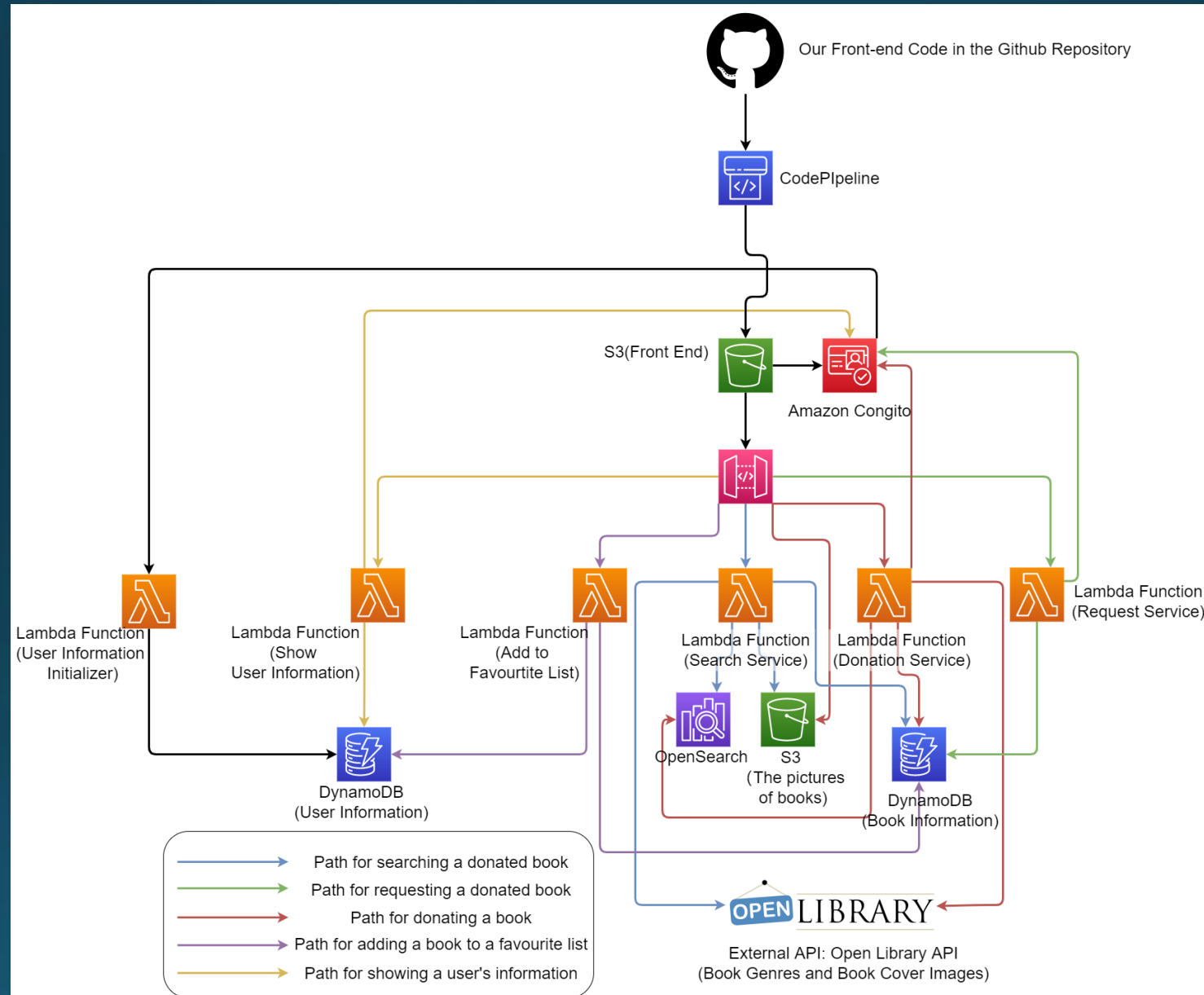


## Motivation

- ▶ A platform encourages people to exchange books
- ▶ Save money to buy new books
- ▶ The more books you donate, the more books you can request from others
- ▶ Welcome users with similar interests to contact each other


Group member: Dantong Zhu(dz2451), Tianhang Cui(tc3158), Anni Chen(ac4779), Zhenrui Chen(zc2569)

# Structure Overview



# Front-End Overview

Login/Registration Panel



**Username**

**Password**

Login

Register


☒ Remember me


Cancel


Forgot [password?](#)


Function  
Panel


Login


 **Book Exchange**


 Home


 Donate


 Account


 Favorite


 Message

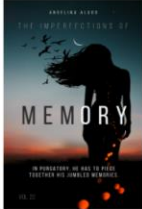
 Cart


Sort By 

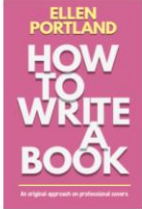
Min. Condition Requ 


Category 

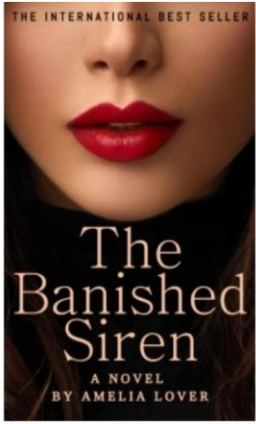
  
Book Detail

  
Book Detail

  
Book Detail

  
Book Detail

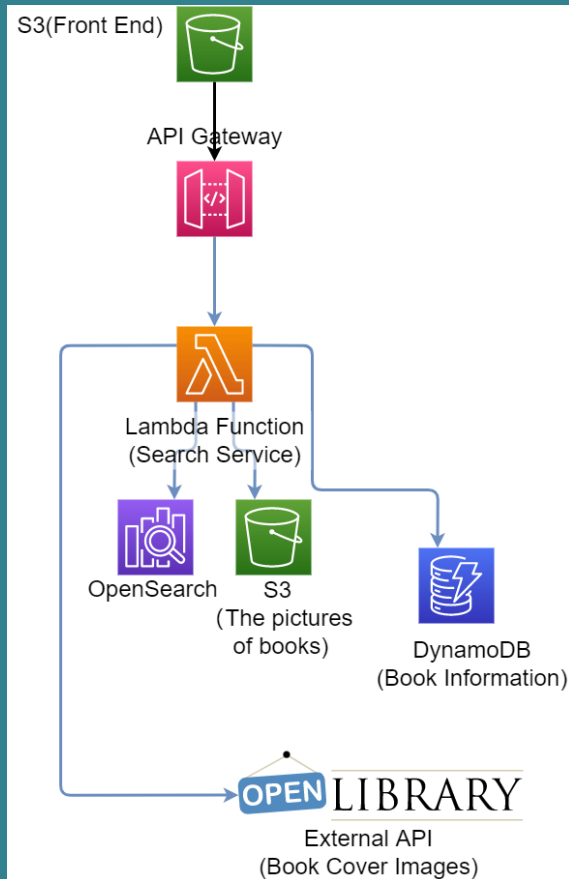
  
Book Detail

  
The Banished Siren  
A Story about betray and betrayed;  
love and loved

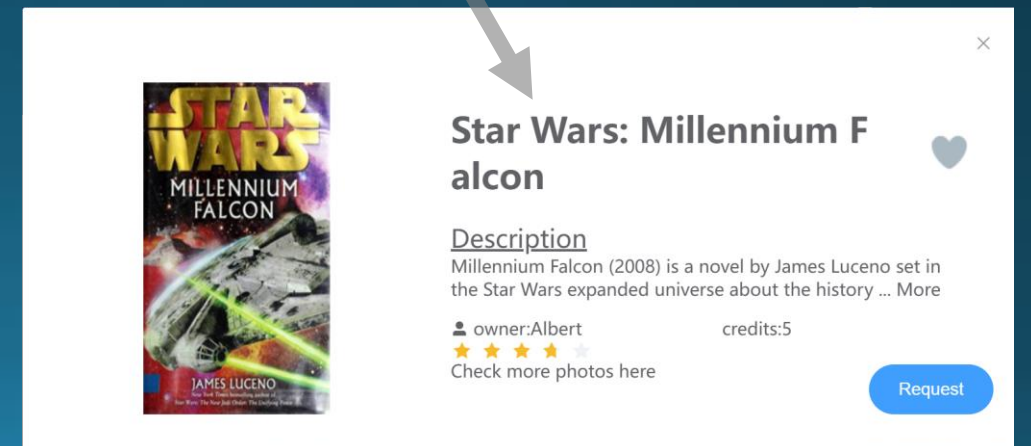
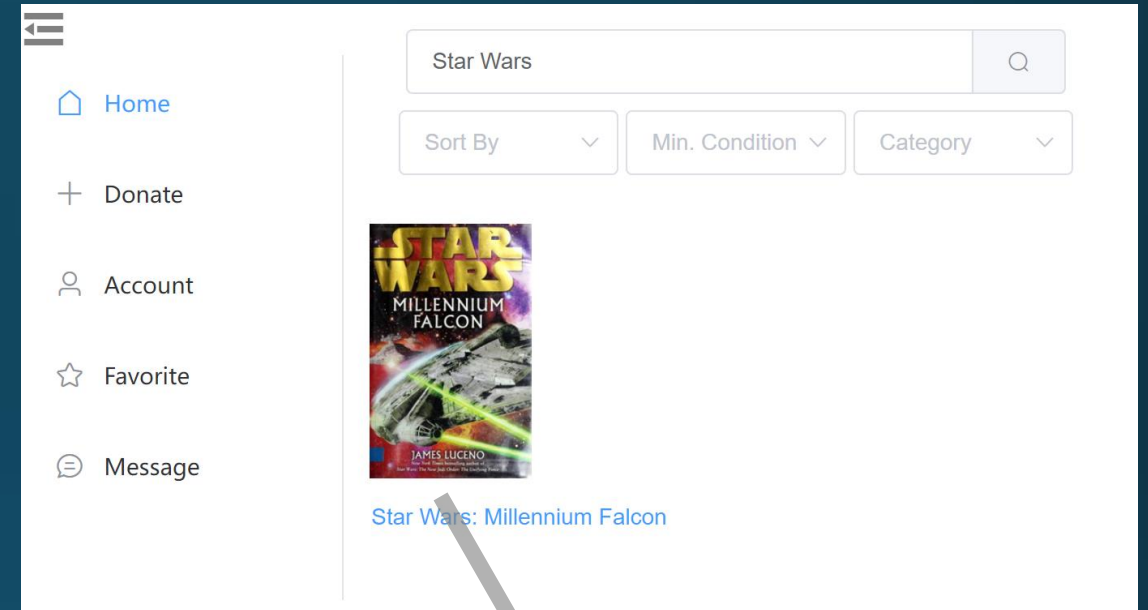
Editor's choice

# Search for a second-hand book

## Implementation Detail

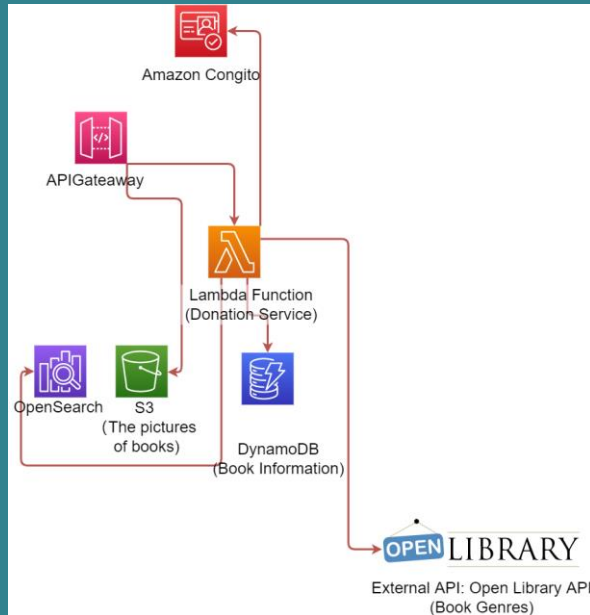


- Using API Gateway to pass the query information from frontend input to lambda function
- Using a lambda function to query corresponding available donation books from DynamoDB through OpenSearch.
- The lambda function also automatically get the book cover by using Open Library API and upload pic from S3
- Frontend get the response from the image and display it under the search bar



# Donate a second-hand book

## Implementation Detail



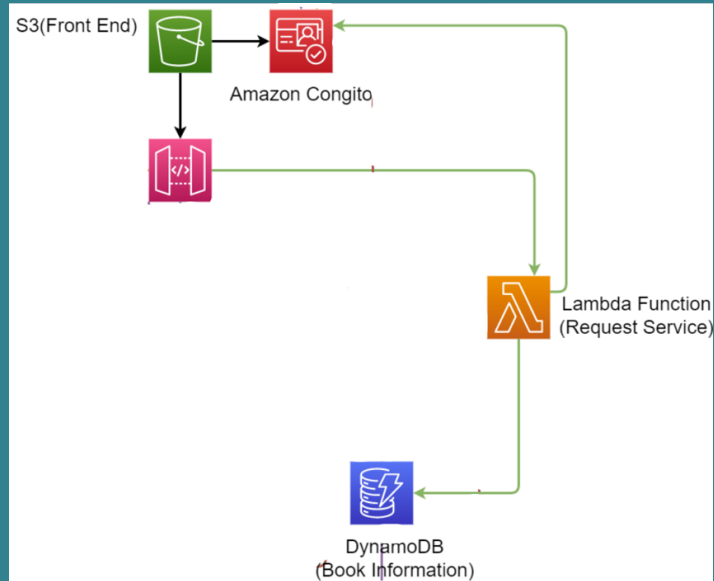
- Using API Gateway to pass the photos directly to a S3 bucket
- Using a Lambda Function to add the donation information to DynamoDB and OpenSearch.
- The lambda function automatically generates the genre information for the donated book by using Open Library API. This can help other users find the book they want by genre.

The screenshot shows the 'Donate' panel of a mobile application. On the left is a navigation menu with options: Home, Donate (highlighted), Account, Favorite, Message, and Cart. The main content area is titled 'Name of your donated book:' and includes a text input field with the example 'Ex. Harry Potter'. Below this is a 'Credit Required for requesting this book:' section with a numeric input set to '5'. The 'Upload a photo:' section features a dashed box with a plus sign for image upload. The 'Book condition:' section has five radio buttons: 'Poor', 'Fair', 'Good' (selected), 'Very good', and 'As New'. At the bottom is a large blue 'Donate' button with a cloud upload icon.

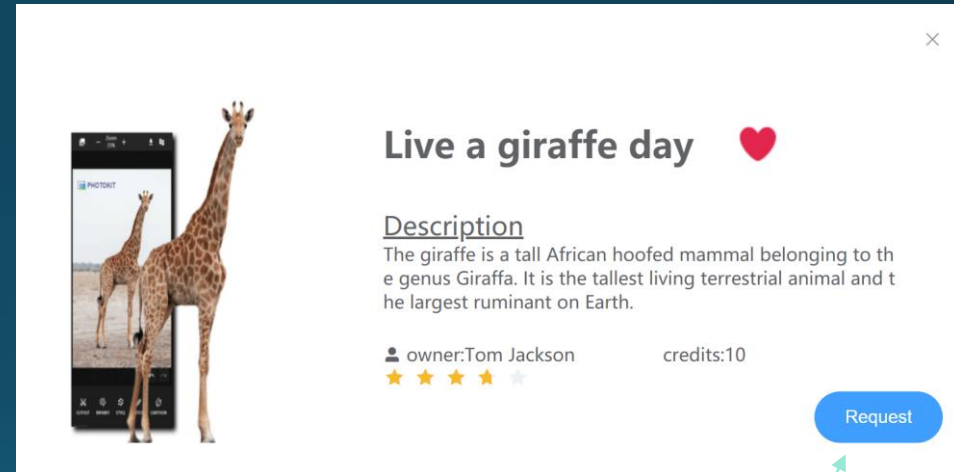
Donation Panel: The user can only donate a book when they have logged in.

# Request a second-hand book

## Implementation Detail



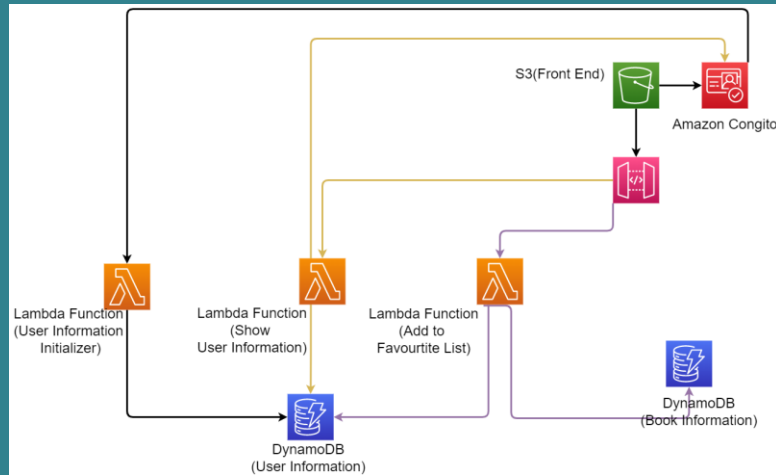
- Determine whether the user has enough credits to make request
- Once successful requested, update user history (add book\_id)
- Update book information to set status='unavailable', so it will no longer be searched



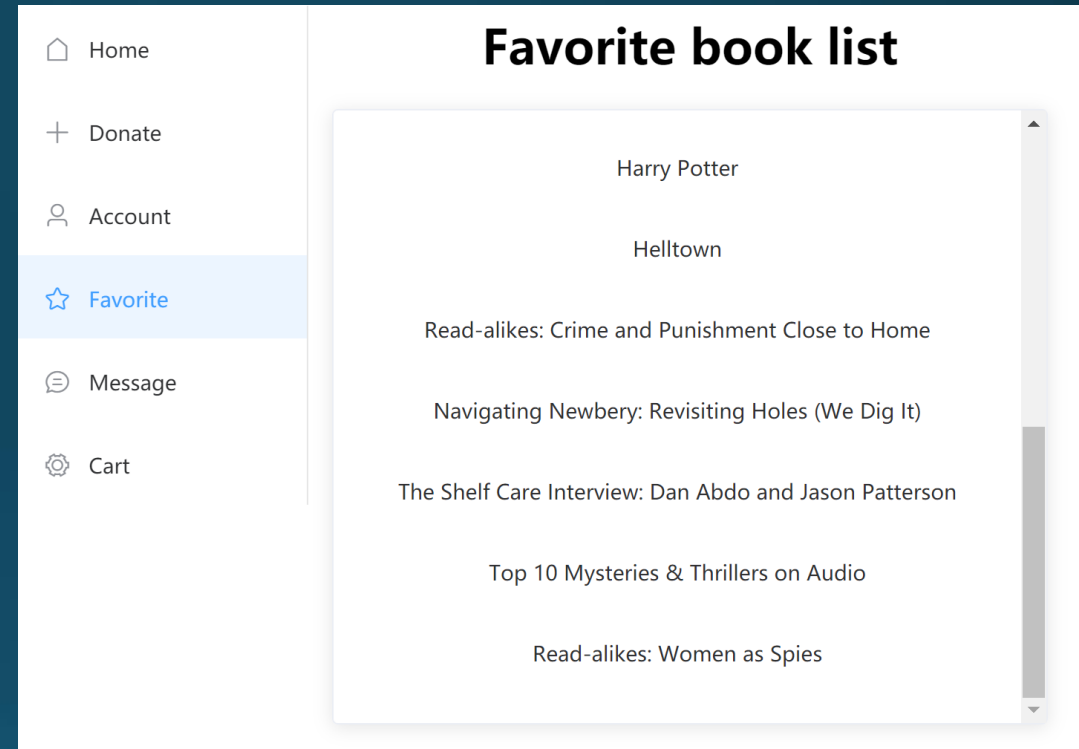
Click to request  
the book you  
want!

# Show user's info & add to favorite list

## Implementation Detail



- Every time the user selects a book they like, the system firstly will store this book's id in Dynamo DB(user information) corresponding to the user.
- When the user jumps to the favorite list page and want more information about these books, the system will search it in the DynamoDB(book info) by id
- When the user chooses to add a book to their favorite list, the count of favorite' for that book will increase by 1 and the book is removed when the user double click



You can add a book to your list here!



# Some Design Choices

Although we can link user's account and their information directly in Cognito (which is convenient and good for keeping user's information secure!), we decided to use **DynamoDB** to store the users' non-sensitive information, rather than storing them in **Cognito**.

- The attributes stored in Cognito has to be key-value(string) pair, not good for complicated data structures.
- We need to store many users' requested/donate books, this list could be large and Cognito is not scalable as the DynamoDB.
- The DynamoDB can handle query much more efficient than Cognito
- We realized this is the pattern often done in the industry after searching the online information

We does not use OpenSearch to store the availability of the book (i.e. already rent/available for request)

- We stores the genres, book name, and donation id to the OpenSearch to improve the performance of searching.
- Storing availability can help us filter out those unavailable books for search.
- However, this would requires us to update OpenSearch each time there is a book was requested.
- Given the current expected number of users (~10000 users in total) is not too large, it seems okay to get all the books fulfilled the search request and then filter out those unavailable.
- This design may need to be changed when the expected number of user become larger.