D424 – Software Engineering

Task 3

Capstone Proposal Project Name: Borrow My Bookshelf

Student Name: Chanel O'Reilly

Table of Contents

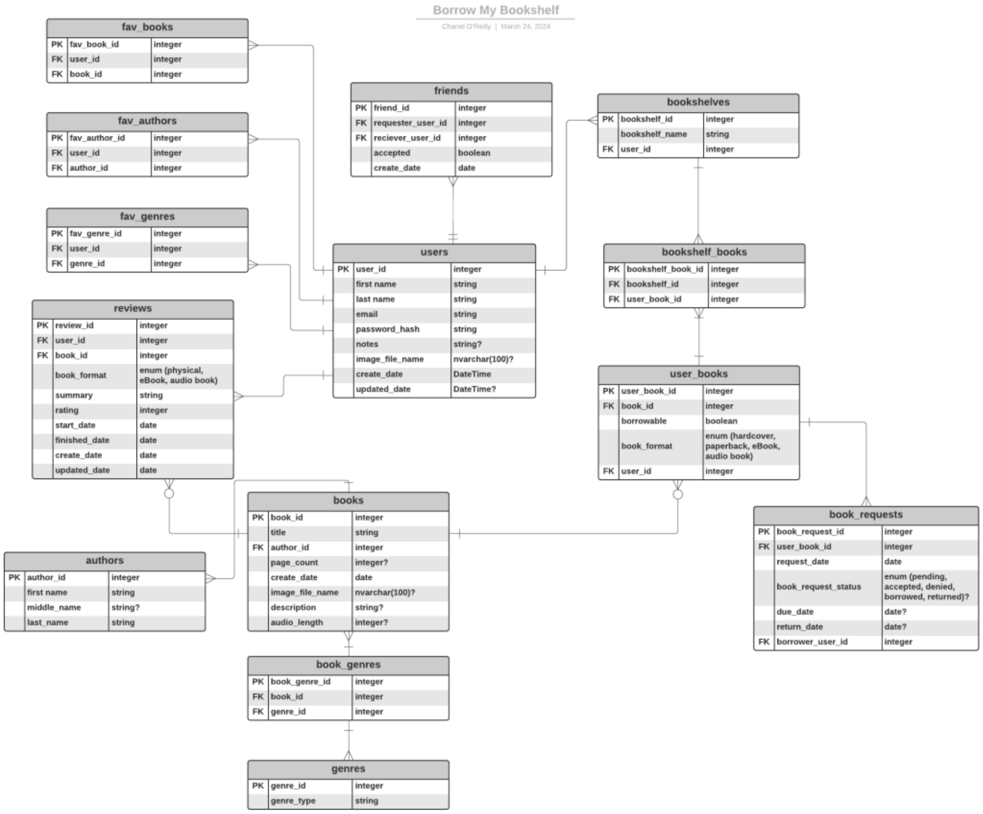
Table of Contents

Application Design and Testing

Class Design

This ERD describes the relationships between the primary components for the creation and use of the Borrow My Bookshelf application. This displays all the classes that are used on the backend in C# and used on the frontend with Typescript with React and follow the shown class and variable representation. This diagram was a helpful reference throughout the design and coding implementation process of my project.

The application is designed to be able to manage and sort your books on bookshelves, create a user profile, add friends and view their profile, bookshelves and books, request and lend books, and leave reviews on finished books. All classes within this diagram were essential to making Borrow My Bookshelf work as intended.



	fav_books	
PK	fav_book_id	integer
FK	user_id	integer
FK	book_id	integer

	fav_authors	
PK	fav_author_id	integer
FK	user_id	integer
FK	author_id	integer

	fav_genres	
PK	fav_genre_id	integer
FK	user_id	integer
FK	genre_id	integer

	reviews	
PK	review_id	integer
FK	user_id	integer
FK	book_id	integer
	book_format	enum (physical, eBook, audio book)
	summary	string
	rating	integer
	start_date	date
	finished_date	date
	create_date	date
	updated_date	date

	authors	
PK	author_id	integer
	first name	string
	middle_name	string?
	last_name	string

	friends	
PK	friend_id	integer
FK	requester_user_id	integer
FK	reciever_user_id	integer
	accepted	boolean
	create_date	date

	users	
PK	user_id	integer
	first name	string
	last name	string
	email	string
	password_hash	string
	notes	string?
	image_file_name	nvarchar(100)?
	create_date	DateTime
	updated_date	DateTime?

	books	
PK	book_id	integer
	title	string
FK	author_id	integer
	page_count	integer?
	create_date	date
	image_file_name	nvarchar(100)?
	description	string?
	audio_length	integer?

	book_genres	
PK	book_genre_id	integer
FK	book_id	integer
FK	genre_id	integer

	genres	
PK	genre_id	integer
	genre_type	string

	bookshelves	
PK	bookshelf_id	integer
	bookshelf_name	string
FK	user_id	integer

	bookshelf_books	
PK	bookshelf_book_id	integer
FK	bookshelf_id	integer
FK	user_book_id	integer

	user_books	
PK	user_book_id	integer
FK	book_id	integer
	borrowable	boolean
	book_format	enum (hardcover, paperback, eBook, audio book)
FK	user_id	integer

	book_requests	
PK	book_request_id	integer
FK	user_book_id	integer
	request_date	date
	book_request_status	enum (pending, accepted, denied, borrowed, returned)?
	due_date	date?
	return_date	date?
FK	borrower_user_id	integer

It should be noted that in this diagram all classes are in their simplest form and connected via id as they are in the database. However, there are versions of these classes that have instances of the connected classes for ease of access (e.g. DetailedUserBooks vs. UserBooks).

UI Design

The User Interface (UI) design of Borrow My Bookshelf was made to be a fun and intuitive experience for the user that allows them to seamlessly and easily log, sort, borrow and lend their books. As there are many pages, I will include the most important wireframes in creation of Borrow My Bookshelf. After a user logs into their account, they will be taken to a page that resembles Figure 1 of the home page that will include all of their books. This page will give them easy access to any page that they would like to go to. It also has functionality to add a new book, new bookshelf, or sort and filter any books on their homepage. In production, it was changed that books must be made on a bookshelf, but those books will be viewable from 'All User's Books'.

When a user clicks on the drop-down menu button connected to a book, they will be taken to a 'View Book' page like Figure 2 that will show all the current information on a book as well as any user reviews connected to that book. Some changes that were made in production was making sure that the book's general information was separate from the user, besides a review that would allow them to add the date they started and finished the book, their star rating, their thoughts, and what version of the book they read.

Figure 3 represents what a user's 'Friends' page will look like. An adjustment made in production was a button that linked to their friend's profile and gives the user the ability to view their friend's bookshelves/books.

In addition to these pages shown below, is the borrowing page which allows users to easily see which of their books they are currently lending as well as who they are borrowing books from. There is also a profile page that allows the user to select their favorite authors, favorite books, and favorite genres and leave a note for their friends to see. All pages with bookshelves with books on them also allow the user to create a 'report' by filtering, sorting, or searching what is currently on the bookshelf they are looking at, their main page bookshelf that combines all books they own currently, or the 'all books' page that contains all books within the database. Obviously before you get to any of this the user will be introduced to the login page, and possibly the sign-up page if they don't have an account yet.

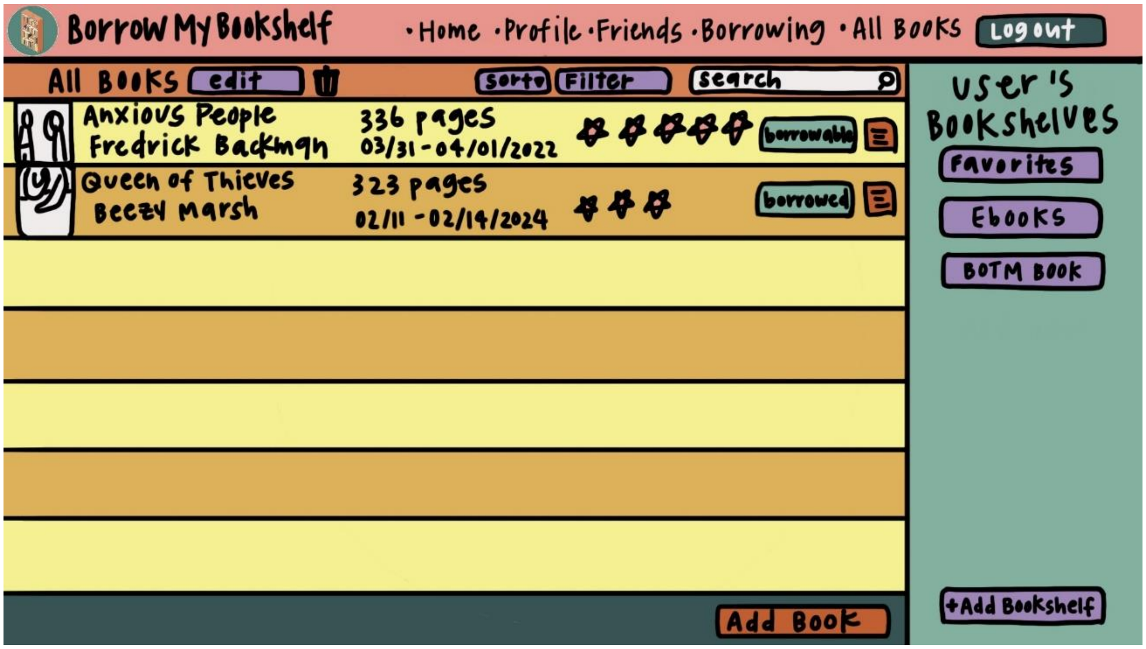
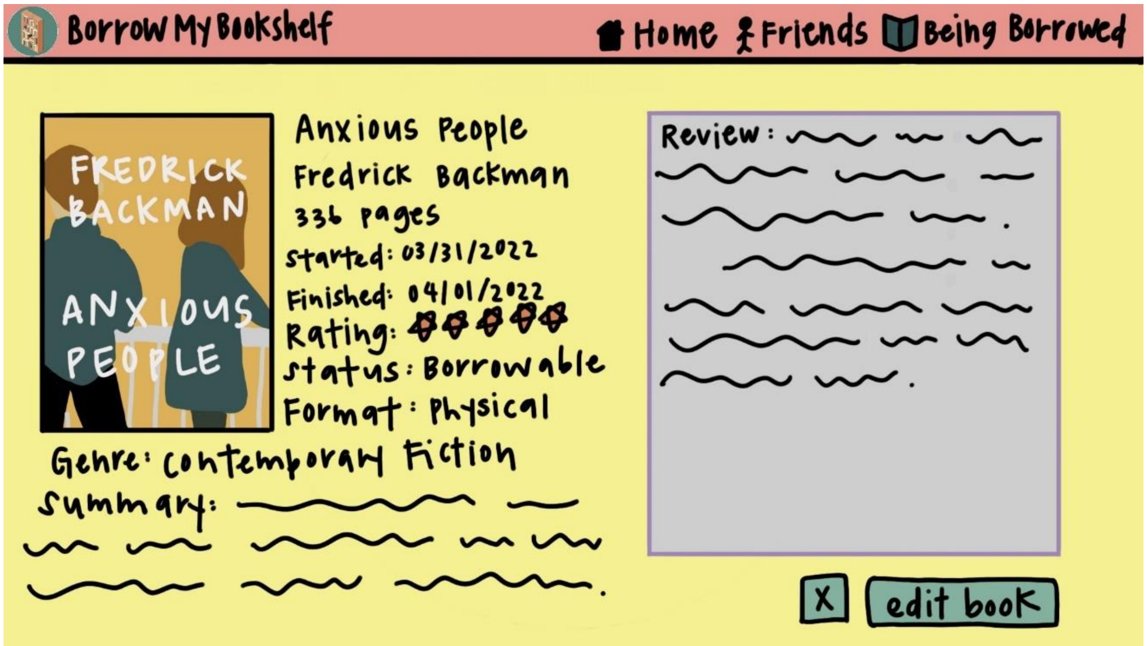


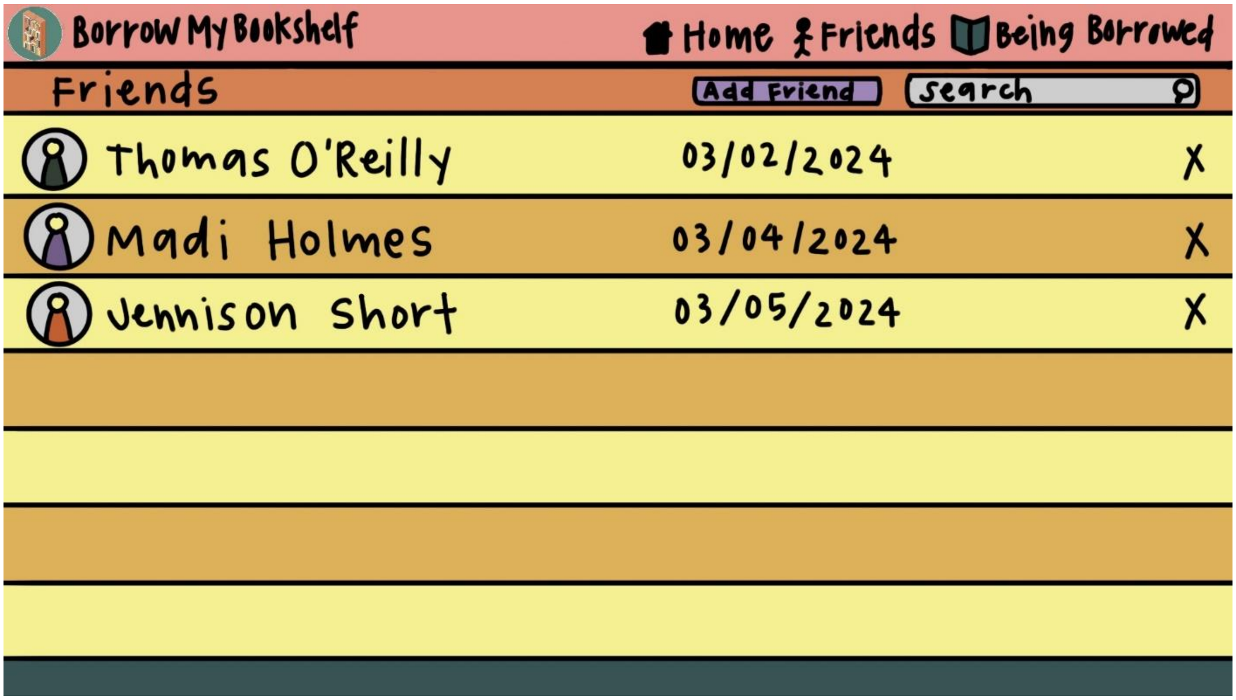
Figure 1: High Fidelity Main Page Design



Anxious People
Fredrick Backman
336 pages
Started: 03/31/2022
Finished: 04/01/2022
Rating:
Status: Borrowable
Format: Physical
Genre: contemporary Fiction
Summary:

Review:

Figure 2: High Fidelity View Book Details and Reviews



Thomas O'Reilly	03/02/2024	X
Madi Holmes	03/04/2024	X
Jennison Short	03/05/2024	X

Figure 3: High Fidelity Friends View Page

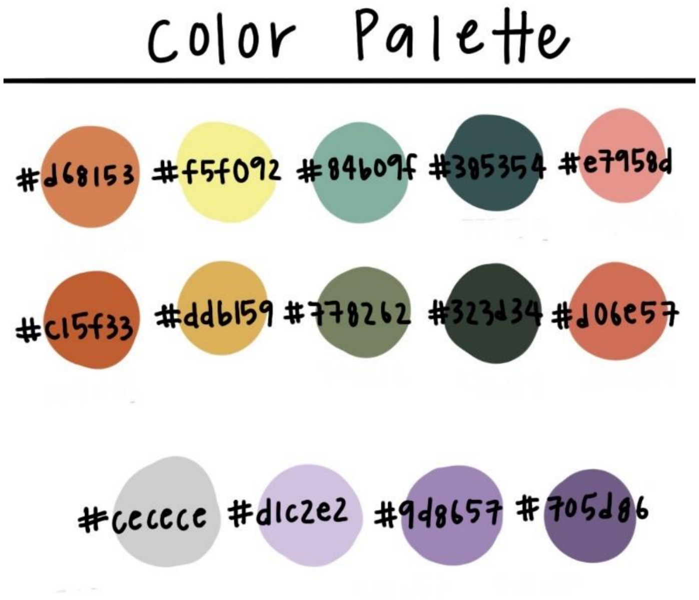


Figure 4: Color Palette Used for Borrow My Bookshelf

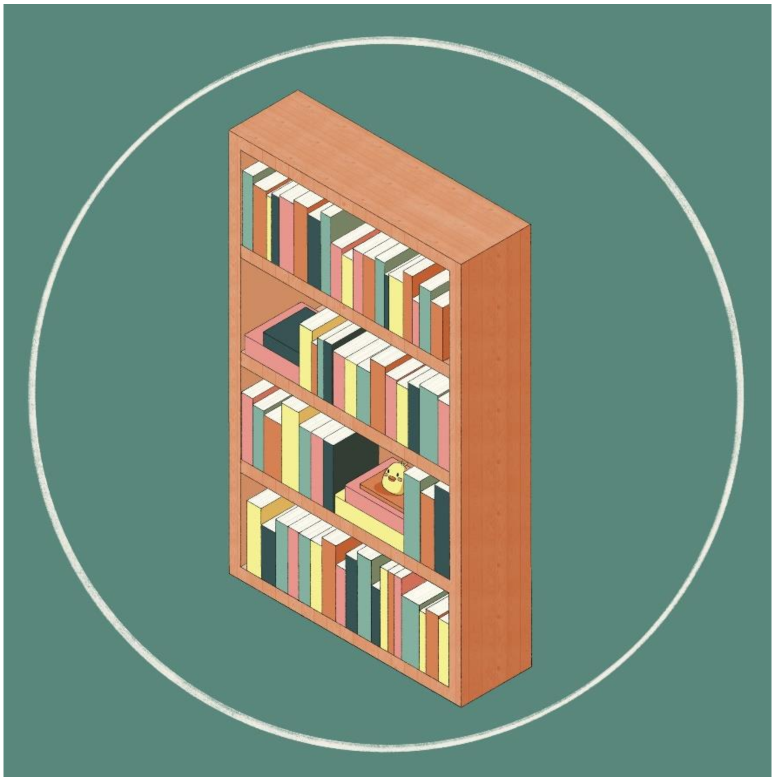


Figure 5: Logo For Borrow My Bookshelf

Borrow My Bookshelf also uses an extensive number of icons that can be found in the icon section of Google Fonts that allows developers to use their icons for free on their website.

Unit Test Plan

Introduction

Purpose

The purpose of the unit test plan is to ensure that the code written for Borrow My Bookshelf works as it is meant to and is reliable for users. Unit testing requires isolating individual components made on the front end and running a script to make sure that inputs, buttons, creation of components, deletion, etc. are working as expected and to verify their functionality. This approach allows for easily finding and fixing bugs, ensuring that the application can run without any hitches, as well as keeping us from having a regression where a bug reoccurs.

Overview

Unit testing is a very good resource for making sure that the developer has written bug free code and that their components will work as intended. The specific tests ran and shown in this document are very critical to the overall user experience. These tests are creating a new book, and making sure that modals are popping up when the appropriate button is clicked. These specific tests were chosen as many of the pages on Borrow My Bookshelf rely on correct form validation and making sure that certain buttons relay the correct information through a pop-up modal using a developer made component called ‘ModalButton’.

- Form Validation: Tested that when the correct information is inserted into the HTMLInputElements that it can create a book. As well as, making sure that the book cannot be created before the required fields are entered, ensuring that the button stays disabled until the form is filled out correctly and that it doesn’t become disabled after the form has been filled out.

- Modal Functionality: Tested that there was no modal on the page already, and that after firing the correct button that a modal with the selected message text and button text pop up.

Test Plan

Items

- Development Environment: Visual Studios 2022 was set up with React installed for the front end which is essential to our unit tests.
- Test Framework: Jest, a popular testing framework for JavaScript based applications, including React.
- Appropriate Configs: Configurations needed to be added to mock navigation and allow for testing Typescript.
- Application source code components: The finished components were completed and already part of the codebase before creating tests based on them.

Features

- Create Book Validation: Test that adding the correct information to the input boxes will allow for creating a book. As well as, testing that the submission button stays disabled until all required fields are filled in and that it stays enabled after the fact.
- Modal Button Functionality: Tests that there isn’t already a modal up, and that after clicking the button that a modal of the correct type appears with correct message text and button text.

Deliverables

- Test Scripts: A collection of test scripts is provided as part of the source codebase as a fileName.spec.tsx file, written using the Jest testing framework.
- Test Results: A report detailing the results of the ran test scripts showing the test results.

Tasks

- Set up the development environment using Visual Studios as the coding IDE for making the application and the following user tests. This includes downloading npm, React, Typescript, and Jest.
- Make user scripts for each component to be tested using the same name but ending with spec.tsx. In this case we need CreateBook.spec.tsx and ModalButton.spec.tsx. These now can be easily run from our Developer PowerShell.
- Execute the jest commands to test the scripts made in the Developer PowerShell. Monitor the results for failure or success.
- Look over the returned results and check for any failed tests or unexpected results.
- Adjust the unit test or the component's code until unit tests are passing and are acting as expected.

Needs

- Software Development IDE: Using Microsoft Visual Studios, I was able to download the most current version of React for this project that was used to write the components and unit tests.

- Source Code and Test Scripts: Access to the specific component's frontend code written in Typescript with React and the following spec.tsx file for running the written test were needed to run the test.
- Testing Tools: For this application's test purposes, Jest was used as the testing framework downloading the most recent version via npm. It is connected to the project files and was ran through Developer PowerShell.

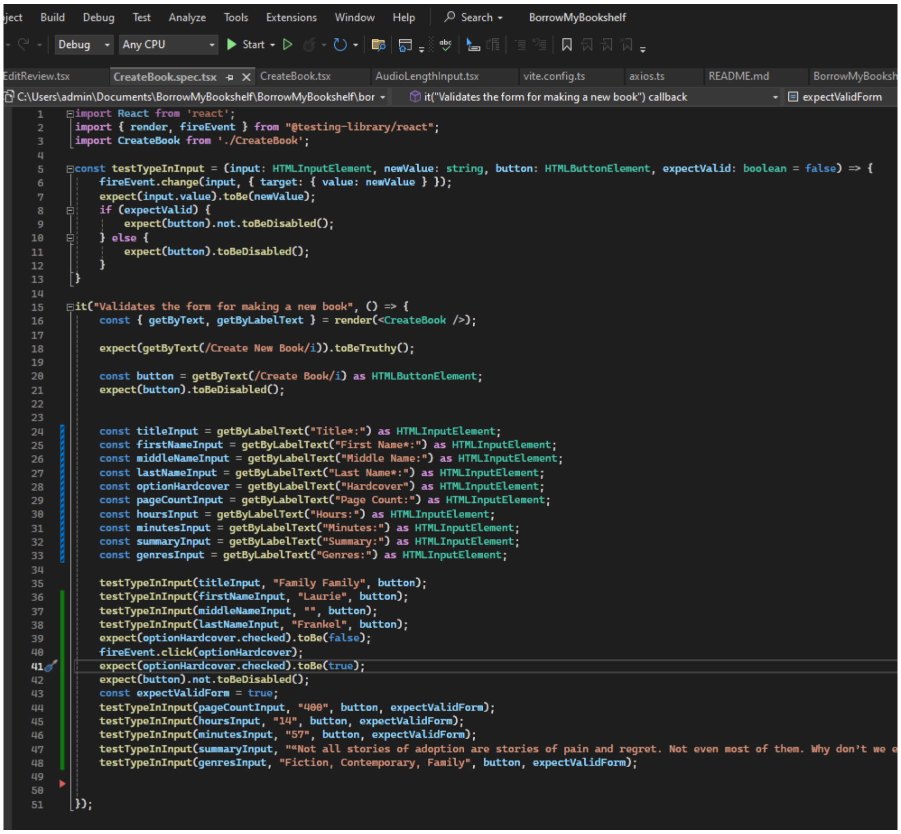
Pass/Fail Criteria

- Making sure that validation on Create Book works as expected:
  - Pass: 'Create Book' button is disabled until all required fields are filled out correctly.
  - Fail: 'Create Book' button either becomes enabled earlier than expected, does not become enabled at all, or does not stay enabled.
- Making sure that creation of modals works upon clicking button:
  - Pass: There is no modal before clicking the intended button, but after clicking the button there is a modal with the correctly set message text and button text.
  - Fail: There is either already a modal existing or modal doesn't appear correctly rendered after clicking the modal button.

Specifications

Here is the code that shows what was written in Jest for Typescript with React.

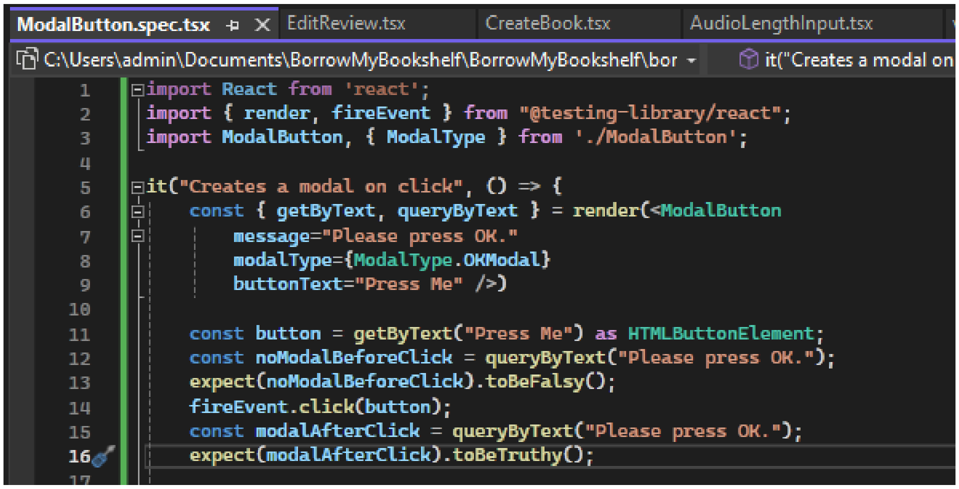
- Unit Test One – Making sure that validation on Create Book works as expected:



ject Build Debug Test Analyze Tools Extensions Window Help
Search
BorrowMyBookshelf
Debug
Any CPU
Start
EditReview.tsx
CreateBook.spec.tsx
CreateBook.tsx
AudioLengthInput.tsx
vite.config.ts
axios.ts
README.md
BorrowMyBooksh
C:\Users\admin\Documents\BorrowMyBookshelf\BorrowMyBookshelf\bor
it("Validates the form for making a new book") callback
expectValidForm

```
import React from 'react';
import { render, fireEvent } from "@testing-library/react";
import CreateBook from './CreateBook';

const testTypeInInput = (input: HTMLInputElement, newValue: string, button: HTMLButtonElement, expectValid: boolean = false) => {
    fireEvent.change(input, { target: { value: newValue } });
    expect(input.value).toBe(newValue);
    if (expectValid) {
        expect(button).not.toBeDisabled();
    } else {
        expect(button).toBeDisabled();
    }
}

it("Validates the form for making a new book", () => {
    const { getByText, getByLabelText } = render(<CreateBook />);

    expect(getByText(/Create New Book/i)).toBeTruthy();

    const button = getByText(/Create Book/i) as HTMLButtonElement;
    expect(button).toBeDisabled();


    const titleInput = getByLabelText("Title*:") as HTMLInputElement;
    const firstNameInput = getByLabelText("First Name*:") as HTMLInputElement;
    const middleNameInput = getByLabelText("Middle Name:") as HTMLInputElement;
    const lastNameInput = getByLabelText("Last Name*:") as HTMLInputElement;
    const optionHardcover = getByLabelText("Hardcover") as HTMLInputElement;
    const pageCountInput = getByLabelText("Page Count:") as HTMLInputElement;
    const hoursInput = getByLabelText("Hours:") as HTMLInputElement;
    const minutesInput = getByLabelText("Minutes:") as HTMLInputElement;
    const summaryInput = getByLabelText("Summary:") as HTMLInputElement;
    const genresInput = getByLabelText("Genres:") as HTMLInputElement;

    testTypeInInput(titleInput, "Family Family", button);
    testTypeInInput(firstNameInput, "Laurie", button);
    testTypeInInput(middleNameInput, "", button);
    testTypeInInput(lastNameInput, "Frankel", button);
    expect(optionHardcover.checked).toBe(false);
    fireEvent.click(optionHardcover);
    expect(optionHardcover.checked).toBe(true);
    expect(button).not.toBeDisabled();
    const expectValidForm = true;
    testTypeInInput(pageCountInput, "400", button, expectValidForm);
    testTypeInInput(hoursInput, "14", button, expectValidForm);
    testTypeInInput(minutesInput, "57", button, expectValidForm);
    testTypeInInput(summaryInput, "“Not all stories of adoption are stories of pain and regret. Not even most of them. Why don’t we e
    testTypeInInput(genresInput, "Fiction, Contemporary, Family", button, expectValidForm);


});
```

- Unit Test Two – Making sure that creation of modals works upon clicking the button:



```
import React from 'react';
import { render, fireEvent } from "@testing-library/react";
import ModalButton, { ModalType } from './ModalButton';

it("Creates a modal on click", () => {
    const { getByText, queryByText } = render(<ModalButton
        message="Please press OK."
        modalType={ModalType.OKModal}
        buttonText="Press Me" />)

    const button = getByText("Press Me") as HTMLButtonElement;
    const noModalBeforeClick = queryByText("Please press OK.");
    expect(noModalBeforeClick).toBeFalsy();
    fireEvent.click(button);
    const modalAfterClick = queryByText("Please press OK.");
    expect(modalAfterClick).toBeTruthy();
```

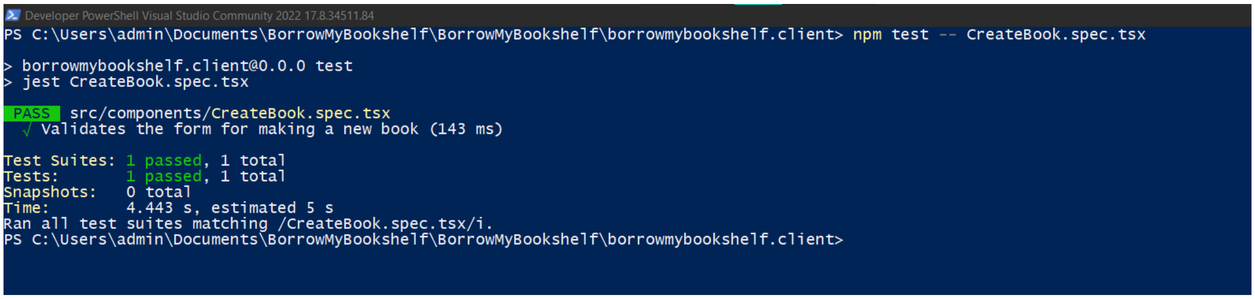
Procedures

- After the creation of the unit tests, they were run in the Developer PowerShell using the Jest framework.
- The Modal Button had no problems with testing and passed with the correct results from the first iteration.
- The 'Create Book' unit test had to be ran a couple times as I had issues with how I set up the test the first time. There was also a more significant bug:
    - I found that the 'Create Book' button was being enabled before the required field of format was selected. This is because the BookFormat had a hidden default value allowing the user to proceed without selecting a format. A little correction of typing within the format's functions made the button function as expected.
    - This made me grateful that I did a unit test for this feature as it helped me catch a bug that I don't think I would have easily caught myself.

Results

The test results for Borrow My Bookshelf were obtained after running the test scripts using the Jest testing framework ran through the Developer PowerShell.

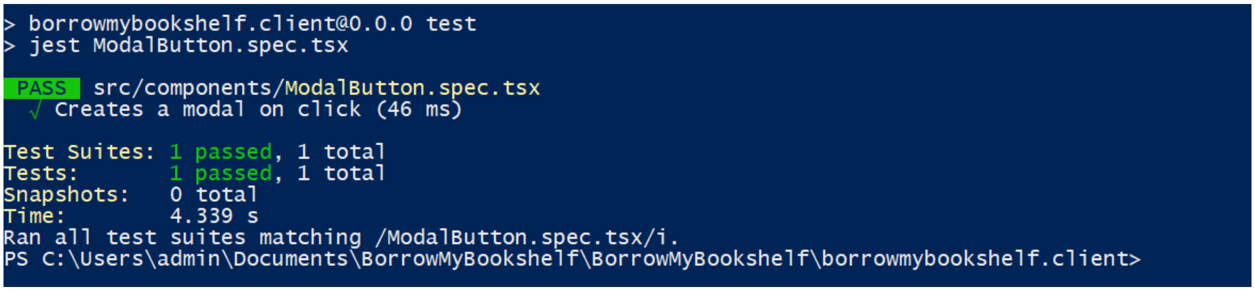
- Unit Test One – Making sure that validation on Create Book works as expected:
    - This test passed when the 'Create Book' form was filled out with correct information and the button was either kept in a disabled state or enabled and stayed enabled after all required information was filled out in the form.



Developer PowerShell Visual Studio Community 2022 17.8.34511.84

```
PS C:\Users\admin\Documents\BorrowMyBookshelf\BorrowMyBookshelf\borrowmybookshelf.client> npm test -- CreateBook.spec.tsx

> borrowmybookshelf.client@0.0.0 test
> jest CreateBook.spec.tsx

 PASS  src/components/CreateBook.spec.tsx
  √ Validates the form for making a new book (143 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        4.443 s, estimated 5 s
Ran all test suites matching /CreateBook.spec.tsx/i.
PS C:\Users\admin\Documents\BorrowMyBookshelf\BorrowMyBookshelf\borrowmybookshelf.client>
```

- Unit Test Two – Making sure that creation of modals works upon clicking the button:
    - This test passed when there was no pre-existing modal already up and after clicking the button, the correct type of modal popped up with the correctly made message text and button text.



```
> borrowmybookshelf.client@0.0.0 test
> jest ModalButton.spec.tsx

 PASS  src/components/ModalButton.spec.tsx
  √ Creates a modal on click (46 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        4.339 s
Ran all test suites matching /ModalButton.spec.tsx/i.
PS C:\Users\admin\Documents\BorrowMyBookshelf\BorrowMyBookshelf\borrowmybookshelf.client>
```

Hosted Web Application

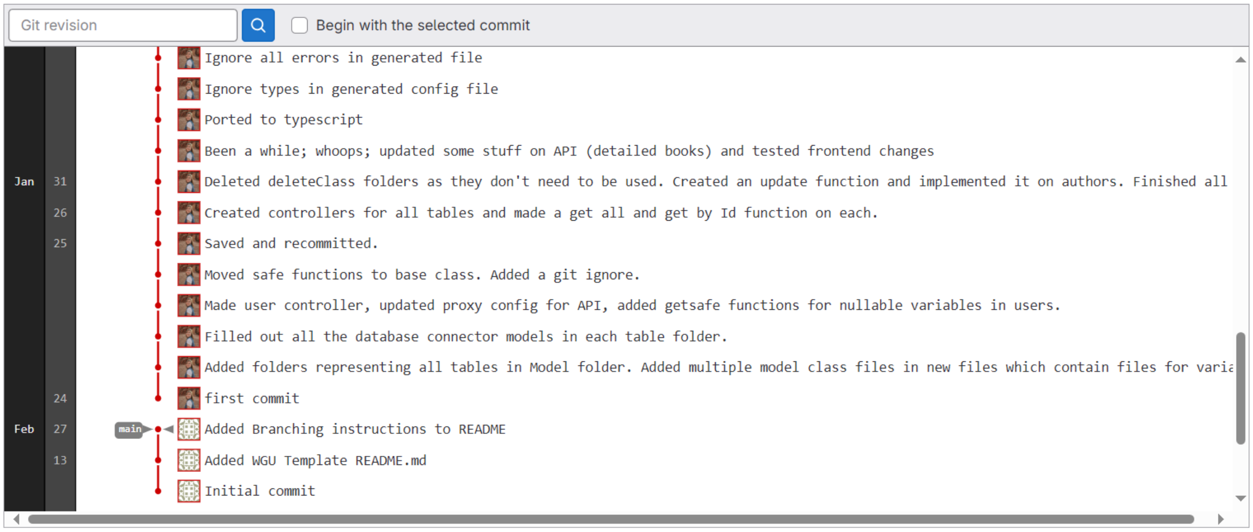
Hosted Web Application Link:

https://borrowmybookshelf.com/

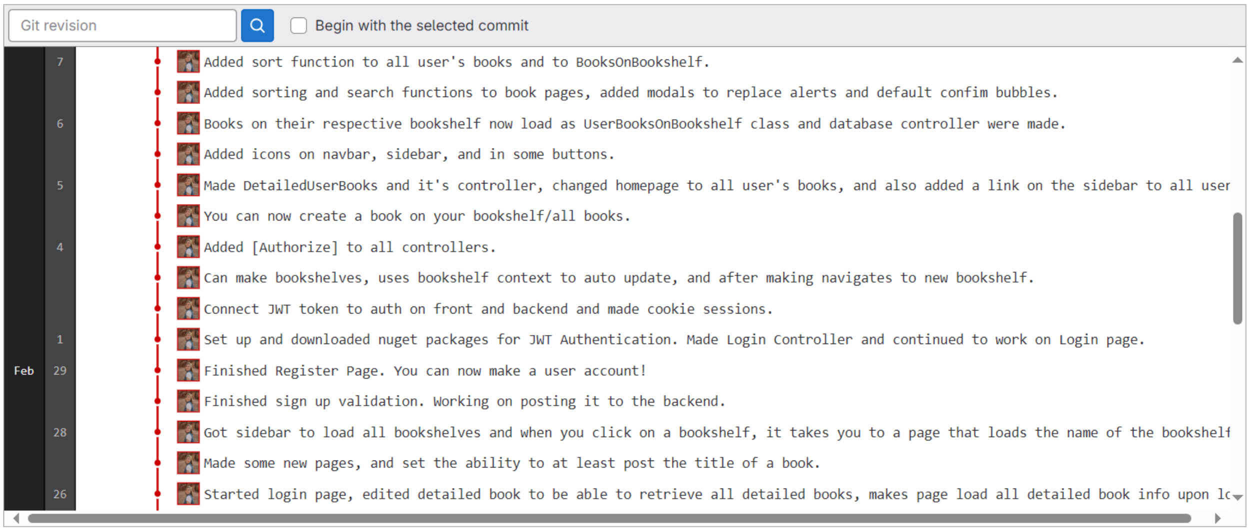
GitLab Repository:

WGU GitLab Environment / Student Repos / corei11 / D424 Software Engineering Capstone · GitLab

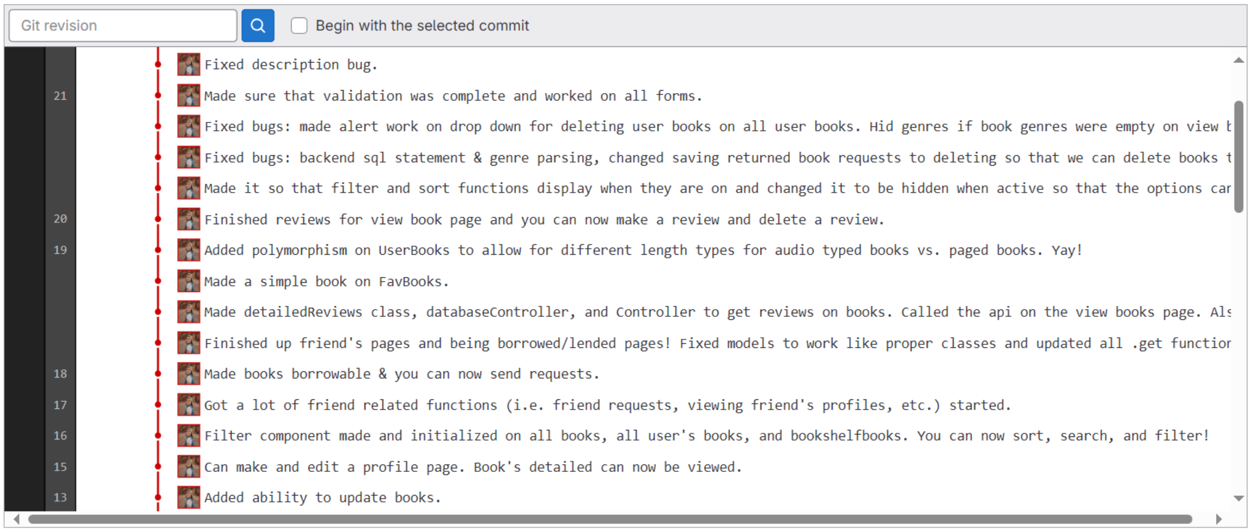
Version History:



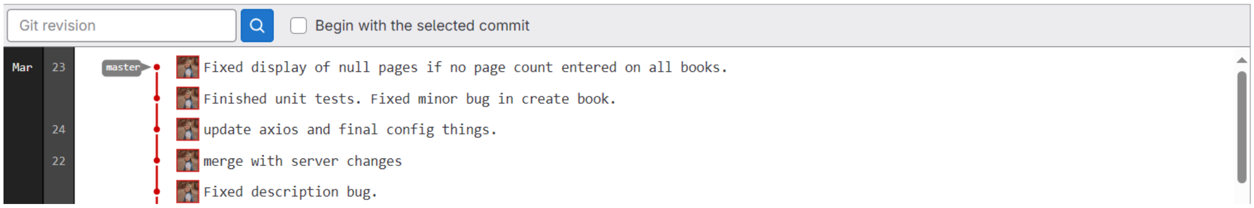
```
            Ignore all errors in generated file
            Ignore types in generated config file
            Ported to typescript
            Been a while; whoops; updated some stuff on API (detailed books) and tested frontend changes
Jan  31     Deleted deleteClass folders as they don't need to be used. Created an update function and implemented it on authors. Finished all
     26     Created controllers for all tables and made a get all and get by Id function on each.
     25     Saved and recommitted.
            Moved safe functions to base class. Added a git ignore.
            Made user controller, updated proxy config for API, added getsafe functions for nullable variables in users.
            Filled out all the database connector models in each table folder.
            Added folders representing all tables in Model folder. Added multiple model class files in new files which contain files for varia
     24     first commit
Feb  27     main  Added Branching instructions to README
     13     Added WGU Template README.md
            Initial commit
```



```
     7      Added sort function to all user's books and to BooksOnBookshelf.
            Added sorting and search functions to book pages, added modals to replace alerts and default confim bubbles.
     6      Books on their respective bookshelf now load as UserBooksOnBookshelf class and database controller were made.
            Added icons on navbar, sidebar, and in some buttons.
     5      Made DetailedUserBooks and it's controller, changed homepage to all user's books, and also added a link on the sidebar to all user
            You can now create a book on your bookshelf/all books.
     4      Added [Authorize] to all controllers.
            Can make bookshelves, uses bookshelf context to auto update, and after making navigates to new bookshelf.
            Connect JWT token to auth on front and backend and made cookie sessions.
     1      Set up and downloaded nuget packages for JWT Authentication. Made Login Controller and continued to work on Login page.
Feb  29     Finished Register Page. You can now make a user account!
            Finished sign up validation. Working on posting it to the backend.
     28     Got sidebar to load all bookshelves and when you click on a bookshelf, it takes you to a page that loads the name of the bookshelf
            Made some new pages, and set the ability to at least post the title of a book.
     26     Started login page, edited detailed book to be able to retrieve all detailed books, makes page load all detailed book info upon lo
```



```
            Fixed description bug.
     21     Made sure that validation was complete and worked on all forms.
            Fixed bugs: made alert work on drop down for deleting user books on all user books. Hid genres if book genres were empty on view b
            Fixed bugs: backend sql statement & genre parsing, changed saving returned book requests to deleting so that we can delete books t
            Made it so that filter and sort functions display when they are on and changed it to be hidden when active so that the options car
     20     Finished reviews for view book page and you can now make a review and delete a review.
     19     Added polymorphism on UserBooks to allow for different length types for audio typed books vs. paged books. Yay!
            Made a simple book on FavBooks.
            Made detailedReviews class, databaseController, and Controller to get reviews on books. Called the api on the view books page. Als
            Finished up friend's pages and being borrowed/lended pages! Fixed models to work like proper classes and updated all .get functio
     18     Made books borrowable & you can now send requests.
     17     Got a lot of friend related functions (i.e. friend requests, viewing friend's profiles, etc.) started.
     16     Filter component made and initialized on all books, all user's books, and bookshelfbooks. You can now sort, search, and filter!
     15     Can make and edit a profile page. Book's detailed can now be viewed.
     13     Added ability to update books.
```



```
Mar  23   Fixed display of null pages if no page count entered on all books.
          Finished unit tests. Fixed minor bug in create book.
     24   update axios and final config things.
     22   merge with server changes
          Fixed description bug.
```

User Guide for Running the Application and Maintenance

Introduction

This user guide is specifically designed to help someone set up this application and get it running from their own machine and maintain the production server.

Setting-up the Dev Environment:

1. Install Microsoft Visual Studios 2022.
2. Make sure you have Node.js installed on your VS 2022 in order to be able to access Typescript and Javascript with React.
    a. Open Visual Studios Installer.
    b. Make sure Node.js development is selected and click 'install'.
3. Open Visual Studios and use its git tools to connect to and clone the git repository.
4. In the Developer PowerShell navigate to 'borrowmybookshelf.client' and run `npm install` to install all dependencies.
5. Click start button at top and it should run.
6. In the client directory from the Developer PowerShell use `npm test --filename.spec.tsx` to run a specific unit test or just `npm test` to run all tests.

Key Classes and Configs:

Frontend:

- src/api/axios.ts: This file defines the base url for all frontend requests to the api. Keep this using windows.location.origin, this uses the current url which is convenient for running the application on both local machine and server.
- /public: Contains all static assets such as images used for icons and the favicon.

- vite.config.ts: Is the main config for the frontend project. It defines which urls are handled by React and which are handled by the API. Currently only urls beginning with /api are forwarded to the API.
- jest.config.ts: Defines how files are mocked and how tests are run using Jest. Currently all css and images are mocked and should not be tested directly.
- src/components: Contains all React components, styles, and unit tests. Names should be clear enough to recognize what pages they point to.
- src/App.tsx: Defines all routes; There are two sets of routes:
    - When the user is not authorized, they are only presented with the Login routes.
    - When the user is authorized, they are only presented with the main application routes.
- src/models: Contains all classes mirroring how they are constructed on the backend for easy use on the frontend components.
- src/helpers: Contains helper functions and objects critical to many components.
    - AuthHelper.ts: Used to check and set authentication status and get current user info using the authentication cookie.
    - BookHelper.ts and UpdateBookHelper.ts: Used to help with creating and updating logic involving books. It is broken down into a helper because it gets quite complicated.
    - NetworkHelper.ts: Handles axios post, put, and delete requests so that authentication and other boiler plate does not need to be repeated throughout the project.

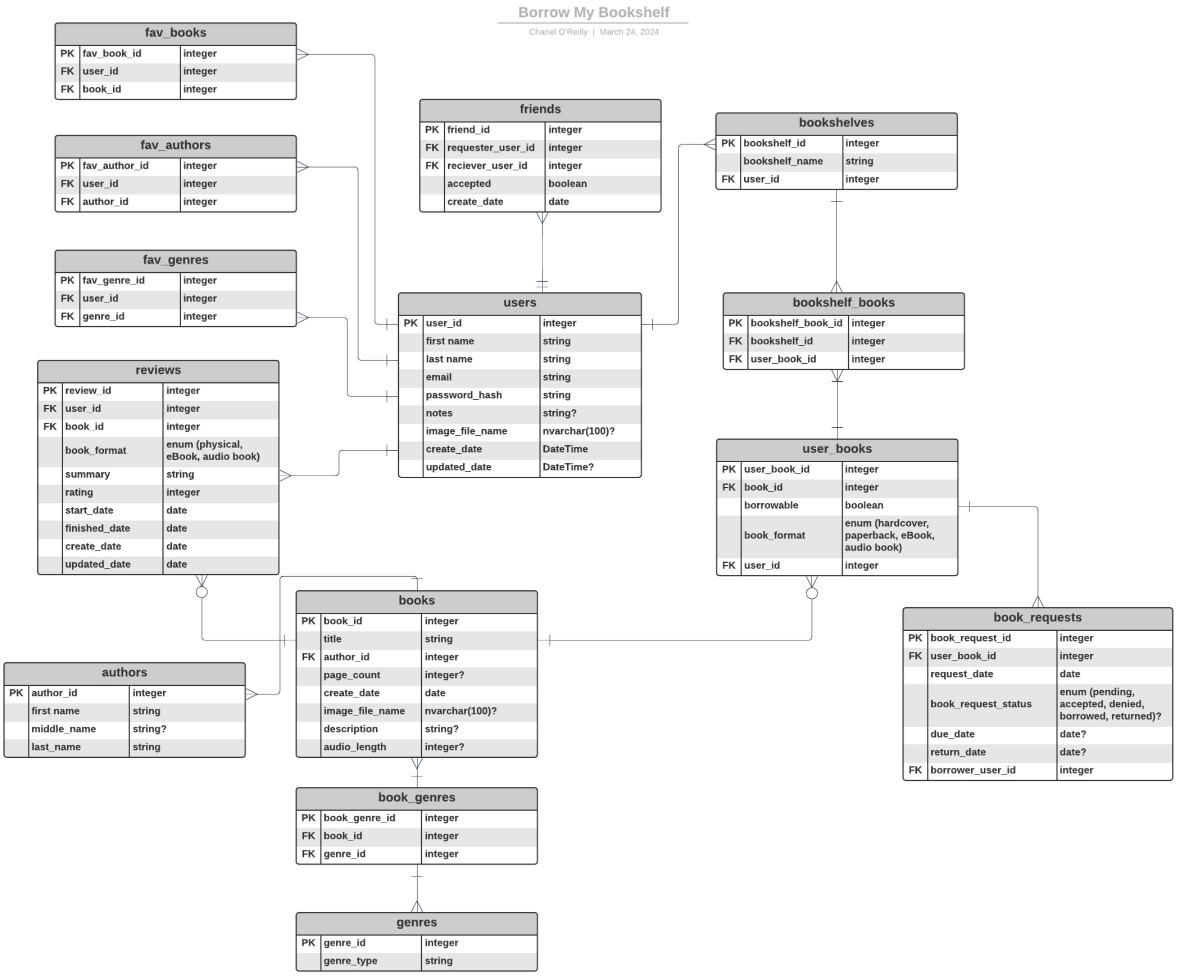
- src/context: Contains state that needs to be shared between components. Such as, authentication, bookshelves, and userBooks. These contexts can be used to get or refresh current data.
    - Take care as each state needs to be managed properly. If data within a context should change due to clicking a link or updating the data, the refresh function must be called.
- src/config/setupTests.ts: Runs before every unit test. Currently it only mocks navigation. This and adjacent files could be adjusted for simplifying complex tests, especially when it should be run on every test.
- Program.cs: Is the main config. Set up for authentication via JWT tokens is in this file. It also has boilerplate for setting up the controllers, etc.

Backend:

- Controllers: Contains all routes unique to each of the sixteen tables, most call through to Database Connectors with very little business logic. Besides the login controller, each is annotated with [authorize] which ensures that users can only get into certain data if they have been authenticated.
- Models: Contains a class for creating, updating, and accessing each table. The Database Connector for each table contains the business logic. More complex queries are in separate detailed connectors and use associated joins as necessary.
- Models/DatabaseConnector.cs: handles all database connection logic and the parameterization of all user input. Having it in a centralized location ensures that we can handle all connections properly and protect against MySQL injection attacks.

- Models/PasswordHelper.cs: Handles password hashing before storage and comparison to login a user.

To understand the relationships of classes that are used on both the front and backend, feel free to examine this ERD:



Setting-up the Production Environment:

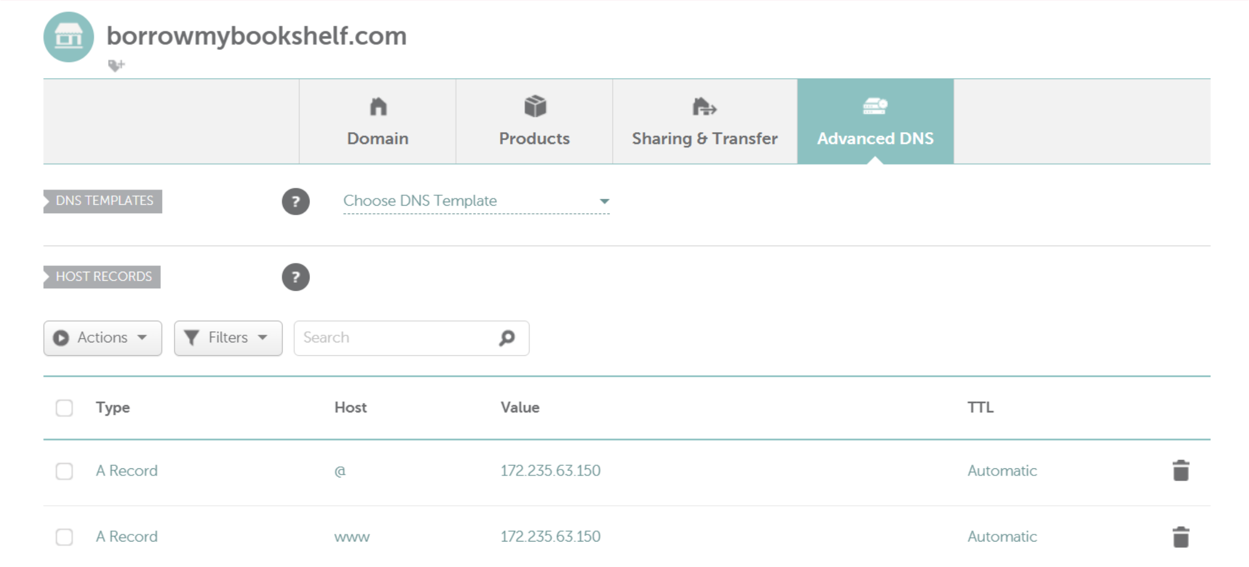
Pre-requisites:

Register the Domain

1. Register the website domain at a domain registration site. This project used https://namecheap.com/.

Create and Connect Cloud Server

1. Create an account at a cloud server site. I used Akamai, formerly known as Linode.
2. Buy a linode server in the region best suited to your location. For me, I chose the smallest size in the Los Angeles region running on Ubuntu as I am running my database on an Ubuntu Linex server.
3. Once your linode server is running, you need to make two A Records. One for '@' and 'www' pointing to the IP address found on your cloud server.



Type	Host	Value	TTL
A Record	@	172.235.63.150	Automatic
A Record	www	172.235.63.150	Automatic

4. Using a WSL terminal, connect to your server via ssh. The command is:
   a. Set up an SSH key on your server. Use these commands:
      i. mkdir /.ssh
      ii. cd /.ssh
      iii. ssh-keygen
         1. Press enter when prompted to choose a file location to generate the key in the default location.
         2. No need to make passphrase if not desired.

iv. cat id_rsa.pub

1. Copy the result into your server setting on the web portal.

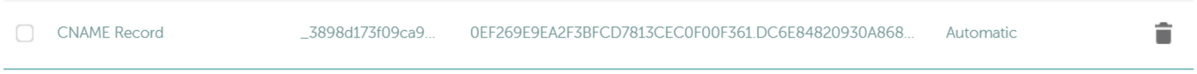
v. ssh root@ip.address

vi. Change out ip.address with the one given to you through your cloud server.

vii. When prompted enter the password you set up.

Add SSL with Namecheap

1. Go to Namecheap and after you are logged in, click on SSL certificates.
2. Buy a ‘PositiveSSL’ for however long you plan on running your site at a minimum a year.
3. Make a ‘Certificate Signing Request’(csr) on your server by running the following commands:
   a. openssl genrsa -out borrowmybookshelf.key 2048
   b. openssl req -new -key borrowmybookshelf.key -out borrowmybookshelf.csr
4. Fill out the information it asks you for. Once you finish filling that out, make sure to use `cat borrowmybookshelf.csr` and copy the whole certificate including the beginning and ending headers.
5. Go to your SSL certificate and click ‘Start SSL’ to start the process of setting up your SSL certificate by pasting in your newly made csr. This should get your domain from the certificate and start the process for getting your SSL to start working.
6. Now go to your domain DNS advanced settings for borrowmybookshelf.com and add a CNAME record using the given target and value your SSL provided.



□	CNAME Record	_3898d173f09ca9...	0EF269E9EA2F3BFCD7813CEC0F00F361.DC6E84820930A868...	Automatic

7. After you do this, wait for an email from Namecheap with the required files you will need to put into your server.
8. Download the files and copy them to your server using the following commands in the folder that you downloaded the files to:
    a. scp borrowmybookshelf_com.ca-bundle root@172.235.63.150:/ssl/borrowmybookshelf_com.ca-bundle
    b. scp borrowmybookshelf_com.crt root@172.235.63.150:/ssl/borrowmybookshelf_com.crt
9. On your server in the ssl folder combine the .ca-bundle and .crt file into one chaining file into your ssl file using the command: ` cat borrowmybookshelf_com.crt borrowmybookshelf_com.ca-bundle >> borrowmybookshelf_chain.crt`.
10. After combining them, use an editor of your choice to ensure that each end and start delimiter are on their own line. If not, add a return in between the delimiters. Make sure each has an equal amount of hyphens.

Server Set-Up

MySQL

1. Set up MySQL tables and root password.
    a. First, run `sudo apt update && sudo apt install mysql-server`.
    b. Run `sudo service mysql start` to ensure MySQL continuously runs on the server.
    c. Access MySQL with `mysql`.
    d. Run the query `ALTER USER ‘root’@’localhost’ IDENTIFIED BY ‘1789’;`

e. Then run `FLUSH PRIVILEGES;`.

f. Create the database and tables based on the code provided in the Create_Tables.sql file in the BorrowMyBookshelf folder.

g. Type `exit;` to exit MySQL.

Cloning Git Repository

2. Clone git repository to server.

    a. Install git by running `sudo apt install git`.

    b. Set up an SSH key on your server. Use these commands:

        i. mkdir /.ssh

        ii. cd /.ssh

        iii. ssh-keygen

            1. Press enter when prompted to choose a file location to generate the key in the default location.

            2. No need to make passphrase if not desired.

        iv. cat id_rsa.pub

            1. Copy the result into your GitLab profile.

    c. To get the repo onto the server run `git clone git@gitlab.com:wgu-gitlab-environment/student-repos/corei11/d424-software-engineering-capstone.git`.

Setting Up API

3. Build and run API code on server.

    a. Run the following commands to set up .NET on your server:

        i. wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
        ii. sudo dpkg -i packages-microsoft-prod.deb
        iii. rm packages-microsoft-prod.deb
        iv. sudo apt-get update
        v. sudo apt-get install -y dotnet-sdk-8.0
        vi. sudo apt-get update
        vii. sudo apt-get install -y aspnetcore-runtime-8.0

b. Navigate to BorrowMyBookshelf.Server/ in the cloned repository.

c. Run `dotnet publish -c release -r ubuntu.14.04-x64` to compile the code for running on the server.

d. Set up virtual session and run the application by running the following commands:

    i. screen -S backend
    ii. Navigate to BorrowMyBookshelf.Server/publish
    iii. dotnet BorrowMyBookshelf.Server.dll --urls https://localhost:7136
    iv. Press ctrl + A then d to leave the session.
        1. You can return to this session anytime by running `screen -r backend`.
        2. If you make any code changes, run `git pull` then republish following step c above then return to this screen, stop the command with ctrl + c, and then rerun step iii above,

making sure to disconnect from the session when you’re done.

Git puSetting-up The React App

4. Build and run React App on server.
    a. Set Node.js and npm by running the following commands:
        i. curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash –
        ii. sudo apt install -y nodejs
            1. If you want to check that the last step worked run:
            2. `node -v` and `npm -v`
    b. Navigate to borrowmybookshelf.client in the cloned repository.
    c. Run `npm run build` to transpile and optimize the React application for production.
    d. Set up virtual session and run the application by running the following commands:
        i. `screen -S frontend`
        ii. Navigate to borrowmybookshelf.client in the cloned repository.
        iii. `npm run preview` to serve the build file.
        iv. Press ctrl + A then d to leave the session.
            1. You can return to this session anytime by running `screen -r frontend`.
            2. If you make any code changes, run `git pull` then run `npm run build` in the borrowmybookshelf.client directory. You do not need to stop or rerun `npm run preview`.

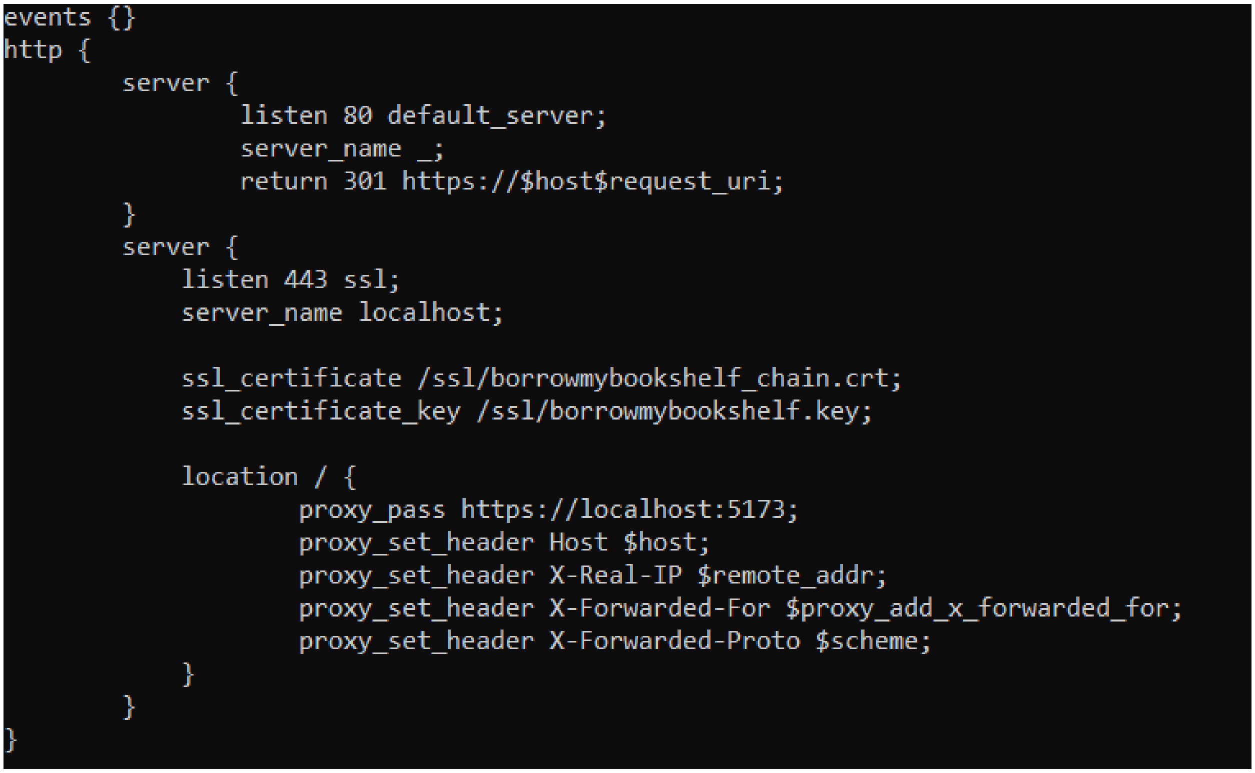
Setting-up the Reverse Proxy

5. Install and set up Nginx.

   a. To install Nginx, run `sudo apt install nginx`.

   b. Set-up the Nginx proxy config with the editor of your choice. For example, you could run `nano /etc/nginx/nginx.conf`.

   c. Put the following contents into the config:



```
events {}
http {
        server {
                listen 80 default_server;
                server_name _;
                return 301 https://$host$request_uri;
        }
        server {
            listen 443 ssl;
            server_name localhost;

            ssl_certificate /ssl/borrowmybookshelf_chain.crt;
            ssl_certificate_key /ssl/borrowmybookshelf.key;

            location / {
                    proxy_pass https://localhost:5173;
                    proxy_set_header Host $host;
                    proxy_set_header X-Real-IP $remote_addr;
                    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                    proxy_set_header X-Forwarded-Proto $scheme;
            }
        }
}
```

   d. Run `sudo systemctl restart nginx` to ensure it's always running.

The application should now be up and running, you can check at the domain you purchased.

User Guide for Application Users

Introduction

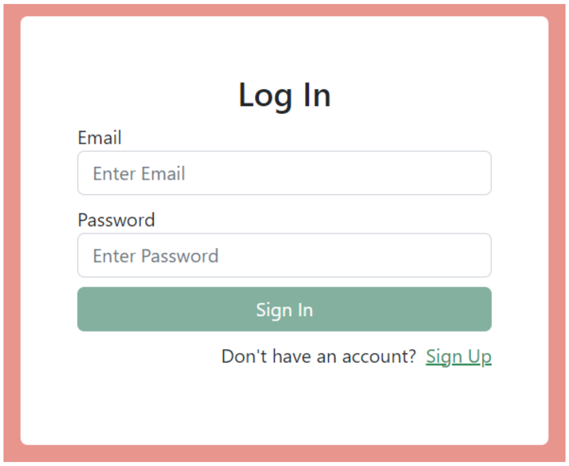
This User Guide will give a thorough overview of how to access and use Borrow My Bookshelf.

Installation and Using the Application

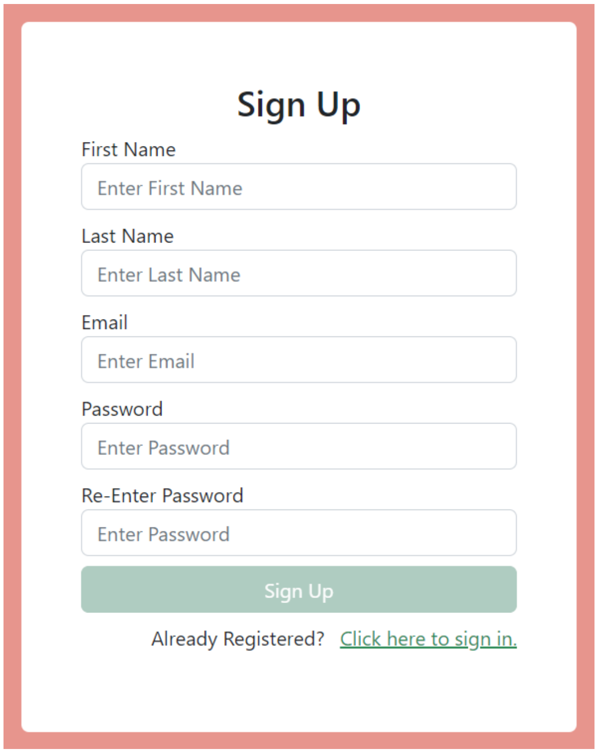
Borrow My Bookshelf doesn't need to be downloaded as it is a website running on the cloud. You can access it at BorrowMyBookshelf.com. I will go into depth about how the site works below with detailed explanations.

Login and Signup

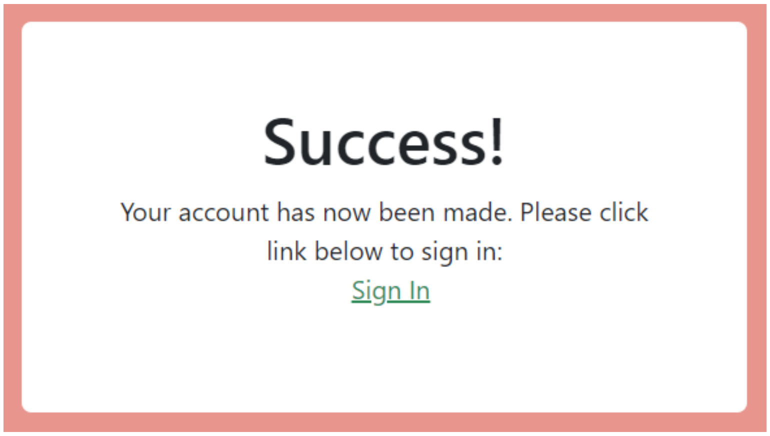
1. When you first arrive at BorrowMyBookshelf.com, it will take you to the login page.
2. If you already have an account, log in with your email and password. If you need an account, click on the link below that says "Don't have an account? Sign Up".



3. If you need to create an account, fill in your first and last name, email, and unique password. The password requires at least 8 characters, an upper and lowercase letter, a number, and special character. You won't be allowed to sign up until you fill out your information correctly as the button will be disabled.



4. After you create an account, you will be alerted that your account was successfully created and be shown a screen that allows you to be rerouted to the 'Login Page' to sign in from there.



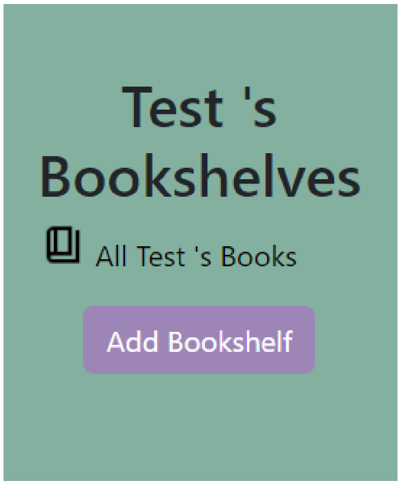
Success!

Your account has now been made. Please click link below to sign in:
Sign In

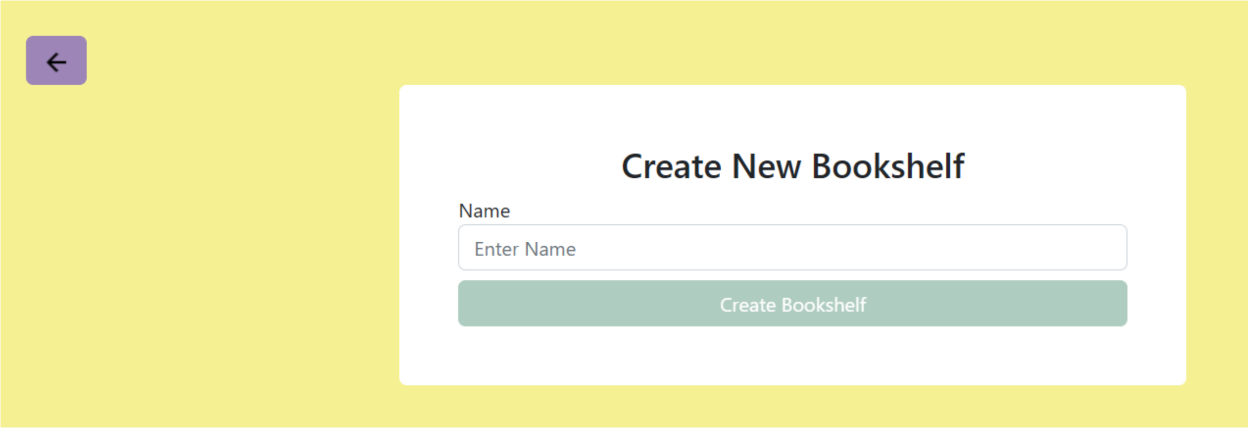
Bookshelves

Create a New Bookshelf

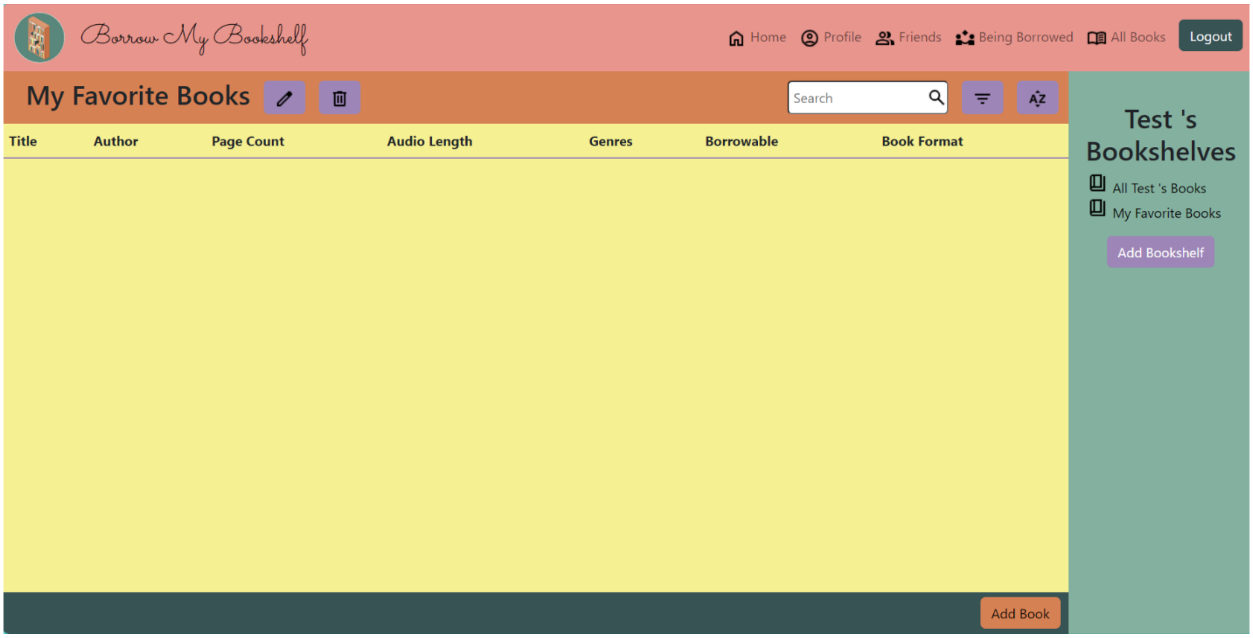
1. Once logged in, you will be shown your home page that contains all books that you have between all your bookshelves. Books can't be created from there though; they must be put on bookshelves. To make a bookshelf, click on the 'Add Bookshelf' button on your sidebar.



2. Enter a bookshelf name. The name must be unique.
3. Click "Create Bookshelf" to make the bookshelf, otherwise click the back button to cancel adding the bookshelf.

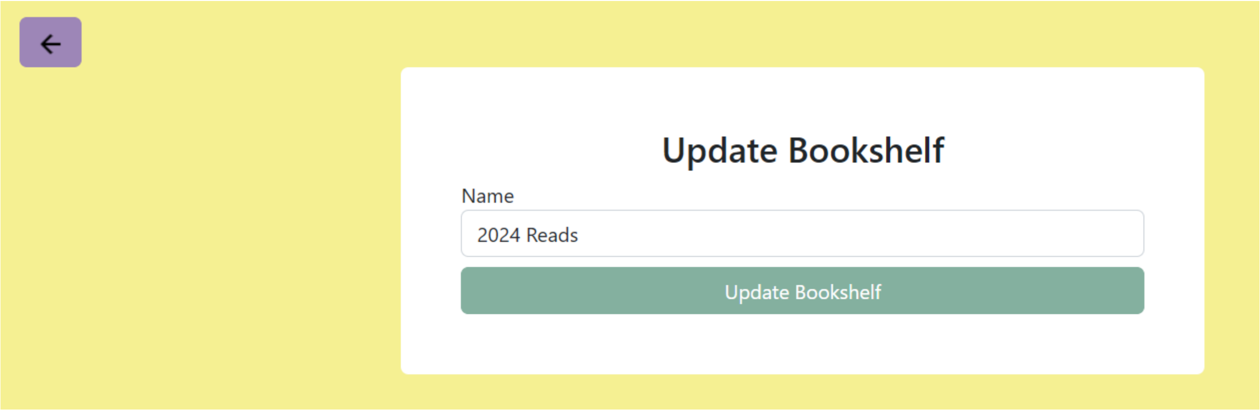


4. After creating your new bookshelf, it will appear in your sidebar, and you will be redirected to your new bookshelf.



Update a Bookshelf’s Name

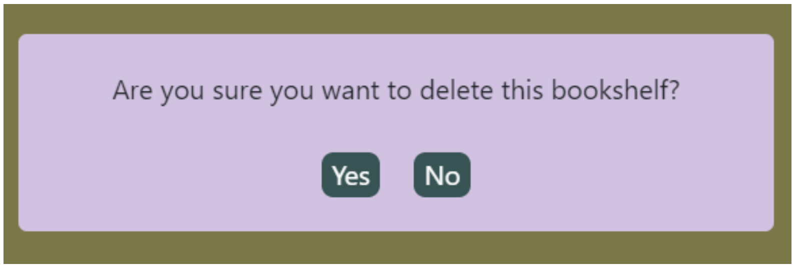
5. A bookshelf’s name can be edited at the top left by clicking the ‘Edit’ button which contains a pencil next to the bookshelf’s name, in this case ‘My Favorite Books’. Change the name on this page and click the ‘Update Bookshelf’ button or press the back button to go back to your bookshelf.



Name
2024 Reads

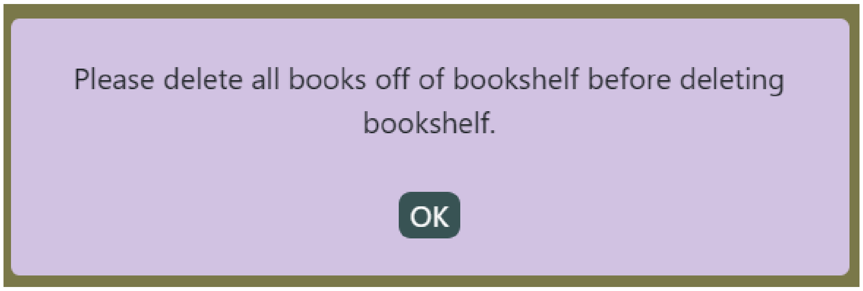
Delete a Bookshelf

1. A bookshelf can be deleted by hitting the trashcan at the top left next to the name of the shelf and choosing ‘Yes’ when the modal that makes sure you want to delete the bookshelf pops up.



Are you sure you want to delete this bookshelf?

2. You won’t be able to delete the bookshelf, until you remove all the books from it first.



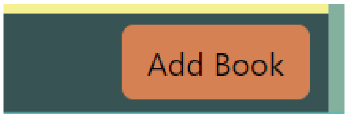
Please delete all books off of bookshelf before deleting bookshelf.

3. If you successfully delete the bookshelf, you will be redirected to ‘All User Books’.

Books

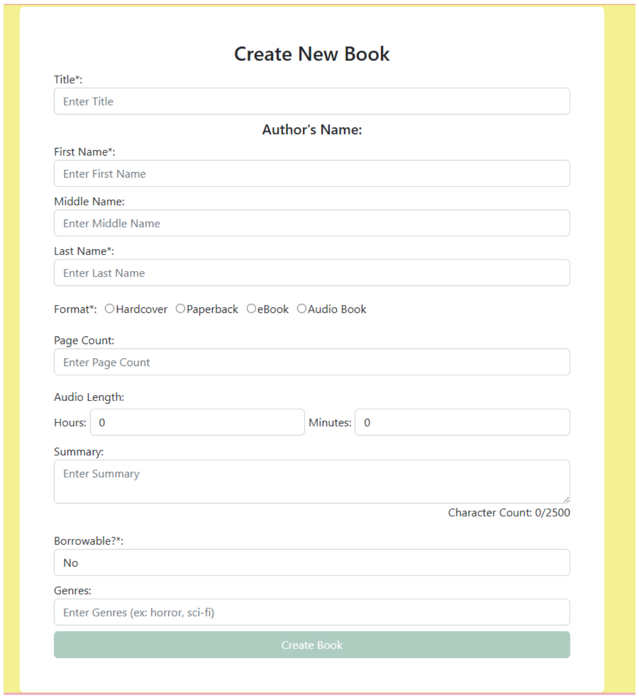
Creating a New Book

1. You can create a book by clicking the ‘Add Book’ button at the bottom right of the screen on the bookshelf footer. This will add the book directly on to the bookshelf that you have made it on.



2. This will take you to the ‘Create New Book’ page. Here you can fill out the title, first, middle, and last names of the author, what format you own the book in, the audio

length and page count, summary of the book, whether you will allow friends to borrow this book or not, and its genres. Fill in the fields with a * next to the label for the button to become enabled and for it to be added to your shelf.



Create New Book

Title*: Enter Title

Author's Name:

First Name*: Enter First Name

Middle Name: Enter Middle Name

Last Name*: Enter Last Name

Format*: ○Hardcover ○Paperback ○eBook ○Audio Book

Page Count: Enter Page Count

Audio Length:

Hours: 0 Minutes: 0

Summary: Enter Summary

Character Count: 0/2500

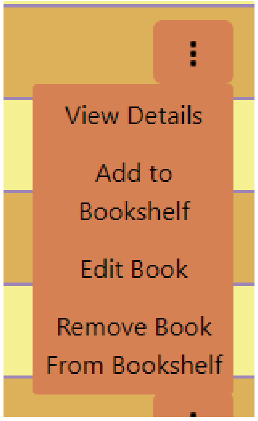
Borrowable?*: No

Genres: Enter Genres (ex: horror, sci-fi)

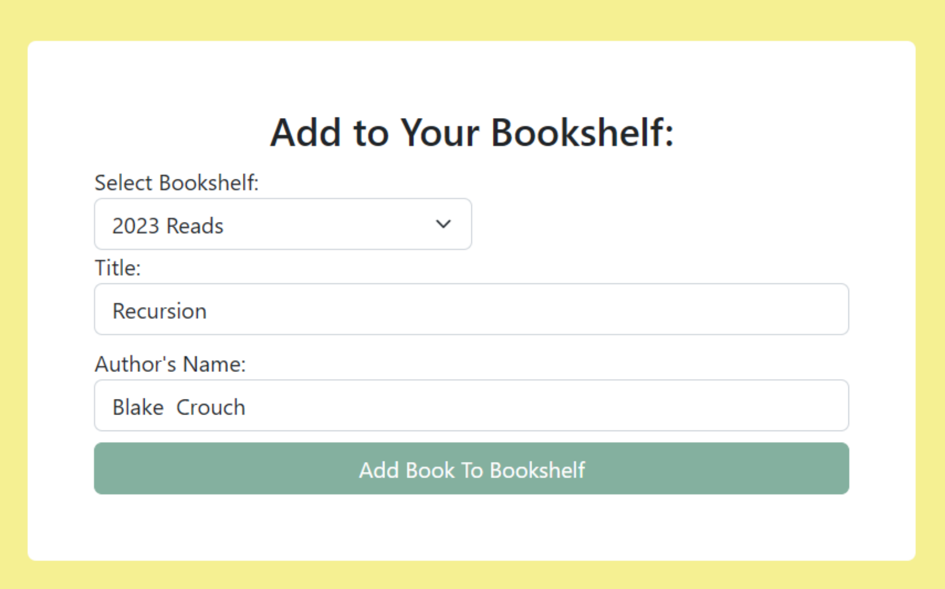
3. After you create the book, you will be redirected to the bookshelf that you added it to.

Adding a Book from Another Existing User Bookshelf

1. You can take the same book and duplicate it so that you can have it on multiple shelves. This allows you to have flexibility in sorting purposes.
2. To add an existing book to another bookshelf, click on the drop-down menu on the row containing the book you want to add to another bookshelf and click on the 'add to bookshelf' option.



3. This direct you to an ‘add to your bookshelf’ page that will allow you to choose which other bookshelf you want to see this book on. You cannot edit the title and author’s name from here.



Add to Your Bookshelf:

Select Bookshelf: 2023 Reads

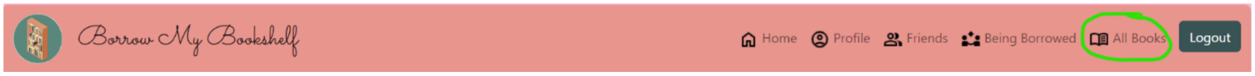
Title: Recursion

Author's Name: Blake Crouch

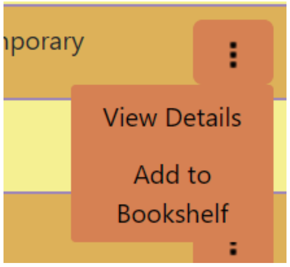
4. After you click ‘Add Book to Bookshelf’ it will redirect you to the bookshelf that you added that book to.

Adding a Book from All Books to Your Bookshelf

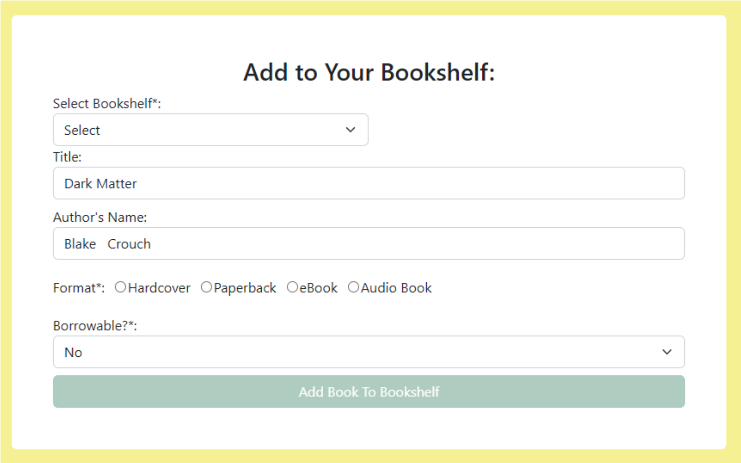
1. If a book already exists in the database, it can be found on the ‘All Books’ tab at the top of the page.



2. On this page, click the drop-down menu on the book you would like to add to your shelf and click on ‘add to bookshelf’.



3. This will take you to a different version of ‘Add to Your Bookshelf’ page. Since this is a generic book and not already a book you own or a different version of a book you own you need to fill out which bookshelf you want this book to be put on as well as what format you own it in and whether you are making the book ‘borrowable’ or not.


Add to Your Bookshelf:

Select Bookshelf*: Select

Title: Dark Matter

Author's Name: Blake Crouch

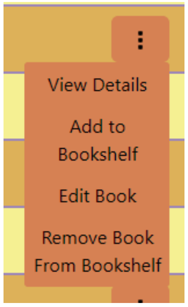
Format*: ○Hardcover ○Paperback ○eBook ○Audio Book

Borrowable?*: No

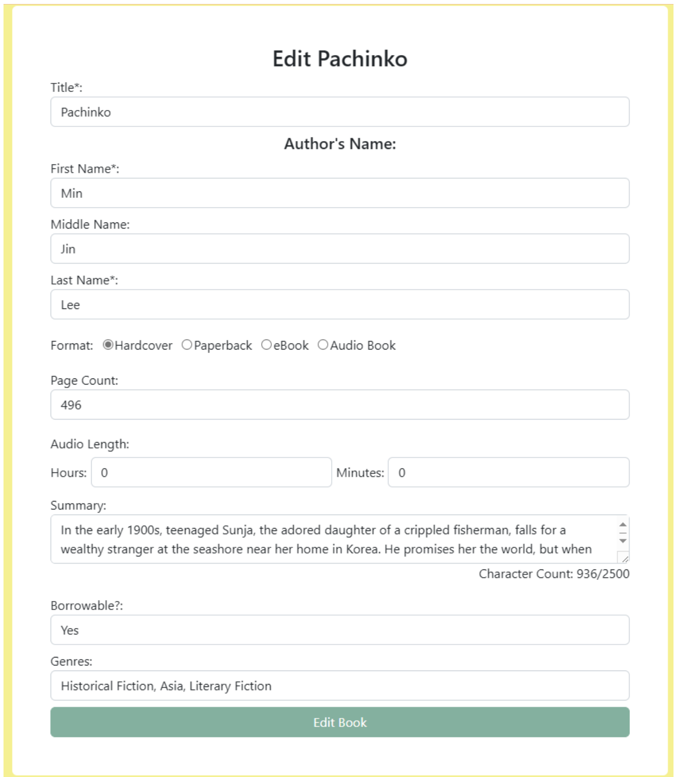
4. Once you hit ‘Add Book to Bookshelf’ you will be redirected to the bookshelf you added this book to.

Editing Books

1. To edit a book, click on the drop-down menu on the row that contains the book information that you want and click on ‘edit book’.



2. This will take you to an “Edit Book” page where it will autofill all the information that it has already. Simply replace what you need to change and click the ‘Edit Book’ button.



Edit Pachinko

Title*: Pachinko

Author's Name:

First Name*: Min

Middle Name: Jin

Last Name*: Lee

Format: ● Hardcover ○ Paperback ○ eBook ○ Audio Book

Page Count: 496

Audio Length:

Hours: 0 Minutes: 0

Summary: In the early 1900s, teenaged Sunja, the adored daughter of a crippled fisherman, falls for a wealthy stranger at the seashore near her home in Korea. He promises her the world, but when

Character Count: 936/2500

Borrowable?: Yes

Genres: Historical Fiction, Asia, Literary Fiction

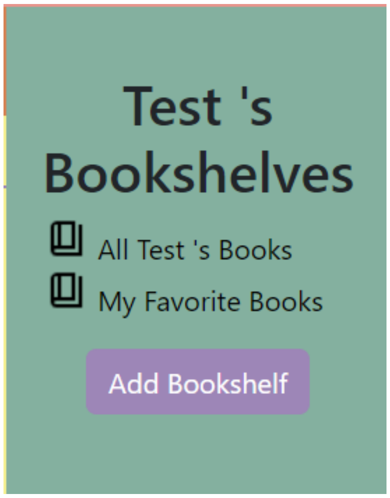
Edit Book

3. Once you submit your changes, you’ll be taken back to the page the user came from.

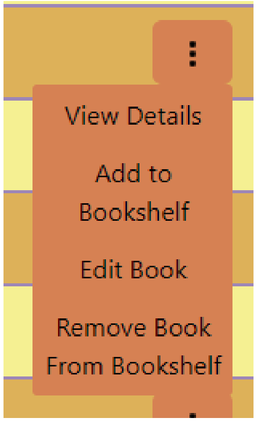
Removing Books From a Bookshelf

1. To delete a book, you need to be on a bookshelf that contains books and it cannot be the ‘All User’s Books’ as that isn’t a bookshelf but a collection of all books on

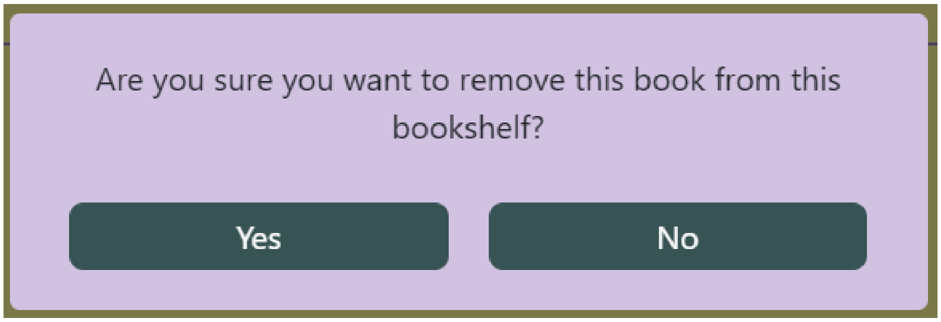
bookshelves. For example, you could choose "My Favorite Books" from the example below.



2. Once you have gotten to a bookshelf, find the book you wanted to remove from the shelf. Click on the drop-down menu connected to that book and click on 'Remove Book From Bookshelf'.



3. A pop up asking if you are sure you want to remove the book from the bookshelf will appear. Choose 'yes' to follow through with removing it from your shelf.

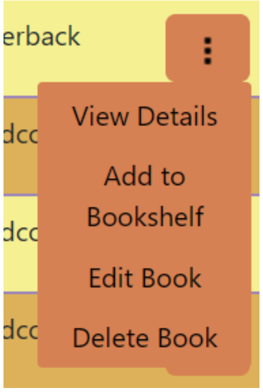


Are you sure you want to remove this book from this bookshelf?

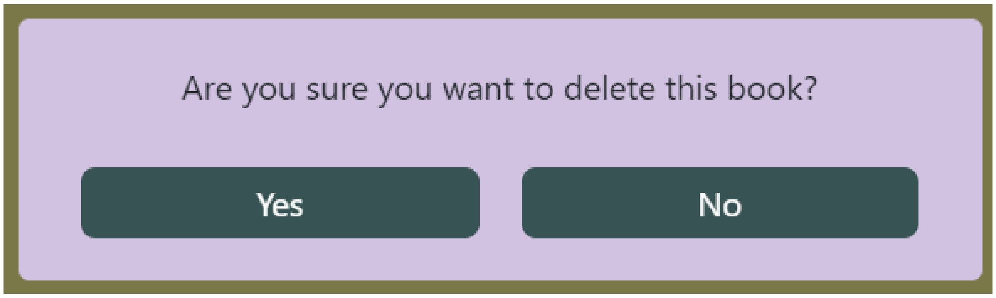
4. You will stay on the same bookshelf after selecting your answer.

Deleting Books from All User's Books

1. To delete a book officially from a user's collection you have to go to the sidebar to what says 'All User's Books'. From here click on the drop-down menu and click on 'Delete Book'.

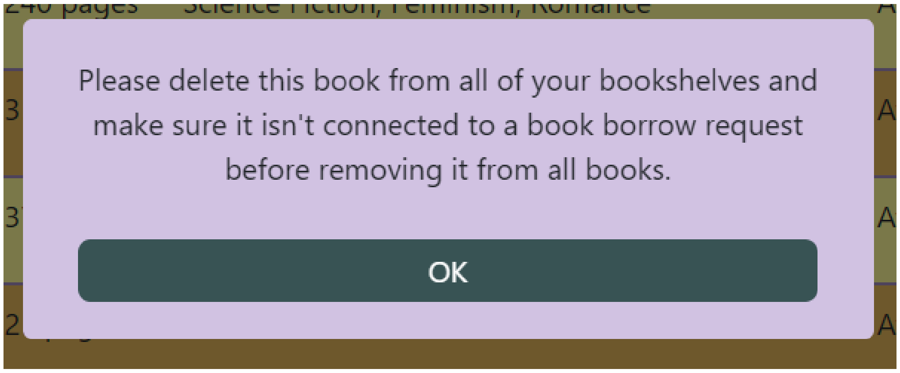


2. A pop up will appear asking if you are sure you want to delete the book. Press 'yes' to confirm deletion of the book.



Are you sure you want to delete this book?

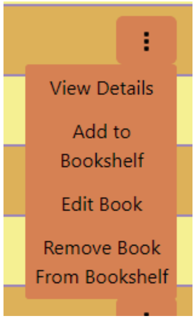
3. You cannot delete a book from All User's Books if a book still exists on a bookshelf or has a book borrowing status on it. Make sure you remove it from your bookshelves and delete any borrowing requests on it if you want to delete it.



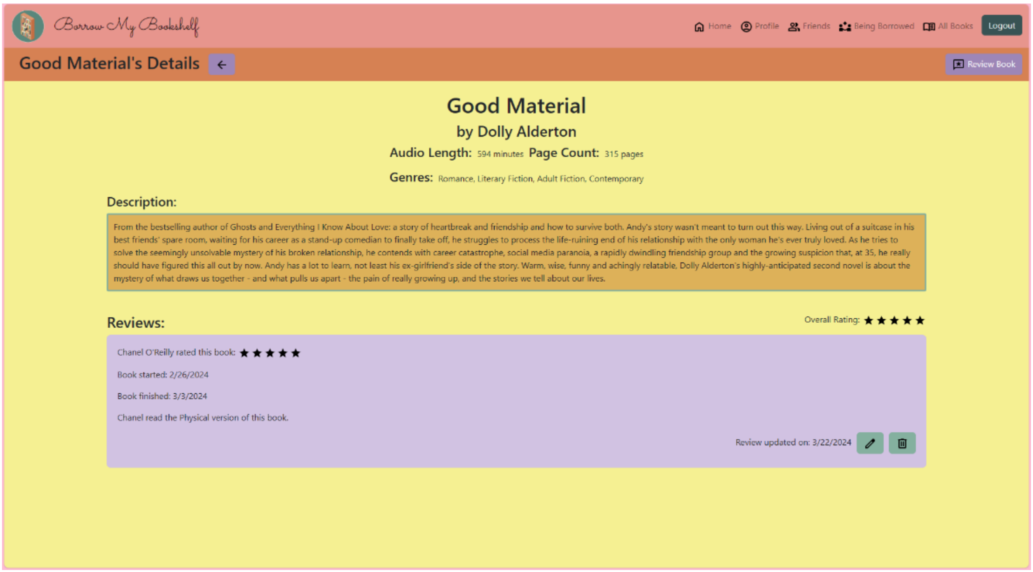
Please delete this book from all of your bookshelves and make sure it isn't connected to a book borrow request before removing it from all books.

View Books

1. On a book, you can click on the drop-down menu affiliated with the book and click on ‘View Details’.



2. This will display all the information available for that book including title, author, page count, audio length, genres, and any reviews that may exist on it.



Good Material's Details

Good Material

by Dolly Alderton

Audio Length: 594 minutes Page Count: 315 pages

Genres: Romance, Literary Fiction, Adult Fiction, Contemporary

Description:

From the bestselling author of Ghosts and Everything I Know About Love: a story of heartbreak and friendship and how to survive both. Andy's story wasn't meant to turn out this way. Living out of a suitcase in his best friends' spare room, waiting for his career as a stand-up comedian to finally take off, he struggles to process the life-ruining end of his relationship with the only woman he's ever truly loved. As he tries to solve the seemingly unsolvable mystery of his broken relationship, he contends with career catastrophe, social media paranoia, a rapidly dwindling friendship group and the growing suspicion that, at 35, he really should have figured this all out by now. Andy has a lot to learn, not least his ex-girlfriend's side of the story. Warm, wise, funny and achingly relatable, Dolly Alderton's highly-anticipated second novel is about the mystery of what draws us together - and what pulls us apart - the pain of really growing up, and the stories we tell about our lives.

Reviews:

Overall Rating: ★★★★★

Chanel O'Reilly rated this book: ★★★★★

Book started: 2/26/2024

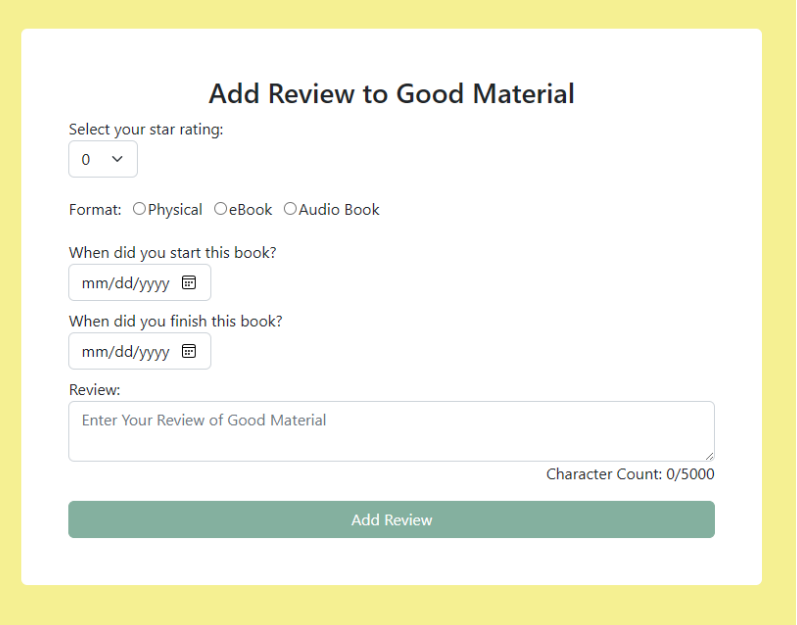
Book finished: 3/3/2024

Chanel read the Physical version of this book.

Review updated on: 3/22/2024

Adding Reviews

1. Clicking on the ‘Review Book’ button on the top right of the page will allow the user to leave a review on the book. Nothing on this page is required except for a star rating. When you are finished filling out what you’d like to include in your review, click on ‘Add Review’.



Add Review to Good Material

Select your star rating:

0

Format: ○Physical ○eBook ○Audio Book

When did you start this book?

mm/dd/yyyy

When did you finish this book?

mm/dd/yyyy

Review:

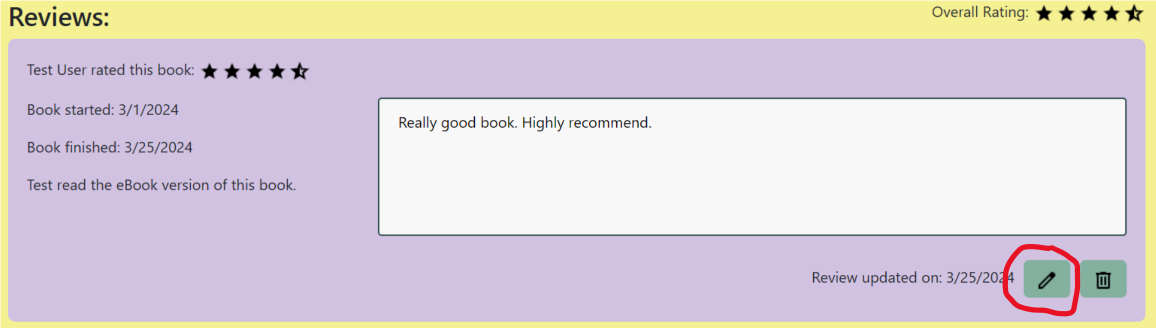
Enter Your Review of Good Material

Character Count: 0/5000

2. A single user can add as many reviews as wanted to a book, as sometimes people like to reread books or read different versions.

Updating Review

1. On the 'View Details' page of the book, if you have a review on the book, on the bottom right of the review a pencil edit button will be displayed. Click that if you'd like to edit your review.



Reviews:

Overall Rating: ★★★★⯪

Test User rated this book: ★★★★⯪

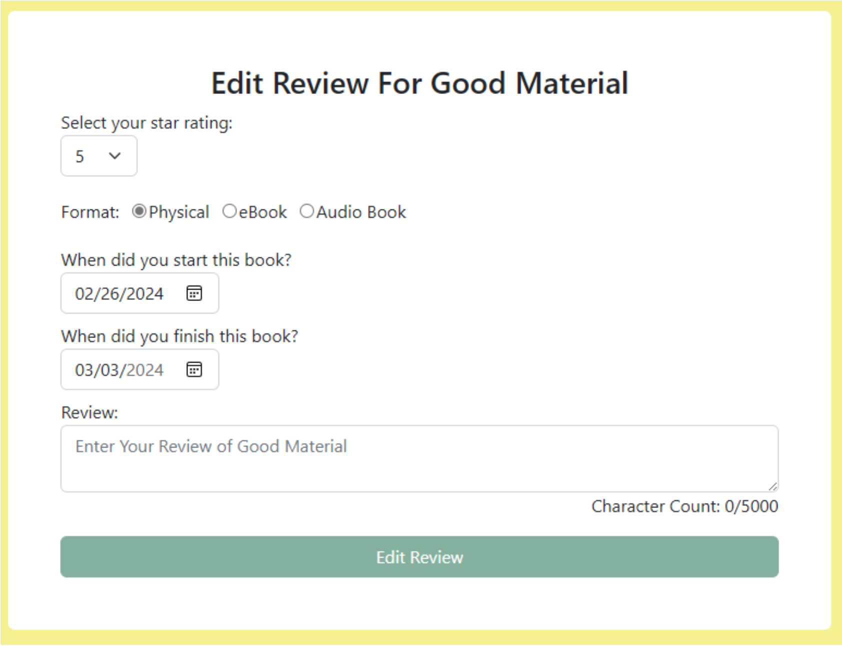
Book started: 3/1/2024

Book finished: 3/25/2024

Test read the eBook version of this book.

Really good book. Highly recommend.

Review updated on: 3/25/2024



Edit Review For Good Material

Select your star rating: 5

Format: ● Physical ○ eBook ○ Audio Book

When did you start this book? 02/26/2024

When did you finish this book? 03/03/2024

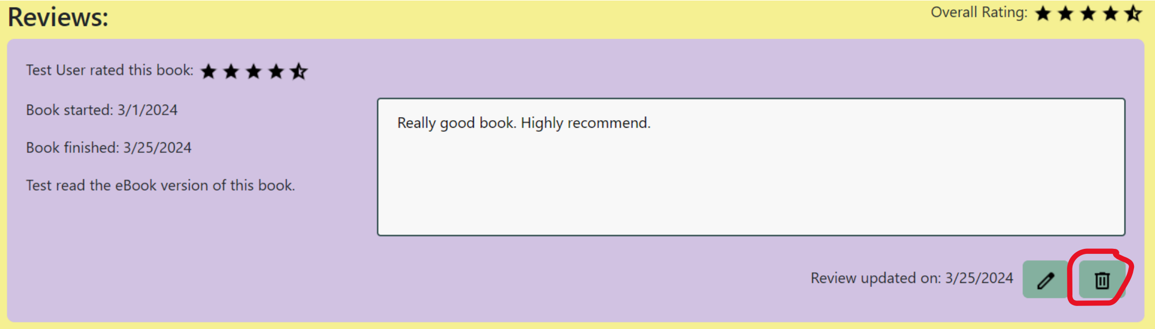
Review: Enter Your Review of Good Material

Character Count: 0/5000

2. Update the fields you'd like to update on your review and hit 'Update Review'.

Deleting a Review

1. In order to delete a review, click on the trashcan button on which of your reviews you'd like to delete.



Reviews:

Overall Rating: ★★★★⯪

Test User rated this book: ★★★★⯪

Book started: 3/1/2024

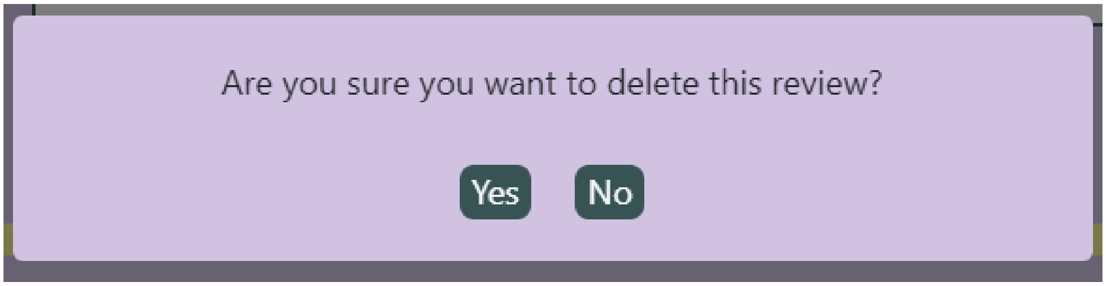
Book finished: 3/25/2024

Test read the eBook version of this book.

Really good book. Highly recommend.

Review updated on: 3/25/2024

2. After clicking delete, a pop-up modal will appear to make sure you'd like to delete your review. Click 'yes' if you want to delete your review.

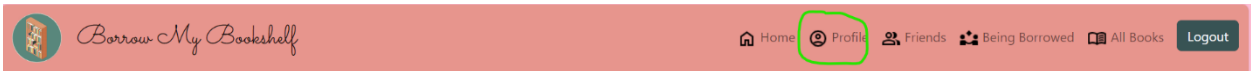


Are you sure you want to delete this review?

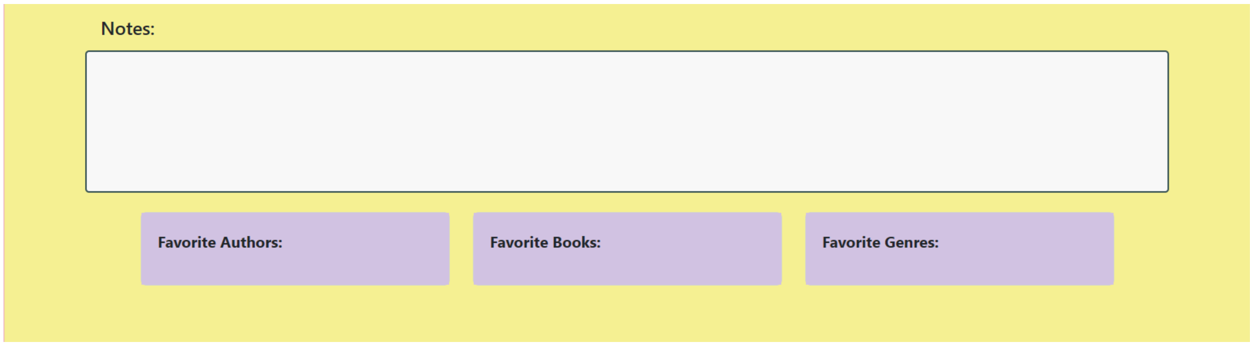
Profile

Viewing Profile

1. To view your user profile, click on the 'Profile' tab on the top navigation bar.

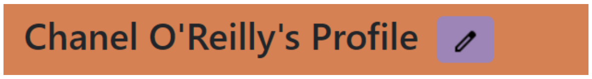


2. From here, you will be able to see your notes, favorite authors, favorite books, and favorite genres.

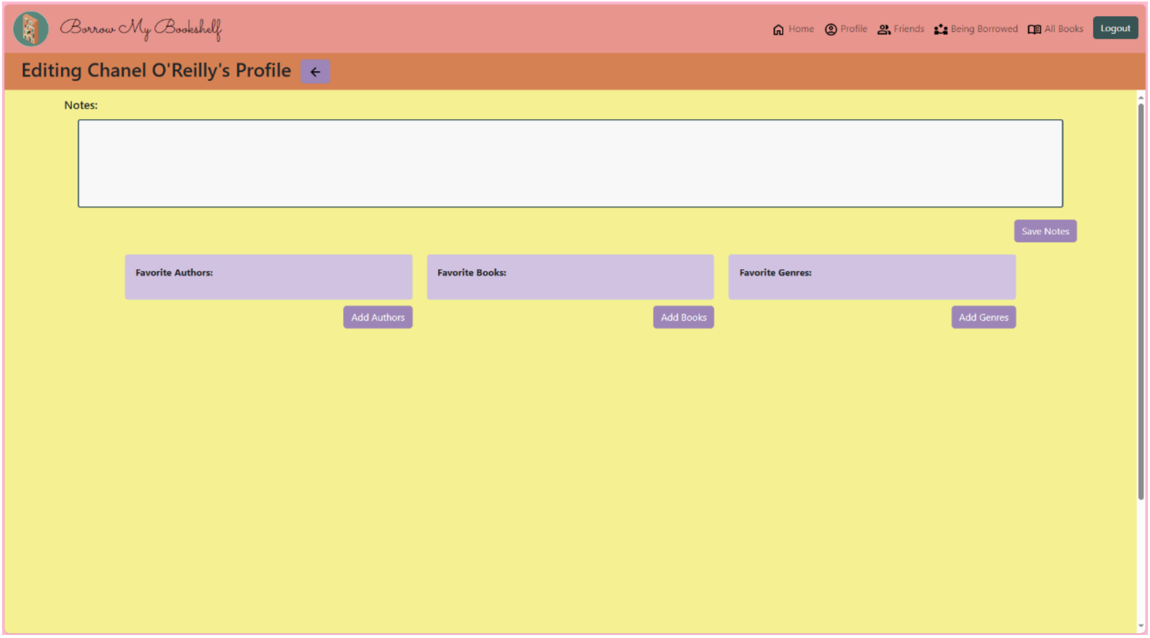


Editing Profile

1. Click on the ‘edit’ button shaped as a pencil on the navbar that has your name on it.

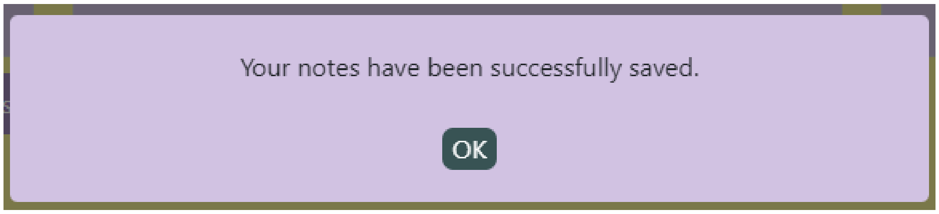


2. This will take you to the ‘Edit User’s Profile’ page. From here you will be able to add some notes and favorites.



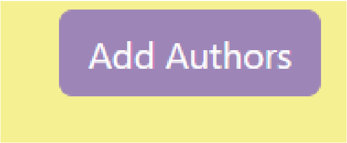
Adding Notes

1. To add notes, simply change what is in the text box and click ‘Save Notes’ when you’re done. You’ll know it worked if a modal pops up saying it worked.

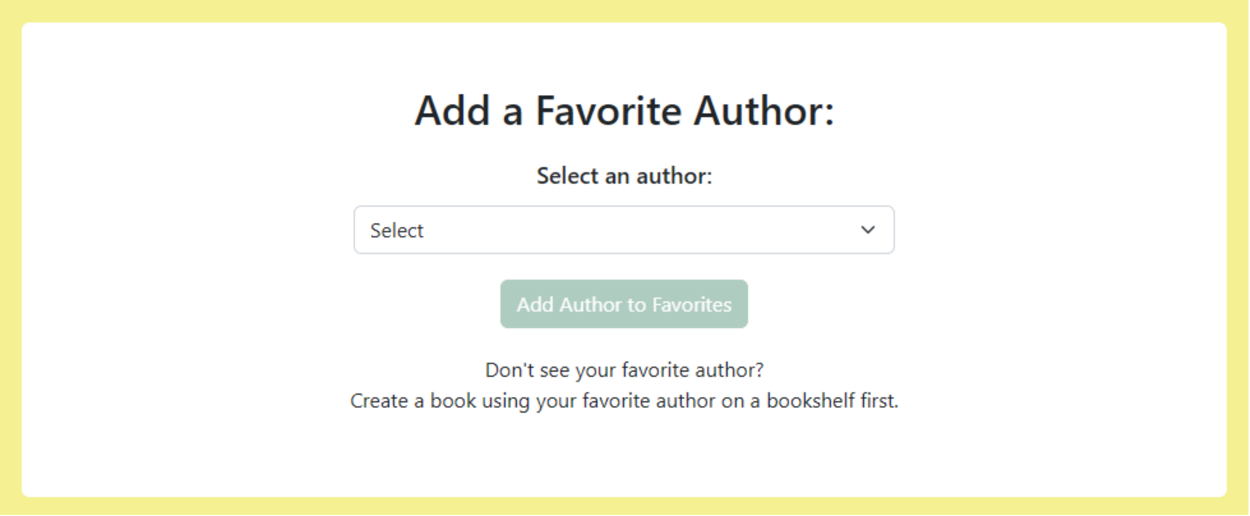

Your notes have been successfully saved.

Editing Favorites

1. To add a favorite author, book, or genre, the process will be about the same.
2. First, click on the 'add' button under the category you'd like to add a favorite too. In this example, we will be adding a 'favorite author'.



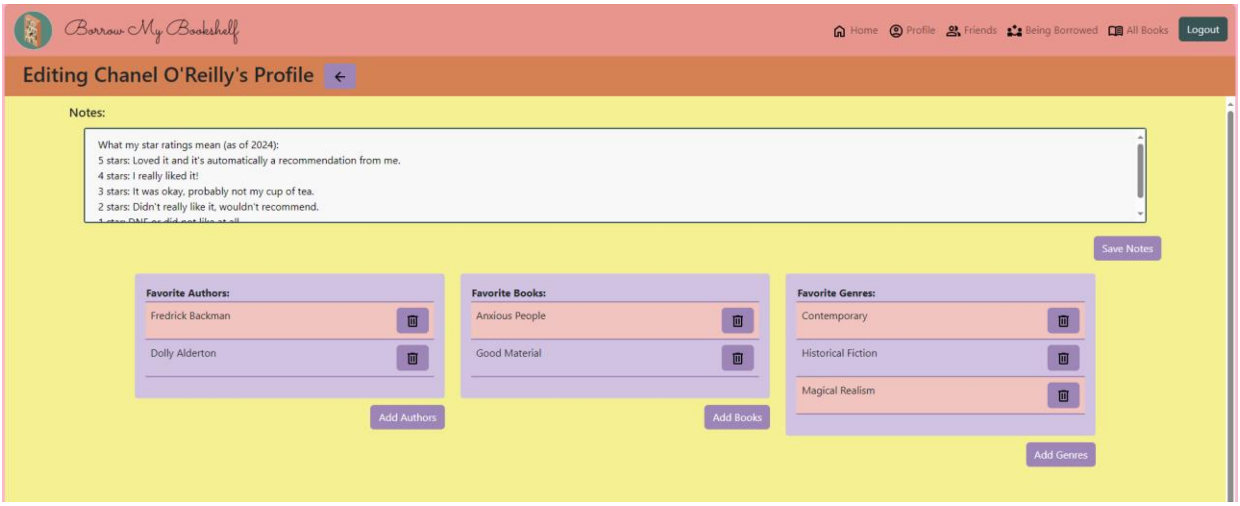
3. You will be redirected to this page where there will be a select box where you can choose from all the available authors, books, or genres in the database. If you don't see your favorite of that category, be sure to add a book that contains what you are looking for.


Add a Favorite Author:

Select an author: Select

Don't see your favorite author?
Create a book using your favorite author on a bookshelf first.

4. Choose what you were looking for and then click 'Add to Favorites'. After you do this, you will be redirected to editing profile page.



Editing Chanel O'Reilly's Profile

Notes:

What my star ratings mean (as of 2024):
5 stars: Loved it and it's automatically a recommendation from me.
4 stars: I really liked it!
3 stars: It was okay, probably not my cup of tea.
2 stars: Didn't really like it, wouldn't recommend.
1 star: DNF or did not like at all

Favorite Authors:
Fredrick Backman
Dolly Alderton

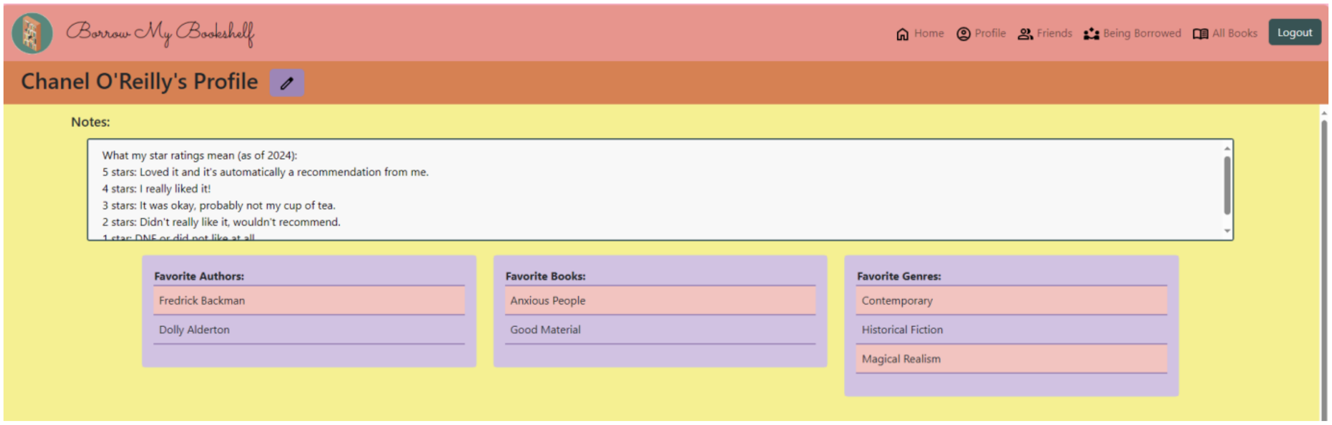
Favorite Books:
Anxious People
Good Material

Favorite Genres:
Contemporary
Historical Fiction
Magical Realism

5. If you want to delete a favorite, click the trashcan on the associated row.

6. Go back to viewing your profile by clicking the back arrow next to the 'Editing User's Profile' title in the navigation bar.

7. Now you will be able to view your profile with all your edits.



Chanel O'Reilly's Profile

Notes:

What my star ratings mean (as of 2024):
5 stars: Loved it and it's automatically a recommendation from me.
4 stars: I really liked it!
3 stars: It was okay, probably not my cup of tea.
2 stars: Didn't really like it, wouldn't recommend.
1 star: DNF or did not like at all

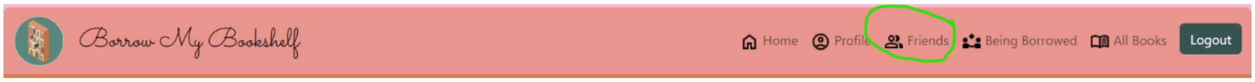
Favorite Authors:
Fredrick Backman
Dolly Alderton

Favorite Books:
Anxious People
Good Material

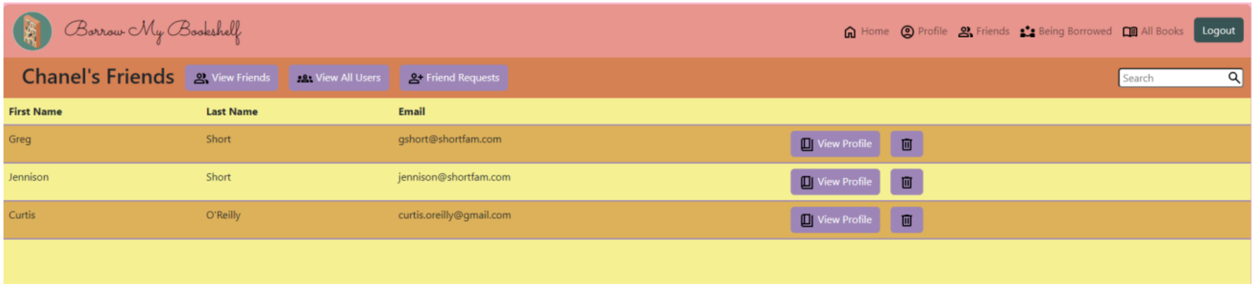
Favorite Genres:
Contemporary
Historical Fiction
Magical Realism

Friends

1. To get to your 'Friends' page. Click on the top navigation bar on the 'Friends' button.



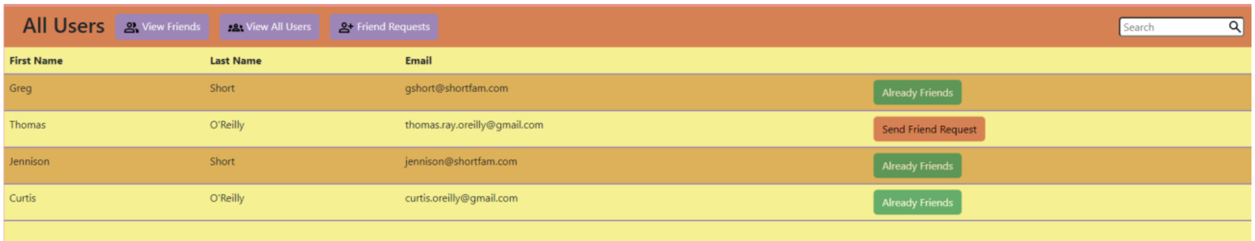
2. When you get to your friend's page, it will show all your current friends.



First Name	Last Name	Email	
Greg	Short	gshort@shortfam.com	View Profile
Jennison	Short	jennison@shortfam.com	View Profile
Curtis	O'Reilly	curtis.oreilly@gmail.com	View Profile

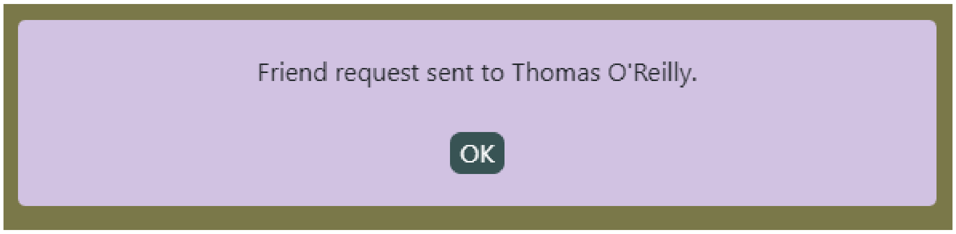
Adding Friends

1. To add a friend, click on the ‘View All Users’ button. From here, you’ll be able to view all users on the application.



First Name	Last Name	Email	
Greg	Short	gshort@shortfam.com	Already Friends
Thomas	O'Reilly	thomas.ray.oreilly@gmail.com	Send Friend Request
Jennison	Short	jennison@shortfam.com	Already Friends
Curtis	O'Reilly	curtis.oreilly@gmail.com	Already Friends

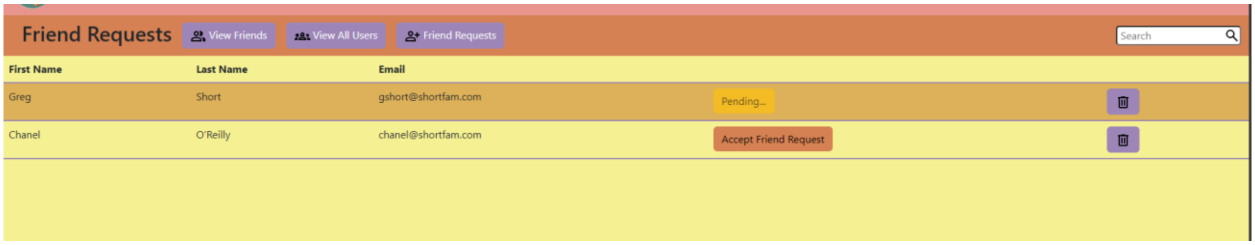
2. As you can see, people you are already friends with will say ‘Already Friends’, people who you aren’t friends with will have a ‘Send Friend Request’ button. If you click ‘Send Friend Request’, you will be notified that you sent a friend request.


Friend request sent to Thomas O'Reilly.

3. If there is already a friend request sent from you, it will say ‘Pending…’ and if someone sent you a friend request, it will say 'Accept Friend Request’. This can be seen better from the ‘Friend Requests’ page. Click there to see all current requests between you and another user.

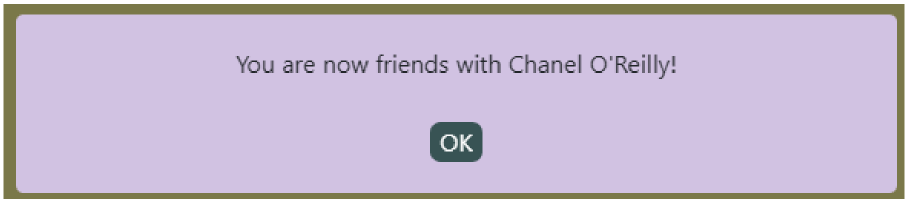
Friend Requests

1. From the ‘Friend Requests’ page, you will be able to see all requests between you and another user. You can delete a request by clicking the trash can icon on the right in the row containing the user you don’t want to be friends with, or you can accept the friend request by clicking ‘Accept Friend Request’.



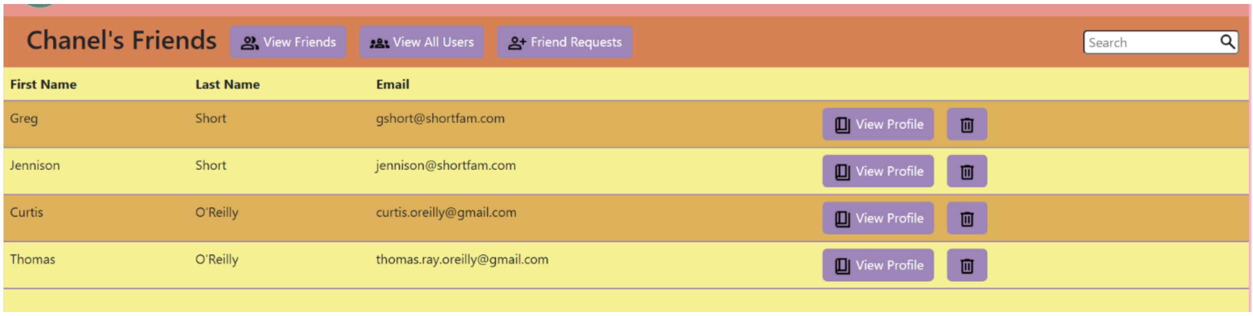
First Name	Last Name	Email		
Greg	Short	gshort@shortfam.com	Pending...	🗑
Chanel	O'Reilly	chanel@shortfam.com	Accept Friend Request	🗑

2. If you accept the friend request, a pop up will let you know that it has been accepted, the request will disappear, and the friend will be added to your ‘View Friends’ page.


You are now friends with Chanel O'Reilly!

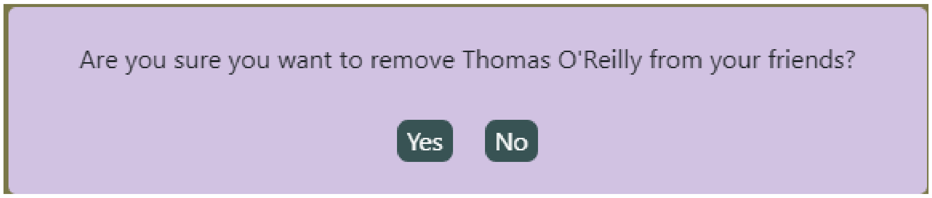
View Friends

1. The ‘View Friends’ page is the page that initially loads when you click on ‘Friends’ in the top navigation bar. From here, you will be able to see all your friends that either you or they have accepted a friend request.



First Name	Last Name	Email		
Greg	Short	gshort@shortfam.com	View Profile	🗑
Jennison	Short	jennison@shortfam.com	View Profile	🗑
Curtis	O'Reilly	curtis.oreilly@gmail.com	View Profile	🗑
Thomas	O'Reilly	thomas.ray.oreilly@gmail.com	View Profile	🗑

2. On this page you can remove a friend by clicking the trashcan icon affiliated with their row. Before officially removing them from your friends list, a modal will pop-up and ask if you want to delete your friend. Click ‘yes’ to remove them from the list or ‘no’ to keep them as a friend.

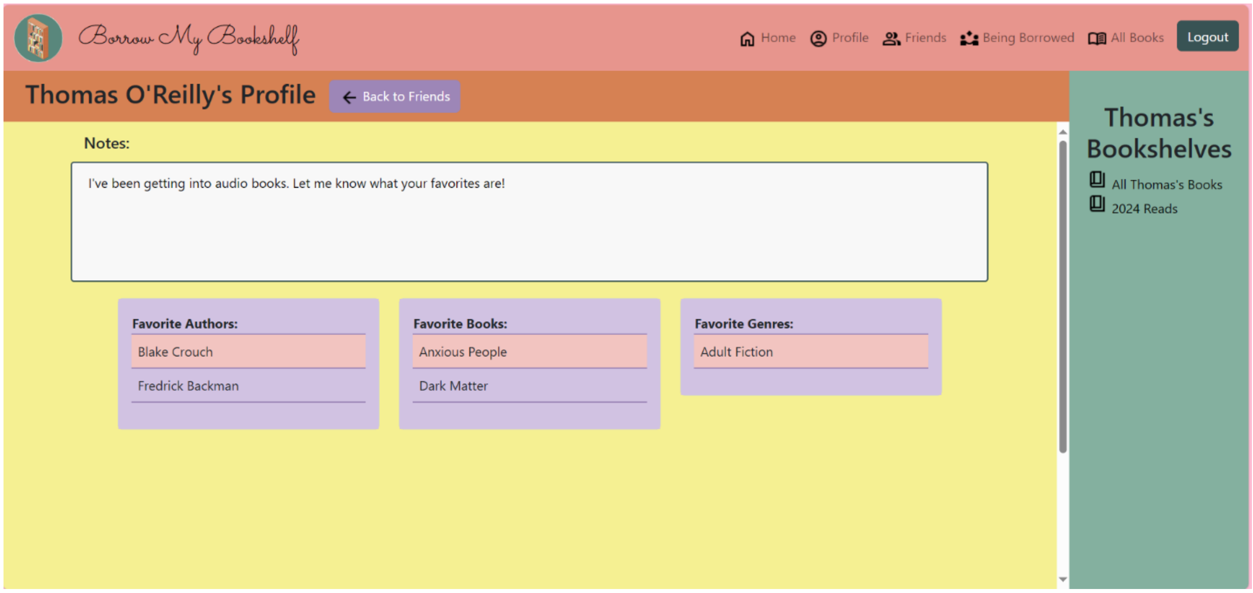


Are you sure you want to remove Thomas O'Reilly from your friends?

3. You can also click the “View Profile” button to view your friend’s profile and bookshelves.

Friend’s Profile

1. On your friend's profile you can see their user profile which includes a note, favorite authors, favorite books, and favorite genres. You can go back to your ‘All Friends’ page by clicking on the ‘Back to Friends’ button on the navbar.



Thomas O'Reilly's Profile

Notes:

I've been getting into audio books. Let me know what your favorites are!

Favorite Authors:	Favorite Books:	Favorite Genres:
Blake Crouch	Anxious People	Adult Fiction
Fredrick Backman	Dark Matter	

Thomas's Bookshelves

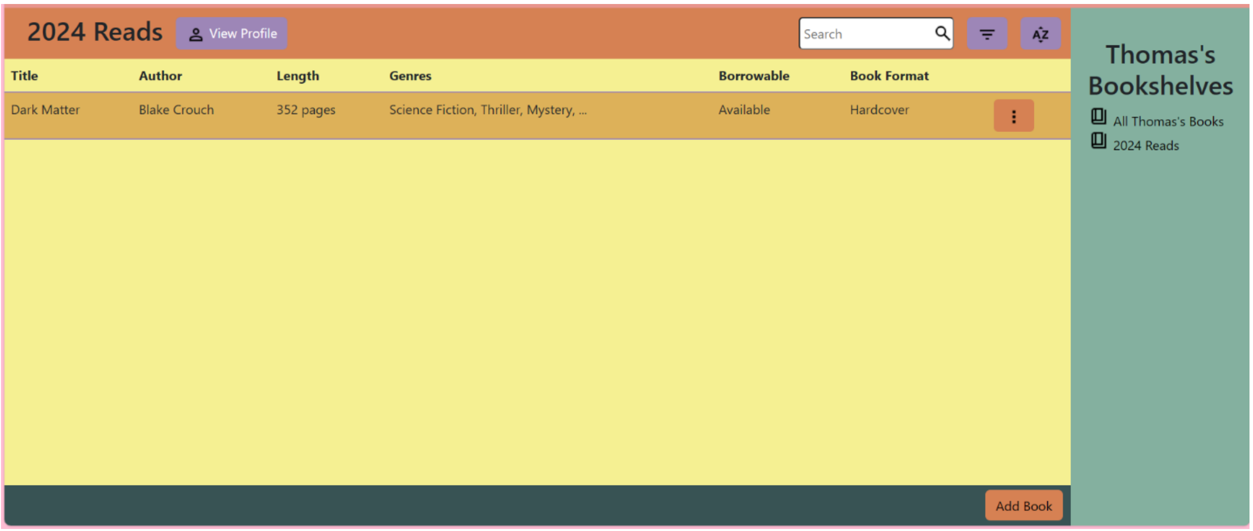
- All Thomas's Books
- 2024 Reads

2. On the right, your bookshelves get replaced with your friend’s bookshelves. You can click on their bookshelves to look through their books, see what they own, and send a borrow request.

Borrowing

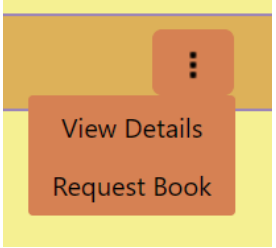
Borrow from Friend’s Bookshelves

1. If you go to your friend’s profile, you will be able to view the books on your friend’s bookshelves.

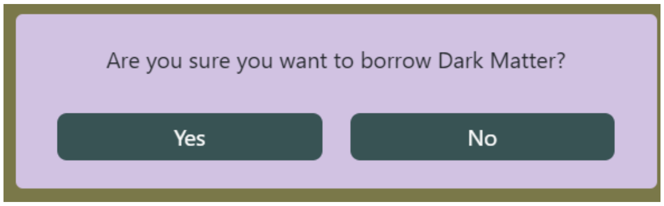


Title	Author	Length	Genres	Borrowable	Book Format
Dark Matter	Blake Crouch	352 pages	Science Fiction, Thriller, Mystery, ...	Available	Hardcover

2. From there you can click on the drop-down menu on the associated row and click ‘Request Book’, unless the book is specifically marked as ‘Not Borrowable’. If the book is ‘Pending’ or ‘Borrowed’ the owner will receive your request after the current borrowing request on the book has been resolved.

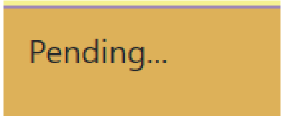


3. If you click on 'Request Book', you'll get a pop up asking if you are sure you'd like to borrow that book.



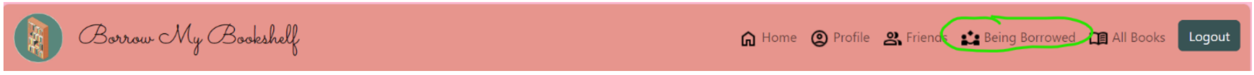
Are you sure you want to borrow Dark Matter?

4. If 'yes' is selected, the borrowable status will change from 'Available' to 'Pending'.

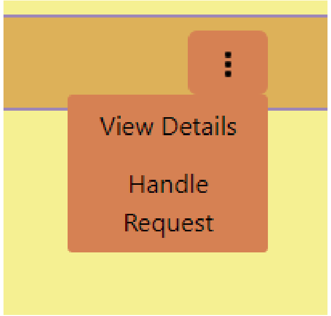


Being Borrowed Page

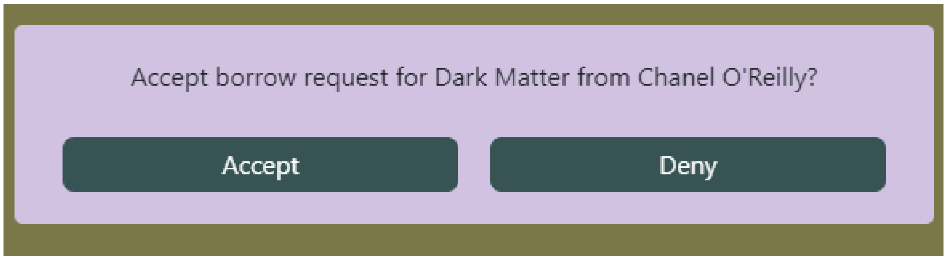
1. Through the borrowing process the status on books will change depending on where they are in the process. They can be viewed on the book that the request is on, but all books with a borrowing status tied to you as either the borrower or lender can be easily viewed from the 'Being Borrowed' tab in the top navigation bar.



2. From here, if you see a 'Pending…' status on a book you own, click on the drop-down menu connected to that book and click on 'Handle Request'.

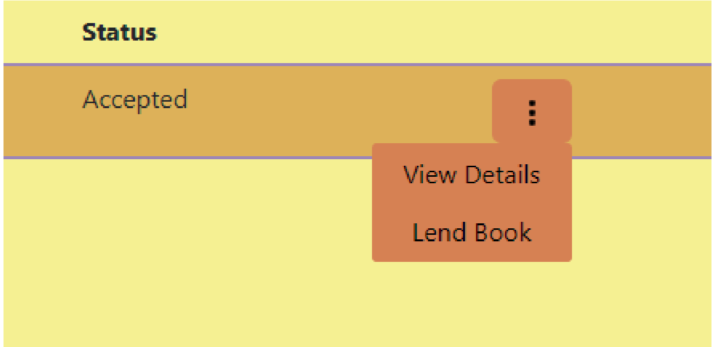


3. When you click on ‘Handle Request’ a pop-up modal will appear asking if you will accept or deny the borrow request.



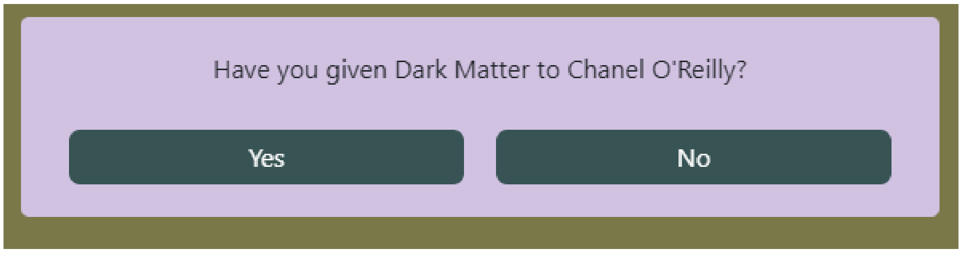
Accept borrow request for Dark Matter from Chanel O'Reilly?

4. If you press on ‘Deny’ it will delete the book request. If you press on 'Accept’ it will accept the book request and change the borrowable status from ‘Pending’ to ‘Accepted’.



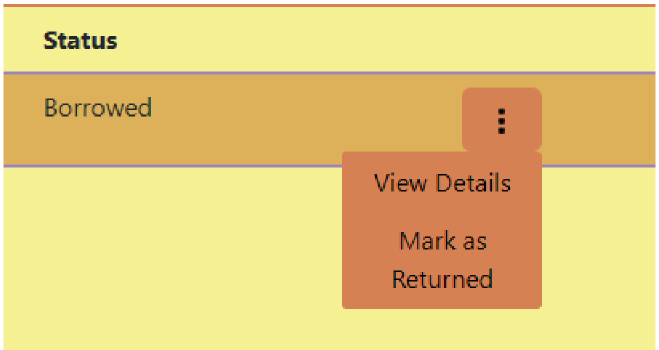
Status
Accepted

5. Once the status is ‘Accepted’, the lender can press on the drop-down menu to select ‘Lend Book’. This will result in a pop-up asking if the book has been given to the borrower.



Have you given Dark Matter to Chanel O'Reilly?

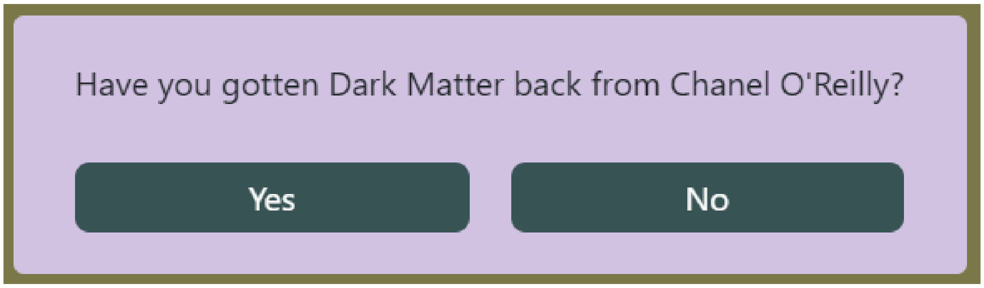
6. If the user hits ‘Yes’, then the status of the request will turn from ‘Accepted’ to ‘Borrowed’.



Status

Borrowed

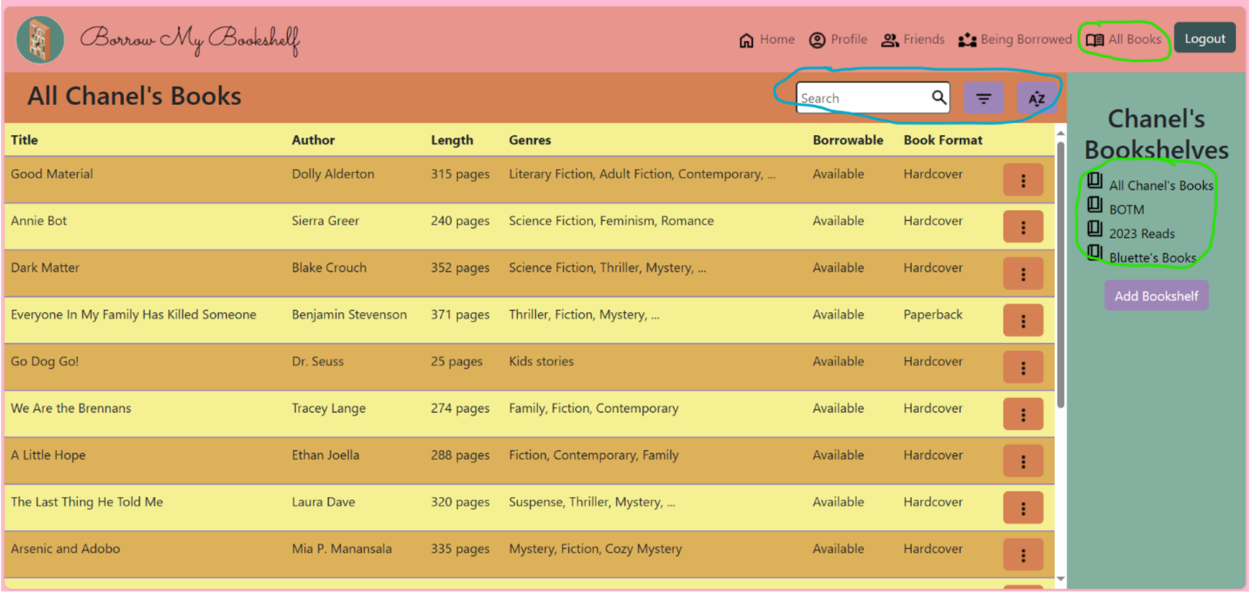
7. Once the book has been finished and returned to the owner of the book, the user can go to the drop-down menu and select the ‘Mark as Returned’ button. This will result in a pop-up asking if you have gotten the book back from the borrower. If ‘Yes’ is selected, the book request will be cleared out of the lender and borrowers feed as the whole process has been completed. If there is another pending request for the book, the owner will immediately see the request after this one is resolved completely.



Have you gotten Dark Matter back from Chanel O'Reilly?

Creating Reports

1. You can search, filter, and/or sort books in three different places. All Books, All User’s Books, and on all the user’s bookshelves. Shown in the below picture in green is how you get to those different areas and in teal is where you can choose to search, sort, or filter.



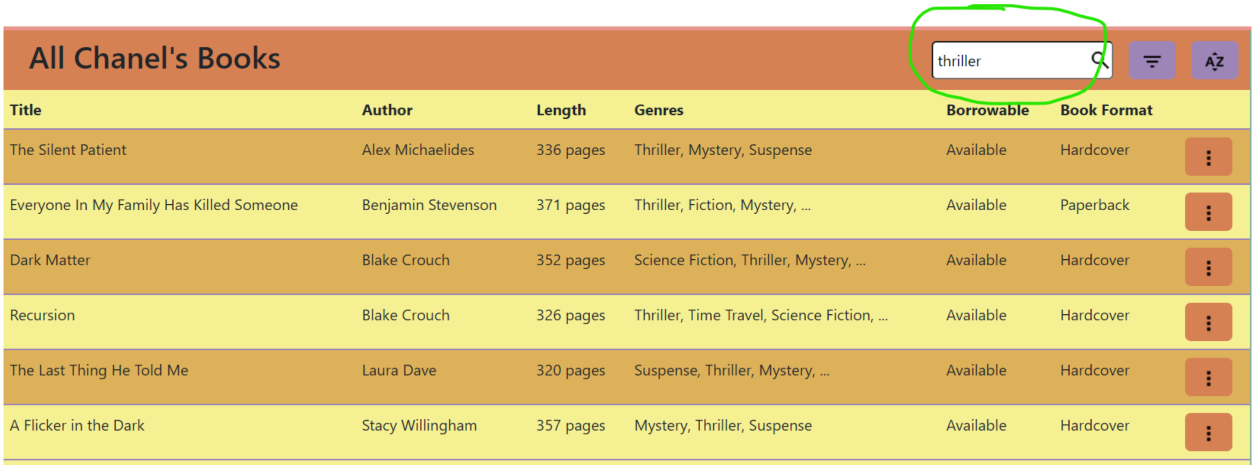
All Chanel's Books

Title	Author	Length	Genres	Borrowable	Book Format
Good Material	Dolly Alderton	315 pages	Literary Fiction, Adult Fiction, Contemporary, ...	Available	Hardcover
Annie Bot	Sierra Greer	240 pages	Science Fiction, Feminism, Romance	Available	Hardcover
Dark Matter	Blake Crouch	352 pages	Science Fiction, Thriller, Mystery, ...	Available	Hardcover
Everyone In My Family Has Killed Someone	Benjamin Stevenson	371 pages	Thriller, Fiction, Mystery, ...	Available	Paperback
Go Dog Go!	Dr. Seuss	25 pages	Kids stories	Available	Hardcover
We Are the Brennans	Tracey Lange	274 pages	Family, Fiction, Contemporary	Available	Hardcover
A Little Hope	Ethan Joella	288 pages	Fiction, Contemporary, Family	Available	Hardcover
The Last Thing He Told Me	Laura Dave	320 pages	Suspense, Thriller, Mystery, ...	Available	Hardcover
Arsenic and Adobo	Mia P. Manansala	335 pages	Mystery, Fiction, Cozy Mystery	Available	Hardcover

2. Note that all of these features can be combined to display the books you want in the order you want to view them in.

Search

1. All columns on a page containing books are searchable except for ‘borrowable’ and ‘book format’. Those options can be filtered though.
2. Simply type in what you are searching for, and all matching options should come up.



All Chanel's Books

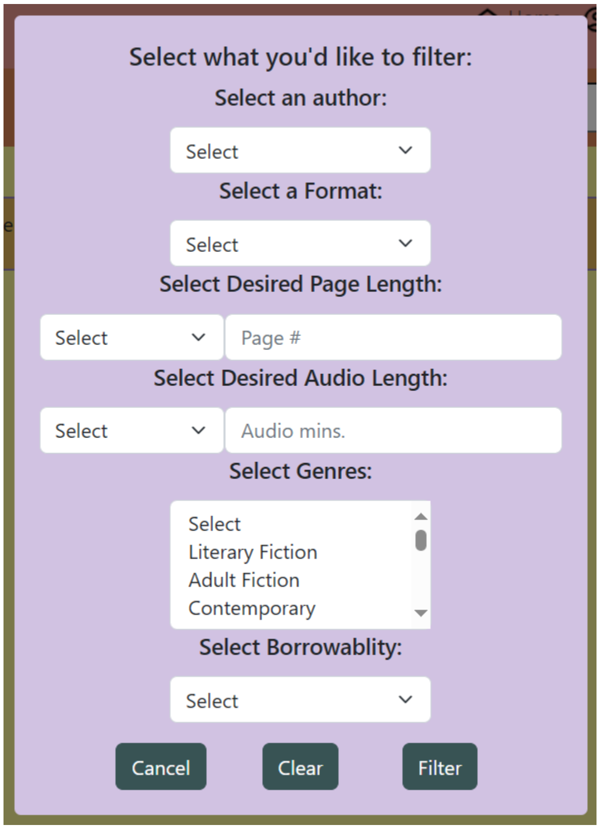
thriller

Title	Author	Length	Genres	Borrowable	Book Format
The Silent Patient	Alex Michaelides	336 pages	Thriller, Mystery, Suspense	Available	Hardcover
Everyone In My Family Has Killed Someone	Benjamin Stevenson	371 pages	Thriller, Fiction, Mystery, ...	Available	Paperback
Dark Matter	Blake Crouch	352 pages	Science Fiction, Thriller, Mystery, ...	Available	Hardcover
Recursion	Blake Crouch	326 pages	Thriller, Time Travel, Science Fiction, ...	Available	Hardcover
The Last Thing He Told Me	Laura Dave	320 pages	Suspense, Thriller, Mystery, ...	Available	Hardcover
A Flicker in the Dark	Stacy Willingham	357 pages	Mystery, Thriller, Suspense	Available	Hardcover

3. As you can see, typing in ‘thriller’ in the search bar returned all books that contained the word ‘thriller’.

Filter

1. To the right of the search bar is the filter button. When you click on the filter button, your filter options will appear in a modal.



Select what you'd like to filter:

Select an author:

Select

Select a Format:

Select

Select Desired Page Length:

Select | Page #

Select Desired Audio Length:

Select | Audio mins.

Select Genres:

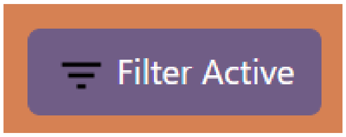
Select
Literary Fiction
Adult Fiction
Contemporary

Select Borrowablity:

Select

2. You can choose multiple filter options. When you are done selecting what you’d like to see, hit ‘Filter’ and all the books fulfilling those requirements will appear.

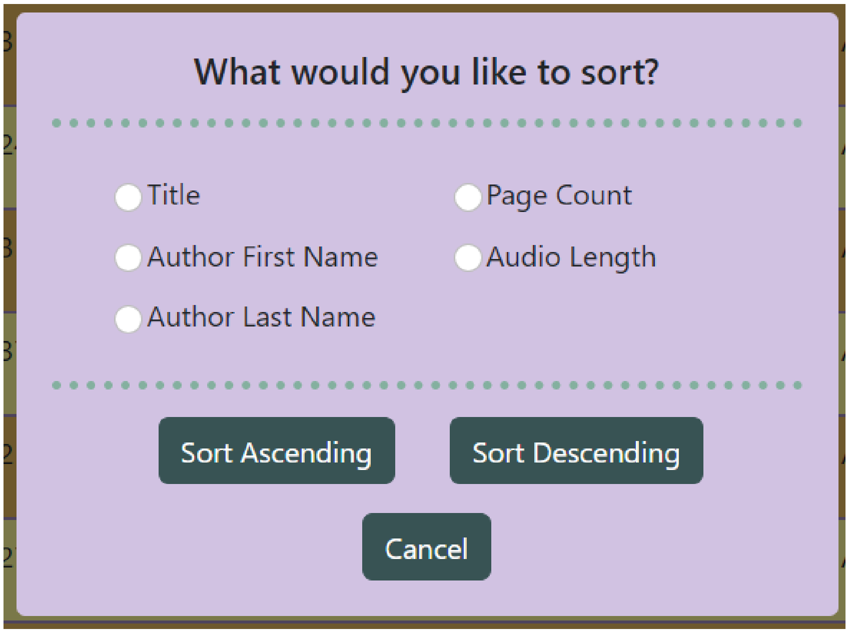
3. You can tell if the filter is on when the filter button is darkened and it says ‘Filter Active’.



4. You can disable the filter by clicking back onto the filter button and clicking ‘clear’.

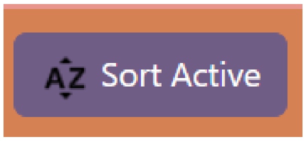
Sorting

1. The last option in the orange navigation bar on the very right is the Sort button. When you click it, a menu will appear with various things that you can sort by either ascending or descending.



What would you like to sort?

2. When the sorting method is on, the button is darkened and says ‘Sort Active’.



3. In order to remove the sort filter, click back onto the sort button and click ‘cancel’.

Panopto Video Link

Link: Borrow My Bookshelf Capstone - Chanel O'Reilly (panopto.com)

Second Link for Deletion of Books: Deleting Books - Borrow My Bookshelf - Chanel O'Reilly (panopto.com)