Team Perrinie - Perrin Chhagan & Stephanie Benamati (Catlett) Group 3

URL: http://flip1.engr.oregonstate.edu:2526/activeLoans

Summary:

Step 1 Changes:

- Report typos fixed and variable/Entity names changed to be consistent
- Made the 1:M relationship between Books and Genre Nullable (you can now input null as the books genre).

Step 2 Changes:

- Updated overview to make it easier to identify the problem the database is solving
- Removed ageRestricted attribute from Genres

Step 3 Changes:

- CSS styling changes for form titles to make them stand out more.
- Changing text on submit buttons added to be reflective of the forms (‘delete’ vs ‘submit’)
- Organized all the pages to be in the same order in regards to the layout of the table and forms (Add->Search->Update)
- Added visuals to dropdown items to make it clearer that the selection can be changed
- Fixed error where tab ‘titles’ was incorrectly labeled

Step 4 Changes:

- Fixed syntax errors in data manipulation queries
- Connected UI to database

Step 5 Changes:

- Tab order changed in nav bar so Authors and Books would come before Authored Works
- Pre-populated dropdowns for foreign keys
- Changed input typing so only numbers allowed in some places instead of generic ‘text’
- Added minimum values to some fields
- Made some fields required
- Changed the authored works page to display name and book title alongside IDs to make more human readable
- Standardized placeholder texts
- Standardized dropdown menu formats
- Added update function to books page
- Added human readable values (non-numbers) to tables where foreign keys were present

Reflection:

The initial version of our website was a basic html layout with the skeleton of required pages. The first major change we made was switching from strictly html/css styling to bootstrap, which modernized the appearance of the site significantly. The second major change was when the UI needed to be connected to the database. When this occurred, we traded out static HTML for a handlebars framework.

Other than these two things, not much has shifted from the initial planning stage. Some naming conventions were changed, as well as some quality-of-life changes for the user at the suggestion of our peers (i.e. styling changes for easier readability, descriptive text placeholders and button text, etc.), but most of our progression was smooth. In fact, it was thanks to the efforts of our peers in finding hidden bugs and pointing out usability issues that allowed us to progress so smoothly overall, not letting any one thing build up and become a larger issue.

In conclusion, as a result of our strong starting design and the efforts of reviewers along the way, we have crafted a workable website to be proud of.

Project Outline:

The Public Library of Perrinie is one of several mid-sized libraries in the city of Fictopia. Fictopia is home to around 500,000 people, and on average, 5-10% of those people visit this particular library semi-regularly. The library contains around 50,000 books with around 25% of that amount being duplicate copies of the various titles in their collection. Each year they archive the least popular books or books that have been damaged (taking them out of circulation) to make room for the newest releases. As such, the number of books on their shelves stays approximately the same even though the titles available are frequently changing. With all the changes to library inventory, it's important to keep track of what books are available to be borrowed. This database-driven website will be used by library clerks to record the ***Available Books***(***Author*** & ***Genre*** info included) that can be ***Actively Loaned*** to ***Library Patrons***.

Database Outline:

Authors: records the authors of the books the library holds in their collection

- authorID: int, auto_increment, unique, not NULL, PK
- firstName: varchar, not NULL
- lastName: varchar, not NULL
- Relationship: a M:M relationship between books and authors is implemented by combining authorID(FK) and bookID(FK) in the Authored Works intersection table (Below).
    - A book can have multiple authors, and an author can have written multiple books.

Authored_Works: an intersection table for authors and books

- authorID: int, not NULL, FK/PK
- bookID: int, not NULL, FK/PK
- Relationship: the intersection of the M:M relationship for Authors and Books. Combines the foreign keys of both tables to make a primary key and join the two tables' data together.

Books: records the books the library has in their collection

- bookID: int, auto_increment, unique, not NULL, PK
- genreID: int, FK
- title: varchar, not NULL
- publisher: varchar, not NULL
- datePublished: date, not NULL
- copyAmount: int, not NULL,
- amountAvailable: int, not NULL
- timesRented: int, not NULL
- Relationship: a M:M relationship between books and authors is implemented by combining authorID and bookID in the Authored Works intersection table (Above).
    - A book can have multiple authors, and an author can have written multiple books.
- Relationship: A 1:M relationship (**can be null**) between genres and books is implemented with genreID as a FK inside of books.
    - A book may not fit into any existing genre, so the field is left null. There may also be genres listed that no books fall under. If a genre is applicable, there can only be one per book, however many books can fall under the category of a single genre.

Genres: a list of the book genres a book can fall under (**restriction: a book can only have one genre)

- genreID: int, auto_increment, unique, not NULL, PK
- name: varchar, not NULL,
- isFiction: boolean
- Relationship: a 1:M relationship (**can be null**) between genres and books is implemented with genreID as a FK inside of books.
    - A book may not fit into any existing genre, so the field is left null. There may also be genres listed that no books fall under. If a genre is applicable, there can only be one per book, however many books can fall under the category of a single genre.

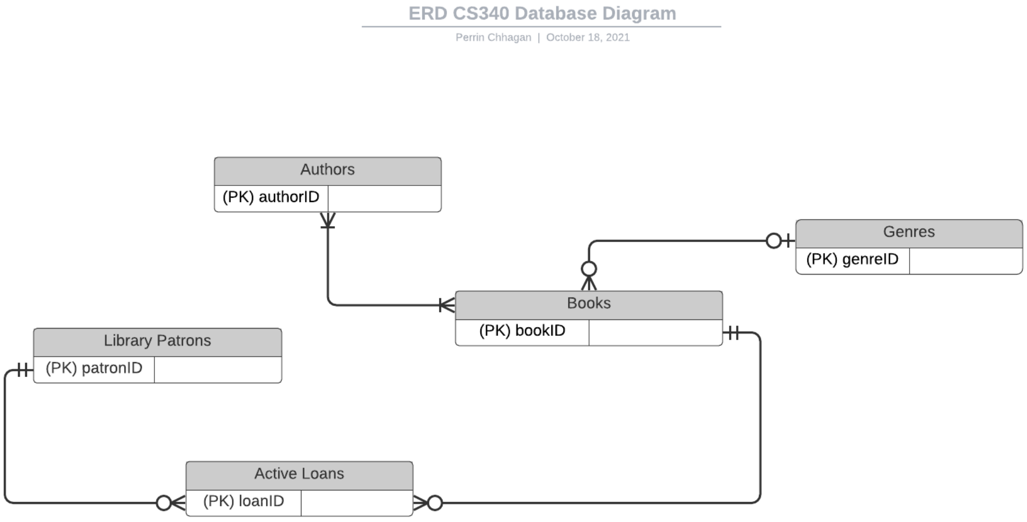
Active_Loans: records the details of each instance of a book being currently loaned out (e.g. one book per active loan)

- loanID: int, auto_increment, unique, not NULL, PK
- bookID: int, not NULL, FK
- patronID: int, not NULL, FK
- returnDate: date, not NULL
- isOverdue: boolean
- Relationship:A 1:M relationship (0 or more) between library patrons and active loans is implemented with patronID as a FK inside active loans.
    - A library patron can be currently borrowing many books, or none.
- Relationship:Also, a 1:M relationship (0 or more) between books and active loans is implemented with bookID as a FK inside active loans.
    - Multiple copies of a book may be loaned out, or none may be currently loaned out.

Library_Patrons: records the details of each person who is a member of the library

- patronID: int, auto_increment, unique, not NULL, PK
- firstName: varchar, not NULL
- lastName: varchar, not NULL
- Relationship: A 1:M relationship (0 or more) between library patrons and active loans is implemented with patronID as a FK inside active loans.
    - A library patron can be currently borrowing many books, or none.

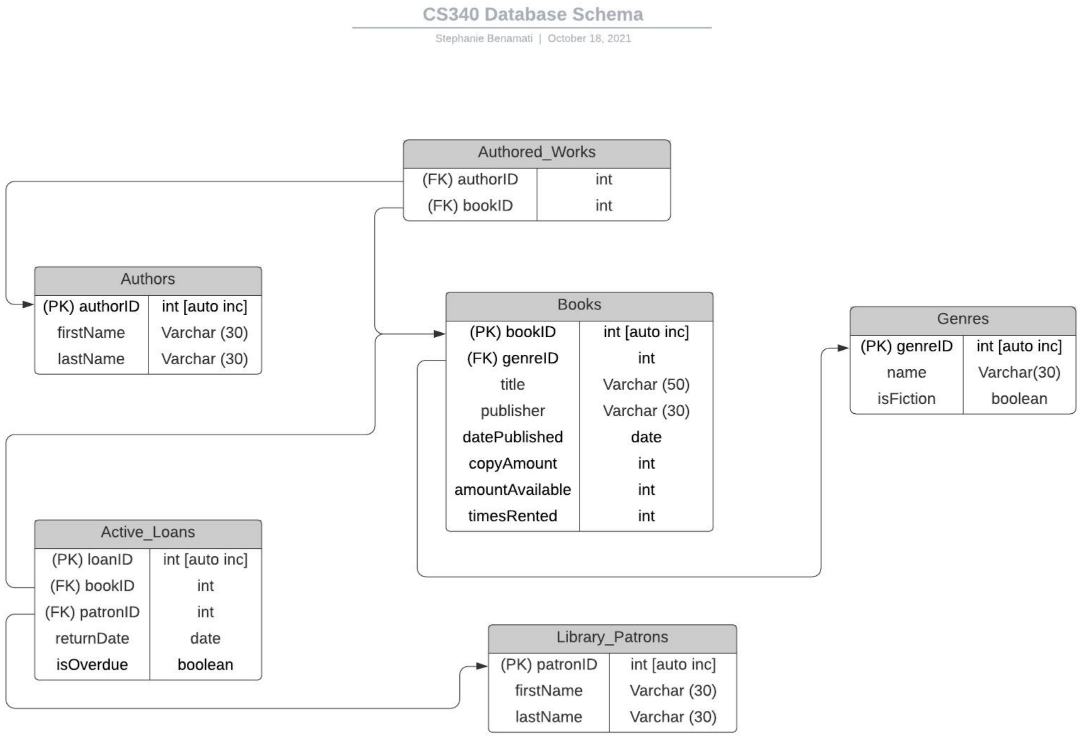
Entity-Relationship Diagram (ERD):



ERD CS340 Database Diagram

Perrin Chhagan | October 18, 2021

Schema:



CS340 Database Schema

Stephanie Benamati | October 18, 2021

Authored_Works

(FK) authorID	int
(FK) bookID	int

Authors

(PK) authorID	int [auto inc]
firstName	Varchar (30)
lastName	Varchar (30)

Books

(PK) bookID	int [auto inc]
(FK) genreID	int
title	Varchar (50)
publisher	Varchar (30)
datePublished	date
copyAmount	int
amountAvailable	int
timesRented	int

Genres

(PK) genreID	int [auto inc]
name	Varchar(30)
isFiction	boolean

Active_Loans

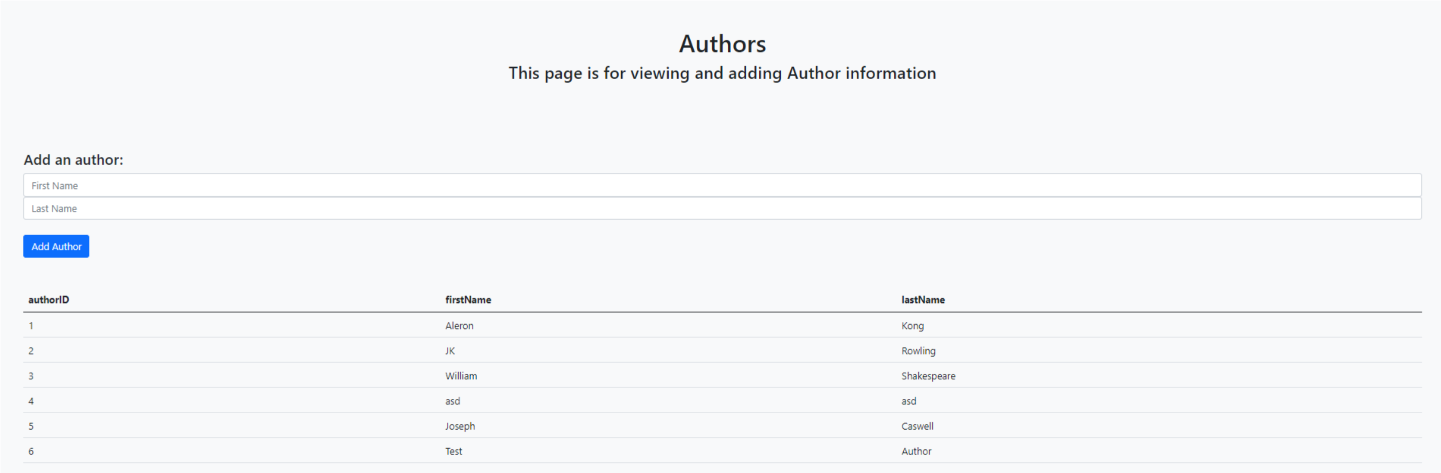
(PK) loanID	int [auto inc]
(FK) bookID	int
(FK) patronID	int
returnDate	date
isOverdue	boolean

Library_Patrons

(PK) patronID	int [auto inc]
firstName	Varchar (30)
lastName	Varchar (30)

Screen Captures:

View & Add New Authors Page

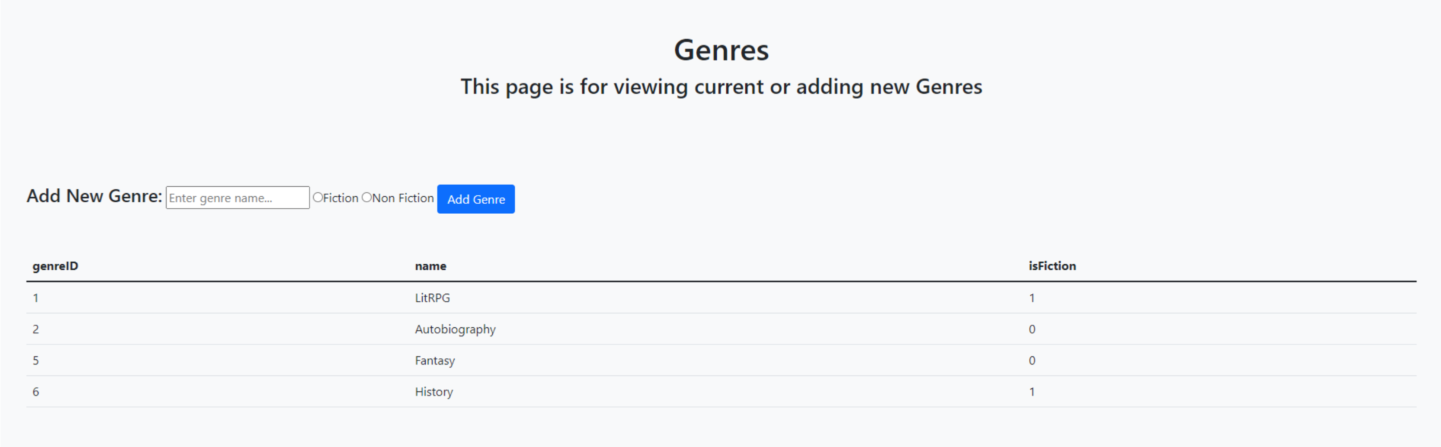


Authors

This page is for viewing and adding Author information

authorID	firstName	lastName
1	Aleron	Kong
2	JK	Rowling
3	William	Shakespeare
4	asd	asd
5	Joseph	Caswell
6	Test	Author

View & Add New Genres Page

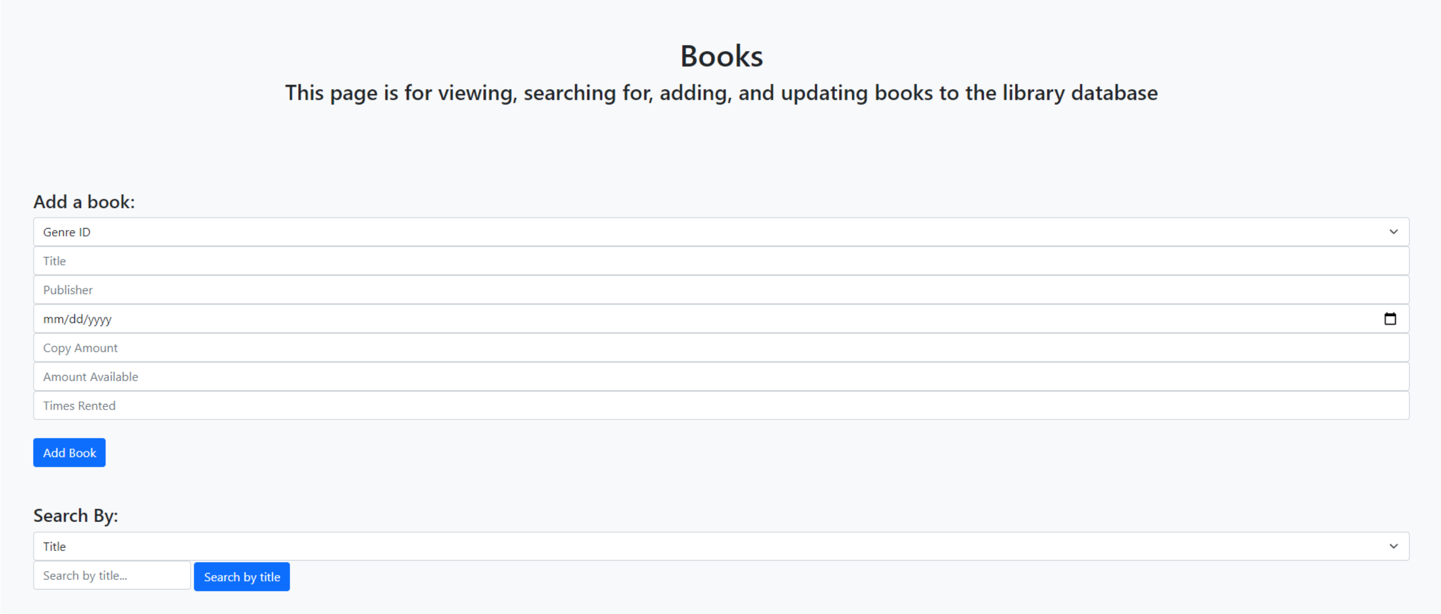


Genres

This page is for viewing current or adding new Genres

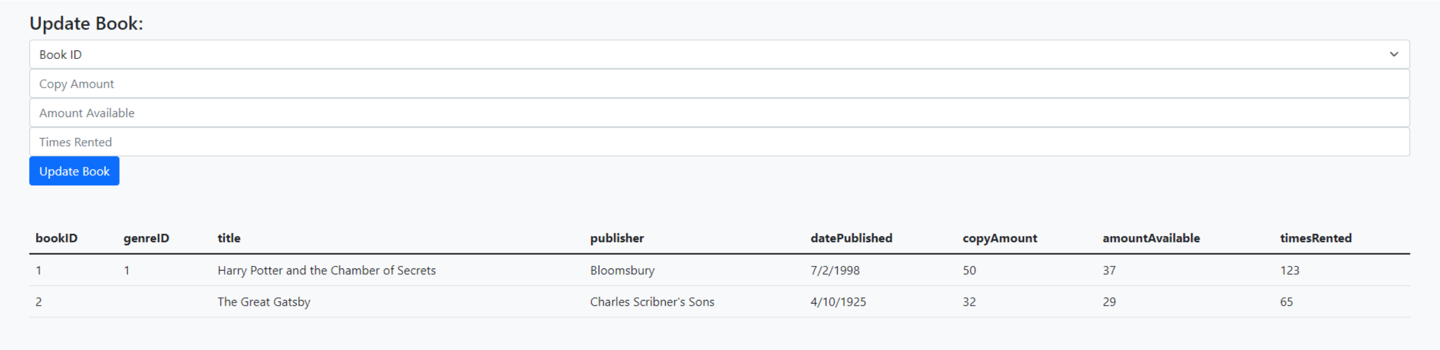
genreID	name	isFiction
1	LitRPG	1
2	Autobiography	0
5	Fantasy	0
6	History	1

View, Search For, Add New, and Update Books Page



Books

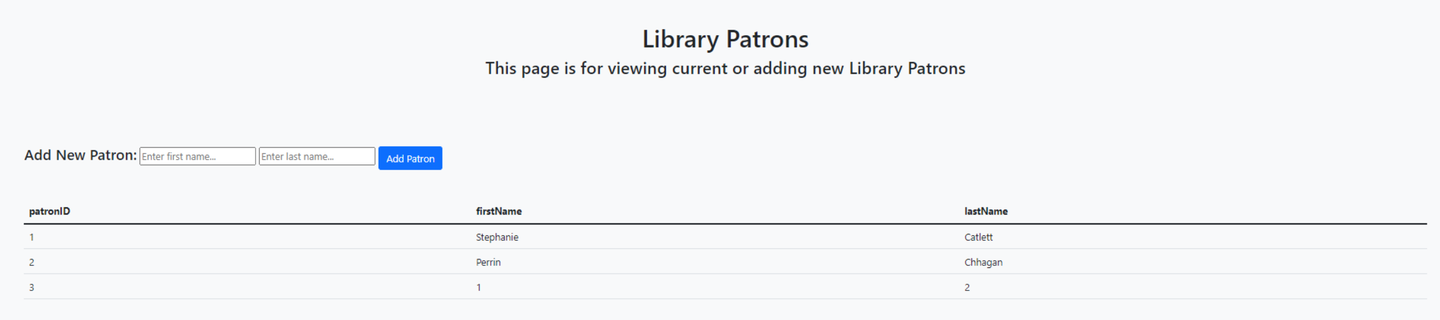
This page is for viewing, searching for, adding, and updating books to the library database



Update Book:

bookID	genreID	title	publisher	datePublished	copyAmount	amountAvailable	timesRented
1	1	Harry Potter and the Chamber of Secrets	Bloomsbury	7/2/1998	50	37	123
2		The Great Gatsby	Charles Scribner's Sons	4/10/1925	32	29	65

View & Add New Library Patrons Page



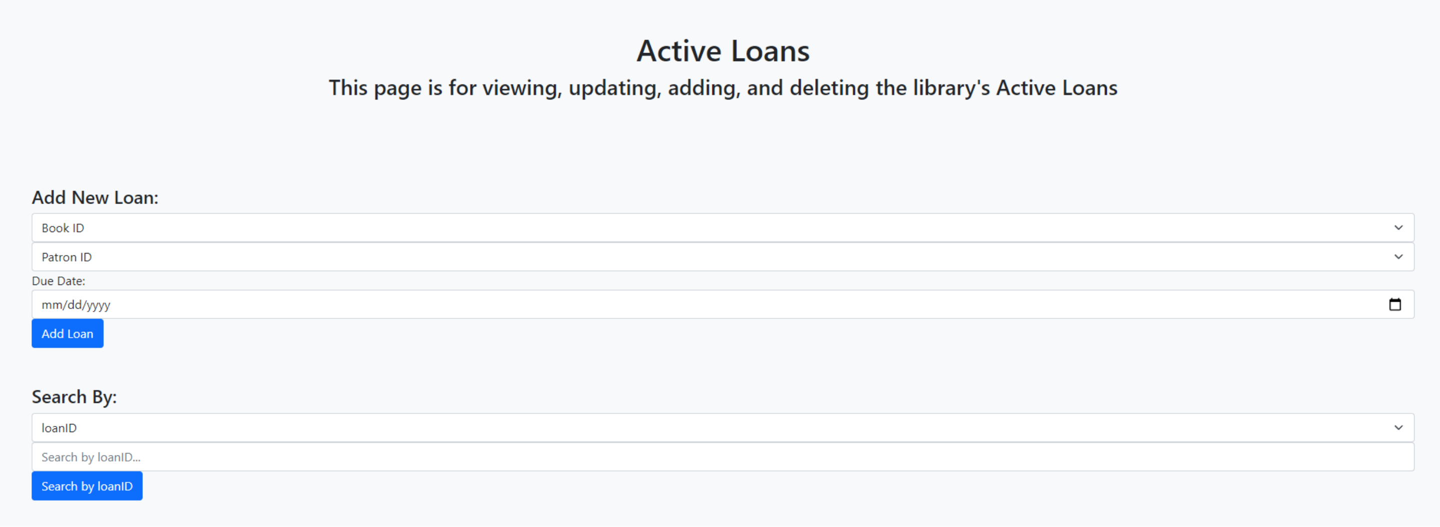
Library Patrons

This page is for viewing current or adding new Library Patrons

Add New Patron:

patronID	firstName	lastName
1	Stephanie	Catlett
2	Perrin	Chhagan
3	1	2

View, Search For, Add New, Update, & Delete Active Loans Page



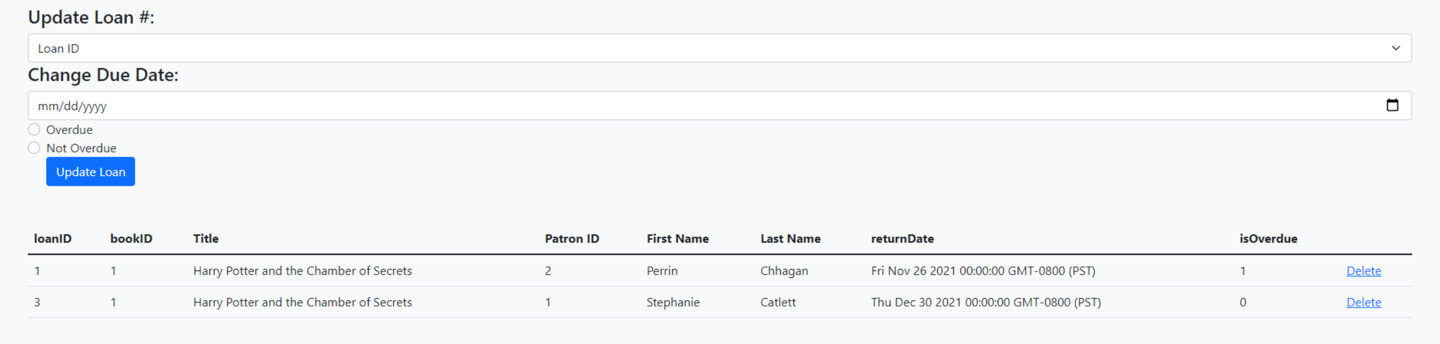
Active Loans

This page is for viewing, updating, adding, and deleting the library's Active Loans

Add New Loan:

Due Date:

Search By:



Update Loan #:

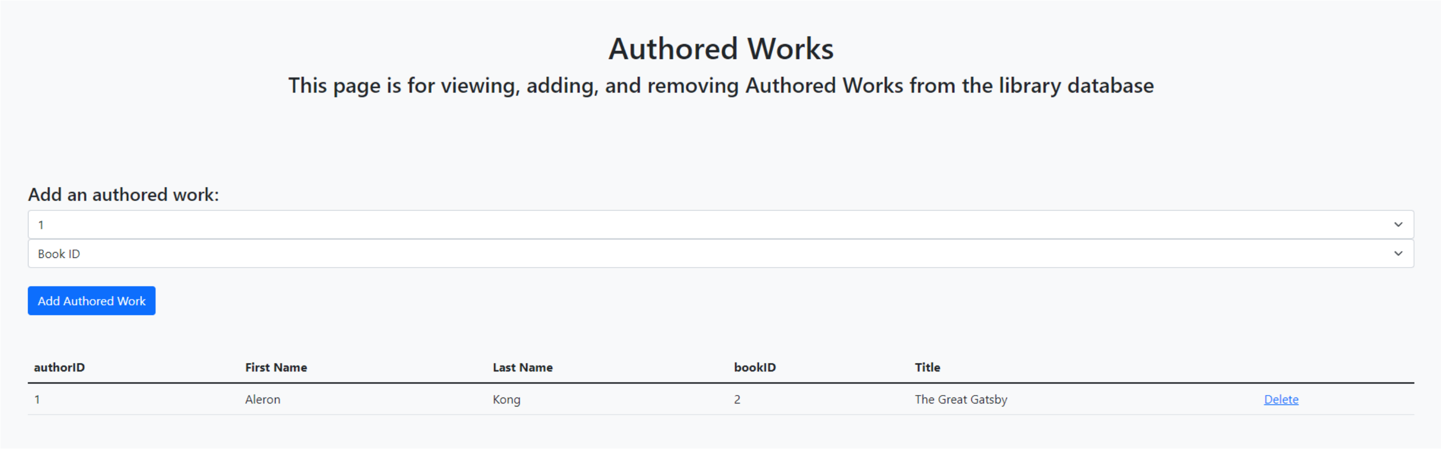
Change Due Date:

○ Overdue

○ Not Overdue

loanID	bookID	Title	Patron ID	First Name	Last Name	returnDate	isOverdue	
1	1	Harry Potter and the Chamber of Secrets	2	Perrin	Chhagan	Fri Nov 26 2021 00:00:00 GMT-0800 (PST)	1	Delete
3	1	Harry Potter and the Chamber of Secrets	1	Stephanie	Catlett	Thu Dec 30 2021 00:00:00 GMT-0800 (PST)	0	Delete

View, Add New, & Delete Authored Works Page



Authored Works

This page is for viewing, adding, and removing Authored Works from the library database

Add an authored work:

1

Book ID

authorID	First Name	Last Name	bookID	Title	
1	Aleron	Kong	2	The Great Gatsby	Delete