Ruby workshop task

You are tasked with building a basic library system where users can borrow and return books. The system should track books, users, and the borrowing process.

The goal of this project is to learn the following:

- Learning about the basic data types used in Ruby
- Reading user input and writing an output to the terminal screen
- Writing data into a text file
- Basics of error handling
- Reading and parsing data from a text file
- How objects can be used to store state
- How objects interact with each other
- Blocks and the capabilities the Enumerable module provides
- Iterative workflow using irb - interactive Ruby console

Requirements

- Use git for version control.
- The program should accept user input from the console and write output to the console. (Hint: take a look at gets and puts methods)
- Upon running the program, the program should ask for a username, and if a user doesn't exist, invite the user to create an account by entering a username.
- Existing books are provided in the file books.csv. To work with CSV library you will need to import it. To do that write require 'csv' at the top of your program.
- Usernames should be stored inside a plain text file called users.db.
- Borrowed books should be stored inside a plain text file called borrowed_books.db, where each entry is book_id username.
- Once the user is logged in, the program should provide the following options:
    - *List available books*
    - *Borrow a book*
    - *Return a book*
    - *Exit*
- The *list available books* option should return a list of books that are available for borrowing. Each book should be printed in the following format - <book_id> <book_name> <book_author> <release_year> and separated by a new line.
- The *borrow a book* option should ask the user for a book ID. If the book is available, the program should write the borrowing fact to the borrowed_books.db file and print a success message. If the book is not available, the program should print a failure message.
- The *return a book* option should ask the user for a book ID. If the book being returned was borrowed, the program should delete the borrowing fact from the

borrowed_books.db file and print a thank you message. If the book was not borrowed, the program should print a message specifying that the book was not borrowed.

- The *exit* option should terminate the program and write a goodbye message.
- You must create at least one custom Exception class.
- You must create at least two classes and have the program communicate with those classes while doing the computation.

Extension ideas:

- Implement a user authentication system where users must log in with a password to perform actions.
- Use a SQLite database instead of storing usernames and passwords inside a plain text file.
- Implement a late fee mechanism when the user returns a book after the borrowing period.
- Write a test suite for your program using the RSpec testing framework.
- [Advanced] allow multiple users to borrow books simultaneously by converting the program into two separate programs - client program and server program. Take a look at the sockets API and this article - https://workingwithruby.com/wwtcps/intro/