**Requirements**

* **Registration**: Users should be able to register for the website, providing (at minimum) a username and password.
* **Login**: Users, once registered, should be able to log in to the website with their username and password.
* **Logout**: Logged in users should be able to log out of the site.
* **Import**: Provided in this project is a file called books.csv, which is a spreadsheet in CSV format of 5000 different books. Each one has an ISBN number, a title, an author, and a publication year. In a Python file called import.py separate from the web application, write a program that will take the books and import them into the PostgreSQL database. I will first need to decide what table(s) to create, what columns those tables should have, and how they should relate to one another.
* **Search**: Once a user has logged in, they should be taken to a page where they can search for a book. Users should be able to type in the ISBN number of a book, the title of a book, or the author of a book. After performing the search, the website should display a list of possible matching results, or some sort of message if there were no matches. If the user typed in only part of a title, ISBN, or author name, my search page finds matches for those as well!
* **Book Page**: When users click on a book from the results of the search page, they should be taken to a book page, with details about the book: its title, author, publication year, ISBN number, and any reviews that users have left for the book on the website.
* **Review Submission**: On the book page, users should be able to submit a review: consisting of a rating on a scale of 1 to 5, as well as a text component to the review where the user can write their opinion about a book. Users should not be able to submit multiple reviews for the same book.
* **Goodreads Review Data**: On my book page, I also display (if available) the average rating and number of ratings the work has received from Goodreads.
* **API Access**: If users make a GET request to my website’s /api/<isbn> route, where <isbn> is an ISBN number, my website should return a JSON response containing the book’s title, author, publication date, ISBN number, review count, and average score. The resulting JSON should follow the format:
* **Error**: If the requested ISBN number is not in my database, my website returns a 404 error.