LEAD BACKEND ENGINEER ASSESSMENT

You are required to build the backend of a web app that handles book loans for a small local library. As part of the exercise, you will need to setup and populate a database and create an API that meets the criteria outlined below. The API needs to be built using Python with SQLite used for the database. You are recommended to spend around 5 hours on the assessment. Submissions that do not implement all the functionality will still be accepted so be sure to adhere to best practices during development.

You have been provided with a CSV file containing the initial book inventory data and should design the table structure as you see appropriate.

The API requires endpoints that allow the library staff to perform the following operations:

For Library Website Users:

a. An endpoint that allows the database searching by title and by author, returning books and their availability.
b. Add/remove unavailable books to/from a wishlist such that they are notified when they become available

For Library Staff:

c. Change the rental status (available/borrowed) for a book (which should also trigger the email notifications to users with the book in their wishlist)
d. Generate a report on the number of books being rented and how many days they've been rented for.
e. The frontend of the library website displays affiliate links to copies of the book available on Amazon for each book. The Amazon book IDs can be retrieved from the OpenLibrary API (no developer key required). An endpoint is required that will update the Amazon IDs stored in the database for all the books.

The function in endpoint (c) that requires emails to be sent out should be implemented by printing the email text to the output or logging to a file.

To submit your assessment, return a zip file via email with all the files as necessary. Please include your source code, database and any other relevant files (e.g. shell scripts, config files, etc.). Also include brief documentation that lists the endpoint URL, HTTP method and data to be submitted to the endpoint for each of the tasks above.