

SOFTWARE DEVELOPER 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 which meets the data and functionalities requirements. A CSV file containing the initial book inventory data and should design the table structure as you see appropriate.
- create the APIs that meets the criteria outlined below.

The APIs require endpoints that allow the library staff to perform the following operations:

MINIMUM

1. *For Library Website Users:*

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

2. *For Library Staff:*

- Change the rental status (available/borrowed) for a book which should also trigger the email notifications to users with the book in their wishlist. No real email is required to be sent, the implementation could be printing the email text or logging to file.
- Generate a report on the number of books being rented and how many days they've been rented for.
- The frontend of the library website displays affiliate links to copies of the book available on Amazon for each book. An endpoint is required that will update the Amazon IDs stored in the database for all the books.

3. *Unit tests*

Implement unit tests. There are no specific requirements around test coverage.

OPTIONAL

On the top above features, you are encouraged to explore solutions for the following extra features. You can choose to implement them or just have a think.

4. *Front end*

Build a simple frontend for example, a simple page of showing the user's wishlist.

5. *Performance optimisation*

Performance considerations around sending emails to large number of book subscribers.

6. *Integration*

Using the OpenLibrary API (<https://openlibrary.org/dev/docs/api/search> , No API key is required) to retrieve the Amazon IDs(asin) programmatically

Note - The code assessment requires to use Python and SQLite for the database.

How to submit

You are recommended to spend around 5-7 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.

To submit your assessment, you can either share a repo url (e.g. github) or 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.) as we will run your solution to check the result. 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.