



## YSD B03 J2EE\_Mini Project Description Online Book Library Application

---

Your goal is to design and develop an **Online Book Library** application. This application will be a back-end only application. So no front-end is required. Just creating the REST end-points is enough.

All the REST end-points of this application need to be protected using **Spring Security role based authentication**. Also use MySQL for the DB.

The application will have the below API / REST end-points:

### 1. User Management Endpoints:

- **/user/register**: This end-point will register an user with user information: firstName, lastName, email, password, address.
- **/user/login**: This end-point will login with user email and password and will return a token as response. User email should be present at payload data.
- **/users/{userId}**: Retrieve user details by userId. Only users with ADMIN roles should have access to this API.
- **/users/{userId}/books**: Retrieve the books borrowed or owned by a specific user. This endpoint can be accessible by the user to view their own books or by an ADMIN to view any user's books.
- **/users/{userId}/borrowed-books**: Retrieve the books currently borrowed by a specific user. This endpoint can be accessible by the user to view their own borrowed books or by an ADMIN to view any user's borrowed books.

### 2. Books Management:

- **/books/create**: This end-point will be used to add new book to the database. User must have "ADMIN" role to use this API.
- **/books/update**: This end-point will be used to update a book's data that is already saved in the database. User must have "ADMIN" role to use this API.
- **/books/delete**: This end-point will be used to delete a book from the database. User must have "ADMIN" role to use this API.
- **/books/all**: This end-point will fetch and show all the books data stored in the database. User must have either "ADMIN" or "CUSTOMER" role to use this API.



### 3. Book Borrowing/Returning Endpoints:

- `/books/{bookId}/borrow`: Allow users (CUSTOMER) to borrow a book by bookId. Implement logic to track the book's availability and due date.
- `/books/{bookId}/return`: Allow users (CUSTOMER) to return a borrowed book by bookId. Update the book's status accordingly.

### 4. Book Reservation Endpoints (Optional):

- `/books/{bookId}/reserve`: Allow users (CUSTOMER) to reserve a book that is currently unavailable. Implement logic to notify the user when the book becomes available.
- `/books/{bookId}/cancel-reservation`: Allow users (CUSTOMER) to cancel a book reservation. Update the reservation status accordingly.

### 5. Book Reviews and Ratings:

- `/books/{bookId}/reviews`: Retrieve reviews and ratings for a specific book by bookId.
- `/books/{bookId}/reviews/create`: Allow users (CUSTOMER) to create a review and rating for a book.
- `/books/{bookId}/reviews/{reviewId}/update`: Allow users (CUSTOMER) to update their own review and rating for a book.
- `/books/{bookId}/reviews/{reviewId}/delete`: Allow users (CUSTOMER) to delete their own review and rating for a book.

### 6. User History (Optional):

- `/users/{userId}/history`: Allow users to view their borrowing history, including borrowed books, due dates, and return dates.

**N.B:** Please read the requirement carefully and create the necessary Java classes for this application according to your understanding.