**Running the Application:**

1. **Prerequisites**:
   * Ensure you have Java Development Kit (JDK) installed on your system.
   * Make sure you have Apache Maven.
2. **Clone the Repository**:
   * Clone the project repository from your version control system (e.g., GitHub).
3. **Navigate to Project Directory**:
   * Open a terminal or command prompt.
   * Change directory to the root directory of the project.
4. **Build the Application**:
   * run the following command:

mvn clean package

1. **Run the Application**:
   * Once the build is successful, you can run the application using Maven.

mvn spring-boot:run

or you can just run it in any IDE (Eclipse , Intellij)

1. **Import Database**:

* Importing database in mysql workbench (schema: maids, username:root, password:123456)

1. **Accessing the Application**:
   * Once the application starts, it will be accessible at **http://localhost:8080** by default.

**Interacting with API Endpoints:**

Using Postman:

1. Open Postman.
2. Import the provided Postman collection or manually create requests for each API endpoint.
3. Set the appropriate request method (GET, POST, PUT, DELETE).
4. Enter the endpoint URL (e.g., http://localhost:8080/api/books).
5. Add authentication username=admin, password=admin (basic auth).
6. Click "Send" to execute the request.
7. View the response in the Postman interface.

**Below are all endpoints:**

* Books management endpoints:

● GET /api/books: Retrieve a list of all books.

● GET /api/books/{id}: Retrieve details of a specific book by ID.

● POST /api/books: Add a new book to the library.

● PUT /api/books/{id}: Update an existing book's information.

● DELETE /api/books/{id}: Remove a book from the library.

* Patron management endpoints:

● GET /api/patrons: Retrieve a list of all patrons.

● GET /api/patrons/{id}: Retrieve details of a specific patron by ID.

● POST /api/patrons: Add a new patron to the system.

● PUT /api/patrons/{id}: Update an existing patron's information.

● DELETE /api/patrons/{id}: Remove a patron from the system.

* Borrowing endpoints:

● POST /api/borrow/{bookId}/patron/{patronId}: Allow a patron to

borrow a book.

● PUT /api/return/{bookId}/patron/{patronId}: Record the return of a borrowed book by a patron.