Library Management System

Table of Contents

Introduction

The Library Management System is designed to manage user registrations, book loans, and returns with secure authentication.

Setup

Prerequisites

- Java 11 or higher
- Maven
- MySQL database

Installation

1. Clone the repository:

   ```sh
   git clone https://github.com/yourusername/library-management-system.git
   cd library-management-system
   ```

2. Set up the MySQL database:

   ```sh
   mysql -u root -p
   CREATE DATABASE librarysystem;
   ```

3. Update `application.properties` with your database credentials.

Running the Application

1. Build the project:

   ```sh
   ./mvnw clean install
   ```

2. Run the application:

   ```sh
   ./mvnw spring-boot:run
   ```

The application will start at `http://localhost:8080`.

API Usage

User Management

- **Register**: `POST /user/register`
  - Request:
   ```json
   {
      “Firstname”:”subhankar”,
     “lastname”:”shaw”
     "email": "user@example.com",
     "password": "password"
   }
   ```

- **Login**: `POST /users/authenticate
  - Request:
   ```json
   {
     "email": "user@example.com",
     "password": "password"
   }
   ```

Book Management

- **Get all books**: `GET /api/books`
- **Get book by ID**: `GET /books/{id}`
- **Add a new book**: `POST /books`

  - Request:
    ```json
    {
      "title": "Book Title",
      "author": "Author Name",
      "category": "Category"
    }
    ```
- **Update a book**: `PUT /book/{id}`
  - Request:
    ```json
    {
      "title": "Updated Title",
      "author": "Updated Author",
      "category": "Updated Category"
    }
    ```
- **Delete a book**: `DELETE /book/{id}`

Loan Management

- **Loan a book**: `POST /loans`
  - Request:
    ```json
    {
      "userId": 1,
      "bookId": 1
    }
    ```
- **Return a book**: `POST /loan/return/{loanId}`
- **Get loans by user ID**: `GET /loan/{userId}`

User Authentication

The application uses JWT for secure authentication. Include the JWT token in the `Authorization` header for requests that require authentication.

Obtaining a Token

After logging in, you will receive a JWT token. Include this token in the `Authorization` header as follows: