



# Library Management System Design and Analytics

# Project Description

This RESTful API is designed to support a Library Management System (LMS). Its primary purpose is to digitally manage the library's data on books, authors, and members. The API will handle collecting, updating, and deleting book information, as well as managing borrowing and returning transactions for members. It will provide an efficient and automated system for library staff.

## Functionalities

1. Complete CRUD (Create, Read, Update, Delete) management for books, authors, and members.
  2. Tracking of book borrowing and returning by members.
  3. Automatic updating of a book's availability status.

## Database Model

Here's a detailed plan for each model, including the field names, data types, and relationships. This covers all the entities and their interactions for the Library Management System.

## 1. Member Model

Field Name	Data Type	Description
id	Primary Key	Unique Identifier for Member
first_name	CharField	Required FirstName for member
last_name	CharField	Required LastName for member
email	EmailField	Unique email for the member
address	TextField	Optional address of the member
phone	CharField	Optional address of the member
password	CharField (hashed)	Hashed password for authentication

## 2. Author Model

Field Name	Data Type	Description
id	Primary Key	Unique Identifier for Author
name	CharField	Name of the author
bio	CharField	Author sort biography

## 3. Category Model

Field Name	Data Type	Description
id	Primary Key	Unique Identifier for category
name	CharField	Name of the category
description	TextField	description of category

## 4. Book Model

Field Name	Data Type	Description	Relationship
id	Primary Key	Unique Identifier for Book	

title	CharField	Name of the book	
author	Foreign Key	Link to the author model	Many To One
isbn	CharField	International Book Number(blank & null is True)	
category	Foreign Key	Link to the category model	Many to One
stock	PositiveIntegerField	How many copies of this book are available in the library	

## 5. BorrowRecord

Field Name	Data Type	Description	Relationship
id	Primary Key	Unique identifier for borrowrecord	
book	Foreign Key	Link to the book model	Many To One
member	Foreign Key	Link to the member model	Many To One
borrow_date	DateField	date of the borrow	
return_date	DateField	date of the return	
due_date	DateField	date of the due	

## Api Endpoints

### 1. Member

HTTP method	Endpoint	Description
GET	/api/members/	Retrieve a list of all members
POST	/api/members/	Create a new member
GET	/api/members/{id}	Retrieve a single member's details

## 2. Categories

HTTP method	Endpoint	Description
GET	/api/categories/	Retrieve a list of all book categories
POST	/api/categories/	Create a new book category
PUT	/api/categories/{id}	Update details of a specific category
DELETE	/api/categories/{id}	Delete a category

## 3. Author

HTTP method	Endpoint	Description
GET	/api/authors/	Retrieve a list of all author
POST	/api/authors/	List a new Author
PUT	/api/authors/{id}/	Update Author
DELETE	/api/authors/{id}/	Delete a Author

## 4. Book

HTTP method	Endpoint	Description
GET	/api/books/	Retrieve a list of all books
GET	/api/books/{id}/	Retrieve details of a single book
POST	/api/books/	Create (list) a new book
PUT	/api/books/{id}/	<b>Full update</b> of a single book's details
DELETE	/api/books/{id}/	Delete a book

## 5. BorrowRecord

HTTP method	Endpoint	Description
GET	/api/borrow-records/	Retrieve a list of all loan records (active and past).

POST	/api/borrow-records/	Create a new borrow record (requires <code>book</code> and <code>member</code> )
PATCH	/api/borrow-records/{id}	Mark a specific borrow record as returned (updates <code>return_date</code> , and <code>stock</code> )
GET	/api/borrow-records/? active=true	Retrieve a list of only currently active (not yet returned) loans