

# Library Management System Plan

## Step 1: API Design

### Project Description

A API for a library Management System. The API runs library core functions, including cataloging books, managing member information, an tracking the borrowing and returning of books.

### Database Schema (Models Definition)

#### Book Model

- **title:** CharField - title of the book.
- **author:** ForeignKey to Author - links author model
- **isbn:** CharField - unique ISBN of the book
- **category:** CharField - defines category of the book
- **availability\_status:** BooleanField - if books available (default- true)

#### Author Model

- **name:** CharField - author name
- **biography:** TextField - author biography

#### Member Model

- **name:** CharField - library member name
- **email:** EmailField - library member email
- **membership\_date:** DateField - membership date

#### BorrowRecord Model

- **book:** ForeignKey to Book - links the borrow book
- **member:** ForeignKey to Member - link borrow book to member.
- **borrow\_date:** DateField - When the book taken
- **return\_date:** DateField - Book return date

### API Endpoints

#### Books Endpoints

- GET- api /books/ - gets the list of books
- POST- api /books/ - add new book
- GET- api /books/{id}/ - details of specific books
- PUT- api /books/{id}/ - update details of a book
- DELETE- api /books/{id}/ - remove book from library

## Members Endpoints

- GET- api /members/ - list of members.
- POST- api /members/ - add new member
- GET- api /members/{id}/ - retrieve details of specific member
- PUT- api /members/{id}/ - update member details
- DELETE- api /members/{id}/ - remove a member

## Borrowing Endpoints

- POST- api/borrow/ - record book borrow details
- POST- api /return/ - record book returns

# Step 2: API Implementation

## Implement the Models

1. Book, Author, Member, and BorrowRecord models in api/models.py.

## Implement Serializers

1. Create api/serializers.py.

## Implement API Views

1. Use ModelViewSet from DRF to handle CRUD operations for Book, Author, and Member models.
2. Use APIView or function-based views for the custom borrow and return logic.

## Test Your API

## API Endpoints

1. Books: <http://127.0.0.1:8000/api/books/>
  2. Authors: <http://127.0.0.1:8000/api/authors/>
  3. Members: <http://127.0.0.1:8000/api/members/>
1. Access the browsable API at <http://127.0.0.1:8000/api/>
  2. Use the browsable API forms or tools like Postman to test each endpoint.