**Library Management Project**

**Advance Application Development**

**Sidharth Goenka**

**2006543**

**December 05, 2023**

**Overview**

This project is a Library Management System with a RESTful API for CRUD operations.

**Dependencies**

.NET SDK (at least version 6.4.0)

SQLite Database (or other supported database)

**Prerequisites**

**Install .NET SDK**:

Install the .NET SDK version 6.4.0 or higher if you haven't already.

**Database Configuration**:

The Database connection string should be updated in the **appsettings.json** file:

"ConnectionStrings": {

"DefaultConnection": "Data Source=library.db"

}

**How to Run**

Open a terminal or command prompt and navigate to the project directory.

Run the following command to restore NuGet packages:

dotnet restore

**Database Migration**:

Run the following command to apply database migrations:

dotnet ef database update

**Run the Application:**

Run the following command to start the application:

dotnet run --launch-profile https

The application will start, and you can access it using :

<https://localhost:7101/swagger>

**API Endpoints**

Swagger Documentation:

Access the Swagger documentation at <https://localhost:7101/swagger>

A screenshot of a computer

Description automatically generated

**API Controllers**

**Schemas**

Figure: Swagger Index page

Testing CRUD Operations

**Create**: POST API

Use a tool like Postman or curl to send a POST request to create a new record.

**Read**: GET API

Retrieve data using GET requests. For example, https://localhost:5001/api/books to get a list of books.

**Update**: PUT API

Use a tool to send a PUT request to update an existing record.

**Delete**: DELETE API

Send a DELETE request to remove a record.

**Author APIs:**

A group of colorful lines

Description automatically generated with medium confidence

Figure: CRUD APIs for Author

1. GET: <https://localhost:7101/api/Author> -> GET API fetches list of all the saved Authors in DB.
   1. Request Body: N/A
   2. Response Body: [

{

"authorId": 0,

"name": "String"

}

]

1. POST: https://localhost:7101/api/Author -> POST API insert data of Author in DB.
   1. Request Body:

{

"authorId": 0,

"name": "String"

}

* 1. Response Body:

{

"authorId": 0,

"name": "String"

}

1. GET: https://localhost:7101/api/Author/{id} -> GET API fetches data of Author with a specific ID in DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body:

{

"authorId": 0,

"name": "String"

}

1. PUT: https://localhost:7101/api/Author/{id} -> PUT API updates data of Author in DB.
   1. Request Parameter: 0
   2. Request Body:

{

"authorId": 0,

"name": "String"

}

* 1. Response Body: N/A, but response status code is 200

1. DELETE: https://localhost:7101/api/Author/{id} -> DELETE API deletes the data of Author with the mentioned id from DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body: N/A, but response status code is 200

**Book APIs:**

A group of colorful lines

Description automatically generated

Figure: CRUD APIs for Book

1. GET: <https://localhost:7101/api/Book> -> GET API fetches list of all the saved Books in DB.
   1. Request Body: N/A
   2. Response Body: [

{

"bookId": 1,

"title": "Book 1",

"authorId": 1,

"libraryBranchId": 1

}

]

1. POST: https://localhost:7101/api/Book -> POST API insert data of Book in DB.
   1. Request Body:

{

"title": "string",

"authorId": 0,

"libraryBranchId": 0

}

* 1. Response Body:

{

"bookId": 0,

"title": "string",

"authorId": 0,

"libraryBranchId": 0

}

1. GET: https://localhost:7101/api/Book/{id} -> GET API fetches data of Book with a specific ID in DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body:

{

"bookId": 0,

"title": "string",

"authorId": 0,

"libraryBranchId": 0

}

1. PUT: https://localhost:7101/api/Book/{id} -> PUT API updates data of Book in DB.
   1. Request Parameter: 0
   2. Request Body:

{

"bookId": 0,

"title": "string",

"authorId": 0,

"libraryBranchId": 0

}

* 1. Response Body: N/A, but response status code is 200

1. DELETE: https://localhost:7101/api/Book/{id} -> DELETE API deletes the data of Book with the mentioned id from DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body: N/A, but response status code is 200

**Customer APIs:**

A group of colorful lines

Description automatically generated with medium confidence

Figure: CRUD APIs for Customer

1. GET: <https://localhost:7101/api/Customer> -> GET API fetches list of all the saved Customers in DB.
   1. Request Body: N/A
   2. Response Body: [

{

"customerId": 1,

"name": "Customer 1"

}

]

1. POST: https://localhost:7101/api/Customer -> POST API insert data of Customer in DB.
   1. Request Body:

{

"name": "string"

}

* 1. Response Body:

{

"customerId": 0,

"name": "string"

}

1. GET: https://localhost:7101/api/Customer/{id} -> GET API fetches data of Customer with a specific ID in DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body:

{

"customerId": 0,

"name": "string"

}

1. PUT: https://localhost:7101/api/Customer/{id} -> PUT API updates data of Customer in DB.
   1. Request Parameter: 0
   2. Request Body:

{

"customerId": 0,

"name": "string"

}

* 1. Response Body: N/A, but response status code is 200

1. DELETE: https://localhost:7101/api/Customer/{id} -> DELETE API deletes the data of Customer with the mentioned id from DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body: N/A, but response status code is 200

**Library Branch APIs:**

A group of rectangular colored lines

Description automatically generated with medium confidence

Figure: CRUD APIs for Library Branch

1. GET: https://localhost:7101/api/LibraryBranch -> GET API fetches list of all the saved Library Branch in DB.
   1. Request Body: N/A
   2. Response Body: [

{

"libraryBranchId": 0,

"branchName": "string"

}

]

1. POST: https://localhost:7101/api/ LibraryBranch -> POST API insert data of Library Branch in DB.
   1. Request Body:

{

" branchName ": "string"

}

* 1. Response Body:

{

"libraryBranchId": 0,

"branchName": "string"

}

1. GET: https://localhost:7101/api/ LibraryBranch /{id} -> GET API fetches data of Library Branch with a specific ID in DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body:

{

"libraryBranchId": 0,

"branchName": "string"

}

1. PUT: https://localhost:7101/api/ LibraryBranch /{id} -> PUT API updates data of Library Branch in DB.
   1. Request Parameter: 0
   2. Request Body:

{

" libraryBranchId ": 0,

" branchName ": "string"

}

* 1. Response Body: N/A, but response status code is 200

1. DELETE: https://localhost:7101/api/ LibraryBranch /{id} -> DELETE API deletes the data of Library Branch with the mentioned id from DB.
   1. Request Parameter: 0
   2. Request Body: N/A
   3. Response Body: N/A, but response status code is 200

**Database Tables:**

**A screenshot of a computer

Description automatically generated**

Figure: library.db