**Asp.netwebservice1**

**Case study Process:**

To implement a library management system using ASP.NET, you need to create a web service that supports (Create, Read, Update, Delete) operations for a table with the following columns: `Bookno`, `Bookname`, `Authorname`, and `Price`. Here’s a steps:

* Step 1: Create the Database Table

Refer create database file

* Step 2: Set Up the ASP.NET Web Service

Create a new ASP.NET Web Application\*\*:

Open Visual Studio, and create a new project. Select "ASP.NET Web Application" and then choose "Web API".

In the `Models` folder, add a new class named Refer `Book.cs` file

1. \*\*Set Up the Data Context\*\*:

Add a new class named “LibraryContext.cs` in the `Models` folder Refer ‘LibraryContext.cs`

Update the `Web.config` file with the connection string:

1. \*\*Create the Web API Controller\*\*:

In the `Controllers` folder, add a new Web API controller named `BooksController.cs`: Refer the same file

* Step 3: Test the Web Service

1. \*\*Run the application\*\*:

- Press F5 to run the application.

- Use tools like Postman to test the API endpoints (GET, POST, PUT, DELETE).

* Step 4: Create the ASP.NET Web Application for the Frontend

1. Create a new ASP.NET Web Application

Open Visual Studio, and create a new project. Select "ASP.NET Web Application" and then choose "MVC".

2. Set Up the Views and Controllers

. Create views for listing books, adding a new book, updating, and deleting books.

. Use jQuery or JavaScript to make AJAX calls to the Web API endpoints from the views.

3. View (Index.cshtml)\*\*:

* Step 5: Test the Entire Application

1. Run the MVC application

2. Ensure the frontend properly communicates with the backend web service

3. Test all CRUD operations from the web application

By following these steps, you will have a fully functional library management system with CRUD operations using ASP.NET Web API and MVC.