

## Evaluation Points for Project (Online BookStore in .Net 6) Submission

### 1. MVC Concepts:

- Applied proper separation of concerns (Model, View, Controller).
- Book model handles data structure; BooksController manages logic, and Razor Views focus on UI.

### 2. Code Quality:

- Followed clean and readable code practices with proper naming conventions and comments.
- Ensured methods are concise and maintainable.

### 3. Database Interaction:

- Used Entity Framework Core for CRUD operations with efficient LINQ queries.
- Set up migrations to define and manage the database schema.

### 4. Validation:

- Implemented server-side validation with attributes like `[Required]`, `[Range]`, etc.
- Added client-side validation scripts for a seamless user experience.
- Provided clear error messages (e.g., ISBN must be 13 digits, price must be up to 999).

### 5. Routing:

- Used conventional and attribute routing for clean and user-friendly URLs.
- Ensured intuitive navigation (e.g., `/Books/Details/{id}`).

### 6. UI/UX:

- Designed a responsive interface with Bootstrap for consistent styling.
- Navigation bar added for ease of access; buttons and forms are well-styled and spaced.

### 7. Additional Features:

- Implemented pagination (5 books per page).
- Added a search bar for filtering by title or author.
- User-friendly error messages and confirmation dialogs for delete operations.

### 8. NuGet Packages Used:

- Microsoft.EntityFrameworkCore**
  - For database operations and ORM (Object-Relational Mapping) functionality.
- Microsoft.EntityFrameworkCore.SqlServer**

- a. To enable SQL Server database provider for Entity Framework Core.
- iii. **Microsoft.EntityFrameworkCore.Tools**
  - a. For enabling Entity Framework Core tools like migrations and database updates.
- iv. **X.PagedList.Mvc.Core**
  - a. For implementing pagination on the books listing page.
- v. **Microsoft.AspNetCore.Mvc.ViewFeatures**
  - a. To support Razor Views and manage HTML helpers like validation messages.
- vi. **Microsoft.AspNetCore.Mvc**
  - a. Essential for MVC functionality (controllers, views, and routing).

These packages helped with the core functionality of the application, including database management, pagination, MVC structure, and validation handling.