



[CLO 3]

Class: **BSE-6 (A/B)**

Max Marks: **5**

Course Instructor: **Engr. Muhammad Faisal**

[The marks of this assignment may increase or decrease]

Read Carefully:

- The deadline for this assignment is *before or on* **Monday, 27th May, 2023.**

WARNINGS:

- This is an individual assignment; you must implement it by yourself. Any form of plagiarism will result in receiving zero in the assignment.
- Late submission will not be accepted. Any assignment submitted after the cutoff time will receive zero.

In this assignment you are supposed to develop a web API that will contain the following

1) Model:

Class → Book

Attributes:

- a) public int BookId { get; set; }
- b) public string Title { get; set; }
- c) public string Isbn { get; set; }
- d) public string PublisherName { get; set; }
- e) public string AuthorName { get; set; }

2) Repository:

The Repository contain an interface IBookRepository and a class BookRepository

Interface → IBookRepository

Methods:

- a) IEnumerable<Book> GetAll();
- b) Book Get(int id);
- c) Book Add(Book item);
- d) void Remove(int id);
- e) bool Update(Book item);

Methods Description:

This code creates an IBookRepository interface and declares the following methods:

- a) GetAll(): An implementation of this method should return an IEnumerable<Book> object that contains details of all the Book.
- b) Get(int id): An implementation of this method should return an Book object of the specified Id passed as parameters to the method.

- c) Add(Book item): An implementation of this method should add a new Book object to the BookRepository object. Once added, this method should return the new Book object.
- d) Remove(int id): An implementation of this method should remove an Book object specified by the Id passed as parameter from the BookRepository object.
- e) Update(Book item): An implementation of this method should update the BookRepository object with the Book object passed as parameter

BookRepository

class BookRepository : IBookRepository

In this code, the BookRepository class implements the IBookRepository interface that it created. For each of the methods declared in the IBookRepository interface, the BookRepository class provides implementation to retrieve, add, and delete albums that the Album model represents.

3) Controller

Class BookController

- a) The Get() method accesses the Book repository to return all Books as an IEnumerable<Book> object.
- b) The Get(int id) method accesses the Book repository to return an Book with the specified Id as an Book object. Similarly, the GetBookByAutName(string AName) method returns all Books of the specified author as an IEnumerable<Book> object.
- c) The Post(Book Book) method adds the Book object passed as parameter to the Book repository. The Put(int id, Book Book) method updates a book in the book repository based on the specified id.
- d) The Delete(int id) method deletes an Book from the Book repository based on the specified id.

Defining Routes:

Once you have created the Web API controller, you need to register it with the ASP.NET routing Framework. When the Web API application receives a request, the routing Framework tries to match the Uniform Resource Identifier (URI) against one of the route templates defined in the WebApiConfig.cs file.

Book Model:

```
public class Book
{
    public int BookId { get; set; }
    public string Title { get; set; }
    public string Isbn { get; set; }
    public string PublisherName { get; set; }
    public string AuthorName { get; set; }
}
```

IBookRepository:

```
public interface IBookRepository
{
    IEnumerable<Book> GetAll();
    Book Get(int id);
    Book Add(Book book);
    bool Remove(int id);
    bool Update(int id, Book book);
}
```

BookRepository Class:

```
public class BookRepository : IBookRepository
{
    private readonly List<Book> _books = new List<Book>();

    public BookRepository()
    {
        _books.Add(new Book
        {
            BookId = 1,
            Title = "The Alchemist",
            Isbn = "978-0062013381",
            PublisherName = "HarperCollins",
            AuthorName = "Paulo Coelho"
        });

        _books.Add(new Book
        {
            BookId = 2,
            Title = "The Da Vinci Code",
            Isbn = "978-0307289608",
            PublisherName = "Doubleday",
            AuthorName = "Dan Brown"
        });

        _books.Add(new Book
        {
            BookId = 3,
            Title = "Harry Potter and the Sorcerer's Stone",
            Isbn = "978-0439116399",
            PublisherName = "Scholastic Press",
            AuthorName = "J.K. Rowling"
        });
    }

    public IEnumerable<Book> GetAll()
    {
        return _books;
    }
}
```

```

    }

    public Book Get(int id)
    {
        return _books.FirstOrDefault(book => book.BookId == id);
    }

    public Book Add(Book book)
    {
        _books.Add(book);
        return book;
    }

    public bool Remove(int id)
    {
        _books.RemoveAll(book => book.BookId == id);
        return true;
    }

    public bool Update(int id, Book book)
    {
        var existingBook = _books.FirstOrDefault(books => book.BookId == id);
        if (existingBook == null)
        {
            return false;
        }

        existingBook.Title = book.Title;
        existingBook.Isbn = book.Isbn;
        existingBook.PublisherName = book.PublisherName;
        existingBook.AuthorName = book.AuthorName;

        return true;
    }
}

```

WebApiConfig.cs

```

public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.MapHttpAttributeRoutes();

        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{BookController}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}

```