

VIJAY M

Day-12

Date-20/08/25

1. To the existing books table, add additional column called price and add at least ten entries with some price values.

1.1 Create a post request 'GetBooksOrderedByPrice()' with a boolean parameter

1.1.1 when parameter is true, then sort by price ascending order

1.1.2 when parameter is false, then sort by price descending order

1.2 Create a get request 'GetTop5Books()' which will return top 5 costliest books

1.3 Create a get request 'GetCostDetails()' which will return the total price and average price of all books.

Update database

Add-Migration

Update-Database

Source Code:

BookController.cs

```
using Microsoft.AspNetCore.Mvc;
```

```
using WebApi.Context;
```

```
using WebApi.Models;
```

```
namespace WebApi.Controllers
```

```
{
```

```
    [ApiController]
```

```
    [Route("[Controller]")]
```

```
    public class BookController : Controller
```

```
    {
```

```
AppDbContext context { get; set; }

public BookController(AppDbContext context)
```

```
{

    this.context = context;

}
```

```
[HttpGet("getallbooks")]
```

```
public IEnumerable<Book> GetAllBooks()
```

```
{

    //List<Book> allBooks = new List<Book>();

    //allBooks.Add(new Book { Title = "harrypotter", Description = "fantasy", StarRating = 4.5f });

    //allBooks.Add(new Book { Title = "agnisiragu", Description = "biography", StarRating = 4.6f });


    var allBooks = context.books.ToList();

    return allBooks;

}
```

```
[HttpPost("addbooks")]
```

```
public IActionResult AddBooks(Book book)
```

```
{

    context.books.Add(book);

    context.SaveChanges();

    return Ok("Added to the db");

}
```

```
[HttpDelete("deletebyid")]
```

```
public IActionResult deleteById(int id)
```

```

{

    var book = context.books.FirstOrDefault(a => a.Id == id);

    if (book == null)
    {
        return NotFound("book not found");
    }

    else

    {
        context.Remove(book);

        context.SaveChanges();

        return Ok("deleted successfully");
    }

}

[HttpPost("GetBooksOrderedByPrice")]
public IActionResult GetBooksOrderedByPrice([FromBody] bool isAscending)
{
    var books = isAscending
        ? context.books.OrderBy(b => b.Price).ToList()
        : context.books.OrderByDescending(b => b.Price).ToList();

    return Ok(books);
}

[HttpGet("GetTop5Books")]
public IActionResult GetTop5Books()
{
    var top5Books = context.books
        .OrderByDescending(b => b.Price)

```

```

        .Take(5)

        .ToList();

    return Ok(top5Books);
}

[HttpGet("GetCostDetails")]
public IActionResult GetCostDetails()
{
    var totalPrice = context.books.Sum(b => b.Price);

    var averagePrice = context.books.Average(b => b.Price);

    return Ok(new
    {
        TotalPrice = totalPrice,
        AveragePrice = averagePrice
    });
}
}
}

```

Book.cs

```

using System.ComponentModel.DataAnnotations;

namespace WebApi.Models
{
    public class Book
    {

```

```

        public int Id { get; set; }

        [Display(Name = "Name of the books")]
        [StringLength(50, MinimumLength = 5, ErrorMessage = "must be minimum 5 letters")]

        public string Title { get; set; }

        public string Description { get; set; }

        public float StarRating { get; set; }


        public decimal Price { get; set; }

    }
}

```

AppDbContext.cs

```

using Microsoft.EntityFrameworkCore;

using WebApi.Models;

namespace WebApi.Context
{
    public class AppDbContext:DbContext

    {
        public AppDbContext(DbContextOptions<AppDbContext> options):base(options) { }

        public DbSet<Book> books { get; set; }

        public DbSet<User> users { get; set; }

    }
}

```

Program.cs

```

using Microsoft.AspNetCore.Authentication.JwtBearer;

```

```
using Microsoft.EntityFrameworkCore;

using Microsoft.IdentityModel.Tokens;

using System.Net.WebSockets;

using System.Security.Claims;

using System.Security.Cryptography;

using System.Text;

using WebApi.Context;

using WebApi.Security;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();

https://aka.ms/aspnetcore/swashbuckle

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

//const string connectionString = "Data Source=PTPLL605;" +

//      "Initial Catalog=sampled;" +

//      "Integrated Security=True;" +

//      "TrustServerCertificate=True;";

var sql_section = builder.Configuration.GetSection("sql");

string connectionString = sql_section["connection"];

builder.Services.Configure<JWTOptions>(builder.Configuration.GetSection("Jwt"));

var jwt = builder.Configuration.GetSection("Jwt");

var issuer = jwt["Issuer"];

var audience = jwt["Audience"];

var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwt["Key"]!));

builder.Services.AddAuthentication(options =>

{

    options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

    options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
```

```
}).AddJwtBearer(options =>
{
    options.TokenValidationParameters = new TokenValidationParameters
    {
        ValidIssuer = issuer,
        ValidAudience = audience,
        IssuerSigningKey = key,
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateIssuerSigningKey = true,
        ValidateLifetime = true,
        ClockSkew = TimeSpan.Zero,
        RoleClaimType = ClaimTypes.Role,
        NameClaimType = ClaimTypes.NameIdentifier
    };
});

builder.Services.AddDbContext<AppDbContext>(x => x.UseSqlServer(connectionstring));

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

appsettings.json

```
{  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "Microsoft.AspNetCore": "Warning"  
    }  
  },  
  "Jwt": {  
    "Issuer": "https://www.payoda.com",  
    "Audience": "students_CSharp",  
    "Key": "use-a-256-agshagfhassfhasfhasfahsgfashfghsagfgegwwkeg",  
    "ExpiresMinutes": 30  
  },  
  "sql": {  
    "connection": "Data Source=PTPL605;Initial Catalog=sampled;Integrated  
Security=True;TrustServerCertificate=True;"  
  },  
  "AllowedHosts": "*"   
}
```

Output:

/Book/GetBooksOrderedByPrice

form body-true

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7095/Book/GetBooksOrderedByPrice' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d 'true'
```

Request URL

```
https://localhost:7095/Book/GetBooksOrderedByPrice
```

Server response

Code

Details

```
[
  {
    "id": 7,
    "title": "The Great Gatsby",
    "description": "American classic by F. Scott Fitzgerald",
    "starRating": 4.4,
    "price": 10.99
  },
  {
    "id": 11,
    "title": "The Catcher in the Rye",
    "description": "Coming-of-age novel by J.D. Salinger",
    "starRating": 4,
    "price": 12.99
  },
  {
    "id": 6,
    "title": "1984",
    "description": "Dystopian novel by George Orwell",
    "starRating": 4.6,
    "price": 15.5
  },
  {
    "id": 5,
    "title": "To Kill a Mockingbird",
    "description": "Classic novel by Harper Lee",
    "starRating": 4.7,
    "price": 18
  }
]
```

```
{
  "id": 12,
  "title": "Educated",
  "description": "Memoir by Tara Westover",
  "starRating": 4.7,
  "price": 19.49
},
{
  "id": 10,
  "title": "Atomic Habits",
  "description": "Self-help book by James Clear",
  "starRating": 4.9,
  "price": 21.99
},
{
  "id": 8,
  "title": "The Hobbit",
  "description": "Fantasy adventure by J.R.R. Tolkien",
  "starRating": 4.8,
  "price": 22.75
},
{
  "id": 4,
  "title": "Harry Potter and the Sorcerer's Stone",
  "description": "Fantasy novel by J.K. Rowling",
  "starRating": 4.8,
  "price": 24.99
},
{
```

/Book/GetBooksOrderedByPrice

form body-false

Responses

Curl

```
curl -X 'POST' \
'https://localhost:7095/Book/GetBooksOrderedByPrice' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d 'false'
```

Request URL

<https://localhost:7095/Book/GetBooksOrderedByPrice>

```
[
  {
    "id": 1,
    "title": "harry potter",
    "description": "fantacy",
    "starRating": 4.5,
    "price": 550
  },
  {
    "id": 2,
    "title": "agnisiragu",
    "description": "kalam biography",
    "starRating": 4.6,
    "price": 406
  },
  {
    "id": 13,
    "title": "Becoming",
    "description": "Autobiography by Michelle Obama",
    "starRating": 4.9,
    "price": 28.5
  },
  {
    "id": 9,
    "title": "Sapiens: A Brief History of Humankind",
    "description": "Historical analysis by Yuval Noah Harari",
    "starRating": 4.5,
    "price": 27
  }
]
```

```

  {
    "id": 5,
    "title": "To Kill a Mockingbird",
    "description": "Classic novel by Harper Lee",
    "starRating": 4.7,
    "price": 18
  },
  {
    "id": 6,
    "title": "1984",
    "description": "Dystopian novel by George Orwell",
    "starRating": 4.6,
    "price": 15.5
  },
  {
    "id": 11,
    "title": "The Catcher in the Rye",
    "description": "Coming-of-age novel by J.D. Salinger",
    "starRating": 4,
    "price": 12.99
  },
  {
    "id": 7,
    "title": "The Great Gatsby",
    "description": "American classic by F. Scott Fitzgerald",
    "starRating": 4.4,
    "price": 10.99
  }
]
```

/Book/GetTop5Books

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7095/Book/GetTop5Books' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7095/Book/GetTop5Books
```

Server response

Code

Details

200

Response body

```
[
  {
    "id": 1,
    "title": "harry potter",
    "description": "fantasy",
    "starRating": 4.5,
    "price": 550
  },
  {
    "id": 2,
    "title": "agnisiragu",
    "description": "kalam biography",
    "starRating": 4.6,
    "price": 406
  },
  {
    "id": 13,
    "title": "Becoming",
    "description": "Autobiography by Michelle Obama",
    "starRating": 4.9,
    "price": 28.5
  },
  {
    "id": 9,
    "title": "Sapiens: A Brief History of Humankind",
    "description": "Historical analysis by Yuval Noah Harari",
    "starRating": 4.5,
    "price": 27
  }
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 20 Aug 2025 11:33:28 GMT
server: Kestrel
```

/Book/GetCostDetails

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7095/Book/GetCostDetails' \
  -H 'accept: */*'
```

Request URL

<https://localhost:7095/Book/GetCostDetails>

Server response

Code	Details
------	---------

200	
-----	--

Response body

```
{
  "totalPrice": 1158.2,
  "averagePrice": 96.516666
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 20 Aug 2025 11:32:12 GMT
server: Kestrel
```

Responses