

Bank Web API


Example of ASP.Net Core 3.1, MVC, C#, API, JWT, EF Core, LINQ, jQuery, Bootstrap, MSTest, Selenium

Database : SQL Server Express LocalDB, Database Design, PK, FK, Index, Paging

Function : Create account, Show details, Deposit, Withdraw, Transfer, Fee, Transaction List

Source code : <https://github.com/Ligerking007/BankWebAPI>

Bank System					
			API ▾	TRANSACTION ▾	CUSTOMER ACCOUNT ▾
Customer Account List					
Show	10 ▾	entries		Search:	<input type="text"/>
No ↑↓	Account Name ↑↓	Account No. ↑↓	Balance ↑↓	Created Date ↑↓	Created By ↑↓
1	Somchai Jaiyai	00000000000000000004	0.00	25/06/2021 09:53	System
2	Somchai Jaiyai2	00000000000000000003	0.00	25/06/2021 09:51	System
3	Suchat JaiDee	00000000000000000002	0.00	25/06/2021 09:07	System
4	Jakapan Kanta	00000000000000000001	179.80	25/06/2021 08:42	System
Showing 1 to 4 of 4 entries					Previous 1 Next



Swagger

Supported by SMARTBEAR

Select a definition

Bank Web API


Bank Web API

v1

OAS3

/swagger/v1/swagger.json


Authorize



AuthenAPI

POST /api/AuthenAPI/GetToken


Get token for authentication and call API



TransactionAPI


POST /api/TransactionAPI/CreateAccount

Create account by full name and Id card or passport




POST /api/TransactionAPI/GetAccount

Get account detail by account id




POST /api/TransactionAPI/DepositMoney

Deposit money to destination account no




POST /api/TransactionAPI/WithdrawMoney

Withdraw money from source account no




POST /api/TransactionAPI/TransferMoney

Transfer money from source account no to destination account no




POST /api/TransactionAPI/GetTransactionListById

Get all transaction by account id




POST /api/TransactionAPI/GetTransactionList

Get all transaction by grid account id




POST /api/TransactionAPI/GetTransactionListCount

Get transaction count number by account id (for paging)



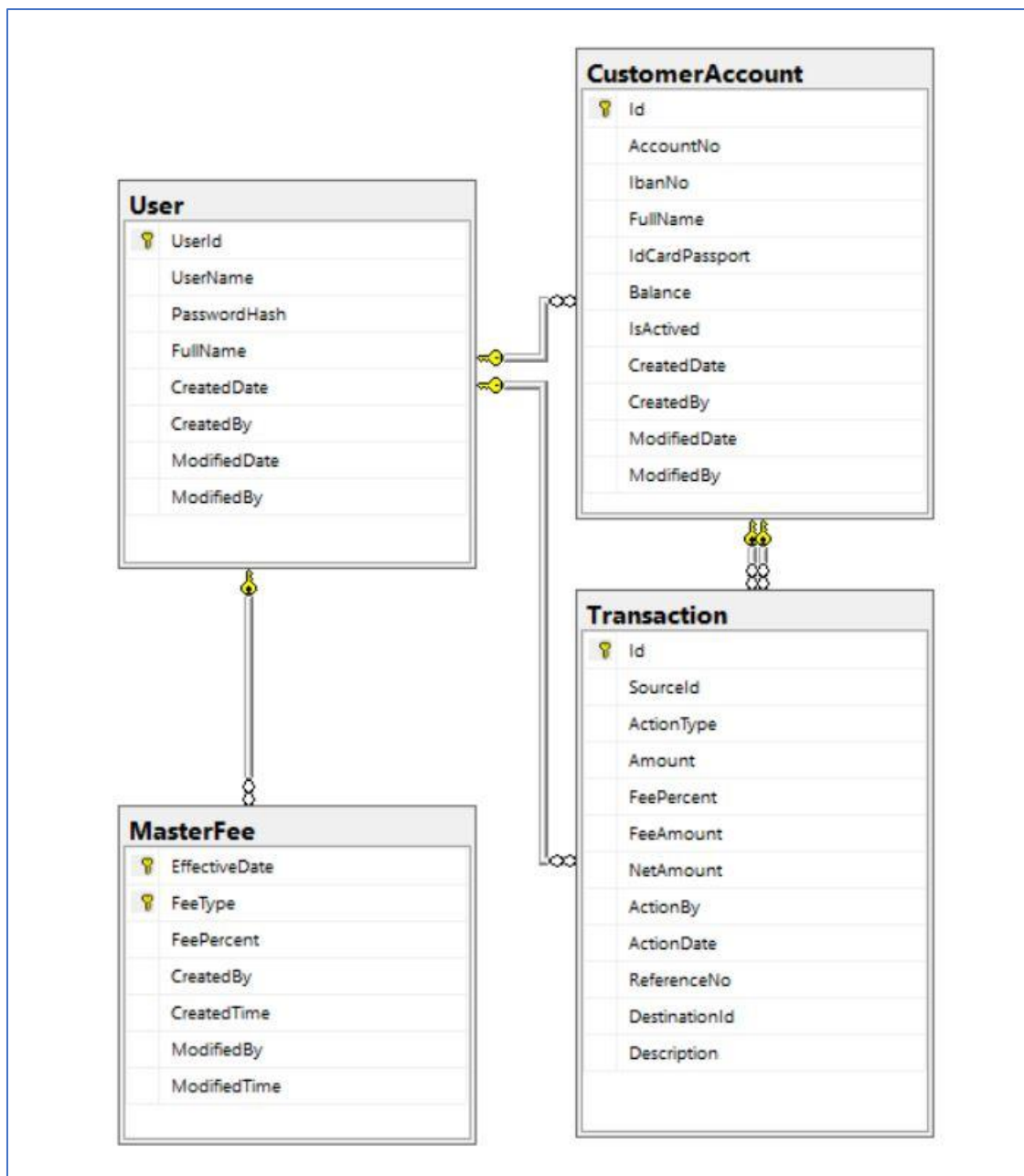
POST /api/TransactionAPI/GetCustomerAccountList

Get all account details



API (Swagger)

Present by Jakapan Kanta



Database Design, PK, FK, Index

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

Create customer account

Customer name

New Normal

ID Card / Passport

1559900111126

Cancel

Create

Create new account

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

Deposit money

Source No

Jakapan Kanta - Rp409.61 ▾

Amount

Rp

200

Fee Percent

Rp

0.10

Fee Amount

Rp

0.2

Total Amount

Rp

199.8

Description (optional)

Cancel

Deposit

Deposit

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

Withdraw money

Source No

Jakapan Kanta - ₪609.41 ▾

Amount

₪ 150

Fee Percent

₪ 0.03

Fee Amount

₪ 0.05

Total Amount

₪ 150.05

Description *(optional)*

Cancel

Withdraw

Withdraw

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

⊙ ▾

Transfer money

Source No

Jakapan Kanta - 00000000000000000001 - ฿459.37 ▾

Destination No

00000000000000000002

Suchat JaiDee - 00000000000000000002 - ฿0.00 ▾

Amount

฿ 300

Fee Percent

฿ 0.05

Fee Amount

฿ 0.15

Total Amount

฿ 300.15

Description (optional)

Cancel Transfer

Transfer

Bank System

API ▾TRANSACTION ▾CUSTOMER ACCOUNT ▾

⌵

Jakapan Kanta

Account details

ID Card / Passport

1559900111123

Current balance

฿159.22

Account number

00000000000000000001

IBAN number

NL40ABNA6579776978

Date of creation

25/06/2021 08:42

Last transactions

Show

10

entries

Search:

No	Action Type	Source No	Destination No	Amount	Fee Percent	Fee Amount	Total Amount	Action Date	Ac
1	Transfer	00000000000000000001	00000000000000000002	300.00	0.05	0.15	300.15	14/07/2021 20:40	฿
2	Withdraw	00000000000000000001		150.00	0.03	0.04	150.04	14/07/2021 20:39	฿
3	Deposit		00000000000000000001	200.00	0.10	0.20	199.80	14/07/2021 20:35	฿
4	Withdraw	00000000000000000001		120.00	0.03	0.04	120.04	14/07/2021 20:11	฿
5	Deposit		00000000000000000001	150.00	0.10	0.15	149.85	14/07/2021 20:09	฿
6	Transfer	00000000000000000003	00000000000000000001	200.00	0.05	0.10	200.10	14/07/2021 20:06	฿
7	Withdraw	00000000000000000001		20.00	0.00	0.00	20.00	29/06/2021 18:16	฿
8	Deposit		00000000000000000001	200.00	0.10	0.20	199.80	25/06/2021 09:11	฿

Showing 1 to 8 of 8 entries

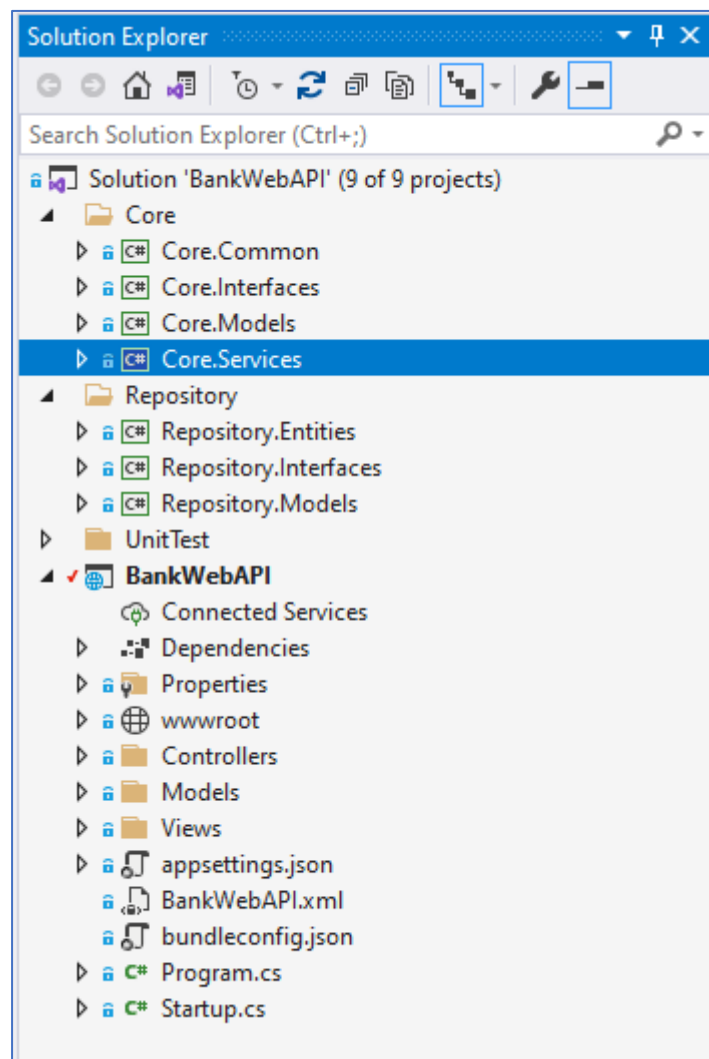
Previous

1

Next

Account details and transaction

Example Source Code



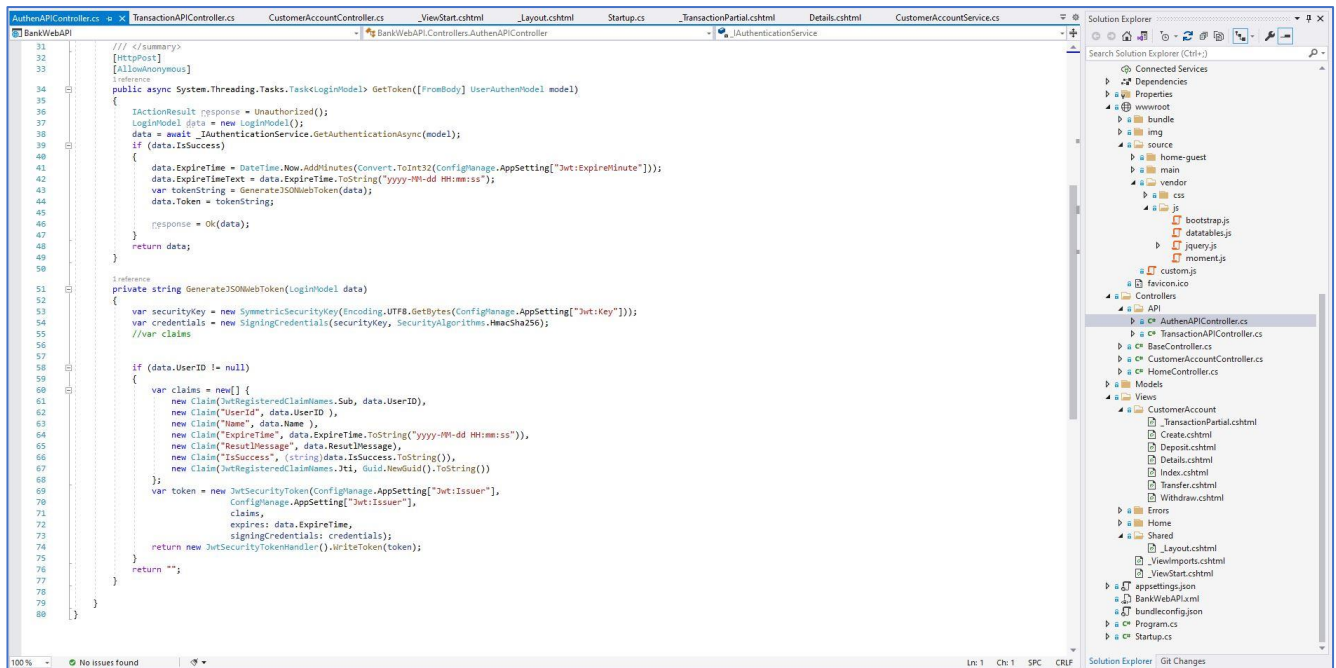
Project Structure

Core : Services, Models, Common libraries

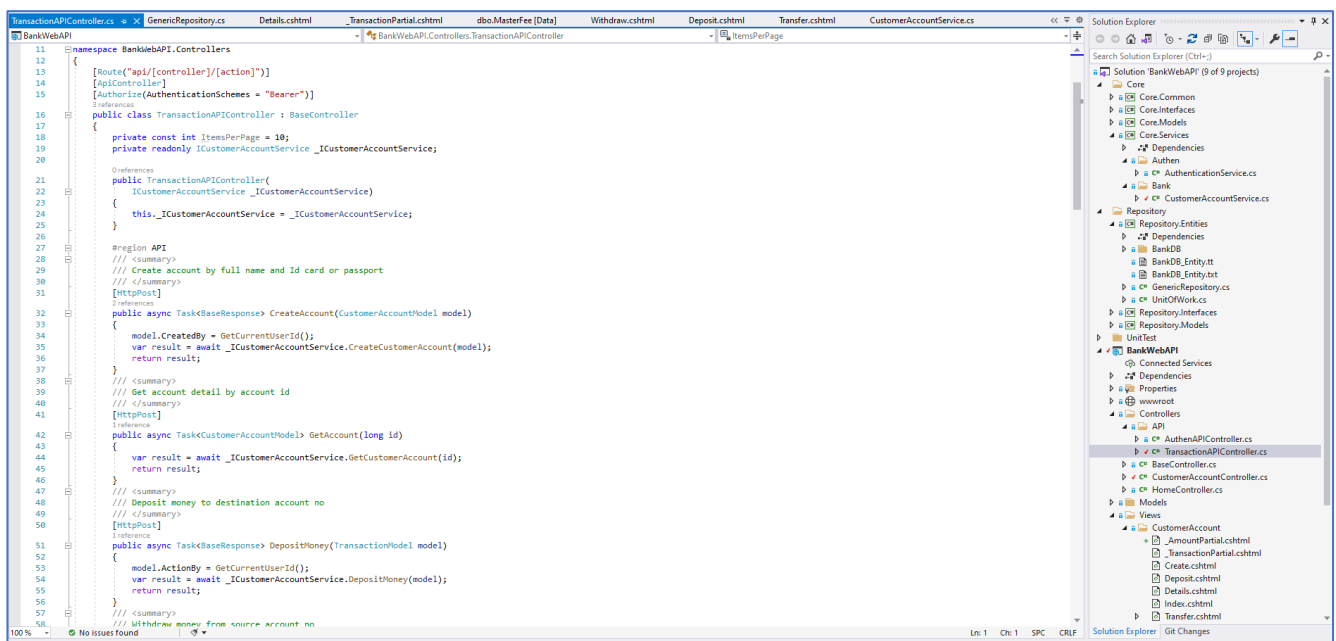
Repository : Access Database

Unit Test

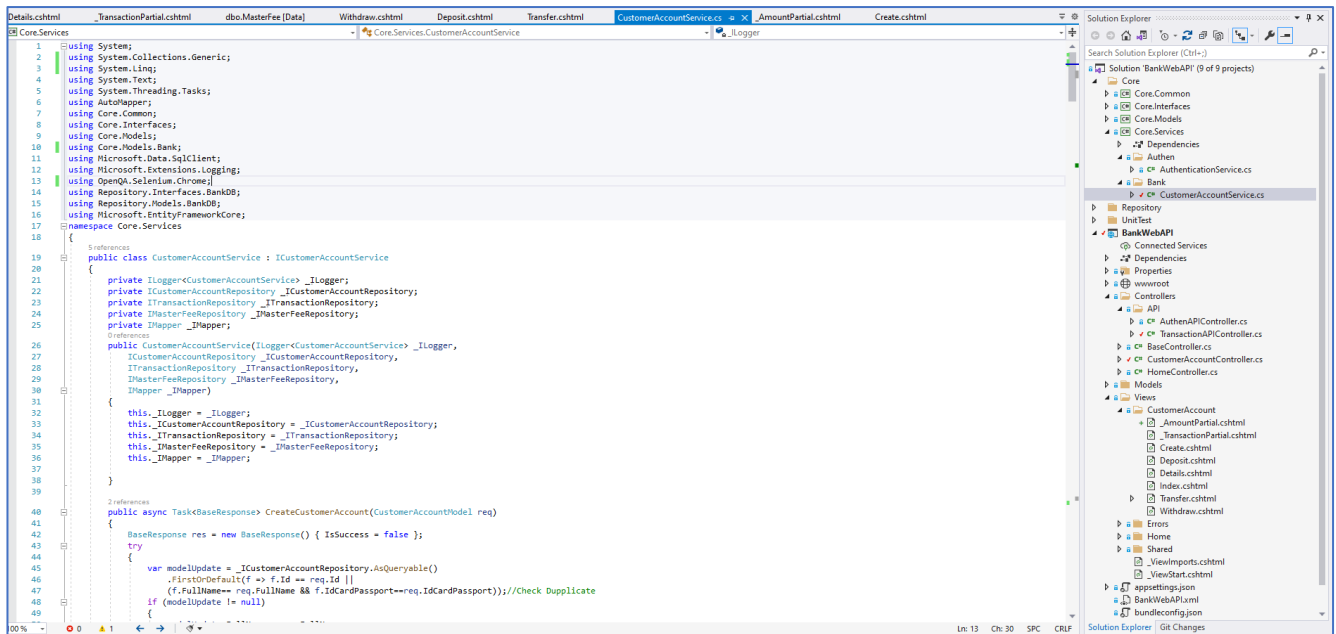
Web MVC & API



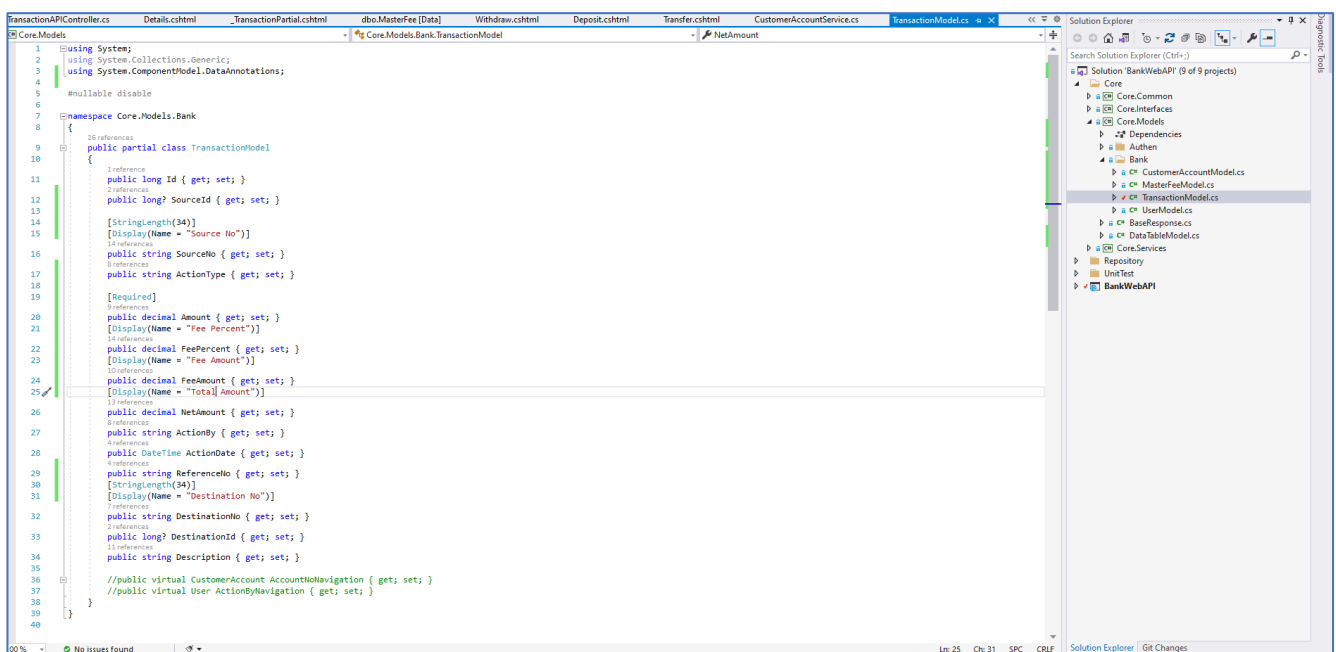
Authentication API



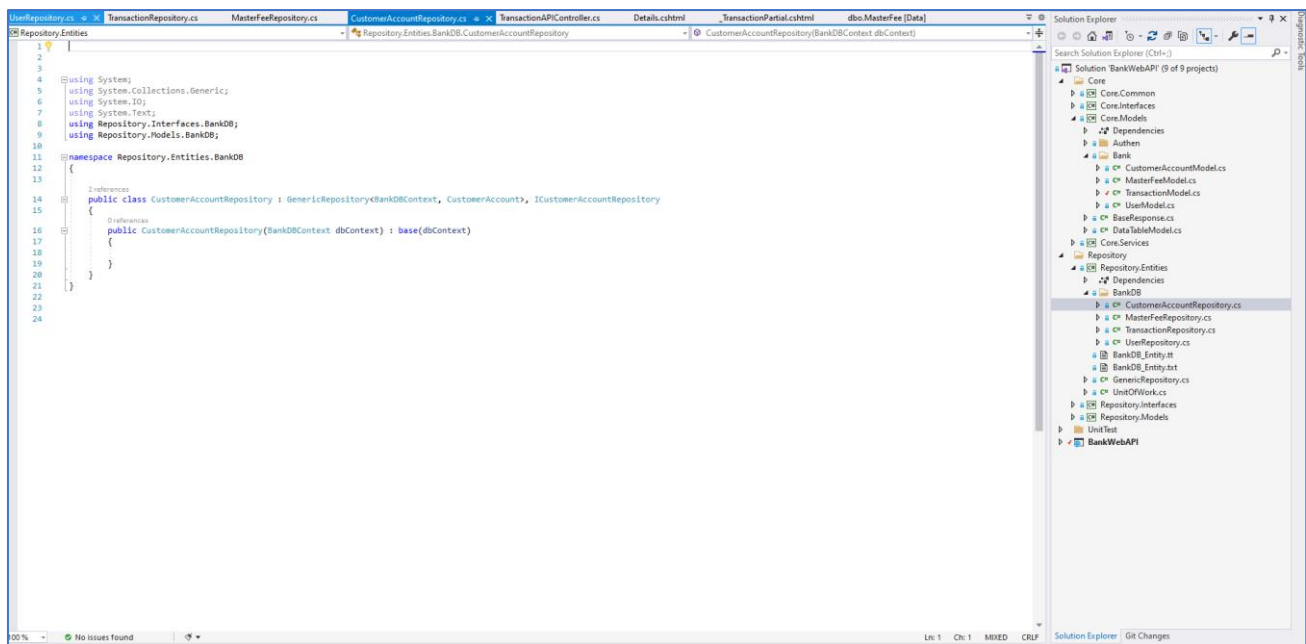
Controller API



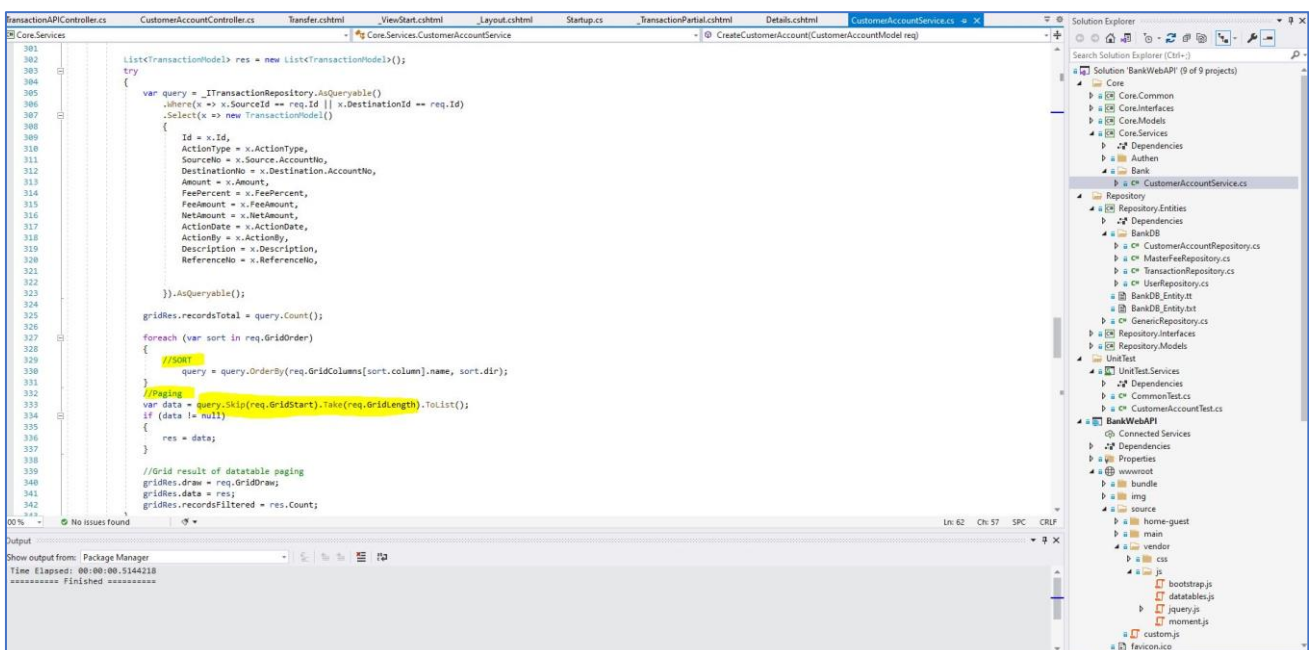
Services & Dependency Injection



Model



Repository



Server Paging and Sorting

```

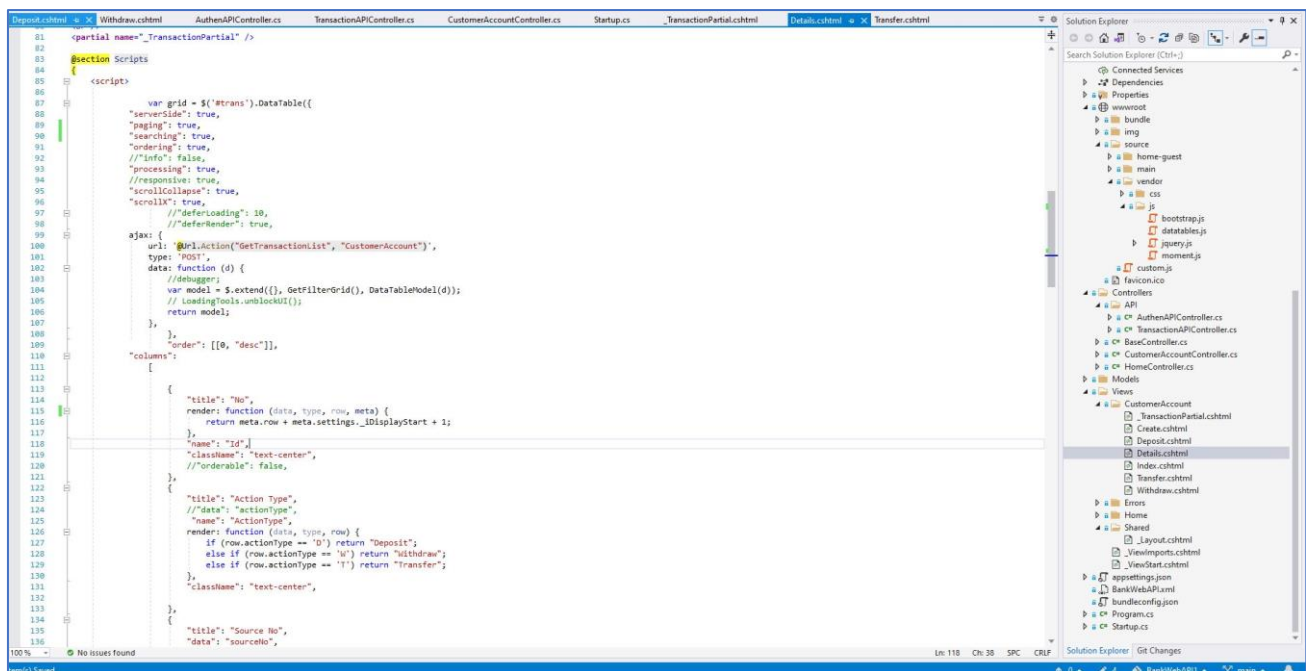
--Execute [dbo].[SP_GetCustomerAccountList] '1','Id','Desc','0','2',1
ALTER PROCEDURE [dbo].[SP_GetCustomerAccountList]
(
    @IsActive bit = 1 ,
    @SortBy nvarchar (100) = null ,
    @SortDirection nvarchar (10) = null ,
    @PageStart int = 0 ,
    @PageSize int = 1000 ,
    @TotalRows int output
)
AS
BEGIN
    -- Build WHERE Clause
    DECLARE @WhereClause [nvarchar] (2000)
    SET @WhereClause = ' IsActive = ' + convert(nvarchar, @IsActive) + ' '
    DECLARE @OrderBy [nvarchar] (100)
    SET @OrderBy = @SortBy + ' ' + @SortDirection
    IF (@SortBy is null or LEN(@SortBy) = 0)
    BEGIN
        -- default order by to first column
        SET @OrderBy = 'Id Desc'
    END
    -- SQL Server 2005 Paging
    declare @SqlCount as nvarchar(max)
    declare @SqlTable as nvarchar(max)
    declare @SQL as nvarchar(max)
    -- get row count
    SET @SqlCount = ' SELECT @TotalRows = COUNT(*) '
    -- sql : from table and where ...
    SET @SqlTable = ' FROM [BankDB].[dbo].[CustomerAccount] '
    IF LEN(@WhereClause) > 0
    BEGIN
        SET @SqlTable = @SqlTable + ' WHERE ' + @WhereClause
    END
    SET @SqlCount = @SqlCount + @SqlTable
    EXEC sp_executesql @SqlCount, N'@TotalRows int OUTPUT', @TotalRows OUTPUT ;
    --print @TotalRows;
    -- Get Data
    SET @SQL = ' SELECT '
    SET @SQL = @SQL + ' ROW_NUMBER() OVER (ORDER BY ' + @OrderBy + ') asRowIndex '
    SET @SQL = @SQL + ' , * --All field '
    -- query @SqlTable : from table where ...
    SET @SQL = @SQL + @SqlTable
    SET @SQL = @SQL + ' ORDER BY ' + @OrderBy
    SET @SQL = @SQL + ' OFFSET ' + convert(nvarchar, @PageStart) + ' ROWS FETCH NEXT ' + convert(nvarchar, @PageSize) + ' ROWS ONLY '
    -- Get Data
    EXEC sp_executesql @SQL
    --PRINT @SQL
END

```

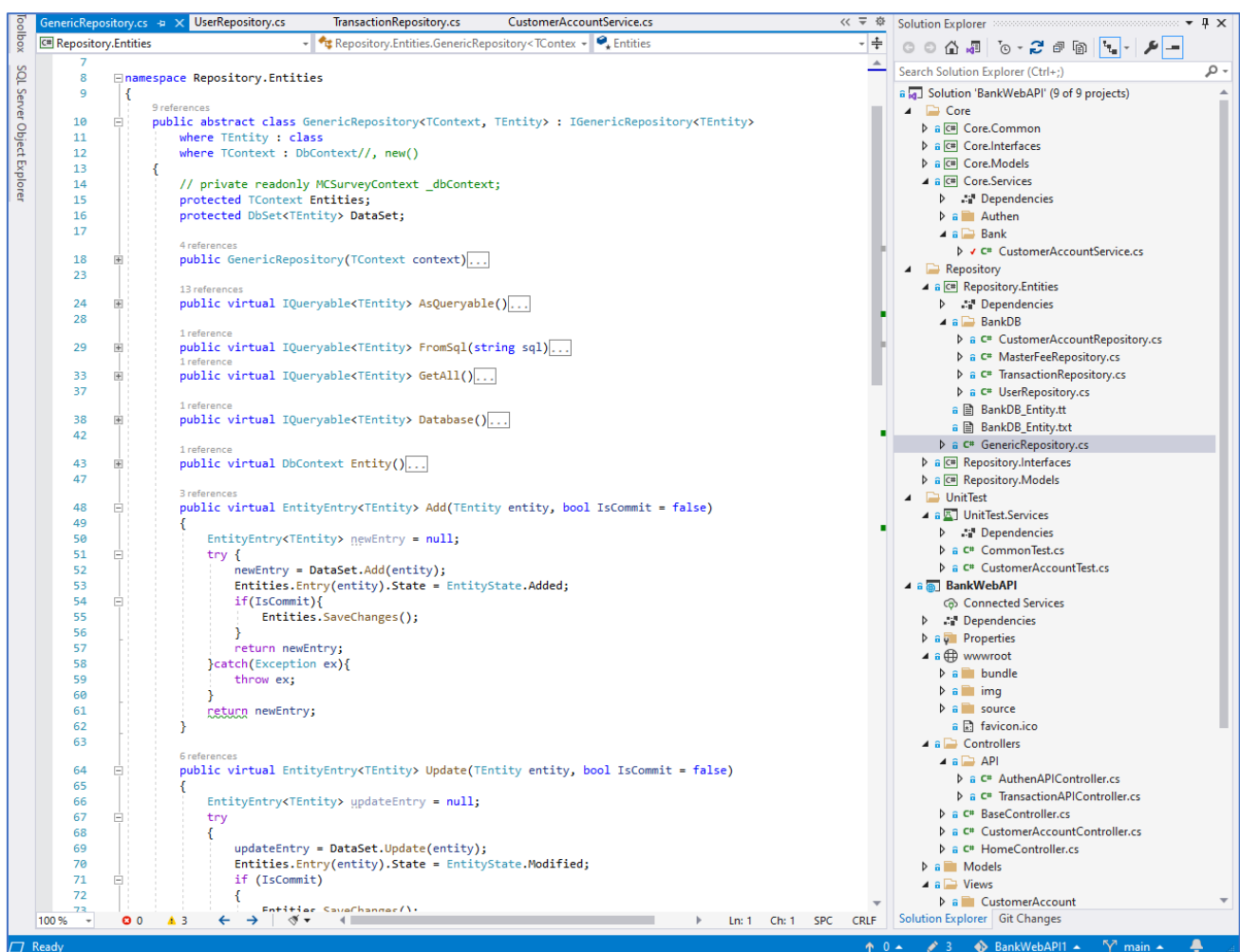
Store Procedure Paging and Sorting

The screenshot displays the Visual Studio IDE with the 'Details.cshtml' file open. The HTML code uses Bootstrap classes to create a modal dialog titled 'Edit account details'. It features a form with a hidden 'accountId' field, a 'Name' text input, and an 'ID Card / Passport' text input. The modal includes 'Close', 'Cancel', and 'OK' buttons. The Solution Explorer on the right shows the project structure for 'BankWebAPI', including folders for Core, Repository, UnitTest, and BankWebAPI, with various sub-files like Controllers, Models, and Views.

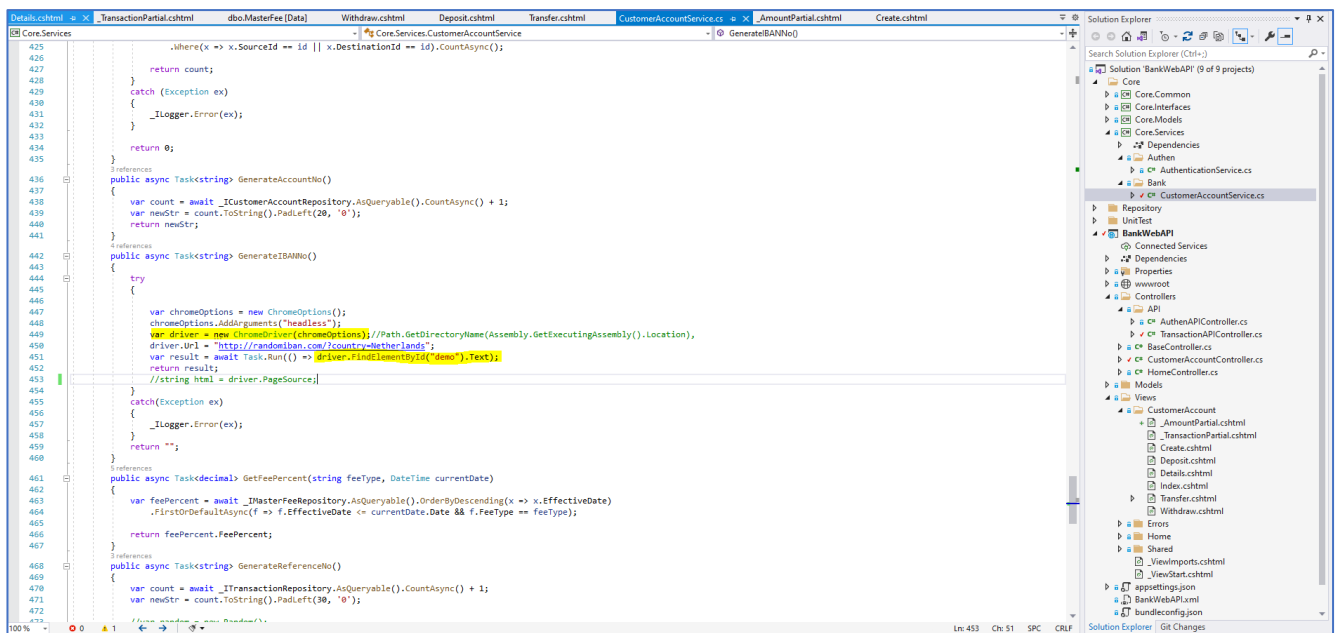
HTML & Bootstrap



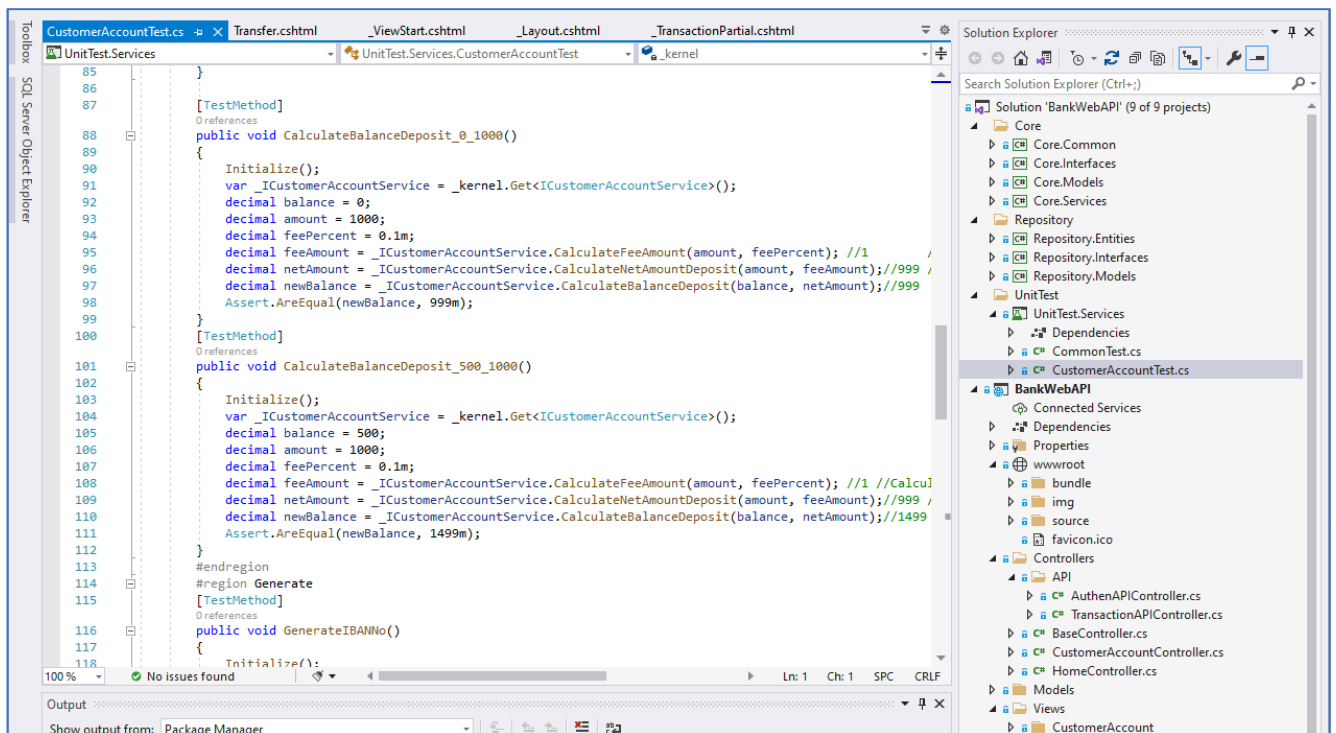
jQuery (DataTables, Grid)



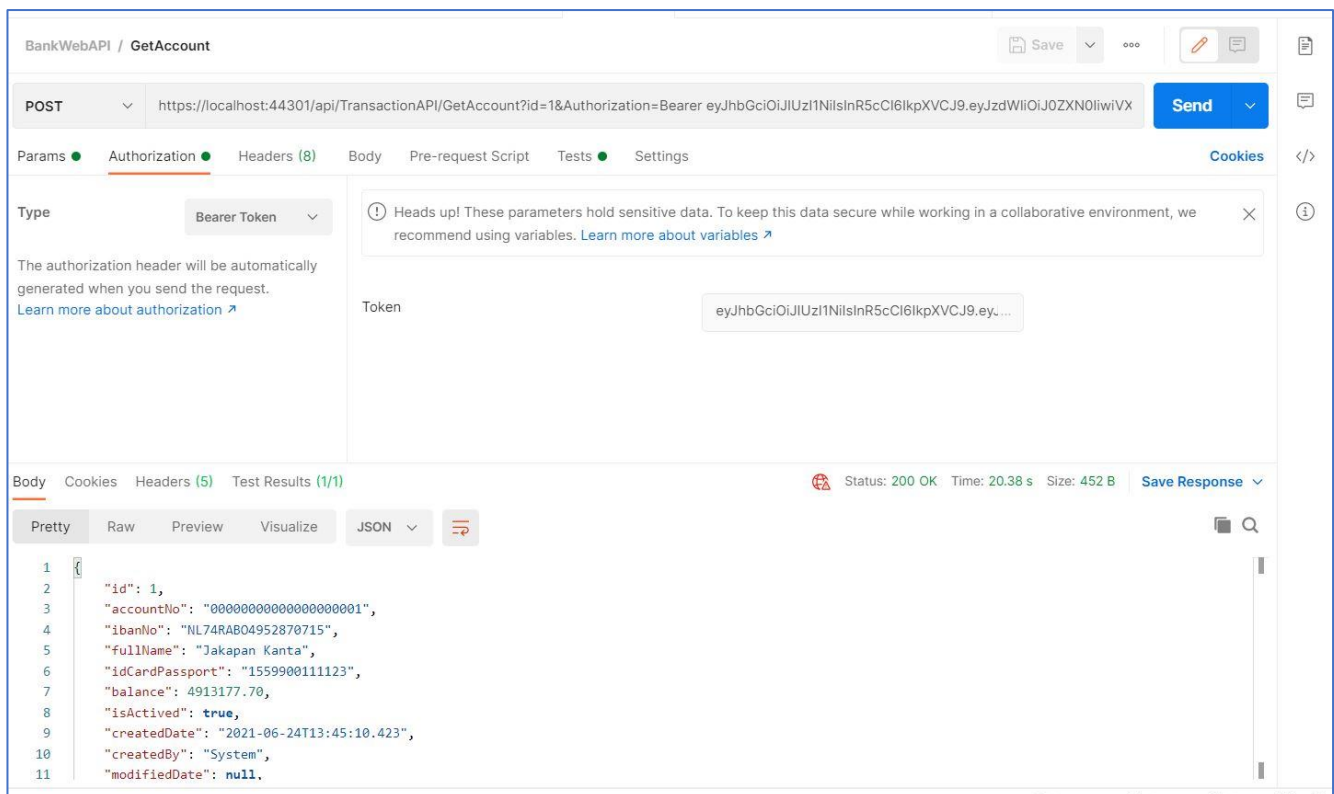
Add new function on TEntity class for CRUD database



Selenium – Get object from website



Unit Test (MSTest)



Postman call API

My technique of coding skill

- OOP concept
- SQL transaction (begin, commit, rollback)
- DB design PK, FK, Indexing
- Server side paging
- Reduce looping
- Reduce connect database
- Parallel run, Split job
- Temp data, Temp memory, Temp table
- Thread async await
- Security coding (OWASP), JWT, reCAPTCHA
- Caching
- Logging
- Memory controlling
- Dispose object (clears memory)
- Garbage collection (clears memory)
- String builder (reusable memory)
- Data conversion (date, number)
- Validation (null, number, date, max length)
- Clean code
- Load balancing
- Scaling