

## Assignment 03 – Summer 2023 HCM

# Apply security, Ajax request with RESTful API and Web Application

### 1. Introduction

A flower bouquet shop is a retail store that specializes in selling flower arrangements. These shops offer a wide variety of flower arrangements and bouquets for different occasions such as birthdays, weddings, anniversaries, and funerals. Customers can choose from a variety of flowers such as roses, lilies, carnations, chrysanthemums, and more. The bouquets can be arranged based on the customer's preferences, including color, flower type, and bouquet size. Some flower bouquet shops also offer delivery services, allowing customers to send flower arrangements to their loved ones. These shops are popular for their ability to provide quick and convenient access to flowers for any occasion. They provide a great option for those looking to make a lasting impression or show appreciation to someone special.

Imagine you're a developer of a FU Flower Bouquet Management System named **FUFlowerSystem**. To implement a part of this system your tasks include:

- Manage customer information.
- Manage flower bouquet information.
- Manage order information.

The application has a default account (admin account) whose email is “**admin@FUflowerbouquet.com**” and password is “**admin@@**” that stored in the **appsettings.json**.

The application has to support adding, viewing, modifying, and removing information - a standardized usage action verb better known as Create, Read, Update, Delete (CRUD) and Search. This assignment explores creating an ASP.NET Core Web API with C#, Identity or JWT and

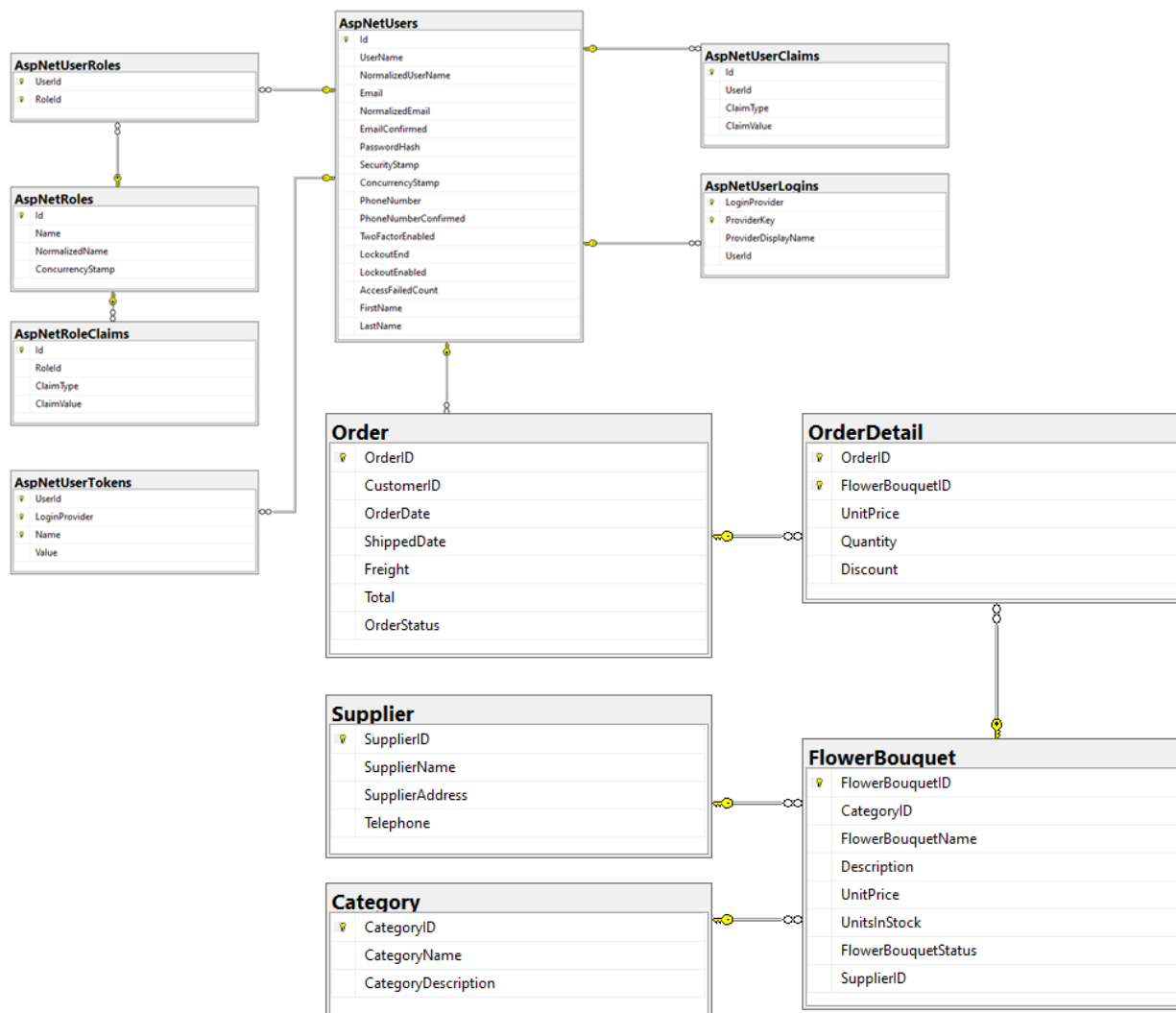
Entity Framework Core, the client application can be used as Web Application (ASP.NET Core Web MVC or Razor Pages) combination with AJAX request. An MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

## 2. Assignment Objectives

In this assignment, you will:

- Use the Visual Studio.NET to create a Web application and ASP.NET Core Web API project.
- Perform CRUD actions using ADO.NET or Entity Framework Core.
- Use LINQ to query and sort data.
- Apply 3-layers architecture to develop the application.
- Apply Repository pattern and Singleton pattern in a project.
- Add CRUD and searching actions to the Web application combination with AJAX request.
- Apply to validate data type for all fields.
- Run the project and test the actions of the Web application.

### 3. Database Design



### 4. Main Functions

- Use Identity to manage User, manage login authentication process, build application with ASP.NET's account registration, login/logout function, configure Identity lockout to lock user if login fails many times.
- Create Web API with JWT for flower bouquet management, and order management: Read, Create, Update and Delete actions.
- Create Client application (Web application) interactive with ASP.NET Core WebAPI using AJAX techniques to perform these functions:

- Search by name (keyword of FlowerBouquetName) and UnitPrice
- Create a report statistics sales by the period from StartDate to EndDate, and sort sales in descending order
- Manage flower bouquet information
- Add flower bouquet to cart, manage cart and make order
- Member (Admin/Customer) authentication by Email and Password. If the user is “**Admin**” then allows to perform all actions, otherwise, the normal user is allowed to view/update the profile, add product to cart, manage cart and view their orders history.

## Notes

- You must use **Visual Studio 2019 or above (.NET5/.NET6/.NET7)**, **MSSQL Server 2012 or above** for your development tools.
- To do your BE program, you must use **ASP.NET Core Web API** with JWT. *Note that you are not allow to connect direct to database from ASP.NET Core Web API, every database connection must be used through Repository and Data Access Objects. The database connection string must get from appsettings.json file.*
- Create Solution in Visual Studio named **StudentName\_ClassCode\_A03.sln**.
- Set the default user interface for your project as **Login** page/window.