**PRN221 Assignment 02 - Summer 2023 HCM**

**Building a Flower Bouquet Management website with ASP.NET Core Razor Pages**

# 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**. Your manager has asked you to develop a website with ASP.NET Core Razor Pages for supplier management, flower bouquet management, customer management, and order management. To implement a part of this system your tasks include:

* Manage customer information.
* Manage flower bouquet information.
* Manage order information.

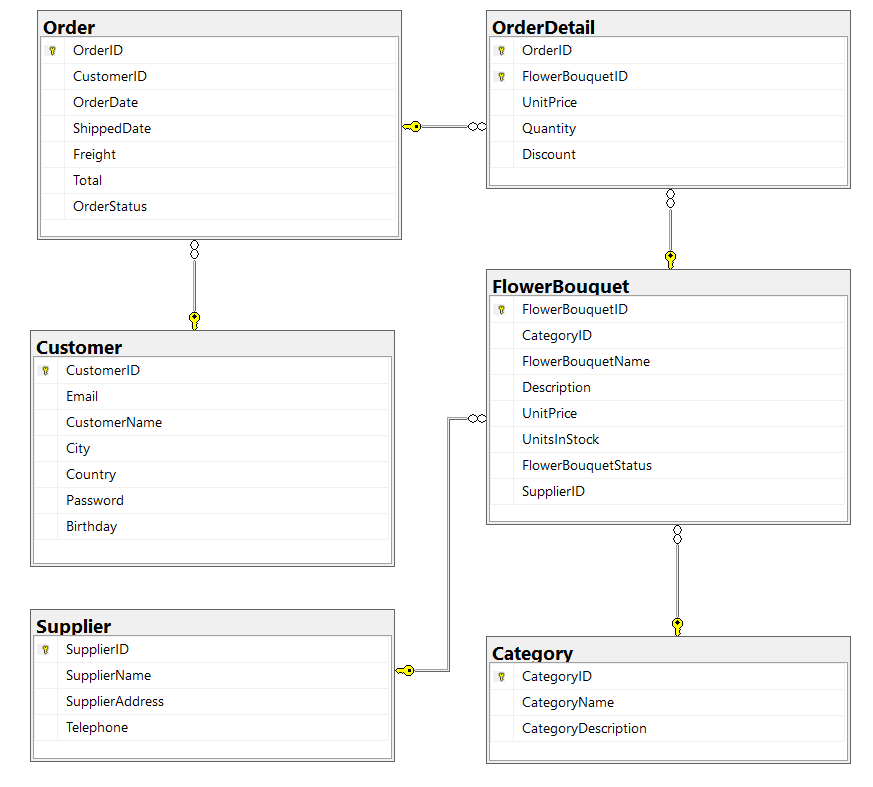
The application has a default account (admin account) whose email is “**admin@FUflowerbouquet.com**” and password is “**admin@@**” that stored in the **appsettings.json**.

# 2. Assignment Objectives

In this assignment, you will:

* Use the Visual Studio.NET to create ASP.NET Core Razor Pages and Class Library (.dll) projects.
* Perform CRUD actions using Entity Framework Core.
* Use LINQ to query and sort data.
* Apply 3-Layers architecture to develop the website.
* Apply Repository pattern and Singleton pattern in a project.
* Add CRUD and searching actions to the website.
* Apply to validate data type for all fields.

# 3. Database Design



A flower bouquet will belong to only one category and only one supplier.

A customer can order many orders. An order will have one or many flower bouquets’ information. A flower bouquet’s information will belong to many orders

# 4. Main Functions

* Member (Admin/Customer) authentication by Email and Password.
* If the user is an “**Admin**” then his/her is allowed to
  + Manage customer information.
  + Manage flower bouquet information. (The delete action will delete flower bouquet information in the case this information is not belong to any order. If the flower bouquet information is already stored in an order, just change the status.)
  + **Manage order information (includes order details management)**
  + Create an order report statistic by the period from StartDate to EndDate, and sort sales in descending order.
* If user is a **“Customer”**, this customer role is allowed to:
  + Register an account.
  + Create an online order with one or many flower bouquets’ information.
  + Manage his/her the profile
  + View order history.
* Note that: Customer management, Flower Bouquet information management, and Order management: Read, Create, Update, Delete and Search actions. Creating and Updating actions must be performed by popup dialog. Delete action always combines with confirmation.

# 5. Note

* You must use **Visual Studio 2019 or above (.NET5/.NET6/.NET7), MSSQL Server 2012 or above** for your development tools.
* To do your program, you must use **ASP.NET Core Razor Pages**. Note that *you are not allow to connect direct to database from Razor pages, 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\_A02.sln**. Inside your Solution, the Project Razor Pages must be named: **StudentNameRazorPages.**
* Create your MS SQL database named **FUFlowerBouquetManagement** by running code in script **FUFlowerBouquetManagement.sql.**
* Set the default user interface for your project as **Login** page/window.