**PRN231 Assignment OData, JWT – Fall 2024 HCM**

**Building ASP.NET Web API OData with JWT and Web Application**

# 1. Introduction

A News Management System (NMS) is a software application that helps universities and educational institutions to efficiently manage, organize, and publish news and content to their website and other channels. The NMS typically includes features such as content creation, approval workflow, scheduling, publishing, and analytics. This can help universities to arrange appropriately their news operations, improve communication with students and the wider community, and better engage with their target audience.

Imagine you're a developer of a News Management System named **FUNewsManagementSystem**. To implement a part of this system your tasks include:

* Manage account information.
* Manage news article.

The application has a default account (admin account) whose email is “**admin@FUNewsManagementSystem.org**” and password is “**@@abc123@@**” 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 OData with C#, JWT and Entity Framework Core, the client application can be used as Web Application (ASP.NET Core  
Web MVC or Razor Pages).* An MS SQL Server database will be  
created to persist the data and it will be used for reading and managing data.

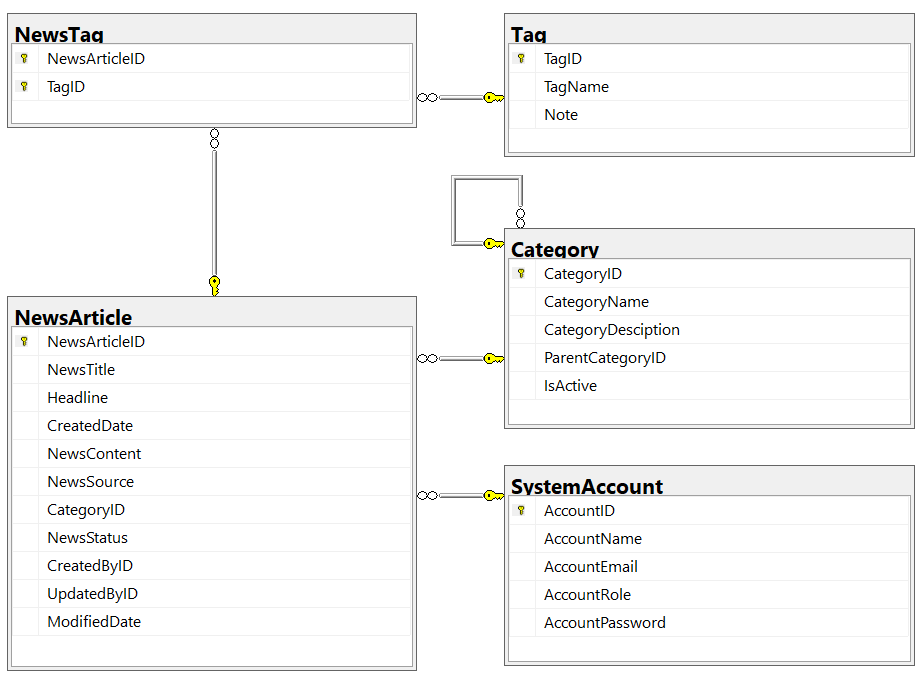
# 2. Assignment Objectives

In this assignment, you will:

* Use the Visual Studio.NET to create a Web application and ASP.NET Core Web API  
  project (with OData support).
* Perform CRUD actions using Entity Framework Core*.*
* Using OData (a data access protocol for the web) to query and manipulate data sets.
* Apply 3-layers architecture to develop the application.
* Apply Repository pattern and Singleton pattern in a project.
* Add CRUD and searching actions to the Client application with ASP.NET Core Web API.
* Apply to validate data type for all fields.
* Run the project and test the actions of the Client Web app and ASP.NET Core Web API

# 3. Database Design

A part of News Management Database:



* A news article will belong to only one news category (Category).
* An account with staff’s role can create many news articles in this system. (Staff role = 1, Lecturer role = 2; Admin role will get from appsettings.json file).
* A news article will have many tags and one tag will belong to zero or one news article (News status = active(1)/inactive(0)).
* Category status = active(1)/inactive(0).

# 4. Main Functions

* *Do not need authentication to view the news article (news status must be active) in this system.*
* Member (Admin/Staff) authentication by Email and Password.
* Create **Web API OData** with **JWT**:
  + **“Admin”** role: Account management (CRUD, the delete action will delete an account in the case this account is not belong to any news articles (created), if the account is already created any news article, cannot delete.)
  + **“Staff”** role:
    - Category management (The delete action will delete an item in the case this item is not belong to any news articles. If the item is already stored in a news article cannot delete.)
    - , and News article management (Read, Create, Update and Delete actions).
* Create Client application (with Desktop/Web application) interactive with Web API to  
  perform these functions:
  + If the user is an “**Admin**” then his/her is allowed
    - Manage account information.
    - Create a report statistic by the period from StartDate to EndDate (it depends to the news’ created date), and sort data in descending order.
  + If user is a “**Staff**”, this staff role is allowed to:
    - Manage category information.
    - Manage news article (includes tags)
    - Manage his/her the profile.
    - View news history created by him/her.
  + Note that: News article Management, Account Management, and Category 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/.NET8), MSSQL Server 2012 or above** for your development tools.
* To do your BE program, you must use **ASP.NET Core Web API OData with JWT**. Note that *you are not allow to connect direct to database from API Controller, every database connection must be used through Repository and Data Access Objects. The database connection string must get from appsettings.json file.*
* Create 2 Solutions in Visual Studio named **StudentName\_ClassCode\_Fall24BE.sln, StudentName\_ClassCode\_ Fall24FE.sln**.