**PRACTICAL EXAMINATION 01**Subject code: PRN221 – FA2024  
Duration: 80 minutes

Develop an Art Gallery Management System using a WPF application and an ASP.NET Core Razor Pages application. These applications will interact with a shared database managed using Entity Framework Core.

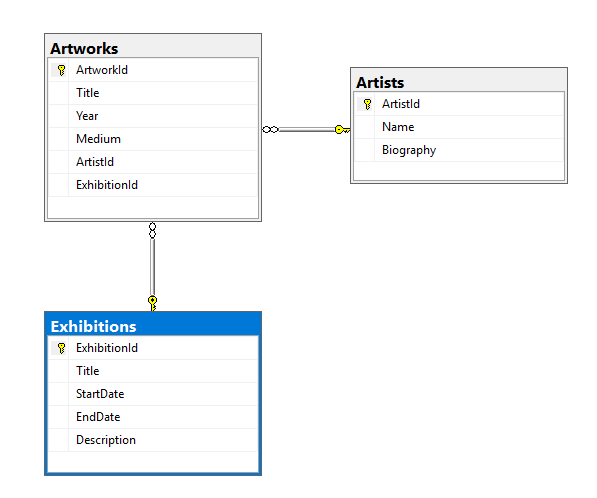
**Solution Name**: FA24\_PRN221

**Projects to Create**:

* **GalleryBusiness** - Manages the DbContext and Entity Framework Core configuration.
* **GalleryDataAccess** - Implements business logic for managing exhibitions, artists, and artworks.
* **GalleryRepositories** - Defines and implements repository interfaces for exhibitions, artists, and artworks.

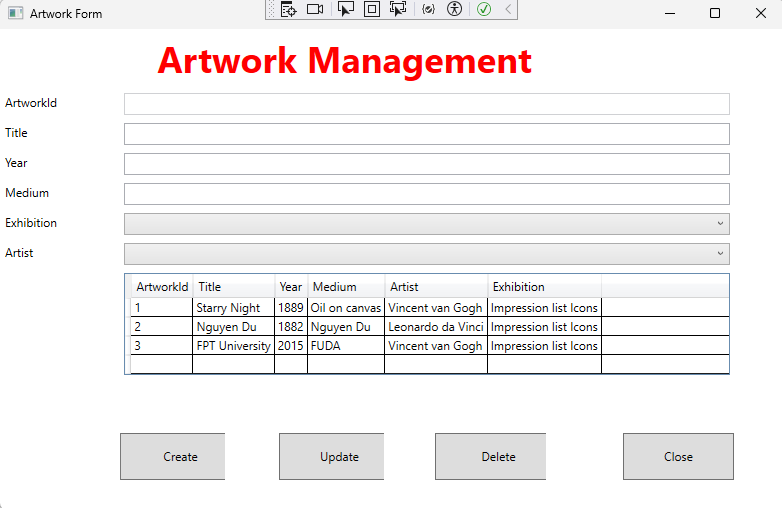
1. **Question 1** (2.5 marks): Project Setup and Database Design

* **Create the Required Projects**: GalleryBusiness, GalleryDataAccess, GalleryRepositories (as Projects to Create) (1.5 marks)
* Design and implement the database using the **Code First approach** with the following entities (0.5 mark):
  + **Exhibitions**: ExhibitionId, Title, StartDate, EndDate, Description
  + **Artists**: ArtistId, Name, Biography
  + **Artworks**: ArtworkId, Title, Year, Medium, ArtistId
* **Establish relationships:** Ensure artworks are correctly linked to artists and included in exhibitions (0.5 mark).



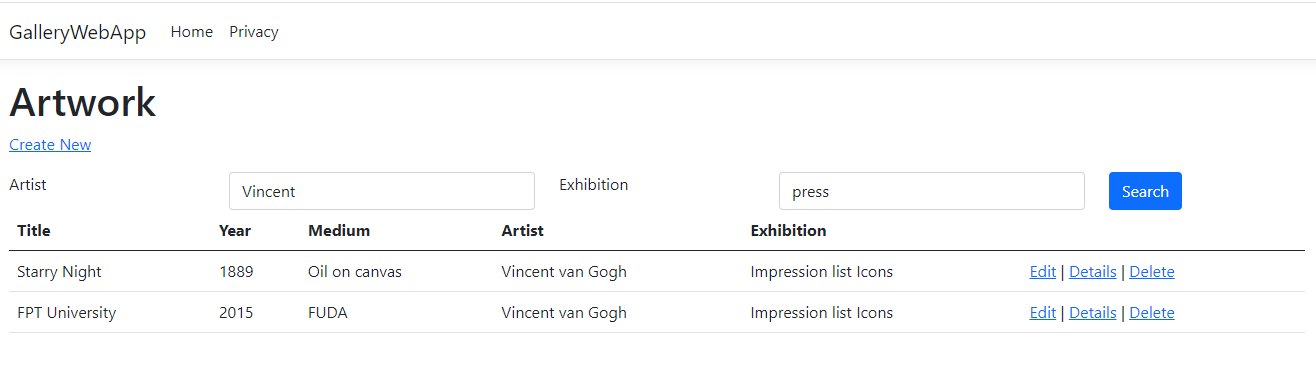
1. **Question 2: WPF Application Development** (2 marks)

* Develop a WPF application named GalleryWPFApp.
* Construct a user interface for managing artwork details within exhibitions, with capabilities to:
  + Create, Update, and Delete artworks.
  + View a list of artworks in a DataGrid with data binding.



1. **Question 3: ASP.NET Core Razor Pages Application** (5.5 marks):

* Develop an ASP.NET Core Razor Pages application named GalleryWebApp.
* Display Artworks: Create a page to list all artworks, including sorting and filtering capabilities:
  + Artworks should be listed with details such as title, year, medium, artist, and exhibition (1 mark).
  + Sort Artworks by Title in Descending Order: Ensure that the main artwork listing page includes functionality to sort the artworks by their title in descending order (0.5 mark).
  + Include dropdown filters for artist and exhibition (1 mark).
* CRUD Operations: Implement pages that allow users to (3 marks):
  + Add New Artworks: Provide a form to input details about new artworks, including associated artist and exhibition.
  + Edit Artworks: Allow users to modify existing artwork details.
  + Delete Artworks: Enable users to remove artworks from the database.
  + View Detailed Information: Each artwork should have a 'Details' page showing comprehensive information.



**Important Note:**

* Functionality Requirement: Your submission must be fully functional. Projects that do not compile or run will receive a score of zero. Ensure that all features are implemented correctly and tested thoroughly before submission.
* Incorrect naming of the Solution or Project will result in a deduction of 1 point from the total score. Additionally, failure to include a backup SQL script will also result in a deduction of 1 point.

**Submission Requirements:**

* Submit a single compressed file (FPTUniversity.zip) containing all project files.
* Include a backup of the SQL Server database script.
* Ensure no personal identifiers (name or student ID) are included in the submission.

**Additional Instructions:**

* You may use local resources but restrict internet use to downloading the exam, submitting your work, and installing packages via Nuget.
* Adherence to the project structure (Business, DataAccess, Repositories) is crucial for full marks.