FINAL ASSIGNMENT

In this assignment, you will develop a Course Management System, in which the records are properly stored in an **SQL database.** This application is similar to the one you have developed for the lab assignment of Module 11. You are required to create the application from scratch including the user interface.

The goal of this application is to allow users create departments (in a university) and add courses to department. A department should have name, abbreviation, and id properties. The id should be automatically generated in the database. For a selected department, the users should be able to add new courses. A course should have title, course code, type (elective vs mandatory), and capacity, and id properties.

Based on this description, you are supposed to create **two** classes: Department and Course. Then, you need to configure a 1-TO-MANY relationship between them: A department can contain/offer **MANY** courses, while a course can be delivered by only **ONE** department. You are required to use Entity Framework Code First to create and manage the database. Using Nuget Package manager, you should install the following libraries to your project:

- EntityFrameworkCore,
- EntityFrameworkCore.Tools,
- EntityFrameworkCore.Design,
- EntityFrameworkCore.SqlServer

You should create a DbContext file and add the necessary code in there (DbSet properties and the connection string). Also, please add the proper *Navigation* properties in the Department and Course classes to configure the 1-To-Many relationship. Do not forget to use the **ForeignKey** data annotation to explicitly indicate a custom foreign key property.

Error Handling When Creating Department and Courses

The application should print an error message if any of the fields are empty and should allow creating a department or course record. Also, the application should not allow entering a non-numeric value to the capacity field (when creating a course). If all inputs are provided properly, then the corresponding record should be created in the database and a confirmation message should be displayed.

The Interface

You are responsible for creating a proper interface for the application. All controls in the form MUST have a proper name. For any improper names, you will lose points. Through the interface, the user should be able to perform these operations:

1. Create a department,
2. List the departments
    a. The departments should be listed with abbreviation and department name combined, such as *[CENG] Computer Engineering*,
    b. When the program is run, the existing departments should be listed,

    c. When a new department is added, the list should be updated immediately.

3. Create a course for the selected department
4. List the courses for the selected department
    a. The courses should be listed with department abbreviation, course code, and course title combined, such as *[CENG 419] : Web Design*,
    b. Whenever a department is selected, the existing courses should be listed,
    c. When a new course is added, the list should be updated.
5. Delete selected courses
    a. The user should be able to select multiple courses for deletion
6. Delete a selected department
    a. The user can delete a single department at a time,
    b. The deletion of a department should be allowed only if the department has no course. Otherwise, an error message should be displayed.
7. Filter departments by their names
8. Filter courses by their titles

Other Requirements

Your code MUST contain the following components at least once:

- method,
- for loop,
- if/else statement.

Remember that for any change in the classes you should create a new migration and update the database.