TRƯỜNG ĐẠI HỌC AN GIANG

A yellow circle with blue and green text

AI-generated content may be incorrect.KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

Lập trình .Net (COS525)

XÂY DỰNG ỨNG DỤNG QUẢN LÝ HỌC SINH

VỚI VISUAL STUDIO VÀ SQL SEVER

Giảng viên hướng dẫn

Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện

Tạ Hồng Quân – DTH235741 – DH24TH2

Đoàn Thanh Phong – DTH235738 – DH24TH2

AN GIANG, 26/11/2025

MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 5](#_Toc214875333)

[1.1. Đặt vấn đề 5](#_Toc214875334)

[1.2. Mục tiêu đề tài 5](#_Toc214875335)

[1.3. Phạm vi nghiên cứu 5](#_Toc214875336)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc214875337)

[2.1. Ngôn ngữ lập trình C# và Nền tảng .NET 6](#_Toc214875338)

[2.2. Môi trường Windows Forms Application (WinForms) 6](#_Toc214875339)

[2.3. Hệ quản trị cơ sở dữ liệu MS SQL Server 6](#_Toc214875340)

[2.4. Công nghệ ADO.NET và Kiến trúc N-Layer 6](#_Toc214875341)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 7](#_Toc214875342)

[3.1. Phân tích yêu cầu chức năng 7](#_Toc214875343)

[3.2. Thiết kế Cơ sở dữ liệu 7](#_Toc214875344)

[Khởi tạo database: 7](#_Toc214875345)

[Code tạo các bảng: 7](#_Toc214875346)

[CHƯƠNG 4. CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM 12](#_Toc214875347)

[4.1. Cài đặt Môi trường và Công nghệ 12](#_Toc214875348)

[4.2. Các Chức năng Chính và Kết quả Cài đặt 12](#_Toc214875349)

[4.2.1. Màn hình Chính (Main Form) và Cơ chế Phân quyền (RBAC) 12](#_Toc214875350)

[4.2.1. Màn hình Chính (Main Form) và Cơ chế Phân quyền (RBAC) 13](#_Toc214875351)

[4.2.2. Chức năng Thời khóa biểu Đa ngữ cảnh (frmThoikhoabieu.cs) 13](#_Toc214875352)

[4.2.3. Quản lý Điểm số (frmQuanLyDiem.cs và frmbangdiemcanhan.cs) 13](#_Toc214875353)

[4.2.4. Quản lý Học sinh (frmHS.cs) và Tính toàn vẹn dữ liệu 14](#_Toc214875354)

[4.2.5. Quản lý Giáo viên (frmQuanLyGiaoVien.cs) và Ràng buộc Khóa ngoại 14](#_Toc214875355)

[4.2.6. Các from quản lý giao diện 15](#_Toc214875356)

[Vai trò học sinh: 22](#_Toc214875357)

[4.2.6. Các code quản lý 24](#_Toc214875358)

[Code kết nối: 24](#_Toc214875359)

[namespace Quanlyhocsinh 24](#_Toc214875360)

[Code program: 24](#_Toc214875361)

[Code form đăng nhập 25](#_Toc214875362)

[Code from trang chủ 27](#_Toc214875363)

[Code form quản lý học sinh 31](#_Toc214875364)

[Code form thông tin cá nhân 42](#_Toc214875365)

[Code form quản lý giáo viên 48](#_Toc214875366)

[Code form quản lý điểm 55](#_Toc214875367)

[Code form bảng điểm cá nhân 62](#_Toc214875368)

[Code form thời khóa biểu 73](#_Toc214875369)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 80](#_Toc214875370)

[5.1. Kết quả Đạt được (Achieved Results) 80](#_Toc214875371)

[5.2. Hạn chế (Limitations) 81](#_Toc214875372)

[5.3. Hướng phát triển (Future Development) 81](#_Toc214875373)

**MỤC LỤC HÌNH ẢNH**

[Hình 1. Sơ đồ quan hệ thực thể của hệ thống 7](#_Toc214874875)

[Hình 2. Giao diện đăng nhập và điều hướng hệ thống 15](#_Toc214874876)

[Hình 3. Giao diện sau khi đăng nhập tài khoản admin 16](#_Toc214874877)

[Hình 4. Giao diện sau khi nhấn vào quản lý của tài khoản admin 16](#_Toc214874878)

[Hình 5. Giao diện sau khi nhấn xem thông tin chi tiết học sinh của tài khoản admin 17](#_Toc214874879)

[Hình 6. Giao diện sau khi nhấn quản lý giáo viên của tài khoản admin 17](#_Toc214874880)

[Hình 7. Giao diện sau khi nhấn quản lý điểm của tài khoản admin 18](#_Toc214874881)

[Hình 8. Giao diện sau khi nhấn xem thông tin chi tiết điểm học sinh của tài khoản admin 18](#_Toc214874882)

[Hình 9. Giao diện sau khi nhấn quản thời khóa biểu của tài khoản admin 19](#_Toc214874883)

[Hình 10. Giao diện sau khi đăng nhập tài khoản giáo viên 19](#_Toc214874884)

[Hình 11. Giao diện sau khi nhấn vào quản lý của tài khoản giáo viên 20](#_Toc214874885)

[Hình 12. Giao diện sau khi nhấn xem thông tin chi tiết học sinh của tài khoản giáo viên 20](#_Toc214874886)

[Hình 13. Giao diện sau khi nhấn quản lý điểm của tài khoản giáo viên 21](#_Toc214874887)

[Hình 14. Giao diện sau khi nhấn xem thông tin chi tiết điểm học sinh của tài khoản giáo viên 21](#_Toc214874888)

[Hình 15. Giao diện sau khi nhấn quản thời khóa biểu của tài khoản giáo viên 22](#_Toc214874889)

[Hình 16. Giao diện sau khi đăng nhập tài khoản học sinh 22](#_Toc214874890)

[Hình 17. Giao diện sau khi nhấn quản lý học sinh của tài khoản học sinh 23](#_Toc214874891)

[Hình 18. Giao diện sau khi nhấn quản thời khóa biểu của tài khoản học sinh 23](#_Toc214874892)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1. Đặt vấn đề

Trong bối cảnh chuyển đổi số hiện nay, việc ứng dụng công nghệ thông tin vào quản lý trường học là vô cùng cấp thiết. Tại nhiều trường học, công tác quản lý hồ sơ học sinh vẫn còn thực hiện thủ công hoặc sử dụng các công cụ rời rạc như Excel, dẫn đến khó khăn trong việc tra cứu, thống kê và bảo mật thông tin. Đề tài "Xây dựng ứng dụng Quản lý Học sinh với C# và SQL Server" được thực hiện nhằm giải quyết các vấn đề trên, mang lại công cụ quản lý hiệu quả, chính xác và trực quan cho cán bộ quản lý.

## 1.2. Mục tiêu đề tài

Mục tiêu chính của đồ án là xây dựng một phần mềm Desktop (WinForms Application) hoàn chỉnh giúp số hóa quy trình quản lý. Cụ thể, hệ thống tập trung vào việc quản lý toàn diện thông tin hồ sơ học sinh, cung cấp đầy đủ các chức năng nghiệp vụ cốt lõi (thêm mới, cập nhật, xóa, tìm kiếm). Bên cạnh đó, hệ thống hướng tới việc lưu trữ dữ liệu tập trung và an toàn trên hệ quản trị cơ sở dữ liệu MS SQL Server, đồng thời cung cấp một giao diện người dùng thân thiện, dễ sử dụng được xây dựng trên nền tảng .NET Framework/Core.

## 1.3. Phạm vi nghiên cứu

Về đối tượng quản lý, đề tài tập trung xử lý các thông tin cá nhân và quá trình học tập của học sinh cấp Trung học phổ thông (THPT). Về công cụ và kỹ thuật thực hiện, phần mềm được phát triển dựa trên Ngôn ngữ lập trình C#, sử dụng môi trường Windows Forms Application (.NET) để thiết kế giao diện, kết hợp với thư viện ADO.NET và SqlClient để kết nối và thao tác với hệ quản trị cơ sở dữ liệu MS SQL Server.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Ngôn ngữ lập trình C# và Nền tảng .NET

C# (C-Sharp) là một ngôn ngữ lập trình hướng đối tượng (Object-Oriented Programming - OOP) mạnh mẽ, được phát triển bởi Microsoft. C# là ngôn ngữ chính trong hệ sinh thái .NET Framework hoặc .NET Core/5+, cung cấp một môi trường thực thi (CLR) ổn định và thư viện lớp (FCL) phong phú. Trong đồ án này, C# đóng vai trò là ngôn ngữ xử lý logic nghiệp vụ (Business Logic) và kiến tạo toàn bộ cấu trúc ứng dụng Desktop.

## 2.2. Môi trường Windows Forms Application (WinForms)

Windows Forms (WinForms) là framework giao diện người dùng đồ họa (Graphical User Interface - GUI) chính thức của .NET để xây dựng các ứng dụng Desktop cho Windows. WinForms cung cấp mô hình lập trình hướng sự kiện (Event-Driven Programming) và bộ công cụ (Controls) tích hợp sẵn như TextBox, Button, và đặc biệt là DataGridView để hiển thị và thao tác dữ liệu bảng một cách hiệu quả.

## 2.3. Hệ quản trị cơ sở dữ liệu MS SQL Server

MS SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft. Nó sử dụng ngôn ngữ Transact-SQL (T-SQL) để quản lý, truy vấn và thao tác dữ liệu. Trong hệ thống này, SQL Server đảm nhận vai trò lưu trữ bền vững, đảm bảo tính toàn vẹn dữ liệu (Data Integrity) thông qua các ràng buộc khóa chính/khóa ngoại và cơ chế Transaction (sử dụng trong các thao tác thêm/sửa phức tạp).

## 2.4. Công nghệ ADO.NET và Kiến trúc N-Layer

ADO.NET là tập hợp các thư viện được sử dụng để truy cập và thao tác với dữ liệu trong môi trường .NET. Cụ thể, các đối tượng như SqlConnection (kết nối), SqlCommand (thực thi truy vấn), và SqlDataAdapter (tải dữ liệu vào DataTable/DataSet) được sử dụng để tương tác với MS SQL Server.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Phân tích yêu cầu chức năng

Hệ thống cần đảm bảo các chức năng chính (CRUD): Thêm học sinh(cho phép nhập thông tin học sinh mới và lưu vào CSDL). Kiểm tra ràng buộc dữ liệu (không được để trống các trường bắt buộc). Sửa thông tin (cho phép chọn một học sinh từ danh sách và cập nhật thông tin). Xóa học sinh (cho phép xóa hồ sơ học sinh khỏi hệ thống, có cảnh báo xác nhận). Hiển thị danh sách (load toàn bộ dữ liệu từ SQL sever lên bảng (Treeview) khi khởi động ứng dụng). Thoát (đóng chương trình an toàn).

## 3.2. Thiết kế Cơ sở dữ liệu

Cơ sở dữ liệu qlhocsinh gồm bảng chính là hocsinh với cấu trúc như sau:

A computer screen shot of a computer

AI-generated content may be incorrect.

Hình . Sơ đồ quan hệ thực thể của hệ thống

### Khởi tạo database:

CREATE DATABASE qlhs COLLATE Vietnamese\_CI\_AS;

### Code tạo các bảng:

-- =============================================

-- PHẦN 1: TẠO CÁC BẢNG (DDL CHO T-SQL)

-- =============================================

-- 📚 BẢNG MÔN HỌC

CREATE TABLE monhoc (

id INT NOT NULL IDENTITY(1,1),

tenmonhoc NVARCHAR(100) UNIQUE NOT NULL,

PRIMARY KEY (id)

);

GO

-- 🧑‍🏫 BẢNG GIÁO VIÊN

CREATE TABLE giaovien (

magv INT NOT NULL IDENTITY(1,1),

hoten NVARCHAR(100),

sodt VARCHAR(15),

chuyenmon NVARCHAR(50),

chunhiemlop NVARCHAR(20),

monhoc\_id INT,

PRIMARY KEY (magv),

FOREIGN KEY (monhoc\_id) REFERENCES monhoc(id)

);

GO

-- 🏫 BẢNG LỚP HỌC

CREATE TABLE lop (

id INT NOT NULL IDENTITY(1,1),

tenlop NVARCHAR(20) UNIQUE NOT NULL,

magv\_chunhiem INT,

PRIMARY KEY (id),

FOREIGN KEY (magv\_chunhiem) REFERENCES giaovien(magv) ON DELETE SET NULL

);

GO

-- 👨‍🎓 BẢNG HỌC SINH

CREATE TABLE hocsinh (

mahs INT NOT NULL IDENTITY(1,1),

holot NVARCHAR(50),

ten NVARCHAR(50),

lop\_id INT,

gioitinh NVARCHAR(10),

ngaysinh DATE,

chucvu NVARCHAR(50),

hanhkiem NVARCHAR(20),

hocluc NVARCHAR(20),

tbcanam FLOAT,

diachi NVARCHAR(255),

dienthoai VARCHAR(20),

PRIMARY KEY (mahs),

FOREIGN KEY (lop\_id) REFERENCES lop(id)

);

GO

-- 👨‍👩‍👧 BẢNG PHỤ HUYNH

CREATE TABLE phuhuynh (

id INT NOT NULL IDENTITY(1,1),

mahs INT NOT NULL,

tenphu NVARCHAR(100) NOT NULL,

sdt VARCHAR(20),

diachi NVARCHAR(255),

quanhe NVARCHAR(50),

nghenghiep NVARCHAR(100),

PRIMARY KEY (id),

FOREIGN KEY (mahs) REFERENCES hocsinh(mahs) ON DELETE CASCADE

);

GO

-- 📚 BẢNG ĐIỂM (ĐÃ SỬA: Thêm cột hocky)

CREATE TABLE diem (

    id INT NOT NULL IDENTITY(1,1),

    mahs INT NOT NULL,

    monhoc\_id INT NOT NULL,

hocky TINYINT NOT NULL DEFAULT 1, -- 1: Học kỳ 1, 2: Học kỳ 2

    mieng FLOAT,

    kt\_15p FLOAT,

    kt\_15p\_lan2 FLOAT,

    giua\_ky FLOAT,

    cuoi\_ky FLOAT,

    dtb\_mon FLOAT,

    PRIMARY KEY (id),

    FOREIGN KEY (mahs) REFERENCES hocsinh(mahs) ON DELETE CASCADE,

    FOREIGN KEY (monhoc\_id) REFERENCES monhoc(id) ON DELETE CASCADE,

-- >> RÀNG BUỘC UNIQUE MỚI <<

    CONSTRAINT uk\_diem\_hs\_mh\_hk UNIQUE (mahs, monhoc\_id, hocky) -- Chỉ 1 bản ghi cho mỗi HS, Môn học, Học kỳ

);

GO

-- 📅 BẢNG THỜI KHÓA BIỂU

CREATE TABLE thoikhoabieu (

id INT NOT NULL IDENTITY(1,1),

lop\_id INT NOT NULL,

thu NVARCHAR(20) NOT NULL,

tiet INT NOT NULL,

monhoc\_id INT NOT NULL,

magv INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (lop\_id) REFERENCES lop(id) ON DELETE CASCADE,

FOREIGN KEY (monhoc\_id) REFERENCES monhoc(id) ON DELETE CASCADE,

FOREIGN KEY (magv) REFERENCES giaovien(magv) ON DELETE CASCADE,

CONSTRAINT uk\_tkb\_lop\_thu\_tiet UNIQUE (lop\_id, thu, tiet)

);

GO

-- 🔐 BẢNG TÀI KHOẢN NGƯỜI DÙNG

CREATE TABLE users (

id INT NOT NULL IDENTITY(1,1),

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

fullname NVARCHAR(150),

role VARCHAR(20) NOT NULL,

PRIMARY KEY (id)

);

GO

**3.3. Thiết kế Giao diện và Trải nghiệm người dùng (UX) 🎨**

Giao diện phần mềm được thiết kế theo tư duy "Context-Aware" (Nhận biết ngữ cảnh). Thay vì sử dụng một thiết kế cứng nhắc, các Form và thành phần điều khiển sẽ tự động biến đổi linh hoạt dựa trên quyền hạn (Role) của người dùng đang đăng nhập (Admin, Giáo viên, Học sinh).

**3.3.1. Nguyên tắc thiết kế giao diện động (Role-Based UI) 🛡️**

Nguyên tắc này đảm bảo người dùng chỉ tương tác với các chức năng mà họ được ủy quyền, từ đó giảm thiểu thao tác sai và giữ cho giao diện gọn gàng, tập trung.

* Thanh điều hướng (Side Panel/Menu): Thanh điều hướng không hiển thị cố định mà được quản lý logic trong Form Trangchu.cs (sử dụng hàm PhanQuyenTheoVaiTro). Người dùng chỉ nhìn thấy những nhóm chức năng mà họ được phép truy cập.
* Các nút tác vụ nghiệp vụ (Action Buttons): Các nút Thêm, Sửa, Xóa, Lưu chỉ được kích hoạt (Enabled = true) hoặc hiển thị (Visible = true) khi hệ thống xác nhận người dùng là Quản trị viên (Admin) hoặc Giáo viên (ví dụ, trong các Form quản lý như frmHS.cs, frmQuanLyGiaoVien.cs). Đối với tài khoản Học sinh, các nút này bị vô hiệu hóa hoặc ẩn đi để đảm bảo giao diện chỉ phục vụ mục đích tra cứu thông tin.

Giá trị hướng dẫn: Minh họa sự thay đổi của giao diện ngay từ màn hình chính.

**3.3.2. Bố cục Master-Detail và Bảng điểm trực quan**

Để tối ưu hóa trải nghiệm tương tác với dữ liệu, hệ thống áp dụng các mô hình bố cục hiệu quả:

1. **Bố cục Master-Detail (Master-Detail Layout):**
   * Áp dụng trong các Form quản lý hồ sơ (frmHS.cs, frmQuanLyGiaoVien.cs).
   * Master (Chính): Danh sách tổng quan hiển thị trên điều khiển DataGridView giúp tra cứu nhanh chóng (ví dụ: Danh sách Học sinh).
   * Detail (Chi tiết): Khi chọn một dòng dữ liệu trên Master, thông tin chi tiết sẽ được tải và hiển thị ngay ở các điều khiển nhập liệu (TextBox, ComboBox) phía dưới.
   * Lợi ích: Người dùng có thể chỉnh sửa dữ liệu mà không bị mất ngữ cảnh làm việc (luôn thấy danh sách tổng thể).

Giá trị hướng dẫn: Minh họa bố cục cơ bản của một Form quản lý hồ sơ trên WinForms.

1. **Bảng điểm trực quan (Pivot Table Logic):**
   * Áp dụng cho giao diện xem điểm tổng hợp (frmthongtincanhan.cs).
   * Thay vì hiển thị điểm mỗi môn trên một dòng (như trong CSDL), phần mềm sử dụng logic Pivot Table (thông qua truy vấn SQL phức tạp hoặc ADO.NET DataTable).
   * Các thông số Điểm TB Học kỳ 1, Điểm TB Học kỳ 2 và ĐTB Cả Năm được trải ngang thành các cột tiêu đề.
   * Lợi ích: Cung cấp cho người dùng (Giáo viên, Học sinh) một cái nhìn tổng quát về học lực ở tất cả các môn trên cùng một dòng, tránh việc phải chuyển đổi qua lại giữa các màn hình khác nhau để so sánh kết quả học tập giữa các kỳ.

# CHƯƠNG 4. CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM

## 4.1. Cài đặt Môi trường và Công nghệ

Việc cài đặt và cấu hình môi trường tập trung vào các công nghệ của Microsoft:

* Môi trường Phát triển (IDE): Visual Studio.
* Ngôn ngữ Lập trình: C#.
* Kết nối Cơ sở dữ liệu (DAL): Sử dụng đối tượng SqlConnection và các lớp ADO.NET (SqlCommand, SqlDataAdapter) từ thư viện System.Data.SqlClient để đảm bảo giao tiếp ổn định với MS SQL Server.
* Quản lý Kết nối: Lớp KetNoi.cs được thiết kế dưới dạng private static và cung cấp hàm public static SqlConnection GetConnection(), giúp các Form có thể tạo và mở kết nối một cách tập trung, dễ quản lý (tham khảo KetNoi.cs).

## 4.2. Các Chức năng Chính và Kết quả Cài đặt

Hệ thống được tổ chức theo kiến trúc 3 lớp, trong đó mỗi Form chịu trách nhiệm xử lý logic trình bày và tương tác dữ liệu thông qua lớp KetNoi..

## 4.2.1. Màn hình Chính (Main Form) và Cơ chế Phân quyền (RBAC)

Form Trangchu.cs (Trang Chủ) đóng vai trò là Dashboard và bộ điều hướng chính. Cơ chế phân quyền được thực hiện thông qua hàm PhanQuyenTheoVaiTro() dựa trên biến currentUserRole được lấy từ frmDangNhap.cs.

* Logic Phân quyền:
  + Admin: Toàn quyền truy cập (hiện tất cả các nút: QL Học sinh, QL Giáo viên, QL Điểm, TKB).
  + Giáo viên: Bị ẩn nút "Quản lý Giáo viên".
  + Học sinh: Chỉ thấy các nút tra cứu cá nhân (QL Học sinh/Thông tin cá nhân, TKB).
* Logic Định tuyến: Khi click vào các nút chức năng (ví dụ: btnQuanlyhocsinh\_Click), hệ thống kiểm tra vai trò người dùng và mở Form tương ứng, đồng thời truyền ID thực tế (currentUserID - là mahs cho học sinh hoặc magv cho giáo viên) vào Constructor của Form đích.

## 4.2.1. Màn hình Chính (Main Form) và Cơ chế Phân quyền (RBAC)

Form Trangchu.cs (Trang Chủ) đóng vai trò là Dashboard và bộ điều hướng chính. Cơ chế phân quyền được thực hiện thông qua hàm PhanQuyenTheoVaiTro() dựa trên biến currentUserRole được lấy từ frmDangNhap.cs.

* Logic Phân quyền:
  + Admin: Toàn quyền truy cập (hiện tất cả các nút: QL Học sinh, QL Giáo viên, QL Điểm, TKB).
  + Giáo viên: Bị ẩn nút "Quản lý Giáo viên".
  + Học sinh: Chỉ thấy các nút tra cứu cá nhân (QL Học sinh/Thông tin cá nhân, TKB).
* Logic Định tuyến: Khi click vào các nút chức năng (ví dụ: btnQuanlyhocsinh\_Click), hệ thống kiểm tra vai trò người dùng và mở Form tương ứng, đồng thời truyền ID thực tế (currentUserID - là mahs cho học sinh hoặc magv cho giáo viên) vào Constructor của Form đích.

## 4.2.2. Chức năng Thời khóa biểu Đa ngữ cảnh (frmThoikhoabieu.cs)

Form này sử dụng một DataGridView duy nhất để hiển thị thời khóa biểu, với logic lọc dữ liệu động:

* Ngữ cảnh hóa Dữ liệu (Dựa trên Role):
  + Admin/Teacher: Lọc theo Lớp thông qua ComboBox (cboChonLop2). Giáo viên chỉ thấy các lớp mình dạy (hàm LoadComboBoxLop\_GiaoVien() sử dụng WHERE tkb.magv = @gvId).
  + Student: Tự động chỉ tải lịch của lớp mình (sử dụng Sub-query WHERE tkb.lop\_id = (SELECT lop\_id FROM hocsinh WHERE mahs = @hsId)).
* Kiểm tra Trùng lịch (Collision Check): (Chỉ cho Admin) Khi Thêm/Sửa, hệ thống kiểm tra bằng các truy vấn SELECT COUNT(\*) đồng thời để đảm bảo không có Giáo viên hoặc Lớp nào bị trùng tiết trong cùng một khung giờ (thu và tiet).

## 4.2.3. Quản lý Điểm số (frmQuanLyDiem.cs và frmbangdiemcanhan.cs)

Phân hệ này được tách thành hai Form để tối ưu hóa việc nhập liệu (Giáo viên) và xem báo cáo tổng hợp (Admin).

a) Bảng Điểm Tổng hợp (Admin/Teacher View - frmQuanLyDiem.cs):

* Cơ chế Pivot Table (T-SQL): Form sử dụng truy vấn GROUP BY kết hợp với hàm tổng hợp MAX(CASE WHEN...) để chuyển đổi dữ liệu điểm từ dạng dòng thành dạng cột ngang (Môn học), giúp Admin có cái nhìn tổng quan về ĐTB của cả lớp.
* Thống kê & Xếp loại: Hàm ThongKe() tính toán Sĩ số, Điểm TB Chung, và phân loại số lượng học sinh Giỏi, Khá, TB, Yếu ngay trên tầng ứng dụng (C#), hiển thị kết quả trực tiếp trên txtThongKe.

b) Bảng Điểm Chi tiết (Student View - frmthongtincanhan.cs):

* Tra cứu đa Học kỳ: Form này cho phép Học sinh (hoặc Admin/Teacher xem chi tiết) chuyển đổi giữa Học kỳ 1, Học kỳ 2, và Cả năm thông qua cboHocKy.
* Logic ĐTB Cả Năm (T-SQL): Khi chọn "Cả năm" (ID=3), hệ thống thực hiện truy vấn phức tạp để tính điểm trung bình của môn học qua hai kỳ (theo công thức hệ số) và hiển thị kết quả trên DataGridView.

## 4.2.4. Quản lý Học sinh (frmHS.cs) và Tính toàn vẹn dữ liệu

Form quản lý hồ sơ học sinh sử dụng mô hình Master-Detail và tập trung vào tính toàn vẹn dữ liệu:

* Logic Lọc Linh hoạt: Hàm LoadData() cho phép lọc đồng thời theo ID Lớp đang chọn hoặc theo từ khóa tìm kiếm (keyword là mahs).
* Giao dịch Dữ liệu (Transaction): Chức năng Thêm và Sửa (hàm btnLuu\_Click) sử dụng đối tượng SqlTransaction (conn.BeginTransaction()) để xử lý các thao tác ghi lên nhiều bảng (hocsinh, phuhuynh):
  + Nếu thao tác thành công trên tất cả các bảng, lệnh trans.Commit() được thực thi.
  + Nếu có bất kỳ lỗi nào xảy ra (ví dụ: lỗi SQL khi thêm Phụ huynh), lệnh trans.Rollback() được gọi, đảm bảo rằng dữ liệu Học sinh vừa thêm cũng bị hủy bỏ, tránh dữ liệu mồ côi.

## 4.2.5. Quản lý Giáo viên (frmQuanLyGiaoVien.cs) và Ràng buộc Khóa ngoại

* Kiểm tra Ràng buộc Khóa ngoại: Chức năng Xóa Giáo viên được bảo vệ nghiêm ngặt. Mặc dù bạn đã thiết lập khóa ngoại ở mức Database, đoạn mã C# trong btnXoa\_Click được đặt trong try-catch để bắt lỗi SqlException nếu Giáo viên đang là Chủ nhiệm lớp hoặc có lịch dạy, và hiển thị cảnh báo thân thiện thay vì để chương trình gặp lỗi ngắt quãng.
* Tìm kiếm chính xác: Chức năng tìm kiếm trong LoadData() được giới hạn chỉ tìm theo Mã Giáo viên (dạng số) để đảm bảo kết quả chính xác, loại bỏ tìm kiếm theo tên không rõ ràng.

## 4.2.6. Các from quản lý giao diện

A computer screen shot of a school login

AI-generated content may be incorrect.

Hình . Giao diện đăng nhập và điều hướng hệ thống

A building with a flag on top

AI-generated content may be incorrect.Vai trò Admin:

Hình . Giao diện sau khi đăng nhập tài khoản admin



Hình . Giao diện sau khi nhấn vào quản lý của tài khoản admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn xem thông tin chi tiết học sinh của tài khoản admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản lý giáo viên của tài khoản admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản lý điểm của tài khoản admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn xem thông tin chi tiết điểm học sinh của tài khoản admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản thời khóa biểu của tài khoản admin

Vai trò giáo viên:

A computer screen shot of a building

AI-generated content may be incorrect.

Hình . Giao diện sau khi đăng nhập tài khoản giáo viên

A screenshot of a computer

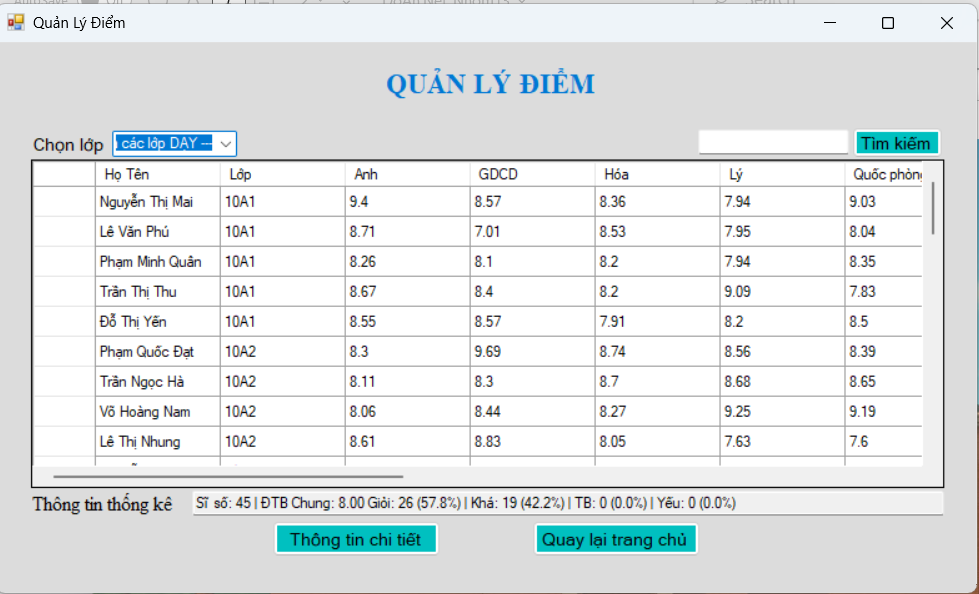
AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn vào quản lý của tài khoản giáo viên

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn xem thông tin chi tiết học sinh của tài khoản giáo viên



Hình . Giao diện sau khi nhấn quản lý điểm của tài khoản giáo viên

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn xem thông tin chi tiết điểm học sinh của tài khoản giáo viên

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản thời khóa biểu của tài khoản giáo viên

### Vai trò học sinh:

A building with a flag on top

AI-generated content may be incorrect.

Hình . Giao diện sau khi đăng nhập tài khoản học sinh

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản lý học sinh của tài khoản học sinh

A screenshot of a computer

AI-generated content may be incorrect.

Hình . Giao diện sau khi nhấn quản thời khóa biểu của tài khoản học sinh

## 4.2.6. Các code quản lý

### Code kết nối:

using System;

using System.Collections.Generic;

using System.Data.SqlClient; // <-- BẠN PHẢI THÊM THƯ VIỆN NÀY

using System.Linq;

using System.Text;

using System.Threading.Tasks;

### namespace Quanlyhocsinh

{

internal class KetNoi

{

// Chúng ta dùng 'private static' để nó là biến của lớp, và không ai bên ngoài thay đổi được.

private static string connectionString = @"Data Source=MSI\MSSQL;Initial Catalog=qlhs;Integrated Security=True";

// 2. TẠO MỘT HÀM (METHOD) public VÀ static

// - 'public': Để các Form khác có thể gọi nó.

// - 'static': Để ta có thể gọi thẳng từ tên lớp (KetNoi.GetConnection())

// mà không cần tạo đối tượng (new KetNoi()).

// - 'SqlConnection': Kiểu dữ liệu trả về là một đối tượng kết nối.

public static SqlConnection GetConnection()

{

// Hàm này chỉ làm một việc là tạo và trả về một đối tượng kết nối mới

// với chuỗi kết nối đã định nghĩa ở trên.

return new SqlConnection(connectionString);

}

}

}

### Code program:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frmDangNhap());

}

}

}

### Code form đăng nhập

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmDangNhap : Form

{

// Chuỗi kết nối (Giữ nguyên của bạn)

private readonly string connectionString = "Server=MSI\\MSSQL;Database=qlhs;Integrated Security=True;";

public frmDangNhap()

{

InitializeComponent();

}

// --- SỰ KIỆN 1: FORM LOAD (Khớp với dòng code bạn vừa gửi) ---

private void frmDangNhap\_Load(object sender, EventArgs e)

{

// Khi form hiện lên thì con trỏ chuột nằm ở ô Tên đăng nhập

txtTendangnhap.Select();

}

// --- SỰ KIỆN 2: NÚT ĐĂNG NHẬP ---

private void btnDangNhap\_Click(object sender, EventArgs e)

{

// 1. Kiểm tra nhập liệu (Giữ nguyên)

string tenDangNhap = txtTendangnhap.Text.Trim();

string matKhau = txtMatkhau.Text.Trim();

if (string.IsNullOrEmpty(tenDangNhap) || string.IsNullOrEmpty(matKhau))

{

MessageBox.Show("Vui lòng nhập đầy đủ thông tin!");

return;

}

using (SqlConnection conn = new SqlConnection(connectionString))

{

try

{

conn.Open();

// 2. Lấy thông tin cơ bản từ bảng USERS

string sqlCheck = "SELECT id, role, fullname FROM dbo.users WHERE username = @User AND password = @Pass";

string userRole = "";

string userFullname = "";

string realID = "";

using (SqlCommand cmd = new SqlCommand(sqlCheck, conn))

{

cmd.Parameters.AddWithValue("@User", tenDangNhap);

cmd.Parameters.AddWithValue("@Pass", matKhau);

using (SqlDataReader reader = cmd.ExecuteReader())

{

if (reader.Read())

{

userRole = reader["role"].ToString().Trim();

userFullname = reader["fullname"].ToString();

realID = reader["id"].ToString();

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu!");

return;

}

}

}

// 3. XỬ LÝ PHÂN BIỆT: GIÁO VIÊN DÙNG 'HOTEN' - HỌC SINH DÙNG 'HOLOT + TEN'

if (userRole == "teacher")

{

// Bảng GIAOVIEN dùng cột 'hoten'

string sqlGetGV = "SELECT magv FROM giaovien WHERE hoten = @TenGV";

using (SqlCommand cmdGV = new SqlCommand(sqlGetGV, conn))

{

cmdGV.Parameters.AddWithValue("@TenGV", userFullname);

object result = cmdGV.ExecuteScalar();

if (result != null) realID = result.ToString();

}

}

else if (userRole == "student")

{

// Bảng HOCSINH dùng 'holot' + 'ten'

string sqlGetHS = "SELECT mahs FROM hocsinh WHERE (holot + ' ' + ten) = @TenHS";

using (SqlCommand cmdHS = new SqlCommand(sqlGetHS, conn))

{

cmdHS.Parameters.AddWithValue("@TenHS", userFullname);

object result = cmdHS.ExecuteScalar();

if (result != null) realID = result.ToString();

}

}

// 4. Chuyển Form

this.Hide();

Trangchu frmMain = new Trangchu(userRole, realID, userFullname);

frmMain.Show();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message);

}

}

}

private void btnThoat\_Click\_1(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Bạn có chắc chắn muốn thoát khỏi ứng dụng?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

Application.Exit();

}

}

}

}

### Code from trang chủ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class Trangchu : Form

{

// 1. Khai báo 3 biến cấp Class

private string currentUserRole;

private string currentUserID;

private string currentUserFullname; // BIẾN MỚI ĐỂ LƯU TÊN

// Constructor Mặc định

public Trangchu()

{

InitializeComponent();

}

// 2. CONSTRUCTOR CHÍNH (NHẬN 3 THAM SỐ)

public Trangchu(string role, string userId, string fullname)

{

InitializeComponent();

// Lưu 3 giá trị lại

this.currentUserRole = role.Trim().ToLower();

this.currentUserID = userId;

this.currentUserFullname = fullname;

PhanQuyenTheoVaiTro();

}

// 3. HÀM PHÂN QUYỀN (Hiển thị tên)

private void PhanQuyenTheoVaiTro()

{

// SỬA: Hiển thị tên đầy đủ

this.Text = $"Trang Chủ (Chào mừng: {this.currentUserFullname})";

try

{

// Logic ẩn/hiện nút

btnQuanlyhocsinh.Visible = false;

btnQuanlygiaovien.Visible = false;

btnTrangchu.Visible = false;

btnQuanlydiem.Visible = false;

btnDangxuat.Visible = false;

btnTKB.Visible = false;

if (this.currentUserRole == "admin")

{

btnQuanlyhocsinh.Visible = true;

btnQuanlygiaovien.Visible = true;

btnTrangchu.Visible = true;

btnQuanlydiem.Visible = true;

btnDangxuat.Visible = true;

btnTKB.Visible = true;

}

else if (this.currentUserRole == "teacher")

{

btnQuanlyhocsinh.Visible = true;

btnQuanlygiaovien.Visible = false;

btnTrangchu.Visible = true;

btnQuanlydiem.Visible = true;

btnDangxuat.Visible = true;

btnTKB.Visible = true;

}

else if (this.currentUserRole == "student")

{

btnTrangchu.Visible = true;

btnQuanlyhocsinh.Visible = true;

btnQuanlydiem.Visible = false;

btnTKB.Visible = true;

btnDangxuat.Visible = true;

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi phân quyền nút: " + ex.Message);

}

}

// --- CÁC SỰ KIỆN CLICK (ĐÃ SỬA CHO ADMIN) ---

private void btnQuanlyhocsinh\_Click(object sender, EventArgs e)

{

string userIdString = this.currentUserID;

if (this.currentUserRole == "student")

{

if (int.TryParse(userIdString, out int studentId))

{

frmthongtincanhan form = new frmthongtincanhan(studentId);

form.Show();

}

else

{

MessageBox.Show("Lỗi: Mã học sinh không hợp lệ.", "Lỗi dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else if (this.currentUserRole == "teacher")

{

if (int.TryParse(userIdString, out int teacherId))

{

frmHS frmquanlyhocsinh = new frmHS(teacherId);

frmquanlyhocsinh.ShowDialog();

}

}

else if (this.currentUserRole == "admin")

{

// Admin mở Form quản lý chung

frmHS frmquanlyhocsinh = new frmHS();

frmquanlyhocsinh.ShowDialog();

}

}

private void btnQuanlydiem\_Click\_1(object sender, EventArgs e)

{

if (this.currentUserRole == "teacher" && int.TryParse(this.currentUserID, out int teacherId))

{

frmQuanLyDiem frmQuanLyDiem = new frmQuanLyDiem(teacherId);

frmQuanLyDiem.ShowDialog();

}

else if (this.currentUserRole == "admin")

{

// Admin mở Form quản lý chung

frmQuanLyDiem frmQuanLyDiem = new frmQuanLyDiem();

frmQuanLyDiem.ShowDialog();

}

}

private void btnTKB\_Click\_1(object sender, EventArgs e)

{

if (int.TryParse(this.currentUserID, out int userId))

{

frmThoikhoabieu frmThoikhoabieu = new frmThoikhoabieu(userId, this.currentUserRole);

frmThoikhoabieu.ShowDialog();

}

else

{

MessageBox.Show("Không thể lấy ID người dùng để tải Thời khóa biểu.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnQuanlygiaovien\_Click\_1(object sender, EventArgs e)

{

frmQuanLyGiaoVien frmQuanLyGiaoVien = new frmQuanLyGiaoVien();

frmQuanLyGiaoVien.ShowDialog();

}

private void btnDangxuat\_Click\_1(object sender, EventArgs e)

{

this.Hide();

frmDangNhap frmDangNhap = new frmDangNhap();

frmDangNhap.Show();

}

// --- CÁC SỰ KIỆN PHỤ (ĐỂ TRỐNG) ---

private void Trangchu\_Load\_1(object sender, EventArgs e) { }

private void btnTrangchu\_Click\_1(object sender, EventArgs e) { }

private void panel1\_Paint(object sender, PaintEventArgs e) { }

}

}

### Code form quản lý học sinh

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient; // Thư viện kết nối SQL

namespace Quanlyhocsinh

{

public partial class frmHS : Form

{

// Cờ trạng thái

private bool isAdding = false;

private bool isLoaded = false; // Biến này để chặn sự kiện combobox chạy khi form chưa load xong

private int currentTeacherId = 0; // 0 là Admin

public frmHS()

{

InitializeComponent();

}

// Nếu bạn có constructor nhận quyền giáo viên

public frmHS(int teacherId)

{

InitializeComponent();

this.currentTeacherId = teacherId;

}

// ==========================================

// 1. FORM LOAD

// ==========================================

private void frmHS\_Load(object sender, EventArgs e)

{

isLoaded = false; // Tạm khóa

LoadLopComboBoxes();

LoadData(); // Load toàn bộ danh sách ban đầu

SetButtonState(true);

isLoaded = true; // Mở khóa

}

// ==========================================

// 2. HÀM TẢI DỮ LIỆU (QUAN TRỌNG NHẤT)

// ==========================================

private void LoadData(string keyword = null, int? lopId = null)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

// SỬA SQL: JOIN BẢNG LỚP ĐỂ LẤY TÊN LỚP

string sqlQuery = @"

SELECT

hs.mahs,

hs.holot,

hs.ten,

l.tenlop, -- Lấy tên lớp hiển thị

hs.gioitinh,

hs.ngaysinh,

hs.chucvu,

hs.hanhkiem,

hs.lop\_id, -- Lấy ID để xử lý ngầm

hs.diachi

FROM HocSinh hs

LEFT JOIN Lop l ON hs.lop\_id = l.id

WHERE 1=1";

SqlCommand cmd = new SqlCommand(sqlQuery, conn);

// --- ƯU TIÊN 1: TÌM KIẾM THEO MÃ HS ---

if (!string.IsNullOrEmpty(keyword))

{

if (int.TryParse(keyword, out int maHSTimKiem))

{

sqlQuery += " AND hs.mahs = @mahs";

cmd.Parameters.AddWithValue("@mahs", maHSTimKiem);

}

}

// --- ƯU TIÊN 2: LỌC THEO LỚP (Chỉ khi không tìm kiếm) ---

else if (lopId.HasValue && lopId.Value > 0)

{

sqlQuery += " AND hs.lop\_id = @lopId";

cmd.Parameters.AddWithValue("@lopId", lopId.Value);

}

// --- PHÂN QUYỀN GIÁO VIÊN (Nếu có) ---

if (currentTeacherId > 0)

{

// Chỉ hiện học sinh thuộc lớp giáo viên chủ nhiệm (ví dụ)

// Tùy logic của bạn, có thể bỏ qua nếu không cần

sqlQuery += " AND hs.lop\_id IN (SELECT id FROM lop WHERE magv\_chunhiem = @gvId)";

cmd.Parameters.AddWithValue("@gvId", currentTeacherId);

}

cmd.CommandText = sqlQuery;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvHocSinh.DataSource = dt;

// --- ĐỊNH DẠNG CỘT HIỂN THỊ ---

if (dgvHocSinh.Columns.Contains("lop\_id")) dgvHocSinh.Columns["lop\_id"].Visible = false; // Ẩn ID

if (dgvHocSinh.Columns.Contains("diachi")) dgvHocSinh.Columns["diachi"].Visible = false; // Ẩn Địa chỉ cho gọn

dgvHocSinh.Columns["mahs"].HeaderText = "Mã HS";

dgvHocSinh.Columns["holot"].HeaderText = "Họ Lót";

dgvHocSinh.Columns["ten"].HeaderText = "Tên";

if (dgvHocSinh.Columns.Contains("tenlop"))

{

dgvHocSinh.Columns["tenlop"].HeaderText = "Lớp";

dgvHocSinh.Columns["tenlop"].DisplayIndex = 3; // Đưa cột Lớp ra sau Tên

}

dgvHocSinh.Columns["gioitinh"].HeaderText = "Giới Tính";

dgvHocSinh.Columns["ngaysinh"].HeaderText = "Ngày Sinh";

dgvHocSinh.Columns["chucvu"].HeaderText = "Chức Vụ";

dgvHocSinh.Columns["hanhkiem"].HeaderText = "Hạnh Kiểm";

dgvHocSinh.ClearSelection();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải dữ liệu: " + ex.Message);

}

}

}

// ==========================================

// 3. TẢI COMBOBOX LỚP

// ==========================================

private void LoadLopComboBoxes()

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "SELECT id, tenlop FROM lop ORDER BY tenlop";

SqlDataAdapter da = new SqlDataAdapter(sql, conn);

DataTable dt = new DataTable();

da.Fill(dt);

// 1. Setup cho ComboBox LỌC (Có thêm dòng "Tất cả")

DataTable dtFilter = dt.Copy();

DataRow row = dtFilter.NewRow();

row["id"] = 0;

row["tenlop"] = "--- Tất cả các lớp ---";

dtFilter.Rows.InsertAt(row, 0);

cboChonlop.DataSource = dtFilter;

cboChonlop.DisplayMember = "tenlop";

cboChonlop.ValueMember = "id";

// 2. Setup cho ComboBox NHẬP LIỆU (Chỉ danh sách lớp thật)

cboLop.DataSource = dt;

cboLop.DisplayMember = "tenlop";

cboLop.ValueMember = "id";

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải lớp: " + ex.Message);

}

}

}

// ==========================================

// 4. SỰ KIỆN LỌC KHI CHỌN COMBOBOX

// ==========================================

private void cboChonlop\_SelectedIndexChanged(object sender, EventArgs e)

{

// 1. Chặn sự kiện khi Form chưa tải xong để tránh lỗi

if (!isLoaded) return;

// 2. XÓA NỘI DUNG Ô TÌM KIẾM (Theo yêu cầu của bạn)

// Để người dùng biết là đang lọc theo danh sách lớp chứ không phải tìm kiếm cũ

txtTimKiem.Text = "";

// 3. Lấy ID của lớp đang chọn

int selectedId = 0;

if (cboChonlop.SelectedValue != null)

{

int.TryParse(cboChonlop.SelectedValue.ToString(), out selectedId);

}

// 4. Gọi hàm LoadData

// - Tham số keyword: truyền null (vì đã xóa tìm kiếm)

// - Tham số lopId: truyền ID lớp vừa chọn

LoadData(null, selectedId);

}

// ==========================================

// 5. NÚT TÌM KIẾM (CHỈ TÌM MÃ HS)

// ==========================================

private void btnTimKiem\_Click(object sender, EventArgs e)

{

string keyword = txtTimKiem.Text.Trim();

// =================================================================

// TRƯỜNG HỢP 1: Ô TÌM KIẾM RỖNG -> LOAD LẠI THEO LỚP ĐANG CHỌN

// =================================================================

if (string.IsNullOrEmpty(keyword))

{

// Lấy ID lớp đang chọn hiện tại

int selectedLopId = 0;

if (cboChonlop.SelectedValue != null)

{

int.TryParse(cboChonlop.SelectedValue.ToString(), out selectedLopId);

}

// Load lại danh sách theo lớp đó (coi như là nút Làm mới)

LoadData(null, selectedLopId);

return;

}

// =================================================================

// TRƯỜNG HỢP 2: CÓ NHẬP MÃ -> TÌM TRÊN TOÀN TRƯỜNG

// =================================================================

// Kiểm tra phải là số

if (!int.TryParse(keyword, out int maHS))

{

MessageBox.Show("Mã học sinh phải là số!");

txtTimKiem.Text = "";

txtTimKiem.Focus();

return;

}

// [QUAN TRỌNG] Reset ComboBox về "Tất cả" để người dùng hiểu là đang tìm toàn trường

// Phải dùng biến isLoaded chặn sự kiện để không bị load 2 lần

if (cboChonlop.Items.Count > 0) // Kiểm tra an toàn

{

isLoaded = false; // Khóa sự kiện

cboChonlop.SelectedIndex = 0; // Chuyển về "Tất cả các lớp"

isLoaded = true; // Mở khóa

}

// Gọi hàm tìm kiếm (Truyền null vào tham số lớp để tìm tất cả)

LoadData(keyword, null);

}

// ==========================================

// 6. HIỂN THỊ CHI TIẾT KHI CLICK BẢNG

// ==========================================

private void dgvHocSinh\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0 && btnThem.Enabled == true) // Chỉ cho xem khi không ở chế độ thêm/sửa

{

try

{

DataGridViewRow row = dgvHocSinh.Rows[e.RowIndex];

// Lấy Mã HS từ cột ẩn hoặc cột hiện

int mahs = Convert.ToInt32(row.Cells["mahs"].Value);

LoadChiTietHocSinh(mahs);

}

catch { }

}

}

private void LoadChiTietHocSinh(int mahs)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

// 1. Load thông tin HS

string sqlHS = "SELECT \* FROM hocsinh WHERE mahs = @mahs";

SqlCommand cmdHS = new SqlCommand(sqlHS, conn);

cmdHS.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader rd = cmdHS.ExecuteReader();

if (rd.Read())

{

txtMaHS.Text = rd["mahs"].ToString();

txtHolot.Text = rd["holot"].ToString();

txtTen.Text = rd["ten"].ToString();

dtpNgaySinh.Value = Convert.ToDateTime(rd["ngaysinh"]);

txtDiaChi.Text = rd["diachi"].ToString();

cboGioiTinh.Text = rd["gioitinh"].ToString();

txtChucVu.Text = rd["chucvu"].ToString();

cboLop.SelectedValue = rd["lop\_id"];

}

rd.Close();

// 2. Load thông tin Phụ huynh

ClearParentFields();

string sqlPH = "SELECT \* FROM phuhuynh WHERE mahs = @mahs";

SqlCommand cmdPH = new SqlCommand(sqlPH, conn);

cmdPH.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader rdPH = cmdPH.ExecuteReader();

while (rdPH.Read())

{

string quanhe = rdPH["quanhe"].ToString();

if (quanhe == "Cha")

{

txtTencha.Text = rdPH["tenphu"].ToString();

txtNghecha.Text = rdPH["nghenghiep"].ToString();

txtSDTcha.Text = rdPH["sdt"].ToString();

txtDiaChicha.Text = rdPH["diachi"].ToString();

}

else if (quanhe == "Mẹ" || quanhe == "Me")

{

txtTenme.Text = rdPH["tenphu"].ToString();

txtNgheme.Text = rdPH["nghenghiep"].ToString();

txtSDTme.Text = rdPH["sdt"].ToString();

txtDiaChime.Text = rdPH["diachi"].ToString();

}

}

rdPH.Close();

}

}

// ==========================================

// 7. CÁC NÚT THÊM - SỬA - XÓA - LƯU

// ==========================================

private void btnThem\_Click(object sender, EventArgs e)

{

isAdding = true;

SetButtonState(false);

ClearAllFields();

txtMaHS.Text = "[Tự động]";

txtHolot.Focus();

}

private void btnSua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaHS.Text) || txtMaHS.Text == "[Tự động]")

{

MessageBox.Show("Vui lòng chọn học sinh cần sửa!");

return;

}

isAdding = false;

SetButtonState(false);

txtHolot.Focus();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaHS.Text) || txtMaHS.Text == "[Tự động]") return;

if (MessageBox.Show("Bạn có chắc muốn xóa học sinh này?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

// Xóa phụ huynh trước (nếu có khóa ngoại) hoặc xóa HS trực tiếp nếu set Cascade

SqlCommand cmd = new SqlCommand("DELETE FROM hocsinh WHERE mahs = @mahs", conn);

cmd.Parameters.AddWithValue("@mahs", Convert.ToInt32(txtMaHS.Text));

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!");

ClearAllFields();

LoadData(); // Load lại

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa: " + ex.Message);

}

}

}

}

private void btnLuu\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtTen.Text) || cboLop.SelectedValue == null)

{

MessageBox.Show("Tên và Lớp không được để trống!");

return;

}

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

SqlTransaction trans = conn.BeginTransaction();

try

{

int currentMaHS = 0;

// 1. XỬ LÝ BẢNG HOCSINH

if (isAdding)

{

string sql = @"INSERT INTO hocsinh (holot, ten, lop\_id, gioitinh, ngaysinh, chucvu, diachi)

VALUES (@holot, @ten, @lopId, @gioitinh, @ngaysinh, @chucvu, @diachi);

SELECT SCOPE\_IDENTITY();";

SqlCommand cmd = new SqlCommand(sql, conn, trans);

cmd.Parameters.AddWithValue("@holot", txtHolot.Text);

cmd.Parameters.AddWithValue("@ten", txtTen.Text);

cmd.Parameters.AddWithValue("@lopId", cboLop.SelectedValue);

cmd.Parameters.AddWithValue("@gioitinh", cboGioiTinh.Text);

cmd.Parameters.AddWithValue("@ngaysinh", dtpNgaySinh.Value);

cmd.Parameters.AddWithValue("@chucvu", txtChucVu.Text);

cmd.Parameters.AddWithValue("@diachi", txtDiaChi.Text);

currentMaHS = Convert.ToInt32(cmd.ExecuteScalar()); // Lấy ID vừa thêm

}

else // Sửa

{

currentMaHS = Convert.ToInt32(txtMaHS.Text);

string sql = @"UPDATE hocsinh SET holot=@holot, ten=@ten, lop\_id=@lopId,

gioitinh=@gioitinh, ngaysinh=@ngaysinh, chucvu=@chucvu, diachi=@diachi

WHERE mahs=@mahs";

SqlCommand cmd = new SqlCommand(sql, conn, trans);

cmd.Parameters.AddWithValue("@mahs", currentMaHS);

cmd.Parameters.AddWithValue("@holot", txtHolot.Text);

cmd.Parameters.AddWithValue("@ten", txtTen.Text);

cmd.Parameters.AddWithValue("@lopId", cboLop.SelectedValue);

cmd.Parameters.AddWithValue("@gioitinh", cboGioiTinh.Text);

cmd.Parameters.AddWithValue("@ngaysinh", dtpNgaySinh.Value);

cmd.Parameters.AddWithValue("@chucvu", txtChucVu.Text);

cmd.Parameters.AddWithValue("@diachi", txtDiaChi.Text);

cmd.ExecuteNonQuery();

}

// 2. XỬ LÝ BẢNG PHUHUYNH (CHA)

// Logic đơn giản: Xóa cũ thêm mới hoặc Update. Ở đây dùng Update if exists.

SavePhuHuynh(conn, trans, currentMaHS, "Cha", txtTencha.Text, txtNghecha.Text, txtSDTcha.Text, txtDiaChicha.Text);

// 3. XỬ LÝ BẢNG PHUHUYNH (MẸ)

SavePhuHuynh(conn, trans, currentMaHS, "Mẹ", txtTenme.Text, txtNgheme.Text, txtSDTme.Text, txtDiaChime.Text);

trans.Commit();

MessageBox.Show("Lưu thành công!");

SetButtonState(true);

// Load lại dữ liệu và focus vào dòng vừa thêm/sửa

if (isAdding) ClearAllFields();

LoadData();

}

catch (Exception ex)

{

trans.Rollback();

MessageBox.Show("Lỗi khi lưu: " + ex.Message);

}

}

}

// Hàm phụ trợ lưu phụ huynh cho gọn

private void SavePhuHuynh(SqlConnection conn, SqlTransaction trans, int mahs, string quanhe, string ten, string nghe, string sdt, string diachi)

{

if (string.IsNullOrEmpty(ten)) return; // Không nhập tên thì thôi

string sqlCheck = "SELECT COUNT(\*) FROM phuhuynh WHERE mahs = @mahs AND quanhe = @quanhe";

SqlCommand cmdCheck = new SqlCommand(sqlCheck, conn, trans);

cmdCheck.Parameters.AddWithValue("@mahs", mahs);

cmdCheck.Parameters.AddWithValue("@quanhe", quanhe);

int exists = (int)cmdCheck.ExecuteScalar();

string sql;

if (exists > 0)

sql = "UPDATE phuhuynh SET tenphu=@ten, nghenghiep=@nghe, sdt=@sdt, diachi=@diachi WHERE mahs=@mahs AND quanhe=@quanhe";

else

sql = "INSERT INTO phuhuynh (mahs, tenphu, nghenghiep, sdt, diachi, quanhe) VALUES (@mahs, @ten, @nghe, @sdt, @diachi, @quanhe)";

SqlCommand cmd = new SqlCommand(sql, conn, trans);

cmd.Parameters.AddWithValue("@mahs", mahs);

cmd.Parameters.AddWithValue("@quanhe", quanhe);

cmd.Parameters.AddWithValue("@ten", ten);

cmd.Parameters.AddWithValue("@nghe", nghe);

cmd.Parameters.AddWithValue("@sdt", sdt);

cmd.Parameters.AddWithValue("@diachi", diachi);

cmd.ExecuteNonQuery();

}

private void btnHuy\_Click(object sender, EventArgs e)

{

isAdding = false;

SetButtonState(true);

ClearAllFields();

if (dgvHocSinh.CurrentRow != null)

{

int mahs = Convert.ToInt32(dgvHocSinh.CurrentRow.Cells["mahs"].Value);

LoadChiTietHocSinh(mahs);

}

}

private void btnBack\_Click(object sender, EventArgs e)

{

this.Close();

}

// ==========================================

// 8. CÁC HÀM TIỆN ÍCH UI

// ==========================================

private void SetButtonState(bool isViewing)

{

btnThem.Enabled = isViewing;

btnSua.Enabled = isViewing;

btnXoa.Enabled = isViewing;

btnLuu.Enabled = !isViewing;

btnHuy.Enabled = !isViewing;

btnBack.Enabled = isViewing;

// Khóa/Mở ô nhập liệu

txtHolot.ReadOnly = isViewing;

txtTen.ReadOnly = isViewing;

txtDiaChi.ReadOnly = isViewing;

txtChucVu.ReadOnly = isViewing;

dtpNgaySinh.Enabled = !isViewing;

cboLop.Enabled = !isViewing;

cboGioiTinh.Enabled = !isViewing;

// Phụ huynh

txtTencha.ReadOnly = isViewing;

txtNghecha.ReadOnly = isViewing;

txtSDTcha.ReadOnly = isViewing;

txtDiaChicha.ReadOnly = isViewing;

txtTenme.ReadOnly = isViewing;

txtNgheme.ReadOnly = isViewing;

txtSDTme.ReadOnly = isViewing;

txtDiaChime.ReadOnly = isViewing;

}

private void ClearAllFields()

{

txtMaHS.Text = "";

txtHolot.Text = "";

txtTen.Text = "";

txtDiaChi.Text = "";

txtChucVu.Text = "";

cboLop.SelectedIndex = -1;

cboGioiTinh.SelectedIndex = -1;

dtpNgaySinh.Value = DateTime.Now;

ClearParentFields();

}

private void ClearParentFields()

{

txtTencha.Text = ""; txtNghecha.Text = ""; txtSDTcha.Text = ""; txtDiaChicha.Text = "";

txtTenme.Text = ""; txtNgheme.Text = ""; txtSDTme.Text = ""; txtDiaChime.Text = "";

}

private void btnThongTin\_Click(object sender, EventArgs e)

{

// Kiểm tra xem có dòng nào đang được chọn không

if (dgvHocSinh.CurrentRow != null)

{

// 1. Lấy Mã Học Sinh từ dòng đang chọn trên lưới (DataGridView)

// Lưu ý: "mahs" là tên cột trong DataGridView (Name), không phải tiêu đề (HeaderText)

int mahs = Convert.ToInt32(dgvHocSinh.CurrentRow.Cells["mahs"].Value);

// 2. Truyền biến mahs vào trong ngoặc

frmthongtincanhan frm = new frmthongtincanhan(mahs);

frm.ShowDialog();

}

else

{

MessageBox.Show("Vui lòng chọn một học sinh trong danh sách!", "Thông báo");

}

}

}

}

### Code form thông tin cá nhân

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmthongtincanhan : Form

{

private int mahs;

public frmthongtincanhan(int mahs)

{

InitializeComponent();

this.mahs = mahs;

}

private void frmthongtincanhan\_Load(object sender, EventArgs e)

{

SetControlsReadOnly();

LoadThongTinHocSinh();

LoadThongTinPhuHuynh();

LoadHocKyComboBox();

// Gọi sự kiện chọn Học kỳ (sẽ Load Diem cho lần đầu tiên)

cboHocKy\_SelectedIndexChanged(null, null);

}

// Đặt tất cả các control ở chế độ chỉ đọc (Read-Only)

private void SetControlsReadOnly()

{

txtMaHS.ReadOnly = true;

txtHolot.ReadOnly = true;

txtTen.ReadOnly = true;

dtpNgaySinh.Enabled = false; // Dùng Enabled=false cho DateTimePicker

txtDiaChi.ReadOnly = true;

cboGioiTinh.Enabled = false; // Dùng Enabled=false cho ComboBox

txtChucVu.ReadOnly = true;

txtLop.ReadOnly = true;

txtTencha.ReadOnly = true;

txtNghecha.ReadOnly = true;

txtSDTcha.ReadOnly = true;

txtDiaChicha.ReadOnly = true;

txtTenme.ReadOnly = true;

txtNgheme.ReadOnly = true;

txtSDTme.ReadOnly = true;

txtDiaChime.ReadOnly = true;

txtTrungBinh.ReadOnly = true;

txtHocLuc.ReadOnly = true;

}

private void LoadThongTinHocSinh()

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = @"

SELECT hs.\*, l.tenlop

FROM hocsinh hs

JOIN lop l ON l.id = hs.lop\_id

WHERE hs.mahs = @mahs";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

txtMaHS.Text = rd["mahs"].ToString();

txtHolot.Text = rd["holot"].ToString();

txtTen.Text = rd["ten"].ToString();

// SỬA LỖI 1: Kiểm tra DBNull cho Ngày sinh

if (rd["ngaysinh"] != DBNull.Value)

{

dtpNgaySinh.Value = Convert.ToDateTime(rd["ngaysinh"]);

}

txtDiaChi.Text = rd["diachi"].ToString();

cboGioiTinh.Text = rd["gioitinh"].ToString();

txtChucVu.Text = rd["chucvu"].ToString();

txtLop.Text = rd["tenlop"].ToString();

}

}

}

private void LoadThongTinPhuHuynh()

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = "SELECT \* FROM phuhuynh WHERE mahs = @mahs";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader rd = cmd.ExecuteReader();

txtTencha.Clear(); txtNghecha.Clear(); txtSDTcha.Clear(); txtDiaChicha.Clear();

txtTenme.Clear(); txtNgheme.Clear(); txtSDTme.Clear(); txtDiaChime.Clear();

while (rd.Read())

{

string qh = rd["quanhe"].ToString();

if (qh == "Cha")

{

txtTencha.Text = rd["tenphu"].ToString();

txtNghecha.Text = rd["nghenghiep"].ToString();

txtSDTcha.Text = rd["sdt"].ToString();

txtDiaChicha.Text = rd["diachi"].ToString();

}

else if (qh == "Mẹ")

{

txtTenme.Text = rd["tenphu"].ToString();

txtNgheme.Text = rd["nghenghiep"].ToString();

txtSDTme.Text = rd["sdt"].ToString();

txtDiaChime.Text = rd["diachi"].ToString();

}

}

}

}

private void LoadHocKyComboBox()

{

// Tạo DataTable thủ công cho ComboBox Học kỳ

DataTable dtHocKy = new DataTable();

dtHocKy.Columns.Add("ID", typeof(int));

dtHocKy.Columns.Add("Name", typeof(string));

dtHocKy.Rows.Add(1, "Học kỳ 1");

dtHocKy.Rows.Add(2, "Học kỳ 2");

dtHocKy.Rows.Add(3, "Cả năm");

cboHocKy.DataSource = dtHocKy;

cboHocKy.DisplayMember = "Name";

cboHocKy.ValueMember = "ID";

// Mặc định chọn Cả năm để hiển thị tổng quan

cboHocKy.SelectedValue = 3;

}

private void LoadDiem(int mahs, int hocky)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "";

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

if (hocky == 3) // >> Tính điểm Cả năm (Tổng hợp)

{

sql = @"

SELECT

mh.tenmonhoc AS [Môn học],

AVG(CASE WHEN d.hocky = 1 THEN d.dtb\_mon END) AS [ĐTB HK1],

AVG(CASE WHEN d.hocky = 2 THEN d.dtb\_mon END) AS [ĐTB HK2],

-- ĐTB Cả Năm tính là AVG của tất cả dtb\_mon trong 2 HK (giả định hệ số 1:1:2 cho 3 loại điểm trung bình)

(

AVG(CASE WHEN d.hocky = 1 THEN d.dtb\_mon END) \* 1 +

AVG(CASE WHEN d.hocky = 2 THEN d.dtb\_mon END) \* 2

) / 3 AS [ĐTB Cả Năm]

FROM diem d

JOIN monhoc mh ON mh.id = d.monhoc\_id

WHERE d.mahs = @mahs AND d.dtb\_mon IS NOT NULL

GROUP BY mh.tenmonhoc

ORDER BY mh.tenmonhoc";

// Xóa các cột cũ để nạp cấu trúc mới

dgvBangDiem.Columns.Clear();

}

else // >> Chỉ xem điểm theo Học kỳ 1 hoặc 2 (Chi tiết điểm thành phần)

{

sql = @"

SELECT

mh.tenmonhoc AS [Môn học],

d.mieng AS [Miệng],

d.kt\_15p AS [KT 15p (1)],

d.kt\_15p\_lan2 AS [KT 15p (2)],

d.giua\_ky AS [Giữa kỳ],

d.cuoi\_ky AS [Cuối kỳ],

d.dtb\_mon AS [ĐTB Môn]

FROM diem d

JOIN monhoc mh ON mh.id = d.monhoc\_id

WHERE d.mahs = @mahs AND d.hocky = @hocky

ORDER BY mh.tenmonhoc";

cmd.Parameters.AddWithValue("@hocky", hocky);

// Cần xóa cột khi chuyển từ Cả năm sang HK1/HK2

dgvBangDiem.Columns.Clear();

}

cmd.CommandText = sql;

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvBangDiem.DataSource = dt;

dgvBangDiem.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

dgvBangDiem.RowHeadersVisible = false;

// Định dạng cột điểm

foreach (DataGridViewColumn col in dgvBangDiem.Columns)

{

if (hocky == 3) // Định dạng chế độ CẢ NĂM

{

if (col.Name.Contains("ĐTB"))

col.DefaultCellStyle.Format = "N2"; // 2 số thập phân

}

else // Định dạng chế độ HK1/HK2

{

if (col.Name == "ĐTB Môn")

col.DefaultCellStyle.Format = "N2";

else if (col.Name != "Môn học")

col.DefaultCellStyle.Format = "N1"; // Điểm thành phần 1 số TP

}

}

dgvBangDiem.ClearSelection();

// Tải ĐTB Cả năm và Học lực (thông tin này phải cố định)

LoadDTBVaHocLuc(conn, mahs);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tải bảng điểm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void LoadDTBVaHocLuc(SqlConnection conn, int mahs)

{

// KHÔNG MỞ CONNECTION VÌ ĐÃ CÓ CONN ĐƯỢC TRUYỀN VÀO TỪ HÀM LoadDiem

string sql = "SELECT tbcanam, hocluc FROM hocsinh WHERE mahs = @mahs";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

// Kiểm tra DBNull cho Trung bình cả năm

if (rd["tbcanam"] != DBNull.Value)

{

double dtb = Convert.ToDouble(rd["tbcanam"]);

txtTrungBinh.Text = dtb.ToString("0.00");

}

else

{

txtTrungBinh.Text = "N/A";

}

txtHocLuc.Text = rd["hocluc"] != DBNull.Value ? rd["hocluc"].ToString() : "N/A";

}

rd.Close();

}

private void panel6\_Paint(object sender, PaintEventArgs e)

{

// Có thể để trống

}

private void btnQuaylai\_Click(object sender, EventArgs e)

{

// Thêm lệnh đóng Form

this.Close();

}

private void cboHocKy\_SelectedIndexChanged(object sender, EventArgs e)

{

// Xử lý lỗi InvalidCastException khi form load

if (cboHocKy.SelectedValue == null) return;

if (int.TryParse(cboHocKy.SelectedValue.ToString(), out int selectedHocKy))

{

LoadDiem(this.mahs, selectedHocKy);

}

// Khóa các cột điểm thành phần khi LoadDiem xong

dgvBangDiem.ClearSelection();

}

}

}

### Code form quản lý giáo viên

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmQuanLyGiaoVien : Form

{

private bool isAdding = false;

public frmQuanLyGiaoVien()

{

InitializeComponent();

}

private void frmQuanLyGiaoVien\_Load(object sender, EventArgs e)

{

LoadData();

SetButtonState(true);

dgvGV.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvGV.MultiSelect = false;

this.dgvGV.CellClick += new System.Windows.Forms.DataGridViewCellEventHandler(this.dgvGV\_CellClick\_1);

}

// ==========================================

// 1. LOAD DANH SÁCH GIÁO VIÊN (SỬA LOGIC TÌM MÃ)

// ==========================================

private void LoadData(string keyword = null)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sqlQuery = @"

SELECT magv, hoten, chuyenmon, chunhiemlop, sodt

FROM giaovien

WHERE 1 = 1";

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

// --- LOGIC TÌM KIẾM THEO MÃ GV ---

if (!string.IsNullOrEmpty(keyword))

{

// Kiểm tra nếu là số thì tìm theo Mã

if (int.TryParse(keyword, out int maGVTimKiem))

{

sqlQuery += " AND magv = @maGV";

cmd.Parameters.AddWithValue("@maGV", maGVTimKiem);

}

else

{

// Nếu nhập chữ -> Không tìm thấy kết quả nào

sqlQuery += " AND 1 = 0";

}

}

// ----------------------------------

cmd.CommandText = sqlQuery;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvGV.DataSource = dt;

dgvGV.ClearSelection();

// Đặt tên cột hiển thị

dgvGV.Columns["magv"].HeaderText = "Mã GV";

dgvGV.Columns["hoten"].HeaderText = "Họ tên";

dgvGV.Columns["chuyenmon"].HeaderText = "Chuyên môn";

dgvGV.Columns["chunhiemlop"].HeaderText = "Chủ nhiệm lớp";

dgvGV.Columns["sodt"].HeaderText = "Số điện thoại";

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải dữ liệu: " + ex.Message);

}

}

}

// ============================

// 2. LOAD CHI TIẾT LÊN TEXTBOX

// ============================

private void LoadChiTietGV(int magv)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "SELECT \* FROM giaovien WHERE magv = @magv";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@magv", magv);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

txtMaGV.Text = rd["magv"].ToString();

txtHoTenGV.Text = rd["hoten"].ToString();

txtChuyenMon.Text = rd["chuyenmon"] != DBNull.Value ? rd["chuyenmon"].ToString() : "";

txtChuNhiem.Text = rd["chunhiemlop"] != DBNull.Value ? rd["chunhiemlop"].ToString() : "";

txtSDT.Text = rd["sodt"] != DBNull.Value ? rd["sodt"].ToString() : "";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải chi tiết: " + ex.Message);

}

}

}

// ============================

// 3. CÁC HÀM HỖ TRỢ (UI)

// ============================

private void SetButtonState(bool isViewing)

{

btnThem.Enabled = isViewing;

btnSua.Enabled = isViewing;

btnXoa.Enabled = isViewing;

btnBack.Enabled = isViewing;

btnLuu.Enabled = !isViewing;

btnHuy.Enabled = !isViewing; // Đảm bảo nút Hủy sáng khi đang thêm/sửa

txtHoTenGV.ReadOnly = isViewing;

txtChuyenMon.ReadOnly = isViewing;

txtChuNhiem.ReadOnly = isViewing;

txtSDT.ReadOnly = isViewing;

txtMaGV.ReadOnly = true;

}

private void ClearAllFields()

{

txtMaGV.Text = "";

txtHoTenGV.Text = "";

txtChuyenMon.Text = "";

txtChuNhiem.Text = "";

txtSDT.Text = "";

}

// ============================

// 4. SỰ KIỆN CLICK GRID

// ============================

private void dgvGV\_CellClick\_1(object sender, DataGridViewCellEventArgs e)

{

if (btnLuu.Enabled == true) return; // Đang thêm/sửa thì không cho chọn dòng khác

if (e.RowIndex >= 0)

{

try

{

if (dgvGV.Rows[e.RowIndex].Cells["magv"].Value != DBNull.Value)

{

int magv = Convert.ToInt32(dgvGV.Rows[e.RowIndex].Cells["magv"].Value);

LoadChiTietGV(magv);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi chọn dòng: " + ex.Message);

}

}

}

// ============================

// 5. CÁC NÚT CHỨC NĂNG CHÍNH

// ============================

private void btnThem\_Click(object sender, EventArgs e)

{

isAdding = true;

SetButtonState(false);

ClearAllFields();

txtMaGV.Text = "[Tự động]";

txtHoTenGV.Focus();

}

private void btnSua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaGV.Text) || txtMaGV.Text == "[Tự động]")

{

MessageBox.Show("Vui lòng chọn giáo viên để sửa!");

return;

}

isAdding = false;

SetButtonState(false);

txtHoTenGV.Focus();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaGV.Text) || txtMaGV.Text == "[Tự động]")

{

MessageBox.Show("Chọn giáo viên cần xóa!");

return;

}

if (MessageBox.Show("Bạn có chắc chắn muốn xóa giáo viên này?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "DELETE FROM giaovien WHERE magv = @magv";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@magv", Convert.ToInt32(txtMaGV.Text));

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!");

LoadData();

ClearAllFields();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa (có thể giáo viên này đang chủ nhiệm lớp hoặc có dữ liệu liên quan): " + ex.Message);

}

}

}

}

private void btnLuu\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtHoTenGV.Text))

{

MessageBox.Show("Tên giáo viên không được trống!");

return;

}

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

if (isAdding)

{

cmd.CommandText = @"INSERT INTO giaovien (hoten, chuyenmon, chunhiemlop, sodt) VALUES (@hoten, @chuyenmon, @chunhiem, @sodt)";

}

else

{

cmd.CommandText = @"UPDATE giaovien SET hoten=@hoten, chuyenmon=@chuyenmon, chunhiemlop=@chunhiem, sodt=@sodt WHERE magv=@magv";

cmd.Parameters.AddWithValue("@magv", txtMaGV.Text);

}

cmd.Parameters.AddWithValue("@hoten", txtHoTenGV.Text);

cmd.Parameters.AddWithValue("@chuyenmon", txtChuyenMon.Text);

cmd.Parameters.AddWithValue("@chunhiem", txtChuNhiem.Text);

cmd.Parameters.AddWithValue("@sodt", txtSDT.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Lưu thành công!");

SetButtonState(true);

LoadData();

// Nếu vừa thêm mới xong, xóa trắng để đẹp

if (isAdding) ClearAllFields();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi lưu: " + ex.Message);

}

}

}

private void btnHuy\_Click(object sender, EventArgs e)

{

isAdding = false;

SetButtonState(true);

ClearAllFields();

// Nếu đang chọn dòng nào thì load lại dòng đó

if (dgvGV.CurrentRow != null)

{

int rowIndex = dgvGV.CurrentRow.Index;

dgvGV\_CellClick\_1(dgvGV, new DataGridViewCellEventArgs(0, rowIndex));

}

}

// ==========================================

// 6. NÚT TÌM KIẾM (ĐÃ SỬA CHỈ TÌM MÃ)

// ==========================================

private void button1\_Click(object sender, EventArgs e)

{

string keyword = textBox1.Text.Trim(); // Giả sử tên ô nhập là textBox1

// TRƯỜNG HỢP 1: Nếu ô tìm kiếm RỖNG -> Load lại tất cả

if (string.IsNullOrEmpty(keyword))

{

LoadData(); // Gọi hàm không tham số để lấy toàn bộ danh sách

return;

}

// TRƯỜNG HỢP 2: Nếu có nhập gì đó -> Kiểm tra phải là số không

if (!int.TryParse(keyword, out int checkNum))

{

MessageBox.Show("Mã Giáo viên phải là số để tìm kiếm!", "Lỗi");

textBox1.Focus();

return;

}

// TRƯỜNG HỢP 3: Tìm theo mã

LoadData(keyword);

}

private void btnBack\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

### Code form quản lý điểm

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmQuanLyDiem : Form

{

private bool isLoaded = false;

private int currentTeacherId = 0;

// Constructor Mặc định (cho Admin)

public frmQuanLyDiem()

{

InitializeComponent();

txtThongKe.ReadOnly = true;

}

// Constructor mới (cho Teacher)

public frmQuanLyDiem(int teacherId)

{

InitializeComponent();

this.currentTeacherId = teacherId;

txtThongKe.ReadOnly = true;

}

// Tải Lớp (ĐÃ LỌC THEO LỚP DẠY)

private void LoadLop()

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = @"

SELECT DISTINCT l.id, l.tenlop

FROM lop l

LEFT JOIN thoikhoabieu tkb ON tkb.lop\_id = l.id

WHERE 1=1";

SqlCommand cmd = new SqlCommand(sql, conn);

// Lọc lớp mình dạy bằng tkb.magv

if (currentTeacherId > 0)

{

sql += " AND tkb.magv = @teacherId";

cmd.Parameters.AddWithValue("@teacherId", currentTeacherId);

}

sql += " ORDER BY tenlop";

cmd.CommandText = sql;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

DataRow allRow = dt.NewRow();

allRow["id"] = DBNull.Value;

allRow["tenlop"] = (currentTeacherId > 0) ? "--- Tất cả các lớp DẠY ---" : "--- Tất cả các lớp ---";

dt.Rows.InsertAt(allRow, 0);

cboChonLop.DisplayMember = "tenlop";

cboChonLop.ValueMember = "id";

cboChonLop.DataSource = dt;

}

cboChonLop.SelectedIndex = 0;

}

// Tải Điểm (ĐÃ SỬA LỖI VÀ THÊM LẠI BỘ LỌC)

private void LoadDiem(int? lopId = null, string keyword = "")

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

// SQL Cơ bản (Giữ nguyên)

string sql = @"

SELECT

hs.mahs AS [Mã HS],

(hs.holot + ' ' + hs.ten) AS [Họ Tên],

l.tenlop AS [Lớp],

MAX(CASE WHEN mh.tenmonhoc = N'Anh' THEN d.dtb\_mon END) AS [Anh],

MAX(CASE WHEN mh.tenmonhoc = N'GDCD' THEN d.dtb\_mon END) AS [GDCD],

MAX(CASE WHEN mh.tenmonhoc = N'Hóa' THEN d.dtb\_mon END) AS [Hóa],

MAX(CASE WHEN mh.tenmonhoc = N'Lý' THEN d.dtb\_mon END) AS [Lý],

MAX(CASE WHEN mh.tenmonhoc = N'Quốc phòng' THEN d.dtb\_mon END) AS [Quốc phòng],

MAX(CASE WHEN mh.tenmonhoc = N'Sinh' THEN d.dtb\_mon END) AS [Sinh],

MAX(CASE WHEN mh.tenmonhoc = N'Sử' THEN d.dtb\_mon END) AS [Sử],

MAX(CASE WHEN mh.tenmonhoc = N'Thể chất' THEN d.dtb\_mon END) AS [Thể chất],

MAX(CASE WHEN mh.tenmonhoc = N'Toán' THEN d.dtb\_mon END) AS [Toán],

MAX(CASE WHEN mh.tenmonhoc = N'Văn' THEN d.dtb\_mon END) AS [Văn],

MAX(CASE WHEN mh.tenmonhoc = N'Địa' THEN d.dtb\_mon END) AS [Địa],

AVG(d.dtb\_mon) AS [ĐTB],

hs.hocluc AS [Học Lực],

hs.hanhkiem AS [Hạnh Kiểm]

FROM hocsinh hs

JOIN lop l ON l.id = hs.lop\_id

LEFT JOIN diem d ON d.mahs = hs.mahs

LEFT JOIN monhoc mh ON mh.id = d.monhoc\_id

WHERE 1 = 1";

SqlCommand cmd = new SqlCommand(sql, conn);

// --- 1. LỌC GIÁO VIÊN (Giữ nguyên) ---

if (currentTeacherId > 0)

{

sql += " AND hs.lop\_id IN (SELECT DISTINCT lop\_id FROM thoikhoabieu WHERE magv = @teacherId)";

cmd.Parameters.AddWithValue("@teacherId", currentTeacherId);

}

// --- 2. XỬ LÝ TÌM KIẾM THEO TÊN (SỬA ĐOẠN NÀY) ---

if (!string.IsNullOrWhiteSpace(keyword))

{

// Tìm gần đúng theo Họ tên (ghép họ lót + tên)

// Ví dụ: Nhập "Tung" sẽ ra "Nguyen Van Tung", "Tung Son"...

sql += " AND (hs.holot + ' ' + hs.ten) LIKE @keyword";

cmd.Parameters.AddWithValue("@keyword", "%" + keyword + "%");

}

// --- 3. NẾU KHÔNG TÌM KIẾM -> MỚI LỌC THEO LỚP ---

else if (lopId.HasValue && lopId.Value != 0)

{

sql += " AND hs.lop\_id = @lopid";

cmd.Parameters.AddWithValue("@lopid", lopId.Value);

}

// --- GROUP BY và ORDER BY (Giữ nguyên) ---

sql += @"

GROUP BY

hs.mahs, hs.holot, hs.ten, l.tenlop, hs.hocluc, hs.hanhkiem

ORDER BY

l.tenlop, hs.ten";

cmd.CommandText = sql;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvDanhSachHocSinh.DataSource = dt;

foreach (DataColumn col in dt.Columns)

{

if (col.ColumnName == "ĐTB" || col.ColumnName == "Toán" || col.ColumnName == "Văn" /\*...\*/)

{

if (dgvDanhSachHocSinh.Columns.Contains(col.ColumnName))

dgvDanhSachHocSinh.Columns[col.ColumnName].DefaultCellStyle.Format = "N2";

}

}

if (dgvDanhSachHocSinh.Columns.Contains("Mã HS"))

dgvDanhSachHocSinh.Columns["Mã HS"].Visible = false;

dgvDanhSachHocSinh.ClearSelection();

ThongKe(dt);

}

}

// SỬA 3: HÀM THỐNG KÊ (Sửa lỗi logic và tên cột)

private void ThongKe(DataTable dt)

{

if (dt.Rows.Count == 0)

{

txtThongKe.Text = "Không có dữ liệu";

return;

}

// Biến tính tổng điểm TB và sĩ số chung

double tongDiemTB = 0;

int hocSinhCoDiem = 0; // Số học sinh có điểm hợp lệ để tính TB

// Biến đếm học lực

int countGioi = 0;

int countKha = 0;

int countTrungBinh = 0;

int countYeu = 0;

int totalStudents = dt.Rows.Count; // Tổng sĩ số

// 1. Duyệt qua DataRow và thống kê

foreach (DataRow r in dt.Rows)

{

if (r["ĐTB"] != DBNull.Value && double.TryParse(r["ĐTB"].ToString(), out double dtb))

{

tongDiemTB += dtb;

hocSinhCoDiem++;

// Xếp loại học lực

if (dtb >= 8.0)

{

countGioi++;

}

else if (dtb >= 6.5)

{

countKha++;

}

else if (dtb >= 5.0)

{

countTrungBinh++;

}

else

{

countYeu++;

}

}

else

{

// Nếu không có điểm, mặc định coi là Yếu hoặc xếp loại riêng

// Tùy theo logic của trường, ở đây ta coi là Yếu (0-4.99)

// countYeu++;

// KHÔNG NÊN ĐẾM vào Yếu nếu điểm là NULL. Chỉ đếm vào sĩ số chung.

}

}

// 2. Tính Điểm TB Chung và Phần trăm

double tbChung = (hocSinhCoDiem > 0) ? tongDiemTB / hocSinhCoDiem : 0;

double percentGioi = (double)countGioi / totalStudents \* 100;

double percentKha = (double)countKha / totalStudents \* 100;

double percentTrungBinh = (double)countTrungBinh / totalStudents \* 100;

double percentYeu = (double)countYeu / totalStudents \* 100;

// 3. Hiển thị kết quả thống kê

string thongKeText =

$"Sĩ số: {totalStudents} | ĐTB Chung: {tbChung:0.00} " +

$"Giỏi: {countGioi} ({percentGioi:0.0}%) | " +

$"Khá: {countKha} ({percentKha:0.0}%) | " +

$"TB: {countTrungBinh} ({percentTrungBinh:0.0}%) | " +

$"Yếu: {countYeu} ({percentYeu:0.0}%)";

txtThongKe.Text = thongKeText;

}

// Sự kiện Form Load

private void frmQuanLyDiem\_Load(object sender, EventArgs e)

{

LoadLop();

isLoaded = true;

cboChonLop\_SelectedIndexChanged(null, null);

}

// Sự kiện ComboBox Lọc

private void cboChonLop\_SelectedIndexChanged(object sender, EventArgs e)

{

if (!isLoaded) return;

int? lopId = null;

if (cboChonLop.SelectedValue != DBNull.Value && cboChonLop.SelectedValue != null)

{

if (int.TryParse(cboChonLop.SelectedValue.ToString(), out int id))

{

lopId = id;

}

}

txtTimKiem.Clear();

LoadDiem(lopId, null);

}

// Nút Tìm kiếm

private void button1\_Click(object sender, EventArgs e)

{

string kw = txtTimKiem.Text.Trim();

// ===========================================================

// TRƯỜNG HỢP 1: Ô TÌM KIẾM RỖNG -> LOAD LẠI THEO LỚP ĐANG CHỌN

// ===========================================================

if (string.IsNullOrEmpty(kw))

{

// Lấy ID lớp đang chọn trong ComboBox

int? lopId = null;

if (cboChonLop.SelectedValue != null && cboChonLop.SelectedValue != DBNull.Value)

{

if (int.TryParse(cboChonLop.SelectedValue.ToString(), out int id))

{

lopId = id;

}

}

// Load lại danh sách chỉ theo Lớp (bỏ qua từ khóa)

LoadDiem(lopId, null);

return;

}

// TRƯỜNG HỢP 2: CÓ NHẬP TÊN -> TÌM TRÊN TOÀN BỘ (HOẶC TOÀN KHỐI DẠY)

// [Tùy chọn] Reset ComboBox về "Tất cả" để người dùng hiểu là đang tìm rộng

if (cboChonLop.Items.Count > 0)

{

isLoaded = false; // Khóa sự kiện

cboChonLop.SelectedIndex = 0; // Về "Tất cả"

isLoaded = true; // Mở khóa

}

// Truyền null vào tham số lopId để BỎ QUA bộ lọc lớp

LoadDiem(null, kw);

}

// Nút Xem chi tiết (Bảng điểm cá nhân)

private void button2\_Click(object sender, EventArgs e)

{

if (dgvDanhSachHocSinh.CurrentRow == null)

{

MessageBox.Show("Vui lòng chọn học sinh muốn xem chi tiết!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// SỬA 4: Sửa tên cột "mahs" thành "Mã HS"

int mahs = Convert.ToInt32(dgvDanhSachHocSinh.CurrentRow.Cells["Mã HS"].Value);

frmbangdiemcanhan frm = new frmbangdiemcanhan();

frm.MaHocSinh = mahs;

frm.ShowDialog();

cboChonLop\_SelectedIndexChanged(null, null);

}

// Nút Đóng

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

// Xóa hàm dư thừa này trong Designer

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

cboChonLop\_SelectedIndexChanged(sender, e);

}

}

}

### Code form bảng điểm cá nhân

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmbangdiemcanhan : Form

{

private bool isAdding = false;

// private bool isLoaded = false; // Đã bỏ dòng này để loại bỏ cảnh báo CS0414 nếu không dùng

// Lưu ID của dòng điểm đang chọn để sửa/xóa

private int selectedDiemId = -1;

// Dùng thuộc tính này để nhận Mã Học sinh từ Form gọi

public int MaHocSinh { get; set; }

// Hàm khởi tạo mặc định (VS cần)

public frmbangdiemcanhan()

{

InitializeComponent();

// Đặt các TextBox thông tin học sinh chỉ hiển thị

txtTenHocSinh.ReadOnly = true;

txtTenLop.ReadOnly = true;

}

// --- HÀM HỖ TRỢ VÀ TÍNH TOÁN ---

private float? ConvertToFloat(string text)

{

if (string.IsNullOrWhiteSpace(text)) return null;

// Chấp nhận cả dấu phẩy và dấu chấm, và chỉ cho phép giá trị từ 0 đến 10

if (float.TryParse(text.Replace(",", "."), out float value))

{

if (value >= 0 && value <= 10) return value;

}

return null;

}

private float CalculateDiemTB(float? mieng, float? kt15p1, float? kt15p2, float? giuaKy, float? cuoiKy)

{

float sumWeightedScores = (mieng ?? 0) \* 1 + (kt15p1 ?? 0) \* 1 + (kt15p2 ?? 0) \* 1 + (giuaKy ?? 0) \* 2 + (cuoiKy ?? 0) \* 3;

int actualTotalWeight = (mieng.HasValue ? 1 : 0) + (kt15p1.HasValue ? 1 : 0) + (kt15p2.HasValue ? 1 : 0) + (giuaKy.HasValue ? 2 : 0) + (cuoiKy.HasValue ? 3 : 0);

if (actualTotalWeight == 0) return 0f;

return (float)Math.Round(sumWeightedScores / actualTotalWeight, 1);

}

// --- LOAD DỮ LIỆU CƠ SỞ ---

private void LoadThongTinHocSinh(int mahs)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = @"

SELECT

(hs.holot + ' ' + hs.ten) AS [Họ Tên], l.tenlop

FROM hocsinh hs

JOIN lop l ON l.id = hs.lop\_id

WHERE hs.mahs = @mahs";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

txtTenHocSinh.Text = reader["Họ Tên"].ToString();

txtTenLop.Text = reader["tenlop"].ToString();

}

else

{

txtTenHocSinh.Text = "Không tìm thấy học sinh";

txtTenLop.Text = "";

MessageBox.Show("Không tìm thấy thông tin học sinh với mã " + mahs, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

this.Close();

}

reader.Close();

}

}

private void LoadMonHoc()

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = "SELECT id, tenmonhoc FROM monhoc ORDER BY tenmonhoc";

SqlDataAdapter da = new SqlDataAdapter(sql, conn);

DataTable dt = new DataTable();

da.Fill(dt);

cboMonHoc.DataSource = dt;

cboMonHoc.DisplayMember = "tenmonhoc";

cboMonHoc.ValueMember = "id";

}

cboMonHoc.SelectedIndex = -1;

}

// --- LOAD ĐIỂM CỦA HỌC SINH ĐƯỢC TRUYỀN VÀO ---

private void LoadDiem(int mahs, int hocky) // >> Đã thêm tham số hocky

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "";

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

if (hocky == 3) // >> Tính điểm Cả năm

{

sql = @"

SELECT

m.tenmonhoc AS [Môn học],

AVG(CASE WHEN d.hocky = 1 THEN d.dtb\_mon END) AS [ĐTB HK1],

AVG(CASE WHEN d.hocky = 2 THEN d.dtb\_mon END) AS [ĐTB HK2],

AVG(d.dtb\_mon) AS [ĐTB Chung]

FROM diem d

JOIN monhoc m ON m.id = d.monhoc\_id

WHERE d.mahs = @mahs AND d.dtb\_mon IS NOT NULL -- Chỉ tính môn đã có điểm

GROUP BY m.tenmonhoc

ORDER BY m.tenmonhoc";

// Xóa hết các cột cũ để nạp cấu trúc mới

dgvDiemCaNhan.Columns.Clear();

// Ẩn các nút thao tác khi xem điểm Cả năm

SetButtonState(true);

btnThem.Enabled = false;

btnSua.Enabled = false;

btnXoa.Enabled = false;

}

else // >> Chỉ xem điểm theo Học kỳ 1 hoặc 2

{

sql = @"

SELECT

d.id, -- Phải giữ cột ID để Sửa/Xóa

m.tenmonhoc AS [Môn học],

d.mieng AS [Miệng],

d.kt\_15p AS [KT 15p (1)],

d.kt\_15p\_lan2 AS [KT 15p (2)],

d.giua\_ky AS [Giữa kỳ],

d.cuoi\_ky AS [Cuối kỳ],

d.dtb\_mon AS [ĐTB Môn]

FROM diem d

JOIN monhoc m ON m.id = d.monhoc\_id

WHERE d.mahs = @mahs AND d.hocky = @hocky -- >> THÊM ĐIỀU KIỆN LỌC HỌC KỲ

ORDER BY m.tenmonhoc";

cmd.Parameters.AddWithValue("@hocky", hocky);

SetButtonState(true); // Mở lại các nút thao tác

}

cmd.CommandText = sql;

cmd.Parameters.AddWithValue("@mahs", mahs);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvDiemCaNhan.DataSource = dt;

// Định dạng cột

if (hocky == 3)

{

if (dgvDiemCaNhan.Columns.Contains("ĐTB HK1"))

dgvDiemCaNhan.Columns["ĐTB HK1"].DefaultCellStyle.Format = "N2";

if (dgvDiemCaNhan.Columns.Contains("ĐTB HK2"))

dgvDiemCaNhan.Columns["ĐTB HK2"].DefaultCellStyle.Format = "N2";

if (dgvDiemCaNhan.Columns.Contains("ĐTB Chung"))

dgvDiemCaNhan.Columns["ĐTB Chung"].DefaultCellStyle.Format = "N2"; // Hiển thị 2 số TP

}

else

{

// Định dạng điểm thành phần (1 số TP) và ĐTB Môn (2 số TP)

foreach (DataGridViewColumn col in dgvDiemCaNhan.Columns)

{

if (col.Name == "Miệng" || col.Name == "KT 15p (1)" || col.Name == "KT 15p (2)" || col.Name == "Giữa kỳ" || col.Name == "Cuối kỳ")

{

col.DefaultCellStyle.Format = "N1";

}

else if (col.Name == "ĐTB Môn")

{

col.DefaultCellStyle.Format = "N2";

}

}

}

if (dt.Columns.Contains("id"))

dgvDiemCaNhan.Columns["id"].Visible = false;

dgvDiemCaNhan.ClearSelection();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi tải bảng điểm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

// --- ĐIỀU CHỈNH TRẠNG THÁI FORM ---

private void SetButtonState(bool view)

{

btnThem.Enabled = view;

// Bật SỬA/XÓA chỉ khi đang ở chế độ VIEW VÀ đã chọn một ID hợp lệ

btnSua.Enabled = view && (selectedDiemId != -1);

btnXoa.Enabled = view && (selectedDiemId != -1);

btnLuu.Enabled = !view;

btnHuy.Enabled = !view;

// Bật/tắt các ô nhập điểm

cboMonHoc.Enabled = !view;

txtMieng.ReadOnly = view;

txtKT15p1.ReadOnly = view;

txtKT15p2.ReadOnly = view;

txtGiuaKy.ReadOnly = view;

txtCuoiKy.ReadOnly = view;

}

private void ClearInputFields()

{

cboMonHoc.SelectedIndex = -1;

txtMieng.Clear();

txtKT15p1.Clear();

txtKT15p2.Clear();

txtGiuaKy.Clear();

txtCuoiKy.Clear();

// CHỈ RESET selectedDiemId khi clear/kết thúc chỉnh sửa

selectedDiemId = -1;

SetButtonState(true); // Cập nhật lại trạng thái nút (Sửa/Xóa bị tắt)

}

private void frmbangdiemcanhan\_Load(object sender, EventArgs e)

{

// isLoaded = false; // Bỏ nếu không dùng

// Kiểm tra MaHocSinh (ID = 0 là không hợp lệ)

if (MaHocSinh <= 0)

{

MessageBox.Show("Mã học sinh không hợp lệ. Vui lòng thử lại.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

this.Close();

return;

}

LoadMonHoc();

LoadThongTinHocSinh(MaHocSinh);

LoadHocKyComboBox();

comboBox1\_SelectedIndexChanged(null, null);

// Bổ sung: Tự động chọn dòng đầu tiên nếu có data để hiển thị và kích hoạt nút Sửa/Xóa

if (dgvDiemCaNhan.Rows.Count > 0)

{

// Chọn dòng đầu tiên

dgvDiemCaNhan.CurrentCell = dgvDiemCaNhan.Rows[0].Cells[1];

// Gọi sự kiện CellClick để load data và bật nút Sửa/Xóa

dgvDiemCaNhan\_CellClick(null, new DataGridViewCellEventArgs(1, 0)); // giả sử cột 1 là cột Môn học

}

else

{

ClearInputFields();

}

SetButtonState(true);

// isLoaded = true; // Bỏ nếu không dùng

}

// --- SỰ KIỆN CELL CLICK (ĐÃ SỬA VÀ THAY THẾ CellContentClick) ---

private void dgvDiemCaNhan\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// 1. Nếu đang ở chế độ Lưu/Hủy (Edit/Add), không làm gì cả

if (!btnThem.Enabled)

return;

// 2. Nếu click vào hàng hợp lệ

if (e.RowIndex >= 0 && e.RowIndex < dgvDiemCaNhan.Rows.Count)

{

DataGridViewRow row = dgvDiemCaNhan.Rows[e.RowIndex];

// --- Điền dữ liệu và Lưu ID ---

// Kiểm tra null cho cột ID (Rất quan trọng)

if (row.Cells["id"].Value != DBNull.Value)

selectedDiemId = Convert.ToInt32(row.Cells["id"].Value);

else

selectedDiemId = -1;

// Kiểm tra giá trị Mon học trước khi gán

if (row.Cells["Môn học"].Value != DBNull.Value)

cboMonHoc.Text = row.Cells["Môn học"].Value.ToString();

else

cboMonHoc.SelectedIndex = -1;

// Kiểm tra DBNull cho điểm số trước khi gán

txtMieng.Text = row.Cells["Miệng"].Value != DBNull.Value ? row.Cells["Miệng"].Value.ToString() : "";

txtKT15p1.Text = row.Cells["KT 15p (1)"].Value != DBNull.Value ? row.Cells["KT 15p (1)"].Value.ToString() : "";

txtKT15p2.Text = row.Cells["KT 15p (2)"].Value != DBNull.Value ? row.Cells["KT 15p (2)"].Value.ToString() : "";

txtGiuaKy.Text = row.Cells["Giữa kỳ"].Value != DBNull.Value ? row.Cells["Giữa kỳ"].Value.ToString() : "";

txtCuoiKy.Text = row.Cells["Cuối kỳ"].Value != DBNull.Value ? row.Cells["Cuối kỳ"].Value.ToString() : "";

// Gọi SetButtonState(true) để BẬT nút SỬA và XÓA (nếu selectedDiemId != -1)

SetButtonState(true);

}

else

{

// Nếu click vào vùng trống, reset

ClearInputFields();

SetButtonState(true);

}

}

// --- CÁC NÚT THAO TÁC ---

private void LoadHocKyComboBox()

{

// Tạo DataTable thủ công cho ComboBox Học kỳ

DataTable dtHocKy = new DataTable();

dtHocKy.Columns.Add("ID", typeof(int));

dtHocKy.Columns.Add("Name", typeof(string));

dtHocKy.Rows.Add(1, "Học kỳ 1");

dtHocKy.Rows.Add(2, "Học kỳ 2");

dtHocKy.Rows.Add(3, "Cả năm"); // ID 3 để tính điểm trung bình cả năm

cboHocKy.DataSource = dtHocKy;

cboHocKy.DisplayMember = "Name";

cboHocKy.ValueMember = "ID";

// Mặc định chọn Học kỳ 1 (ID=1)

cboHocKy.SelectedValue = 1;

}

private void btnLuu\_Click(object sender, EventArgs e)

{

if (cboMonHoc.SelectedValue == null)

{

MessageBox.Show("Vui lòng chọn Môn học.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// >> Bổ sung: Lấy Học kỳ

int hocky = Convert.ToInt32(cboHocKy.SelectedValue);

if (hocky == 3) // Không cho Thêm/Sửa khi đang chọn "Cả năm"

{

MessageBox.Show("Không thể Thêm/Sửa khi đang xem điểm Cả năm.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// Lấy monhoc\_id và mahs từ thuộc tính của form

int mahs = this.MaHocSinh;

int monhocId = Convert.ToInt32(cboMonHoc.SelectedValue);

// Chuyển đổi điểm

float? mieng = ConvertToFloat(txtMieng.Text);

float? kt15p1 = ConvertToFloat(txtKT15p1.Text);

float? kt15p2 = ConvertToFloat(txtKT15p2.Text);

float? giuaKy = ConvertToFloat(txtGiuaKy.Text);

float? cuoiKy = ConvertToFloat(txtCuoiKy.Text);

float dtbMon = CalculateDiemTB(mieng, kt15p1, kt15p2, giuaKy, cuoiKy);

try

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

if (isAdding)

{

// >> Sửa Kiểm tra trùng lặp (Thêm điều kiện hocky)

string checkSql = "SELECT COUNT(\*) FROM diem WHERE mahs = @mahs AND monhoc\_id = @monhocId AND hocky = @hocky";

SqlCommand checkCmd = new SqlCommand(checkSql, conn);

checkCmd.Parameters.AddWithValue("@mahs", mahs);

checkCmd.Parameters.AddWithValue("@monhocId", monhocId);

checkCmd.Parameters.AddWithValue("@hocky", hocky);

if ((int)checkCmd.ExecuteScalar() > 0)

{

MessageBox.Show("Điểm cho môn học này đã tồn tại trong học kỳ này. Vui lòng sử dụng chức năng Sửa.", "Lỗi trùng lặp", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// >> Sửa INSERT (Thêm hocky)

string sqlAdd = @"

INSERT INTO diem (mahs, monhoc\_id, mieng, kt\_15p, kt\_15p\_lan2, giua\_ky, cuoi\_ky, dtb\_mon, hocky)

VALUES (@mahs, @monhocId, @mieng, @kt15p1, @kt15p2, @giuaKy, @cuoiKy, @dtbMon, @hocky)";

SqlCommand cmd = new SqlCommand(sqlAdd, conn);

cmd.Parameters.AddWithValue("@mahs", mahs);

cmd.Parameters.AddWithValue("@monhocId", monhocId);

cmd.Parameters.AddWithValue("@mieng", (object)mieng ?? DBNull.Value);

cmd.Parameters.AddWithValue("@kt15p1", (object)kt15p1 ?? DBNull.Value);

cmd.Parameters.AddWithValue("@kt15p2", (object)kt15p2 ?? DBNull.Value);

cmd.Parameters.AddWithValue("@giuaKy", (object)giuaKy ?? DBNull.Value);

cmd.Parameters.AddWithValue("@cuoiKy", (object)cuoiKy ?? DBNull.Value);

cmd.Parameters.AddWithValue("@dtbMon", dtbMon);

cmd.Parameters.AddWithValue("@hocky", hocky); // >> THAM SỐ HỌC KỲ ĐÃ ĐƯỢC THÊM

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm điểm thành công!");

}

else // Chức năng Sửa

{

if (selectedDiemId == -1)

{

MessageBox.Show("Không có môn học nào được chọn để sửa.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string sqlUpdate = @"

UPDATE diem SET

mieng = @mieng,

kt\_15p = @kt15p1,

kt\_15p\_lan2 = @kt15p2,

giua\_ky = @giuaKy,

cuoi\_ky = @cuoiKy,

dtb\_mon = @dtbMon

WHERE id = @id";

SqlCommand cmd = new SqlCommand(sqlUpdate, conn);

cmd.Parameters.AddWithValue("@id", selectedDiemId);

cmd.Parameters.AddWithValue("@mieng", (object)mieng ?? DBNull.Value);

cmd.Parameters.AddWithValue("@kt15p1", (object)kt15p1 ?? DBNull.Value);

cmd.Parameters.AddWithValue("@kt15p2", (object)kt15p2 ?? DBNull.Value);

cmd.Parameters.AddWithValue("@giuaKy", (object)giuaKy ?? DBNull.Value);

cmd.Parameters.AddWithValue("@cuoiKy", (object)cuoiKy ?? DBNull.Value);

cmd.Parameters.AddWithValue("@dtbMon", dtbMon);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật điểm thành công!");

}

isAdding = false;

ClearInputFields(); // Dùng ClearInputFields để reset selectedDiemId và trạng thái nút

LoadDiem(mahs, hocky); // >> Đã thêm tham số hocky

}

}

catch (SqlException sqlex)

{

MessageBox.Show("Lỗi SQL: " + sqlex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnXoa\_Click(object sender, EventArgs e)

{

if (selectedDiemId == -1)

{

MessageBox.Show("Chọn điểm môn học muốn xóa!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (MessageBox.Show("Bạn có chắc chắn muốn xóa điểm môn này không?", "Xác nhận xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)

return;

try

{

using (SqlConnection conn = KetNoi.GetConnection())

{

conn.Open();

string sql = "DELETE FROM diem WHERE id = @id";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@id", selectedDiemId);

cmd.ExecuteNonQuery();

}

MessageBox.Show("Xóa thành công!");

}

catch (Exception ex)

{

MessageBox.Show("Lỗi xóa điểm: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

int hocky = Convert.ToInt32(cboHocKy.SelectedValue);

LoadDiem(this.MaHocSinh, hocky); // >> Đã thêm tham số hocky

ClearInputFields();

}

private void btnThem\_Click(object sender, EventArgs e)

{

isAdding = true;

ClearInputFields();

SetButtonState(false);

cboMonHoc.Enabled = true;

}

private void btnSua\_Click(object sender, EventArgs e)

{

if (selectedDiemId == -1)

{

MessageBox.Show("Vui lòng chọn môn học cần sửa trong bảng.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

isAdding = false;

SetButtonState(false);

// Ngăn người dùng sửa môn học khi sửa điểm

cboMonHoc.Enabled = false;

}

private void btnHuy\_Click(object sender, EventArgs e)

{

isAdding = false;

ClearInputFields();

SetButtonState(true);

// >> Load lại điểm của Học kỳ đang chọn

int hocky = Convert.ToInt32(cboHocKy.SelectedValue);

LoadDiem(this.MaHocSinh, hocky);

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

// 1. Kiểm tra SelectedValue là null (trường hợp khởi tạo)

if (cboHocKy.SelectedValue == null)

{

return;

}

// 2. Ép kiểu SelectedValue một cách an toàn

// Đảm bảo rằng SelectedValue có thể chuyển đổi thành int (giả định cột ID là INT)

if (int.TryParse(cboHocKy.SelectedValue.ToString(), out int selectedHocKy))

{

// >> Gọi hàm LoadDiem mới

LoadDiem(this.MaHocSinh, selectedHocKy);

ClearInputFields();

// Bổ sung: Tự động chọn dòng đầu tiên nếu có data để kích hoạt nút Sửa/Xóa (chỉ áp dụng cho HK1/HK2)

if (dgvDiemCaNhan.Rows.Count > 0 && selectedHocKy != 3)

{

dgvDiemCaNhan.CurrentCell = dgvDiemCaNhan.Rows[0].Cells[1];

dgvDiemCaNhan\_CellClick(null, new DataGridViewCellEventArgs(1, 0));

}

}

// Nếu không parse được (lỗi DataRowView), ta bỏ qua, tránh crash.

}

}

}

### Code form thời khóa biểu

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Quanlyhocsinh

{

public partial class frmQuanLyGiaoVien : Form

{

private bool isAdding = false;

public frmQuanLyGiaoVien()

{

InitializeComponent();

}

private void frmQuanLyGiaoVien\_Load(object sender, EventArgs e)

{

LoadData();

SetButtonState(true);

dgvGV.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

dgvGV.MultiSelect = false;

this.dgvGV.CellClick += new System.Windows.Forms.DataGridViewCellEventHandler(this.dgvGV\_CellClick\_1);

}

// ==========================================

// 1. LOAD DANH SÁCH GIÁO VIÊN (SỬA LOGIC TÌM MÃ)

// ==========================================

private void LoadData(string keyword = null)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sqlQuery = @"

SELECT magv, hoten, chuyenmon, chunhiemlop, sodt

FROM giaovien

WHERE 1 = 1";

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

// --- LOGIC TÌM KIẾM THEO MÃ GV ---

if (!string.IsNullOrEmpty(keyword))

{

// Kiểm tra nếu là số thì tìm theo Mã

if (int.TryParse(keyword, out int maGVTimKiem))

{

sqlQuery += " AND magv = @maGV";

cmd.Parameters.AddWithValue("@maGV", maGVTimKiem);

}

else

{

// Nếu nhập chữ -> Không tìm thấy kết quả nào

sqlQuery += " AND 1 = 0";

}

}

// ----------------------------------

cmd.CommandText = sqlQuery;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dgvGV.DataSource = dt;

dgvGV.ClearSelection();

// Đặt tên cột hiển thị

dgvGV.Columns["magv"].HeaderText = "Mã GV";

dgvGV.Columns["hoten"].HeaderText = "Họ tên";

dgvGV.Columns["chuyenmon"].HeaderText = "Chuyên môn";

dgvGV.Columns["chunhiemlop"].HeaderText = "Chủ nhiệm lớp";

dgvGV.Columns["sodt"].HeaderText = "Số điện thoại";

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải dữ liệu: " + ex.Message);

}

}

}

// ============================

// 2. LOAD CHI TIẾT LÊN TEXTBOX

// ============================

private void LoadChiTietGV(int magv)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "SELECT \* FROM giaovien WHERE magv = @magv";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@magv", magv);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.Read())

{

txtMaGV.Text = rd["magv"].ToString();

txtHoTenGV.Text = rd["hoten"].ToString();

txtChuyenMon.Text = rd["chuyenmon"] != DBNull.Value ? rd["chuyenmon"].ToString() : "";

txtChuNhiem.Text = rd["chunhiemlop"] != DBNull.Value ? rd["chunhiemlop"].ToString() : "";

txtSDT.Text = rd["sodt"] != DBNull.Value ? rd["sodt"].ToString() : "";

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tải chi tiết: " + ex.Message);

}

}

}

// ============================

// 3. CÁC HÀM HỖ TRỢ (UI)

// ============================

private void SetButtonState(bool isViewing)

{

btnThem.Enabled = isViewing;

btnSua.Enabled = isViewing;

btnXoa.Enabled = isViewing;

btnBack.Enabled = isViewing;

btnLuu.Enabled = !isViewing;

btnHuy.Enabled = !isViewing; // Đảm bảo nút Hủy sáng khi đang thêm/sửa

txtHoTenGV.ReadOnly = isViewing;

txtChuyenMon.ReadOnly = isViewing;

txtChuNhiem.ReadOnly = isViewing;

txtSDT.ReadOnly = isViewing;

txtMaGV.ReadOnly = true;

}

private void ClearAllFields()

{

txtMaGV.Text = "";

txtHoTenGV.Text = "";

txtChuyenMon.Text = "";

txtChuNhiem.Text = "";

txtSDT.Text = "";

}

// ============================

// 4. SỰ KIỆN CLICK GRID

// ============================

private void dgvGV\_CellClick\_1(object sender, DataGridViewCellEventArgs e)

{

if (btnLuu.Enabled == true) return; // Đang thêm/sửa thì không cho chọn dòng khác

if (e.RowIndex >= 0)

{

try

{

if (dgvGV.Rows[e.RowIndex].Cells["magv"].Value != DBNull.Value)

{

int magv = Convert.ToInt32(dgvGV.Rows[e.RowIndex].Cells["magv"].Value);

LoadChiTietGV(magv);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi chọn dòng: " + ex.Message);

}

}

}

// ============================

// 5. CÁC NÚT CHỨC NĂNG CHÍNH

// ============================

private void btnThem\_Click(object sender, EventArgs e)

{

isAdding = true;

SetButtonState(false);

ClearAllFields();

txtMaGV.Text = "[Tự động]";

txtHoTenGV.Focus();

}

private void btnSua\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaGV.Text) || txtMaGV.Text == "[Tự động]")

{

MessageBox.Show("Vui lòng chọn giáo viên để sửa!");

return;

}

isAdding = false;

SetButtonState(false);

txtHoTenGV.Focus();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtMaGV.Text) || txtMaGV.Text == "[Tự động]")

{

MessageBox.Show("Chọn giáo viên cần xóa!");

return;

}

if (MessageBox.Show("Bạn có chắc chắn muốn xóa giáo viên này?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

string sql = "DELETE FROM giaovien WHERE magv = @magv";

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@magv", Convert.ToInt32(txtMaGV.Text));

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!");

LoadData();

ClearAllFields();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi xóa (có thể giáo viên này đang chủ nhiệm lớp hoặc có dữ liệu liên quan): " + ex.Message);

}

}

}

}

private void btnLuu\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txtHoTenGV.Text))

{

MessageBox.Show("Tên giáo viên không được trống!");

return;

}

using (SqlConnection conn = KetNoi.GetConnection())

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = conn;

if (isAdding)

{

cmd.CommandText = @"INSERT INTO giaovien (hoten, chuyenmon, chunhiemlop, sodt) VALUES (@hoten, @chuyenmon, @chunhiem, @sodt)";

}

else

{

cmd.CommandText = @"UPDATE giaovien SET hoten=@hoten, chuyenmon=@chuyenmon, chunhiemlop=@chunhiem, sodt=@sodt WHERE magv=@magv";

cmd.Parameters.AddWithValue("@magv", txtMaGV.Text);

}

cmd.Parameters.AddWithValue("@hoten", txtHoTenGV.Text);

cmd.Parameters.AddWithValue("@chuyenmon", txtChuyenMon.Text);

cmd.Parameters.AddWithValue("@chunhiem", txtChuNhiem.Text);

cmd.Parameters.AddWithValue("@sodt", txtSDT.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Lưu thành công!");

SetButtonState(true);

LoadData();

// Nếu vừa thêm mới xong, xóa trắng để đẹp

if (isAdding) ClearAllFields();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi lưu: " + ex.Message);

}

}

}

private void btnHuy\_Click(object sender, EventArgs e)

{

isAdding = false;

SetButtonState(true);

ClearAllFields();

// Nếu đang chọn dòng nào thì load lại dòng đó

if (dgvGV.CurrentRow != null)

{

int rowIndex = dgvGV.CurrentRow.Index;

dgvGV\_CellClick\_1(dgvGV, new DataGridViewCellEventArgs(0, rowIndex));

}

}

// ==========================================

// 6. NÚT TÌM KIẾM (ĐÃ SỬA CHỈ TÌM MÃ)

// ==========================================

private void button1\_Click(object sender, EventArgs e)

{

string keyword = textBox1.Text.Trim(); // Giả sử tên ô nhập là textBox1

// TRƯỜNG HỢP 1: Nếu ô tìm kiếm RỖNG -> Load lại tất cả

if (string.IsNullOrEmpty(keyword))

{

LoadData(); // Gọi hàm không tham số để lấy toàn bộ danh sách

return;

}

// TRƯỜNG HỢP 2: Nếu có nhập gì đó -> Kiểm tra phải là số không

if (!int.TryParse(keyword, out int checkNum))

{

MessageBox.Show("Mã Giáo viên phải là số để tìm kiếm!", "Lỗi");

textBox1.Focus();

return;

}

// TRƯỜNG HỢP 3: Tìm theo mã

LoadData(keyword);

}

private void btnBack\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết quả Đạt được (Achieved Results)

Sau quá trình nghiên cứu, phân tích và triển khai, đồ án "Xây dựng Ứng dụng Quản lý Học sinh với C# và MS SQL Server" đã hoàn thành các mục tiêu đề ra ban đầu. Cụ thể, hệ thống đã đạt được những kết quả chính sau:

* Xây dựng Ứng dụng Desktop Hoàn chỉnh: Xây dựng thành công ứng dụng Desktop dựa trên nền tảng .NET WinForms, hoạt động ổn định và đáp ứng các yêu cầu nghiệp vụ quản lý cơ bản của một trường học.
* Áp dụng Kiến trúc Phân lớp: Mã nguồn được tổ chức theo kiến trúc 3 lớp (3-Tier Architecture) với lớp Data Access Layer (DAL) tách biệt, đảm bảo tính dễ bảo trì và khả năng mở rộng.
* Đảm bảo Tính Toàn vẹn Dữ liệu: Đã sử dụng hiệu quả công nghệ ADO.NET kết hợp với cơ chế Transaction (SqlTransaction) trong MS SQL Server để xử lý các nghiệp vụ phức tạp (như thêm Học sinh và Phụ huynh), đảm bảo dữ liệu luôn nhất quán.
* Triển khai Cơ chế Phân quyền (RBAC): Hệ thống thực hiện phân quyền chặt chẽ ngay tại Tầng Giao diện, sử dụng logic Context-Aware để ẩn/hiện chức năng linh hoạt cho Admin, Giáo viên và Học sinh.
* Trực quan hóa Dữ liệu: Thành công trong việc sử dụng truy vấn T-SQL phức tạp (dùng MAX(CASE WHEN...) và GROUP BY) để mô phỏng tính năng Pivot Table trong Form quản lý điểm, giúp trực quan hóa kết quả học tập.

## 5.2. Hạn chế (Limitations)

Bên cạnh những kết quả đạt được, ứng dụng vẫn còn một số hạn chế cần khắc phục trong các phiên bản tiếp theo:

* Khả năng Xuất Báo cáo: Hiện tại, ứng dụng chưa hỗ trợ chức năng xuất báo cáo, danh sách học sinh hay bảng điểm ra các định dạng văn phòng phổ biến như Excel (.xlsx) hay PDF để phục vụ công tác in ấn chính thức.
* Thiếu Biểu đồ Trực quan: Chức năng thống kê báo cáo còn ở mức cơ bản (chỉ là văn bản thông báo), chưa tích hợp các biểu đồ trực quan (như Biểu đồ tròn, Biểu đồ cột) để phân tích xu hướng học tập của học sinh.
* Giao diện Tĩnh: Do sử dụng WinForms, giao diện còn khá truyền thống và chưa thể hỗ trợ chế độ giao diện sáng/tối hoặc các yếu tố hiện đại như trong các framework mới hơn (như WPF hoặc web app).

## 5.3. Hướng phát triển (Future Development)

Để ứng dụng hoàn thiện và có tính thực tiễn cao hơn, hướng phát triển trong tương lai bao gồm:

* Tích hợp Báo cáo (Reporting): Nghiên cứu và tích hợp các công cụ báo cáo chuyên nghiệp của .NET như ReportViewer hoặc sử dụng thư viện bên ngoài (ví dụ: EPPlus, iTextSharp) để tạo file Excel và PDF cho các phiếu điểm và danh sách.
* Trực quan hóa Dữ liệu (Data Visualization): Sử dụng thư viện Chart Control (trong WinForms) hoặc thư viện OxyPlot để vẽ các biểu đồ thống kê tỷ lệ học lực, hạnh kiểm trực quan ngay trên giao diện Dashboard.
* Nâng cấp Kiến trúc và Kết nối: Chuyển đổi kiến trúc sang mô hình WPF hoặc phát triển phiên bản Web API (sử dụng ASP.NET Core) để cho phép phụ huynh và học sinh tra cứu thông tin từ xa qua mạng Internet.
* Kiểm tra Ràng buộc Nghiêm ngặt: Hoàn thiện thuật toán kiểm tra trùng lịch (Collision Check) để không chỉ kiểm tra Giáo viên và Lớp mà còn kiểm tra ràng buộc về Khối (Grade Level) khi phân công môn học.