

Đề tài 05: Phát triển Ứng dụng quản lý đăng ký và thi chứng chỉ kỹ năng lái xe an toàn

Sinh viên dựa vào gợi ý đề tài (chỉ mang tính chất tham khảo) để tự thiết kế và xây dựng một Ứng dụng quản lý đăng ký và thi chứng chỉ kỹ năng lái xe an toàn hoàn chỉnh.

1. Yêu cầu kỹ thuật và công nghệ

- **Nền tảng:** WPF (Windows Presentation Foundation).
- **Cơ sở dữ liệu:** SQL Server để lưu trữ thông tin + EntityFrameworkCore.
- **Mô hình:** Ba lớp (Three Layers), hoặc MVC (Model-View-Controller), hoặc MVVM (Model-View-ViewModel).
- **Giao diện:** Thân thiện, trực quan

2. Yêu cầu về báo cáo

- Phân tích, thiết kế chức năng hệ thống và cơ sở dữ liệu.
- Phân công rõ ràng nhiệm vụ cho từng thành viên nhóm.
- Kết quả thực hiện bao gồm:
 - Mã nguồn ứng dụng.
 - Tài liệu thiết kế và báo cáo phân tích.
 - Hình ảnh giao diện và chức năng của ứng dụng.

GỢI Ý ĐỀ TÀI

1. Giới thiệu

Ứng dụng hỗ trợ học sinh trung học phổ thông, giảng viên dạy kỹ năng lái xe an toàn, và cảnh sát giao thông trong việc quản lý đăng ký, tổ chức và thi cấp chứng chỉ kỹ năng lái xe an toàn. Hệ thống đảm bảo minh bạch, hiệu quả và dễ sử dụng.

2. Tính năng chính

2.1. Quản lý thông tin người dùng

- **Học sinh:** Thêm, cập nhật thông tin cá nhân, lớp, trường.
- **Giảng viên:** Quản lý danh sách khóa học, học sinh tham gia, và kết quả học tập.
- **Cảnh sát giao thông:** Theo dõi và phê duyệt các kỳ thi, chứng chỉ cấp cho học sinh.

2.2. Đăng ký tham gia khóa học và thi

- Học sinh đăng ký tham gia khóa học và kỳ thi trực tuyến.
- Giảng viên xác nhận danh sách tham gia.
- Cảnh sát giao thông theo dõi tình trạng phê duyệt chứng chỉ.

2.3. Quản lý khóa học

- Giảng viên tạo và quản lý khóa học: thông tin giáo viên, lịch học, và nội dung khóa học.
- Theo dõi tiến trình và kết quả học tập của học sinh.

2.4. Quản lý thi và cấp chứng chỉ

- Lập danh sách thi: ngày thi, phòng thi, phân công giám sát.
- Ghi nhận điểm số và trạng thái đạt chứng chỉ.
- Cấp chứng chỉ cho học sinh đã hoàn thành kỳ thi.

2.5. Thông báo và giao tiếp

- Gửi thông báo tự động về lịch học, lịch thi, kết quả, và trạng thái chứng chỉ.
- Hỗ trợ giao tiếp giữa học sinh, giảng viên và cảnh sát giao thông.

2.6. Báo cáo và thống kê

- Báo cáo số lượng học sinh đăng ký, tỷ lệ tham gia và đạt chứng chỉ.
- Thống kê kết quả học và thi theo lớp/trường/khóa học.

3. Vai trò và chức năng người dùng

3.1. Học sinh

- Đăng ký khóa học và thi trực tuyến.
- Theo dõi tiến trình học và kết quả thi.
- Nhận chứng chỉ sau khi đạt yêu cầu.

3.2. Giảng viên

- Tạo và quản lý khóa học.
- Ghi nhận điểm số, đánh giá kỹ năng học sinh.
- Phối hợp với cảnh sát giao thông trong quá trình tổ chức thi.

3.3. Cảnh sát giao thông

- Xác nhận và giám sát kỳ thi.
- Quản lý việc cấp chứng chỉ kỹ năng lái xe an toàn.
- Đảm bảo tuân thủ quy định pháp luật.

3.4. Quản trị viên hệ thống

- Quản lý tài khoản người dùng (học sinh, giảng viên, cảnh sát giao thông).
- Cấu hình hệ thống và bảo mật.

GỢI Ý THIẾT KẾ CSDL:

1. Các bảng trong hệ thống

1.1. Bảng Users

- Lưu thông tin tất cả người dùng (học sinh, giảng viên, cảnh sát giao thông).
- Cột:
 - UserID (INT, PRIMARY KEY)
 - FullName (VARCHAR(100), NOT NULL)
 - Email (VARCHAR(100), NOT NULL, UNIQUE)
 - Password (VARCHAR(255), NOT NULL)
 - Role (ENUM('Student', 'Teacher', 'TrafficPolice'), NOT NULL)
 - Class (VARCHAR(50), NULL - chỉ dành cho học sinh)
 - School (VARCHAR(100), NULL - chỉ dành cho học sinh)
 - Phone (VARCHAR(15), NULL).

1.2. Bảng Courses

- Quản lý thông tin khóa học.
- **Cột:**
 - CourseID (INT, PRIMARY KEY)
 - CourseName (VARCHAR(100), NOT NULL)
 - TeacherID (INT, FOREIGN KEY REFERENCES Users(UserID))
 - StartDate (DATE, NOT NULL)
 - EndDate (DATE, NOT NULL).

1.3. Bảng Registrations

- Quản lý đăng ký của học sinh vào khóa học/kỳ thi.
- **Cột:**
 - RegistrationID (INT, PRIMARY KEY)
 - UserID (INT, FOREIGN KEY REFERENCES Users(UserID))
 - CourseID (INT, FOREIGN KEY REFERENCES Courses(CourseID))
 - Status (ENUM('Pending', 'Approved', 'Rejected'), DEFAULT 'Pending')
 - Comments (TEXT, NULL).

1.4. Bảng Exams

- Lưu thông tin kỳ thi.
- **Cột:**
 - ExamID (INT, PRIMARY KEY)
 - CourseID (INT, FOREIGN KEY REFERENCES Courses(CourseID))
 - Date (DATE, NOT NULL)
 - Room (VARCHAR(50), NOT NULL).

1.5. Bảng Results

- Kết quả thi.
- **Cột:**
 - ResultID (INT, PRIMARY KEY)
 - ExamID (INT, FOREIGN KEY REFERENCES Exams(ExamID))
 - UserID (INT, FOREIGN KEY REFERENCES Users(UserID))
 - Score (DECIMAL(5, 2), NOT NULL)
 - PassStatus (BOOLEAN, NOT NULL).

1.6. Bảng Certificates

- Quản lý chứng chỉ đã cấp.
- **Cột:**
 - CertificateID (INT, PRIMARY KEY)
 - UserID (INT, FOREIGN KEY REFERENCES Users(UserID))
 - IssuedDate (DATE, NOT NULL)
 - ExpirationDate (DATE, NOT NULL)

- CertificateCode (VARCHAR(50), UNIQUE).

2. Câu lệnh SQL tạo bảng

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY AUTO_INCREMENT,  
    FullName VARCHAR(100) NOT NULL,  
    Email VARCHAR(100) NOT NULL UNIQUE,  
    Password VARCHAR(255) NOT NULL,  
    Role ENUM('Student', 'Teacher', 'TrafficPolice') NOT NULL,  
    Class VARCHAR(50),  
    School VARCHAR(100),  
    Phone VARCHAR(15)  
);
```

```
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY AUTO_INCREMENT,  
    CourseName VARCHAR(100) NOT NULL,  
    TeacherID INT NOT NULL,  
    StartDate DATE NOT NULL,  
    EndDate DATE NOT NULL,  
    FOREIGN KEY (TeacherID) REFERENCES Users(UserID)  
);
```

```
CREATE TABLE Registrations (  
    RegistrationID INT PRIMARY KEY AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    CourseID INT NOT NULL,  
    Status ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending',  
    Comments TEXT,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID),  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```

```
CREATE TABLE Exams (  
    ExamID INT PRIMARY KEY AUTO_INCREMENT,  
    CourseID INT NOT NULL,  
    Date DATE NOT NULL,  
    Room VARCHAR(50) NOT NULL,  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```

```
CREATE TABLE Results (  
    ResultID INT PRIMARY KEY AUTO_INCREMENT,  
    ExamID INT NOT NULL,  
    UserID INT NOT NULL,  
    Score DECIMAL(5, 2) NOT NULL,  
    PassStatus BOOLEAN NOT NULL,  
    FOREIGN KEY (ExamID) REFERENCES Exams(ExamID),  
    FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);
```

```
CREATE TABLE Certificates (  
    CertificateID INT PRIMARY KEY AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    IssuedDate DATE NOT NULL,  
    ExpirationDate DATE NOT NULL,  
    CertificateCode VARCHAR(50) UNIQUE,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);
```

```
CREATE TABLE Notifications (  
    NotificationID INT PRIMARY KEY AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    Message TEXT NOT NULL,  
    SentDate DATETIME DEFAULT CURRENT_TIMESTAMP,  
    IsRead BOOLEAN DEFAULT FALSE,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);
```