**Báo Cáo**

**Đề Tài Quản Lí Sách**

Giới thiệu thành viên:

Phan Khánh Vương - 2312802

Nguyễn Ngọc Trường Dân - 2312590

Vũ Thế Huỳnh - 2312639

Phùng Nguyễn Hoài Bo - 2312585

Ngô Văn Chương - 2312588

**I. Tìm Hiểu Ngôn Ngữ C#**

**1. Giới thiệu ngôn ngữ C#**

C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft. Nó là một phần của nền tảng .NET Framework và được sử dụng để xây dựng các ứng dụng Windows, web, di động và máy tính đám mây.

C# là một ngôn ngữ mạnh mẽ và linh hoạt, được thiết kế để dễ học và sử dụng. Nó hỗ trợ nhiều tính năng hiện đại, chẳng hạn như lập trình hướng đối tượng, kiểm tra kiểu động, và biên dịch thời gian chạy.

2. Nhập và xuất dữ liệu

⦁ Trong C#, có hai phương thức phổ biến để nhập dữ liệu từ console:

⦁ Console.Read()`: Phương thức này đọc một ký tự từ console.

⦁ Console.ReadLine()`: Phương thức này đọc một chuỗi ký tự từ console.

Ví dụ:

// Nhập một ký tự

char c = Console.Read();

// Nhập một chuỗi ký tự

string str = Console.ReadLine();

⦁ Có hai phương thức phổ biến để xuất dữ liệu ra console:

⦁ Console.Write(): Phương thức này in một ký tự ra console.

⦁ Console.WriteLine(): Phương thức này in một chuỗi ký tự ra console và đưa con trỏ xuống dòng.

Ví dụ:

// In một ký tự

Console.Write("Ký tự: ");

Console.Write(c);

// In một chuỗi ký tự

Console.WriteLine("Chuỗi ký tự: ");

Console.WriteLine(str);

**3. Kiểu dữ liệu**

⦁ C# cung cấp nhiều kiểu dữ liệu như int, float, double, string, và nhiều kiểu dữ liệu tự định nghĩa.

⦁ Việc sử dụng kiểu dữ liệu phụ thuộc vào yêu cầu cụ thể của biến.

**4. Câu lệnh điều khiển**

⦁ C# cung cấp nhiều câu lệnh điều khiển khác nhau, bao gồm:

⦁ C# hỗ trợ nhiều câu lệnh điều khiển như if, else, switch, for, while, do-while.

⦁ Câu lệnh điều khiển giúp kiểm soát luồng thực thi của chương trình.

**5. Hàm**

Trong C#, hàm là một khối mã được sử dụng để thực hiện một tác vụ cụ thể. Hàm có thể được khai báo trong một class, struct, hoặc namespace.

Cấu trúc của hàm trong C# như sau:

[Từ khóa] [Kiểu dữ liệu trả về] [Tên hàm] ([Danh sách tham số])

{

// Khối lệnh xử lý của hàm

}

Trong đó:

⦁ [Từ khóa] là các từ khóa tùy chọn, có thể sử dụng để chỉ định các thuộc tính của hàm, chẳng hạn như:

⦁ static : Hàm tĩnh, có thể được gọi mà không cần tạo đối tượng của class chứa hàm.

⦁ public : Hàm công khai, có thể được gọi từ bất kỳ vị trí nào trong chương trình.

⦁ private : Hàm riêng tư, chỉ có thể được gọi từ bên trong class chứa hàm.

⦁ [Kiểu dữ liệu trả về] là kiểu dữ liệu của giá trị mà hàm trả về. Nếu hàm không trả về giá trị thì sử dụng từ khóa void.

⦁ [Tên hàm] là tên của hàm. Tên hàm phải là một tên hợp lệ theo quy tắc đặt tên trong C#.

⦁ [Danh sách tham số] là danh sách các tham số được truyền vào hàm. Nếu hàm không có tham số thì danh sách tham số là một dấu ngoặc nhọn rỗng (()).

II. Tìm Hiểu Winforms

1. Khái niệm Winforms

Winforms là một thư viện lớp đồ họa (GUI) mã nguồn mở và miễn phí được bao gồm như một phần của Microsoft.NET Framework hoặc Mono Framework, cung cấp nền tảng để viết các ứng dụng khách phong phú cho máy tính để bàn, máy tính xách tay và máy tính bảng.

Winforms là một thư viện lớp đồ họa cung cấp một bộ các điều khiển đồ họa sẵn có, chẳng hạn như nút, hộp văn bản, hộp danh sách, v.v., có thể được sử dụng để tạo giao diện người dùng cho ứng dụng. Winforms cũng cung cấp một số tính năng nâng cao, chẳng hạn như hỗ trợ ràng buộc dữ liệu và hoạt ảnh.

Dưới đây là một số khái niệm và đặc điểm chính của WinForms:

⦁ Form (Biểu mẫu): Là cửa sổ chính của ứng dụng, nơi các điều khiển và thành phần được đặt để tạo giao diện người dùng.

⦁ Controls (Điều khiển): Các thành phần như nút, ô nhập liệu, danh sách thả xuống, v.v., được sử dụng để xây dựng giao diện người dùng của ứng dụng.

⦁ Event-driven programming (Lập trình sự kiện): WinForms sử dụng mô hình lập trình dựa trên sự kiện, tức là các hành động như nhấn nút, di chuyển chuột sẽ kích hoạt các sự kiện để xử lý.

⦁ Threading (Luồng): WinForms hỗ trợ việc quản lý các luồng để xử lý công việc đồng thời, giúp tránh tình trạng đóng băng giao diện người dùng trong quá trình thực hiện công việc lâu dài.

⦁ Data Binding (Ràng buộc dữ liệu): Cho phép liên kết dữ liệu giữa các điều khiển và nguồn dữ liệu, giúp cập nhật tự động khi dữ liệu thay đổi.

⦁ Visual Studio: Môi trường phát triển tích hợp (IDE) chính được sử dụng để phát triển ứng dụng WinForms, thường được sử dụng với ngôn ngữ lập trình C# hoặc VB.NET.

**II. Đặc Điểm Chính:**

**1.** **Giao Diện Người Dùng (GUI) Trực Quan:**

⦁ WinForms cung cấp một loạt các controls như buttons, textboxes, labels, và grids giúp tạo ra giao diện người dùng trực quan và dễ sử dụng.

**2.** **Sự Linh Hoạt và Dễ Sử Dụng:**

⦁ Thiết kế giao diện trực tiếp trong Visual Studio, với việc kéo và thả controls, làm cho việc phát triển ứng dụng trở nên dễ dàng và nhanh chóng.

**3.** **Sự Tích Hợp với .NET Framework:**

⦁ WinForms tích hợp chặt chẽ với .NET Framework, có thể sử dụng các tính năng của .NET như quản lý bộ nhớ tự động, xử lý ngoại lệ, và thư viện lớp.

**4.** **Sự Mở Rộng và Tùy Chọn:**

⦁ Có thể tạo các ứng dụng đơn giản hoặc phức tạp tùy thuộc vào yêu cầu của dự án.

⦁ Hỗ trợ tạo các control tùy chỉnh và linh hoạt trong việc xử lý sự kiện.

**5.** **Hỗ Trợ Đa Ngôn Ngữ và Đa Thiết Bị:**

⦁ WinForms hỗ trợ quốc tế hóa, giúp dễ dàng chuyển đổi và hiển thị ứng dụng trên nhiều ngôn ngữ và thiết bị khác nhau.

**III. Các Control Phổ Biến trong WinForms:**

**1.** **Form:**

⦁ Là cửa sổ chính của ứng dụng, nơi chứa các controls khác.

**2.** **Button:**

⦁ Dùng để thực hiện các hành động khi được nhấn.

**3.** **TextBox:**

⦁ Cho phép người dùng nhập văn bản hoặc số liệu.

**4.** **Label:**

⦁ Hiển thị văn bản không thể chỉnh sửa, thường được sử dụng để hiển thị thông tin.

**5.** **DataGridView:**

⦁ Cho phép hiển thị và chỉnh sửa dữ liệu dưới dạng bảng.

**6.** **ComboBox và ListBox:**

⦁ Cho phép người dùng chọn một hoặc nhiều mục từ danh sách.

**7.** **MenuStrip và ToolStrip:**

⦁ Cung cấp thanh menu và thanh công cụ để tổ chức và thực hiện các chức năng của ứng dụng.

**IV. Sự Phát Triển và Tiện Ích:**

**1.** **Visual Studio:**

⦁ WinForms được hỗ trợ mạnh mẽ trong Visual Studio, môi trường phát triển tích hợp của Microsoft.

**2.** **Thư Viện Bên Thứ Ba:**

⦁ Cộng đồng phát triển đã tạo ra nhiều thư viện bên thứ ba mở rộng chức năng và kiểu dáng cho WinForms.

**V. Nhược Điểm và Tương Lai:**

**1.** **Giao Diện Cổ Điển:**

⦁ Giao diện của WinForms có vẻ cổ điển so với các công nghệ giao diện hiện đại hơn.

**2.** **Khả Năng Tích Hợp với Công Nghệ Mới:**

⦁ Trong khi WinForms vẫn được sử dụng rộng rãi, nó không tích hợp tự nhiên với các công nghệ mới như Blazor hay MAUI của Microsoft.

**III. Tìm Hiểu SQL Server**

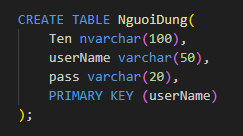
SQL Server là một hệ quản trị cơ sở dữ liệu của công ty Microsoft, dùng để thiết kế dữ liệu bằng cách sử dụng các bảng (Tables), các mối quan hệ (Relationships), các truy vấn (Query), thủ tục (Store procedure), hàm (Function) và bẩy sự kiện (Trigger). Giống như những hệ quản trị khác (Oracle, My SQL, DB2, MongoDB…), khi làm việc với SQL Server, người dùng phải thực hiện việc kết nối bằng câu lệnh truy vấn.

Một số thao tác thường dùng trên hệ quản trị cơ sở dữ liệu SQL Server như sau:

⦁ Tạo Cơ sở dữ liệu: Click phải lên Databases, chọn New Database, đặt tên và nhấn OK.



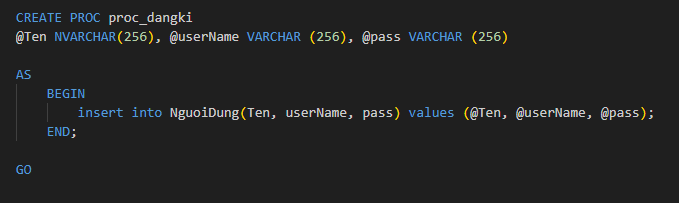
⦁ Tạo bảng: Vào Database, click phải lên Table, chọn New, chọn Table: Đặt tên cột, kiểu dữ liệu, chọn cho phép Null hay không, nhấn lưu để đặt tên cho bảng.



⦁

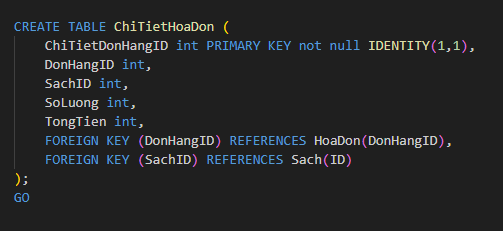
⦁ Tạo truy vấn: Click phảo lên Database, chọn New Query (hoặc chọn New Query trên thanh công cụ).

⦁ Tạo ràng buộc: Click phải lên Database Diagrams, chọn New Database Diagram, thiết lập các mối quan hệ, sau đó lưu lại.

⦁ Tạo thủ tục (Store Procedure): Vào Programmability, click phải lên Store Procedures chọn Store Procedure... để viết các thủ tụ

⦁

⦁ Tạo hàm (Function): Vào Programmability, click phải lên Function chọn New, chọn Scalar-valued Functions để viết các hàm.



⦁ Sao lưu và phục hồi cơ sở dữ liệu: Click phải lên Database, chọn Task, chọn Backup hoặc Restore.

II. Đặc điểm chính:

**1.** **Quản lý Dữ liệu:**

⦁ SQL Server hỗ trợ lưu trữ và quản lý dữ liệu quan hệ dưới dạng bảng.

⦁ Dữ liệu được tổ chức thành các hàng và cột, giúp dễ dàng tìm kiếm và truy xuất.

**2.** **Ngôn ngữ Truy vấn SQL:**

⦁ SQL Server sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để tương tác với cơ sở dữ liệu.

⦁ Ngôn ngữ này cung cấp các lệnh như SELECT, INSERT, UPDATE, DELETE để thực hiện các thao tác trên dữ liệu.

**3.** **Bảo mật và Quản lý Quyền:**

⦁ SQL Server cung cấp các chức năng bảo mật mạnh mẽ, bao gồm quản lý quyền và phân quyền người dùng.

⦁ Admin có thể kiểm soát quyền truy cập, sửa đổi và xóa dữ liệu.

**4.** **Khả năng Mở rộng:**

⦁ SQL Server hỗ trợ khả năng mở rộng bằng cách tăng kích thước cơ sở dữ liệu và số lượng người dùng.

⦁ Các tính năng như phân vùng và đám mây giúp nâng cao khả năng mở rộng.

**5.** **Tích Hợp với Công Nghệ Microsoft:**

**⦁** SQL Server tương thích chặt chẽ với nền tảng Microsoft, đặc biệt là .NET Framework.

⦁ Có thể tích hợp dễ dàng với các ứng dụng và dịch vụ khác của Microsoft.

**6.** **Dự trữ và Phục hồi Dữ liệu:**

⦁ Cung cấp các phương tiện để sao lưu và khôi phục dữ liệu, bảo vệ dữ liệu khỏi mất mát và tổn thất.

**IV. Mô tả chức năng**

**2. Đăng nhập:**

⦁ Người dùng cung cấp thông tin đăng nhập (tên đăng nhập, mật khẩu).

⦁ Hệ thống kiểm tra thông tin với cơ sở dữ liệu SQL Server.

⦁ Nếu thông tin chính xác, người dùng được chuyển đến giao diện chính của ứng dụng.

⦁ Đăng ký:

⦁ Người dùng mới chọn chức năng đăng ký.

⦁ Nhập thông tin cá nhân (tên, email, tên đăng nhập, mật khẩu).

⦁ Hệ thống kiểm tra tính hợp lệ và tạo tài khoản mới.

**2.2. Quản lý Sách**

⦁ Thêm sách:

⦁ Người dùng nhập thông tin sách mới vào giao diện.

⦁ Dữ liệu mới được thêm vào cơ sở dữ liệu SQL Server.

⦁ Xóa sách:

⦁ Người dùng chọn một cuốn sách trong danh sách và xác nhận xóa.

⦁ Dữ liệu liên quan sẽ được cập nhật trong cơ sở dữ liệu.

⦁ Sửa thông tin sách:

⦁ Người dùng chọn một cuốn sách và cập nhật thông tin.

⦁ Dữ liệu được lưu vào cơ sở dữ liệu.

**2.3. Quản lý Hóa đơn và Chi tiết Hóa đơn**

⦁ Bán sách:

⦁ Người dùng chọn cuốn sách và nhập số lượng cần bán.

⦁ Hệ thống tạo một hóa đơn mới và cập nhật chi tiết hóa đơn.

⦁ Hóa đơn và Chi tiết Hóa đơn:

⦁ Mỗi giao dịch bán sách tạo một hóa đơn mới.

⦁ Chi tiết hóa đơn liên kết với sách, số lượng, đơn giá và thành tiền.

**2.4. Biểu đồ Thống kê**

⦁ Thống kê Sách:

⦁ Tạo biểu đồ thống kê số lượng sách theo thể loại.

⦁ Biểu đồ giúp người quản lý dễ dàng theo dõi xu hướng và phân loại sách.

⦁ Thống kê Doanh thu:

⦁ Tạo biểu đồ thống kê doanh thu theo tháng.

⦁ Biểu đồ giúp đánh giá hiệu suất kinh doanh và dự đoán xu hướng tương lai.

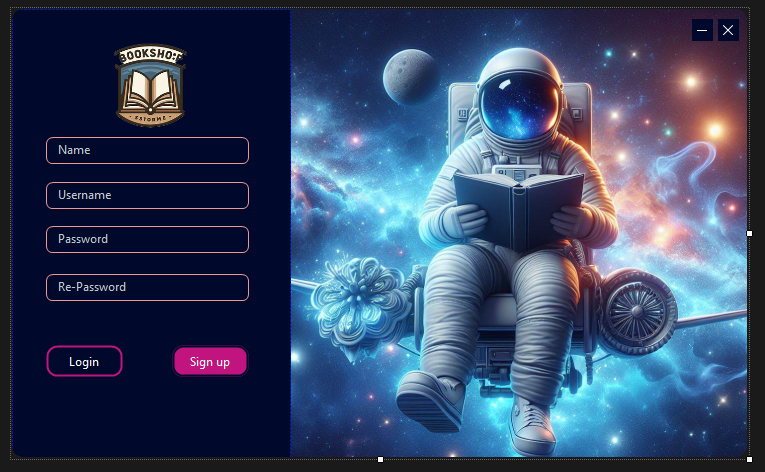
**BÁO CÁO CẬP NHẬT CHƯƠNG TRÌNH QUẢN LÝ SÁCH VỚI CHỨC NĂNG ĐĂNG KÝ**

Ngày: [Ngày/Tháng/Năm]

**I. Giới thiệu:** Chương trình quản lý sách đã được bổ sung thêm chức năng đăng ký, nhằm mở rộng quyền truy cập và quản lý người dùng. Điều này giúp tăng cường bảo mật và quản lý thông tin người dùng một cách hiệu quả.

**II. Chức năng mới:**

**1.** **Đăng ký:**



⦁ Người dùng mới có thể chọn chức năng "Đăng ký" từ giao diện đăng nhập.

⦁ Họ sẽ nhập thông tin cần thiết như tên, email, tên đăng nhập và mật khẩu.

⦁ Hệ thống sẽ kiểm tra thông tin và tạo một tài khoản mới nếu thông tin là hợp lệ.

**3.** **Xác nhận đăng ký:**

⦁ Sau khi đăng ký, người dùng sẽ nhận được một email xác nhận hoặc mã xác nhận để kích hoạt tài khoản.

⦁ Mục này được thêm vào để bảo đảm tính bảo mật và xác thực người dùng.

**4.** **Quản lý thông tin người dùng:**

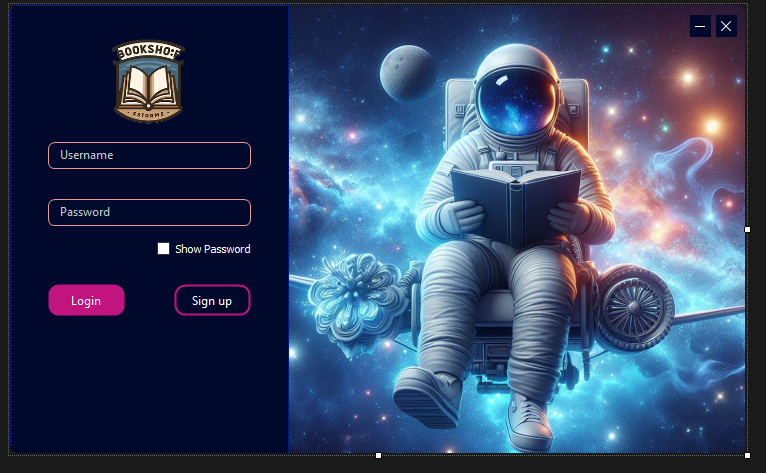
⦁ Người dùng có thể truy cập thông tin cá nhân, thay đổi mật khẩu và cập nhật thông tin cá nhân từ giao diện người dùng.

III. Kết nối với SQL Server:

⦁ Bảng dữ liệu mới được thêm vào cơ sở dữ liệu để lưu trữ thông tin người dùng, bao gồm tên, email, tên đăng nhập và mật khẩu đã được mã hóa.

⦁ Các truy vấn SQL đã được cập nhật để thêm, xác nhận và quản lý thông tin người dùng.

**IV. Giao diện người dùng:**



⦁ Giao diện đăng nhập đã được cập nhật để bao gồm chức năng đăng ký.

⦁ Giao diện quản lý người dùng được thêm vào để người dùng có thể quản lý thông tin cá nhân của họ.

**V. Mở rộng và Tính năng tương lai:**

⦁ Cải thiện chức năng đăng ký và xác nhận để tối ưu hóa trải nghiệm người dùng.

⦁ Thêm tính năng quên mật khẩu để người dùng có thể đặt lại mật khẩu nếu cần.

**Phần 3: Kết luận**

Chương trình quản lý sách kết hợp sự linh hoạt của ngôn ngữ C# và tính tiện ích của giao diện WinForms cùng với khả năng quản lý dữ liệu của SQL Server. Việc tích hợp chức năng đăng nhập, đăng ký, quản lý sách, hóa đơn và thống kê tạo ra một ứng dụng mạnh mẽ, linh hoạt và dễ sử dụng. Qua báo cáo này, chúng ta đã đi sâu vào cả hai khía cạnh này, cung cấp cái nhìn tổng quan vững chắc về lập trình C# và phát triển phần mềm sử dụng WinForms và SQL Server.