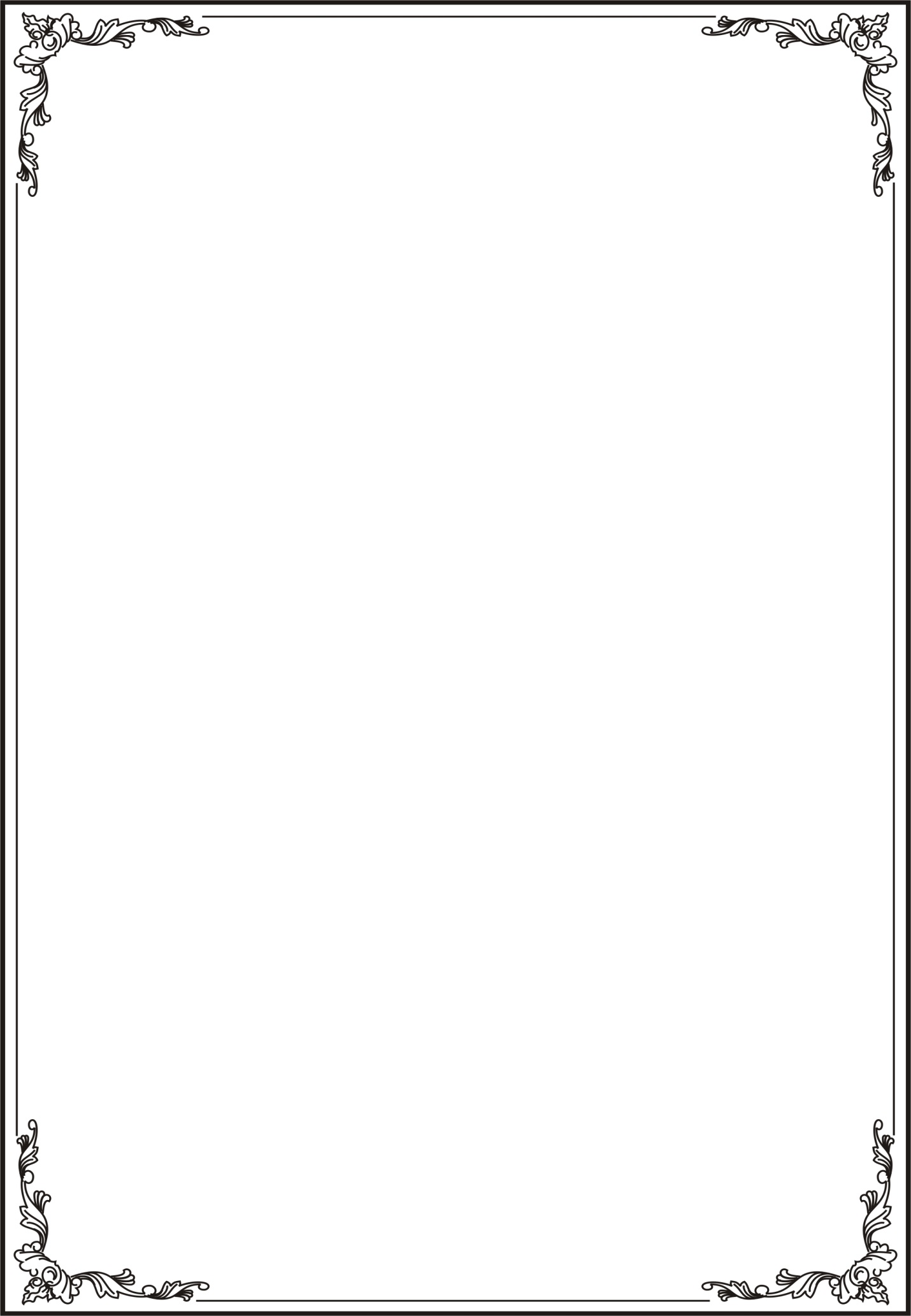
****

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN LẬP TRÌNH.NET**

**XÂY DỰNG PHẦN MỀM**

**QUẢN LÝ CỬA HÀNG QUẦN ÁO NEM**

**GVHD:Ths.Nguyễn Thị Mỹ Truyền**

**Sinh viên thực hiện:**

**Nguyễn Thị Huyền Trân - DTH235791 - DH24TH3**

**Nguyễn Thị Như Ý - DTH235824 - DH24TH3**

**Lê Thị Mỹ Tiên - DTH235781 - DH24TH3**

**AN GIANG, tháng 11 năm 2025**

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn Cô Mỹ Truyền Khoa Công nghệ Thông tin – Trường Đại học An Giang, đã tận tình giảng dạy và hướng dẫn em trong quá trình học tập cũng như thực hiện đồ án này.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến Cô khi đã tận tình chỉ bảo, định hướng và giúp em hoàn thiện đề tài **“Xây dựng ứng dụng Quản lý cửa hàng quần áo bằng C# .NET và SQL Server”**. Sự hướng dẫn của Cô đã giúp em hiểu rõ hơn về cách thiết kế và lập trình ứng dụng quản lý, từ việc tạo cơ sở dữ liệu, xử lý dữ liệu đến xây dựng giao diện người dùng thân thiện.

Do thời gian thực hiện đồ án có hạn, báo cáo này khó tránh khỏi những thiếu sót. Em rất mong nhận được ý kiến đóng góp quý báu của Cô để bài làm của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Nội dung

[**CHƯƠNG 1: THÔNG TIN TỔNG QUAN 4**](#_Toc214901845)

[***I. Mục đích yêu cầu của đồ án* 4**](#_Toc214901846)

[***II. Giới thiệu đồ án* 4**](#_Toc214901847)

[**1.Thông tin tổng quan 4**](#_Toc214901848)

[**2. Giới hạn phạm vi đồ án 4**](#_Toc214901849)

[**3. Đối tượng sử dụng 5**](#_Toc214901850)

[**CHƯƠNG 2: PHÂN TÍCH HIỆN TRẠNG VÀ YÊU CẦU 5**](#_Toc214901851)

[***I. Phân tích hiện trạng* 5**](#_Toc214901852)

[***II. Yêu cầu hệ thống* 5**](#_Toc214901853)

[***III. Bảng yêu cầu chức năng hệ thống* 6**](#_Toc214901854)

[***IV. Sơ đồ luồng dữ liệu (DFD) cấp 0* 8**](#_Toc214901855)

[***V. Sơ đồ chức năng (Functional Diagram)* 9**](#_Toc214901856)

[***VI.Cơ sở dữ liệu* 10**](#_Toc214901857)

[**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN 11**](#_Toc214901858)

[***I. Form Đăng Nhập* 11**](#_Toc214901859)

[***II. Form Main* 14**](#_Toc214901860)

[***III. Form Quản Lý Sản Phẩm* 20**](#_Toc214901861)

[***IV. Form Quản Lý Khách Hàng* 29**](#_Toc214901862)

[***V. Form Hóa Đơn* 38**](#_Toc214901863)

[***VI. Form Quản Lý Doanh Thu* 49**](#_Toc214901864)

[***VII. Form Quản Lý Nhập Hàng* 59**](#_Toc214901865)

[***VIII. Form Quản Lý Kho* 68**](#_Toc214901866)

[**CHƯƠNG 4: KẾT LUẬN VÀ TÀI LIỆU THAM KHẢO 81**](#_Toc214901867)

[***I. Kết luận* 81**](#_Toc214901868)

## 

## **CHƯƠNG 1: THÔNG TIN TỔNG QUAN**

## ***I. Mục đích yêu cầu của đồ án***

**1.Mục đích:**

Xây dựng hệ thống quản lý cửa hàng quần áo NEM giúp quản lý **sản phẩm, khách hàng, nhân viên, hóa đơn** một cách nhanh chóng và chính xác.

Hỗ trợ việc **tra cứu, thống kê, báo cáo doanh thu** và quản lý kho hàng hiệu quả.

Giúp sinh viên nắm vững **thiết kế CSDL, lập trình .NET, giao diện Windows Forms, kết nối SQL Server**.

**2.Yêu cầu:**

Ứng dụng phải có **giao diện thân thiện**, dễ sử dụng.

Chức năng đầy đủ: **Quản lý sản phẩm, khách hàng, nhân viên, hóa đơn, thống kê doanh thu**.

Dữ liệu phải được **lưu trữ và truy xuất từ SQL Server**.

Hỗ trợ **thêm, sửa, xóa, tìm kiếm và xuất báo cáo**.

## ***II. Giới thiệu đồ án***

### 1.Thông tin tổng quan

Tên đồ án: **Xây dựng ứng dụng Quản lý cửa hàng quần áo NEM bằng C# .NET và SQL Server**

Loại đồ án: **Ứng dụng Windows Form**

Ngôn ngữ lập trình: **C# .NET (Windows Forms)**

Hệ quản trị CSDL: **SQL Server**

### 2. Giới hạn phạm vi đồ án

Chỉ quản lý **sản phẩm, khách hàng, hóa đơn bán hàng, kho, nhập hàng, doanh thu**.

Chưa tích hợp **thanh toán online hay quản lý chuỗi cửa hàng**.

Chưa có chức năng **quản lý tồn kho nâng cao, báo cáo phân tích thống kê nâng cao**.

## 3. Đối tượng sử dụng

**Ai có thể sử dụng:**

Chủ cửa hàng quần áo NEM và nhân viên bán hàng.

**Mô tả hệ thống:**

Hệ thống quản lý trên máy tính, giúp **theo dõi thông tin khách hàng, sản phẩm, kho, hóa đơn**, **nhập hàng** và **doanh thu**.

Dễ dàng **cập nhật dữ liệu, tìm kiếm nhanh, báo cáo tổng hợp**.

## CHƯƠNG 2: PHÂN TÍCH HIỆN TRẠNG VÀ YÊU CẦU

### ***I. Phân tích hiện trạng***

Qua khảo sát thực tế, nhiều cửa hàng quần áo hiện nay vẫn quản lý **sản phẩm, hóa đơn, khách hàng và kho hàng** theo phương pháp thủ công hoặc sử dụng Excel. Điều này dẫn đến một số hạn chế sau:

**Khó đồng bộ dữ liệu:** Thông tin tồn kho, sản phẩm và doanh thu thường bị rời rạc, gây khó khăn trong việc kiểm tra và cập nhật chính xác.

**Tốc độ tra cứu chậm:** Việc tìm kiếm thông tin sản phẩm, hóa đơn hay khách hàng mất nhiều thời gian, đặc biệt khi số lượng dữ liệu lớn.

**Dễ xảy ra sai sót:** Nhập liệu thủ công có thể dẫn đến lỗi số liệu, nhầm lẫn trong thống kê tồn kho và doanh thu.

**Thiếu thống kê tổng quan:** Các báo cáo doanh thu, nhập hàng, xuất kho và sản phẩm bán chạy không được cập nhật kịp thời, gây khó khăn trong việc ra quyết định kinh doanh.

Những hạn chế trên ảnh hưởng trực tiếp đến **hiệu quả quản lý và hoạt động kinh doanh** của cửa hàng, khiến việc theo dõi kho, doanh thu và quản lý nhập hàng trở nên phức tạp và tốn thời gian.

### ***II. Yêu cầu hệ thống***

Hệ thống quản lý cửa hàng quần áo cần đáp ứng các chức năng cơ bản sau:

1. **Quản lý sản phẩm:**

Lưu trữ thông tin sản phẩm như tên, mã sản phẩm, loại, giá bán.

Theo dõi số lượng tồn kho và tình trạng sản phẩm.

Hỗ trợ cập nhật, sửa đổi và xóa sản phẩm khi cần.

1. **Quản lý hóa đơn bán hàng:**

Lập hóa đơn bán hàng nhanh chóng, chính xác.

Lưu trữ và tra cứu các hóa đơn đã xuất.

Hỗ trợ thống kê doanh thu từ hóa đơn bán hàng.

1. **Quản lý kho:**

Theo dõi số lượng hàng tồn kho theo từng sản phẩm.

Cảnh báo khi tồn kho thấp.

Quản lý xuất – nhập – tồn kho chính xác, hỗ trợ quyết định nhập hàng.

1. **Quản lý khách hàng:**

Lưu trữ thông tin khách hàng: tên, số điện thoại, địa chỉ, lịch sử mua hàng.

Hỗ trợ chăm sóc khách hàng và xây dựng chương trình khách hàng thân thiết.

1. **Quản lý doanh thu:**

Tổng hợp doanh thu theo ngày, tuần, tháng.

Hiển thị các báo cáo doanh thu chi tiết và biểu đồ tổng quan.

Hỗ trợ đánh giá hiệu quả kinh doanh và ra quyết định.

1. **Quản lý nhập hàng:**

Lưu trữ thông tin phiếu nhập hàng từ nhà cung cấp.

Cập nhật số lượng hàng vào kho tự động.

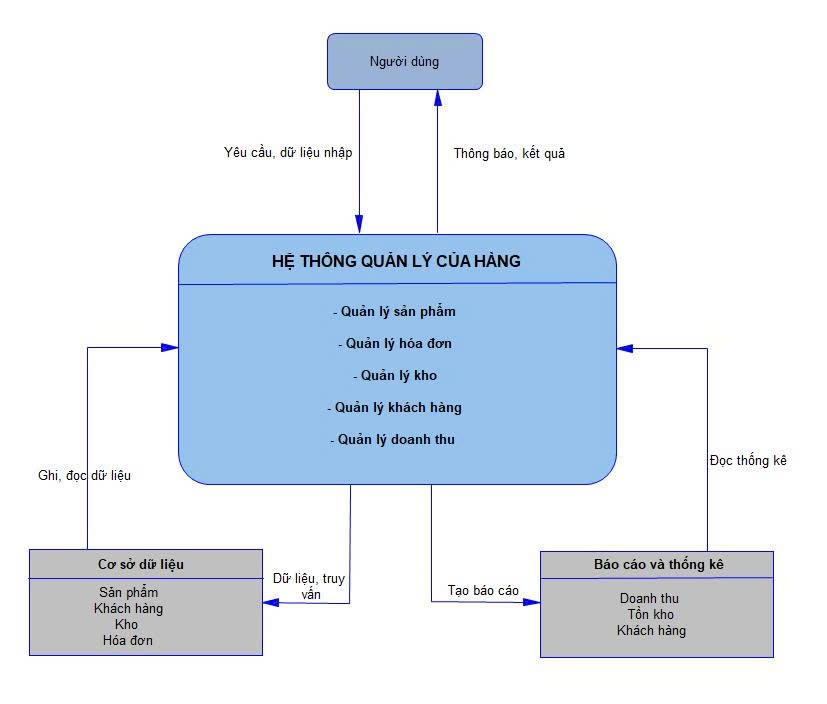
Theo dõi chi phí nhập hàng và tình trạng sản phẩm nhập.

Hệ thống cần được thiết kế **dễ sử dụng, ổn định và có khả năng mở rộng**, đáp ứng đầy đủ nhu cầu quản lý hàng ngày của cửa hàng quần áo vừa và nhỏ.

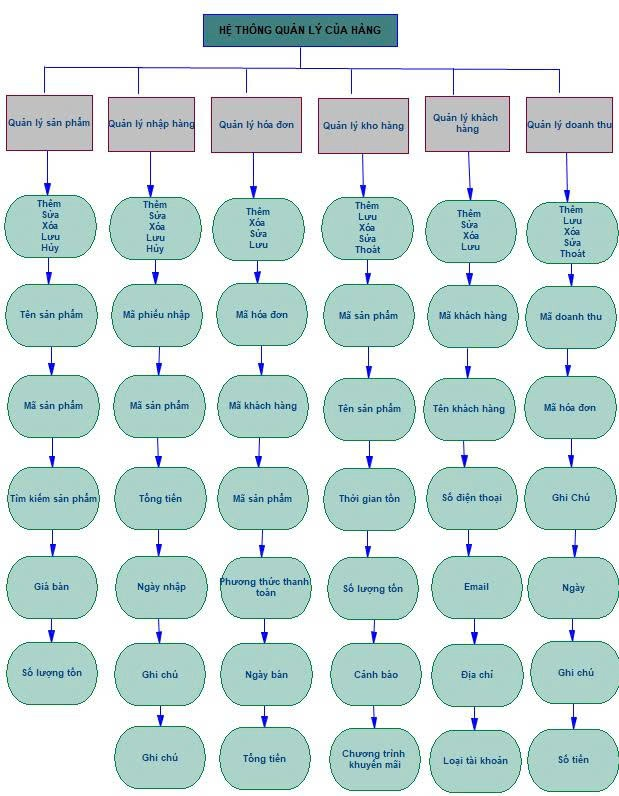
### ***III. Bảng yêu cầu chức năng hệ thống***

| **STT** | **Chức năng** | **Mô tả chi tiết** | **Kết quả/Đầu ra** |
| --- | --- | --- | --- |
| 1 | Quản lý sản phẩm | Thêm, sửa, xóa, tra cứu sản phẩm; lưu trữ thông tin sản phẩm như mã, tên, loại, giá bán, tình trạng tồn kho | Danh sách sản phẩm được cập nhật, tra cứu nhanh |
| 2 | Quản lý hóa đơn bán hàng | Lập hóa đơn bán hàng, lưu trữ, tra cứu, xuất báo cáo doanh thu từ hóa đơn | Hóa đơn được lưu trữ đầy đủ, thống kê doanh thu chính xác |
| 3 | Quản lý kho | Theo dõi số lượng tồn kho, cập nhật khi xuất/nhập hàng; cảnh báo khi tồn kho thấp | Kho hàng được kiểm soát, tránh thiếu hoặc thừa hàng |
| 4 | Quản lý khách hàng | Lưu trữ thông tin khách hàng, lịch sử mua hàng; hỗ trợ chăm sóc khách hàng và chương trình khách hàng thân thiết | Danh sách khách hàng chi tiết, dễ tra cứu và quản lý |
| 5 | Quản lý doanh thu | Tổng hợp doanh thu theo ngày, tuần, tháng; hiển thị báo cáo và biểu đồ tổng quan | Thống kê doanh thu chính xác, hỗ trợ ra quyết định kinh doanh |
| 6 | Quản lý nhập hàng | Lưu trữ phiếu nhập hàng, cập nhật kho tự động, theo dõi chi phí nhập | Kho hàng được cập nhật, chi phí nhập rõ ràng, sản phẩm sẵn sàng bán |

### ***IV. Sơ đồ luồng dữ liệu (DFD) cấp 0***



### ***V. Sơ đồ chức năng (Functional Diagram)***



# ***VI.Cơ sở dữ liệu***

Ảnh có chứa văn bản, biểu đồ, bản phác thảo, Kế hoạch

Mô tả được tạo tự động

# CHƯƠNG 3: THIẾT KẾ GIAO DIỆN

### ***I. Form Đăng Nhập***

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Code của giao diện:

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 DoAnMonHoc

{

public partial class frmDangNhap : Form

{

public frmDangNhap()

{

InitializeComponent();

}

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

{

string username = txtTen.Text.Trim();

string password = txtMatKhau.Text.Trim();

if (string.IsNullOrEmpty(username))

{

MessageBox.Show("Tên đăng nhập không được để trống!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTen.Focus();

return;

}

if (string.IsNullOrEmpty(password))

{

MessageBox.Show("Mật khẩu không được để trống!", "Cảnh báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMatKhau.Focus();

return;

}

if (username == "admin" && password == "1234")

{

MessageBox.Show("Đăng nhập thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Hide();

frmMain mainForm = new frmMain();

mainForm.Show();

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu!", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

txtTen.Clear();

txtMatKhau.Clear();

txtTen.Focus();

}

}

}

### ***II. Form Main***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phần mềm đa phương tiện, phần mềm

Mô tả được tạo tự động

Code của giao diện:

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 DoAnMonHoc

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

}

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

{

frmKho f = new frmKho();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

Application.Exit();

}

}

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

{

this.Hide(); // Ẩn form hiện tại

frmKho frm = new frmKho();

frm.ShowDialog();

this.Show();

}

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

{

frmDoanhThu f = new frmDoanhThu();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

this.Hide(); // Ẩn form hiện tại

frmDoanhThu frm = new frmDoanhThu();

frm.ShowDialog();

this.Show();

}

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

{

frmHoaDon f = new frmHoaDon();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmNhap f = new frmNhap();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmNhap f = new frmNhap();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmSanPham f = new frmSanPham();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmSanPham f = new frmSanPham();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmKhachHang f = new frmKhachHang();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmKhachHang f = new frmKhachHang();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

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

{

frmHoaDon f = new frmHoaDon();

f.FormClosed += (s, args) => this.Show(); // khi form mới đóng → hiện lại form main

f.Show();

this.Hide();

}

}

}

### ***III. Form Quản Lý Sản Phẩm***

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Code của giao diện:

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 DoAnMonHoc

{

public partial class frmSanPham : Form

{

public frmSanPham()

{

InitializeComponent();

}

DataSet ds = new DataSet("dsQLCH");

SqlDataAdapter daSanPham;

SqlConnection conn;

SqlCommandBuilder cb;

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

{

conn = new SqlConnection(

@"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True");

string query = "SELECT \* FROM SanPham";

daSanPham = new SqlDataAdapter(query, conn);

cb = new SqlCommandBuilder(daSanPham);

daSanPham.Fill(ds, "tblDSSanPham");

dgvSanPham.DataSource = ds.Tables["tblDSSanPham"];

FormatGrid();

}

private void FormatGrid()

{

dgvSanPham.Columns["MaSP"].HeaderText = "Mã Sản Phẩm";

dgvSanPham.Columns["MaSP"].Width = 150;

dgvSanPham.Columns["TenSP"].HeaderText = "Tên Sản Phẩm";

dgvSanPham.Columns["TenSP"].Width = 200;

dgvSanPham.Columns["Gia"].HeaderText = "Giá Bán";

dgvSanPham.Columns["Gia"].Width = 150;

dgvSanPham.Columns["SoLuongTon"].HeaderText = "SL Tồn";

dgvSanPham.Columns["SoLuongTon"].Width = 80;

dgvSanPham.Columns["MauSac"].HeaderText = "Màu Sắc";

dgvSanPham.Columns["MauSac"].Width = 150;

dgvSanPham.Columns["SoSize"].HeaderText = "Size";

dgvSanPham.Columns["SoSize"].Width = 80;

dgvSanPham.Columns.Remove("MaSize");

}

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

{

if (string.IsNullOrWhiteSpace(txtMaSP.Text) ||

string.IsNullOrWhiteSpace(txtTenSP.Text) ||

string.IsNullOrWhiteSpace(txtGiaBan.Text))

{

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

return;

}

DataTable tbl = ds.Tables["tblDSSanPham"];

// kiểm tra trùng mã

bool exists = tbl.AsEnumerable()

.Any(r => r.Field<string>("MaSP")

.Equals(txtMaSP.Text, StringComparison.OrdinalIgnoreCase));

if (exists)

{

MessageBox.Show("Mã sản phẩm đã tồn tại!", "Lỗi");

return;

}

DataRow row = tbl.NewRow();

row["MaSP"] = txtMaSP.Text.Trim();

row["TenSP"] = txtTenSP.Text.Trim();

row["Gia"] = txtGiaBan.Text.Trim();

row["SoLuongTon"] = txtSoLuongTon.Text.Trim();

row["MauSac"] = cboMauSac.Text.Trim();

row["SoSize"] = cboSize.Text.Trim();

tbl.Rows.Add(row);

MessageBox.Show("Đã thêm sản phẩm!", "Thành công");

}

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

{

DataTable tbl = ds.Tables["tblDSSanPham"];

DataRow row = tbl.AsEnumerable()

.FirstOrDefault(r => r["MaSP"].ToString() == txtMaSP.Text);

if (row == null)

{

MessageBox.Show("Không tìm thấy sản phẩm!", "Lỗi");

return;

}

row["TenSP"] = txtTenSP.Text.Trim();

row["Gia"] = txtGiaBan.Text.Trim();

row["SoLuongTon"] = txtSoLuongTon.Text.Trim();

row["MauSac"] = cboMauSac.Text.Trim();

row["SoSize"] = cboSize.Text.Trim();

MessageBox.Show("Đã sửa!", "Thành công");

}

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

{

// Nếu bấm vào header thì bỏ qua

if (e.RowIndex < 0) return;

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

txtMaSP.Text = row.Cells["MaSP"].Value?.ToString();

txtTenSP.Text = row.Cells["TenSP"].Value?.ToString();

txtGiaBan.Text = row.Cells["Gia"].Value?.ToString();

txtSoLuongTon.Text = row.Cells["SoLuongTon"].Value?.ToString();

cboMauSac.Text = row.Cells["MauSac"].Value?.ToString();

cboSize.Text = row.Cells["SoSize"].Value?.ToString();

}

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

{

if (string.IsNullOrWhiteSpace(txtMaSP.Text))

{

MessageBox.Show("Vui lòng chọn sản phẩm!", "Thông báo");

return;

}

if (MessageBox.Show("Bạn có chắc muốn xóa?", "Xác nhận",

MessageBoxButtons.YesNo) == DialogResult.No)

return;

DataTable tbl = ds.Tables["tblDSSanPham"];

DataRow row = tbl.AsEnumerable()

.FirstOrDefault(r => r["MaSP"].ToString() == txtMaSP.Text);

if (row != null)

{

row.Delete();

MessageBox.Show("Đã xóa!", "Thành công");

}

}

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

{

try

{

daSanPham.Update(ds.Tables["tblDSSanPham"]);

MessageBox.Show("Đã lưu xuống CSDL!", "Thành công");

}

catch (Exception ex)

{

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

}

}

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

{

txtMaSP.Clear();

txtTenSP.Clear();

txtGiaBan.Clear();

txtSoLuongTon.Clear();

cboMauSac.SelectedIndex = -1;

cboSize.SelectedIndex = -1;

txtMaSP.Focus();

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // CHỈ đóng form hiện tại

}

}

}

}

### ***IV. Form Quản Lý Khách Hàng***

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Code của giao diện:

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 DoAnMonHoc

{

public partial class frmKhachHang : Form

{

public frmKhachHang()

{

InitializeComponent();

}

SqlConnection conn;

SqlDataAdapter daKhachHang;

SqlCommandBuilder cb;

DataSet ds = new DataSet();

private void FormatGrid()

{

dgvKhachHang.Columns["MaKH"].HeaderText = "Mã KH";

dgvKhachHang.Columns["MaKH"].Width = 70;

dgvKhachHang.Columns["TenKH"].HeaderText = "Tên KH";

dgvKhachHang.Columns["TenKH"].Width = 150;

dgvKhachHang.Columns["SoDienThoai"].HeaderText = "Số ĐT";

dgvKhachHang.Columns["SoDienThoai"].Width = 120;

dgvKhachHang.Columns["Email"].HeaderText = "Email";

dgvKhachHang.Columns["Email"].Width = 150;

dgvKhachHang.Columns["DiaChi"].HeaderText = "Địa chỉ";

dgvKhachHang.Columns["DiaChi"].Width = 120;

dgvKhachHang.Columns["LoaiTaiKhoan"].HeaderText = "Loại TK";

dgvKhachHang.Columns["LoaiTaiKhoan"].Width = 120;

dgvKhachHang.Columns.Remove("MaTK");

}

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

{

conn = new SqlConnection(

@"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True");

string query = "SELECT \* FROM KhachHang";

daKhachHang = new SqlDataAdapter(query, conn);

// Tự động sinh Insert, Update, Delete

cb = new SqlCommandBuilder(daKhachHang);

daKhachHang.Fill(ds, "tblDSKhachHang");

dgvKhachHang.DataSource = ds.Tables["tblDSKhachHang"];

FormatGrid();

}

//xóa textbox

private void ClearFields()

{

txtMaKH.Clear();

txtTenKH.Clear();

txtSDT.Clear();

txtEmail.Clear();

cboDiaChi.SelectedIndex = -1;

cboTaiKhoan.SelectedIndex = -1;

}

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

{

if (e.RowIndex < 0) return; // Không click vào header

DataGridViewRow r = dgvKhachHang.Rows[e.RowIndex];

txtMaKH.Text = r.Cells["MaKH"].Value?.ToString() ?? "";

txtTenKH.Text = r.Cells["TenKH"].Value?.ToString() ?? "";

txtSDT.Text = r.Cells["SoDienThoai"].Value?.ToString() ?? "";

txtEmail.Text = r.Cells["Email"].Value?.ToString() ?? "";

cboDiaChi.Text = r.Cells["DiaChi"].Value?.ToString() ?? "";

cboTaiKhoan.Text = r.Cells["LoaiTaiKhoan"].Value?.ToString() ?? "";

}

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

{

if (txtMaKH.Text.Trim() == "" || txtTenKH.Text.Trim() == "")

{

MessageBox.Show("Vui lòng nhập Mã KH và Tên KH!");

return;

}

// ===== KIỂM TRA TRÙNG MÃ KH =====

DataTable tbl = ds.Tables["tblDSKhachHang"];

string ma = txtMaKH.Text.Trim();

bool trungMa = tbl.AsEnumerable()

.Any(r => r.RowState != DataRowState.Deleted &&

r["MaKH"].ToString() == ma);

if (trungMa)

{

MessageBox.Show("Mã khách hàng đã tồn tại!\nVui lòng nhập mã khác!");

return;

}

// ===== THÊM MỚI =====

DataRow row = tbl.NewRow();

row["MaKH"] = txtMaKH.Text.Trim();

row["TenKH"] = txtTenKH.Text.Trim();

row["SoDienThoai"] = txtSDT.Text.Trim();

row["Email"] = txtEmail.Text.Trim();

row["DiaChi"] = cboDiaChi.Text.Trim();

row["LoaiTaiKhoan"] = cboTaiKhoan.Text.Trim();

tbl.Rows.Add(row);

MessageBox.Show("Đã thêm khách hàng!");

ClearFields();

}

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

{

if (dgvKhachHang.SelectedRows.Count == 0)

{

MessageBox.Show("Vui lòng chọn dòng cần xóa!");

return;

}

dgvKhachHang.Rows.RemoveAt(dgvKhachHang.SelectedRows[0].Index);

MessageBox.Show("Đã xóa!");

}

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

{

if (dgvKhachHang.SelectedRows.Count == 0)

{

MessageBox.Show("Vui lòng chọn khách hàng để sửa!");

return;

}

DataGridViewRow r = dgvKhachHang.SelectedRows[0];

r.Cells["MaKH"].Value = txtMaKH.Text;

r.Cells["TenKH"].Value = txtTenKH.Text;

r.Cells["SoDienThoai"].Value = txtSDT.Text;

r.Cells["Email"].Value = txtEmail.Text;

r.Cells["DiaChi"].Value = cboDiaChi.Text;

r.Cells["LoaiTaiKhoan"].Value = cboTaiKhoan.Text;

MessageBox.Show("Đã sửa thông tin!");

}

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

{

try

{

daKhachHang.Update(ds.Tables["tblDSKhachHang"]);

MessageBox.Show("Lưu dữ liệu thành công!");

}

catch (Exception ex)

{

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

}

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // CHỈ đóng form hiện tại

}

}

}

}

### ***V. Form Hóa Đơn***

Ảnh có chứa văn bản, ảnh chụp màn hình, số, màn hình

Mô tả được tạo tự động

Code của giao diện:

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;

using System.Xml;

namespace DoAnMonHoc

{

public partial class frmHoaDon : Form

{

public frmHoaDon()

{

InitializeComponent();

}

DataSet ds = new DataSet("dsQLCH");

SqlDataAdapter daHoaDon;

SqlConnection conn;

SqlCommandBuilder cb;

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

{

conn = new SqlConnection(

@"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True");

string query = "SELECT \* FROM HoaDon";

daHoaDon = new SqlDataAdapter(query, conn);

cb = new SqlCommandBuilder(daHoaDon);

daHoaDon.Fill(ds, "tblDSHoaDon");

dgvHoaDon.DataSource = ds.Tables["tblDSHoaDon"];

FormatGrid();

}

private void FormatGrid()

{

dgvHoaDon.Columns["MaKH"].HeaderText = "Mã KH";

dgvHoaDon.Columns["MaKH"].Width = 140;

dgvHoaDon.Columns["MaHD"].HeaderText = "Mã HD";

dgvHoaDon.Columns["MaHD"].Width = 140;

dgvHoaDon.Columns["MaSP"].HeaderText = "Mã SP";

dgvHoaDon.Columns["MaSP"].Width = 140;

dgvHoaDon.Columns["PhuongThucThanhToan"].HeaderText = "PTTT";

dgvHoaDon.Columns["PhuongThucThanhToan"].Width = 150;

dgvHoaDon.Columns["NgayBan"].HeaderText = "Ngày Bán";

dgvHoaDon .Columns["NgayBan"].Width = 150;

dgvHoaDon.Columns["TongTien"].HeaderText = "Tổng Tiền";

dgvHoaDon.Columns["TongTien"].Width = 150;

//dgvHoaDon.Columns.Remove("MaTK");

}

private void ClearFields()

{

txtMaKH.Clear();

txtMaHD.Clear();

txtMaSP.Clear();

txtTongTien.Clear();

cboPTTT.SelectedIndex = -1;

}

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

{

if (e.RowIndex < 0) return; // Không click vào header

DataGridViewRow r = dgvHoaDon.Rows[e.RowIndex];

txtMaKH.Text = r.Cells["MaKH"].Value?.ToString() ?? "";

txtMaHD.Text = r.Cells["MaHD"].Value?.ToString() ?? "";

txtMaSP.Text = r.Cells["MaSP"].Value?.ToString() ?? "";

txtTongTien.Text = r.Cells["TongTien"].Value?.ToString() ?? "";

cboPTTT.Text = r.Cells["PhuongThucThanhToan"].Value?.ToString() ?? "";

dtpNgayBan.Text = r.Cells["NgayBan"].Value?.ToString() ?? "";

}

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

{

if (txtMaKH.Text.Trim() == "" || txtMaHD.Text.Trim() == "")

{

MessageBox.Show("Vui lòng nhập Mã KH và Mã HD!");

return;

}

DataTable tbl = ds.Tables["tblDSHoaDon"];

string ma = txtMaHD.Text.Trim();

bool trungMa = tbl.AsEnumerable()

.Any(r => r.RowState != DataRowState.Deleted &&

r["MaHD"].ToString() == ma);

if (trungMa)

{

MessageBox.Show("Mã HD đã tồn tại!\nVui lòng nhập mã khác!");

return;

}

DataRow row = tbl.NewRow();

row["MaKH"] = txtMaKH.Text.Trim();

row["MaHD"] = txtMaHD.Text.Trim();

row["MaSP"] = txtMaSP.Text.Trim();

row["TongTien"] = txtTongTien.Text.Trim();

row["PhuongThucThanhToan"] = cboPTTT.Text.Trim();

row["NgayBan"] = dtpNgayBan.Value;

tbl.Rows.Add(row);

MessageBox.Show("Đã thêm hóa đơn!");

ClearFields();

}

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

{

if (dgvHoaDon.SelectedRows.Count == 0)

{

MessageBox.Show("Vui lòng chọn dòng cần xóa!");

return;

}

string maHDChon = dgvHoaDon.SelectedRows[0].Cells["MaHD"].Value.ToString();

DataTable tbl = ds.Tables["tblDSHoaDon"];

DataRow row = tbl.AsEnumerable()

.FirstOrDefault(r => r["MaHD"].ToString() == maHDChon);

if (row != null)

{

row.Delete(); // Đánh dấu xóa trong DataSet

MessageBox.Show("Đã xóa hóa đơn!");

}

else

{

MessageBox.Show("Không tìm thấy hóa đơn để xóa!");

}

}

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

{

if (dgvHoaDon.SelectedRows.Count == 0)

{

MessageBox.Show("Vui lòng chọn hóa đơn để sửa!");

return;

}

// Lấy mã hóa đơn của dòng được chọn

string maHDChon = dgvHoaDon.SelectedRows[0].Cells["MaHD"].Value.ToString();

// Tìm row trong DataSet theo MaHD

DataTable tbl = ds.Tables["tblDSHoaDon"];

DataRow row = tbl.AsEnumerable()

.FirstOrDefault(r => r["MaHD"].ToString() == maHDChon);

if (row != null)

{

row["MaKH"] = txtMaKH.Text.Trim();

row["MaSP"] = txtMaSP.Text.Trim();

row["TongTien"] = txtTongTien.Text.Trim();

row["PhuongThucThanhToan"] = cboPTTT.Text.Trim();

row["NgayBan"] = dtpNgayBan.Value;

MessageBox.Show("Đã sửa hóa đơn!");

}

else

{

MessageBox.Show("Không tìm thấy hóa đơn để sửa!");

}

}

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

{

try

{

conn.Open();

SqlCommand cmd1 = new SqlCommand(

"ALTER TABLE HoaDon NOCHECK CONSTRAINT ALL", conn);

cmd1.ExecuteNonQuery();

daHoaDon.Update(ds.Tables["tblDSHoaDon"]);

SqlCommand cmd2 = new SqlCommand(

"ALTER TABLE HoaDon CHECK CONSTRAINT ALL", conn);

cmd2.ExecuteNonQuery();

MessageBox.Show("Lưu dữ liệu thành công!");

}

catch (Exception ex)

{

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

}

finally

{

conn.Close();

}

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // CHỈ đóng form hiện tại

}

}

}

}

### ***VI. Form Quản Lý Doanh Thu***

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

Code của giao diện:

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;

namespace DoAnMonHoc

{

public partial class frmDoanhThu : Form

{

public frmDoanhThu()

{

InitializeComponent();

}

DataSet ds = new DataSet("dsQLCH");

SqlDataAdapter daDoanhThu;

SqlConnection conn;

SqlCommandBuilder cb;

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

{

conn = new SqlConnection(

@"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True");

string query = "SELECT \* FROM DoanhThu";

daDoanhThu = new SqlDataAdapter(query, conn);

cb = new SqlCommandBuilder(daDoanhThu);

daDoanhThu.Fill(ds, "tblDSDoanhThu");

dgvDoanhThu.DataSource = ds.Tables["tblDSDoanhThu"];

FormatGrid();

}

private void FormatGrid()

{

dgvDoanhThu.Columns["MaDT"].HeaderText = "Mã Doanh Thu";

dgvDoanhThu.Columns["MaDT"].Width = 170;

dgvDoanhThu.Columns["MaHD"].HeaderText = "Mã HD";

dgvDoanhThu.Columns["MaHD"].Width = 150;

dgvDoanhThu.Columns["Ngay"].HeaderText = "Ngày";

dgvDoanhThu.Columns["Ngay"].Width = 150;

dgvDoanhThu.Columns["TongTien"].HeaderText = "Tổng Tiền";

dgvDoanhThu.Columns["TongTien"].Width = 150;

dgvDoanhThu.Columns["GhiChu"].HeaderText = "Ghi Chú";

dgvDoanhThu.Columns["GhiChu"].Width = 300;

}

private void ClearFields()

{

txtMaDT.Clear();

txtMaHD.Clear();

txtSoTien.Clear();

txtGhiChu.Clear();

}

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

{

if (txtMaDT.Text.Trim() == "" || txtMaHD.Text.Trim() == "")

{

MessageBox.Show("Vui lòng nhập mã doanh thu và mã hóa đơn ");

return;

}

// ===== KIỂM TRA TRÙNG MÃ DT =====

DataTable tbl = ds.Tables["tblDSDoanhThu"];

string ma = txtMaDT.Text.Trim();

bool trungMa = tbl.AsEnumerable()

.Any(r => r.RowState != DataRowState.Deleted &&

r["MaDT"].ToString() == ma);

if (trungMa)

{

MessageBox.Show("Mã doanh thu đã tồn tại!\nVui lòng nhập mã khác!");

return;

}

// ===== THÊM MỚI =====

DataRow row = tbl.NewRow();

row["MaDT"] = txtMaDT.Text.Trim();

row["MaHD"] = txtMaHD.Text.Trim();

row["TongTien"] = txtSoTien.Text.Trim();

row["GhiChu"] = txtGhiChu.Text.Trim();

row["Ngay"] = dateNgay.Text.Trim();

tbl.Rows.Add(row);

MessageBox.Show("Đã thêm doanh thu!");

ClearFields();

}

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

{

if (e.RowIndex < 0) return; // bỏ header

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

txtMaDT.Text = row.Cells["MaDT"].Value?.ToString() ?? "";

txtMaHD.Text = row.Cells["MaHD"].Value?.ToString() ?? "";

txtSoTien.Text = row.Cells["TongTien"].Value?.ToString() ?? "";

txtGhiChu.Text = row.Cells["GhiChu"].Value?.ToString() ?? "";

if (DateTime.TryParse(row.Cells["Ngay"].Value?.ToString(), out DateTime ngay))

dateNgay.Value = ngay;

}

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

{

DataTable tbl = ds.Tables["tblDSDoanhThu"];

// Không có dữ liệu

if (tbl.Rows.Count == 0)

{

MessageBox.Show("Không có dữ liệu để xóa!");

return;

}

// Chưa chọn dòng

if (dgvDoanhThu.CurrentRow == null)

{

MessageBox.Show("Vui lòng chọn dòng cần xóa!");

return;

}

int index = dgvDoanhThu.CurrentRow.Index;

// Xác nhận

DialogResult dr = MessageBox.Show("Bạn có chắc muốn xóa?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

// ❗ Xóa trên DataTable – không xóa trực tiếp trên DataGrid

tbl.Rows[index].Delete();

MessageBox.Show("Đã xóa!");

}

}

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

{

if (dgvDoanhThu.SelectedRows.Count == 0)

{

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

return;

}

DataGridViewRow r = dgvDoanhThu.SelectedRows[0];

r.Cells["MaDT"].Value = txtMaDT.Text.Trim();

r.Cells["MaHD"].Value = txtMaHD.Text.Trim();

r.Cells["Ngay"].Value = dateNgay.Value;

r.Cells["TongTien"].Value = txtSoTien.Text.Trim();

r.Cells["GhiChu"].Value = txtGhiChu.Text.Trim();

MessageBox.Show("Đã sửa thành công!");

}

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

{

try

{

conn.Open();

SqlCommand cmd1 = new SqlCommand(

"ALTER TABLE DoanhThu NOCHECK CONSTRAINT ALL", conn);

cmd1.ExecuteNonQuery();

daDoanhThu.Update(ds.Tables["tblDSDoanhThu"]);

SqlCommand cmd2 = new SqlCommand(

"ALTER TABLE DoanhThu CHECK CONSTRAINT ALL", conn);

cmd2.ExecuteNonQuery();

MessageBox.Show("Lưu dữ liệu thành công!");

}

catch (Exception ex)

{

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

}

finally

{

conn.Close();

}

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // CHỈ đóng form hiện tại

}

}

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

{

txtMaDT.Clear();

txtMaHD.Clear();

txtGhiChu.Clear();

txtSoTien.Clear();

txtMaDT.Focus();

}

}

}

### ***VII. Form Quản Lý Nhập Hàng***

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Code của giao diện:

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;

namespace DoAnMonHoc

{

public partial class frmNhap : Form

{

public frmNhap()

{

InitializeComponent();

}

SqlConnection conn;

SqlDataAdapter daPhieuNhap;

SqlCommandBuilder cb;

DataSet ds = new DataSet();

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

{

conn = new SqlConnection(

@"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True");

string query = "SELECT \* FROM PhieuNhap";

daPhieuNhap = new SqlDataAdapter(query, conn);

// Tự động sinh Insert, Update, Delete

cb = new SqlCommandBuilder(daPhieuNhap);

daPhieuNhap.Fill(ds, "tblDSPhieuNhap");

dgvPhieuNhap.DataSource = ds.Tables["tblDSPhieuNhap"];

FormatGrid();

}

private void FormatGrid()

{

dgvPhieuNhap.Columns["MaPN"].HeaderText = "Mã Phiếu Nhập";

dgvPhieuNhap.Columns["MaPN"].Width = 170;

dgvPhieuNhap.Columns["MaSP"].HeaderText = "Mã SP";

dgvPhieuNhap.Columns["MaSP"].Width = 100;

dgvPhieuNhap.Columns["TongTien"].HeaderText = "Tổng Tiền";

dgvPhieuNhap.Columns["TongTien"].Width = 150;

dgvPhieuNhap.Columns["NgayNhap"].HeaderText = "Ngày nhập";

dgvPhieuNhap.Columns["NgayNhap"].Width = 150;

dgvPhieuNhap.Columns["GhiChu"].HeaderText = "Ghi Chú";

dgvPhieuNhap.Columns["GhiChu"].Width = 250;

dgvPhieuNhap.Columns["SoLuongNhap"].HeaderText = "SL";

dgvPhieuNhap.Columns["SoluongNhap"].Width = 100;

}

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

{

if (dgvPhieuNhap.SelectedRows.Count == 0)

{

MessageBox.Show("Vui lòng chọn dòng cần xóa!");

return;

}

dgvPhieuNhap.Rows.RemoveAt(dgvPhieuNhap.SelectedRows[0].Index);

MessageBox.Show("Đã xóa!");

}

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

{

try

{

daPhieuNhap.Update(ds.Tables["tblDSPhieuNhap"]);

MessageBox.Show("Lưu dữ liệu thành công!");

}

catch (Exception ex)

{

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

}

}

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

{

if (dgvPhieuNhap.SelectedRows.Count == 0)

{

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

return;

}

DataGridViewRow r = dgvPhieuNhap.SelectedRows[0];

r.Cells["MaPN"].Value = txtMaPN.Text;

r.Cells["MaSP"].Value = txtMaSP.Text;

r.Cells["TongTien"].Value = txtTongTien.Text;

r.Cells["GhiChu"].Value = txtGhiChu.Text;

r.Cells["NgayNhap"].Value =dtpNgayNhap.Text;

r.Cells["SoLuongNhap"].Value = txtSoLuong.Text;

MessageBox.Show("Đã sửa thông tin!");

}

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

{

txtMaPN.Clear();

txtMaSP.Clear();

txtGhiChu.Clear();

txtTongTien.Clear();

txtSoLuong.Clear();

txtMaPN.Focus();

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // CHỈ đóng form hiện tại

}

}

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

{

if (txtMaPN.Text.Trim() == "" || txtMaSP.Text.Trim() == "")

{

MessageBox.Show("Vui lòng nhập Mã PN và Mã SP!");

return;

}

// ===== KIỂM TRA TRÙNG MÃ PN =====

DataTable tbl = ds.Tables["tblDSPhieuNhap"];

string ma = txtMaPN.Text.Trim();

bool trungMa = tbl.AsEnumerable()

.Any(r => r.RowState != DataRowState.Deleted &&

r["MaPN"].ToString() == ma);

if (trungMa)

{

MessageBox.Show("Mã phiếu nhập đã tồn tại!\nVui lòng nhập mã khác!");

return;

}

// ===== THÊM MỚI =====

DataRow row = tbl.NewRow();

row["MaPN"] = txtMaPN.Text.Trim();

row["MaSP"] = txtMaSP.Text.Trim();

row["TongTien"] = txtTongTien.Text.Trim();

row["GhiChu"] = txtGhiChu.Text.Trim();

row["NgayNhap"] = dtpNgayNhap.Text.Trim();

row["SoLuongNhap"] = txtSoLuong.Text.Trim();

tbl.Rows.Add(row);

MessageBox.Show("Đã thêm phiếu nhập!");

ClearFields();

}

//xóa textbox

private void ClearFields()

{

txtMaPN.Clear();

txtMaSP.Clear();

txtTongTien.Clear();

txtGhiChu.Clear();

txtSoLuong.Clear();

}

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

{

DataGridViewRow r = dgvPhieuNhap.Rows[e.RowIndex];

txtMaPN.Text = r.Cells["MaPN"].Value?.ToString() ?? "";

txtMaSP.Text = r.Cells["MaSP"].Value?.ToString() ?? "";

txtTongTien.Text = r.Cells["TongTien"].Value?.ToString() ?? "";

txtGhiChu.Text = r.Cells["GhiChu"].Value?.ToString() ?? "";

txtSoLuong.Text = r.Cells["SoLuongNhap"].Value?.ToString() ?? "";

dtpNgayNhap.Text = r.Cells["NgayNhap"].Value?.ToString() ?? "";

}

}

}

### ***VIII. Form Quản Lý Kho***

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

Code của giao diện:

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;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace DoAnMonHoc

{

public partial class frmKho : Form

{

public frmKho()

{

InitializeComponent();

}

DataSet ds = new DataSet("dsQLCH");

SqlDataAdapter daKho;

void ClearFields()

{

txtMaSP.Clear();

txtTen.Clear();

txtSLTon.Clear();

txtTGTon.Clear();

txtCanhBao.Clear();

cboKM.SelectedIndex = 0;

}

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

{

if (txtMaSP.Text == "" || txtTen.Text == "" || txtSLTon.Text == "")

{

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

return;

}

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

// Kiểm tra trùng mã SP

string check = "SELECT COUNT(\*) FROM Kho WHERE MaSP = @MaSP";

SqlCommand c1 = new SqlCommand(check, conn);

c1.Parameters.AddWithValue("@MaSP", txtMaSP.Text.Trim());

if ((int)c1.ExecuteScalar() > 0)

{

MessageBox.Show("❌ Mã sản phẩm đã tồn tại, không thể thêm!");

return;

}

string query = @"INSERT INTO Kho (MaSP, TenSP, SLTon, ThoiGianTon, CanhBao, KM)

VALUES (@MaSP, @TenSP, @SLTon, @ThoiGianTon, @CanhBao, @KM)";

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text.Trim());

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

cmd.Parameters.AddWithValue("@SLTon", int.Parse(txtSLTon.Text));

cmd.Parameters.AddWithValue("@ThoiGianTon", txtTGTon.Text.Trim());

cmd.Parameters.AddWithValue("@CanhBao", txtCanhBao.Text.Trim());

cmd.Parameters.AddWithValue("@KM", cboKM.Text.Trim());

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm sản phẩm thành công!");

LoadKho();

ClearFields();

}

}

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

{

if (txtMaSP.Text == "")

{

MessageBox.Show("Vui lòng chọn sản phẩm cần sửa!");

return;

}

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = @"

UPDATE Kho

SET

TenSP = @TenSP,

SLTon = @SLTon,

ThoiGianTon = @ThoiGianTon,

CanhBao = @CanhBao,

KM = @KM

WHERE MaSP = @MaSP";

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);

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

cmd.Parameters.AddWithValue("@SLTon", txtSLTon.Text);

cmd.Parameters.AddWithValue("@ThoiGianTon", txtTGTon.Text);

cmd.Parameters.AddWithValue("@CanhBao", txtCanhBao.Text);

cmd.Parameters.AddWithValue("@KM", cboKM.Text);

int rows = cmd.ExecuteNonQuery();

if (rows > 0)

{

MessageBox.Show("Sửa thành công!");

LoadKho();

ClearFields();

}

else

{

MessageBox.Show("❌ Sửa thất bại!\nKiểm tra lại MaSP hoặc chưa chọn dòng.");

}

}

}

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

{

if (txtMaSP.Text == "" || txtTen.Text == "" || txtSLTon.Text == "")

{

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

return;

}

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

// Kiểm tra MaSP đã tồn tại chưa

string checkQuery = "SELECT COUNT(\*) FROM Kho WHERE MaSP = @MaSP";

SqlCommand checkCmd = new SqlCommand(checkQuery, conn);

checkCmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);

int count = (int)checkCmd.ExecuteScalar();

if (count > 0) // Nếu tồn tại → UPDATE

{

string updateQuery = @"

UPDATE Kho

SET TenSP = @TenSP,

SLTon = @SLTon,

ThoiGianTon = @TGTon,

CanhBao = @CanhBao,

KM = @KM

WHERE MaSP = @MaSP";

SqlCommand cmd = new SqlCommand(updateQuery, conn);

cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);

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

cmd.Parameters.AddWithValue("@SLTon", txtSLTon.Text);

cmd.Parameters.AddWithValue("@TGTon", txtTGTon.Text);

cmd.Parameters.AddWithValue("@CanhBao", txtCanhBao.Text);

cmd.Parameters.AddWithValue("@KM", cboKM.Text);

int rows = cmd.ExecuteNonQuery();

if (rows > 0) MessageBox.Show("Cập nhật thành công!");

}

else // Nếu chưa tồn tại → INSERT

{

string insertQuery = @"

INSERT INTO Kho (MaSP, TenSP, SLTon, ThoiGianTon, CanhBao, KM)

VALUES (@MaSP, @TenSP, @SLTon, @TGTon, @CanhBao, @KM)";

SqlCommand cmd = new SqlCommand(insertQuery, conn);

cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);

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

cmd.Parameters.AddWithValue("@SLTon", txtSLTon.Text);

cmd.Parameters.AddWithValue("@TGTon", txtTGTon.Text);

cmd.Parameters.AddWithValue("@CanhBao", txtCanhBao.Text);

cmd.Parameters.AddWithValue("@KM", cboKM.Text);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm thành công!");

}

catch (Exception ex)

{

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

}

}

LoadKho(); // Load lại ListView sau khi lưu

ClearFields(); // Xóa dữ liệu TextBox

}

}

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

{

if (txtMaSP.Text == "")

{

MessageBox.Show("Vui lòng chọn sản phẩm cần xóa!");

return;

}

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "DELETE FROM Kho WHERE MaSP = @MaSP";

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@MaSP", txtMaSP.Text);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!");

LoadKho();

ClearFields();

conn.Close();

}

}

string connectionString = @"Data Source=ADMIN-PC\SQLEXPRESS;Initial Catalog=QLCH;Integrated Security=True";

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

{

LoadKho();

}

void LoadKho()

{

lstKho.Items.Clear();

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "SELECT \* FROM Kho";

SqlCommand cmd = new SqlCommand(query, conn);

SqlDataReader rd = cmd.ExecuteReader();

while (rd.Read())

{

ListViewItem item = new ListViewItem(rd["MaSP"].ToString());

item.SubItems.Add(rd["TenSP"].ToString());

item.SubItems.Add(rd["SLTon"].ToString());

item.SubItems.Add(rd["ThoiGianTon"].ToString());

item.SubItems.Add(rd["CanhBao"].ToString());

item.SubItems.Add(rd["KM"].ToString());

lstKho.Items.Add(item);

}

conn.Close();

}

}

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

{

if (lstKho.SelectedItems.Count > 0)

{

ListViewItem item = lstKho.SelectedItems[0];

txtMaSP.Text = item.SubItems[0].Text;

txtTen.Text = item.SubItems[1].Text;

txtSLTon.Text = item.SubItems[2].Text;

txtTGTon.Text = item.SubItems[3].Text;

txtCanhBao.Text = item.SubItems[4].Text;

cboKM.Text = item.SubItems[5].Text;

}

}

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

{

txtMaSP.Clear();

txtTen.Clear();

txtSLTon.Clear();

txtTGTon.Clear();

txtCanhBao.Clear();

txtMaSP.Focus();

}

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

{

DialogResult dr = MessageBox.Show("Bạn có chắc chắn muốn thoát?",

"Xác nhận",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question);

if (dr == DialogResult.Yes)

{

this.Close(); // hoặc Application.Exit();

}

}

}

}

# CHƯƠNG 4: KẾT LUẬN VÀ TÀI LIỆU THAM KHẢO

## ***I. Kết luận***

Sau quá trình thực hiện đồ án “Xây dựng ứng dụng Quản lý cửa hàng quần áo NEM bằng C# .NET và SQL Server”, nhóm tác giả rút ra một số kết luận quan trọng như sau:

**1. Hoàn thành mục tiêu đề ra**

Hệ thống quản lý cửa hàng quần áo đã được xây dựng thành công trên nền tảng Windows Form với ngôn ngữ C# .NET, kết nối với cơ sở dữ liệu SQL Server.

Hệ thống đáp ứng đầy đủ các yêu cầu cơ bản: quản lý sản phẩm, hóa đơn bán hàng, kho, nhập hàng, doanh thu và quản lý khách hàng.

**2. Cải thiện hiệu quả quản lý**

Hệ thống giúp chủ cửa hàng và nhân viên tiết kiệm thời gian tra cứu và cập nhật thông tin, đồng thời giảm thiểu lỗi nhập liệu thủ công.

Thống kê doanh thu, tồn kho và báo cáo bán hàng được thực hiện nhanh chóng, chính xác, hỗ trợ ra quyết định kinh doanh hiệu quả hơn.

**3. Tính năng mở rộng và ổn định**

Giao diện đơn giản, dễ sử dụng, phù hợp với các cửa hàng vừa và nhỏ.

Hệ thống có khả năng mở rộng trong tương lai, ví dụ: tích hợp thanh toán online, quản lý chuỗi cửa hàng hoặc báo cáo thống kê nâng cao.

**4. Hạn chế và hướng phát triển**

Hiện tại hệ thống chưa tích hợp các chức năng nâng cao như quản lý tồn kho chi tiết, phân tích dữ liệu thông minh hay báo cáo đa chiều.

Trong tương lai, có thể nâng cấp hệ thống với các module quản lý bán hàng online, tích hợp đa nền tảng và nâng cao bảo mật dữ liệu.

**Tóm lại:** đồ án đã hoàn thành mục tiêu xây dựng một hệ thống quản lý cửa hàng quần áo cơ bản nhưng hiệu quả, góp phần hiện đại hóa quá trình quản lý kinh doanh, giảm thiểu sai sót và tăng hiệu quả quản lý kho, doanh thu và khách hàng.

## II. Tài Liệu Tham Khảo

1. Nguyễn Văn A, Lập trình C# .NET cho ứng dụng Windows Form, Nhà xuất bản Bách Khoa, 2020.
2. Trần Thị B, SQL Server cơ bản và nâng cao, Nhà xuất bản Tổng hợp, 2019.
3. Microsoft Docs, Windows Forms Overview, <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/>
4. Microsoft Docs, SQL Server Documentation, https://learn.microsoft.com/en-us/sql/sql-server/
5. Nguyễn Văn C, Thiết kế hệ thống quản lý bán hàng, Nhà xuất bản Khoa học & Kỹ thuật, 2018.
6. Sách và tài liệu tham khảo về quản lý kho và bán lẻ: Kotler, Marketing Management, 2017.
7. Giáo trình Lập trình C# và ứng dụng, Khoa Công nghệ thông tin, Trường Đại học An Giang