

Lab 05:

TẠO BÁO CÁO THỐNG KÊ VỚI DATA REPORT

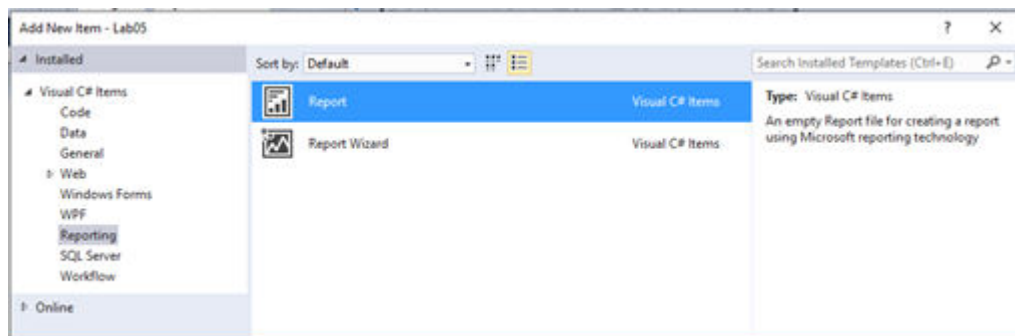
A. MỤC TIÊU:

- ✓ Cài đặt, Thiết kế các báo cáo thống kê với Data Report
- ✓ Ôn tập lại kiến thức đã học

B. HƯỚNG DẪN TẠO DATA REPORT

- ✓ Bước 1: Kiểm tra xem đã cài đặt Report cho VS hay chưa, nếu đã cài đặt rồi thì chuyển qua bước 3

Kiểm tra bằng cách khi Add/ New Item thấy có phần item **Reporting**



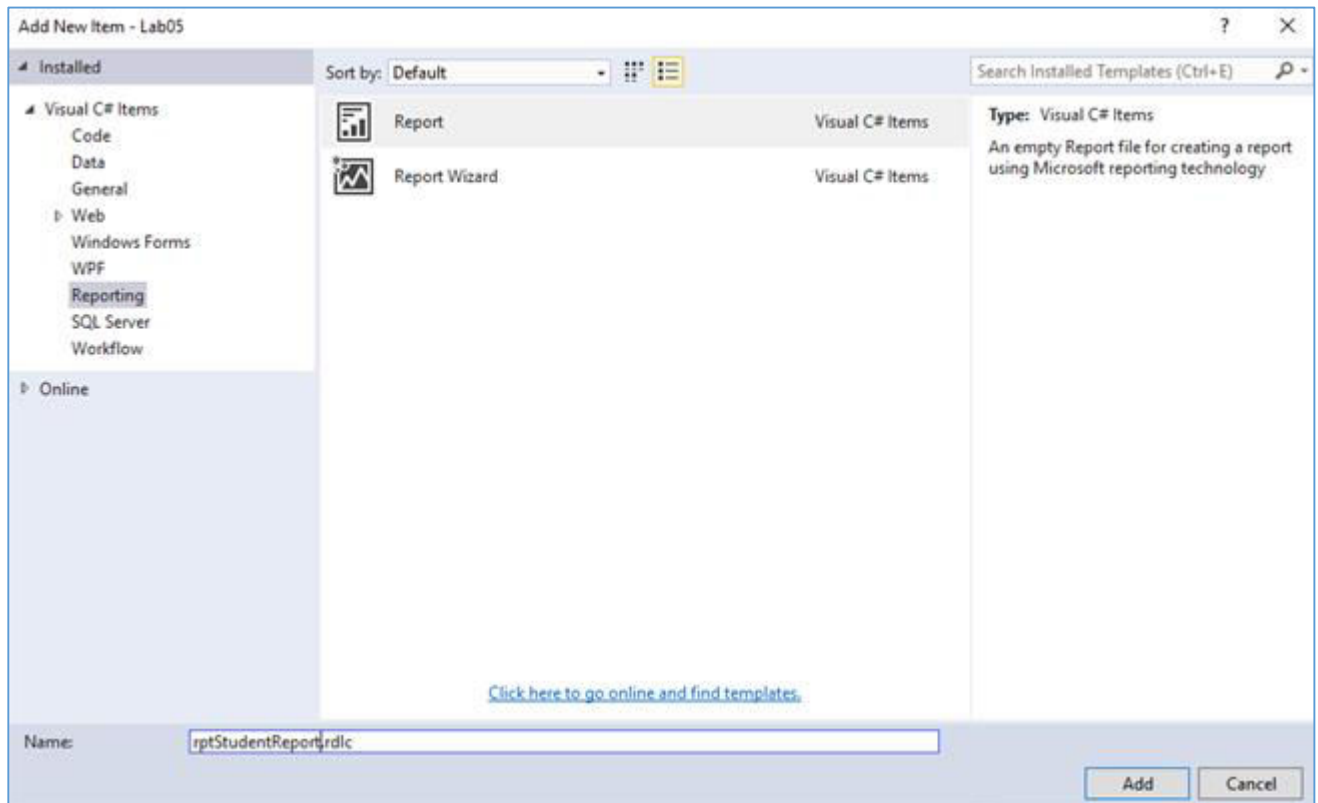
- ✓ Bước 2: Cài đặt **Report Viewer** Cho Visual Studio 2015
- Vào Control Panel chọn Programs And Feature , Chọn Visual Studio Installer , Click phải chọn Change

Microsoft Visual Studio 2008 Shell (integrated mode) ...	Microsoft Corporation	8/11/2019	224 MB	9.0.30729
Microsoft Visual Studio Installer	Microsoft Corporation	8/11/2019		2.2.3073.701
Microsoft Visual Studio 2015 with Updates	Microsoft Corporation	8/11/2019	6.15 GB	14.0.25420.1
Microsoft Visual Studio 2015 Tools - E...	Microsoft Corporation	8/11/2019	244 MB	9.0.35191
Microsoft Web Deploy 4.0	Microsoft Corporation	8/11/2019	9.55 MB	10.0.2606

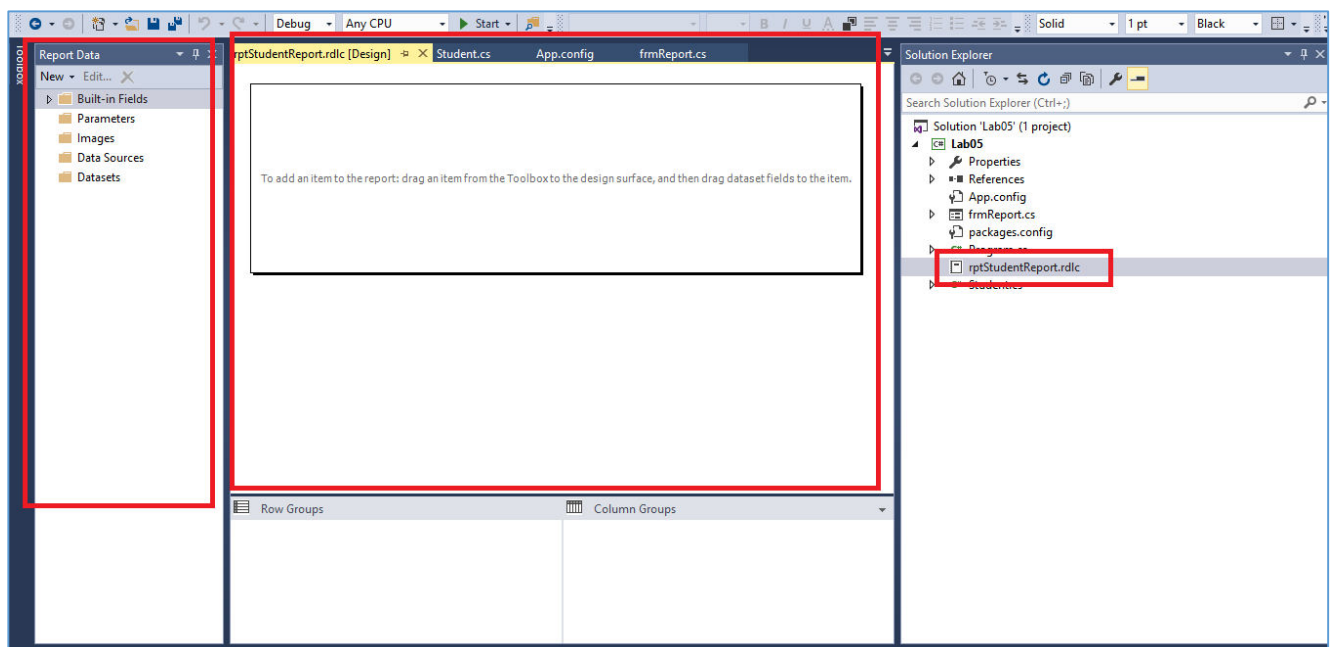
- Check vào Microsoft SQL Server Data Tools, Next để tiến hành hoàn tất



✓ Bước 3: Tạo Data Report

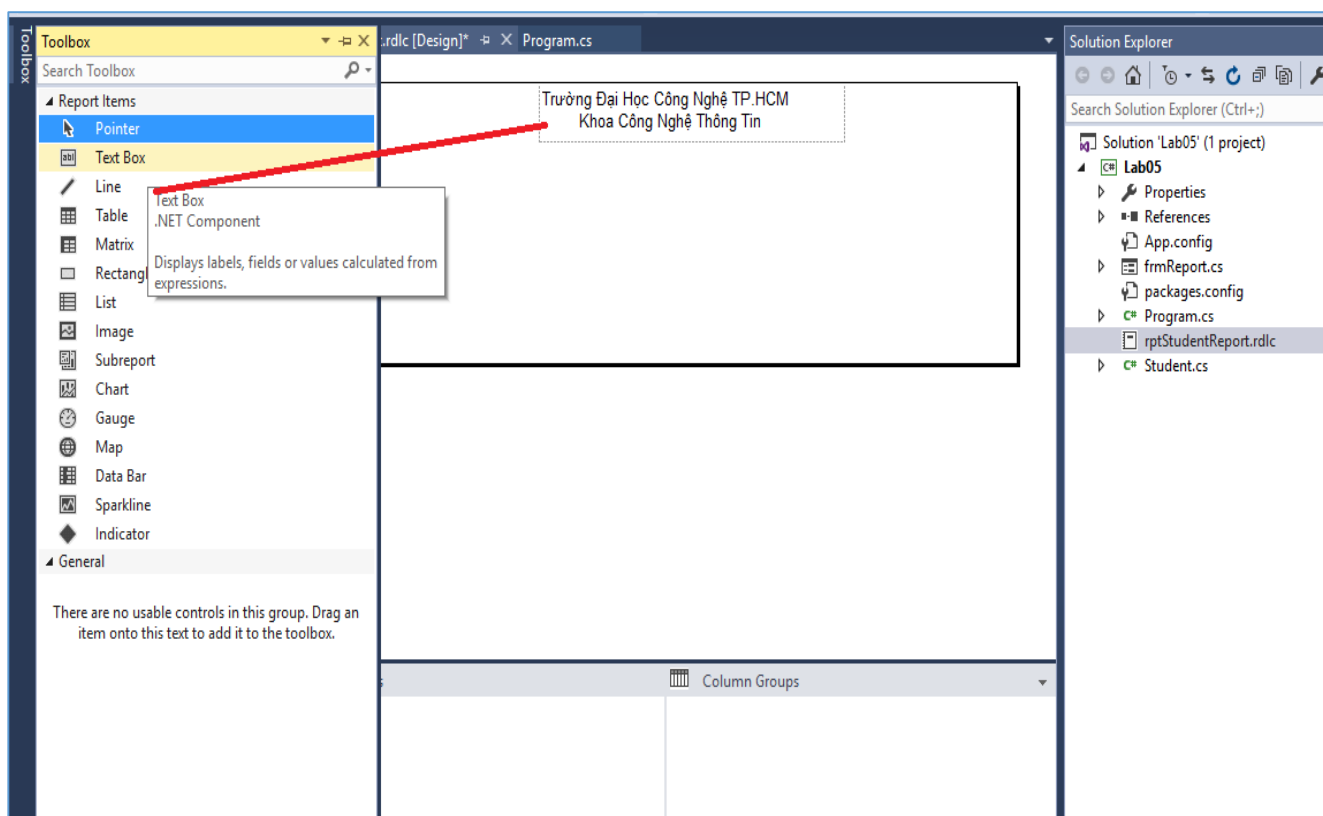


✓ Giao diện của phần report design



✓ Bước 4: Thiết kế mẫu Report Bằng cách sử dụng toolbox

- Sử dụng Text để hiển thị thông tin trên Report file.



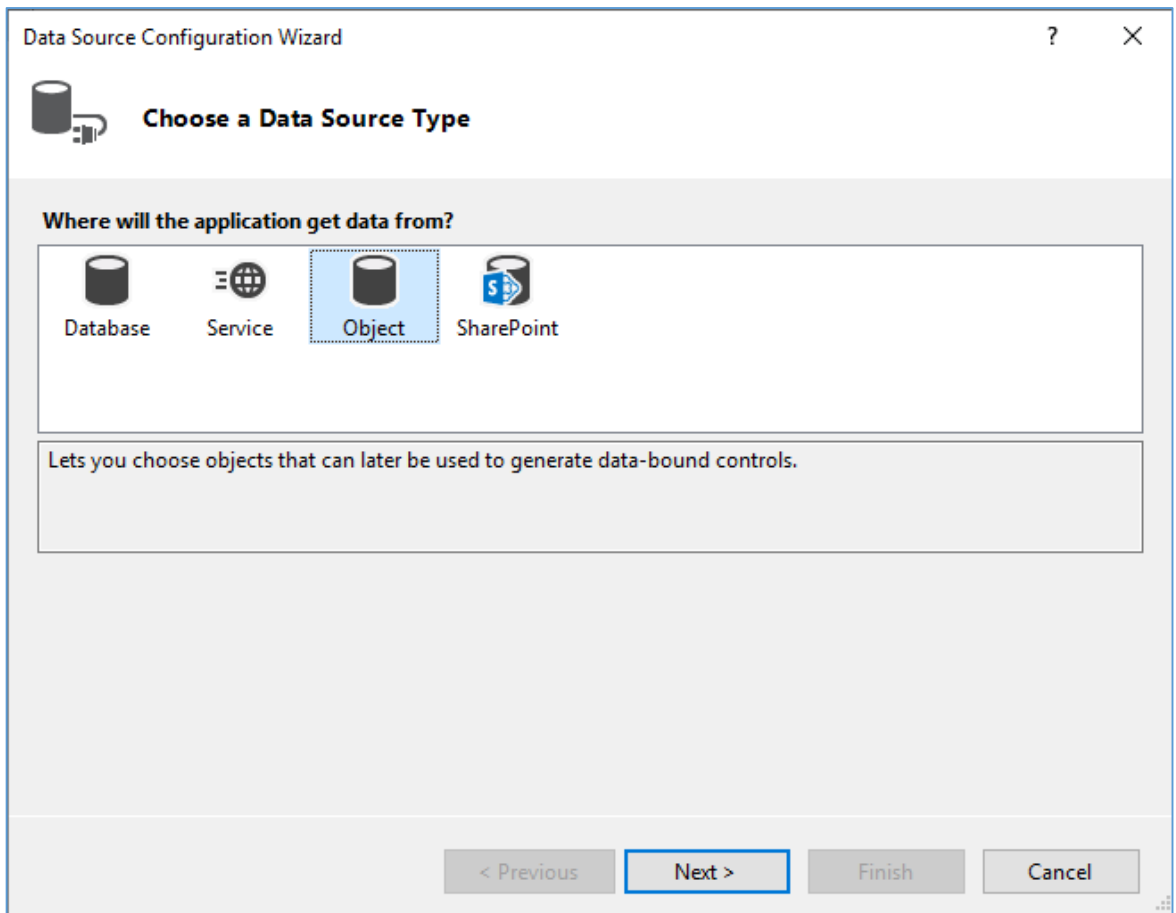
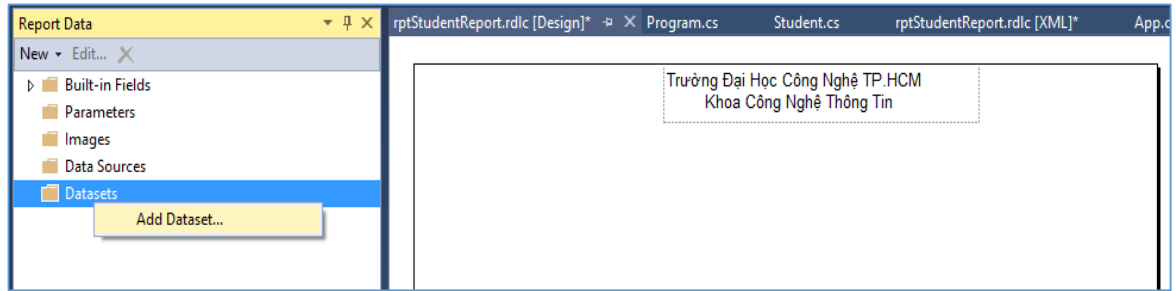
- Sử dụng Table: Để lấy List dữ liệu từ datasource cụ thể (Database, Object, Sharepoint ...).

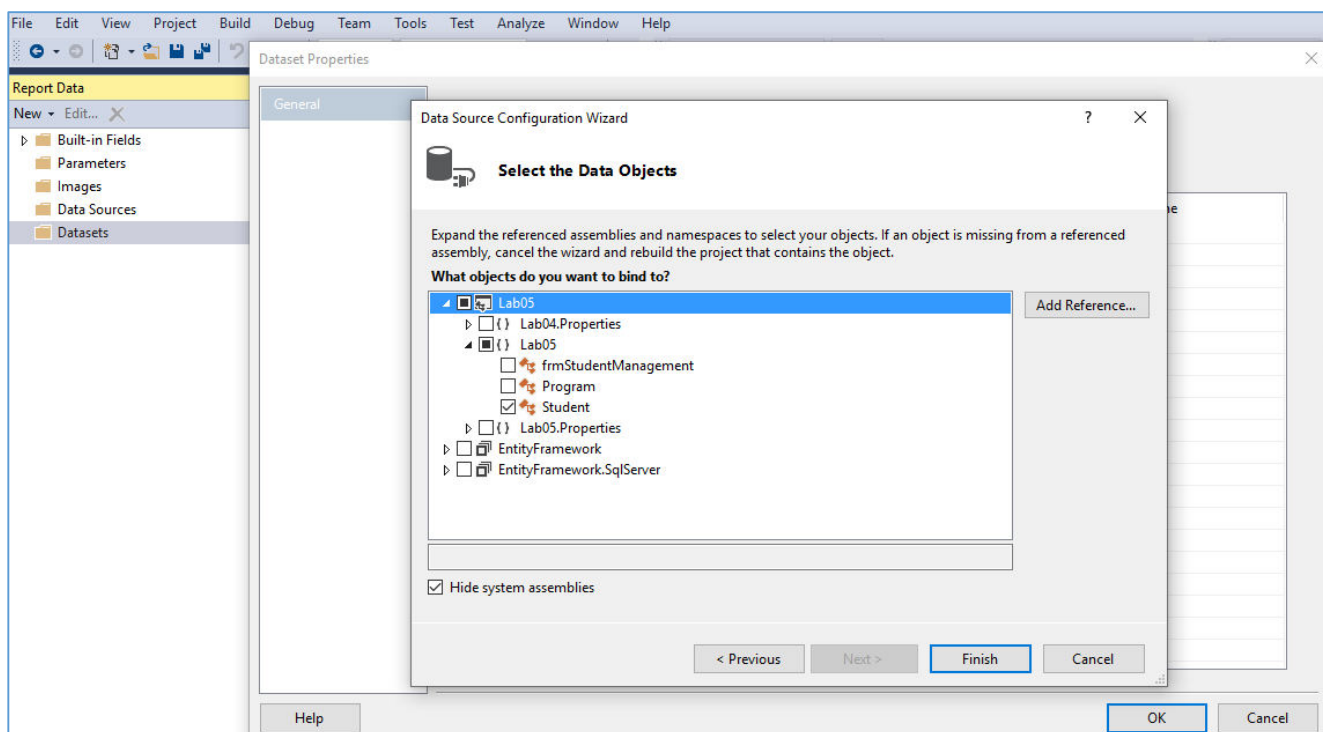
Trong bài thực hành này sẽ mô tả lấy datasource từ Object

+ Xây Dựng 1 class Student có sẵn như sau

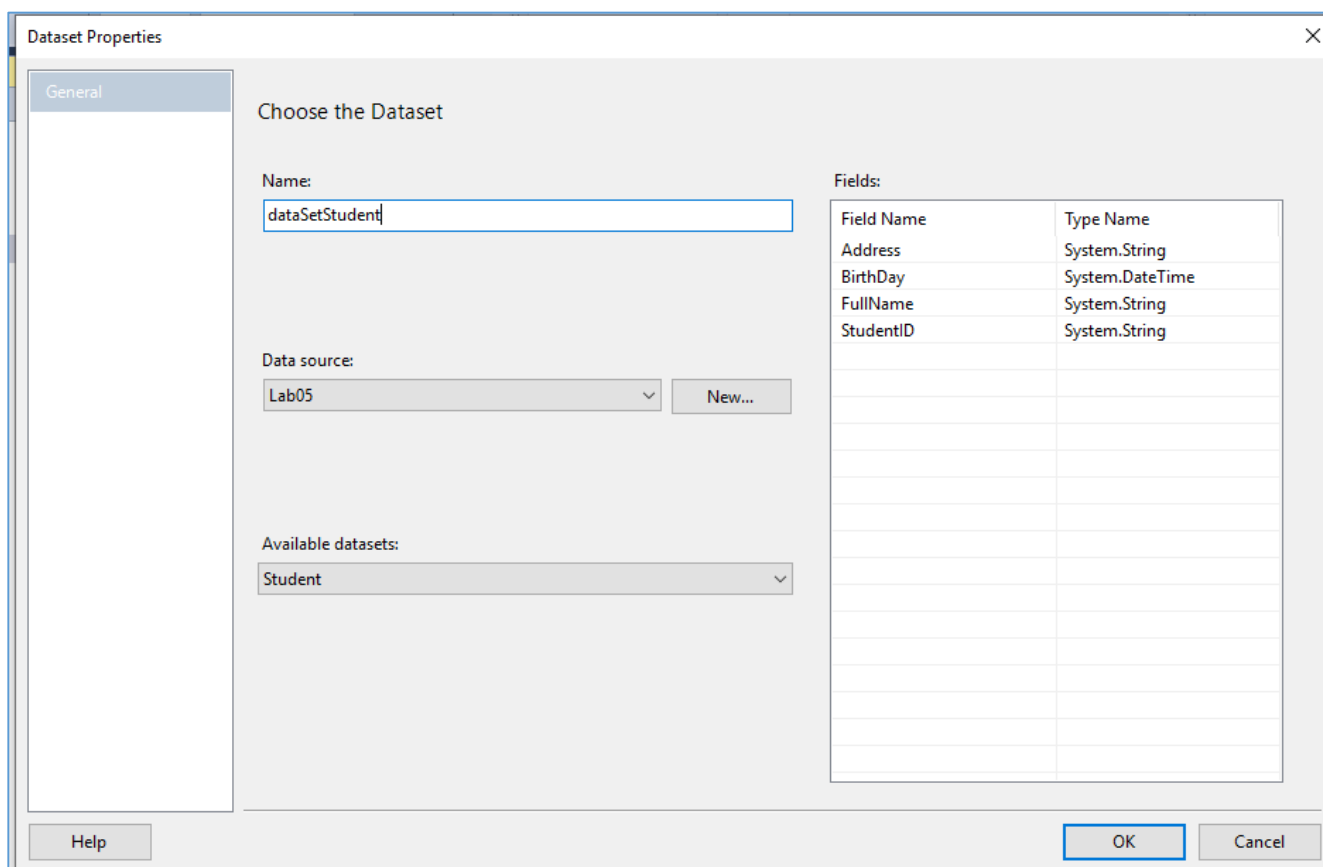
```
public class Student
{
    public string StudentID { get; set; }
    public string FullName { get; set; }
    public DateTime BirthDay { get; set; }
    public string Address { get; set; }
    public Student()
    {
    }
    public Student(string id, string name, DateTime dtBirthDay, string add)
    {
        StudentID = id;
        FullName = name;
        BirthDay = dtBirthDay;
        Address = add;
    }
}
```

- Tạo DataSet từ Object (SV nghiên cứu tạo DataSets từ database)

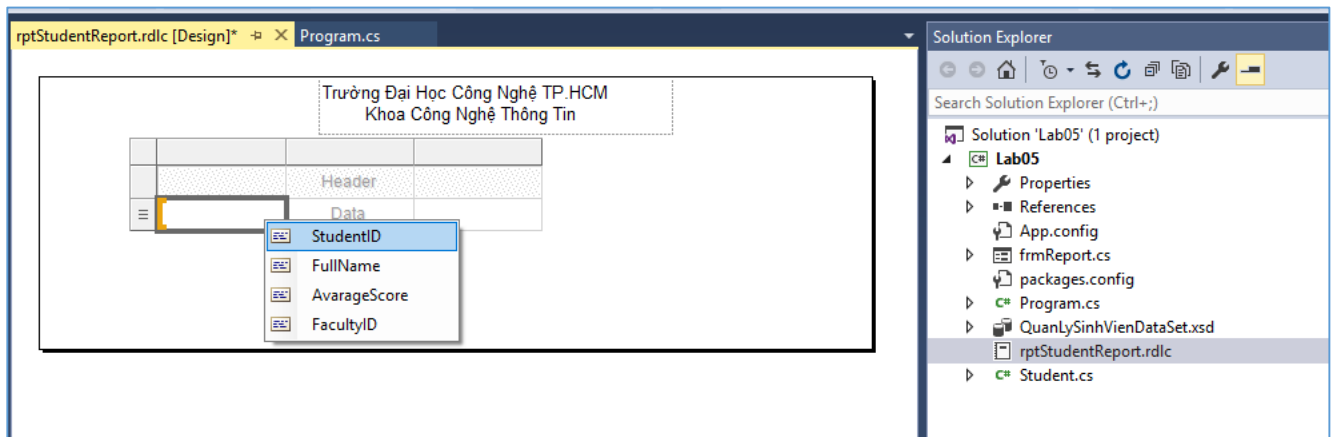




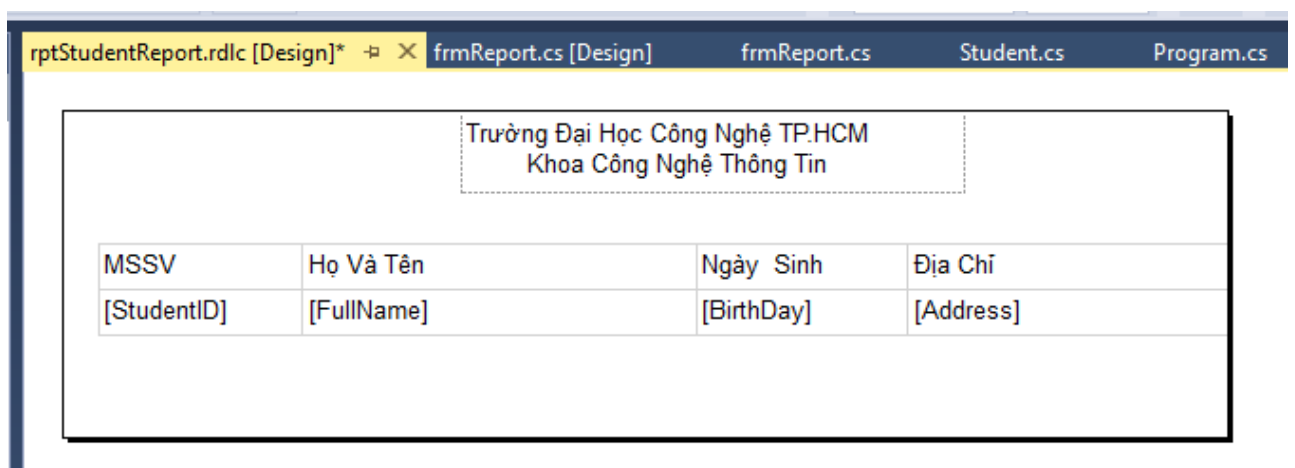
- Đặt tên cho DataSet chứa trong report file



- Chọn các field tương ứng vào report

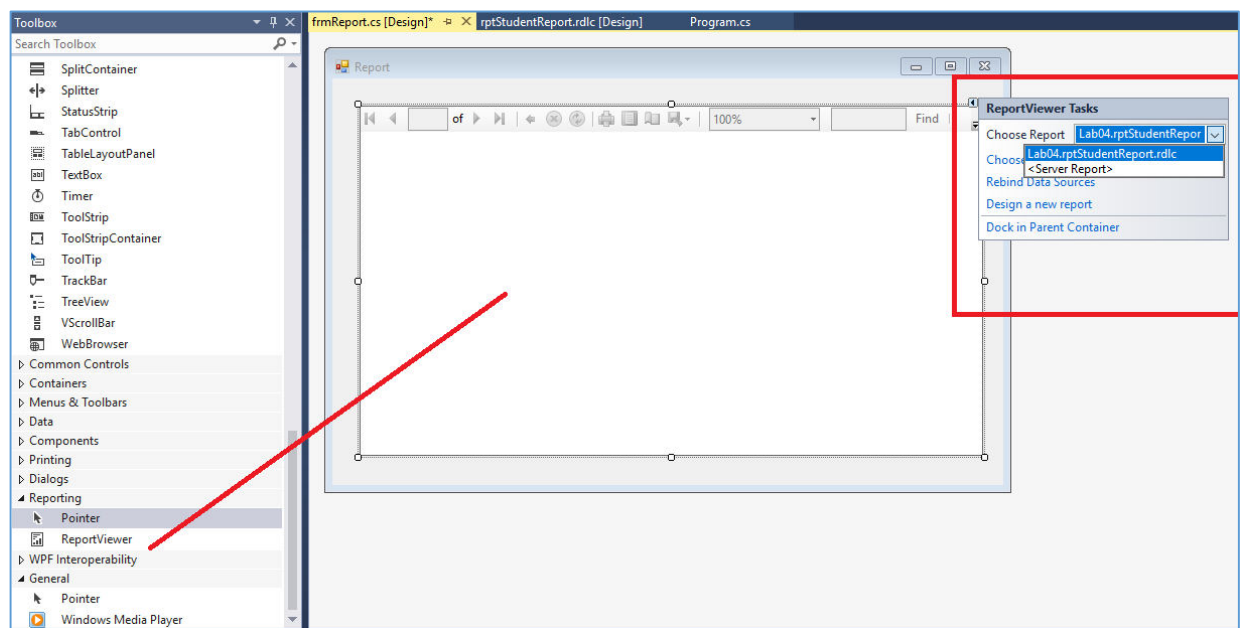


Và thiết kế lại kích thước, tên header tương ứng

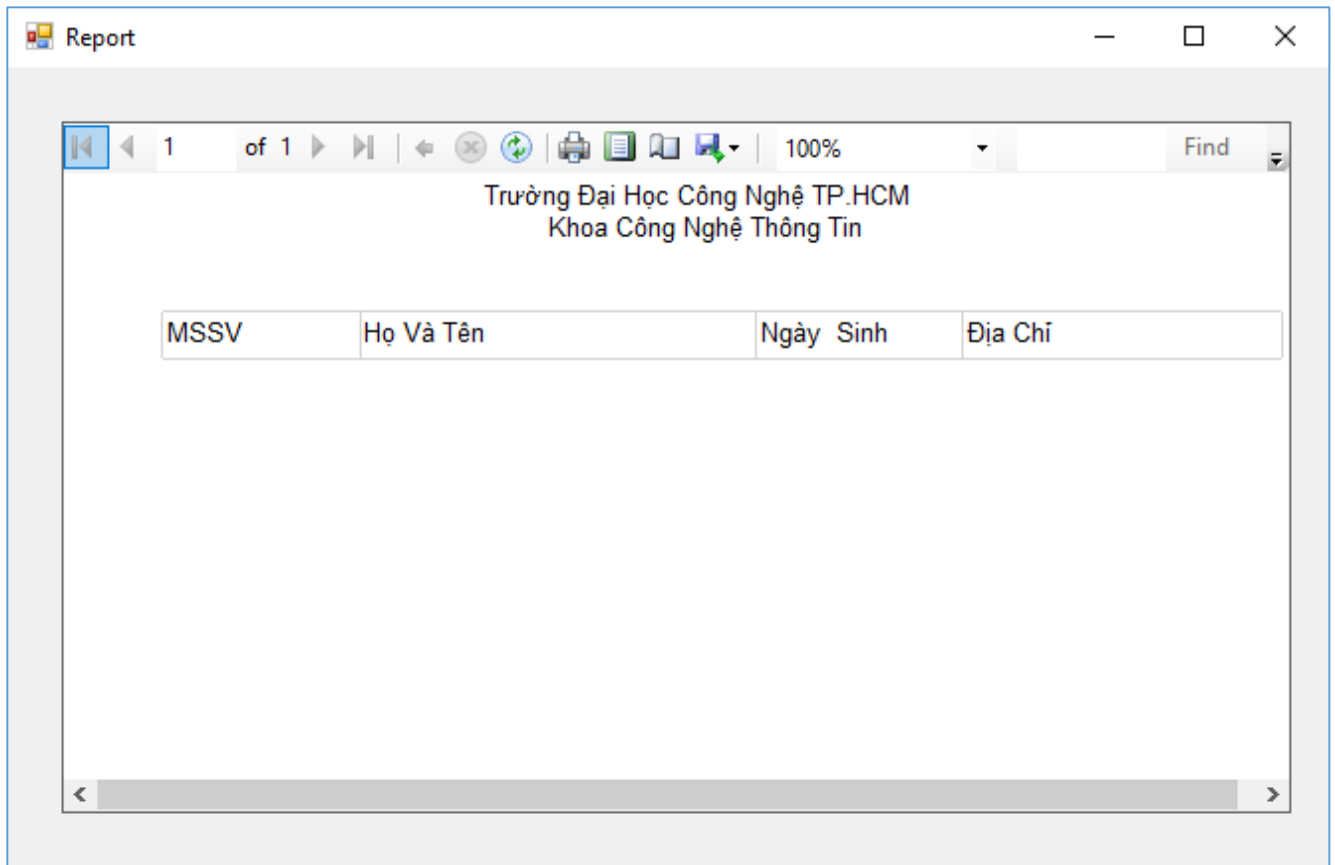


- ✓ Bước 4: Sử dụng ReportViewer để gọi reportData vừa thiết kế

- Kéo ReportView vào form, Chọn tên file report data sử dụng đã thiết kế



- Chạy thử



- Sử dụng code để thêm list dữ liệu

Giả sử có hàm lấy list dữ liệu tạm như sau

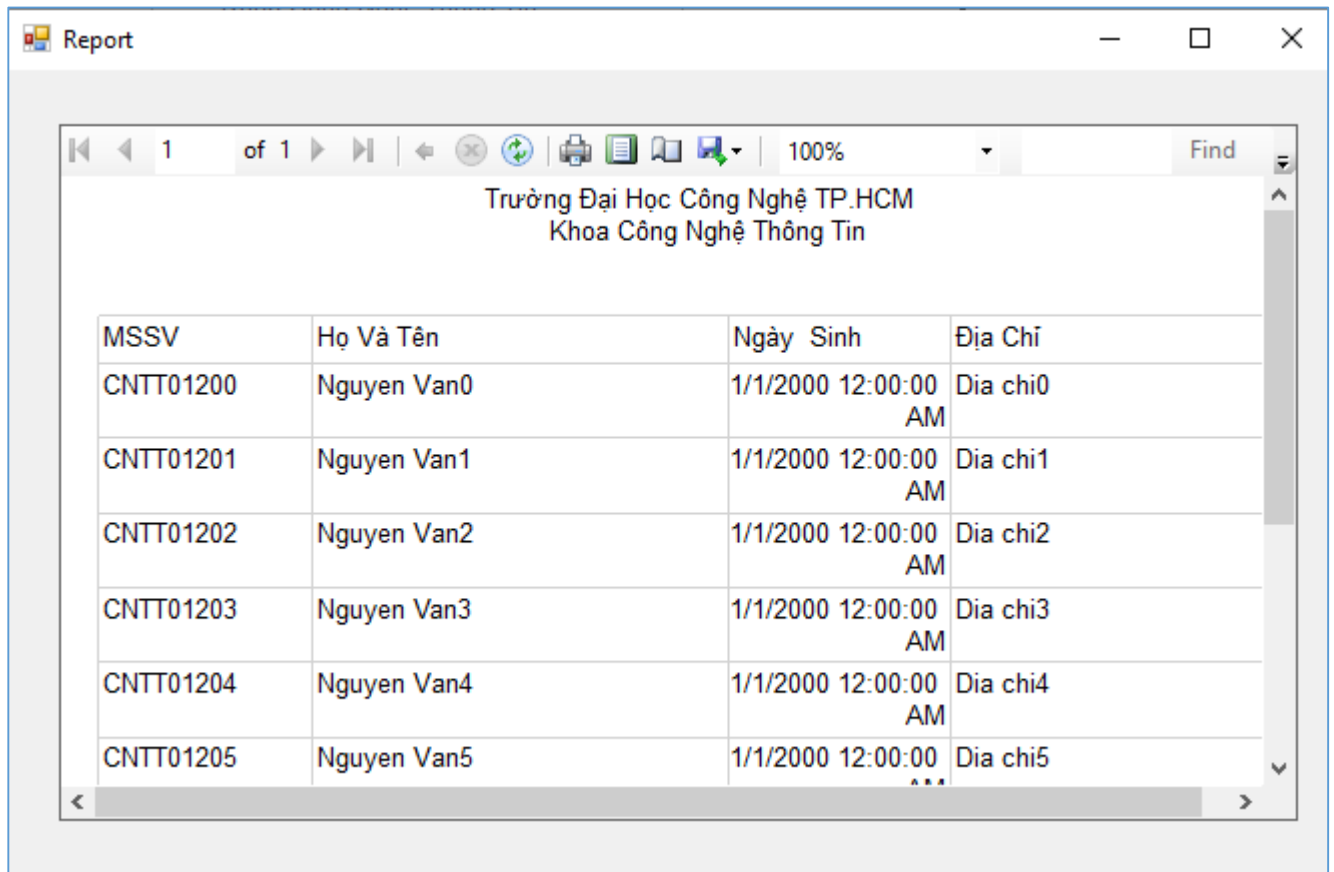
```
public List<Student> GetTempListStudent()
{
    List<Student> listStudent = new List<Student>();
    for(int i=0;i<10;i++)
    {
        Student temp = new Student();
        temp.StudentID = "CNTT0120" + i;
        temp.FullName = "Nguyen Van" + i;
        temp.BirthDay = new DateTime(2000, 1, 1);
        temp.Address = "Dia chi" + i;

        listStudent.Add(temp);
    }
    return listStudent;
}
```

- Viết sự kiện ở form Load

```
List<Student> listStudent = GetTempListStudent();
ReportDataSource rds = new ReportDataSource("dataSetStudent", listStudent);
this.reportViewer1.LocalReport.DataSources.Clear();
this.reportViewer1.LocalReport.DataSources.Add(rds);
this.reportViewer1.RefreshReport();
```

- Chạy lại chương trình



Nếu sử dụng lấy DataSet từ Database, có thể sử dụng ADO.NET để lấy giá trị

```
SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["QuanLySinhVienConnectionString"].Conne
ctionString);
SqlDataAdapter da = new SqlDataAdapter("select * from Student", con);
DataTable dt = new DataTable("DataSet1");
da.Fill(dt);

ReportDataSource rds = new ReportDataSource("DataSet1", dt);
this.reportViewer1.LocalReport.DataSources.Clear();
this.reportViewer1.LocalReport.DataSources.Add(rds);
this.reportViewer1.RefreshReport();
```


C. BÀI TẬP:

Xây dựng chương trình quản lý đơn hàng đơn giản như sau

- ✓ Sử dụng cơ sở dữ liệu SQL server có 3 bảng **Product**, **Order**, **Invoice** lần lượt như sau

Product: Lưu trữ thông tin sản phẩm (Mã sản phẩm, Tên Sản phẩm, Đơn vị Tính, Giá Mua, Giá Bán)

Order: Lưu trữ chi tiết thông tin đơn hàng (Số HĐ, Số TT, Mã SP, Tên SP, ĐVT, Đơn giá, Số lượng)

Invoice: Lưu trữ thông tin hóa đơn đặt hàng (Số HĐ, Ngày đặt hàng, ngày giao hàng, ghi chú)

```
USE [ProductOrder]
GO
/***** Object: Table [dbo].[Product]      Script Date: 09/14/2019 14:10:52
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Product] (
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NOT NULL,
    [Unit] [nvarchar](20) NOT NULL,
    [BuyPrice] [decimal](18, 0) NULL,
    [SellPrice] [decimal](18, 0) NULL,
    CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Order]      Script Date: 09/14/2019 14:10:52 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Order] (
    [InvoiceNo] [nvarchar](20) NOT NULL,
    [No] [int] NOT NULL,
    [ProductID] [nvarchar](20) NOT NULL,
    [ProductName] [nvarchar](100) NULL,
    [Unit] [nvarchar](20) NULL,
    [Price] [decimal](18, 0) NOT NULL,
    [Quantity] [int] NOT NULL,
    CONSTRAINT [PK_Order] PRIMARY KEY CLUSTERED
(
    [InvoiceNo] ASC,
    [No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Invoice]      Script Date: 09/14/2019 14:10:52
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Invoice] (
    [InvoiceNo] [nvarchar](20) NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [DeliveryDate] [datetime] NOT NULL,
    [Note] [nvarchar](255) NULL,
```

```

CONSTRAINT [PK_Invoice] PRIMARY KEY CLUSTERED
(
    [InvoiceNo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

✓ Thiết kế giao diện cho các form tương tự như sau

1. Form giao diện chính xem tất cả các đơn hàng

- Có 2 menu Đơn Hàng và Sản Phẩm

Khi click vào Nhập Đơn Hàng: Tới chi tiết đơn hàng ở Form Quản Lý Đơn Hàng (2)

Khi click vào Quản lý Sản Phẩm: Tới chi tiết form Quản lý sản phẩm (3)

Khi click vào Báo giá xuất report file cho báo giá các sản phẩm (theo mẫu sau)

Report				
1 of 1				
100%				
Find Next				
NHÀ PHÂN PHỐI SẢN PHẨM ABC Số:30/4A Kp2, Trung Mỹ Tây 13,P.Trung Mỹ Tây, Q12,TP Hồ Chí Minh Email: EMAIL@gmail.com ĐT: 09XXXXXXX				
<h2 style="text-align: center;">BẢNG BÁO GIÁ</h2>				
STT	Product ID	Product Name	Unit	Sell Price
1	Product1	Sản phẩm 1	Cái	120000
2	Product2	Sản phẩm 2	Cái	120000
3	Product3	Sản phẩm 3	Cái	70000
4	Product4	Sản phẩm 4	Hộp	300000

- Thông tin tìm kiếm đơn hàng: Tự động tìm kiếm theo thông tin nhập liệu

Người sử dụng có thể xem tất cả các đơn hàng trong 1 khoảng thời gian giao hàng tìm kiếm (mặc định ban đầu là trong ngày hiện tại). Khi ngày thay đổi thì dữ liệu search tương ứng thay đổi
 Nếu checked vào CheckBox (Xem tất cả trong tháng) thì tháng thì khoảng thời gian giao hàng sẽ từ đầu tháng hiện tại tới cuối tháng hiện tại

Tự động tìm kiếm và hiển thị dữ liệu ra DataGridView ở phía dưới

- Nút thêm đơn hàng: Tới form quản lý đơn hàng (2)
- Nút In/Xem: Người dùng chọn 1 dòng trên GridView để xem báo cáo thống kê cho chi tiết 1 đơn hàng (giống ở phần xem report ở quản lý đơn hàng)

2. Form Quản lý đơn hàng

Phần đầu cho phép nhập liệu số hóa đơn , ngày đặt hàng, ngày giao hàng, ghi chú

Phần Dữ liệu DataGridView nhập chi tiết cho từng loại sản phẩm

Cho phép ở từng dòng chọn loại sản phẩm, tự động hiển thị lại thông tin của sản phẩm

Khi có giá trị số lượng thì tự động tính thành tiền = Số lượng * Đơn giá bán

Quản Lý Đơn Hàng

THÔNG TIN ĐƠN HÀNG

Theo hóa đơn số

Ngày Đặt Hàng Ngày Giao Hàng

Ghi Chú

STT	Mã SP	Tên Sản Phẩm	ĐVT	Số Lượng	Giá Bán	Thành Tiền
▶*						

In Đặt Hàng

Tổng Cộng

- **Đặt hàng:** Kiểm tra thông tin đặt hàng phải có hóa đơn, ngày đặt hàng, ngày giao hàng và ít nhất có 1 sản phẩm trong DataGridView

Khi thông tin kiểm tra hợp lệ thì insert dữ liệu vào 2 bảng Order và Invoice

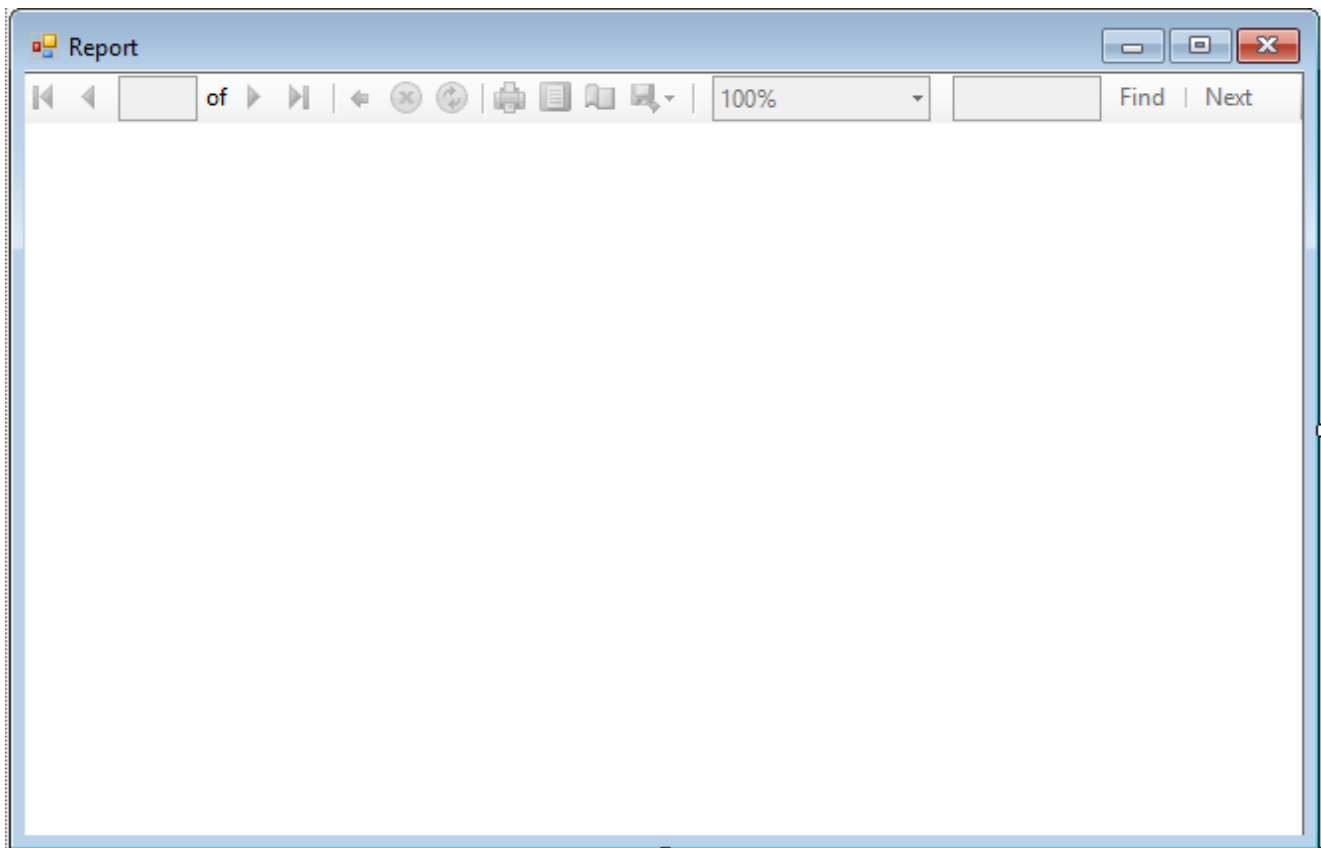
- **Tổng Cộng:** Tổng thành tiền của tất cả sản phẩm

- **In/Xem:** Xuất báo cáo Report của đơn hàng như mẫu sau

- + Kiểm tra các thông tin bắt buộc phải nhập (Mã SP, Tên SP, ĐVT, Giá Mua, Giá Bán).
- + Nếu mã sản phẩm chưa có trong cơ sở dữ liệu thì thêm mới và thông báo
- Nếu mã sản phẩm đã tồn tại trong cơ sở dữ liệu thì cập nhật và thông báo
- Nút Xóa: Kiểm tra mã sản phẩm đã tồn tại và thông báo đã xóa trong cơ sở dữ liệu.

Hướng Dẫn:

- Thiết kế giao diện như yêu cầu
- Sử dụng Entity Framework để tạo model
- Thiết kế 1 form Report chứa report viewer dung chung cho cả Báo giá và chi tiết đơn hàng



```

public frmReport(string invoiceNo)
{
    InitializeComponent();

    if (invoiceNo == null) //TH báo giá
    {
        List<Product> listProduct = Product.GetListProduct();
        this.reportViewer1.LocalReport.ReportPath = "rptProduct.rdlc"; //nhớ
        copy report ra debug
        var reportDataSource = new ReportDataSource("ProductDataSet",
listProduct);
        this.reportViewer1.LocalReport.DataSources.Add(reportDataSource);
        this.reportViewer1.LocalReport.DisplayName = "Bảng Báo Giá";
    }
    else //TH chi tiết đơn hàng
    {
        Invoice invoice = Invoice.GetInvoice(invoiceNo);
        List<Order> listOrder = Order.GetListOrder(invoiceNo);
    }
}

```

```

        if (invoice == null || listOrder.Count() == 0)
        {
            MessageBox.Show("Không tìm thấy đơn hàng");
            return;
        }

        //Sử dụng 2 tham số truyền vào cho report
        ReportParameter[] param = new ReportParameter[2];
        param[0] = new ReportParameter("InvoiceNo", invoiceNo);
        param[1] = new ReportParameter("DeliveryDateStr", string.Format("Ngày " +
invoice.DeliveryDate.ToString("dd/MM/yyyy"))));

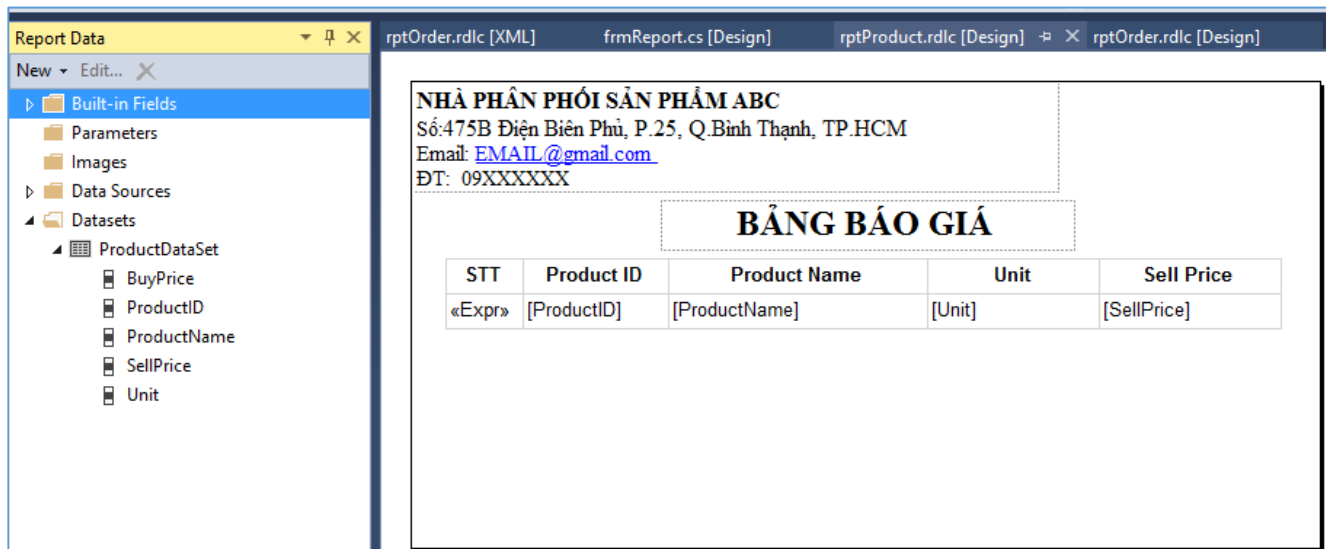
        this.reportViewer1.LocalReport.ReportPath = "rptOrder.rdlc"; //nhớ copy
report ra debug
        this.reportViewer1.LocalReport.SetParameters(param);

        var reportDataSource = new ReportDataSource("OrderDataSet", listOrder);
        this.reportViewer1.LocalReport.DataSources.Add(reportDataSource);

        //tên hiển thị
        this.reportViewer1.LocalReport.DisplayName = "Phiếu giao hàng";
    }
    this.reportViewer1.RefreshReport();
}

```

- Thiết kế Data Report cho bảng báo giá



Số thứ tự sử dụng Expression: =RowNumber([nothing](#))

- Thiết kế Data Report cho phiếu giao hàng


```

        //Sinh viên tự tính tổng tiền
        // dgvInvoice.Rows[row].Cells["Amount"].Value =
    }
}

```

- Hướng dẫn làm form đặt hàng

Thiết kế DataGridView với các column: No, ProductID, ProductName (combobox), Unit, Quantity, Price, Amount

```

private void frmOrder_Load(object sender, EventArgs e)
{
    try{

        ListProduct = Product.GetListProduct();
        DataGridViewComboBoxColumn cmbProduct =
        (DataGridViewComboBoxColumn)dgvExport.Columns[2];
        cmbProduct.DataSource = ListProduct;
        cmbProduct.ValueMember = "ProductID";
        cmbProduct.DisplayMember = "ProductName";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void dgvExport_RowPostPaint(object sender, DataGridViewRowPostPaintEventArgs e)
{
    this.dgvExport.Rows[e.RowIndex].Cells[0].Value = (e.RowIndex + 1).ToString();
}

private void dgvExport_EditingControlShowing(object sender,
DataGridViewEditingControlShowingEventArgs e)
{
    try
    {
        if (dgvExport.CurrentCell.ColumnIndex == 2 && e.Control is ComboBox)
        //khi gặp combobox sản phẩm
        {
            ComboBox combobox = e.Control as ComboBox;
            combobox.DropDownStyle = ComboBoxStyle.DropDown;
            combobox.SelectedIndexChanged += productName_SelectionChanged;
        }
        if (dgvExport.CurrentCell.ColumnIndex == 4) //khi thay đổi quantity
        {
            SetAmount(dgvExport.CurrentCellAddress);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void productName_SelectionChanged(object sender, EventArgs e)
{
    try
    {
        var currentcell = dgvExport.CurrentCellAddress;
        var sendingCB = sender as DataGridViewComboBoxEditingControl;
        if (sendingCB.SelectedValue != null)
        {
            Product findProduct = ListProduct.FirstOrDefault(p => p.ProductID ==
            sendingCB.SelectedValue);

```

```

        if(findProduct!= null)
        {
            DataGridViewTextBoxCell cellProductID =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["ProductID"];
            DataGridViewTextBoxCell cellUnit=
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Unit"];
            DataGridViewTextBoxCell cellQuantity =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Quantity"];
            DataGridViewTextBoxCell cellPrice =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Price"];

            cellProductID.Value = findProduct.ProductID;
            cellUnit.Value = findProduct.Unit;
            cellPrice.Value = findProduct.SellPrice;
            SetAmount(currentcell);
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
private void SetAmount(Point currentcell)
{
    DataGridViewTextBoxCell cellQuantity =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Quantity"];
    DataGridViewTextBoxCell cellPrice =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Price"];
    DataGridViewTextBoxCell cellAmount =
(DataGridViewTextBoxCell)dgvExport.Rows[currentcell.Y].Cells["Amount"];
    cellAmount.Value = "";
    if (cellPrice.Value!= null && cellQuantity.Value!= null)
    {
        decimal amount = Convert.ToDecimal(cellPrice.Value) *
Convert.ToDecimal(cellQuantity.Value);
        cellAmount.Value = amount.ToString();
    }
}
}

```

-----Hết Lab 05-----