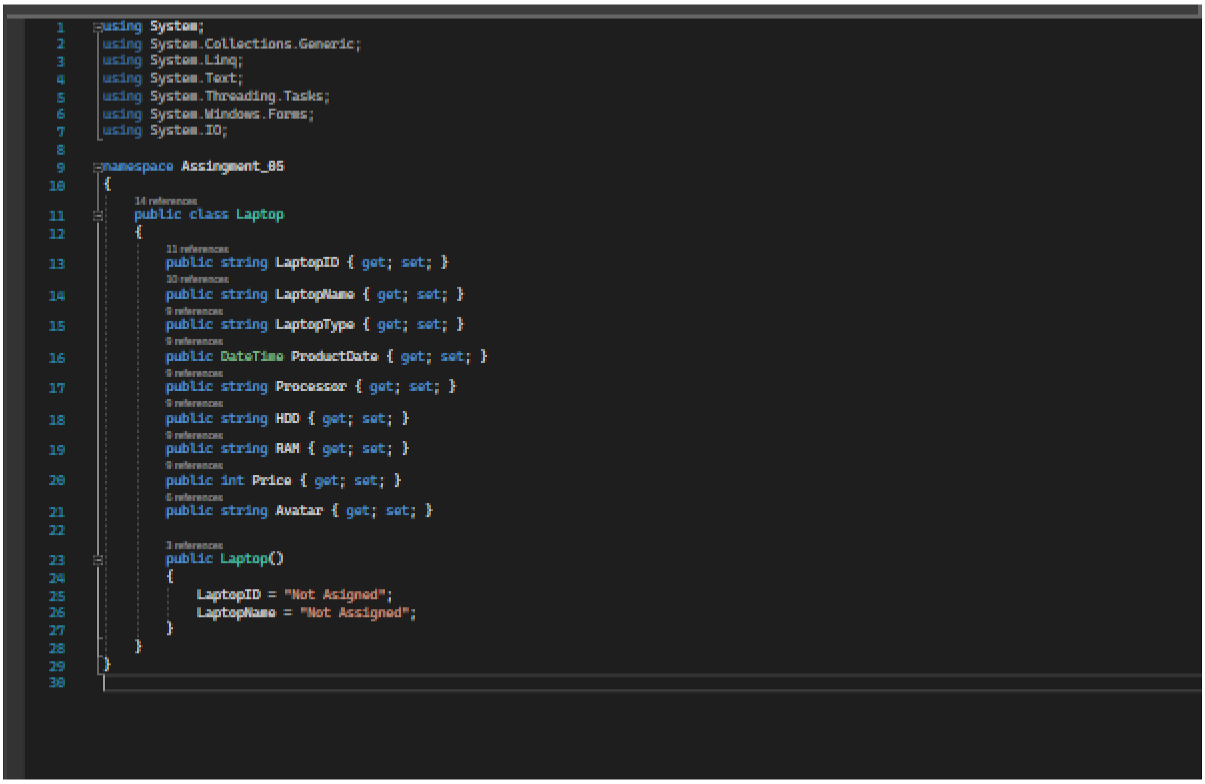
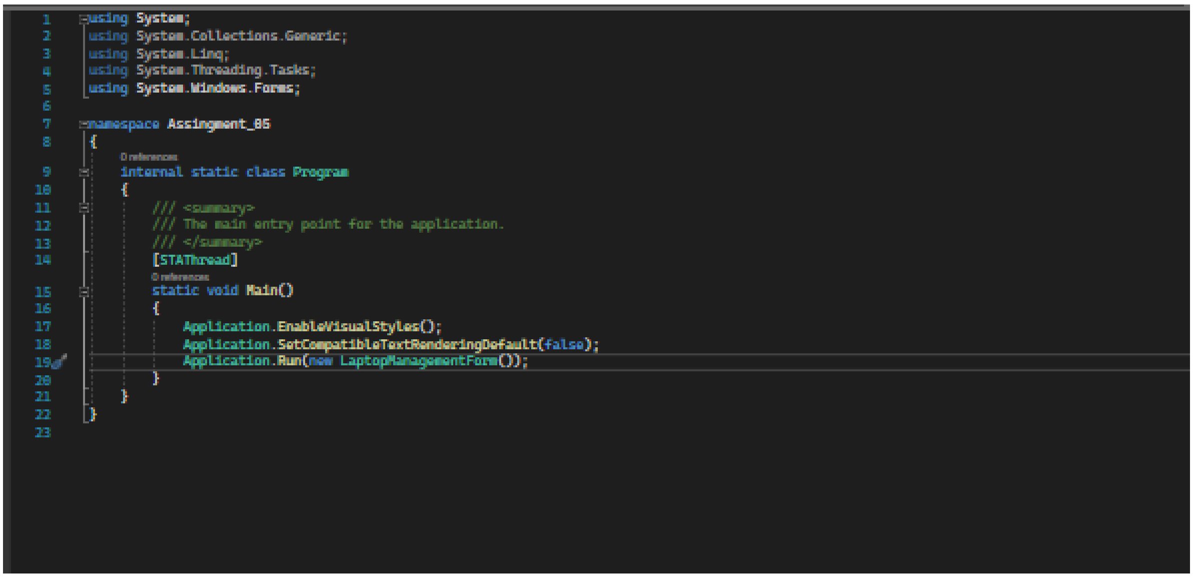
Laptop Management

Laptop.cs



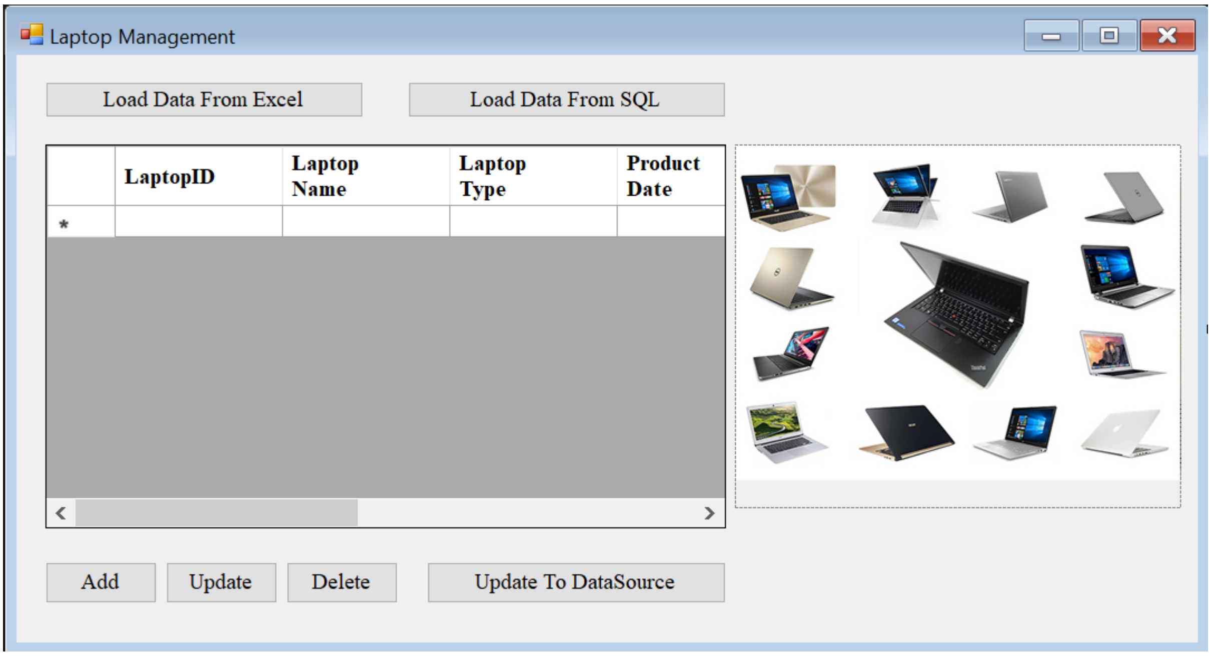
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace Assingment_05
{
    14 references
    public class Laptop
    {
        11 references
        public string LaptopID { get; set; }
        10 references
        public string LaptopName { get; set; }
        9 references
        public string LaptopType { get; set; }
        9 references
        public DateTime ProductDate { get; set; }
        9 references
        public string Processor { get; set; }
        9 references
        public string HDD { get; set; }
        9 references
        public string RAM { get; set; }
        9 references
        public int Price { get; set; }
        6 references
        public string Avatar { get; set; }

        3 references
        public Laptop()
        {
            LaptopID = "Not Asigned";
            LaptopName = "Not Assigned";
        }
    }
}
```

Program.cs

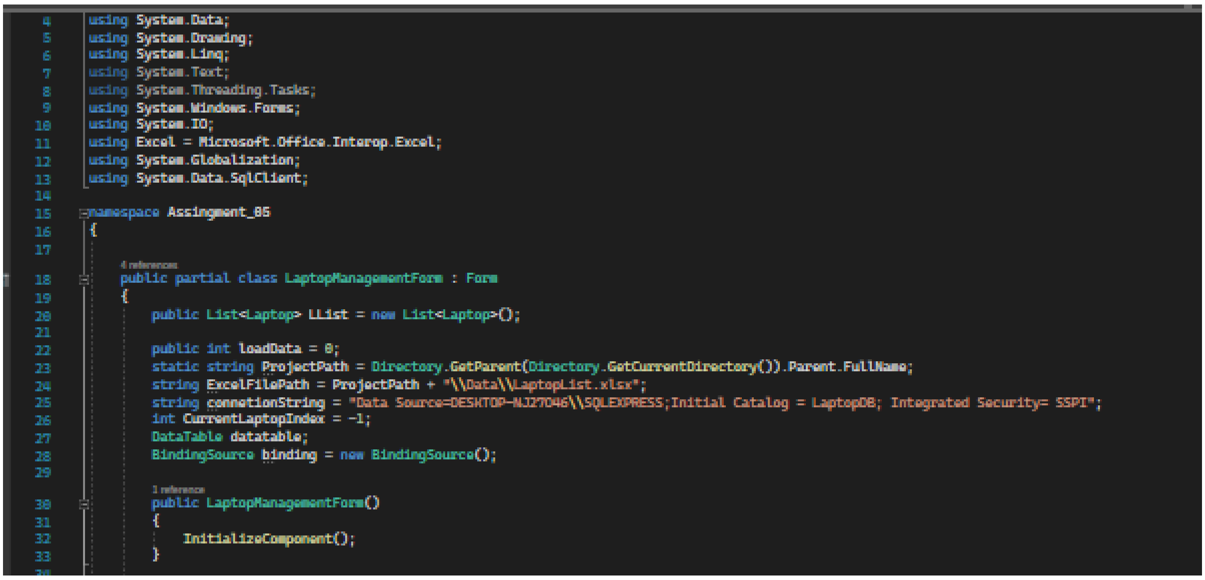


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Assingment_05
{
    0 references
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LaptopManagementForm());
        }
    }
}
```

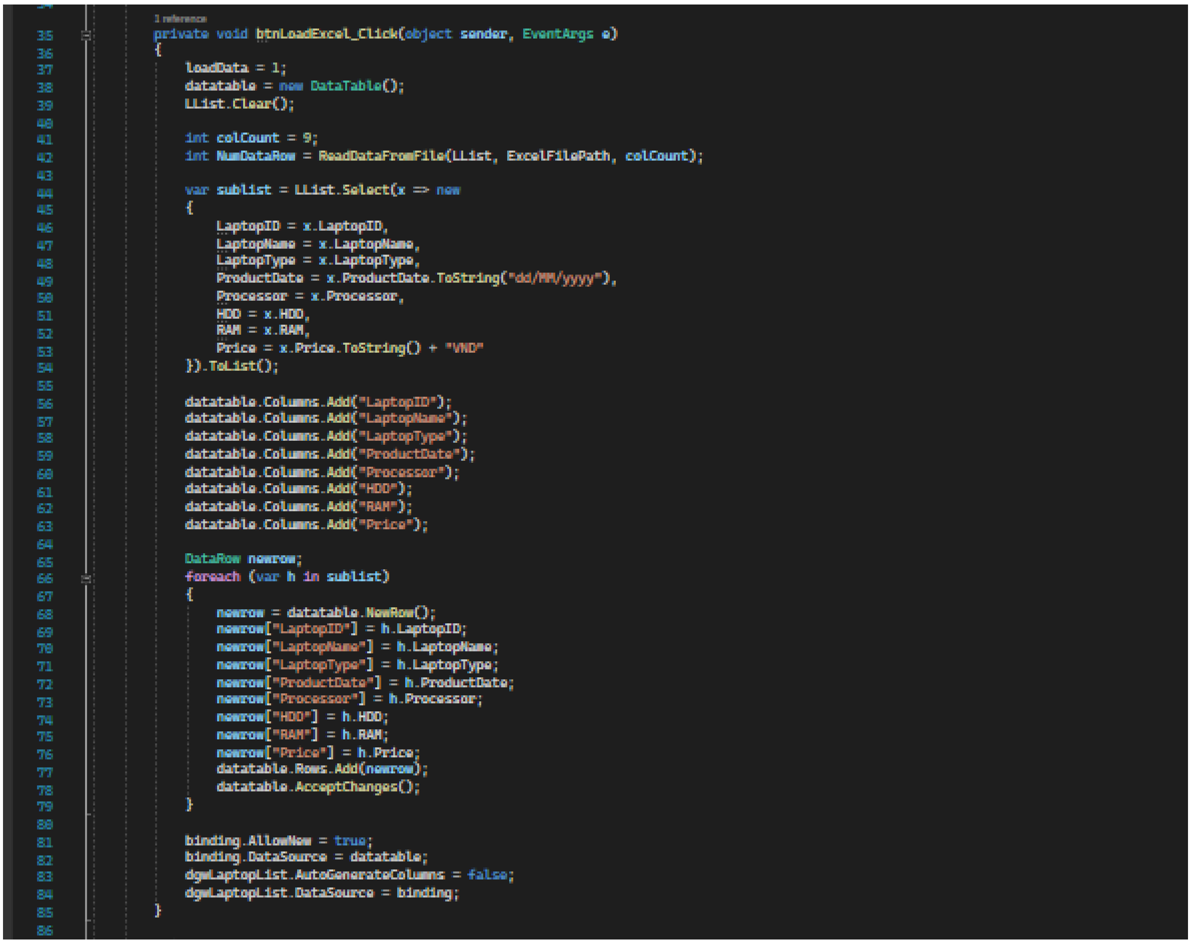
Design



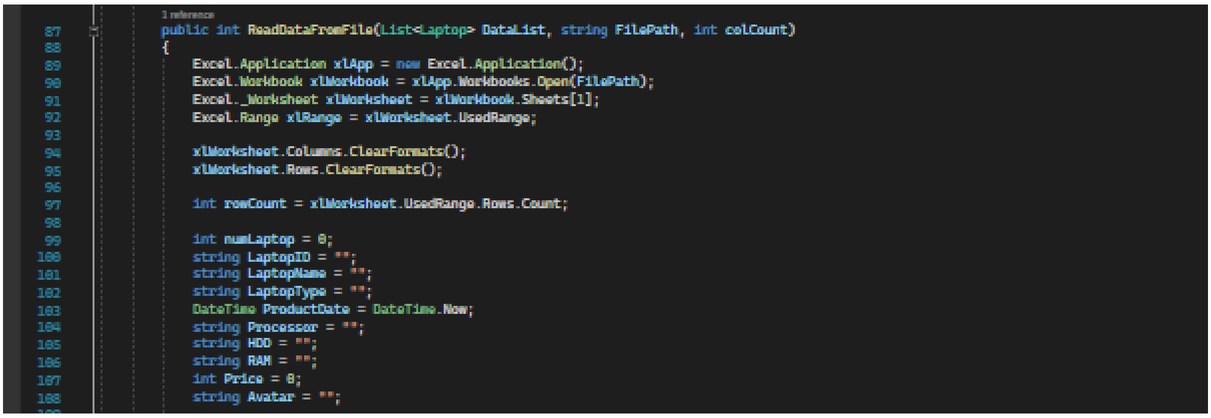
LaptopManagement.cs



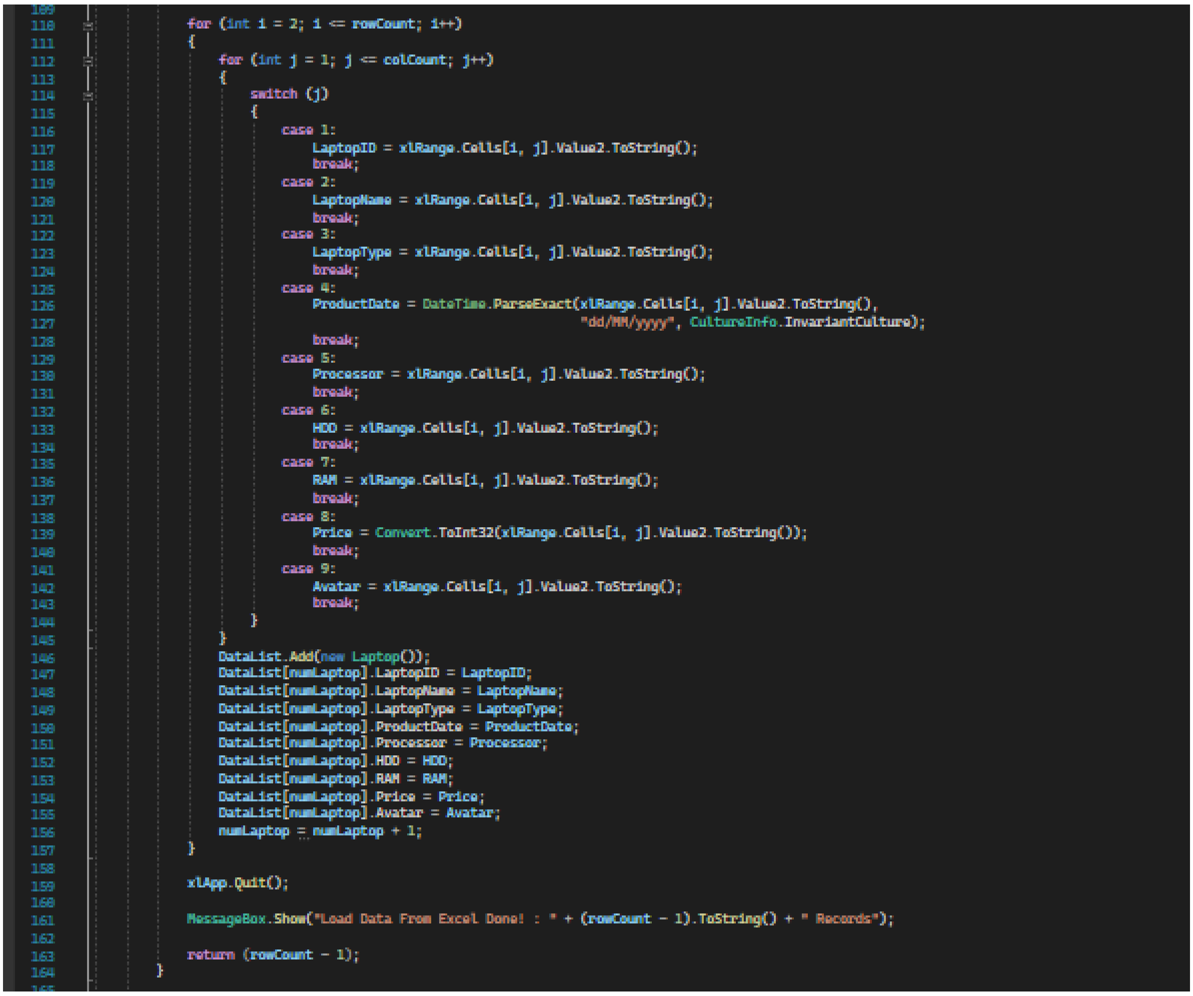
```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using Excel = Microsoft.Office.Interop.Excel;
using System.Globalization;
using System.Data.SqlClient;

namespace Assingment_05
{

    public partial class LaptopManagementForm : Form
    {
        public List<Laptop> LList = new List<Laptop>();

        public int loadData = 0;
        static string ProjectPath = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.FullName;
        string ExcelFilePath = ProjectPath + "\\Data\\LaptopList.xlsx";
        string connetionString = "Data Source=DESKTOP-NJ27O46\\SQLEXPRESS;Initial Catalog = LaptopDB; Integrated Security= SSPI";
        int CurrentLaptopIndex = -1;
        DataTable datatable;
        BindingSource binding = new BindingSource();

        public LaptopManagementForm()
        {
            InitializeComponent();
        }
```



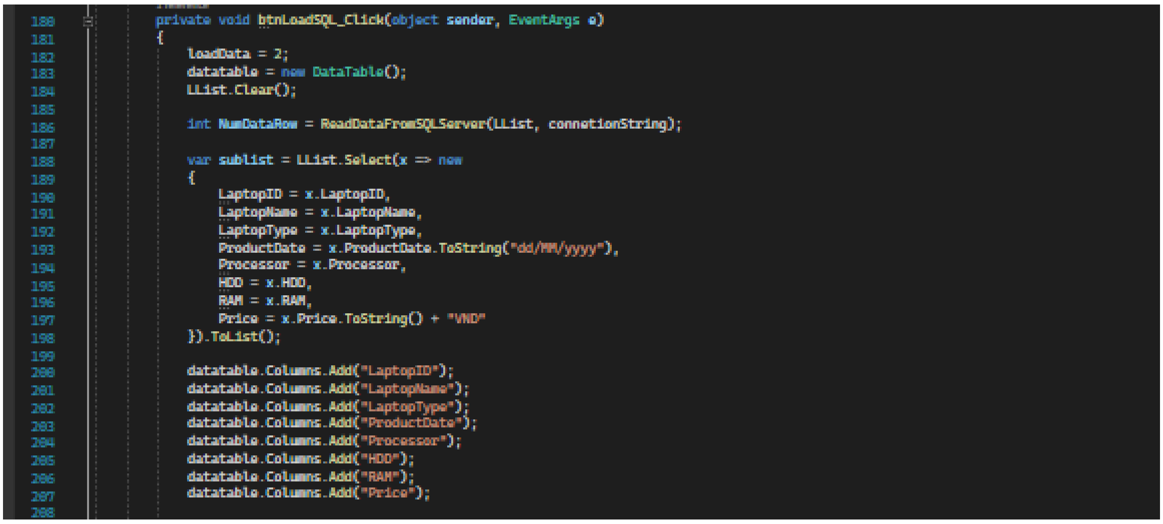
```
private void btnLoadExcel_Click(object sender, EventArgs e)
{
    loadData = 1;
    datatable = new DataTable();
    LList.Clear();

    int colCount = 9;
    int NumDataRow = ReadDataFromFile(LList, ExcelFilePath, colCount);

    var sublist = LList.Select(x => new
    {
        LaptopID = x.LaptopID,
        LaptopName = x.LaptopName,
        LaptopType = x.LaptopType,
        ProductDate = x.ProductDate.ToString("dd/MM/yyyy"),
        Processor = x.Processor,
        HDD = x.HDD,
        RAM = x.RAM,
        Price = x.Price.ToString() + "VND"
    }).ToList();

    datatable.Columns.Add("LaptopID");
    datatable.Columns.Add("LaptopName");
    datatable.Columns.Add("LaptopType");
    datatable.Columns.Add("ProductDate");
    datatable.Columns.Add("Processor");
    datatable.Columns.Add("HDD");
    datatable.Columns.Add("RAM");
    datatable.Columns.Add("Price");

    DataRow newrow;
    foreach (var h in sublist)
    {
        newrow = datatable.NewRow();
        newrow["LaptopID"] = h.LaptopID;
        newrow["LaptopName"] = h.LaptopName;
        newrow["LaptopType"] = h.LaptopType;
        newrow["ProductDate"] = h.ProductDate;
        newrow["Processor"] = h.Processor;
        newrow["HDD"] = h.HDD;
        newrow["RAM"] = h.RAM;
        newrow["Price"] = h.Price;
        datatable.Rows.Add(newrow);
        datatable.AcceptChanges();
    }

    binding.AllowNew = true;
    binding.DataSource = datatable;
    dgwLaptopList.AutoGenerateColumns = false;
    dgwLaptopList.DataSource = binding;
}
```



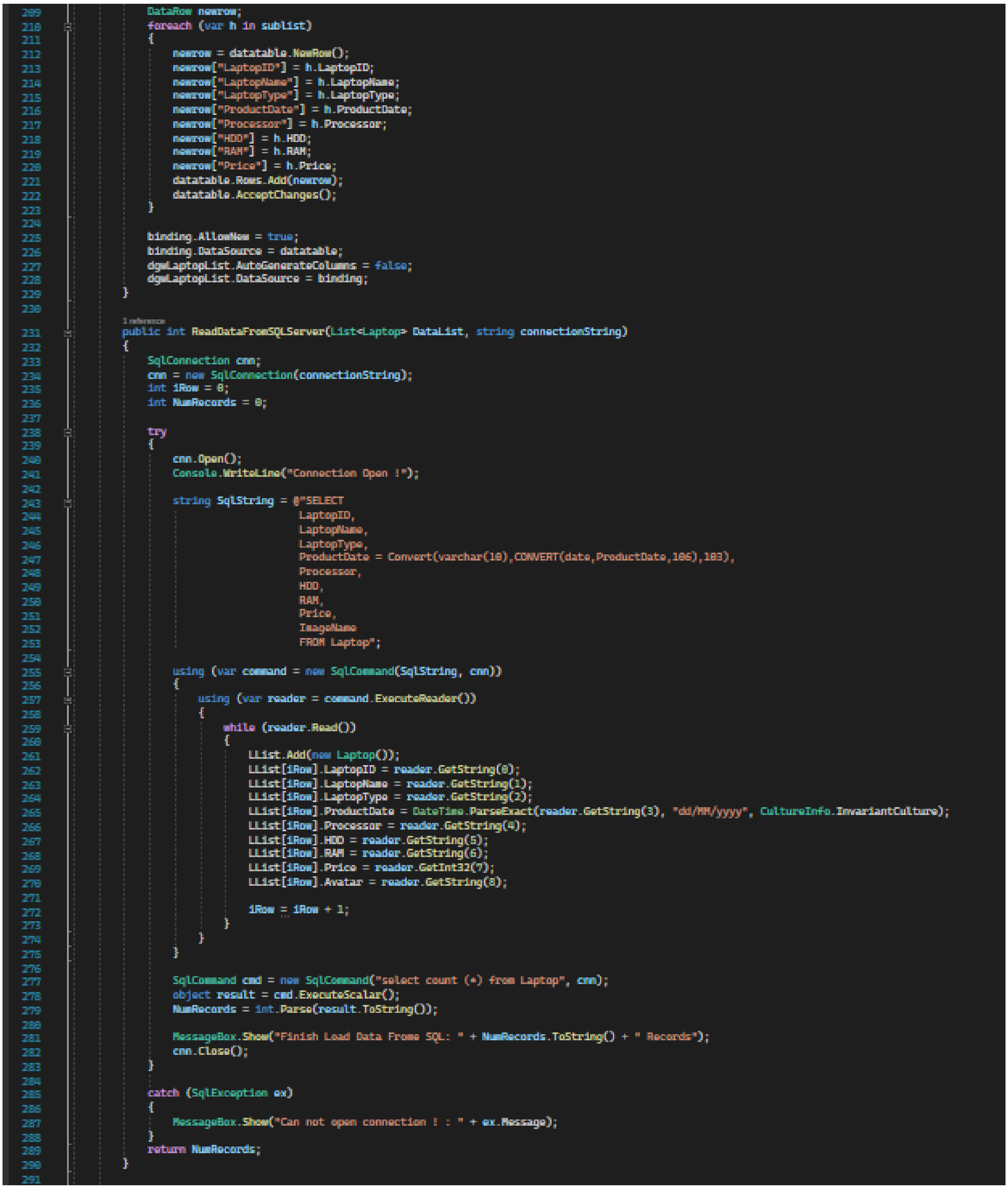
```
public int ReadDataFromFile(List<Laptop> DataList, string FilePath, int colCount)
{
    Excel.Application xlApp = new Excel.Application();
    Excel.Workbook xlWorkbook = xlApp.Workbooks.Open(FilePath);
    Excel._Worksheet xlWorksheet = xlWorkbook.Sheets[1];
    Excel.Range xlRange = xlWorksheet.UsedRange;

    xlWorksheet.Columns.ClearFormats();
    xlWorksheet.Rows.ClearFormats();

    int rowCount = xlWorksheet.UsedRange.Rows.Count;

    int numLaptop = 0;
    string LaptopID = "";
    string LaptopName = "";
    string LaptopType = "";
    DateTime ProductDate = DateTime.Now;
    string Processor = "";
    string HDD = "";
    string RAM = "";
    int Price = 0;
    string Avatar = "";
```



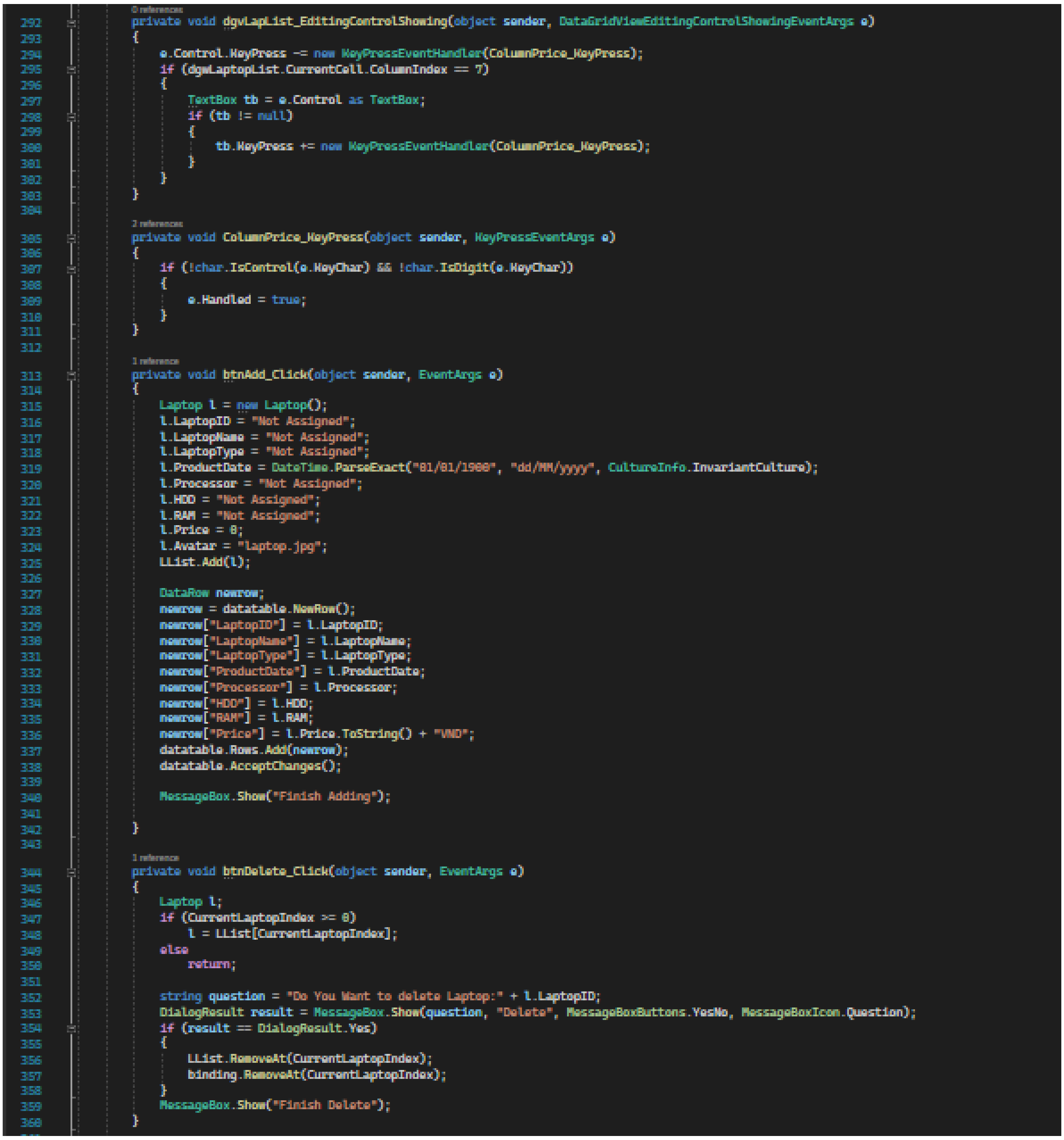
```
            for (int i = 2; i <= rowCount; i++)
            {
                for (int j = 1; j <= colCount; j++)
                {
                    switch (j)
                    {
                        case 1:
                            LaptopID = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 2:
                            LaptopName = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 3:
                            LaptopType = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 4:
                            ProductDate = DateTime.ParseExact(xlRange.Cells[i, j].Value2.ToString(),
                                                              "dd/MM/yyyy", CultureInfo.InvariantCulture);
                            break;
                        case 5:
                            Processor = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 6:
                            HDD = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 7:
                            RAM = xlRange.Cells[i, j].Value2.ToString();
                            break;
                        case 8:
                            Price = Convert.ToInt32(xlRange.Cells[i, j].Value2.ToString());
                            break;
                        case 9:
                            Avatar = xlRange.Cells[i, j].Value2.ToString();
                            break;
                    }
                }
                DataList.Add(new Laptop());
                DataList[numLaptop].LaptopID = LaptopID;
                DataList[numLaptop].LaptopName = LaptopName;
                DataList[numLaptop].LaptopType = LaptopType;
                DataList[numLaptop].ProductDate = ProductDate;
                DataList[numLaptop].Processor = Processor;
                DataList[numLaptop].HDD = HDD;
                DataList[numLaptop].RAM = RAM;
                DataList[numLaptop].Price = Price;
                DataList[numLaptop].Avatar = Avatar;
                numLaptop = numLaptop + 1;
            }

            xlApp.Quit();

            MessageBox.Show("Load Data From Excel Done! : " + (rowCount - 1).ToString() + " Records");

            return (rowCount - 1);
        }
```



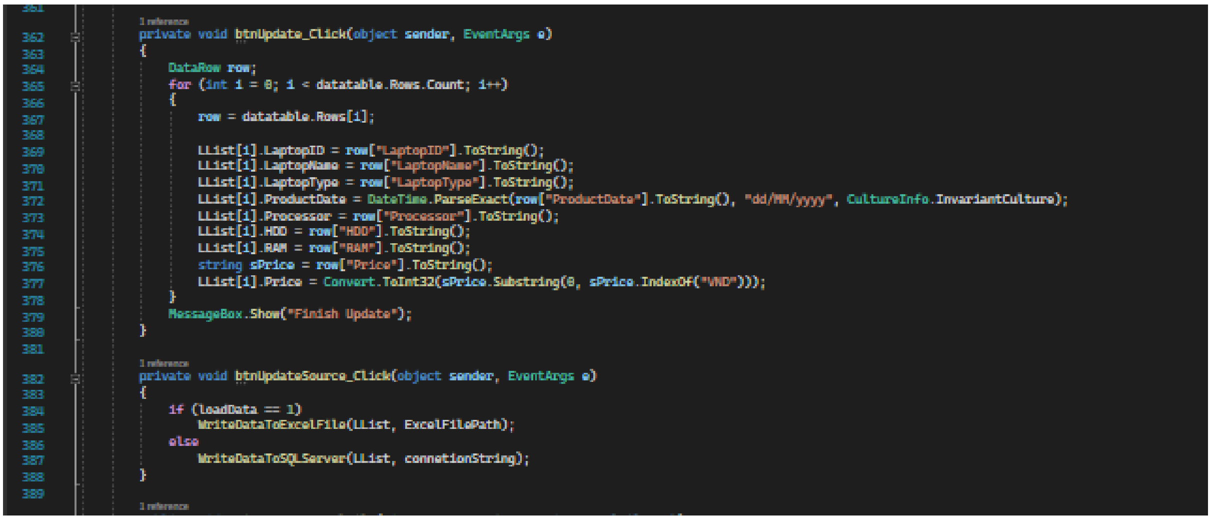
```
        private void btnLoadSQL_Click(object sender, EventArgs e)
        {
            loadData = 2;
            datatable = new DataTable();
            LList.Clear();

            int NumDataRow = ReadDataFromSQLServer(LList, connetionString);

            var sublist = LList.Select(x => new
            {
                LaptopID = x.LaptopID,
                LaptopName = x.LaptopName,
                LaptopType = x.LaptopType,
                ProductDate = x.ProductDate.ToString("dd/MM/yyyy"),
                Processor = x.Processor,
                HDD = x.HDD,
                RAM = x.RAM,
                Price = x.Price.ToString() + "VND"
            }).ToList();

            datatable.Columns.Add("LaptopID");
            datatable.Columns.Add("LaptopName");
            datatable.Columns.Add("LaptopType");
            datatable.Columns.Add("ProductDate");
            datatable.Columns.Add("Processor");
            datatable.Columns.Add("HDD");
            datatable.Columns.Add("RAM");
            datatable.Columns.Add("Price");
```



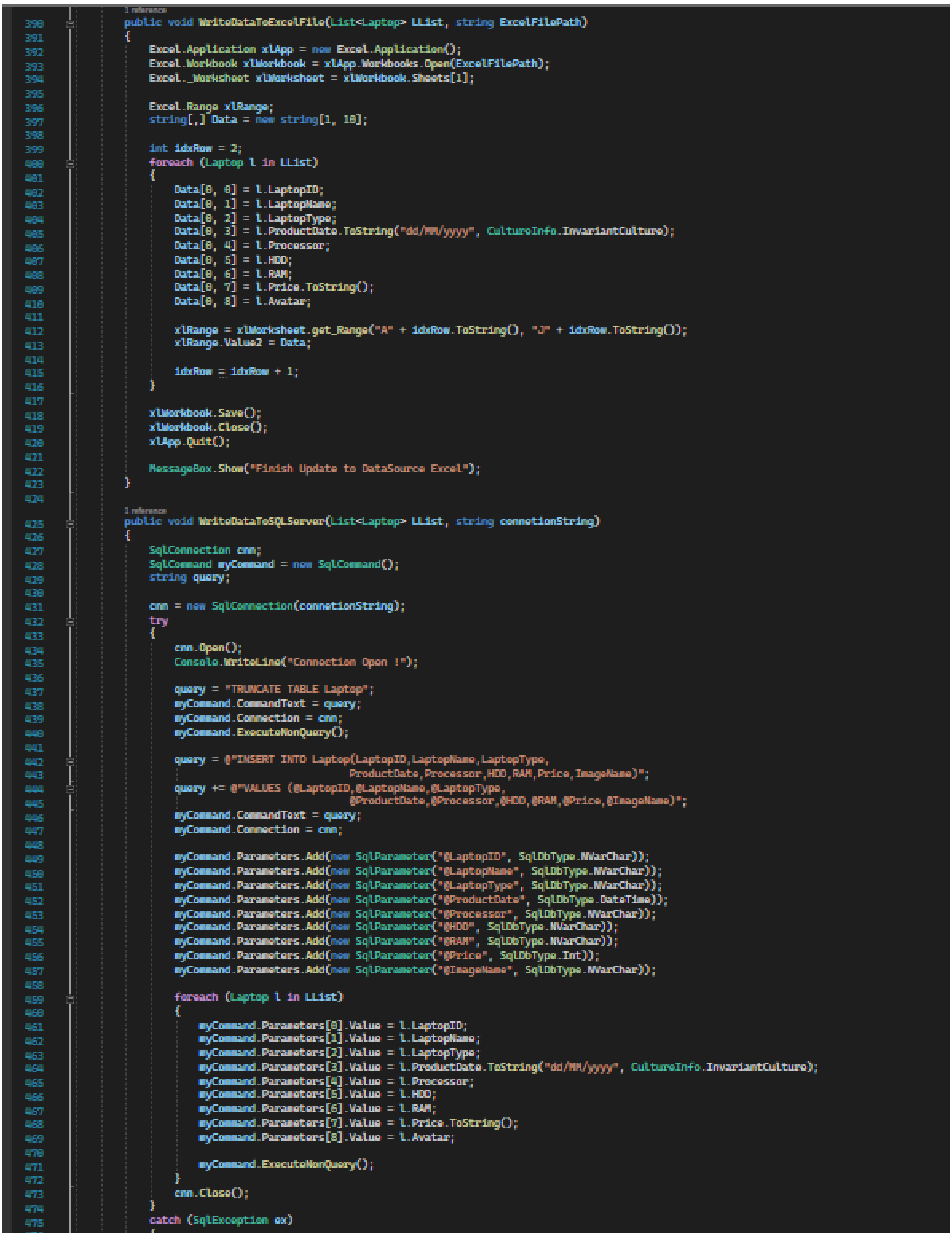
```
            DataRow newrow;
            foreach (var h in sublist)
            {
                newrow = datatable.NewRow();
                newrow["LaptopID"] = h.LaptopID;
                newrow["LaptopName"] = h.LaptopName;
                newrow["LaptopType"] = h.LaptopType;
                newrow["ProductDate"] = h.ProductDate;
                newrow["Processor"] = h.Processor;
                newrow["HDD"] = h.HDD;
                newrow["RAM"] = h.RAM;
                newrow["Price"] = h.Price;
                datatable.Rows.Add(newrow);
                datatable.AcceptChanges();
            }

            binding.AllowNew = true;
            binding.DataSource = datatable;
            dgwLaptopList.AutoGenerateColumns = false;
            dgwLaptopList.DataSource = binding;
        }

        1 reference
        public int ReadDataFromSQLServer(List<Laptop> DataList, string connectionString)
        {
            SqlConnection cnn;
            cnn = new SqlConnection(connectionString);
            int iRow = 0;
            int NumRecords = 0;

            try
            {
                cnn.Open();
                Console.WriteLine("Connection Open !");

                string SqlString = @"SELECT
                                    LaptopID,
                                    LaptopName,
                                    LaptopType,
                                    ProductDate = Convert(varchar(10),CONVERT(date,ProductDate,106),103),
                                    Processor,
                                    HDD,
                                    RAM,
                                    Price,
                                    ImageName
                                    FROM Laptop";

                using (var command = new SqlCommand(SqlString, cnn))
                {
                    using (var reader = command.ExecuteReader())
                    {
                        while (reader.Read())
                        {
                            LList.Add(new Laptop());
                            LList[iRow].LaptopID = reader.GetString(0);
                            LList[iRow].LaptopName = reader.GetString(1);
                            LList[iRow].LaptopType = reader.GetString(2);
                            LList[iRow].ProductDate = DateTime.ParseExact(reader.GetString(3), "dd/MM/yyyy", CultureInfo.InvariantCulture);
                            LList[iRow].Processor = reader.GetString(4);
                            LList[iRow].HDD = reader.GetString(5);
                            LList[iRow].RAM = reader.GetString(6);
                            LList[iRow].Price = reader.GetInt32(7);
                            LList[iRow].Avatar = reader.GetString(8);

                            iRow = iRow + 1;
                        }
                    }
                }

                SqlCommand cmd = new SqlCommand("select count (*) from Laptop", cnn);
                object result = cmd.ExecuteScalar();
                NumRecords = int.Parse(result.ToString());

                MessageBox.Show("Finish Load Data Frome SQL: " + NumRecords.ToString() + " Records");
                cnn.Close();
            }

            catch (SqlException ex)
            {
                MessageBox.Show("Can not open connection ! : " + ex.Message);
            }
            return NumRecords;
        }
```



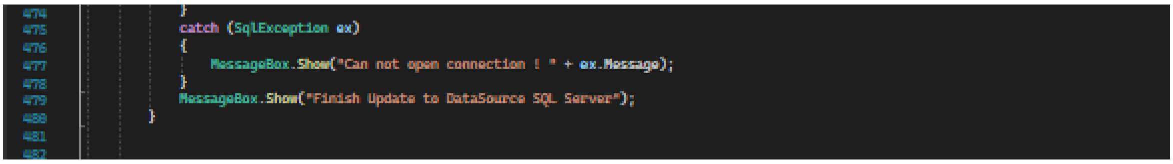
```
        0 references
        private void dgvLapList_EditingControlShowing(object sender, DataGridViewEditingControlShowingEventArgs e)
        {
            e.Control.KeyPress -= new KeyPressEventHandler(ColumnPrice_KeyPress);
            if (dgwLaptopList.CurrentCell.ColumnIndex == 7)
            {
                TextBox tb = e.Control as TextBox;
                if (tb != null)
                {
                    tb.KeyPress += new KeyPressEventHandler(ColumnPrice_KeyPress);
                }
            }
        }

        2 references
        private void ColumnPrice_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
            {
                e.Handled = true;
            }
        }

        1 reference
        private void btnAdd_Click(object sender, EventArgs e)
        {
            Laptop l = new Laptop();
            l.LaptopID = "Not Assigned";
            l.LaptopName = "Not Assigned";
            l.LaptopType = "Not Assigned";
            l.ProductDate = DateTime.ParseExact("01/01/1900", "dd/MM/yyyy", CultureInfo.InvariantCulture);
            l.Processor = "Not Assigned";
            l.HDD = "Not Assigned";
            l.RAM = "Not Assigned";
            l.Price = 0;
            l.Avatar = "laptop.jpg";
            LList.Add(l);

            DataRow newrow;
            newrow = datatable.NewRow();
            newrow["LaptopID"] = l.LaptopID;
            newrow["LaptopName"] = l.LaptopName;
            newrow["LaptopType"] = l.LaptopType;
            newrow["ProductDate"] = l.ProductDate;
            newrow["Processor"] = l.Processor;
            newrow["HDD"] = l.HDD;
            newrow["RAM"] = l.RAM;
            newrow["Price"] = l.Price.ToString() + "VND";
            datatable.Rows.Add(newrow);
            datatable.AcceptChanges();

            MessageBox.Show("Finish Adding");

        }

        1 reference
        private void btnDelete_Click(object sender, EventArgs e)
        {
            Laptop l;
            if (CurrentLaptopIndex >= 0)
                l = LList[CurrentLaptopIndex];
            else
                return;

            string question = "Do You Want to delete Laptop:" + l.LaptopID;
            DialogResult result = MessageBox.Show(question, "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (result == DialogResult.Yes)
            {
                LList.RemoveAt(CurrentLaptopIndex);
                binding.RemoveAt(CurrentLaptopIndex);
            }
            MessageBox.Show("Finish Delete");
        }
```



```
        1 reference
        private void btnUpdate_Click(object sender, EventArgs e)
        {
            DataRow row;
            for (int i = 0; i < datatable.Rows.Count; i++)
            {
                row = datatable.Rows[i];

                LList[i].LaptopID = row["LaptopID"].ToString();
                LList[i].LaptopName = row["LaptopName"].ToString();
                LList[i].LaptopType = row["LaptopType"].ToString();
                LList[i].ProductDate = DateTime.ParseExact(row["ProductDate"].ToString(), "dd/MM/yyyy", CultureInfo.InvariantCulture);
                LList[i].Processor = row["Processor"].ToString();
                LList[i].HDD = row["HDD"].ToString();
                LList[i].RAM = row["RAM"].ToString();
                string sPrice = row["Price"].ToString();
                LList[i].Price = Convert.ToInt32(sPrice.Substring(0, sPrice.IndexOf("VND")));
            }
            MessageBox.Show("Finish Update");
        }

        1 reference
        private void btnUpdateSource_Click(object sender, EventArgs e)
        {
            if (loadData == 1)
                WriteDataToExcelFile(LList, ExcelFilePath);
            else
                WriteDataToSQLServer(LList, connetionString);
        }

        1 reference
```

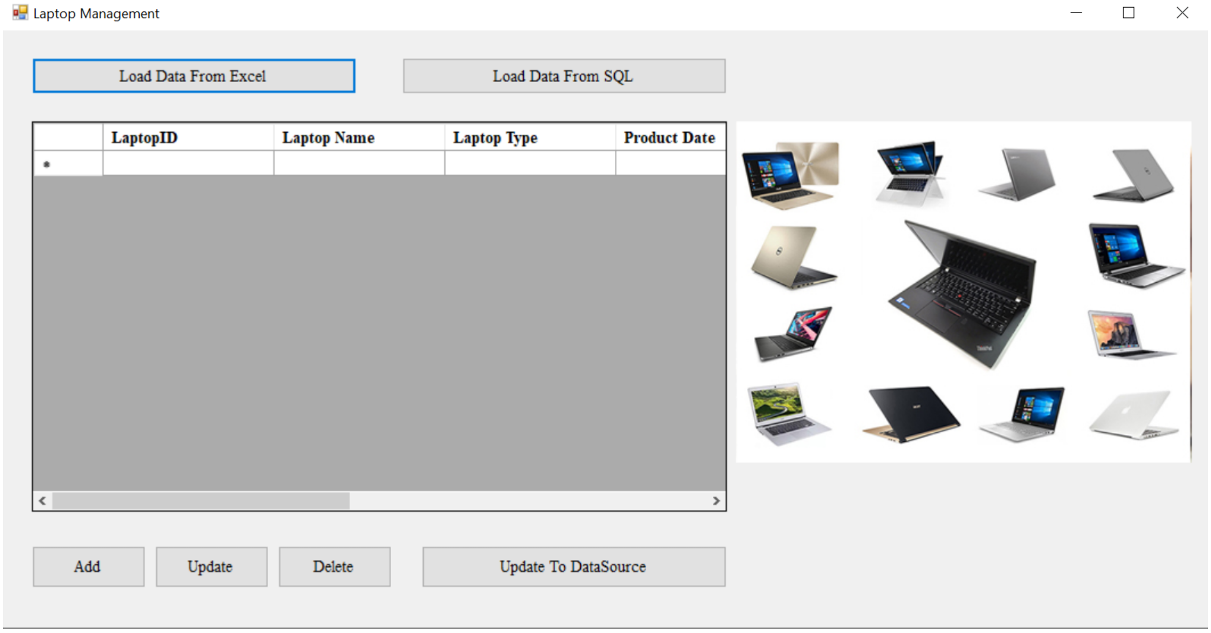


```
1 reference
public void WriteDataToExcelFile(List<Laptop> LList, string ExcelFilePath)
{
    Excel.Application xlApp = new Excel.Application();
    Excel.Workbook xlWorkbook = xlApp.Workbooks.Open(ExcelFilePath);
    Excel._Worksheet xlWorksheet = xlWorkbook.Sheets[1];

    Excel.Range xlRange;
    string[,] Data = new string[1, 10];

    int idxRow = 2;
    foreach (Laptop l in LList)
    {
        Data[0, 0] = l.LaptopID;
        Data[0, 1] = l.LaptopName;
        Data[0, 2] = l.LaptopType;
        Data[0, 3] = l.ProductDate.ToString("dd/MM/yyyy", CultureInfo.InvariantCulture);
        Data[0, 4] = l.Processor;
        Data[0, 5] = l.HDD;
        Data[0, 6] = l.RAM;
        Data[0, 7] = l.Price.ToString();
        Data[0, 8] = l.Avatar;

        xlRange = xlWorksheet.get_Range("A" + idxRow.ToString(), "J" + idxRow.ToString());
        xlRange.Value2 = Data;

        idxRow = idxRow + 1;
    }

    xlWorkbook.Save();
    xlWorkbook.Close();
    xlApp.Quit();

    MessageBox.Show("Finish Update to DataSource Excel");
}

1 reference
public void WriteDataToSQLServer(List<Laptop> LList, string connetionString)
{
    SqlConnection cnn;
    SqlCommand myCommand = new SqlCommand();
    string query;

    cnn = new SqlConnection(connetionString);
    try
    {
        cnn.Open();
        Console.WriteLine("Connection Open !");

        query = "TRUNCATE TABLE Laptop";
        myCommand.CommandText = query;
        myCommand.Connection = cnn;
        myCommand.ExecuteNonQuery();

        query = @"INSERT INTO Laptop(LaptopID,LaptopName,LaptopType,
                                    ProductDate,Processor,HDD,RAM,Price,ImageName)";
        query += @"VALUES (@LaptopID,@LaptopName,@LaptopType,
                                    @ProductDate,@Processor,@HDD,@RAM,@Price,@ImageName)";
        myCommand.CommandText = query;
        myCommand.Connection = cnn;

        myCommand.Parameters.Add(new SqlParameter("@LaptopID", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@LaptopName", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@LaptopType", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@ProductDate", SqlDbType.DateTime));
        myCommand.Parameters.Add(new SqlParameter("@Processor", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@HDD", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@RAM", SqlDbType.NVarChar));
        myCommand.Parameters.Add(new SqlParameter("@Price", SqlDbType.Int));
        myCommand.Parameters.Add(new SqlParameter("@ImageName", SqlDbType.NVarChar));

        foreach (Laptop l in LList)
        {
            myCommand.Parameters[0].Value = l.LaptopID;
            myCommand.Parameters[1].Value = l.LaptopName;
            myCommand.Parameters[2].Value = l.LaptopType;
            myCommand.Parameters[3].Value = l.ProductDate.ToString("dd/MM/yyyy", CultureInfo.InvariantCulture);
            myCommand.Parameters[4].Value = l.Processor;
            myCommand.Parameters[5].Value = l.HDD;
            myCommand.Parameters[6].Value = l.RAM;
            myCommand.Parameters[7].Value = l.Price.ToString();
            myCommand.Parameters[8].Value = l.Avatar;

            myCommand.ExecuteNonQuery();
        }
        cnn.Close();
    }
    catch (SqlException ex)
```

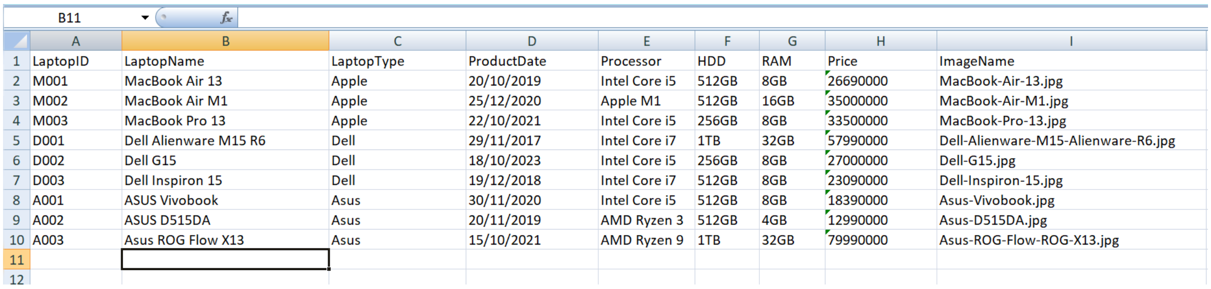


```
    }
    catch (SqlException ex)
```

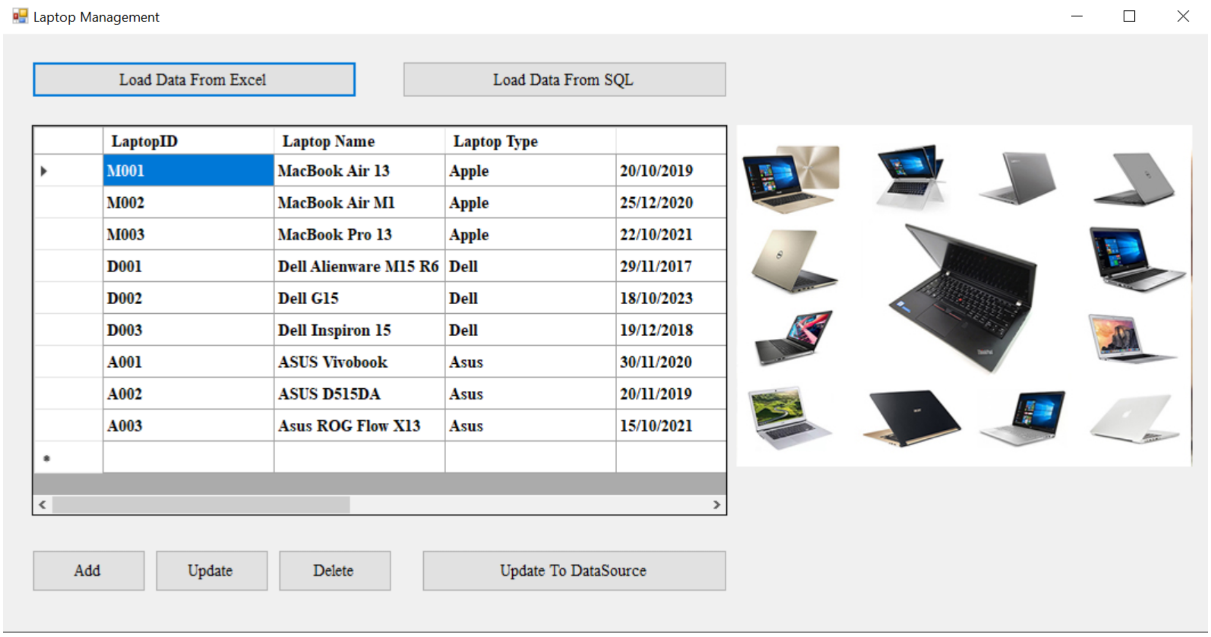
```
    {
        MessageBox.Show("Can not open connection ! " + ex.Message);
    }
    MessageBox.Show("Finish Update to DataSource SQL Server");
}
```



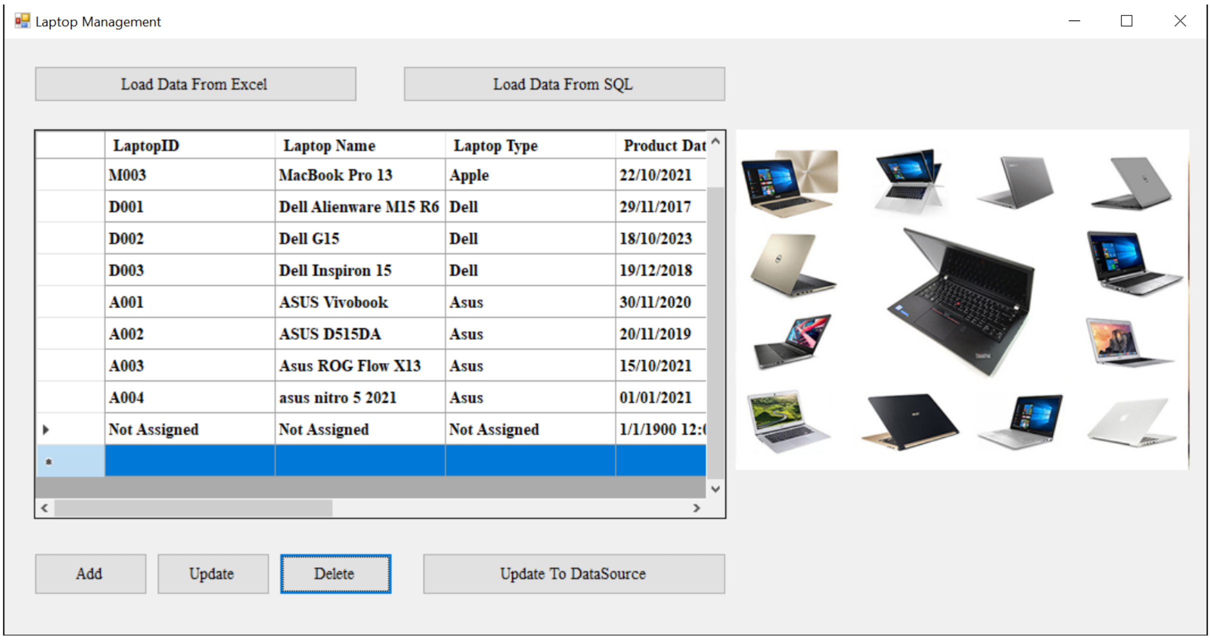
	LaptopID	Laptop Name	Laptop Type	Product Date
*				



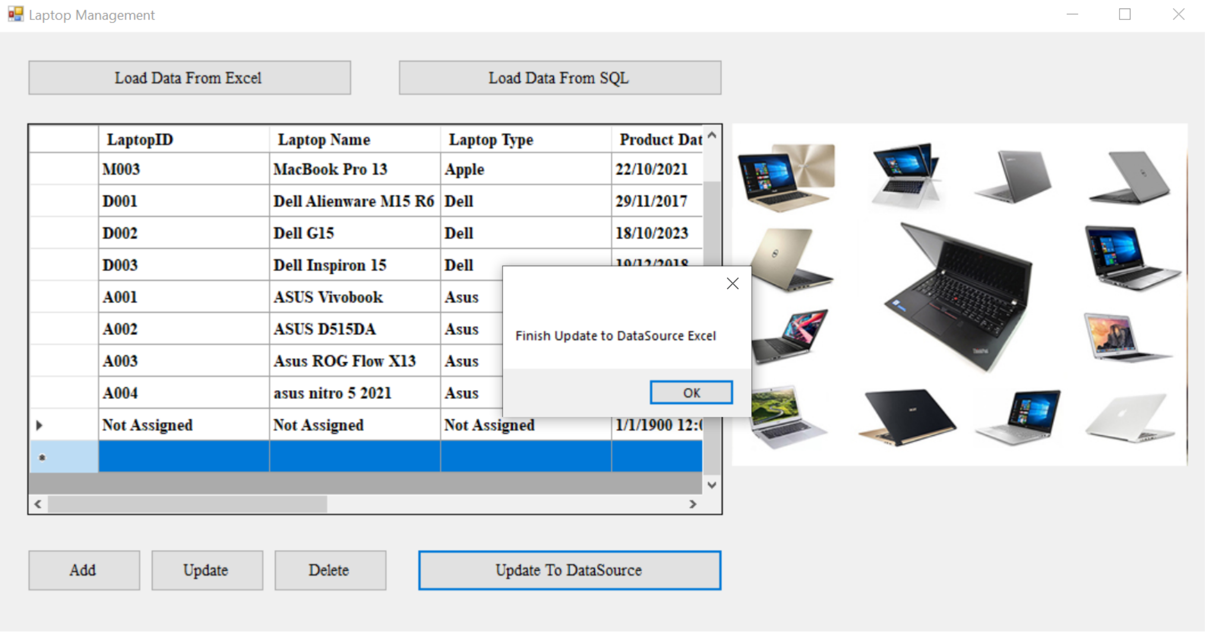
	A	B	C	D	E	F	G	H	I
1	LaptopID	LaptopName	LaptopType	ProductDate	Processor	HDD	RAM	Price	ImageName
2	M001	MacBook Air 13	Apple	20/10/2019	Intel Core i5	512GB	8GB	26690000	MacBook-Air-13.jpg
3	M002	MacBook Air M1	Apple	25/12/2020	Apple M1	512GB	16GB	35000000	MacBook-Air-M1.jpg
4	M003	MacBook Pro 13	Apple	22/10/2021	Intel Core i5	256GB	8GB	33500000	MacBook-Pro-13.jpg
5	D001	Dell Alienware M15 R6	Dell	29/11/2017	Intel Core i7	1TB	32GB	57990000	Dell-Alienware-M15-Alienware-R6.jpg
6	D002	Dell G15	Dell	18/10/2023	Intel Core i5	256GB	8GB	27000000	Dell-G15.jpg
7	D003	Dell Inspiron 15	Dell	19/12/2018	Intel Core i7	512GB	8GB	23090000	Dell-Inspiron-15.jpg
8	A001	ASUS Vivobook	Asus	30/11/2020	Intel Core i5	512GB	8GB	18390000	Asus-Vivobook.jpg
9	A002	ASUS D515DA	Asus	20/11/2019	AMD Ryzen 3	512GB	4GB	12990000	Asus-D515DA.jpg
10	A003	Asus ROG Flow X13	Asus	15/10/2021	AMD Ryzen 9	1TB	32GB	79990000	Asus-ROG-Flow-ROG-X13.jpg
11									
12									



	LaptopID	Laptop Name	Laptop Type	
▶	M001	MacBook Air 13	Apple	20/10/2019
	M002	MacBook Air M1	Apple	25/12/2020
	M003	MacBook Pro 13	Apple	22/10/2021
	D001	Dell Alienware M15 R6	Dell	29/11/2017
	D002	Dell G15	Dell	18/10/2023
	D003	Dell Inspiron 15	Dell	19/12/2018
	A001	ASUS Vivobook	Asus	30/11/2020
	A002	ASUS D515DA	Asus	20/11/2019
	A003	Asus ROG Flow X13	Asus	15/10/2021
*				

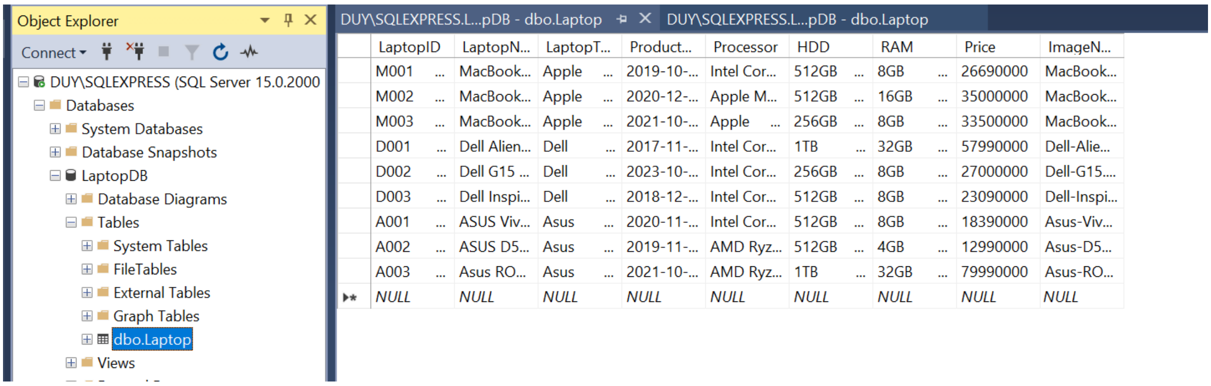


LaptopID	Laptop Name	Laptop Type	Product Dat
M003	MacBook Pro 13	Apple	22/10/2021
D001	Dell Alienware M15 R6	Dell	29/11/2017
D002	Dell G15	Dell	18/10/2023
D003	Dell Inspiron 15	Dell	19/12/2018
A001	ASUS Vivobook	Asus	30/11/2020
A002	ASUS D515DA	Asus	20/11/2019
A003	Asus ROG Flow X13	Asus	15/10/2021
A004	asus nitro 5 2021	Asus	01/01/2021
Not Assigned	Not Assigned	Not Assigned	1/1/1900 12:(

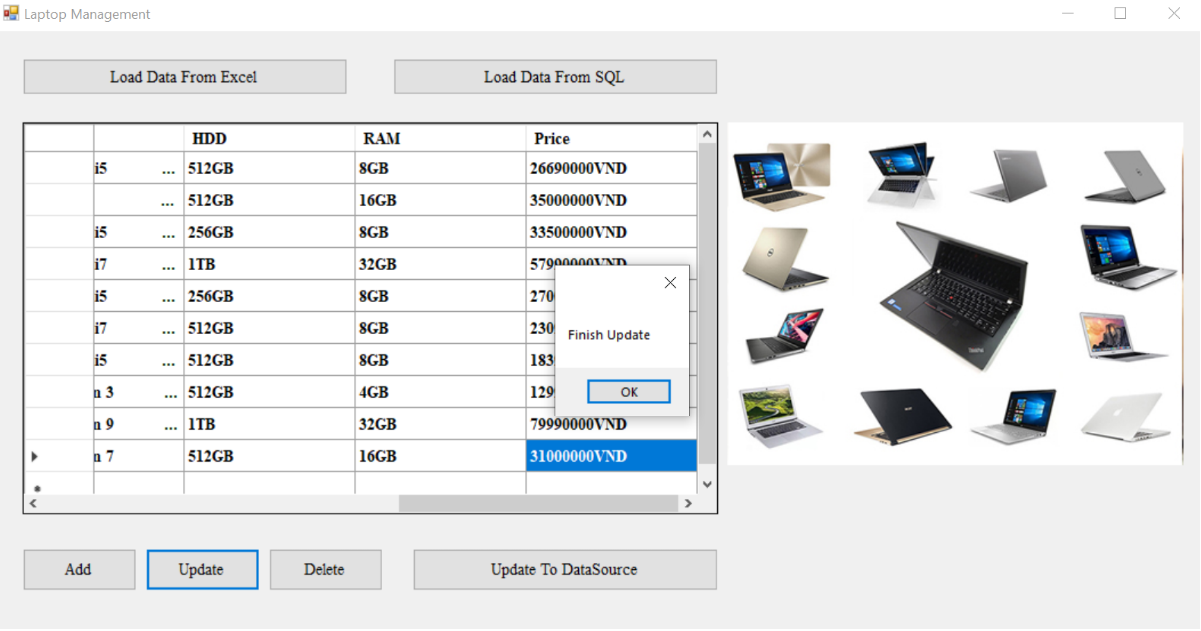


LaptopID	Laptop Name	Laptop Type	Product Dat
M003	MacBook Pro 13	Apple	22/10/2021
D001	Dell Alienware M15 R6	Dell	29/11/2017
D002	Dell G15	Dell	18/10/2023
D003	Dell Inspiron 15	Dell	19/12/2018
A001	ASUS Vivobook	Asus	
A002	ASUS D515DA	Asus	
A003	Asus ROG Flow X13	Asus	
A004	asus nitro 5 2021	Asus	
Not Assigned	Not Assigned	Not Assigned	1/1/1900 12:(

Finish Update to DataSource Excel



	LaptopID	LaptopN...	LaptopT...	Product...	Processor	HDD	RAM	Price	ImageN...
	M001 ...	MacBook...	Apple ...	2019-10-...	Intel Cor...	512GB ...	8GB ...	26690000	MacBook...
	M002 ...	MacBook...	Apple ...	2020-12-...	Apple M...	512GB ...	16GB ...	35000000	MacBook...
	M003 ...	MacBook...	Apple ...	2021-10-...	Apple ...	256GB ...	8GB ...	33500000	MacBook...
	D001 ...	Dell Alien...	Dell ...	2017-11-...	Intel Cor...	1TB ...	32GB ...	57990000	Dell-Alie...
	D002 ...	Dell G15 ...	Dell ...	2023-10-...	Intel Cor...	256GB ...	8GB ...	27000000	Dell-G15....
	D003 ...	Dell Inspi...	Dell ...	2018-12-...	Intel Cor...	512GB ...	8GB ...	23090000	Dell-Inspi...
	A001 ...	ASUS Viv...	Asus ...	2020-11-...	Intel Cor...	512GB ...	8GB ...	18390000	Asus-Viv...
	A002 ...	ASUS D5...	Asus ...	2019-11-...	AMD Ryz...	512GB ...	4GB ...	12990000	Asus-D5...
	A003 ...	Asus RO...	Asus ...	2021-10-...	AMD Ryz...	1TB ...	32GB ...	79990000	Asus-RO...
▸*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



		HDD	RAM	Price
	i5 ...	512GB	8GB	26690000VND
	...	512GB	16GB	35000000VND
	i5 ...	256GB	8GB	33500000VND
	i7 ...	1TB	32GB	57900000VND
	i5 ...	256GB	8GB	270
	i7 ...	512GB	8GB	230
	i5 ...	512GB	8GB	183
	n 3 ...	512GB	4GB	129
	n 9 ...	1TB	32GB	79990000VND
▸	n 7	512GB	16GB	31000000VND
*				

Finish Update