

Laptop Management

Laptop.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows.Forms;
7 using System.IO;
8
9 namespace Assignment_05
10 {
11     public class Laptop
12     {
13         public string LaptopID { get; set; }
14         public string LaptopName { get; set; }
15         public string LaptopType { get; set; }
16         public DateTime ProductDate { get; set; }
17         public string Processor { get; set; }
18         public string HDD { get; set; }
19         public string RAM { get; set; }
20         public int Price { get; set; }
21         public string Avatar { get; set; }
22
23         public Laptop()
24         {
25             LaptopID = "Not Assigned";
26             LaptopName = "Not Assigned";
27         }
28     }
29 }
30
```

Program.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace Assignment_05
8 {
9     internal static class Program
10     {
11         /// <summary>
12         /// The main entry point for the application.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new LaptopManagementForm());
20         }
21     }
22 }
23
```

Design

Load Data From Excel Load Data From SQL

	LaptopID	Laptop Name	Laptop Type	Product Date
*				

Add Update Delete Update To DataSource

LaptopManagement.cs

```

4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.IO;
11 using Excel = Microsoft.Office.Interop.Excel;
12 using System.Globalization;
13 using System.Data.SqlClient;
14
15 namespace Assignment_85
16 {
17     4 references
18     public partial class LaptopManagementForm : Form
19     {
20         public List<Laptop> Llist = new List<Laptop>();
21
22         public int loadData = 0;
23         static string ProjectPath = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.FullName;
24         string ExcelFilePath = ProjectPath + "\\Data\\LaptopList.xlsx";
25         string connectionString = "Data Source=DESKTOP-MJ27046\\SQLEXPRESS;Initial Catalog = LaptopDB; Integrated Security= SSPI";
26         int CurrentLaptopIndex = -1;
27         DataTable datatable;
28         BindingSource binding = new BindingSource();
29
30     1 reference
31     public LaptopManagementForm()
32     {
33         InitializeComponent();
34     }

```

```

1 reference
35 private void btnLoadExcel_Click(object sender, EventArgs e)
36 {
37     loadData = 1;
38     datatable = new DataTable();
39     lList.Clear();
40
41     int colCount = 9;
42     int NumDataRow = ReadDataFromFile(lList, ExcelFilePath, colCount);
43
44     var sublist = lList.Select(x => new
45     {
46         LaptopID = x.LaptopID,
47         LaptopName = x.LaptopName,
48         LaptopType = x.LaptopType,
49         ProductDate = x.ProductDate.ToString("dd/MM/yyyy"),
50         Processor = x.Processor,
51         HDD = x.HDD,
52         RAM = x.RAM,
53         Price = x.Price.ToString() + " VND"
54     }).ToList();
55
56     datatable.Columns.Add("LaptopID");
57     datatable.Columns.Add("LaptopName");
58     datatable.Columns.Add("LaptopType");
59     datatable.Columns.Add("ProductDate");
60     datatable.Columns.Add("Processor");
61     datatable.Columns.Add("HDD");
62     datatable.Columns.Add("RAM");
63     datatable.Columns.Add("Price");
64
65     DataRow newrow;
66     foreach (var h in sublist)
67     {
68         newrow = datatable.NewRow();
69         newrow["LaptopID"] = h.LaptopID;
70         newrow["LaptopName"] = h.LaptopName;
71         newrow["LaptopType"] = h.LaptopType;
72         newrow["ProductDate"] = h.ProductDate;
73         newrow["Processor"] = h.Processor;
74         newrow["HDD"] = h.HDD;
75         newrow["RAM"] = h.RAM;
76         newrow["Price"] = h.Price;
77         datatable.Rows.Add(newrow);
78         datatable.AcceptChanges();
79     }
80
81     binding.AllowNew = true;
82     binding.DataSource = datatable;
83     dgvLaptopList.AutoGenerateColumns = false;
84     dgvLaptopList.DataSource = binding;
85 }
86

```

```

1 reference
87 public int ReadDataFromFile(List<Laptop> DataList, string FilePath, int colCount)
88 {
89     Excel.Application xlApp = new Excel.Application();
90     Excel.Workbook xlWorkbook = xlApp.Workbooks.Open(FilePath);
91     Excel.Worksheet xlWorksheet = xlWorkbook.Sheets[1];
92     Excel.Range xlRange = xlWorksheet.UsedRange;
93
94     xlWorksheet.Columns.ClearFormats();
95     xlWorksheet.Rows.ClearFormats();
96
97     int rowCount = xlWorksheet.UsedRange.Rows.Count;
98
99     int numLaptop = 0;
100     string LaptopID = "";
101     string LaptopName = "";
102     string LaptopType = "";
103     DateTime ProductDate = DateTime.Now;
104     string Processor = "";
105     string HDD = "";
106     string RAM = "";
107     int Price = 0;
108     string Avatar = "";
109

```

```

109     for (int i = 2; i <= rowCount; i++)
110     {
111         for (int j = 1; j <= colCount; j++)
112         {
113             switch (j)
114             {
115                 case 1:
116                     LaptopID = xlRange.Cells[i, j].Value2.ToString();
117                     break;
118                 case 2:
119                     LaptopName = xlRange.Cells[i, j].Value2.ToString();
120                     break;
121                 case 3:
122                     LaptopType = xlRange.Cells[i, j].Value2.ToString();
123                     break;
124                 case 4:
125                     ProductDate = DateTime.ParseExact(xlRange.Cells[i, j].Value2.ToString(),
126                                                         "dd/MM/yyyy", CultureInfo.InvariantCulture);
127                     break;
128                 case 5:
129                     Processor = xlRange.Cells[i, j].Value2.ToString();
130                     break;
131                 case 6:
132                     HDD = xlRange.Cells[i, j].Value2.ToString();
133                     break;
134                 case 7:
135                     RAM = xlRange.Cells[i, j].Value2.ToString();
136                     break;
137                 case 8:
138                     Price = Convert.ToInt32(xlRange.Cells[i, j].Value2.ToString());
139                     break;
140                 case 9:
141                     Avatar = xlRange.Cells[i, j].Value2.ToString();
142                     break;
143             }
144         }
145         Datalist.Add(new Laptop());
146         Datalist[numLaptop].LaptopID = LaptopID;
147         Datalist[numLaptop].LaptopName = LaptopName;
148         Datalist[numLaptop].LaptopType = LaptopType;
149         Datalist[numLaptop].ProductDate = ProductDate;
150         Datalist[numLaptop].Processor = Processor;
151         Datalist[numLaptop].HDD = HDD;
152         Datalist[numLaptop].RAM = RAM;
153         Datalist[numLaptop].Price = Price;
154         Datalist[numLaptop].Avatar = Avatar;
155         numLaptop = numLaptop + 1;
156     }
157 }
158
159 xlApp.Quit();
160
161 MessageBox.Show("Load Data From Excel Done! : " + (rowCount - 1).ToString() + " Records");
162
163 return (rowCount - 1);
164 }
165

```

```

180 private void btnLoadSQL_Click(object sender, EventArgs e)
181 {
182     loadData = 2;
183     datatable = new DataTable();
184     llist.Clear();
185
186     int NumDataRow = ReadDataFromSQLServer(llist, connotionString);
187
188     var sublist = llist.Select(x => new
189     {
190         LaptopID = x.LaptopID,
191         LaptopName = x.LaptopName,
192         LaptopType = x.LaptopType,
193         ProductDate = x.ProductDate.ToString("dd/MM/yyyy"),
194         Processor = x.Processor,
195         HDD = x.HDD,
196         RAM = x.RAM,
197         Price = x.Price.ToString() + " VND"
198     }).ToList();
199
200     datatable.Columns.Add("LaptopID");
201     datatable.Columns.Add("LaptopName");
202     datatable.Columns.Add("LaptopType");
203     datatable.Columns.Add("ProductDate");
204     datatable.Columns.Add("Processor");
205     datatable.Columns.Add("HDD");
206     datatable.Columns.Add("RAM");
207     datatable.Columns.Add("Price");
208 }

```

```

209 DataRow newRow;
210 foreach (var h in sublist)
211 {
212     newRow = datatable.NewRow();
213     newRow["LaptopID"] = h.LaptopID;
214     newRow["LaptopName"] = h.LaptopName;
215     newRow["LaptopType"] = h.LaptopType;
216     newRow["ProductDate"] = h.ProductDate;
217     newRow["Processor"] = h.Processor;
218     newRow["HDD"] = h.HDD;
219     newRow["RAM"] = h.RAM;
220     newRow["Price"] = h.Price;
221     datatable.Rows.Add(newRow);
222     datatable.AcceptChanges();
223 }
224
225 binding.AllowNew = true;
226 binding.DataSource = datatable;
227 dgvLaptopList.AutoGenerateColumns = false;
228 dgvLaptopList.DataSource = binding;
229 }
230
231 1 reference
232 public int ReadDataFromSqlServer(List<Laptop> DataList, string connectionString)
233 {
234     SqlConnection cnn;
235     cnn = new SqlConnection(connectionString);
236     int iRow = 0;
237     int NumRecords = 0;
238
239     try
240     {
241         cnn.Open();
242         Console.WriteLine("Connection Open !");
243
244         string SqlString = @"SELECT
245                               LaptopID,
246                               LaptopName,
247                               LaptopType,
248                               ProductDate = Convert(varchar(10), CONVERT(date, ProductDate, 106), 103),
249                               Processor,
250                               HDD,
251                               RAM,
252                               Price,
253                               ImageName
254                               FROM Laptop";
255
256         using (var command = new SqlCommand(SqlString, cnn))
257         {
258             using (var reader = command.ExecuteReader())
259             {
260                 while (reader.Read())
261                 {
262                     DataList.Add(new Laptop());
263                     DataList[iRow].LaptopID = reader.GetString(0);
264                     DataList[iRow].LaptopName = reader.GetString(1);
265                     DataList[iRow].LaptopType = reader.GetString(2);
266                     DataList[iRow].ProductDate = DateTime.ParseExact(reader.GetString(3), "dd/MM/yyyy", CultureInfo.InvariantCulture);
267                     DataList[iRow].Processor = reader.GetString(4);
268                     DataList[iRow].HDD = reader.GetString(5);
269                     DataList[iRow].RAM = reader.GetString(6);
270                     DataList[iRow].Price = reader.GetInt32(7);
271                     DataList[iRow].Avatar = reader.GetString(8);
272
273                     iRow++;
274                 }
275             }
276
277             SqlCommand cmd = new SqlCommand("select count (*) from Laptop", cnn);
278             object result = cmd.ExecuteScalar();
279             NumRecords = int.Parse(result.ToString());
280
281             MessageBox.Show("Finish Load Data Frome SQL: " + NumRecords.ToString() + " Records");
282             cnn.Close();
283         }
284
285         catch (SqlException ex)
286         {
287             MessageBox.Show("Can not open connection ! : " + ex.Message);
288         }
289
290         return NumRecords;
291     }

```

```

292 private void dgvLapList_EditingControlShowing(object sender, DataGridViewEditingControlShowingEventArgs e)
293 {
294     e.Control.KeyPress += new KeyPressEventHandler(ColumnPrice_KeyPress);
295     if (dgvLapList.CurrentCell.ColumnIndex == 7)
296     {
297         TextBox tb = e.Control as TextBox;
298         if (tb != null)
299         {
300             tb.KeyPress += new KeyPressEventHandler(ColumnPrice_KeyPress);
301         }
302     }
303 }
304
305 2 references
306 private void ColumnPrice_KeyPress(object sender, KeyPressEventArgs e)
307 {
308     if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
309     {
310         e.Handled = true;
311     }
312 }
313
314 1 reference
315 private void btnAdd_Click(object sender, EventArgs e)
316 {
317     Laptop l = new Laptop();
318     l.LaptopID = "Not Assigned";
319     l.LaptopName = "Not Assigned";
320     l.LaptopType = "Not Assigned";
321     l.ProductDate = DateTime.ParseExact("81/81/1988", "dd/MM/yyyy", CultureInfo.InvariantCulture);
322     l.Processor = "Not Assigned";
323     l.HDD = "Not Assigned";
324     l.RAM = "Not Assigned";
325     l.Price = 8;
326     l.Avatar = "Laptop.jpg";
327     lList.Add(l);
328
329     DataRow newrow;
330     newrow = datatable.NewRow();
331     newrow["LaptopID"] = l.LaptopID;
332     newrow["LaptopName"] = l.LaptopName;
333     newrow["LaptopType"] = l.LaptopType;
334     newrow["ProductDate"] = l.ProductDate;
335     newrow["Processor"] = l.Processor;
336     newrow["HDD"] = l.HDD;
337     newrow["RAM"] = l.RAM;
338     newrow["Price"] = l.Price.ToString() + " VND";
339     datatable.Rows.Add(newrow);
340     datatable.AcceptChanges();
341
342     MessageBox.Show("Finish Adding");
343 }
344
345 1 reference
346 private void btnDelete_Click(object sender, EventArgs e)
347 {
348     Laptop l;
349     if (CurrentLaptopIndex >= 0)
350     {
351         l = lList[CurrentLaptopIndex];
352     }
353     else
354     {
355         return;
356     }
357
358     string question = "Do You Want to delete Laptop: " + l.LaptopID;
359     DialogResult result = MessageBox.Show(question, "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
360     if (result == DialogResult.Yes)
361     {
362         lList.RemoveAt(CurrentLaptopIndex);
363         binding.RemoveAt(CurrentLaptopIndex);
364     }
365     MessageBox.Show("Finish Delete");
366 }

```

```

367
368 1 reference
369 private void btnUpdate_Click(object sender, EventArgs e)
370 {
371     DataRow row;
372     for (int i = 0; i < datatable.Rows.Count; i++)
373     {
374         row = datatable.Rows[i];
375
376         lList[i].LaptopID = row["LaptopID"].ToString();
377         lList[i].LaptopName = row["LaptopName"].ToString();
378         lList[i].LaptopType = row["LaptopType"].ToString();
379         lList[i].ProductDate = DateTime.ParseExact(row["ProductDate"].ToString(), "dd/MM/yyyy", CultureInfo.InvariantCulture);
380         lList[i].Processor = row["Processor"].ToString();
381         lList[i].HDD = row["HDD"].ToString();
382         lList[i].RAM = row["RAM"].ToString();
383         string sPrice = row["Price"].ToString();
384         lList[i].Price = Convert.ToInt32(sPrice.Substring(0, sPrice.IndexOf(" VND")));
385     }
386     MessageBox.Show("Finish Update");
387 }
388
389 1 reference
390 private void btnUpdateSource_Click(object sender, EventArgs e)
391 {
392     if (LoadData == 1)
393     {
394         WriteDataToExcelFile(lList, ExcelFilePath);
395     }
396     else
397     {
398         WriteDataToSQLServer(lList, connectionString);
399     }
400 }
401
402 1 reference

```

```

390 public void WriteDataToExcelFile(List<Laptop> lList, string ExcelFilePath)
391 {
392     Excel.Application xlApp = new Excel.Application();
393     Excel.Workbook xlWorkbook = xlApp.Workbooks.Open(ExcelFilePath);
394     Excel.Worksheet xlWorksheet = xlWorkbook.Sheets[1];
395
396     Excel.Range xlRange;
397     string[,] Data = new string[1, 10];
398
399     int idxRow = 2;
400     foreach (Laptop l in lList)
401     {
402         Data[0, 0] = l.LaptopID;
403         Data[0, 1] = l.LaptopName;
404         Data[0, 2] = l.LaptopType;
405         Data[0, 3] = l.ProductDate.ToString("dd/MM/yyyy", CultureInfo.InvariantCulture);
406         Data[0, 4] = l.Processor;
407         Data[0, 5] = l.HDD;
408         Data[0, 6] = l.RAM;
409         Data[0, 7] = l.Price.ToString();
410         Data[0, 8] = l.Avatar;
411
412         xlRange = xlWorksheet.get_Range("A" + idxRow.ToString(), "J" + idxRow.ToString());
413         xlRange.Value2 = Data;
414
415         idxRow = idxRow + 1;
416     }
417
418     xlWorkbook.Save();
419     xlWorkbook.Close();
420     xlApp.Quit();
421
422     MessageBox.Show("Finish Update to DataSource Excel");
423 }
424
425 //reference
426 public void WriteDataToSQLServer(List<Laptop> lList, string connectionString)
427 {
428     SqlConnection cnn;
429     SqlCommand myCommand = new SqlCommand();
430     string query;
431
432     cnn = new SqlConnection(connectionString);
433     try
434     {
435         cnn.Open();
436         Console.WriteLine("Connection Open !");
437
438         query = "TRUNCATE TABLE Laptop";
439         myCommand.CommandText = query;
440         myCommand.Connection = cnn;
441         myCommand.ExecuteNonQuery();
442
443         query = @"INSERT INTO Laptop(LaptopID,LaptopName,LaptopType,
444                                     ProductDate,Processor,HDD, RAM,Price,ImageName)";
445         query += @"VALUES (@LaptopID,@LaptopName,@LaptopType,
446                             @ProductDate,@Processor,@HDD,@RAM,@Price,@ImageName)";
447         myCommand.CommandText = query;
448         myCommand.Connection = cnn;
449
450         myCommand.Parameters.Add(new SqlParameter("@LaptopID", SqlDbType.NVarChar));
451         myCommand.Parameters.Add(new SqlParameter("@LaptopName", SqlDbType.NVarChar));
452         myCommand.Parameters.Add(new SqlParameter("@LaptopType", SqlDbType.NVarChar));
453         myCommand.Parameters.Add(new SqlParameter("@ProductDate", SqlDbType.DateTime));
454         myCommand.Parameters.Add(new SqlParameter("@Processor", SqlDbType.NVarChar));
455         myCommand.Parameters.Add(new SqlParameter("@HDD", SqlDbType.NVarChar));
456         myCommand.Parameters.Add(new SqlParameter("@RAM", SqlDbType.NVarChar));
457         myCommand.Parameters.Add(new SqlParameter("@Price", SqlDbType.Int));
458         myCommand.Parameters.Add(new SqlParameter("@ImageName", SqlDbType.NVarChar));
459
460         foreach (Laptop l in lList)
461         {
462             myCommand.Parameters[0].Value = l.LaptopID;
463             myCommand.Parameters[1].Value = l.LaptopName;
464             myCommand.Parameters[2].Value = l.LaptopType;
465             myCommand.Parameters[3].Value = l.ProductDate.ToString("dd/MM/yyyy", CultureInfo.InvariantCulture);
466             myCommand.Parameters[4].Value = l.Processor;
467             myCommand.Parameters[5].Value = l.HDD;
468             myCommand.Parameters[6].Value = l.RAM;
469             myCommand.Parameters[7].Value = l.Price.ToString();
470             myCommand.Parameters[8].Value = l.Avatar;
471
472             myCommand.ExecuteNonQuery();
473         }
474         cnn.Close();
475     }
476     catch (SqlException ex)
477     {
478     }
479
480     catch (SqlException ex)
481     {
482         MessageBox.Show("Can not open connection ! " + ex.Message);
483     }
484     MessageBox.Show("Finish Update to DataSource SQL Server");
485 }

```

[illegible]

Load Data From Excel

Load Data From SQL

	LaptopID	Laptop Name	Laptop Type	Date
▶	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
✱				

<>

Add

Update

Delete

Update To DataSource

Laptop Management

Load Data From Excel
Load Data From SQL

	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 12:0

Add
Update
Delete
Update To DataSource

Laptop Management

Load Data From Excel
Load Data From SQL

	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 12:0

Add
Update
Delete
Update To DataSource

Finish Update to DataSource Excel
OK

Object Explorer

Connect

DUY\SQLEXPRESS (SQL Server 15.0.2000)

- Databases
 - System Databases
 - Database Snapshots
 - LaptopDB
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Laptop
 - Views

DUY\SQLEXPRESS.L...pDB - dbo.Laptop

	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

Laptop Management

Load Data From Excel Load Data From SQL

		HDD	RAM	Price
	i5	512GB	8GB	26690000VND
		512GB	16GB	35000000VND
	i5	256GB	8GB	33500000VND
	i7	1TB	32GB	57990000VND
	i5	256GB	8GB	27000000VND
	i7	512GB	8GB	23090000VND
	i5	512GB	8GB	18390000VND
	a 3	512GB	4GB	12990000VND
	a 9	1TB	32GB	79990000VND
▶	a 7	512GB	16GB	31000000VND
*				

Finish Update

OK

Add Update Delete Update To DataSource