

Print Name _____

Signature _____ (pledged)

Purchase Order Form In-Class Lab II-B (cont)

In-Class Individual Lab

- 1] We have now discussed a number of ways that we might try to implement the Item Detail tab. We will continue to stress the need for no training and no manuals.
- 2] Suggestions included (1) placing a number of data entry sets on one tab and (b) using a datagrid for the collections of data.
- 3] In the space below, I am going to use the first of these two approaches because it is easier to code.

Part V (cont)

Vendor	Vendor	Company	Company	DATE	DATE	PO #	PO #
PUSH	Search For Item	Item ID	Description	Quantity	Unit Price \$	Total \$	Delete This Item

- 1] Add a gray label near the top of the Item Detail tab. Include the data above in the label.

Vendor	Vendor	Company	Company	DATE	DATE	PO #	PO #
PUSH	Search For Item	Item ID	Description	Quantity \$	Unit Price \$	Total \$	Delete This Item

- 2] Place a large rectangular panel below the gray label.

Vendor	Vendor	Company	Company	DATE	DATE	PO #	PO #
PUSH	Search For Item	Item ID	Description	Quantity	Unit Price \$	Total \$	Delete This Item
Add Row 01							

- 3] Create `btnAddRow01`.

PUSH	Search For Item	Item ID	Description	Quantity	Unit Price \$	Total \$	Delete This Item
Add Row 01	Search	<input type="text" value="**"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete

- 4] Create a thin rectangular panel, called **pnI01**, to the right of **btnAddRow01**. Inside this panel, add **btnSearch01 txtItemID01, txtDescription01, txtQuantity01, txtUnitPrice01, txtTotal01, & btnDelete01**.
- 5] Place 4 pixels between each of the items in pnI01. Adjust the gray label to align the items as shown above.
- 6] Copy/paste → **btnAddRow01.pnI01, btnSearch01 txtItemID01, txtDescription01, txtQuantity01, txtUnitPrice01, txtTotal01, & btnDelete01**.

PUSH	Search For Item	Item ID	Description	Quantity	Unit Price \$	Total \$	Delete This Item
Add Row 01	Search	<input type="text" value="**"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 02	* Search *	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete

- 7] Place this second row of data just below the first. Name the items → **btnAddRow02, pnI02, btnSearch02 txtItemID02, txtDescription02, txtQuantity02, txtUnitPrice02, txtTotal02, & btnDelete02**.

Vendor	Vendor	Company	Company	DATE	DATE	PO #	PO #
PUSH	Search For Item	Item ID	Description	Quantity	Unit Price \$	Total \$	Delete This Item
Add Row 01	Search	<input type="text" value="**"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 02	* Search *	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 03	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 04	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 05	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 06	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 07	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 08	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 09	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 10	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 11	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 12	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 13	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 14	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 15	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 16	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 17	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 18	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 19	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Add Row 20	Search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Delete
Done							

- 8] Add additional rows until your form is filled; I can hold 20. Add **btnDone** at the bottom.
- 9] Name each of the Red buttons → **btnAddRow01, btnAddRow02, ..., btnAddRow20**.

- 10] Name each of the Red buttons → **pnl01, pnl02, ..., pnl20.**

```
public PurchaseOrder()
{
    InitializeComponent();

    btnAddRow02.Hide();
    btnAddRow03.Hide();
    btnAddRow04.Hide();
    btnAddRow05.Hide();
    btnAddRow06.Hide();
    btnAddRow07.Hide();
    btnAddRow08.Hide();
    btnAddRow09.Hide();
    btnAddRow10.Hide();
    btnAddRow11.Hide();
    btnAddRow12.Hide();
    btnAddRow13.Hide();
    btnAddRow14.Hide();
    btnAddRow15.Hide();
    btnAddRow16.Hide();
    btnAddRow17.Hide();
    btnAddRow18.Hide();
    btnAddRow19.Hide();
    btnAddRow20.Hide();
    btnDone.Hide();

    pnl01.Hide();
    pnl02.Hide();
    pnl03.Hide();
    pnl04.Hide();
    pnl05.Hide();
    pnl06.Hide();
    pnl07.Hide();
    pnl08.Hide();
    pnl09.Hide();
    pnl10.Hide();
    pnl11.Hide();
    pnl12.Hide();
    pnl13.Hide();
    pnl14.Hide();
    pnl15.Hide();
    pnl16.Hide();
    pnl17.Hide();
    pnl18.Hide();
    pnl19.Hide();
    pnl20.Hide();
}
```

- 11] Add the code above to your Purchase Order constructor. **THIS IS JUST THE FIRST DRAFT OF THE TESTING - WE WILL REPLACE IT LATER WITH THE FINAL TESTING CODE WHEN ALL IS COMPLETE!**

```
pnl01.Show();
btnAddRow01.Text = "1]";
txtItemID01.Focus();
```

- 12] Create an On-Click event for **btnAddRow01**. Place the code above in the event.

```

txtDescription01.Text = "Mikasa Leather BasketBall";
txtUnitPrice01.Text = "60.00";
txtQuantity01.Text = "1";
txtTotal01.Text = "60.00";
btnAddRow02.Show();
btnDone.Show();
txtUnitPrice01.Focus();

```

- 13] Create a Leave Event for **txtItemID01**. Place the code above in the event.

```

pn102.Show();
btnAddRow02.Text = "2]";
txtItemID02.Focus();

```

- 14] Create an On-Click Event for **btnAddRow02**. Place the code above in the event.

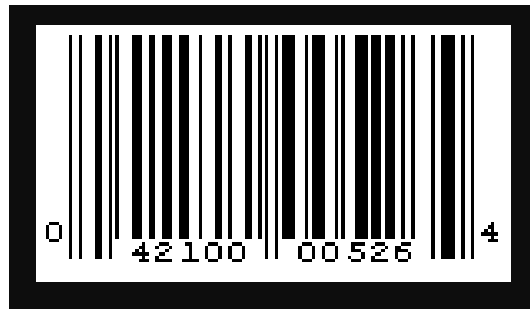
Building Sample Record Part I

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	
Name	VarChar(50)	
Description	VarChar(300)	
VendorID	INT	
VendorItemID	VarChar(12)	
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 1] Suppose we have 101 Inventory items. The Auto-Increment ID, provided by the database, is **102**.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	
VendorID	INT	
VendorItemID	VarChar(12)	
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 2] Suppose we would like the name for this item, in our system, to be **"Mikasa Leather BasketBall"**.



- 3] The UPC-A (Universal Product Code) is 12 digits long. Suppose the barcode, produced by our automatic bar code program, is the one.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	
VendorID	INT	
VendorItemID	VarChar(12)	
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

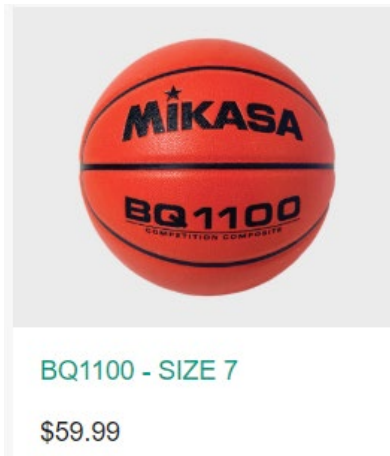
- 4] Suppose we would like the barcode for this item, in our system, to be **"042100005264"**.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	
VendorItemID	VarChar(12)	
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 5] The Description would provide a more detailed explanation of the inventory item. Since this is a demo item, we will keep it short. The Description is **"This is an NBA quality leather indoor basketball"**.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 6] In the Vendor Select ComboBox, used in the first tab, we identified Mikasa as Vendor **2**.



- 7] It is important that we make sure that the Vendor knows exactly what we are trying to order. The ItemID, used by Mikasa, is BQ1100; this will not always be marked on the object, but it is for this basketball. Thus → Our Order should include **BQ1100**.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 8] Our **VendorItemID** is **BQ1100** → Our Purchase Order Must Include This Vendor Item ID!

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 9] When we look at the Mikasa catalog, we see that they describe this item a "Professional Leather Basketball". Our **VendorItemDescription** is "**Professional Leather Basketball**" → Our Purchase Order Must Include This Vendor Description!

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	40.00
UnitPrice	Double	
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 10] Suppose I paid \$40.00 for this basketball.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	40.00
UnitPrice	Double	40.00
SellingPrice	Double	
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 11] Suppose I paid \$40.00 for this basketball → the last time I ordered it. My cost may, or may not change.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	40.00
UnitPrice	Double	40.00
SellingPrice	Double	59.95
QuantityInStock	INT	
MinimumQuantityToStock	INT	
Deleted	Bool/Char	

- 12] As you can see, from the Ad, the **SellingPrice** is **59.95**

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	40.00
UnitPrice	Double	40.00
SellingPrice	Double	59.95
QuantityInStock	INT	3
MinimumQuantityToStock	INT	7
Deleted	Bool/Char	

- 13] Suppose we wish to have 7 in stock at all times. Suppose we currently have 3 in stock. These values would enable us to do automatically generate an order, to Mikasa, → this would produce a single purchase order → WHICH WE COULD MODIFY → that would include all of the Mikasa items that are needed. This Auto-Order would include 4 more of these Leather Basketballs.

Inventory		Sample Part I
ID	AUTO INT	102
BarCode	CHAR(12)	042100005264
Name	VarChar(50)	Mikasa Leather BasketBall
Description	VarChar(300)	This is an NBA quality leather indoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BQ1100
VendorDescription	VarChar(50)	Professional Leather Basketball
LastCost	Double	40.00
UnitPrice	Double	40.00
SellingPrice	Double	59.95
QuantityInStock	INT	3
MinimumQuantityToStock	INT	7
Deleted	Bool/Char	F

- 14] This item is still available from Mikasa. When Mikasa ceases to have this item, we will tag it for deletion.

- 15] The record above represents Part I → ID = 102.
 16] In a real application, there would be many more fields in the Inventory records.

Sample Record Part II



BWLC110 - SIZE 6

\$32.99

Inventory		Sample Part II
ID	AUTO INT	115
BarCode	CHAR(12)	042103335264
Name	VarChar(50)	Mikasa Synthetic Street BasketBall
Description	VarChar(300)	This is a high quality synthetic outdoor basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BWL110
VendorDescription	VarChar(50)	Synthetic Street Basketball
LastCost	Double	20.00
UnitPrice	Double	20.00
SellingPrice	Double	32.99
QuantityInStock	INT	6
MinimumQuantityToStock	INT	20
Deleted	Bool/Char	F

- 1] The record above represents Part II → ID = 115.

Building Sample Record Part III



VIP300-R

\$59.99

Inventory		Sample Part III
ID	AUTO INT	87
BarCode	CHAR(12)	042888335264
Name	VarChar(50)	Mikasa VolleyBall
Description	VarChar(300)	This is a high quality indoor volleyball
VendorID	INT	2
VendorItemID	VarChar(12)	VIP300-R
VendorDescription	VarChar(50)	NFHS Competition game ball series
LastCost	Double	35.00
UnitPrice	Double	35.00
SellingPrice	Double	59.99
QuantityInStock	INT	5
MinimumQuantityToStock	INT	5
Deleted	Bool/Char	F

1] The record above represents Part III → ID = 87.

Building Sample Record Part IV



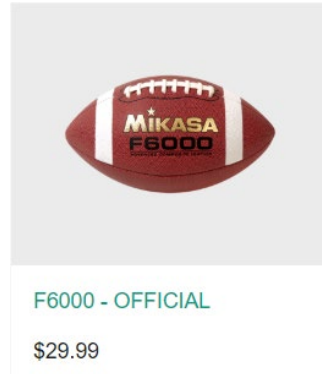
BX1010 - SIZE 6

\$14.99

Inventory		Sample Part IV
ID	AUTO INT	112
BarCode	CHAR(12)	042888335777
Name	VarChar(50)	Mikasa Outdoor Basketball
Description	VarChar(300)	This is an introductory youth basketball
VendorID	INT	2
VendorItemID	VarChar(12)	BX1010
VendorDescription	VarChar(50)	Pebble Design Basketball
LastCost	Double	10.00
UnitPrice	Double	10.00
SellingPrice	Double	14.99
QuantityInStock	INT	20
MinimumQuantityToStock	INT	35
Deleted	Bool/Char	F

1] The record above represents Part IV → ID = 112.

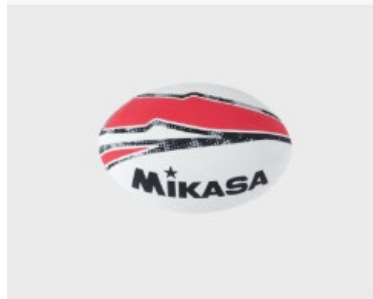
 Building Sample Record Part V



Inventory		Sample Part V
ID	AUTO INT	311
BarCode	CHAR(12)	053100444269
Name	VarChar(50)	Mikasa Football
Description	VarChar(300)	High quality football
VendorID	INT	2
VendorItemID	VarChar(12)	F6000
VendorDescription	VarChar(50)	Football Official Size
LastCost	Double	16.00
UnitPrice	Double	16.00
SellingPrice	Double	29.99
QuantityInStock	INT	5
MinimumQuantityToStock	INT	5
Deleted	Bool/Char	F

- 26] Check to make sure your DATA toggle works. Show the form without Data.
- 27] RECOMMEND → Exit Visual Studio.
- 27] RECOMMEND → Save a copy of project Nike → call it **Nike.5**

Building Sample Record Part VI



RNB7 – OFFICIAL SIZE

\$39.99

Inventory		Sample Part VI
ID	AUTO INT	311
BarCode	CHAR(12)	077770444269
Name	VarChar(50)	Mikasa Rugby Ball
Description	VarChar(300)	Red & Black Rugby Ball Official Size
VendorID	INT	2
VendorItemID	VarChar(12)	RNB7
VendorDescription	VarChar(50)	Rugby Ball Official Size
LastCost	Double	22.50
UnitPrice	Double	22.50
SellingPrice	Double	39.99
QuantityInStock	INT	3
MinimumQuantityToStock	INT	2
Deleted	Bool/Char	F

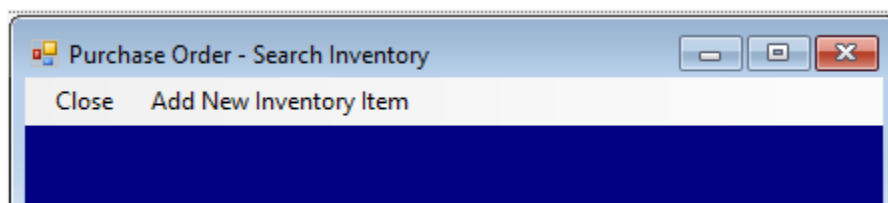
Part VI

Restart Visual Studio With project → Nike

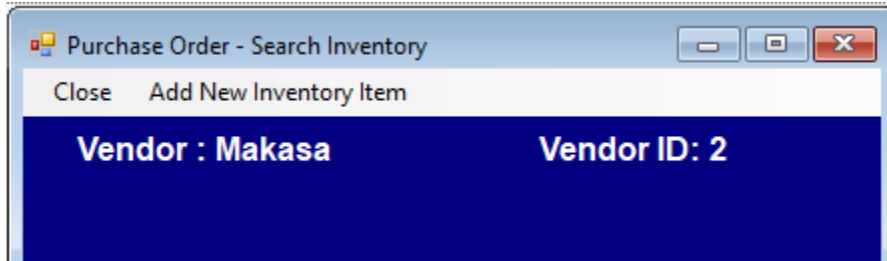
- 1] Create a Windows Form → call it **PurchaseOrderSearch.cs** whose height is the same as **PurchaseOrder.cs**.

```
private void Search()
{
    PurchaseOrderSearch SearchForm = new PurchaseOrderSearch();
    SearchForm.Show();
}
```

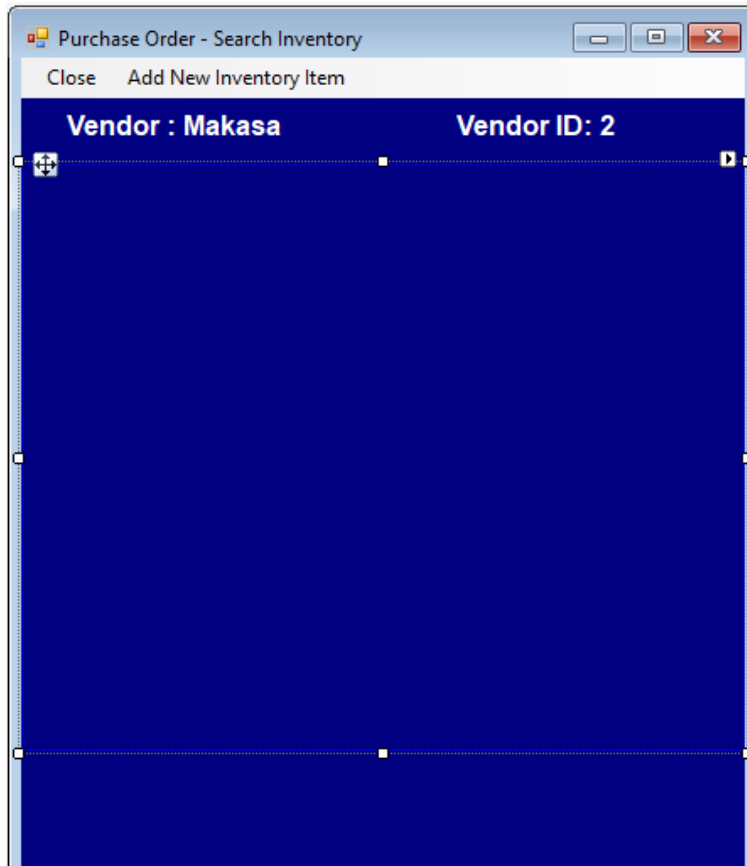
- 2] Create form Search().
- 3] Since we are going to work with this form, place a call to function Search at the bottom of your constructor; we will remove it later. → **Search();**
- 4] Manually open form **PurchaseOrder** to a location in the top left corner of your screen (maybe 10,10).
- 5] Manually open form **PurchaseOrderSearch** at the same height – place it next to the first form. Make your two forms open something like the graphic above.



- 6] Place a title at the top of the form.
- 7] Make the background color the same as that on form PurchaseOrder.cs
- 8] Add a menustrip with the choices seen above.



- 9] In my prototype demonstration, I am choosing to use Vendor 2 (Mikasa). Create a label that contains your Vendor Name and ID; place it below your MenuStrip.



- 10] Create **pnIAIData**; place it beneath the label containing your Vendor Information.

- 11] Place a label in the top of pnlAllData → include the text above.

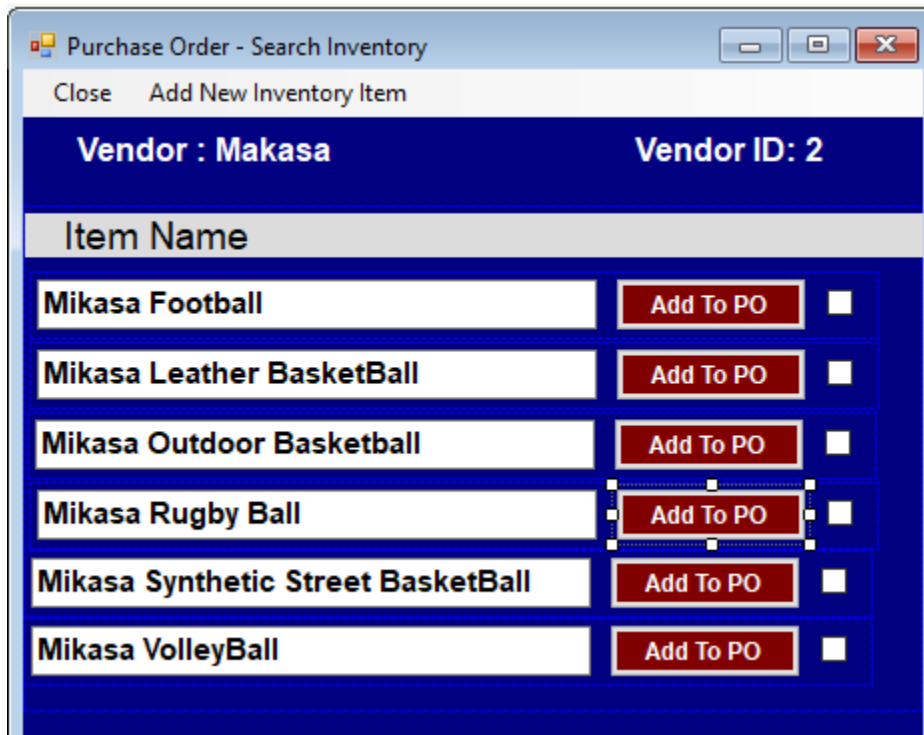
Mikasa Football
Mikasa Leather BasketBall
Mikasa Outdoor Basketball
Mikasa Rugby Ball
Mikasa Synthetic Street BasketBall
Mikasa VolleyBall

- 12] I have 6 data records that might be used in my Prototype Demonstration. They should appear in alphabetical order. Either use my data records or sort your own.

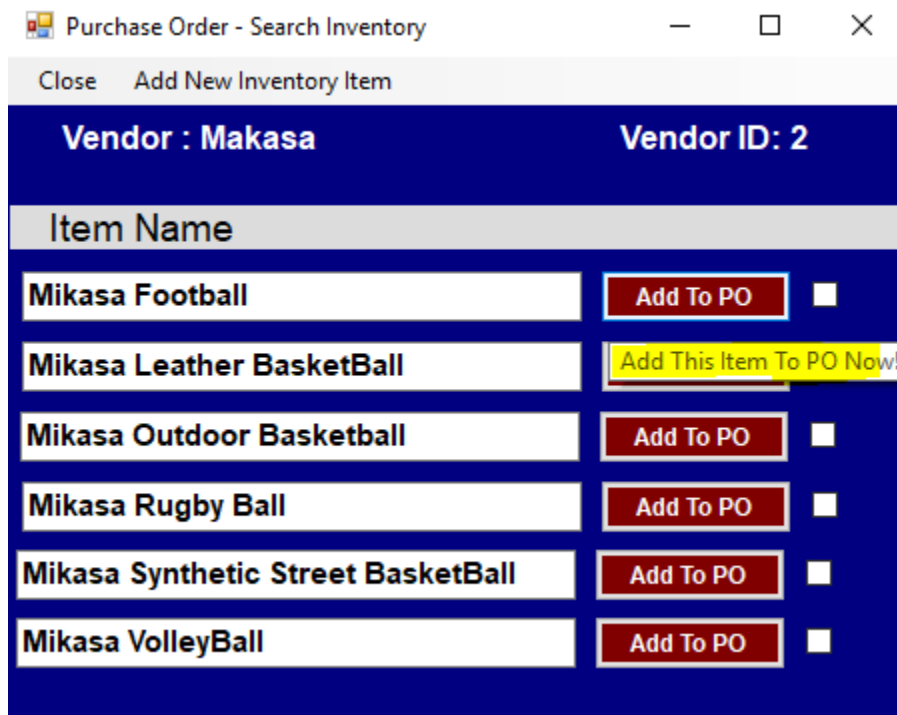
- 13] Were I to implement this form, I would use a DataGrid control and fill it from a database or a file; the DataGrid control is not easy/quick to manipulate for my prototype; I am going to use some panels because I can show the stakeholders something similar to what they will get. I tell them that their final product will not be "exactly like this".

- 14] Create a panel, called **pnlRecord1**; place it beneath the label.

- 15] Add a txtbox, button, and checkbox to panel **pnlRecord1**.



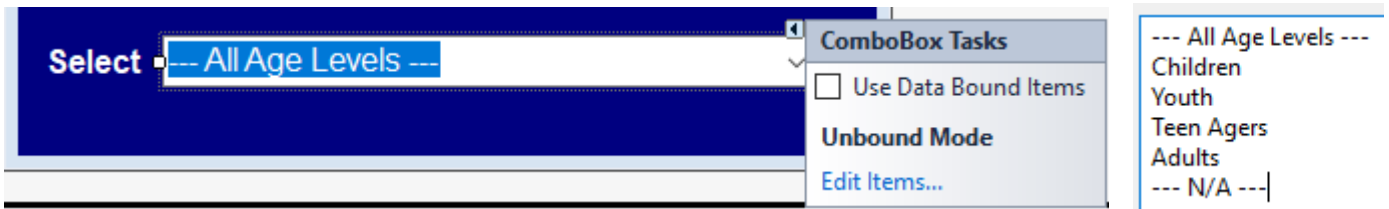
16] Duplicate this panel → Add all 6 of your records.



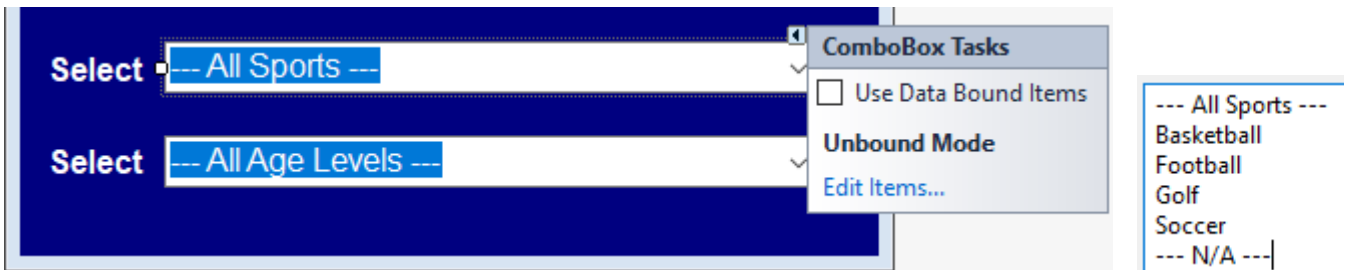
17] Include the tool tips above on all six of the Add To buttons.



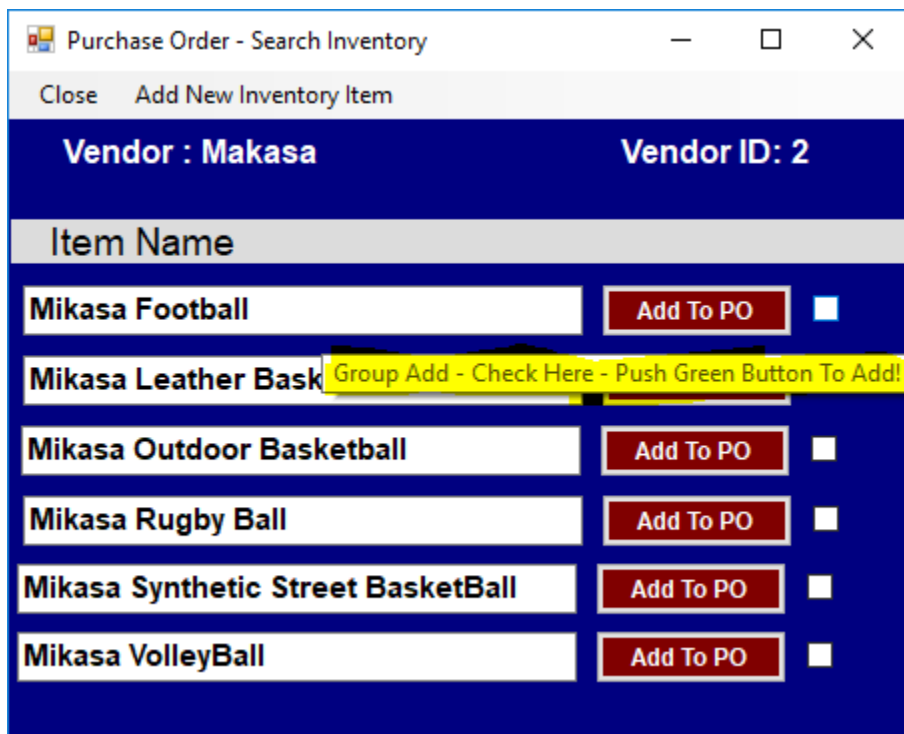
18] Add this button near the bottom of your form.



19] When we display all of the items for this Vendor, it might be a lot to select from. Filters are necessary to reduce this population. Add a filter to the bottom of your form.



20] Add a second filter to the bottom of your form.



21] Add the tool tip above to each of your six Check Boxes.

22] Check to make sure your DATA toggle works. Show the form without Data.

23] RECOMMEND → Exit Visual Studio.

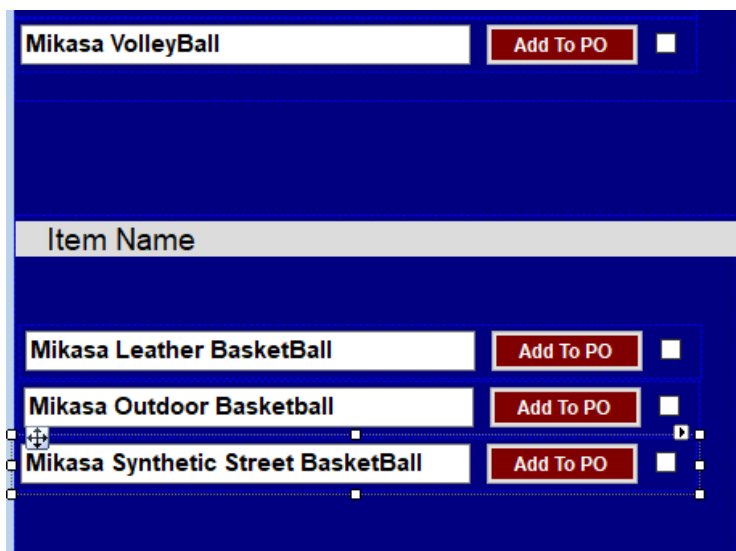
24] RECOMMEND → Save a copy of project Nike → call it **Nike.6**

Part VII

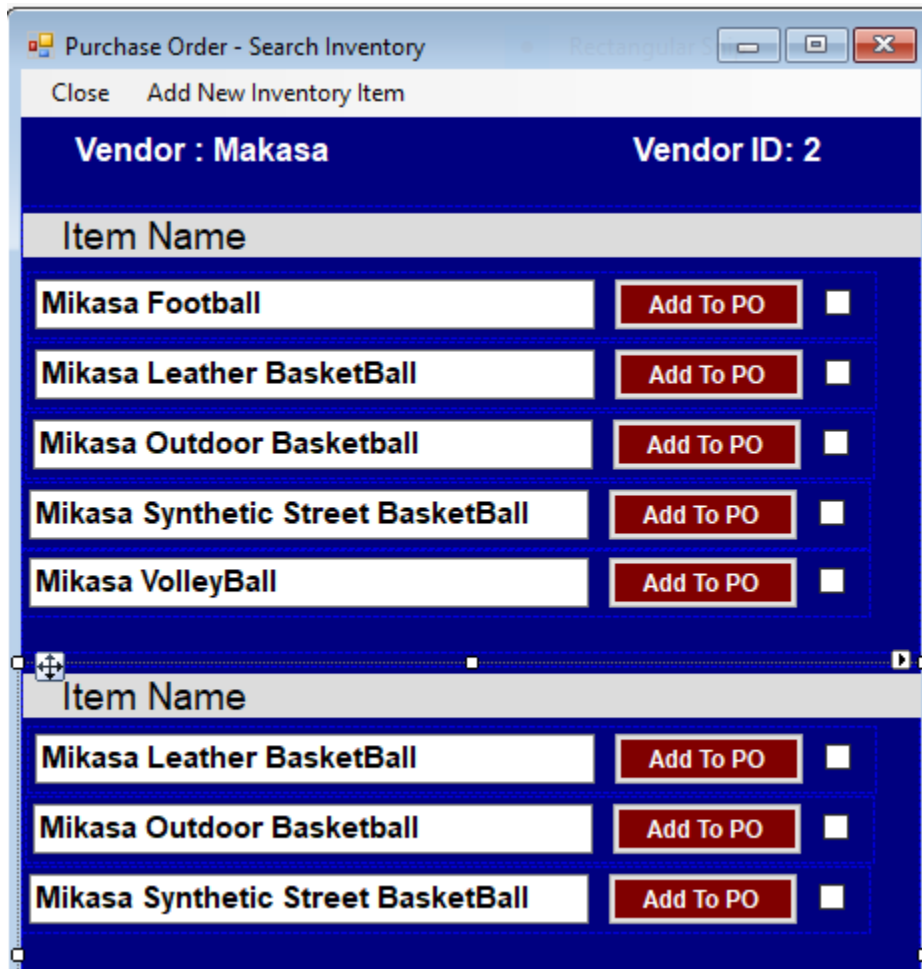
Restart Visual Studio With project → Nike



- 1] Duplicate pnlAllData → Call it pnlBasketBalls. If you have different data, do something similar.



- 2] Delete the non-basketball entries from pnlBasketBalls.



- 3] Organize the data in pnlBasketBalls.

```

public PurchaseOrderSearch()
{
    InitializeComponent();
    pnlBasketBalls.Hide();
    pnlAllData.Location = new Point(0, 70);
}

```

- 4] Add the code above to the constructor.

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    pnlAllData.Hide();
    pnlBasketBalls.Show();
    pnlBasketBalls.Location = new Point(0, 70);
}

```

- 5] Add the code above to the combobox in which you might apply the BasketBall filter.

Push --> To Add All Checked Items To PO

Select --- All Sports ---

Select Adults

Purchase Order - Search Inventory

Close Add New Inventory Item

Vendor : Makasa Vendor ID: 2

Item Name

Mikasa Football	Add To PO	<input type="checkbox"/>
Mikasa Leather BasketBall	Add To PO	<input type="checkbox"/>
Mikasa Outdoor Basketball	Add To PO	<input type="checkbox"/>
Mikasa Synthetic Street BasketBall	Add To PO	<input type="checkbox"/>
Mikasa VolleyBall	Add To PO	<input type="checkbox"/>

- 6] The filters, at the bottom of the Search Inventory are intended to drill down into the data. These filters would differ from application to application. There may be as many as half a dozen filters.

Push --> To Add All Checked Items To PO

Select Basketball

Select Adults

Purchase Order - Search Inventory

Close Add New Inventory Item

Vendor : Makasa Vendor ID: 2

Item Name

Mikasa Leather BasketBall	Add To PO	<input type="checkbox"/>
Mikasa Outdoor Basketball	Add To PO	<input type="checkbox"/>
Mikasa Synthetic Street BasketBall	Add To PO	<input type="checkbox"/>

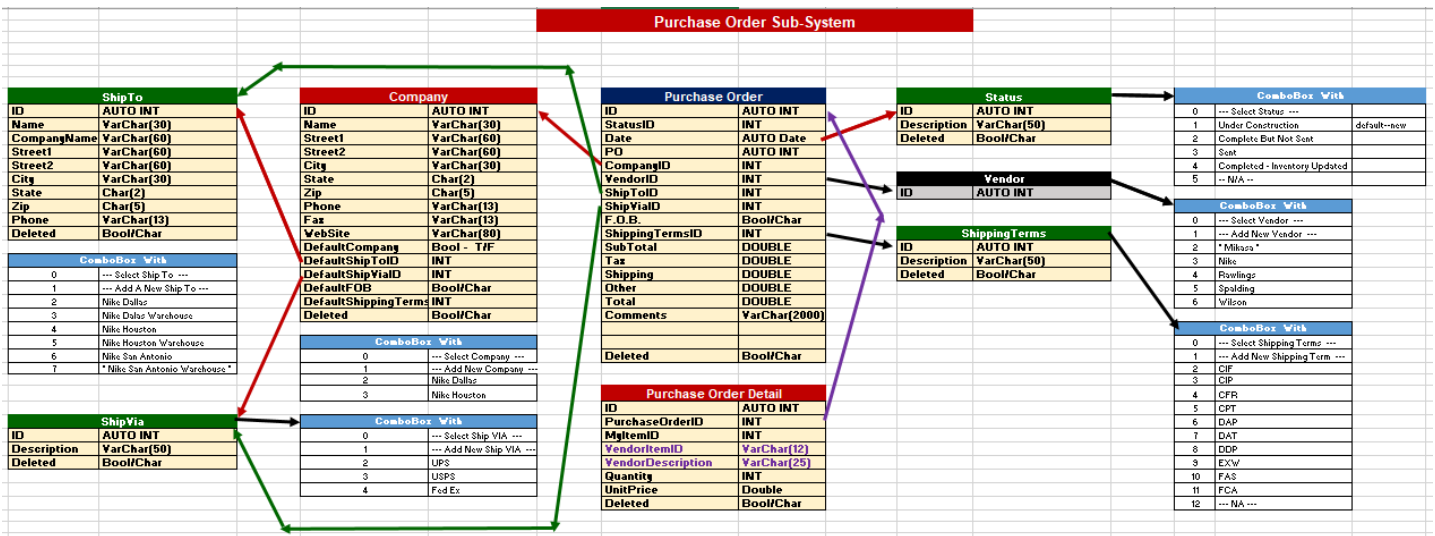
- 7] During the prototype demonstration, you might show the user what happens when you change the sports filter to Basketball → note how the filter reduces the number of items that you have to examine in the effort to quickly and easily complete your purchase order.
- 8] RECOMMEND → Exit Visual Studio.
- 9] RECOMMEND → Save a copy of project Nike → call it **Nike.7**

Purchase Order	
ID	AUTO INT
StatusID	INT
Date	AUTO Date
PO	AUTO INT
CompanyID	INT
VendorID	INT
ShipToID	INT
ShipVialID	INT
F.O.B.	Bool/Char
ShippingTermsID	INT
SubTotal	DOUBLE
Tax	DOUBLE
Shipping	DOUBLE
Other	DOUBLE
Total	DOUBLE
Comments	VarChar(2000)
Deleted	Bool/Char
Purchase Order Detail	
ID	AUTO INT
PurchaseOrderID	INT
MyItemID	INT
VendorItemID	VarChar(12)
VendorDescription	VarChar(25)
MyName	VarChar(25)
Quantity	INT
UnitPrice	Double
Deleted	Bool/Char

- 1] Return to [Purchase-Order-2.xlsx](#) → Add the Supporting Class for **Purchase Order Detail**. The items in purple are optional.

If you include the VendorItemID and the VendorDescription in the detail, you do not have to read all of the Inventory Table records when you display the Purchase Order; this does require a small amount of disk space, but you do get an increase in speed performance. I would recommend it.

Suppose the Purchase order has 15 items. If you do include the VendorItemID and the VendorDescription in the detail, it will require 15 additional reads (from the Inventory Table) to display the Purchase Order; this will save a small amount of disk space, I do not recommend it.



2] The total Purchase Order Class Diagram is shown above.

3] Save **Purchase-Order-2.xlsx**

Place Nike, Nike.5, Nike.6, Nike.7 & Purchase-Order-2.xlsx in Your Work In Progress Folder on Mars.

Place Nike, Nike.5, Nike.6, Nike.7 & Purchase-Order-2.xlsx on your Personal Computer

Place Nike, Nike.5, Nike.6, Nike.7 & Purchase-Order-2.xlsx on your Network Drive or a Flash Drive