

VBA Automated Invoicing System – Excel

In this project, we will be using Excel to automate data entry tasks at the click of a button in a simulated business environment. We will be using several files, including a request form, an invoice template, a product catalog, and a customer database.

Contents

Skills Demonstrated:	1
Required Materials:	1
Simulated Business Environment	2
Walkthrough:	2
Analyzing our Materials	2
Formulating our proposal	6
Additional Features	9
The PO Entry Window	10
Invoice Records and VBA.....	13
Results	21

Skills Demonstrated:

- Communication
- Requirements gathering
- VLOOKUP
- Macros
- VBA
- Data Entry
- Managing Databases
- Conditional formatting

Required Materials:

- The compiled materials can be found at [this link](#) or the [Portfolio Site](#), which includes:
 - A template file from our company.
 - A customer database file.
 - A product catalog file.
 - An order example.

Simulated Business Environment

For the purposes of this project, we will be simulating the role of a data entry clerk for our simulated business *Robison Corporation™*. Our job at Robison corporation will be to enter in data to invoices from PO forms, however with our experience we are aware of how we can cut down the time it takes to enter in the data and can even provide additional features to the business.

We will begin the project with all files being separate and without any references to one another. Through communicating with various individuals at our company, including our boss, we will be able to consolidated these sheets into one file and use formulas and references to drastically cut down on time. Additionally, we will be requested to adjust our file into a full invoicing system after the business integrates a simulated PO entry window to input its information into our file. The end product of our work will include the following:

- An invoice number tracker.
- The invoice template which fills in customer and product information from the simulated PO entry window.
- Several buttons on the invoice sheet which will allow as to save the invoice record, clear the invoice, or save the invoice as an excel file or pdf.
- An invoice record sheet which will track all invoices and whether or not they have been processed.

Walkthrough:

Analyzing our Materials

Congratulations! You've recently been hired on as a data entry clerk at *Robison Corporation™*! Your boss has tasked you with reading PO form information from an outdated integration that neither of you really understand, but will provide you with enough information to fill in the invoices. With the PO request information, you will find the related entries in the product catalog and customer database and fill out the appropriate information using the provided invoice template. You will then save the invoice as a pdf or excel file and print it out if needed. With this all in mind, let's take a look at our materials:

- The Product Catalog

ID	Product Name	Product Description	Product Price
1	Spark Plugs	Genuine OEM spark plugs designed to ignite the air-fuel mixture in your engine for optimal performance.	\$4.99
2	Brake Pads	High-quality ceramic brake pads engineered to provide reliable stopping power and minimize brake dust.	\$39.99
3	Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	\$9.99
4	Air Filter	High-flow air filter that improves airflow to your engine, enhancing horsepower and acceleration.	\$19.99
5	Wiper Blades	Durable wiper blades with a precision rubber wiping edge for streak-free visibility in rain and snow.	\$14.99
6	Cabin Air Filter	Cabin air filter that removes dust, pollen, and other allergens, ensuring clean air inside your vehicle.	\$24.99
7	Engine Coolant	Long-life engine coolant formulated to protect your engine from freezing, boiling, and corrosion.	\$12.99
8	Serpentine Belt	Heavy-duty serpentine belt made from durable materials for reliable power transmission in your vehicle.	\$29.99
9	Fuel Filter	High-efficiency fuel filter that removes impurities and contaminants, ensuring clean fuel reaches your engine.	\$19.99
10	Oxygen Sensor	Precision-engineered oxygen sensor that monitors oxygen levels in your exhaust system for optimal fuel efficiency.	\$49.99
11	Radiator Hose	Reinforced rubber radiator hose designed to withstand high temperatures and pressures in your vehicle's cooling system.	\$16.99
12	Thermostat	Thermostat that regulates engine temperature for efficient performance and prevents overheating.	\$10.99
13	Ignition Coil	Ignition coil that delivers high-voltage electricity to the spark plugs, ensuring reliable ignition and engine performance.	\$29.99
14	Brake Rotors	Premium brake rotors made from high-quality materials for consistent braking performance and reduced noise.	\$49.99
15	Timing Belt	Timing belt that synchronizes the rotation of the engine's camshaft and crankshaft, ensuring precise engine operation.	\$69.99
16	Wheel Bearings	Precision-engineered wheel bearings that reduce friction and support smooth wheel rotation for a comfortable ride.	\$34.99
17	Tie Rod Ends	Heavy-duty tie rod ends that connect the steering rack to the steering knuckle, ensuring responsive steering control.	\$24.99
18	Ball Joints	High-strength ball joints that connect the control arms to the steering knuckles, providing stable wheel alignment.	\$29.99
		Gas-charged shock absorbers that dampen vibrations and provide a smooth ride	

The catalog contains 100 rows of car part information with the following rows:

- **ID:** The product ID
- **Product Name:** The name of the product.
- **Product Description:** A short description of the product.
- **Product Price:** The price of the product in USD.
- Customer Database

Company Name	Address 1	Address 2	Contact Email	Telephone	Bank	Account Number	ID
XYZ Corporation	20000 Northwood St	Phoenix, AZ 85048	john.smith@xyzcorp.com	(123) 456-7890	Chase Bank	1234567890123	1
ABC Industries	300 Oak Avenue	Springfield, IL 62701	sarah.doe@abc.com	(234) 567-8901	Bank of America	2345678901234	2
Acme Co	400 Maple Street	Smalltown, NY 12345	mike.jones@acme.com	(345) 678-9012	Wells Fargo	3456789012345	3
Summit Enterprise	500 Cedar Road	Suburbia, TX 76543	emily.smith@summit.com	(456) 789-0123	Citibank	4567890123456	4
Pacific Logistics	600 Elm Lane	Downtown, FL 32123	david.brown@paclog.com	(567) 890-1234	U.S. Bank	5678901234567	5
Cascade Solutions	700 Pine Drive	Metropolis, GA 45678	jessica.white@cascade.com	(678) 901-2345	PNC Bank	6789012345678	6
Golden Gate Group	800 Sycamore Blvd	Cityville, WA 98765	andrew.johnson@goldengate.com	(789) 012-3456	TD Bank	7890123456789	7
Sunburst Services	900 Cedar Avenue	Countryside, OH 54321	lisa.johnson@sunburst.com	(890) 123-4567	Capital One	8901234567890	8
Bright Ideas Inc.	1000 Birch Street	Midtown, NC 23456	mark.wilson@brightideas.com	(901) 234-5678	HSBC	9012345678901	9
Silver Star Corp	1100 Willow Road	Riverside, CA 67890	ashley.miller@silverstar.com	(012) 345-6789	Barclays	123456789012	10
Swift Solutions	1200 Pine Lane	Uptown, TX 23456	brian.james@swift.com	(123) 456-7890	Goldman Sachs	1234567890123	11
Evergreen Enterpri	1300 Cedar Street	Lakeside, CO 34567	amy.jones@evergreen.com	(234) 567-8901	Morgan Stanley	2345678901234	12
Mountain View Ltd	1400 Elm Avenue	Hillside, CA 91006	info@mountainview.com	(345) 678-9012	JPMorgan Chase	3456789012345	13
Oliver Industries	1500 Oak Street	Parkville, MD 21234	jsteel@oliveri.com	(456) 789-0123	Bank of America	4567890123456	14
Alpha Associates	1600 Maple Avenue	Lakeside, MT 59868	contact@alphaassociates.com	(567) 890-1234	Wells Fargo	5678901234567	15
Omega Solutions	1700 Birch Road	Uptown, CO 80202	jjacobs@omegasolutions.com	(678) 901-2345	U.S. Bank	6789012345678	16
Zenith Enterprises	1800 Cedar Lane	Cityville, NV 89501	gary.king@zenithenterprises.com	(789) 012-3456	PNC Bank	7890123456789	17
Starlight Systems	1900 Pine Drive	Metropolis, IL 62960	jordanu@starlightsystems.com	(890) 123-4567	TD Bank	8901234567890	18
Elite Solutions	2000 Sycamore Blvd	Downtown, TX 77002	sholmes@elitesolutions.com	(901) 234-5678	Capital One	9012345678901	19
Global Group	2100 Cedar Avenue	Countryside, CA 90210	ffhorrigan@globalgroup.com	(012) 345-6789	HSBC	123456789012	20
Fusion Corporation	2200 Birch Street	Springfield, IL 62702	ceastwood@fusioncorp.com	(123) 456-7890	Barclays	1234567890123	21
Liberty Ltd.	2300 Elm Lane	Parkville, MD 21234	calvin.morriganon@libertyltd.com	(234) 567-8901	Goldman Sachs	2345678901234	22
Phoenix Partners	2400 Willow Road	Uptown, AZ 85001	kyle.steward@phoenixpartners.com	(345) 678-9012	Morgan Stanley	3456789012345	23
Diamond Dynamics	2500 Pine Lane	Lakeside, MT 59868	abigail.lockwood@diamonddynamic.com	(456) 789-0123	JPMorgan Chase	4567890123456	24
Infinity Inc.	2600 Cedar Road	Cityville, NV 89502	lyjung@infinityinc.com	(567) 890-1234	Bank of America	5678901234567	25
Nova Solutions	2700 Sycamore Blvd	Metropolis, IL 62961	sean.hanes@novasolutions.com	(678) 901-2345	Wells Fargo	6789012345678	26
Sunflower Systems	2800 Cedar Avenue	Countryside, CA 90211	vincent.richardson@sunflowersys.com	(789) 012-3456	U.S. Bank	7890123456789	27
Platinum Partners	2900 Elm Lane	Downtown, TX 77003	bethany.bell@platinumpartners.com	(890) 123-4567	PNC Bank	8901234567890	28
Quantum Quotient	3000 Pine Drive	Springfield, IL 62703	insert.name@quantumquotient.com	(901) 234-5678	TD Bank	9012345678901	29
Robison Corp	5678 North Way	Phoenix, AZ 85048	patrickrobison17@gmail.com	(012) 345-6789	Capital One	123456789012	30

The customer database contains 30 rows of company information, including the following:

- **Company Name:** The customer company name.
- **Address 1:** The first portion of the company's address.
- **Address 2:** The second portion of the company's address.
- **Contact Email:** The email for the company's contact.
- **Telephone:** The contact's phone number.
- **Bank:** The bank used by the company's account.
- **Account Number:** The simulated routing number of the company.
- **ID:** The ID of the customer company.
- The PO information we receive

ORDER

Company ID 14
Order No. PO00034
Order Date 5/18/2024

Product ID	Quantity
93	3
3	3
48	1
67	2
14	5

Notes

Need as soon as possible

A barebones amount of information that only barely gives us what we need to fill in the information in the invoice. From what we saw earlier we have the date the order was placed, a reference to the company who placed the purchase, references to products, and some customer notes.

- Robison Corporation Invoice Template

Robison Corp		Invoice																																																															
<table><tr><td>FROM:</td><td>BILL TO:</td></tr><tr><td>Robison Corp</td><td></td></tr><tr><td>5678 North Way</td><td></td></tr><tr><td>Phoenix, AZ 85048</td><td></td></tr><tr><td>patrickrobison17@gmail.com</td><td></td></tr></table>		FROM:	BILL TO:	Robison Corp		5678 North Way		Phoenix, AZ 85048		patrickrobison17@gmail.com		<table><tr><td>Invoice No.</td><td></td></tr><tr><td>PO No.</td><td></td></tr><tr><td>Date</td><td></td></tr><tr><td>Due Date</td><td></td></tr></table>		Invoice No.		PO No.		Date		Due Date																																													
FROM:	BILL TO:																																																																
Robison Corp																																																																	
5678 North Way																																																																	
Phoenix, AZ 85048																																																																	
patrickrobison17@gmail.com																																																																	
Invoice No.																																																																	
PO No.																																																																	
Date																																																																	
Due Date																																																																	
<table><thead><tr><th>Product</th><th>Description</th><th>Quantity</th><th>Unit Price</th><th>VAT</th><th>Total (Tax exc.)</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table>						Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)																																																						
Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)																																																												
Customer Comments: <div></div>				<table><tr><td>Subtotal USD</td><td>\$0.00</td></tr><tr><td>Tax</td><td>\$0.00</td></tr><tr><td>Total USD</td><td>\$0.00</td></tr></table>		Subtotal USD	\$0.00	Tax	\$0.00	Total USD	\$0.00																																																						
Subtotal USD	\$0.00																																																																
Tax	\$0.00																																																																
Total USD	\$0.00																																																																
Company Information		Payment Information																																																															
Company ID		Bank Name																																																															
Phone No.		Account Number																																																															
Email		Notes																																																															

The template we will be using to fill out invoices, which can hold a maximum of ten different products. We can fill out billing information based on the company, product information based on the order, and other information from our order form. As it stands there are no formulas or references being used aside from the due date being set to 30 days ahead, the VAT column placing a 20% tax on the unit price for the amount ordered, and the totals of the transactions.

With all these materials in mind, we begin our work and slowly come to realize how inefficient this process is. The catalog and customer database files are each held several folders deep in our file directory, and looking between the several files and scrolling up and down to refer to the IDs in the order form takes far too long.

Formulating our proposal

Initially the work for each form takes on average **6 minutes** to complete without any formulas and taking no other steps to speed things up. After some time, you pin the location of the catalog and customer database files to your commonly accessed files to get to them quicker, and use CTRL+F to find the ID references in each of the other files to quickly input them into the template, cutting the time down to **4 minutes**. You know for a fact that this time can go down further if you include all the different sheets into one file, but you only have authorization to include the product catalog as the customer database is protected information.

You bring this information to your boss and propose that you can cut this time down to **2 minutes** if you can include all sheets and use formulas in the template. We can prove this by implementing some of these features using the catalog information. With **VLOOKUP**, we can automatically fill in description and unit price information from our product catalog. Here is an example of the formula we can use:

```
=IF(C11="", "", VLOOKUP(C11, Table1[["#All"]], [Product Name]:[Product Price], 2, FALSE))
```

Translation: If C11 (The Product column of our invoice) is empty, then keep the description column empty. Else, search the product catalog sheet for the listed product and look at columns 2-4, using the second column from our selection (column 3, the description) to find the answer, and it must be an exact match.

(Note: When using VLOOKUP, the entry we are looking for must be in the leftmost column to work. Though you could use XLOOKUP instead to work around this, I do not have access to it in my version of excel.)

The formula will autofill in the product description information assuming a matching product from the product column can be found. We can use a similar formula for the product price:

```
IF([C11="", "", VLOOKUP(C11, Table1[["#All"]], [Product Name]:[Product Price]), 3, FALSE))
```

This will do the same thing except extract the product price information from our selected columns in the VLOOKUP. Now when we input a product it will automatically fill in both of these selections, but we can make things go even faster.

When selecting a product cell, we can have the option to open a list of all the products in the product catalog so that we don't need to type in the entry manually. We can do this by selecting the product column and then navigating to the Data Ribbon → Selecting Data Validation. The following window will open:


Data Validation ? X

Settings Input Message Error Alert

Validation criteria

Allow: List ☒ Ignore blank ☒ In-cell dropdown

Data: between

Source: 

☐ Apply these changes to all other cells with the same settings

Clear All OK Cancel

In allow, we will select the List option, and in the source entry box we will select the range of cells that encompasses our product name column. By hitting OK, we can then select from our list in each of the cells.

Product	Description
Engine Oil	High-quality engine oil that lubricates engine components, reduces friction and wear, and provides thermal stability.
Spark Plugs	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.
Brake Pads	Replacement supercharger that increases engine power and efficiency by compressing air for improved combustion.
Oil Filter	Replacement head gasket that seals the cylinder head to the engine block,
Air Filter	
Wiper Blades	
Cabin Air Filter	
Engine Coolant	
Serpentine Belt	
Supercharger	

After your explanation, your boss agrees with you and holds a meeting with the company's controls personnel for authorization to include the customer database in the file. Authorization is granted and you now have the ability to include the customer database to perform the same changes outlined prior. Using the data validation list functionality from earlier, we can now quickly select the company on the PO form.

FROM:	BILL TO:
Robison Corp	Oliver Industries
5678 North Way	Oliver Industries
Phoenix, AZ 85048	Alpha Associates
patrickrobison17@gmail.com	Omega Solutions
	Zenith Enterprises
	Starlight Systems
	Elite Solutions
	Global Group
	Fusion Corporation
Product	High-quality engine oil that lubricates

We can then fill in all of the company information by utilizing the same VLOOKUP syntax from earlier:

`=IF(D5 = "", "", VLOOKUP(D5, 'Customer Database'!A:G, 2, FALSE))`

Since the company name, D5, won't be changing, we lock it in with F4. We do this with the companies address, email, phone, and banking information. Now by using the lists in the cells we selected previously, we can quickly fill in most relevant information that we would have to from the PO form. The only remaining information we would have to manually input is the customer comments, the invoice number, the PO number, the date, and the quantity for each product row.

As a result of our work, we successfully cut the time of our job in half which allows us to process invoices faster.

Robison Corp				Invoice	
FROM:		BILL TO:		Invoice No.	1
Robison Corp		Oliver Industries		PO No.	PO00034
5678 North Way		1500 Oak Street		Date	5/18/2024
Phoenix, AZ 85048		Parkville, MD 21234		Due Date	6/17/2024
probison@robco.com		jsteel@oliveri.com			
Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)
Engine Oil	High-quality engine oil that lubricates engine components, reduces friction and wear, and provides thermal stability.	3	\$29.99	\$17.99	\$89.97
Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	3.00	\$9.99	\$5.99	\$29.97
Supercharger	Replacement supercharger that increases engine power and efficiency by compressing air for improved combustion.	1.00	\$399.99	\$80.00	\$399.99
Head Gasket	Replacement head gasket that seals the cylinder head to the engine block, preventing coolant	2.00	\$49.99	\$20.00	\$99.98
Brake Rotors	Premium brake rotors made from high-quality materials for consistent braking performance and reduced noise.	5.00	\$49.99	\$49.99	\$249.95
Customer Comments:				Subtotal USD	\$869.86
Need as soon as possible.				Tax	\$173.97
				Total USD	\$1,043.83
Company Information		Payment Information			
Company ID	14	Bank Name	Bank of America		
Phone No.	(456) 789-0123	Account Number	4567890123456		
Email	jsteel@oliveri.com	Notes			

Additional Features

Your boss is impressed with your work, and seeing how you were able to create the prior solution, they make an additional request of you. As it stands, customers have been complaining from only having a maximum of 10 different products they can select from for each invoice and would like more.

Additionally, it's a waste of ink to include empty rows, so they request you to hide rows if they aren't being used.

We happily oblige and begin our work to include more rows and make them dynamic. The new number of rows we will include is 30 rows per invoice as agreed upon by various stakeholders. For the solution we have in mind, we also insert a new column into our product entry table: LineHelper.

By selecting on our table and going to the Insert tab and selecting data, it will automatically format the information into a table with filters included. Using an automatic filter, we can make the rows of this table dynamic based on whether or not there is information in the row above. In the line helper column, we will include the following formula:

=IF(C10="",TRUE,FALSE)

Translation: If C10 (The product column cell above this row) is empty, set the result to true. Else, set to false. Here is the result:

[illegible]

The first row in our list is set to false because the cell above it is filled with the product column title. If we were to input an entry into the first row, the next row's line helper would adjust to false. Next, we can adjust the line tool's filter to filter out all results with true in them. This will remove all empty rows except for one to allow additional entries and dynamically update. Stylistically, we can hide column H where the line tool is housed and then go to table design and hide the filter button. However, when we input a new row at this point, it will not adjust yet. This is because we need to implement the auto filter functionality.

We can right click on our sheet and select 'View Code' to open up VBA. We will select Worksheet and change, and input the following code:

Me.ListObjects("Table7").AutoFilter.ApplyFilter (Table# will change depending on the table you select)

```

VBAAutomatedInvoices.xlsm - Sheet6 (Code)
Worksheet
Change

Private Sub Worksheet_Change(ByVal Target As Range)
    Me.ListObjects("Table7").AutoFilter.ApplyFilter
End Sub

```

This will now enable the table to automatically update filters (in this case the line helper filter) with new row entries. We'll input the example order from earlier to see how it looks:

Robison Corp				Invoice																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #4f81bd; color: white; padding: 2px;">FROM:</td> <td style="background-color: #4f81bd; color: white; padding: 2px;">BILL TO:</td> </tr> <tr> <td style="padding: 2px;">Robison Corp</td> <td style="padding: 2px;">Oliver Industries</td> </tr> <tr> <td style="padding: 2px;">5678 North Way</td> <td style="padding: 2px;">1500 Oak Street</td> </tr> <tr> <td style="padding: 2px;">Phoenix, AZ 85048</td> <td style="padding: 2px;">Parkville, MD 21234</td> </tr> <tr> <td style="padding: 2px;">erobison@robco.com</td> <td style="padding: 2px;">jsteel@oliveri.com</td> </tr> </table>		FROM:	BILL TO:	Robison Corp	Oliver Industries	5678 North Way	1500 Oak Street	Phoenix, AZ 85048	Parkville, MD 21234	erobison@robco.com	jsteel@oliveri.com	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #4f81bd; color: white; padding: 2px;">Invoice No.</td> <td style="padding: 2px;">1</td> </tr> <tr> <td style="background-color: #4f81bd; color: white; padding: 2px;">PO No.</td> <td style="padding: 2px;">PO00034</td> </tr> <tr> <td style="background-color: #4f81bd; color: white; padding: 2px;">Date</td> <td style="padding: 2px;">5/18/2024</td> </tr> <tr> <td style="background-color: #4f81bd; color: white; padding: 2px;">Due Date</td> <td style="padding: 2px;">6/17/2024</td> </tr> </table>		Invoice No.	1	PO No.	PO00034	Date	5/18/2024	Due Date	6/17/2024																				
FROM:	BILL TO:																																								
Robison Corp	Oliver Industries																																								
5678 North Way	1500 Oak Street																																								
Phoenix, AZ 85048	Parkville, MD 21234																																								
erobison@robco.com	jsteel@oliveri.com																																								
Invoice No.	1																																								
PO No.	PO00034																																								
Date	5/18/2024																																								
Due Date	6/17/2024																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="background-color: #4f81bd; color: white; padding: 2px;">Product</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">Description</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">Quantity</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">Unit Price</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">VAT</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">Total (Tax exc.)</th> </tr> <tr> <td style="padding: 2px;">Engine Oil</td> <td style="padding: 2px;">High-quality engine oil that lubricates engine components, reduces friction and wear, and provides thermal stability.</td> <td style="padding: 2px;">3</td> <td style="padding: 2px;">\$29.99</td> <td style="padding: 2px;">\$17.99</td> <td style="padding: 2px;">\$89.97</td> </tr> <tr> <td style="padding: 2px;">Oil Filter</td> <td style="padding: 2px;">Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.</td> <td style="padding: 2px;">3.00</td> <td style="padding: 2px;">\$9.99</td> <td style="padding: 2px;">\$5.99</td> <td style="padding: 2px;">\$29.97</td> </tr> <tr> <td style="padding: 2px;">Supercharger</td> <td style="padding: 2px;">Replacement supercharger that increases engine power and efficiency by compressing air for improved combustion.</td> <td style="padding: 2px;">1.00</td> <td style="padding: 2px;">\$399.99</td> <td style="padding: 2px;">\$80.00</td> <td style="padding: 2px;">\$399.99</td> </tr> <tr> <td style="padding: 2px;">Head Gasket</td> <td style="padding: 2px;">Replacement head gasket that seals the cylinder head to the engine block, preventing coolant and oil leaks.</td> <td style="padding: 2px;">2.00</td> <td style="padding: 2px;">\$49.99</td> <td style="padding: 2px;">\$20.00</td> <td style="padding: 2px;">\$99.98</td> </tr> <tr> <td style="padding: 2px;">Brake Rotors</td> <td style="padding: 2px;">Premium brake rotors made from high-quality materials for consistent braking performance and reduced noise.</td> <td style="padding: 2px;">5.00</td> <td style="padding: 2px;">\$49.99</td> <td style="padding: 2px;">\$49.99</td> <td style="padding: 2px;">\$249.95</td> </tr> </table>	Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)	Engine Oil	High-quality engine oil that lubricates engine components, reduces friction and wear, and provides thermal stability.	3	\$29.99	\$17.99	\$89.97	Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	3.00	\$9.99	\$5.99	\$29.97	Supercharger	Replacement supercharger that increases engine power and efficiency by compressing air for improved combustion.	1.00	\$399.99	\$80.00	\$399.99	Head Gasket	Replacement head gasket that seals the cylinder head to the engine block, preventing coolant and oil leaks.	2.00	\$49.99	\$20.00	\$99.98	Brake Rotors	Premium brake rotors made from high-quality materials for consistent braking performance and reduced noise.	5.00	\$49.99	\$49.99	\$249.95	<table style="width: 100%;"> <tr> <td colspan="2" style="padding: 5px;"> Customer Comments: <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div> </td> <td style="padding: 5px; text-align: right;"> Subtotal USD \$869.86 Tax \$173.97 Total USD \$1,043.83 </td> </tr> </table>		Customer Comments: <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div>		Subtotal USD \$869.86 Tax \$173.97 Total USD \$1,043.83
Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)																																				
Engine Oil	High-quality engine oil that lubricates engine components, reduces friction and wear, and provides thermal stability.	3	\$29.99	\$17.99	\$89.97																																				
Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	3.00	\$9.99	\$5.99	\$29.97																																				
Supercharger	Replacement supercharger that increases engine power and efficiency by compressing air for improved combustion.	1.00	\$399.99	\$80.00	\$399.99																																				
Head Gasket	Replacement head gasket that seals the cylinder head to the engine block, preventing coolant and oil leaks.	2.00	\$49.99	\$20.00	\$99.98																																				
Brake Rotors	Premium brake rotors made from high-quality materials for consistent braking performance and reduced noise.	5.00	\$49.99	\$49.99	\$249.95																																				
Customer Comments: <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div>		Subtotal USD \$869.86 Tax \$173.97 Total USD \$1,043.83																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="background-color: #4f81bd; color: white; padding: 2px;">Company Information</th> <th style="background-color: #4f81bd; color: white; padding: 2px;">Payment Information</th> </tr> <tr> <td style="padding: 2px;"> Company ID 14 Phone No. (456) 789-0123 Email jsteel@oliveri.com </td> <td style="padding: 2px;"> Bank Name Bank of America Account Number 4567890123456 Notes </td> </tr> </table>		Company Information	Payment Information	Company ID 14 Phone No. (456) 789-0123 Email jsteel@oliveri.com	Bank Name Bank of America Account Number 4567890123456 Notes																																				
Company Information	Payment Information																																								
Company ID 14 Phone No. (456) 789-0123 Email jsteel@oliveri.com	Bank Name Bank of America Account Number 4567890123456 Notes																																								

The boss looks over and is pleased with the work. If you were to input another row in the table, a new empty one would pop up automatically. With these changes you've dramatically improved the efficiency of this process while also improving the visual appeal of the sheet.

The PO Entry Window

With the attention your solution has provided, an employee within the company adjusted the PO Entry window with your help to fill in information directly into your file so that you have the complete information and don't have to worry about the barebones order forms you previously had.

Purchase Order Entry Window
X

Customer	Robison Corp
PO No	PO00001
Date	5/27/2024

ID	Product	Price	Quantity
1	Spark Plugs	\$4.99	1

Comments
Test

Clear

Submit

Information filled in from this window will automatically flow into the excel sheet, meaning that we will no longer need to input the data into the invoice sheet as it will now flow in from the click of a button. This will remove the need to use the lists as that information will automatically be inputted into the sheet, though we will keep that functionality in the case that manual entry is required.

Though this is a simulated window, we will still go over how it functions as it introduces several new concepts that we will use for some of the remaining tasks we will be given by our simulated boss later. Currently this 'window' uses the following concepts and features we implemented previously.

- A list of all possible customers in the customer cell.
- A list for the ID column of all the product IDs in the catalog
- VLOOKUPs for the product and price based on the ID column
- A hidden LineHelper auto-filter to dynamically show rows.

However, there are a few new features in this sheet we haven't seen before that stem from the buttons of this 'window.' Each button has a macro tied to them with the following VBA code:

- Clear: This macro clears the PO table of all it's contents.

Sub ClearPO()

Range("E9,E10,E45").ClearContents ' Clear all non-table content fields.

Dim line As Range

Set line = Sheet6.Range("C14") ' Selects the ID column in the PO table.

Do While line.Value <> "" ' While there are still entries ...

line = "" ' Remove the ID and quantity from the table.

line.Offset(0, 3) = ""

Set line = line.Offset(1, 0)

Loop

End Sub

- Submit: This button will take all the data that is in the PO form and write it into the invoice sheet.

Sub POForm()

' Clear the invoice if there is any information.

ClearInvoice

' Insert the new invoice number

InvNumber

Dim cust As String

Dim poNum As String

Dim invDate As Date

Dim comments As String

Dim commentBox As Range

Dim infoBox As Range

' Save all information from the PO form

cust = Range("E9")

poNum = Range("E10")

invDate = Range("E11")

comments = Range("E45")

' Input the comment box information to the invoice form.

Set commentBox = Sheet5.Range("C43")

commentBox = comments

' Input the non-product information into the invoice.

Set infoBox = Sheet5.Range("D5")

infoBox = cust

Set infoBox = Sheet5.Range("G5")

infoBox = poNum

Set infoBox = Sheet5.Range("G6")

infoBox = invDate

Dim Product As Range

Dim Quantity As Range

Dim line As Range

' Setting the product and quantity column ranges for the PO sheet.

```
Set Product = Sheet6.Range("D14")
```

```
Set Quantity = Sheet6.Range("F14")
```

' Setting the range for the product entry table in the invoice.

```
Set line = Sheet5.Range("C11")
```

```
Do While Product.Value <> "" ' While there are still entries in the product column of the PO form ...
```

```
    line = Product ' Copy the information to the invoice.
```

```
    line.Offset(0, 2) = Quantity
```

```
    Set Product = Product.Offset(1, 0) ' Move on to the next row.
```

```
    Set Quantity = Quantity.Offset(1, 0)
```

```
    Set line = line.Offset(1, 0)
```

```
Loop
```

```
Sheet5.Activate ' Switch over to the invoice sheet.
```

```
End Sub
```

There are some references to other macro functions within this macro, but we will go over them later. Again, these buttons give us an idea of what we'll be working with in our later requests by writing information over into other sheets while also simulating the business aspect of things.

Invoice Records and VBA

With the work that you've done, your boss is looking to shift your priorities and tasks so that you'll be in a position where you can continue to provide the business solutions similar to this invoice sheet. A new data entry clerk is in the process of being hired on, but before the work of data entry shifts over to them, your boss comes to you with an additional request. They want a record of the invoices to be kept on the file seeing as most of the other sheets are already present in this file. They ideally want the saving of records to be as simple as possible and to be done in either a PDF or Excel file so that anyone could do the job. They task you with making this more efficient, and you come up with the following project scope:

- An invoice record sheet that contains information on invoices and whether or not they have been processed or are past due.
- A macro button that writes the invoice information into the record sheet.
- An invoice tracking number that automatically determines the new invoice number to be used and put into the record.
- A macro button to clear the invoice form.
- A macro button to save the invoice as a PDF to a designated folder and create a hyperlink in the records.
- A macro button to save the invoice as an xlsx file to a designated folder and create a hyperlink in the records.

With this agreement, we begin work on the outline of the invoice sheet.

Invoice No	Company	Amount	Date Issued	Date Due	Paid	Excel File	PDF File
1	Oliver Industries	\$1,043.83	5/27/2024	6/26/2024	Processed		
2	Robison Corp	\$5.99	2/27/2024	3/26/2024			
3	ABC Industries	\$11.99	5/27/2024	6/26/2024			

The sheet includes the invoice number from the invoice, the customer company, the total amount, and the related dates. We also have columns for the Excel and PDF files we will include later. The first five columns will be written in using a VBA macro we will write later, while the Paid column will be used to determine whether or not an invoice has been processed, and we can do this using a list where the only entry is "Processed."

Using conditional formatting on all possible rows of the table (=\$A\$2:\$H\$1048576), we can highlight past due rows in red that have yet to be processed using the following logic:

```
=AND($A2<>"", $E2<TODAY(), $F2<>"Processed")
```

Translation: If the invoice number in this row is not empty, and the date is past due, and the invoice is not processed, then apply formatting. With this, any row past due that isn't processed will be highlighted for us to grab our attention and notify us that we need to deal with the issue.

Our next step is to implement the code to create an invoice number tracker. Considering all of our invoices are kept here, it is safe to say that we can simply increment the number by one for each time we make a new invoice entry, using the previous maximum invoice number as a reference. We end up the following code:

Sub InvNumber()

```
Dim rng As Range
```

```
Dim maxInv As Long
```

```
' Check if there is any data in column A
```

```
If Sheet7.Range("A2") = "" Then
```

```
    ' If column A is empty, set the invoice number to 1
```

```
    maxInv = 1
```

```
Else
```

```
    ' Find the last filled cell in column A
```

```
    Set rng = Sheet7.Range("A2").End(xlDown)
```

```
    ' Find the maximum invoice number in column A
```

```
    maxInv = Application.WorksheetFunction.Max(Sheet7.Range("A2", rng))
```

```
    ' Increment the maximum invoice number by 1
```

```
    maxInv = maxInv + 1
```

```
End If
```

```
' Display the maximum invoice number in Sheet5 cell G4
```

```
Sheet5.Range("G4").Value = maxInv
```

End Sub

Note: End(xlDown) is effectively the Excel equivalent of using CTRL+SHIFT+Down

With this we have the invoice number tracker set up to be called in other macros. We'll then move on to our Insert Record Macro:

Sub InsertRecord()

```
Dim InvNum As Long
```

```
Dim custName As String
```

```
Dim amt As Currency
```

```
Dim dateIssued As Date
```

```
Dim dateDue As Date
```

```
Dim newRec As Range
```

```
If Sheet5.Range("G4") = "" Then      ' If the invoice number is blank, put in a new invoice number.
```

```
    InvNumber
```

```
End If
```

```
InvNum = Sheet5.Range("G4")
```

```
custName = Sheet5.Range("D5")
```

```
amt = Sheet5.Range("H45")
```

```
dateIssued = Sheet5.Range("G6")
```

```
dateDue = Sheet5.Range("G7")
```

```
Set newRec = Sheet7.Range("A1048576").End(xlUp).Offset(1, 0) ' Find next empty row.
```

```
newRec = InvNum          ' Insert invoice information
```

```
newRec.Offset(0, 1) = custName
```

```
newRec.Offset(0, 2) = amt
```

```
newRec.Offset(0, 3) = dateIssued
```

```
newRec.Offset(0, 4) = dateDue
```

```
Sheet7.Activate          ' Move to record sheet.
```

End Sub

This macro will capture all the important information from the invoice we are working on and write it in the next available blank row in our invoice record worksheet. In the case that we'd need to create an invoice by hand and not through the PO entry window, we've left that option open by accounting for times when the invoice number is blank. In this case, it will automatically generate the invoice number to put into the record, meaning invoices can still be manually inputted. After a record is saved, we'll automatically shift over to the record sheet to see the new record.

Next, we'll want to check out the Clear Invoice macro that was also included in our PO form macro:

Sub ClearInvoice()

```
Range("G4,D5,G5,G6").ClearContents ' Clear all non-table contents.
```

```
Dim line As Range
```

```
Set line = Sheet5.Range("C43") ' Clear the merged cell.
```

```
line = ""
```

```
Set line = Sheet5.Range("C11")
```

```
Do While line.Value <> "" ' While there are still entries ...
```

```
    line = "" ' Remove all products and quantities from the invoice.
```

```
    line.Offset(0, 2) = ""
```

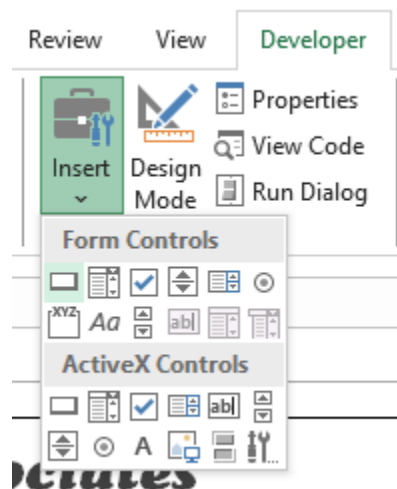
```
    Set line = line.Offset(1, 0)
```

```
Loop
```

End Sub

Again, very similar to what was created in our PO entry window to make sure that the invoice is completely cleared out.

We can then start creating buttons that we can tie our macros to. In the Excel ribbon we can enable the developer tab which will have a lot of the macro/VBA related functionality we'd be using. Click on Insert → Button.



From here we can select the area we'd want to include our button on. We can then adjust the name of our button and the text formatting. I included an icon in mine and group the button and image together for the purposes of design. Once we've selected an area, we can tie a macro to the button.

Assign Macro

Macro name:

InsertRecord

AutoFilter

AutoFilterPO

AutoFilterPOInvoice

'PERSONAL(AutoRecovered).xlsx'!ColumnsAndFilter

InsertRecord

Macros in: All Open Workbooks

Description

OK Cancel

We'll do this for both our Clear Invoice and Insert Record macros. Our window should look like the following at this point:

Robison Corp				Invoice	
FROM:		BILL TO:		Invoice No.	
Robison Corp		ABC Industries		PO No.	
5678 North Way		300 Oak Avenue		Date	
Phoenix, AZ 85048		Springfield, IL 62701		Due Date	
patrickrobison17@gmail.com		sarah.doe@abc.com			
Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)
Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	1	\$9.99	\$2.00	\$9.99
Customer Comments:				Subtotal USD	
Test				Tax	
				Total USD	
Company Information		Payment Information			
Company ID	2	Bank Name	Bank of America		
Phone No.	(234) 567-8901	Account Number	2345678901234		
Email	sarah.doe@abc.com	Notes			

Save Invoice +

Clear Invoice X

Note: Be sure to not place the buttons in line with any of the filtered-out rows of the product table or else it'll mess up or hide the buttons.

Finally we can get to work on the heart of our invoice tracking system which will be saving the files as records. We can do this for both xlsx and PDF files using the following macros:

Sub newExcelFile()

Dim InvNum As Long

Dim Customer As String

Dim Path As String

```

Dim FName As String
Dim recRange As Range
Dim foundRec As Range
InvNum = Range("G4")
Customer = Range("D5")
Path = "C:\Users\Patrick Robison\Documents\Business Documents\Invoices\" ' Set file path of the file
to be saved. Adjust as needed.
Fname = InvNum & "-" & Customer ' Use invoice number and customer for file
name. Ex: 4-Oliver Industries
Sheet5.Copy ' Copy the invoice sheet.
Dim shp As Shape
For Each shp In ActiveSheet.Shapes ' Remove ALL macro buttons from the new
sheet.
    shp.Delete
Next shp
With ActiveWorkbook ' Set sheet name and save the file as an xlsx file.
    .Sheets(1).Name = "Invoice"
    .SaveAs Filename:=Path & Fname, FileFormat:=51
    .Close
End With
Set recRange = Sheet7.Range("A:A") ' Search the invoice records for the
invoice number.
Set foundRec = recRange.Find(What:=InvNum, LookIn:=xlValues, LookAt:=xlWhole)

If foundRec Is Nothing Then ' If the invoice does not yet exist in the
records, insert it.
InsertRecord
Set foundRec = Sheet7.Range("A1048576").End(xlUp)
Sheet7.Hyperlinks.Add anchor:=foundRec.Offset(0, 6), Address:=Path & Fname & ".xlsx"
Else
Sheet7.Hyperlinks.Add anchor:=foundRec.Offset(0, 6), Address:=Path & Fname & ".xlsx" ' Add a
hyperlink to the saved file.
End If
Sheet7.Activate ' Go to the record sheet.
End Sub

```

The macro will save a copy of the invoice sheet to an address of our choosing; making sure to delete the macro buttons on the copy. It will then insert the record if it doesn't yet detect the invoice number on the invoice while also creating a hyperlink to the created file regardless. The same is also done for PDF files in the following macro:

Sub newPDFFile()

Dim InvNum As Long

Dim Customer As String

Dim Path As String

Dim FName As String

Dim recRange As Range

Dim foundRec As Range

InvNum = Range("G4")

Customer = Range("D5")

Path = "C:\Users\Patrick Robison\Documents\Business Documents\Invoices\" ' Set file path of the file to be saved. Adjust as needed.

FName = InvNum & "-" & Customer ' Use invoice number and customer for file name. Ex: 4-Oliver Industries

ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, ignoreprintareas:=False, Filename:=Path & FName ' Export the file as a pdf while keeping the print areas in mind.

Set recRange = Sheet7.Range("A:A") ' Search the invoice records for the invoice number.

Set foundRec = recRange.Find(What:=InvNum, LookIn:=xlValues, LookAt:=xlWhole)

If foundRec Is Nothing Then ' If the invoice does not yet exist in the records, insert it.

InsertRecord

Set foundRec = Sheet7.Range("A1048576").End(xlUp)

Sheet7.Hyperlinks.Add anchor:=foundRec.Offset(0, 7), Address:=Path & FName & ".pdf"

Else

Sheet7.Hyperlinks.Add anchor:=foundRec.Offset(0, 7), Address:=Path & FName & ".pdf" ' Add a hyperlink to the saved file.

End If

Sheet7.Activate ' Go to the record sheet.

End Sub

With this we've effectively finished our macros and can finalize the remaining buttons:

FROM:		BILL TO:		Invoice No.	
Robison Corp		ABC Industries		3	
5678 North Way		300 Oak Avenue		PO No. PO00002	
Phoenix, AZ 85048		Springfield, IL 62701		Date 5/27/2024	
patrickrobison17@gmail.com		sarah.doe@abc.com		Due Date 6/26/2024	

Product	Description	Quantity	Unit Price	VAT	Total (Tax exc.)
Oil Filter	Premium oil filter designed to trap harmful contaminants and keep your engine running smoothly.	1	\$9.99	\$2.00	\$9.99

Customer Comments:		Subtotal USD	
Test		\$9.99	
		Tax \$2.00	
		Total USD \$11.99	

Company Information		Payment Information	
Company ID	2	Bank Name	Bank of America
Phone No.	(234) 567-8901	Account Number	2345678901234
Email	sarah.doe@abc.com	Notes	

Results

After our work we accomplished the following:

- Consolidated several databases of over 100 rows of information into one file; using VLOOKUP formulas to fill in customer and product information.
- Developed 7 VBA macros to perform several functions at the click of a button; including saving files.
- Automated a data entry position; reducing the time to process invoices by around 83% and streamlining the process for new hires.

Thank you for sticking around to the end of this walkthrough. I hope you found the contents interesting, and I hope that you may have learned something new.

Every day is a good day to be better than you were yesterday.