Enhancing Procurement Systems at San Antonio International Airport through Artificial Intelligence Integration

Anthony Sadler

Engineering Science: Environment

asadler1@mail.stmarytx.edu

Sebastian Blackburn

Mechanical Engineering

sblackburn@mail.stmarytx.edu

Project Sponsor:
San Antonio International Airport
Mr. Syed Mehdi
Chief Strategy & Innovation Officer
Adjunct Professor of Engineering at St. Mary's University
Member, St. Mary's School of Science, Engineering, and Technology Advisory Council

A SENIOR DESIGN PROJECT REPORT

Submitted to the Faculty of the St. Mary's University in partial fulfillment of the requirements for their respective Bachelor of Science degrees.

SAN ANTONIO, TEXAS 2024

Table of Contents

Project Charter

GENERAL PROJECT INFORMATION	
PROJECT NAME	Enhancing Procurement Systems at San Antonio International Airport through Artificial Intelligence Integration
Project Champion	Mr. Syed Mehdi
PROJECT SPONSOR	San Antonio International Airport
PROJECT MANAGER	Sebastian Blackburn and Anthony Sadler
Start Date	August 2023
COMPLETION DATE	April 2024

PROJECT DETAILS	
EXECUTIVE SUMMARY	The San Antonio International Airport identified inefficiencies in its procurement process and sought improvements. This project aims to streamline procurement operations through AI automation, specifically in the search for stock keeping units (SKUs) or part numbers. Limited to the airport's procurement office, the project aligns with security and privacy objectives. The solution involves converting and cleaning procurement contracts to make them machine-readable for AI analysis, leveraging existing programs and Excel macros. Key milestones were achieved within the project timeline, resulting in deliverables including methods for PDF-to-XLS conversion, contract cleaning, AI-based querying, and an owner/user manual. Overall, the project provides a scalable and sustainable solution to enhance

	procurement efficiency at the San Antonio International Airport.
OBJECTIVES	To expedite the procurement process at the San Antonio International Airport by developing and implementing an AI-driven solution that automates the search for stock keeping units (SKUs) or part numbers, thereby reducing the time and resources required for procurement, while ensuring alignment with security and privacy objectives.
EXPECTED BENEFITS	- Increased efficiency: By automating the search for stock keeping units (SKUs) or part numbers, the procurement process will be expedited, saving valuable time and resources for the San Antonio International Airport. - Cost savings: Streamlining procurement operations will lead to cost savings by reducing manual labor and improving resource allocation. - Enhanced accuracy: Utilizing AI technology will improve the accuracy and precision of procurement searches, minimizing errors and ensuring procurement of the correct items. - Improved productivity: With a more efficient procurement process, airport personnel can focus on higher-value tasks, leading to improved overall productivity. - Scalability: The AI-driven solution can be scaled and adapted to meet future procurement needs, providing long-term benefits for the airport's operations.
SCOPE	The scope of this project was exclusively limited to the San Antonio International Airport's Procurement Office. This focus was decided upon so that the implementation of the solution remained aligned with the San Antonio International Airport's security and privacy objectives.

MILESTONES	1. Project proposal final draft, completed on 9/22/23. 2. Database Creation begins: 9/6/23 completed on 02/15/24. 3. Data stratifying/ grouping begins: 10/1/23 completed on 12/31/23. 4. GPT integration testing begins: 10/8/23 completed on 02/1/24. 5. Final presentation completed on 04/30/24.
SUCCESS METRICS	1. Reduction in procurement time: Measure the percentage decrease in the time taken to find stock keeping units (SKUs) or part numbers compared to the legacy manual process. 2. Cost savings: Quantify the cost savings achieved by reducing manual labor and improving resource allocation in the procurement process. 3. Productivity enhancement: Measure the increase in productivity of procurement personnel by tracking the completion time of procurement tasks and the ability to focus on higher-value activities.

Abstract

Our study delved into the utilization of artificial intelligence, specifically leveraging Chat GPT 4.0, to address the inefficiencies hindering the Procurement Department's operations at San Antonio International Airport. Recognizing the challenges of the arduous task of manually sifting through vast catalogs to find specific items, we endeavored to implement AI solutions to streamline the procurement process. By integrating Chat GPT 4.0, alongside other AI tools, our objective was to revolutionize the cumbersome part selection process within the Procurement Office, ultimately aiming to expedite the identification of required items while minimizing human intervention.

Our investigation focused on the development and implementation of AI-driven systems capable of navigating through the complexities of the procurement catalogs that belong to the San Antonio International Airport with efficient precision. Through rigorous testing and experimentation, we evaluated the performance of these AI tools in automating the search and identification of items, ensuring compatibility with the unique requirements and constraints of the airport's procurement environment.

The results of our study demonstrate promising advancements in efficiency and effectiveness, significantly reducing the time and effort required to find the suppliers for essential items of the procurement process. The integration of AI technologies has not only expedited the procurement process but has also enhanced the accuracy of item selection, mitigating the risk of errors and discrepancies. Overall, our findings underscore the transformative potential of AI in modernizing procurement practices within airport operations.

By harnessing the power of artificial intelligence, we have laid the groundwork for a more agile, responsive, and efficient procurement system by delivering to San Antonio International Airport a new method of finding part numbers through the use of Artificial Intelligence and an Owner and Operations Manual, which can be found in Appendix A, and allows the Procurement team to be ready to meet the evolving demands and challenges of the aviation industry.

1. Problem Description

Introduction

The San Antonio International Airport serviced 10.6 million people (about five times the population of the city of San Antonio) in 2023 and provided economic, safe, and reliable transportation for business, leisure, and cargo alike (Chief Strategy and Innovation officer at the San Antonio International Airport Prof. Mehdi). The San Antonio International Airport, like any organization, has many factors to consider when ensuring that the goals of safe and reliable travel were met, which made the detection of productivity improvement opportunities an important initiative at the time this project was offered to St. Mary's University. The Procurement team at the San Antonio International Airport proposed a solution to make their jobs more efficient and, if addressed properly, save valuable time and resources.

Traditionally, the San Antonio International Airport's procurement process had been labor-intensive and consumed a significant amount of time, all while having relied upon numerous human intermediaries for the procurement process to function. However, with the development of artificial intelligence, there was great potential and opportunity to streamline the

procurement process through the automation of these time consuming and labor heavy processes. By making the procurement contracts AI readable, the Senior design team used an AI chatbot to make the process of finding stock keeping units (SKUs), or part numbers, easier to find. The motivation behind this project was not to remove the human element of the job, but rather to make the job of the Airport's procurement team easier, such that time and money would be saved from the increased efficiency of the procurement division of the San Antonio International Airport.

Literature Review

OfficeSpace Software (2023) estimated 91% of businesses reported having a company-wide strategy of using and implementing AI systems into their business. The relevance to the project was in the proposed benefits of integrating AI. The Society for Human Resource Management (SHRM, 2023) described possible ways of implementing AI in the workplace. Slalom Build (2023) detailed what a generative AI is, how it worked, and methods of utilizing and crafting them. OpenAI (2023) detailed how it handled the data it is given during the conversations that the system is known for. This information was relevant to our project due to implied concerns from the airport over the security of Chat GPT 4.0, and so the team took the relevant information and passed it along to the airport. PBS (2023) described how AI made images from text prompts. While interesting, it was not immediately relevant to the project, since the team and the airport had agreed that images were not going to be used in the final product, at least in its initial phase. This information, however, pointed at possible extensions to the present project.

Problem Statement

We use an example to describe some of the inefficiencies inherent in the use of Vroozi, the contract aggregation platform used by Airport personnel before this project started. Assume that the Procurement team searched for a 2022 Ford F150 air filter. Not only did Vroozi produce many filters for Ford vehicles but also for various other makes and models that would be vaguely related to Ford. F150, 2022, air, or filter would also bring results with those key words in their name that were irrelevant to the search. Even window tint would appear because the keyword “filtering” was in the description to specify how the tint interacted with light. While a human user would know that window tint was to be immediately disregarded it was harder for the computerized catalog to recognize such a distinction. Hundreds and even thousands of unrelated items may appear when Vroozi is queried on a variety of keywords.

The problem that the Airport described to the Senior Design team at the onset of this senior project was the inefficient way suppliers were found by Vroozi for the items that the airport needed to procure, due to the lack of an optimized process at the time. All items in Vroozi had a unique SKU that could be searched for in the various catalogs it contained, but getting a SKU for what was needed consumed a considerable amount of time. The legacy process had relied heavily on human communication and coordination, where the procurement office sifted through long and numerous pages of items on Vroozi, full of thousands of results, to find one specific item that was to be procured by the team.

Goal

The goal of this senior design project was to expedite the time it took to find a SKU for items that would need to be procured at the Airport. The team understood that contracts were aggregated into Vroozi, where an Airport employee would then look up the items needed. However, this was accomplished by manually searching for specific part numbers, a process where a human would have to comb through the hundreds to thousands of results Vroozi displayed to find the correct part. The proposed solution was to automate the lengthy searching process by delegating the task to an AI, which would be able to methodically search through an entire mock database that would be made by the Senior design, to then determine an optimal option and present these few and specific options to a staff member with a SKU for each one. This SKU would then be searched in Vroozi, making items found through their part numbers quicker than searching based on their description.

Scope

The scope of this project was exclusively limited to the San Antonio International Airport's Procurement Office. This focus was decided upon so that the implementation of the solution remained aligned with the San Antonio International Airport's security and privacy objectives.

One of the constraints we faced was that the AI system was to be operated within a controlled environment, where data were accessed and manipulated through a database made

locally on the Procurement team's computer. There would not be access from the database to external data sources, systems, or networks for the Senior design project.

The Senior design team believed that Vroozi's search algorithms was not only a problem for the Airport's Procurement team, but for most legacy procurement systems that used Vroozi as their main procurement contract aggregator, as it was one of the most widely used procurement aggregation catalogs during the development of the project solution. While the problem the Senior design project had addressed was procurement inefficiency within the Airport's Procurement team, it is recognized that, on a larger scale, the problem lied with finding specific items efficiently in Vroozi.

Requirements and Constraints

Professor Mehdi, the project's sponsor, described the requirements and constraints of the project as follows:

- Development of an AI tool that could query a static database and identify the SKU being searched;
- A user manual such that contracts could be added, modified, and removed from, and
- A chatbot could be used to find SKUs.

If time allowed, the development of an AI interface that would be contained in a widget (a program that performs one task and is small in size that is active on the desktop of a computer, like the clock function or the search bar) would be added, allowing the underlying program to query a static database and return the cost of items on a per-unit basis, along with the availability of items based on the contracts given, and the lead time for delivery. After discussion during

our initial meetings, the team and our senior project sponsor agreed on and created the following constraint: The Airport needed to be able to take complete control of the team's solution when the project is finished, which meant that the Senior Design team needed to purely use company resources, equipment, and standards in having access to software. Therefore, the senior design's sponsor, Mr. Syed Mehdi, requested that Excel and its addons such as Visual Basic Applications, Chat GPT 4.0, and certain free-to-use online resources were to be used in the development of the team's solution.

Metrics

The project would be deemed successful if four objectives were completed:

- The AI chatbot could query a contract;
- The AI chatbot could articulate the information requested in a fashion easily understandable by a human user;
- The time to find SKUs is shorter than the time required by the legacy method; and
- A user manual is provided to the San Antonio International Airport's and has clear instructions and guidance that allows the Procurement Team to update the database, or the AI used to analyze the database's information.

Project Strategy

The first iteration of the project proposal and the corresponding PowerPoint presentation were submitted to the Airport's Procurement team after the approval of the project sponsor, Chief Innovation Officer at the San Antonio International Airport: Mr. Syed Mehdi, and St.

Mary's University project advisor, Dr. Moras. As depicted in the project Gantt chart that can be found in Appendix B, project work commenced immediately upon approval. For the next stage, the team researched how artificial intelligence could be used to make the process of finding SKUs quicker. One of the first ideas the Senior design team investigated was to train an AI tool to predict what someone was trying to find during a search and provide a SKU to that user when, for example, a Ford F-150 air filter is being searched. This would imply having an AI tool that can analyze either a series of documents or a super document (a document consisting of multiple contracts) to then analyze and display the results of a search for a particular part or item, the example used in our testing became a Ford F-150 Air Filter.

A trained AI tool was deemed too complex and impractical for this project, considering the time constraints and the development cycle of a completely new and unique AI tool. After deliberation and negotiation between the parties tied to the project, the Senior Design team and the Procurement team jointly developed a proposal that consisted of developing a method to use a series of cheap and effective tools to complete the task of improving the efficiency of the procurement department. This resulted in the creation of website-based methodology that would use existing programs to conduct a series of smaller tasks that would yield the desired results of, when given to Chat GPT 4.0 for analysis, giving a user all the relevant information of a part, including the part number or SKU, the supplier that the part is contracted from, and the pricing of a unit of the part (either by batch or by individual) that was found in an Excel database consisting of converted and cleaned (reformatted) Excel files of the original PDF contracts.

The project goal was accomplished by initially devising a method to render the procurement files machine-readable. Two methods of querying the procurement contracts that the Airport uses were initially considered. The first was where the senior design team had tried to export the Portable Document Format (PDF) version to Chat GPT 4.0 directly, with the second being that the PDF files would be converted to Microsoft Excel Spreadsheet (XLS) files before making the files more uniform using Excel's internal processes. Through experimentation, it was concluded that the second method for conversion allowed for the file to be accurately read by Chat GPT 4.0 after testing both methods, where the trouble of reading inconsistently formatted PDFs caused Chat GPT 4.0 to not even be able to analyze the files as they were.

To follow the second method, the contract files were cleaned, or made uniform, so that the user would then query the database via Chat GPT 4.0, which was used during the project to test for accuracy and reliability. For example, the question of "If pricing is in column C, what rows contain parts that fit the description of Air Filter and what are their pricing and supplier," would be one of many test questions the Senior design team had used during development. The AI system used by Chat GPT 4.0 would analyze the data set made by our team and would respond with the information found in the cleaned contract files, with the example proposed getting a response that contains one or more suppliers that satisfied the query, along with the price of the parts and where they could be found in the Excel file. Once the ideal product is determined by the human operator, further questions could be presented to Chat GPT 4.0 that would inform personnel, which would then be used when tailoring a search on Vroozi, including stratifying by supplier.

A challenge that became apparent during the initial development was the "contract conversion and cleaning" problem, which was that to make a contract machine-readable, there needed to be an estimated one hour of manual work to reformat the file by hand. To clean a contract, the reformatting would involve removing unnecessary information and to re-align what remains, creating a standard file layout that would be uniform and consistent with all the files that would be used in the contract database the Senior Design team creates. With 68 contracts estimated to be used, the team calculated that approximately 68 hours would be needed to clean the contracts if the team were to do it by hand. The 68 hours of estimated work would be considered problematic for the Procurement team if they oversaw cleaning up the contracts, as they would need to work around their busy schedules to do all this additional work. It would also be impractical for the Senior Design team due to similar problems of tight scheduling as well as the standards for what should constitute a Senior Design project. Recognizing the amount of time that was estimated to clean the 68 contracts by the senior design team, the procurement team at the Airport reduced the number of contracts to be cleaned and used by the senior design team to 12 contracts.

The companies on the three contracts received by this Senior Project team were Automotive Trucks and Springs, Navistar and Crane Carrier, and Herc Rentals Inc. The Senior design team explored how either Chat GPT 4.0 or a Python program could be leveraged to clean the contracts, such that clean contracts were not constructed by manual human labor. We discovered that Chat GPT 4.0 would not be able to clean the contracts as the team needed on its own, as, at that time, GPT 4.0 did not have the capability of restructuring a contract to fit our needs or subsequently recreate the file. In testing the capabilities of GPT 4.0, we realized

that the AI chatbot could be better used to query a clean contract rather than make clean contracts.

The team also explored how Python could be used to clean the contracts, and with the aid of Chat GPT 4.0, the Senior Design team created multiple Python scripts that were tested to evaluate how clean the scripts could get the contracts. While the scripts did clean the contracts, the "work to clean" ratio, or the time and effort put into the program compared to the time saved by the program, was not where the team believed it should be. Neither member of the senior design team was proficient enough in Python to make this ratio ideal in the allotted time granted to the project.

The team explored whether the macro capabilities of would be a more suitable option to clean the contracts with. Macros are subroutines or small programs that are housed in Excel and other Microsoft products that can perform repetitive tasks for a user, with a wide range of coding possibilities, all at the push of a couple buttons. We developed a series of Macros to support and automate the reformatting of the contracts, which then allowed Chat GPT 4.0 to read what the team had given to it to analyze. Once the Macros were developed, finishing touches would be made and the final segment of the project began.

The team created a shortcut in Excel, which is a button that would activate the macro and would remove the need to have the macros on hand, leaving the manual the last thing to create. This took the form of saving the macros directly to the Excel worksheet, making them easily accessed by a user. It's used is fully described in the Owner and Operations Manual included in Appendix B. This manual was created in Word and provides instructions, images, and guidance on basic troubleshooting that may need to be done in unforeseen scenarios, as well as

an easy-to-read instruction set to use the solution the senior design team made. A report and proposal presentation were developed for this project as well, crafted to fit the information in two separate ways: the technical manner found in the report, and a quickly understood and surface level manner found in the presentation.

Tools needed for the project were mostly software-based technologies and applications, such as, but not limited to, Adobe Acrobat, PDFtoEXCEL.com, Google Sheets, Excel, Chat GPT 4.0, email, Microsoft Teams, and Lang Chain. The team's personal computers were the means to access the software, which would emulate the most stringent potential operational use of the project solution. Furthermore, our team divided the tasks found in the Gantt Chart (Appendix B), where tasks were assigned to the two members of the Senior design team separately. However, since our team was composed of two people, there were sections that were charged to both members at once, including the report and presentations.

Communication between the Senior design team was facilitated via Microsoft Teams, where the Senior Design team would have their virtual meetings with the Airport personnel. The team logged a journal to record all progress made on the project, and when the final report was being written, those logs were transferred to the report and kept up to date. The journal entries were submitted to Dr. Moras for review. Conflict was sparse and in-between but when it did arise the St. Mary's Ombuds, Dr. Bob O' Connor, was contacted to facilitate solutions and further the team's understanding of each individual member. We achieved positive results throughout these meetings with Dr. Bob, resulting in growth in both members of the team.

Listed next are the key milestones detailed in the *Project Plan* section and are also displayed in the Gantt Chart for reference to the other various parts of the project. These milestones would

be reached throughout the academic year, culminating in the final presentation that was done on April 30th:

1. Project proposal final draft, completed on 9/22/23.
2. Database Creation begins: 9/6/23 completed on 02/15/24.
3. Data stratifying/ grouping begins: 10/1/23 completed on 12/31/23.
4. GPT integration testing begins: 10/8/23 completed on 02/1/24.
5. Final presentation completed on 04/30/24.

Deliverables

Our sponsor specified four deliverables for this project. Firstly, there was a method to convert PDF files to XLS files. This was important because the procurement team obtained their contracts in PDF not XLS. An XLS file contract was important to our solution as it was easier to query and control an XLS file than a PDF.

The second deliverable was a method to "clean" the newly converted XLS files to the standard format used for the database. A contract was determined to be 'clean' when the Excel file was uniform to the rest of the database the Senior Design team created and only has the relevant information that Chat GPT 4.0 needs for analysis. For example, if an XLS file of the contract contained the date that the contract was sent to the airport in a cell of the file, this would need to be removed. The removal of the date is not something that would aid Chat GPT 4.0's analysis of the Senior Design team's database, but rather hindered and stopped the analysis of the file due to Chat GPT 4.0 being unable to work abstractly. While based on intuition, a human operator could consider the data simply irrelevant and move past the cell,

Chat GPT would not be able to make the same distinctions that would be necessary to properly parse out the data and would be unable to continue.

The third deliverable was a specific way for the procurement team to query the AI solution. The more specific the language used in the query, the more consistent and predictable the results of the query will be. The other deliverable was an owner and user manual such that the procurement team at the airport can easily use and troubleshoot our solution. This became the Owner and Operations Manual found in Appendix A.

2. Solution Approach

Introduction

In addressing the complex challenges posed by the project, this section serves as the record of the Senior Design team's endeavor, which was aimed at delivering an effective solution to the Airport. We outline the methodology used for the project, elucidating the rationale behind the decisions made and the systematic steps undertaken to engineer practical solutions. From conceptualization to implementation, this approach embodies a fusion of creativity, technical prowess, and strategic thinking, tailored to meet the unique demands of this project. By dissecting the intricacies of the solution's framework, the team aims to provide a comprehensive insight into the problem-solving journey that was made, highlighting the ingenuity and efficacy of our methodologies.

Methods

As stated in the 'Deliverables' section, the senior design team delivered four items. To reiterate, the first deliverable was a method to convert PDF files to XLS files. Secondly, a method to clean contracts where "clean" refers to a newly converted XLS contract being uniform with only the information that is useful to find the vendor, or supplier, for a specific item to then be procured. We produced specific language to query the contracts using an AI chatbot; specifically structured and formatted language allows for consistent and precise results when using a Large Language Model (LLM), which can interpret slightly different phrases in wildly different ways. Finally, we created user's manual for the Procurement team at San Antonio International Airport to easily use and troubleshoot the solution the Senior Design team has made.

Converting PDF files to XLS files

The San Antonio International Airport gets procurement contracts in PDF format. It is important that these contracts are converted to XLS files to allow for the answers to queries, via Chat GPT 4.0, to be consistent. We converted the PDF files to XLS files by using the free online tool, PDFtoEXCEL.com, which allows for any PDF file to be converted to a XLS file. PDFtoEXCEL.com does a good job of converting the files but it does not convert without flaws. This is the senior design team made a method to "clean the contracts."

Cleaning Contracts

The clean titling of the files refers to them being formatted in a uniform way such that it is easier for Chat GPT 4.0 to look through the contracts. It is also important for the contracts to be "clean" so that the verbiage that is used by the Procurement team to query for the information within the contracts is consistent. An example of a "dirty" contract is furnished in Figure 1. A dirty contract is defined as a contract that did make the conversion process into a XLS file, and still contains unnecessary in addition to the Vendor, Part Number and List Price. A XLS that only has the Vendor, Part Number and List price is referred to, subsequently, as a "clean contract."

To 'clean' the contract, macros were used in Excel to clean the procurement contracts converted from PDF to XLS format. A macro is a set of instructions or commands that automate tasks within the spreadsheet software that is found in many Microsoft Office products, including Word and Excel, and are crafted using the program's Visual Basic for Applications Studio (or VBAS). VBAS uses its own programming language, VBA (Visual Basic for Applications, to write the code for the macros. The purpose of these macros was to standardize the format of the contracts, ensuring consistency and facilitating the querying process for Chat GPT 4.0. By applying macros, the team automated the cleaning process, making the process more efficient and dramatically reducing the number of man-hours required.

	A	B	C	D	E	F	G	H	I	J
34	b	Product Identification (manufacturer)		CLUB-CAR						
35	c	Type of Price Schedule (dealer, jobber, etc.)		RETAIL						
36	d	Price Schedule Number		124-136						
37	e	Date of Price Schedule		1/19						
38	f	Price Schedule Column on Which Discount		LIST						
39		is based (i.e. distributer, net, wholesale, etc.)								
40	g	Warranty		NO WARRANTY						
41	Item 3B	Specified Items for evaluation purposes only								
42	Item 4	Labor and Service								
43		Labor - Regular Rate (Max 11 hrs. labor								
44		allowable for bench work)		$90.00						
45		Labor - Expedited Rate		$90.00						
46		Payment Terms		Net 30						
47										
48	**Solicitation: Informal IFB**									
49										
50	**Departments: Parks & Recreation; Solid Waste Management; Building and Equipment Services; Aviation**									
51	**Scope: Provide parts and service to approximately 64 carts and utility vehicles utilized within the City's**									
52	fleet.									
53										
54	**Estimated Usage: $10,000.00 per year; the contract period will include 2, 1-year renewal options; not to**									
55	exceed $50,000.									
56										
57	Page 1									
58										
59	EZGO Pricelist 2019									
60		Report #: 124			Customer Price List As of --->			11/18/19		
61	PAGE:	1								
62	Store #:	1		For: CITY OF SAN ANTONIO						
63				LINE 130 E Z GO PARTS						
64		Part Number	Description			SPQ SET UOM CL		LIST	NET	
65		CORE AMOUNT								
66	00015G1		TOW BAR		1	0	EA			
67	00100G4			SCREW 10 16X.5 FOR K	1	0	EA	0.82	0.62	
68	00106G6			SCREW 12 24X3/4 CP	1	0	EA	0.82	0.62	
69	00114G6			SCREW WASHER HEAD 1/4	1	0	EA	0.83	0.63	
70	00114G8		SCREW NLA		1	0	EA	0.93	0.70	
71	00130G5		NLA	10 13 X 5/8 ZP	1	0	EA	0.36	0.27	
72	00139G2			6 32X1 1/4 SC M RS CP	1	0	EA	0.11	0.09	
73	00139G6			SCREW 6 32X1 3/4 M R	1	0	EA	0.85	0.64	
74	00156G4			NLA 10 32X1/2 SC M RS	1	0	EA	0.10	0.08	
75	00205G2		USE 00205G4		1	0	EA	0.51	0.39	
76	00205G4			SC M FP ZP 1/4 20 X 1	1	0	EA	1.58	1.19	
77	00276G6			SCREW 5/16 24 X 3/4	1	0	EA	1.71	1.29	
78	003-70080				1	0	EA			
79	00406G4			SCREW HEX 1/2 13 X 4	1	0	EA	7.97	5.98	
80	00407G6			SCREW 1/2 13 X 6	1	0	EA	5.58	4.19	
81	00414G6			SCREW 1/4 20X3/4 HEX	1	0	EA	1.68	1.26	
82	00414G8			OBS SCREW 1/4 20X1 C	1	0	EA	0.27	0.21	
83	00415G2			SCREW 1/4 20 X 1 1/4	1	0	EA	1.69	1.27	
84	00426G6			SCREW, CAP HEX HEAD 3	1	0	EA	3.18	2.39	
85	00426G8			SCREW 3/8 16X1 CAP HE	1	0	EA	3.18	2.39	
86	00427G6			SCREW CH 3/8 16 X 1 3	1	0	EA	4.04	3.03	
87	00427G8			SCREW HEX 3/8 16X2 CP	1	0	EA	4.78	3.59	
88	00428G2			SCREW 3/8 16X2 1/4 C	1	0	EA	3.20	2.40	
89	00428G4			SCREW HEX 3/8 16 X 2.	1	0	EA	2.41	1.81	
90	00428G6			SCREW 3/8 16X2 3/4 HE	1	0	EA	4.00	3.00	

Figure 1. Dirty EZGO contract after it has been converted from PDF to XLS using PDFtoEXCEL.com.

For example, the Senior design team created a macro (Figure 2) that deletes all columns after the fourth column if there are four consecutive columns, starting at column 1. In Figure 3 displayed is the revised contract first seen in Figure 1, after being cleaned by the macro.

```
(General)                                   (Declarations)

Sub DeleteRowsIfNotAllColumnsOccupied()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim i As Long
    Dim occupiedColumnsCount As Integer

    ' Change "Sheet1" to your actual sheet name
    Set ws = ThisWorkbook.Sheets("Test")

    Application.ScreenUpdating = False ' Turn off screen updating for better performance

    ' Find the last row with data in column 1
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Loop through rows backwards to safely delete rows
    For i = lastRow To 1 Step -1
        ' Reset the count of occupied columns for each row
        occupiedColumnsCount = 0

        ' Count the number of occupied columns (1, 2, 3, and 4)
        If ws.Cells(i, 1).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 2).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 3).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 4).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1

        ' Delete the row if not all 4 columns are occupied
        If occupiedColumnsCount < 4 Then
            ws.Rows(i).Delete
        End If
    Next i

    ' Align all columns to the left
    For j = 1 To 4 ' Assuming you want to align the first 4 columns
        ws.Columns(j).HorizontalAlignment = xlLeft
    Next j

    Application.ScreenUpdating = True ' Turn screen updating back on

    MsgBox "Rows with less than 4 occupied columns have been deleted.", vbInformation
```

Figure 2. Macro code where all columns after fourth are deleted if there are four consecutive columns of information.

The process of using PDFtoEXCEL.com to convert the contracts and then using up to five separate macros to clean the contracts was conducted 12 times for all the contracts that were provided to the Senior Design team by Mr. Jordan Mergele of the Procurement team at the San Antonio International Airport.

Listed next are technical descriptions of the purpose of the four most used macros do. At the end of the last step for a given macro there will be a couple of sentences describing its

purpose, in plain English. The code for all the macros is included in the User Manual (Appendix A).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Part Number	List Price	Vendor													
2	00100G4	0.82	EZGO													
3	00106G6	0.82	EZGO													
4	00114G6	0.83	EZGO													
5	00114G8	0.93	EZGO													
6	00130G5	0.36	EZGO													
7	00139G2	0.11	EZGO													
8	00139G6	0.85	EZGO													
9	00156G4	0.10	EZGO													
10	00205G2	0.51	EZGO													
11	00205G4	1.58	EZGO													
12	00276G6	1.71	EZGO													
13	00406G4	7.97	EZGO													
14	00407G6	5.58	EZGO													
15	00414G6	1.68	EZGO													
16	00414G8	0.27	EZGO													
17	00415G2	1.69	EZGO													
18	00426G6	3.18	EZGO													
19	00426G8	3.18	EZGO													
20	00427G6	4.04	EZGO													
21	00427G8	4.78	EZGO													
22	00428G2	3.20	EZGO													
23	00428G4	2.41	EZGO													
24	00428G6	4.00	EZGO													
25	00429G2	4.04	EZGO													
26	00429G4	2.21	EZGO													
27	00433G1	2.55	EZGO													
28	00438G6	1.68	EZGO													
29	00438G8	1.68	EZGO													
30	00439G1	2.40	EZGO													
31	00440G2	2.35	EZGO													
32	00440G6	1.69	EZGO													
33	00441G2	4.01	EZGO													
34	00441G8	5.88	EZGO													
35	00444G8	1.72	EZGO													
36	00462G7	2.41	EZGO													
37	00463G11	3.95	EZGO													
38	00463G4	3.15	EZGO													

Figure 3. Clean EZGO contract after it has been cleaned using macros.

1. Sub KeepSelectedColumns()

This VBA macro retains only the selected columns in the active worksheet and deletes all other columns. Here is a simplified explanation of its steps:

1. It prompts the user to select columns using an input box.
2. Error handling is implemented to handle cases where the user cancels the selection.
3. If no columns are selected, a message box informs the user and the macro exits.

4. It iterates through each column in the worksheet from right to left.
5. For each column, it checks if the column is not included in the selected columns range.
6. If the column is not part of the selected columns range, it deletes the column.

This macro efficiently filters columns based on user selection, simplifying data management tasks within the worksheet.

2. Sub DeleteRowsWithEmptyCells()

This VBA macro deletes rows from a worksheet where either of two specified columns contains empty cells. Here is a detailed explanation of its functionality:

1. The macro begins by declaring variables for the worksheet, ranges for two selected columns, individual cells within these columns, the last row number, and a loop counter.
2. It prompts the user to select two columns from the worksheet using an input box.
3. Error handling is implemented to handle cases where the user cancels or does not select any column, displaying an error message and exiting the macro accordingly.
4. The worksheet containing the selected range of the first column is set.
5. Screen updating is turned off to improve performance during the deletion process.
6. The last row with data based on the first selected column is determined.
7. It iterates through each row starting from the last row and moving upwards.
8. For each row, it checks if either of the cells in the corresponding columns is empty using the CountA function.
9. If either cell is empty, it deletes the entire row using the Delete method.

10. After completing the loop, screen updating is turned back on.
11. A message box is displayed to inform the user that rows with empty cells in the selected columns have been deleted.

This macro provides a convenient way to clean up data by removing rows with empty cells in specific columns, enhancing data integrity and readability.

3. Sub DeleteRowsWithSpecificText()

This macro deletes rows from the active sheet where the first column contains a specific text. Here is a concise breakdown of its functionality:

1. The macro sets the active sheet as the worksheet to work with (`ws`).
2. It defines the text to search for (`searchText`), which is "CAT-CLASS DESCRIPTION" in this case.
3. Screen updating is turned off to enhance performance.
4. The last row with data in the first column is determined.
5. It iterates through rows backwards to safely delete rows.
6. For each row, it checks if the value in the first column matches the specified text (`searchText`).
7. If the text is found in the first column of a row, the entire row is deleted.
8. After completing the loop, screen updating is turned back on.
9. A message box is displayed, informing the user that rows containing the specified text in the first column have been deleted.

This macro efficiently removes rows based on a specific criterion, enhancing data management within the worksheet.

4. Sub RemoveEmptyRows()

This VBA macro removes empty rows from a worksheet based on the content of two selected columns. Here is a breakdown of its steps:

1. The macro prompts the user to select two columns using an input box.
2. Error handling is implemented to handle cases where the user cancels the selection.
3. If no columns are selected, the macro exits.
4. Screen updating is turned off to enhance performance.
5. It loops through each row in the selected columns.
6. For each row, it selects the corresponding cells in the two columns.
7. It checks if either of the cells in the current row is empty.
8. If either cell is empty, the entire row is deleted.
9. After completing the loop, screen updating is turned back on.

This macro efficiently cleans up data by removing rows where either of the selected cells in each row is empty, thereby streamlining data analysis and presentation.

Language Used

What is the list price for part number ABC123 located in column B? Or if you need the price from a specific vendor: “What is the list price for part number ABC123 from Vendor X located in column B?" This way, you are explicitly stating where the relevant information can be found in

the file, ensuring that the response accurately reflects the data structure with which you are working.

Operations and Owner's Manual

The language used to query Chat GPT 4.0 for a vendor, price and or part number needs to be very deliberate and specific. We explain the language to be used in the Operations and Owner's manual located in Appendix A. Listed is a "cookie cutter" method of asking Chat GPT for what is needed.

What is the list price for part number ABC123 located in column B? Or if the user needs the price from a specific vendor: What is the list price for part number ABC123 from Vendor X located in column B?"

This way, the user is explicitly stating where the relevant information can be found in the file, ensuring that the response accurately reflects the data structure with which you are working.

Data Collection

Months after the initial three contracts were sent to the senior design team members of the Airport's Procurement team sent nine more contracts to work with via email, totaling twelve. The PDF versions of the contracts were downloaded, and the team tried Adobe Acrobat and PDFtoExcel.com to convert the original PDF file to an Excel file. Many Excel cells were double units after the conversion from both applications, where two sets of information were formatted into a single cell. Adobe Acrobat was not a reliable candidate to use because the XLS files it creates were not consistently legible by Chat GPT, on top of the subscription a user would need to even use the exporting function that converted the files. Instead, PDFtoExcel.com became

the program used due to less problems in the conversion and that it was free online software. To ensure uniformity in the resulting Excel files, where a double cell was found, the Senior Design team, as mentioned in previous section *Project Strategy*, unmerged all relevant cells and would reorganize the data, first by hand before the implementation of the macros.

The initial data the Senior Design team collected for this estimate was specific to one contract out of the three that the Airport gave to the team initially. Each of the three contracts was approximately 100 pages.

Evaluation of AI Tools

The team researched and developed a solution and encountered simpler solutions to complex problems. For example, the leading idea to make contracts AI readable was to use Lang Chain, a software neither of the team members were familiar with, but that was presented as an option since the team could then develop a unique AI designed with the project in mind. Anthony scheduled a meeting with Dr. Damiano Torre, an Assistant Professor of Engineering at St. Mary's University, to discuss the use of Lang Chain. With the professor's guidance, we found that Chat GPT 4.0 was a more appropriate and straightforward way to "feed" the contracts to AI. Furthermore, considering the skills of each of the team member Lang Chain appeared to be a potentially overwhelming task considering the time constraint of two semesters.

Development of Macros

Before diving into the step-by-step process, it is important to note that there are two primary methods for creating macros in Excel: (1) writing VBA (Visual Basic for Applications) code (2) using the Record feature. Each method has advantages and may be suitable depending on the analyst's ability and familiarity with the software.

In the next section, we outline both approaches to better encapsulate the possible methods of creating macros for repetition of this project in the future.

A. Using VBA to Create Macros

1. Open Excel: Launch Microsoft Excel and navigate to the Developer tab as outlined in the previous section.
2. Open Visual Basic for Applications (VBA) Editor: Click on the "Visual Basic" button in the Developer tab to access the VBA Editor. This is where you will write the code for your macro.
3. Insert a New Module: Right-click within the Project Explorer window, which is usually located on the left-hand side of the VBA Editor. From the context menu, select "Insert" and then choose "Module." This action creates a new module where you will write your macro code.
4. Write the Macro Code: In the newly created module, write the VBA code to perform the desired task. This may involve using VBA functions and objects to interact with Excel, open web browsers, input data, and navigate through web pages. To better understand the form

the code will need to take, it is recommended to identify each step by hand so that pseudocode can better guide the coding process.

5. Test the Macro: After writing the code, it is essential to test the macro's functionality within the VBA Editor. Run the macro to ensure it executes the desired actions accurately and without errors.
6. Assign Shortcut (Optional): Like the previous section, the user can assign a keyboard shortcut to your VBA macro for quick execution. This can be done by creating a custom shortcut within Excel or by adding a button to the ribbon that triggers the macro.
7. Save the Workbook: Once the user is confident in your VBA macro's performance, save your Excel workbook to preserve the VBA code. This ensures that your macro is accessible for future use and can be easily shared with others.

For more detailed guidance on macro creation and structure, the user is asked to refer to the manual provided by the team. This manual offers comprehensive instructions tailored to the specific needs and objectives of the procurement office optimization project. When writing VBA code, it was essential to have a good understanding of the language's syntax and structure. We took advantage of online resources, tutorials, and forums to develop knowledge and understanding of VBA programming and further enhance Macro development skills. The team decided to use the VBA method to make the macros as the VBA method provided a more dynamic environment to cater to the exact needs of the project. The “Record Macros” feature was included in this section as simple macros can be created using the record feature.

Macro Testing

To test macros and ensure they clean the contracts as desired for the project, the Senior Design team recommends following these steps:

1. Define Test Cases: Based on the problem statement and project goals, define specific scenarios or test cases that represent the types of contracts you expect the macros to handle. For example, create test cases for contracts containing irrelevant information, contracts with multiple items, and contracts with varying formats.
2. Prepare Test Data: Gather sample contracts that reflect the test cases that have been defined. These contracts should cover a range of scenarios and complexities to thoroughly evaluate the macros' performance.
3. Execute Macros: Run the macros on the test data, simulating real-world usage. Pay close attention to how the macros handle each contract, including their ability to identify and remove irrelevant information while preserving essential details.
4. Verify Results: After running the macros, carefully review the output to ensure that the contracts are cleaned as desired. Verify that irrelevant information, such as unrelated items or descriptions, has been effectively removed, leaving behind only pertinent details related to procurement.
5. Compare Against Expected Outcome: Compare the cleaned contracts against your expected outcome based on the defined test cases. Ensure that the macros produce results consistent with the project goals and requirements outlined in the problem statement.

6. Iterate and Refine: If any discrepancies or issues are identified during testing, iterate on your macros to address these issues. Refine the logic, adjust parameters, or incorporate additional checks to improve the accuracy and effectiveness of the cleaning process.
7. Test Edge Cases: In addition to standard test cases, consider testing edge cases or corner cases that may pose challenges to the macros' functionality. This includes contracts with unusual formatting, ambiguous descriptions, or unexpected content.
8. Document Test Results: Document the results of the testing, including any observations, issues encountered, and modifications made to the macros. This documentation will serve as a reference for future testing and troubleshooting.
9. Seek Feedback: Share the test results with project stakeholders, such as team members or stakeholders from the procurement office. Solicit feedback on the macros' performance and effectiveness in meeting the project objectives.

By systematically testing macros against a variety of scenarios and use cases, a developer can verify the macro's effectiveness in cleaning contracts and achieving the desired outcomes outlined in the project's problem statement and goal.

Results

To measure the success of the Senior Design team, the following goals had to be met, articulated in the 'Goals,' 'Requirements and Constraints,' 'Metrics,' and 'Deliverables' subsections of the 'Problem Description' sections.

Firstly, one of objectives described in the 'Goals' subsection was to "expedite the time it took to find a SKU for items that would need to be procured at the airport." The Senior Design team

had decided to measure this objective in how many results would occur when a search was conducted. The initial condition that there were thousands of results to look through when making a search, demonstrated by the image on the following page, Figure 9.

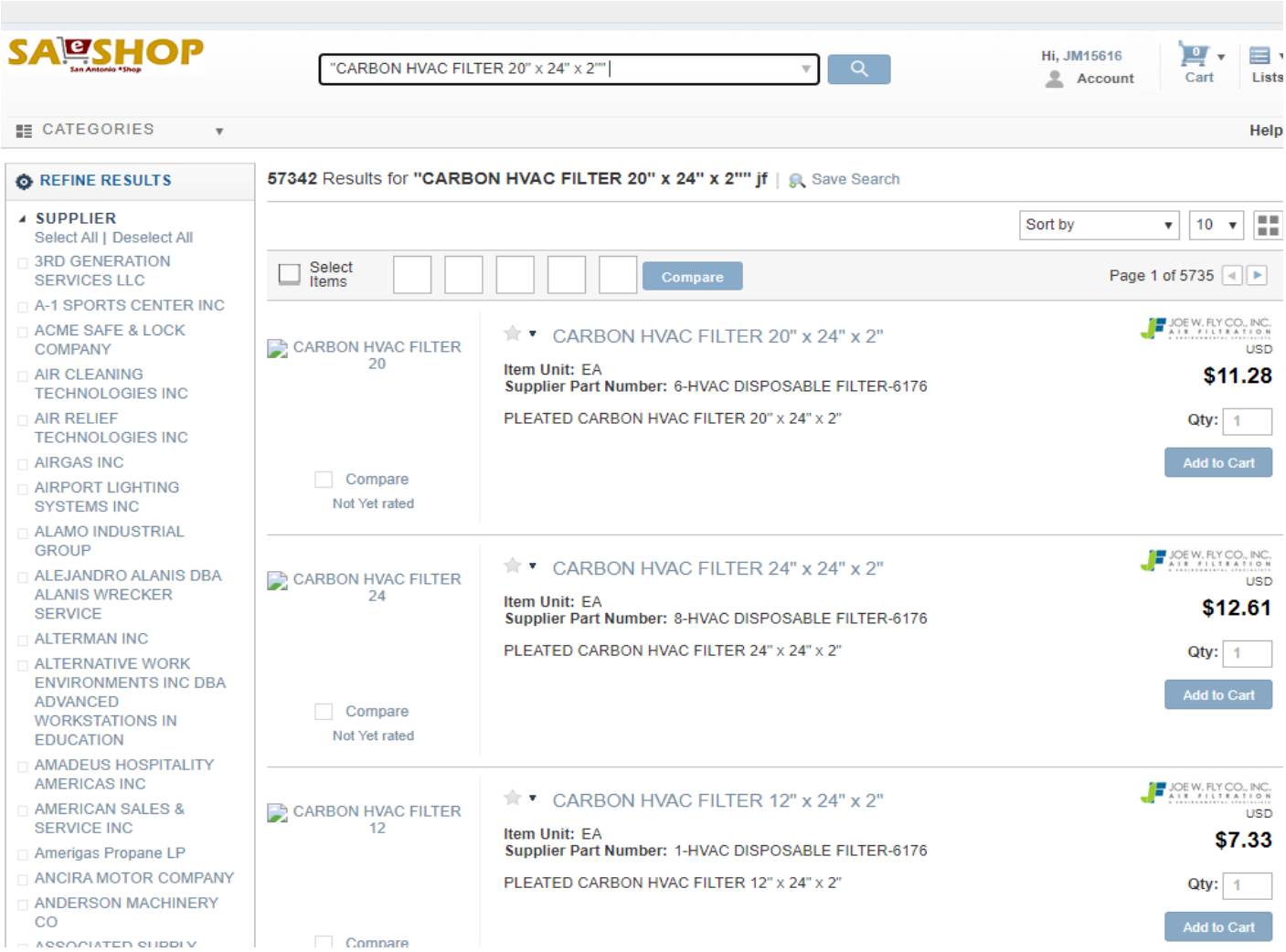


Figure 4. Search results after looking for a part by its description. The catalog shows 57342 separate items.

After the team conducted a search with a part number, only tens of potential results became matches, marking an extremely significant decrease in the time it would take to find the part a procurement team member would need.

Secondly, the objectives in the 'Requirements and Constraints' subsection detailed that the Senior Design team would need to deliver an AI tool "that could query a static database and identify the SKU being searched;" a user manual "such that contracts could be added, modified, and removed;" and a chatbot with the capability to "be used to find SKUs." These requirements were met, because the Senior Design team delivered the AI tool in the form of a web-based method that can query an Excel database comprised of the various contracts given and can identify the SKUs (part numbers) within, delivered the Owner and Operations Manual that allows for contracts to be changed within the Excel database, instructing readers how to do so, and delivered the format on conversing with Chat GPT, a chatbot, that has the capability of analyzing the Excel database and find the SKUs in each contract.

Thirdly, the 'Metrics' subsection detailed four objectives to be met if the project was to be considered completed:

1. Query a contract;
2. Articulate the information requested in a manner that is comfortably understandable for a user to read;
3. Decrease the time it takes to look for a part in Vroozi; and
4. Provide a user's manual to the Airport such as the manual and the chatbot allow for the Procurement team to make further adjustments in accordance with their future needs. As mentioned, the AI Chatbot took the form of Chat GPT and can query a contract, delivering that information to users in an understandable manner, which decreased the amount of time it took to complete a search for a part and accompanies the Owner and

Operations Manual, working jointly with it to provide excellent user instruction on the inner workings of the Excel database, Macros, and contract conversion.

Lastly, the objectives found in the 'Deliverables' subsection are as follows:

1. Design and implement a web-based methodology using pre-existing programs to set up contracts that would allow Chat GPT to analyze them and return information to the user.
2. Deliver a user manual as previously detailed in this very subsection.
3. Deliver the final documentation of the project, encompassing both the final report and the final presentation.

The Senior Design team managed to complete all these objectives in accordance with the previous goals that were analyzed, with the final documentation being completed, presented, and delivered at the end of April 2024.

3. Final Remarks

Introduction

The Senior Design team embarked on a transformative project aimed at enhancing contract management processes for the Airport. With a focus on leveraging AI technologies, specifically Chat GPT 4.0, the team successfully developed a solution tailored to the Airport Procurement team's needs. Initial goals centered on processing a diverse array of contracts, numbering twelve in total, yet challenges emerged during the implementation phase. Despite these obstacles, the team's dedication and ingenuity prevailed, resulting in significant progress. By converting PDF contracts to Excel format and implementing macros to streamline cleaning and formatting, the team achieved

remarkable automation, reducing manual effort substantially. However, issues with data consistency and labeling underscored the ongoing need for refinement in both process and technology. Nonetheless, this project marks a significant milestone, laying the groundwork for a more agile and efficient procurement system at the San Antonio International Airport, poised to meet future challenges with confidence.

Summary of Results

The Senior Design team fulfilled the goals set out by the Airport. We developed an AI-based solution so that the Airport Procurement team would the contracts the airport had supplied. Ultimately, the senior design team got ahold of twelve contracts that were clean and query-able using the solution designed for the project. Under the team's experimentation, Chat GPT accomplished what was expected, and accurately produced the information requested after approximately one minute of search time. The solution that the senior design team made provided a method to find suppliers for the procurement team at the San Antonio International Airport quicker than the legacy system.

Conclusion

The project was considered a success. The Senior Design team had made goals and objectives to follow through and did. The subsection 'Results' details the large undertaking that was done for the project by the team. The contracts were not in a state that was productive for what the Airport needed, and the Senior Design team worked on a solution for that, which culminated in the conversion of the PDFs to Excel, putting the contracts in a state that could be

further analyzed. Once the conversion was made, the data was too messy for even a human to understand the data easily. Another solution was workshopped to solve that, the macro, and was tested, resulting in clean contracts that were now legible to a human being. However, they initially were not legible to a machine, so for automation to occur, the files needed further reformatting. The macros became the critical component in this process, as changes to their code allowed for further modifications of the contracts to become readable by Chat GPT. Chat GPT was also a critical success in this project, as it was able to look through the entire Excel database in search of a key word, and this database comprised of many contracts that would be spanning tens of thousands of rows full of unique results.

However, some issues came up during development. Under testing that stretched the limits of what Chat GPT 4.0 can do, we discovered just how important both the formatting of the Excel database and the initial conditions of the data within the original PDFs were, as it would have problems stemming from missing pieces of data. While not capable of making the same logical leaps a human can, the computational potential of this chatbot more than made up for its deficits. The resulting format the contracts took while in the Excel database allows for other solutions to take the place of Chat GPT and allows for the progress this project made to not be lost. The macros all worked as intended and reformatted every contract they were given, resulting in incredible success as the estimated one hour of manual cleaning is now down to a matter of seconds. The Senior Design team learned new a new coding language, VBA, and had to use many of the soft skills learned while in university. Being their own boss and managing their time became critical to the success of this project, as well as communicating effectively.

The creation of the manual required great documentation of the processes the Senior Design team made, as well as mastery of two writing styles: formal and technical writing, and informal but comprehensive writing. This meant that while data is communicated in this report in a formal manner, appropriate to its audience, the Manual had to be crafted so that many kinds of people with varying skills, abilities, and understanding of the topics discussed could read and understand the solution this team created. That means a deep understanding of the underlying data and concepts had to occur so that the information can be presented in both these manners effectively.

Pictures were taken, small steps had to be recorded, and abstract concepts had to be explained, all so that someone who has zero knowledge about macros, Chat GPT, Excel, or even the Airport's procurement process could learn and repeat the Senior Design team's success. Overall, by leveraging the power of artificial intelligence and the varying skills of the two members of the Senior Design team, this project has laid the groundwork for the San Antonio International Airport to have in their possession a more agile, responsive, and efficient procurement system, ready to meet the challenges of serving the community and the aviation industry in the future to come.

Recommendation for Future Work

The senior design team initially anticipated that automating the contract cleaning process using macros would streamline operations. However, the variability among contracts made it challenging to determine the exact number of macros needed for each contract, leading to unforeseen complexities. Moving forward, it would be beneficial to develop a more standardized

approach to contract cleaning, aiming for a predetermined set of macros that can effectively handle various contract formats. This would enhance automation and simplify the cleaning process, ultimately improving efficiency and reducing manual effort.

Additionally, while AI technologies such as Chat GPT 4.0 were envisioned as comprehensive solutions, the reality proved to be more nuanced. The process of querying the AI for specific information was found to be tedious, with results sometimes lacking relevance. To address these challenges, future projects should explore ways to refine AI integration, focusing on improving user interaction and enhancing result accuracy. By refining both the automated cleaning process and AI capabilities, future endeavors can achieve greater efficiency and effectiveness in contract management tasks.

References

AI In The Workplace: 12 Ways Workplace AI Can Improve The Office (2023) *OfficeSpace Software*. Available at: https://www.officespacesoftware.com/blog/ai-in-the-workplace-ai/ (Accessed: 6 October 2023).

What is artificial intelligence and how is it used in the workplace? (no date). Available at: https://www.shrm.org/resourcesandtools/tools-and-samples/hr-qa/pages/what-is-artificial-intelligence-and-how-is-itused-in-the-workplace.aspx (Accessed: 6 October 2023).

Discover the power of building Intelligent Products with generative AI (no date) *Slalom Build*. Available at: https://www.slalombuild.com/blueprint-articles/code-meets-imagination-using-generative-ai-to-build-intelligentproducts (Accessed: 6 October 2023).

Drapkin, A. (2023) 'Does Chat GPT 4.0 Save My Data? OpenAI's Privacy Policy Explained,' *Tech.co*, 29 June. Available at: https://tech.co/news/does-Chat GPT 4.0-save-my-data (Accessed: 6 October 2023).

How AI turns text into images (2023) *PBS NewsHour*. Available at: https://www.pbs.org/newshour/science/how-aimakes-images-based-on-a-few-words (Accessed: 6 October 2023).

Appendix A: User Manual

User Manual: San Antonio International Airport GPT Procurement Project

Written by:

Sebastian Blackburn

Anthony Sadler

Key Contributors:

Mr. Syed Mehdi

Dr. Rafael Moras

Mr. Jordan Mergele

Table of Contents

1. General Overview

1.1. Introduction

Hello there! Welcome to the Owner's and Operation Manual of the Chat GPT 4.0 Procurement Contract Procedure!

This Manual was made by the St. Mary's 2024 San Antonio International Airport Engineering Senior design Team:

Sebastian B and Anthony S.

Before we get to all that you would need to know regarding this method we've pioneered, we of the Senior design Team would like to give thanks to these individuals, whose support made this project possible:

- Mr. Syed Mehdi, whose generous support made sure we were able to complete this project and acted as our sponsor for this project, representing the San Antonio International Airport and its Procurement Team.

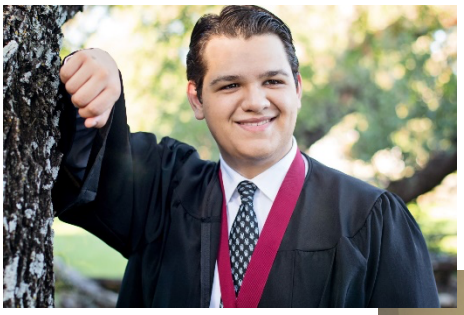
- Mr. Jordan Mergele, whose presence made sure we never strayed from the end goal we had set at the start of the project, and acted as a key contact withing the Procurement Team as the San Antonio International Airport

- Dr. Rafael Moras, whose constant oversight made sure we could bring you this manual at the quality you need.

- Dr. Robert O'Connor, whose intervention allowed us to surpass many of our obstacles during our window of opportunity to do this project.

- And of course, You, the reader, for taking the time to sift through this manual and being able to bring use all that we have put together.

So, sincerely, thank you!



Sebastian

Anthony

1.2. What Is This Manual For

This user manual provides guidance on utilizing the Chat GPT 4.0 procurement solution developed by the Senior design team for San Antonio International Airport.

The solution aims to speed up finding SKUs for items needed for procurement within the airport's system.

Required Tools:

- PDF version of procurement contracts

- PDFTOEXCEL.com

- Microsoft Excel

-Macros (provided by senior design team)

- Chat GPT 4.0 (premium version)

1.3. Where You Come In

So, when should someone be using manual? Well, every time you have a question, of course!

From detailing every step of the process, to troubleshooting, to even designing your own stuff, this manual has it all!

1.4. How to Navigate This

The bigger question now is how do you find stuff within these pages?

Using our table of is a good place to start, as it has the titles of various sections and the headers of the plentiful subsections to make sure you know where you are, like a map.

This manual will describe the general knowledge needed in each step of our process, going into the details of each component in their own sections.

This allows you, the reader, to simply look up the general or specific information based on your needs. How convenient!

2. Process Overview

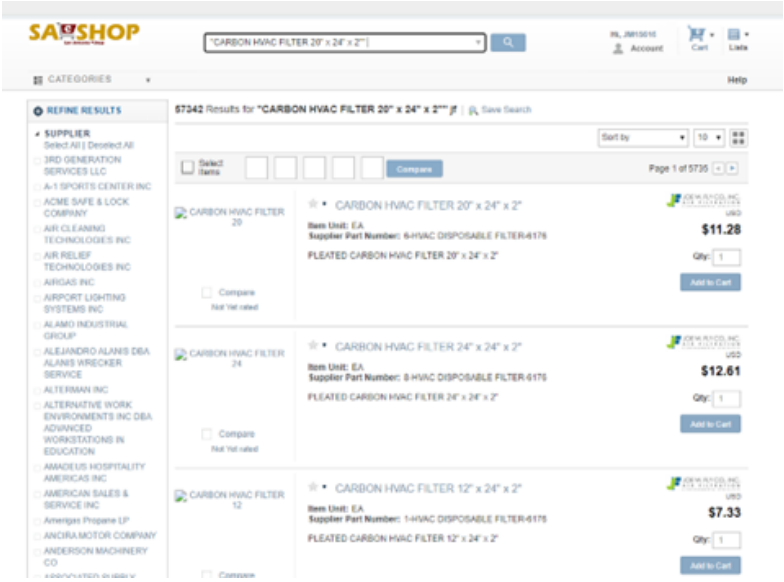
2.1. What is the Solution You Are Going to Use?

The solution contained in these pages is meant to solve an extremely specific problem.

But what is the problem? This is a screen grab of Vroozi in action, the main website that the San Antonio International Airport uses to act as a catalogue to find the parts they need.

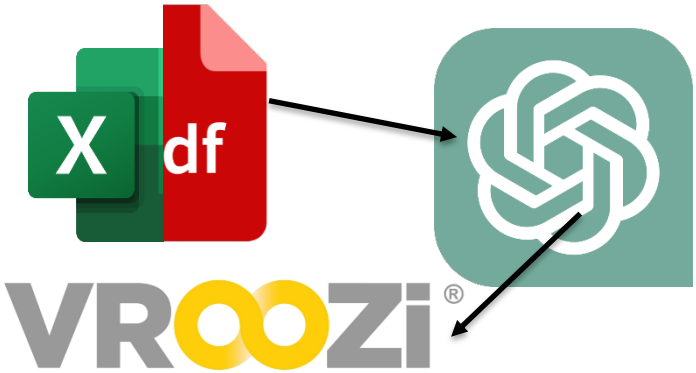
As you can see in the ballooned area, searching for a part yields thousands of results, and the user interface for this system is not enough to properly display each individual part so the user is not able to simply look and find what they need.

Our method consists of using Artificial Intelligence, in the form of Chat GPT 4.0, to take the information found directly in the contracts that the San Antonio International Airport has in their possession and give it to the user, so that the user can then go to Vroozi and search directly for what they need without having to search through the thousands of results found in the original search.



A normal search from Vroozi through the Airport's SAShop system

Potential blurry: The number of results from the search is 57,342 different matches.



2.2. Why Would You Use This Solution

The reason to use this method is simple:

It is simply faster to ask Chat GPT 4.0 for the part number directly and search it on SAShop, than to just search through all the results it would give a user if they did a general description search, so this method acts like a shortcut.

The Senior Design team, as part of the project, supplied San Antonio International Airport with a copy of the contracts that were already converted. That Excel File contains the 12 starting contracts, as proof of concept, to show what the finished product will eventually look like and allows users to add more contracts and make the shortcut better.

3. Steps for Utilization

3.1. Convert Procurement Files

To convert the files, this method is to be adopted:

Ensure you receive the PDF copy of the Contract you wish to convert.

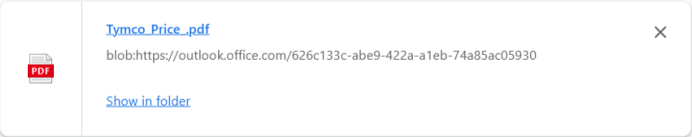


Image of the file found in File Explorer.
(Note: Windows 11)

Once the contract is received, open https://www.PDFtoexcel.com/

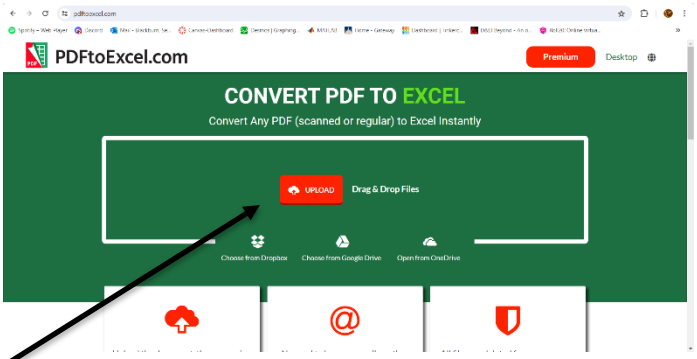
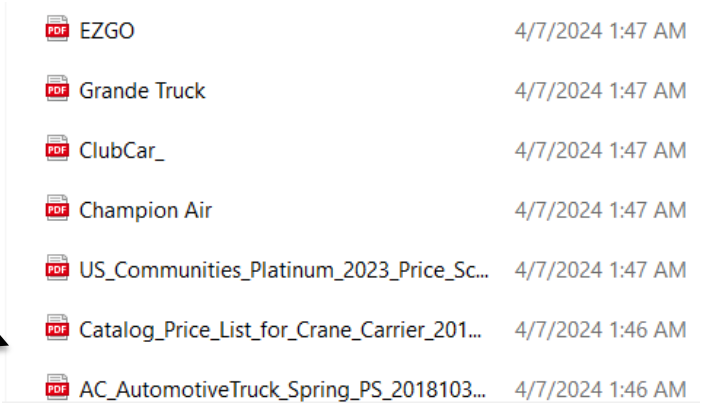


Image of the website PDFtoExcel.com on Google Chrome

Click the red Upload button and then search for your file.

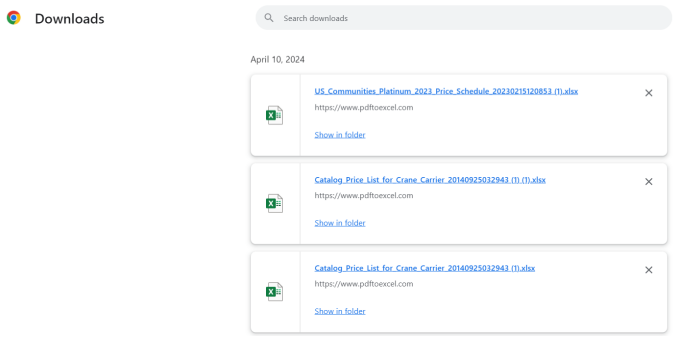


The various downloaded contracts found in File Explorer.
(Note: on Windows 11)

The website will then take time to automatically convert the file into an Excel file. When it finishes, click Free Download.

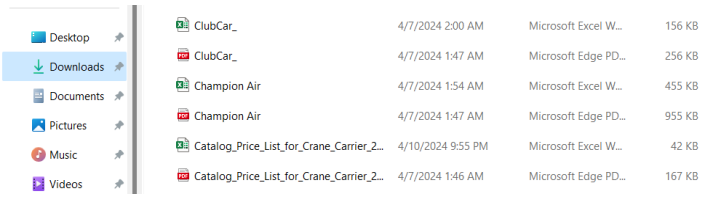
The relevant section of the website PDFtoExcel after the conversion is complete.

Congratulations, the converted file will now be in your device's 'downloads' folder.



Downloads found in Google Chrome.

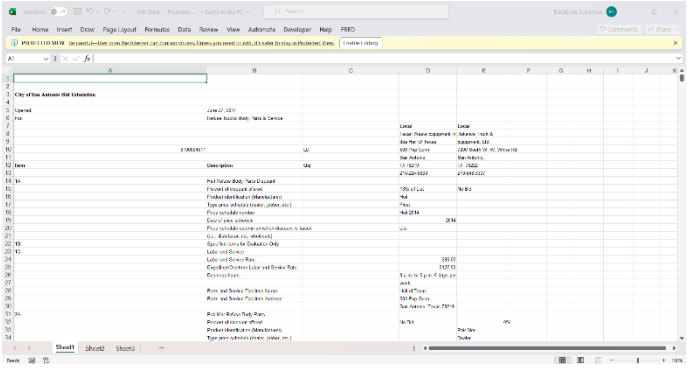
If you are using a browser like Google Chrome, you can find it in your downloads, like what is shown here, or you can find it in your file manager app (Finder for Mac and File Explorer for Windows).



Files found in File Explorer.
(Note: Windows 11)

3.2. Clean Contracts Using Macros

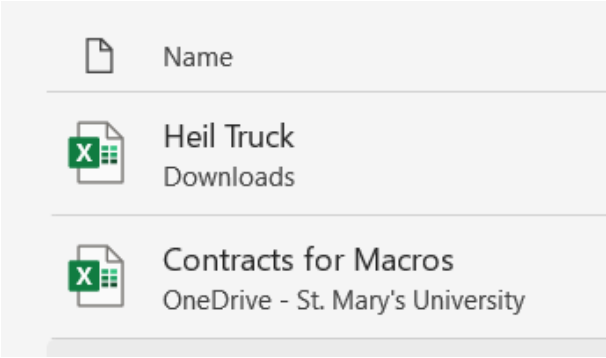
Now that the contract is now in Excel's file format, you can use it to open and edit the file, with this image serving as an example.



Contract from Refuse Trucks and Body Parts opened in Excel.

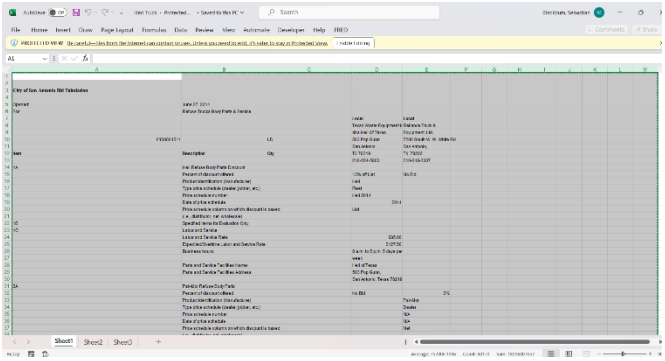
Make sure you have access to the Excel workbook that contains all the converted contract files, as you can add contracts to it and are going to use the Macros that are saved in that workbook. The file is currently titled 'MegaFile.xlsm'

Note: Always make sure the file you have downloaded is the most recent one, as names may change over time



Depicted here is a converted contract ('Heil Truck') and the file we are looking for (example being 'Contracts for Macros').

Once everything is checked out and running, open the contract you just converted and copy the sheet. You can do this by selecting the 'A' column and then, while holding the shift key, select the 'M' column. This should ensure that the whole table is copied.



The contract ('Heil Truck') after columns 'A' through 'M' is selected.

Now create a new sheet in the ‘Contracts for Macros’ file by clicking here. Also rename the sheet you just made, for clarity and later use.

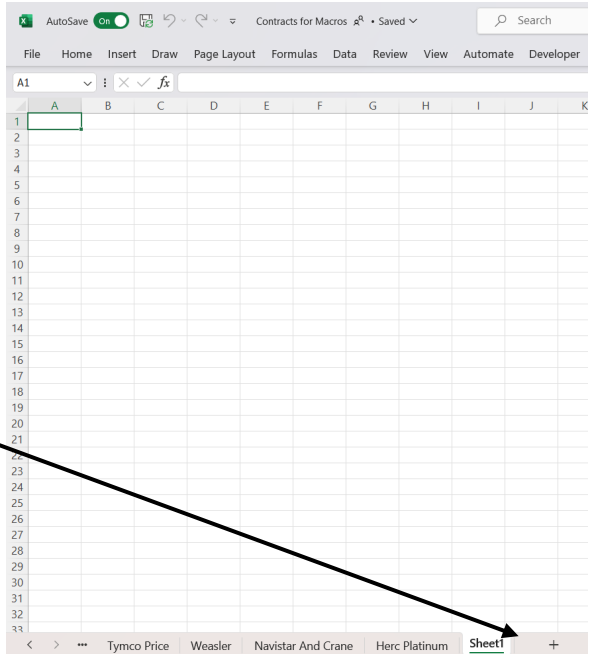
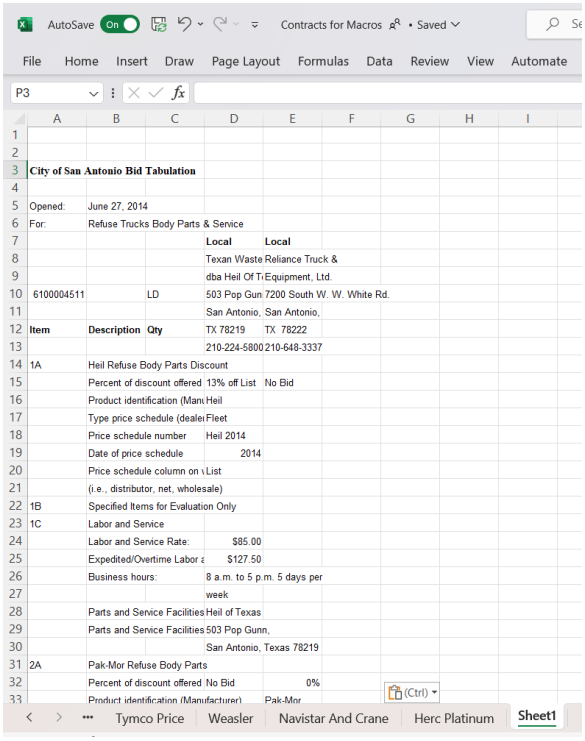


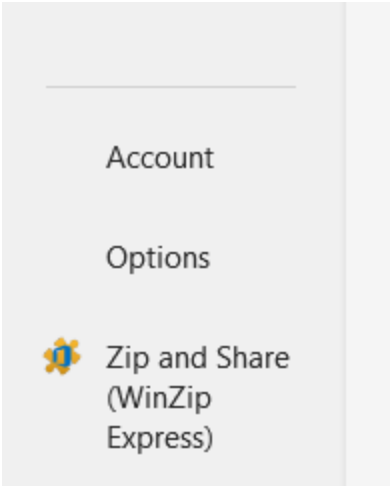
Image of the workbook after a new sheet is made, see the ‘+’ sign circled; that is where you click to make a new sheet.

Paste the contract information into the document. If you find that there is information missing due to a missed column, simply undo the paste, go back to the converted file, and select a different column to the right of ‘H’ and try again.



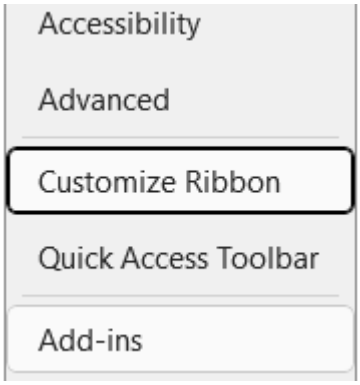
The new contact after being placed into the workbook.

When your new contract is set in place, under the ‘File’ Tab, open ‘Options’ and go to ‘Customize Ribbon.’

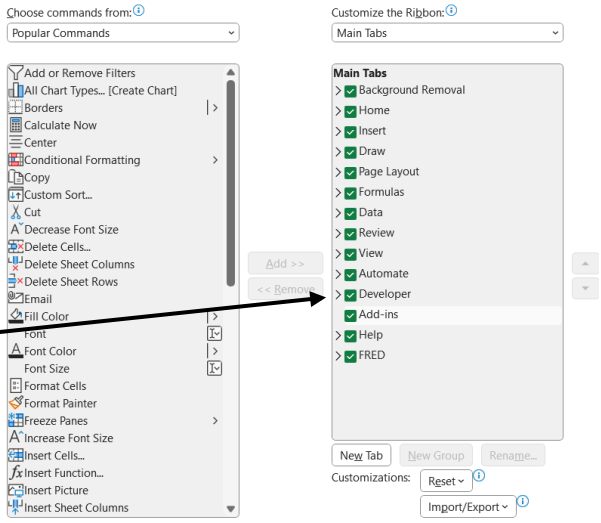


On Windows 11, you can find ‘Options’ towards the bottom.

The menu along the left of the window has ‘Customize Ribbon’ towards the bottom.



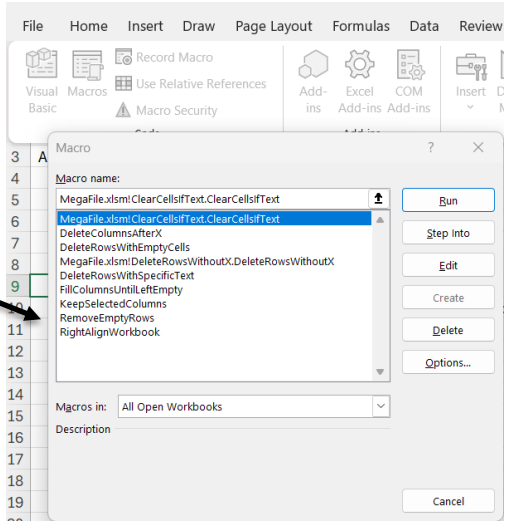
Once there, make sure the ‘Developer’ main tab is checked, as this is needed to access a critical component to the macros, the VBA editor. It should look like this.



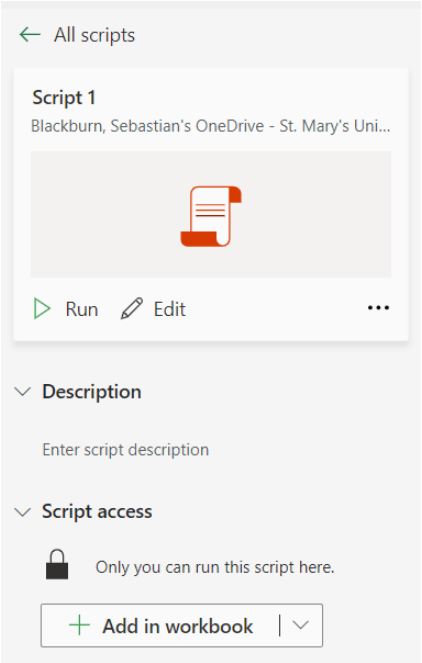
See the ‘Developer’ tab is checked, that means we can access it.

Returning the ‘Contracts for Macros’ workbook, you can now access the Macros found in the ‘Developer’ tab in the main ribbon, click it. This is how it should look like.

This is going to be where you select the various macros from, all organized in the box for you to see.



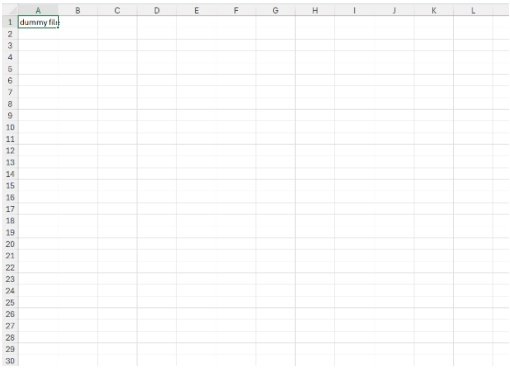
From this screen, you can select what macro you wish to use. For example, we are using ‘Script 1’ on an empty sheet.



After selecting ‘script 1’ from the ‘Office Scripts’

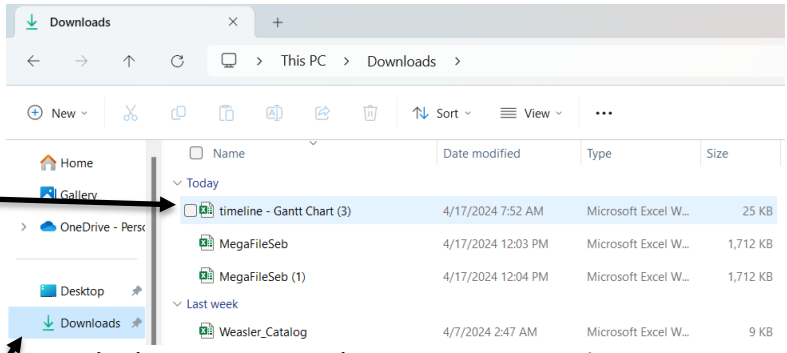
You can hit ‘Run’ to start the macro and before you know it, the macro has activated, done its business, and left. All that remains is a clean macro.

Note: There is no warning when a macro is completed; Be careful to not apply a macro multiple times.



3.3. Sending Contracts to Chat GPT 4.0

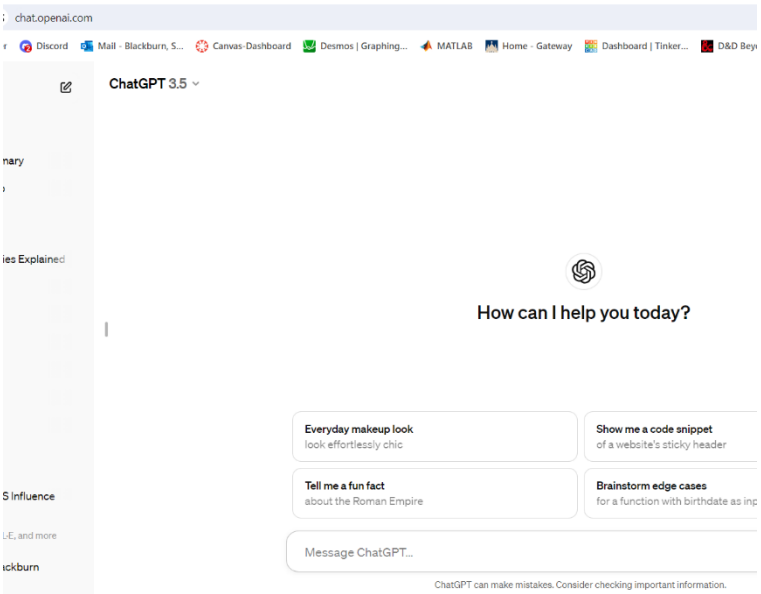
Once the Excel file containing all the cleaned contracts is ready to be given to Chat GPT, you need to make sure you can find your file quickly and easily for later.



File is safely saved in the 'Downloads' folder, this Excel file is easily found and accessed.

This could mean moving it to your desktop or placing it in a folder you can remember or recognize quickly.

Open your browser and login to the Open AI website https://chat.openai.com/. Chat GPT 3.5 is not capable of reading files in the manner required. You need Chat GPT 4.0.



The starting page after logging in.

Note: this is Chat GPT 3.5 for this image.

Once you login and can access Chat GPT 4.0, there are two methods to choose from. The first is to simply click and drag the Excel workbook file, and the second is to click the paperclip icon and select the file.

Congratulations, Chat GPT now can read and analyze your file.

3.4. Query Chat GPT 4.0

Now that everything is in place, the next step is to ask Chat GPT for the information you would need, curated with the help of macros.

The information retained in the workbook file includes:

- The part number in column A
- The contracted price of the part in column B
- And the vendor who is selling the part in column C

Chat GPT 4.0 can even pick up on this you lead it!

You

Well, can you read the sheets in the spreadsheet I sent?

ChatGPT

The Excel workbook you sent contains the following sheets:

1. Main
2. Task Manager
3. Club Car

This snapshot is toward the beginning of analysis, asking if Chat GPT 4.0 had successfully accessed the file.

You

Thank you, can you identify the how information is stored in the Navistar and Crane sheet?

ChatGPT

The "Navistar And Crane" sheet in your workbook is organized with the following columns:

1. **Part #** - This column contains part numbers for various items.
2. **List Price** - This column shows the list price for each part.
3. **Supplier** - This column indicates the supplier, which is "Navistar and Crane" for the rows listed.

Here's a snapshot of the data from the first few rows:

Part #	List Price	Supplier
001 -1033	189.68	Navistar and Crane
002 -0051	1156.6	Navistar and Crane
002 -0064	186.36	Navistar and Crane
002 -0067	160.8	Navistar and Crane
002 -0087	262.68	Navistar and Crane

This snapshot shows that Chat GPT 4.0 can pick up on the formatting of the files and display it to the user.

The suggested format of asking questions should contain the following if not mentioned earlier:

- The column of each piece of information you are using and searching for

- A distinct and clear link between the columns you are using and what that column contains.

Here are a bunch of examples that show and highlight the differences in the questions.

Note that while there is a lot of variety in the specifics of the questions, the underlying format is consistent and gives Chat GPT 4.0 critical markers for its analysis.

You
Can you tell me the part number associated with the lowest listed price in the Navistar and Crane sheet?

ChatGPT
The part number associated with the lowest listed price on the "Navistar And Crane" sheet is `010.5D`, with a price of $57.60. If you need more details or further assistance, feel free to ask!

You
What about in the sheet Club Car?

ChatGPT
In the "Club Car" sheet, the part number with the lowest listed price is `1010028`, priced at $0.45. Let me know if there's anything else you'd like to explore or any other assistance you need!

In this conversation, the same question is asked twice, but the framing only happened once, seen on the previous page (same conversation).

MegaFile2.xlsx
Spreadsheet

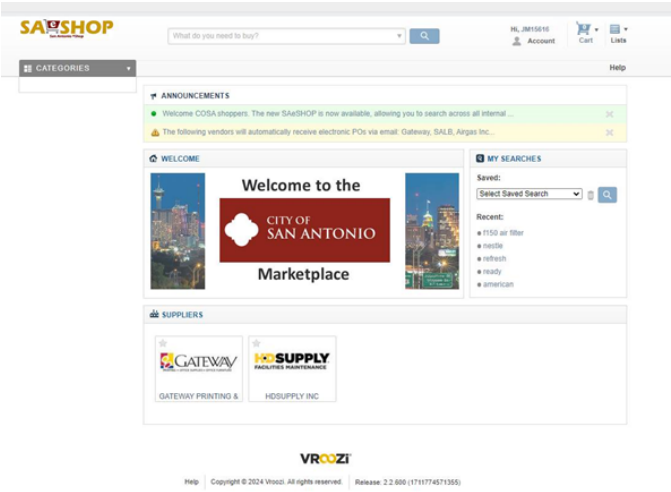
In all sheets column A is the part number, column B is the list price, and column C is the vendor. Give me the part number and vendor for the highest List Price, for a single part, as listed in column B.

ChatGPT
Let's proceed with your updated request by examining the new file you've uploaded. I'll look for the part number and vendor associated with the highest list price across all sheets, considering the columns as you described: column A for part number, column B for list price, and column C for vendor. I'll begin by identifying the highest priced item.
The newly uploaded Excel file contains the same sheet names as before. I will now extract and identify the part number and vendor for the highest priced item from column B across all sheets. Let's proceed.
The part number with the highest list price across all sheets is "43.88", provided by "Champion Air". The highest list price recorded is $431,041.00. If you need more details or further analysis, feel free to ask!

Note: The core components in this conversation are done all at once, showing that Chat GPT can handle a variety of ways of asking the same question

3.5. Searching SAShop

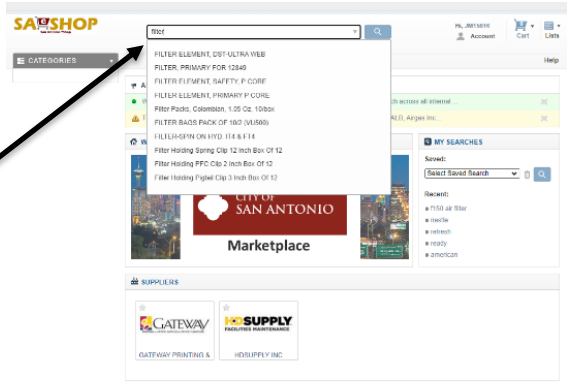
Once you get the Part Number (SKU) and the Vendor, you can now go to Vroozi.



The main SAShop page

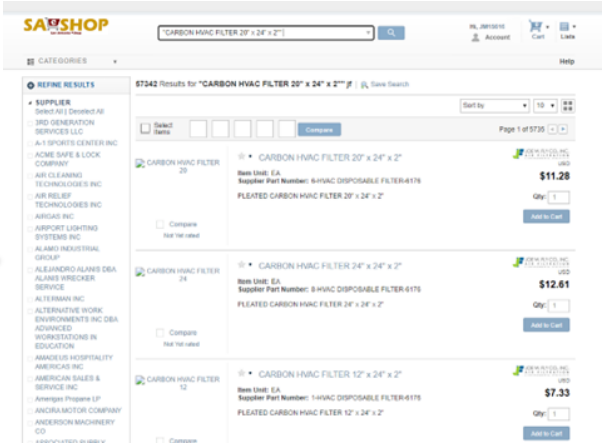
To begin searching, you're going to need to head to your designated shopping website, SAShop.

Image of SAShop with the search bar mid-search.



From the main page and after logging in, you will find the search bar at the top-center, where you can begin your search for whatever part you may need.

Using the vendor, you found with Chat GPT 4.0, you can narrow your search result to that specific vendor. This prevents repeat part numbers from confusing the result.



The suppliers are now displayed on the left.

4. Cleaning Contracts

4.1. Will These Macros Work For Every Scenario?

No, it is impossible to confidently say that we have created all the macros that will be needed for any kind of contract.

The macros that we created were successful during testing, however, unforeseen and unique situations may occur where the current list is insufficient.

In the subsection 'Where Could You Find Sources to Help Edit and Code a Macro' on pg. 47, there will be a brief description on how we created the macros which can give insight on how to make one to suit the needs for a new contract.

There are also 2 methods described in this manual that detail how to create a macro, under the section 'What is a Macro' (pg. 27) using either a 'Record' feature or direct coding.

4.2. Cleaning Contracts With Macros

If you would like a direct path and instruction set to clean a contract with a macro, follow these steps:

- Open the Excel file containing the PDF-converted contracts.

- Access the "Developer" tab in Excel and click on "Macros" to view the list of available macros.

- Select the appropriate macro designed by the Senior design team for contract cleaning.

- Execute the selected macro by clicking "Run" or double-clicking its name.

- Wait for the macro to complete its tasks, which may include formatting, data extraction, or other cleaning operations.

- Review the cleaned contracts to ensure accuracy and completeness.

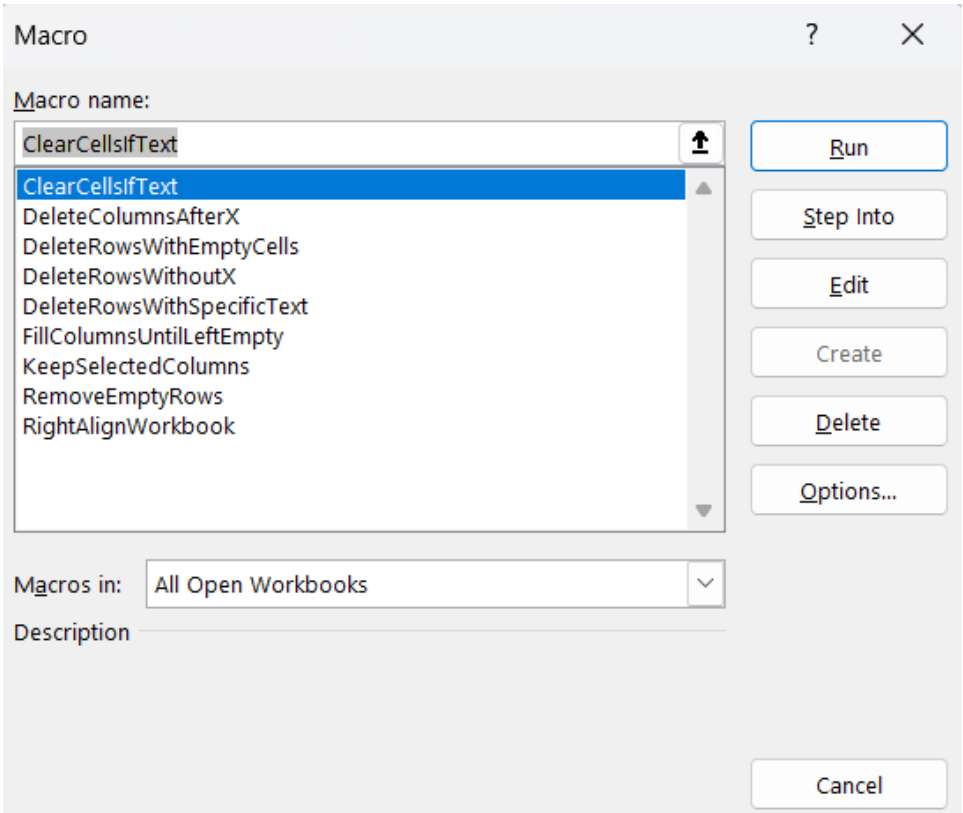
- Save the cleaned contracts in the desired format.

- Repeat the process for any additional contracts that require cleaning.

- If necessary, customize or modify the macros to accommodate specific contract cleaning requirements.

4.3. Selecting the Right Macro

To the right are the macros that were made for this project during the development of the manual. Please ensure that any macros added by your team are logged in some way, so that they are not lost in the future.



Each macro made is designed to solve a specific problem, and so must be chosen for the job at hand. Here are the nine stock macros that come with the Excel workbook and what they do.

1. ClearCellsIfText

This macro is a simple, yet effective, bit of code. It asks users for a key word. Once it received that key word, it will search every sheet in the workbook, search all cells with content in it (meaning data, words, anything), and then if it finds the key word within the cell, it just empties it.

2. DeleteColumnsAfterX

So, Excel uses an alphabet-based system to decide the names of their columns. What this macro does is that it can take a letter decided by the user, identify the column that it is attached to, and begin deleting the information in columns that come after (to the right of) the designated column.

3. DeleteRowsWithEmptyCells

This macro goes through the worksheet and deletes rows with empty cells. It does this by using a key word given by the user, and it looks through each row and the relevant columns given by an automatic range and deletes the row if the condition of having a single empty cell in range is empty.

4. DeleteRowsWithoutX

This macro will take a key word that the user decides on, something critical like the front number of a part number, let's say ATD (Pulled from the contract under the sheet name 'Allison Transmission'), and search through the entire workbook, deleting every row that does NOT contain that key word.

5. DeleteRowsWithTextX

This macro will take a key word a user gives it and search the first column in Excel, column A, and look backwards from the last row for that key word. If it finds the word, it will delete the whole row, so it is best used when deleting a series of rows that contain information you may not want to use.

6. FillColumnUntilLeftEmpty

This macro will take a key phrase a user inputs and fill all empty cells with that text. This is particularly useful to fill in the price list, because if the data for the price goes missing, you can use this macro to fill in the missing data with 'N/A.'

7. KeepColumns

This macro is a simple one. It asks a user to select a series of columns. Once the columns have been selected, it then looks through every column. If it is not in the range of selected columns, it gets deleted. This is a useful tool for quickly removing columns that do not contain essential information like the price or part number in a contract.

8. RemoveEmptyRows

This macro works by asking two questions: The first is to select the first column a user wants to look at, and subsequently, the second question asks the user to select the second column. It will then look backwards, going through every row, deleting the rows that have an empty cell in either column the user looks at. This is best used to look and remove missing data a user might find in the worksheet.

9. Check2ColumnsforEmpty

This macro is a derivative of the previous macro, RemoveEmptyRows. If functions very similarly but with one key distinction: It is pickier. What is meant by that is that the condition to remove a row is now to see if both cells are empty, instead of just 1.

10. RightAlignWorkbook

This macro is completely automatic. It looks for the last row and column with data in the worksheet and it loops inward, going from to the bottom right to the top left, setting the alignment of all the data within the workbook to the right.

4.4. What Is a Macro?

A macro is a set of instructions recorded or written in a programming language like Visual Basic for Applications (VBA) to automate tasks within Microsoft Office products like Microsoft Excel or Word. It allows users to perform repetitive tasks more efficiently by executing a series of predefined actions automatically.

Macros can be triggered manually by the user or automatically based on specified conditions. This improves productivity and enables customization of software functionality.

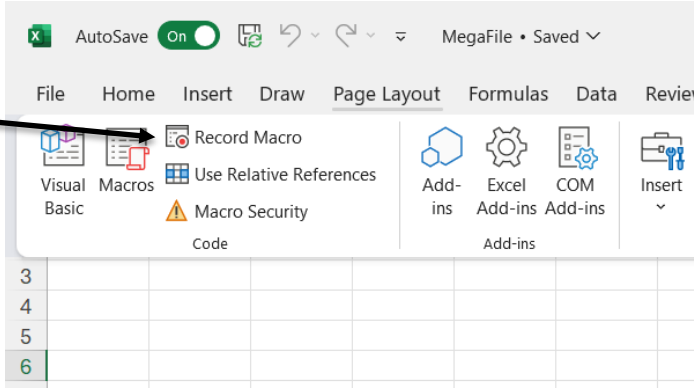
The following subsection contains information on how to make a macro using Excel in two separate ways.

4.4.1. Making Macros Using ‘Record’ Feature

1. **Open Excel:** Open Microsoft Excel on your computer.

2. **Enable Developer Tab:** If the Developer tab is not already enabled in Excel, you'll need to enable it. Go to Excel's Options, then select "Customize Ribbon" and check the box for "Developer" in the right pane.

3. **Open Macro Recorder:** Click on the Developer tab, then click on "Record Macro". This will open the "Record Macro" dialog box.



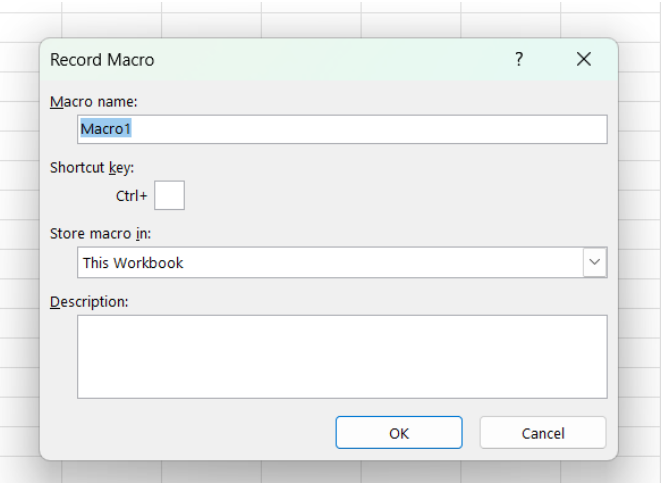
4. **Name Your Macro:** In the "Macro name" field, give your macro a name (e.g., "SimpleMacro").

5. **Shortcut Key (Optional):** If you want to assign a keyboard shortcut to your macro, you can do so by pressing a letter or number while holding down the Ctrl key. Make sure the combination you choose is not already assigned to another function in Excel.

6. **Store Macro In:** Choose where you want to store your macro. You can either store it in the current workbook or in your personal macro workbook, which allows you to use the macro in any Excel workbook on your computer.

7. **Description (Optional):** You can enter a description for your macro to remind yourself or others what it does.

8. **Click "OK":** Once you've set up your macro parameters, click "OK" to start recording.



9. **Perform Actions:** Excel is now recording your actions. Perform the actions you want your macro to automate. For example, you can format cells, enter data, apply formulas, etc.

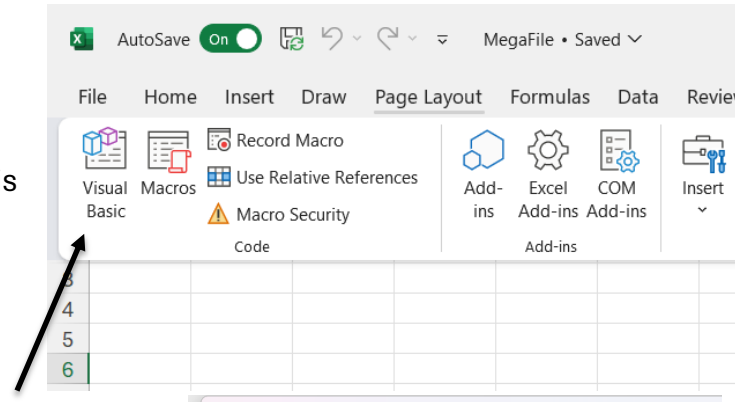
10. **Stop Recording:** After you've completed your actions, click on the Developer tab again and then click on "Stop Recording" in the "Code" group.

Your macro is now created! You can run it by either using the assigned shortcut key (if you set one) or by going to the Developer tab, clicking on "Macros," selecting your macro from the list, and clicking "Run." Excel will automatically generate VBA code based on the actions you performed during the recording. This method allows you to create simple macros without needing to write any code manually.

4.4.2 Making Macros Using VBA

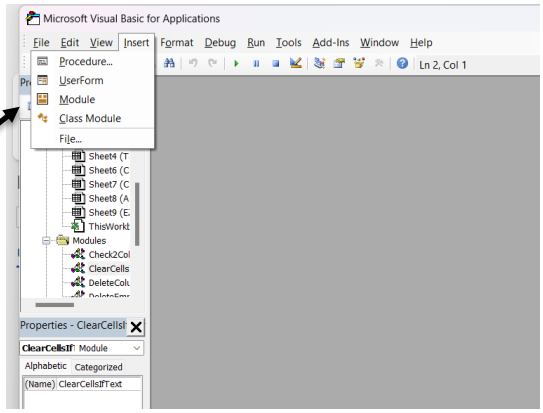
1. **Open Excel:** Launch Microsoft Excel on your computer.

2. **Enable Developer Tab:** If you haven't already, enable the Developer tab in Excel. You can do this by clicking on "File" > "Options" > "Customize Ribbon". Then, check the box next to "Developer" in the right pane and click "OK".

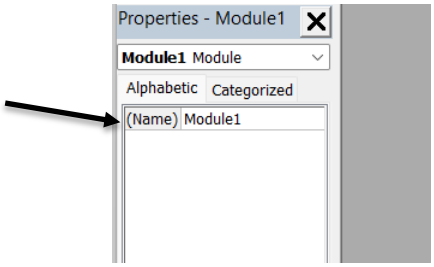


3. **Open Visual Basic Editor (VBE):** Go to the Developer tab and click on "Visual Basic" to open the Visual Basic for Applications (VBA) editor.

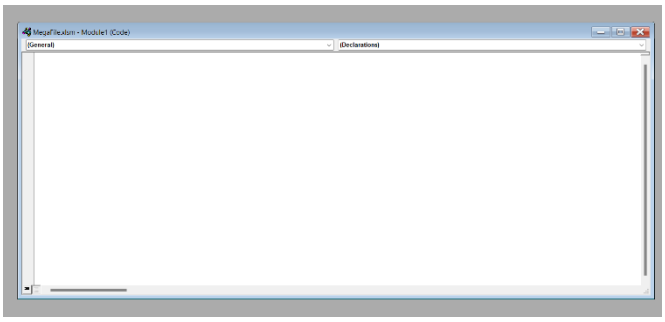
4. **Insert a Module:** In the VBA editor, locate the "Insert" menu at the top and click on it. From the dropdown menu, select "Module". This will create a new module where you'll write your macro.



5. **Name Your Macro:** You can give your macro a name by double-clicking on "Module1" in the project explorer window (left pane) and typing a new name for it.



6. **Define Your Macro:** In the module window (right pane), you'll write the actions your macro will perform. For instance, you might create a macro that formats cell, inserts data, or calculates values.



7. **Close the VBA Editor:** Once you've defined your macro, close the VBA editor by clicking the "X" button in the top-right corner of the window.

8. **Assign a Shortcut (Optional):** If you want to assign a keyboard shortcut to your macro, you can do so via Excel's interface. Go to the Developer tab, click on "Macros", select your macro, and click "Options". Then, you can enter a letter or number in the "Shortcut key" field (while holding down the Ctrl key) and click "OK".

9. **Run Your Macro:** Return to Excel. You can run your macro by going to the Developer tab, clicking on "Macros", selecting your macro from the list, and clicking "Run". If you assigned a shortcut key, you could use that instead.

10. **Verify the Output:** After running the macro, check to see if it performed the desired actions in your Excel workbook.

That's it! You've created and executed a basic macro in Excel without writing any code. This approach allows you to automate tasks and streamline your workflow using Excel's built-in features. As you become more comfortable with macros, you can explore more advanced functionalities and customize your macros further.

4.4.3 What Is the Programming Language Used

Visual Basic for Applications, or VBA, is the programming language used in the development of macros. The very next page, Pg. 33, has sources you can look to on learning how to code macros in this language.

While this manual has gone over how to access VBA studio and the next page details where to find help, here are some examples of the code behind the macros used in this project.

```
(General)                                                    DeleteRow

Sub DeleteRowsWithoutX()
    Dim rng As Range
    Dim cell As Range
    Dim searchText As String
    Dim lastRow As Long
    Dim i As Long

    ' Prompt user for the text to search for
    searchText = InputBox("Enter the text to search for:", "Search Text")

    ' Exit if user cancels or doesn't provide a search text
    If searchText = "" Then Exit Sub

    ' Find the last row in column A with data
    lastRow = Cells(Rows.Count, "A").End(xlUp).Row

    ' Set the range to the entire column A up to the last row
    Set rng = Range("A1:A" & lastRow)

    ' Loop through the range backwards to avoid skipping rows
    For i = rng.Rows.Count To 1 Step -1
        If rng.Cells(i, 1).Value <> searchText Then
            rng.Cells(i, 1).EntireRow.Delete
        End If
```

This screenshot is for the macro DeleteRowsWithoutX, and its functionality is detailed in the section ‘Selecting the Right Macro’ on pg. 24.

4.4.4 Where Could You Find Sources to Help Edit and Code a Macro

By leveraging the available learning space, you can enhance your skills in editing and coding macros for Excel. There are several places where you can find sources to help edit and code a macro:

1. Official Documentation: Check the official documentation provided by Microsoft for Excel macros and VBA for detailed explanations, examples, and references to various functions and objects you can use in your macros.

2. Online Tutorials and Courses: Explore online tutorials and courses on VBA programming and Excel macros. Websites like Microsoft Learn, YouTube, Udemy, and Coursera offer courses ranging from beginner to advanced levels.

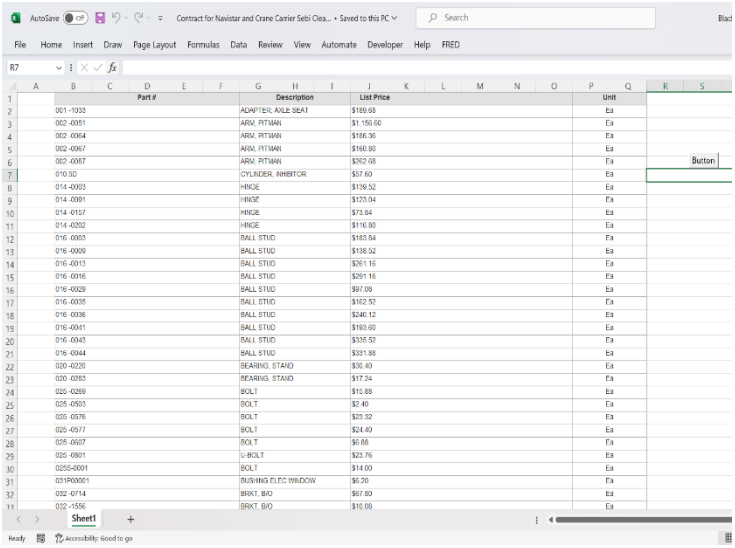
3. Forums and Communities: Join online forums, communities, and websites like Stack Overflow, MrExcel, and Reddit's r/excel.

4. Books and eBooks: Look for books and eBooks on Excel VBA programming. There are many comprehensive resources available that cover various aspects of VBA programming and macro development.

4.4.5 How Do You Test Your Code

1. Identify cleaning requirements:

Before testing, ensure you have a clear understanding of what "cleaning" means for your Excel files. This could include removing duplicates, formatting cells, standardizing data formats, etc.

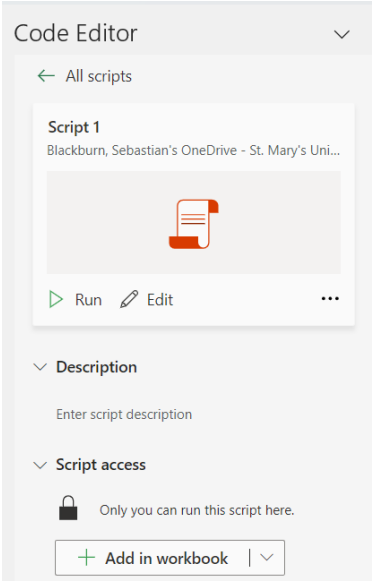


2. Create a sample Excel file:

Generate or collect a few Excel files that represent different states of data cleanliness. Include distinct types of data, such as text, numbers, dates, and special characters.

3. Record or write your macro:

If you haven't already, create the VBA macro to clean the Excel files. This might involve writing code to loop through rows and columns, apply formatting, remove duplicates, etc.



4. Test with sample data:

Run your macro on the sample Excel files. Make sure to test each aspect of your cleaning process to ensure it works as expected. Pay attention to edge cases and unexpected data formats.

5. Verify results:

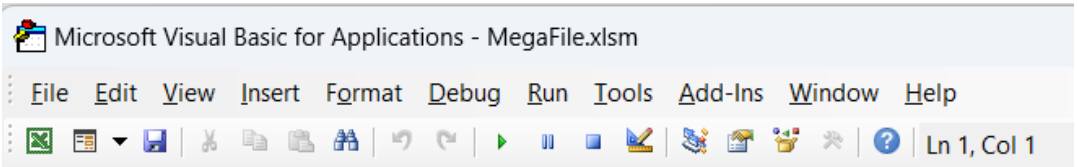
After running the macro, manually inspect the output to ensure that the data has been cleaned according to your expectations. Check for errors or inconsistencies.

6. Iterate and refine:

If you encounter any issues during testing, debug your macro code and make necessary adjustments. Run the tests again until you're satisfied with the results.

7. Automate testing:

If your macro code will be used frequently or on large datasets, consider automating the testing process using tools like Excel VBA's built-in testing features or external testing frameworks.



8. Document your process:

Once you're confident in your macro's ability to clean Excel files, document the testing process and any known limitations or edge cases. This documentation will be helpful for future maintenance or for other users who may interact with your macro.

4.4.6 Where Can You Access a Macro's Code?

To view the code for a macro in Excel, you need to access the Visual Basic for Applications (VBA) editor. Here's how you can do it:

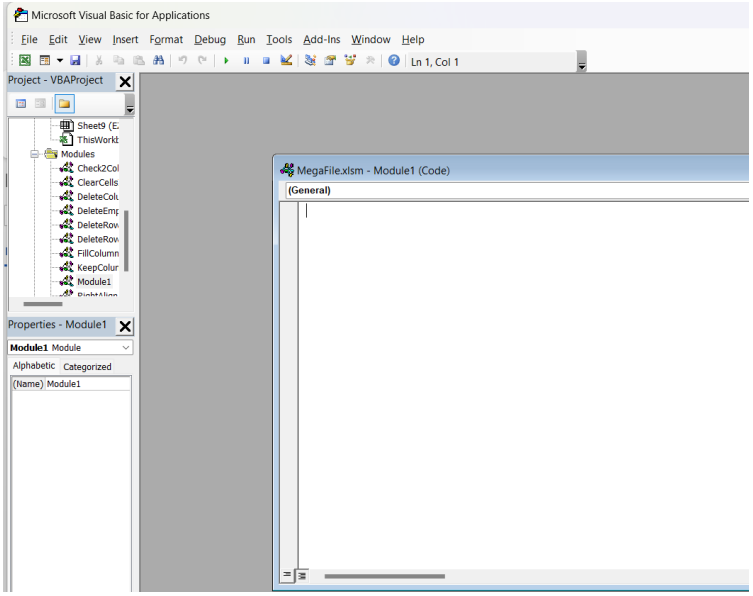
1. Press `Alt + F11` on your keyboard. This keyboard shortcut opens the VBA editor.

2. Alternatively, you can go to the "Developer" tab on the Excel ribbon. If you don't see the "Developer" tab, you may need to enable it in Excel options. Once the "Developer" tab is visible, click on it, and then click on the "Visual Basic" button.

3. In the VBA editor, you'll see a Project Explorer window on the left side. This window displays a list of all open workbooks and their corresponding VBA projects.

4. Find and expand the project for the workbook containing the macro you want to view. It's usually named after the workbook itself.

The Alt key and F11 Key



Screenshot showing the VBA Editor

5. Within the workbook's project, you'll find a folder named "Modules" or "Modules" under "Microsoft Excel Objects". Double-click on the module containing the macro code you want to view.

6. The module window will open, displaying the VBA code for the macro. You can scroll through the code, make edits if necessary, and save your changes.

By following these steps, you can easily view the code for a macro in Excel using the VBA editor.

```
(General)                                   (Declarations)

Sub DeleteRowsIfNotAllColumnsOccupied()
    Dim ws As Worksheet
    Dim lastRow As Long
    Dim i As Long
    Dim occupiedColumnsCount As Integer

    ' Change "Sheet1" to your actual sheet name
    Set ws = ThisWorkbook.Sheets("Test")

    Application.ScreenUpdating = False ' Turn off screen updating for better performance

    ' Find the last row with data in column 1
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Loop through rows backwards to safely delete rows
    For i = lastRow To 1 Step -1
        ' Reset the count of occupied columns for each row
        occupiedColumnsCount = 0

        ' Count the number of occupied columns (1, 2, 3, and 4)
        If ws.Cells(i, 1).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 2).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 3).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1
        If ws.Cells(i, 4).Value <> "" Then occupiedColumnsCount = occupiedColumnsCount + 1

        ' Delete the row if not all 4 columns are occupied
        If occupiedColumnsCount < 4 Then
            ws.Rows(i).Delete
        End If
    Next i

    ' Align all columns to the left
    For j = 1 To 4 ' Assuming you want to align the first 4 columns
        ws.Columns(j).HorizontalAlignment = xlLeft
    Next j

    Application.ScreenUpdating = True ' Turn screen updating back on

    MsgBox "Rows with less than 4 occupied columns have been deleted.", vbInformation
```

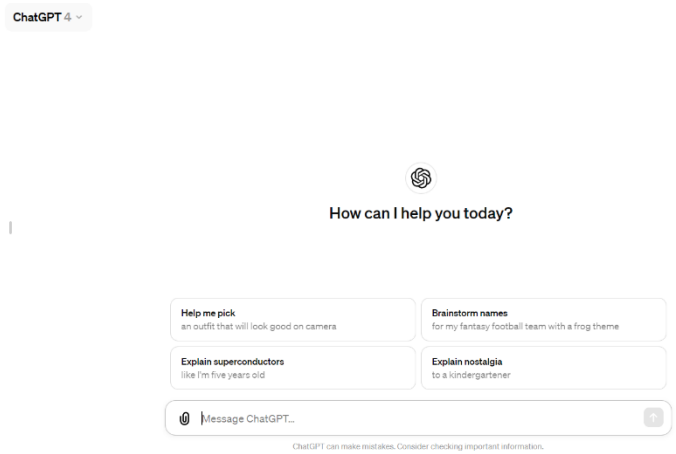
What you see above is an example of the code behind a macro, this one deletes a row if a relevant column is empty.

5 Querying a Database

5.1 Chat GPT 4.0 Component

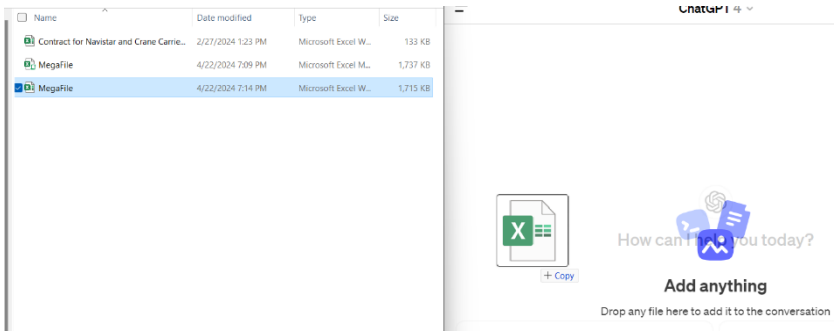
5.1.1 How Do You Query Contracts From Chat GPT 4.0?

Here is a step-by-step method on how to query the database through Chat GPT 4.0:



• Access the Chat GPT 4.0 platform through your browser.

• Upload the cleaned contract files to the platform.



• Formulate a query specifying the desired item details or SKU requirements.

• Initiate the query process by submitting the query to Chat GPT 4.0.

• Wait for the AI system to analyze the contract and generate a response.

You

Thank you, can you identify the how information is stored in the Navistar and Crane sheet?

ChatGPT

The "Navistar And Crane" sheet in your workbook is organized with the following columns:

1. **Part #** - This column contains part numbers for various items.
2. **List Price** - This column shows the list price for each part.
3. **Supplier** - This column indicates the supplier, which is "Navistar and Crane" for the rows listed.

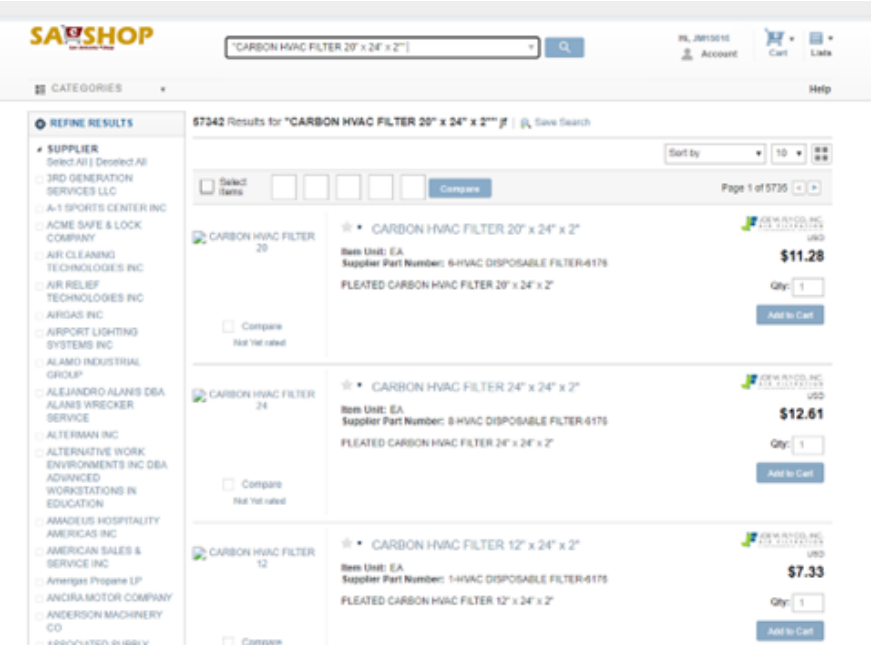
• Review the response provided by Chat GPT 4.0, which will include SKU(s) for relevant items.

• Evaluate the suitability of the suggested SKU(s) based on requirements.

• Use the provided SKU(s) for procurement purposes, referencing them in the search on SAShop

• Repeat the process for additional contracts or queries as necessary.

• Maintain documentation of queries and responses for record-keeping and future reference.

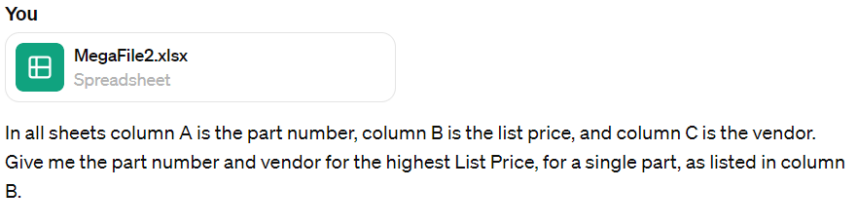


5.1.2 How Should the Interaction With Chat GPT 4.0 Be Framed

When interacting with Chat GPT 4.0, you may need to start the interaction by framing the environment.

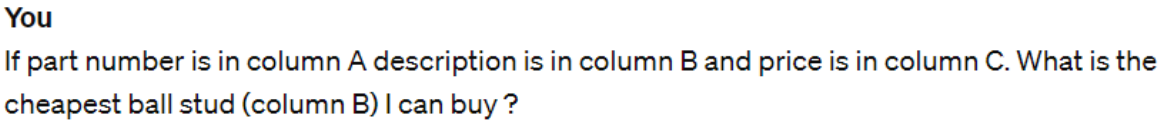
When you ask about the contracts in the Excel file, it is not going to know the formatting immediately. To speed up the process, you can frame your questions before you ask them with this opener:

"In all sheets column A is the part number, column B is the list price, and column C is the supplier."



Then you can ask for the part specifically following this style:

"Give me the (part number, vendor/supplier, or list price) and (part number, excel/supplier, or list price) for the (highest or lowest) List Price, for a single part, as listed in column B."



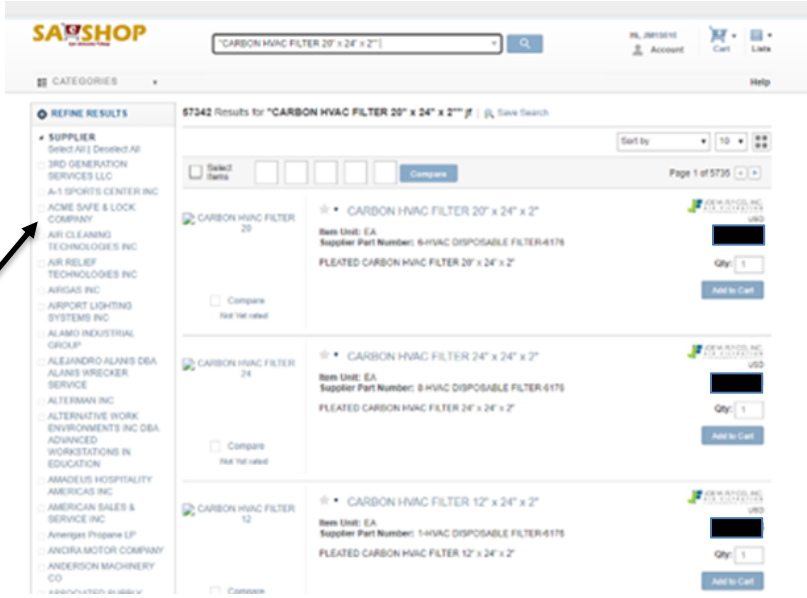
It sounds like a mouthful, but the key takeaway is to be very specific about what you are looking for.

5.2 Vroozi and SAShop

This solution is meant to allow Chat GPT 4.0 to give a user the supplier, the part number, and the listed unit price of any item that is requested, should it be in the Excel workbook containing all the converted and reformatted contracts.

While Vroozi is the software behind the scenes, SAShop is where you'll be interacting and doing your search.

In SAShop, when you make a search, you will see the suppliers listed along the left. You can search for matches from a group of suppliers, down to one. This ability is what we're capitalizing on, and why the Excel file contains the supplier's name.



6 Troubleshooting

6.1 Who Should Have the Contracts

If you desire to add more contracts to the MEGAFILE.xlsm file, you must first make sure that you have access and clearance to work with it.

If you do, then the next step would be to contact the San Antonio International Airport's Procurement team to ask for more contracts.

The point of contact at the procurement team was Mr. Jordan Mergele during the project.

6.2 What if PDFtoExcel.com Fails to Convert?

If PDFtoExcel.com does not wish to convert the PDF contract you give it, do not panic and give up.

While Chat GPT does need the file in Excel format, there is another method you can use to get the part without it.

Computers come equipped with a series of commands that use different key combinations. The combination we're using is 'ctrl' and 'f.' the Ctrl + F command adds a search bar to your page or file and allows you to perform a search by hand.

This search, as shown on the right, highlights the search word, and keeps track of how many times it shows up. You can even use this to look for multiple part numbers in this way.

By using this method, you may not be able to use Chat GPT to conveniently get your answers, but you have the file and now the means to find it without hunting through Vroozi.

For example, this contract was not able to be converted properly, and so had to be

City of San Antonio
Bid Tabulation

Opened:	March 2, 2018
For:	Formal Annual Contract for Captivel Fiat Chrysler Parts

done some other way, paying for a premium converter (Adobe Acrobat)

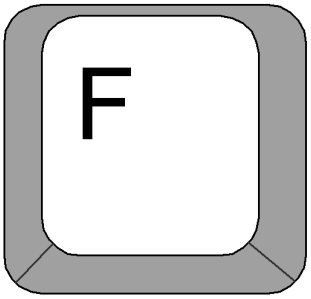
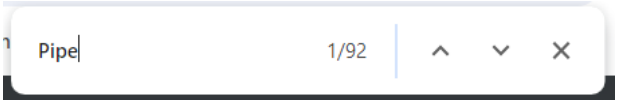


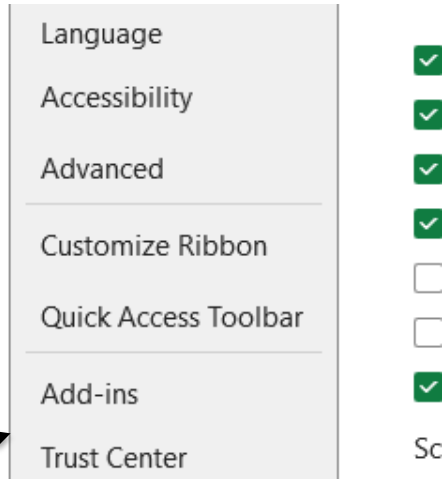
Image of both keys, for clarity.



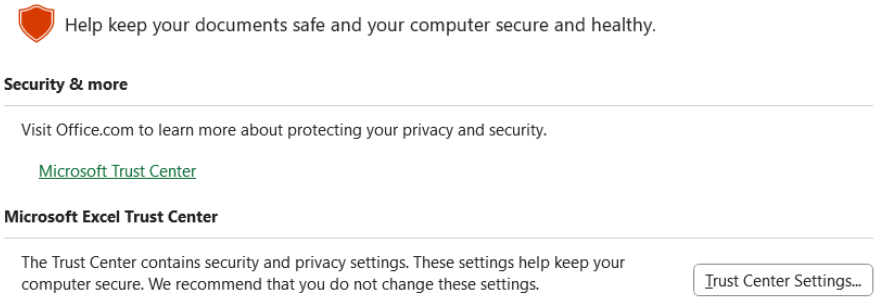
Ctrl + F in action, using an example search of 'Pipe,' the search bar is above and the scroll bar, that bar on the side of your page that you can scroll a large page up and down, is on the right. Note the highlighted areas: that's where the command found matches. Also, this search is inclusive, you can hit stuff that's not relevant, but 92 potential irrelevant matches is better than 1000 pages full of them.

6.3 Macro Troubleshooting

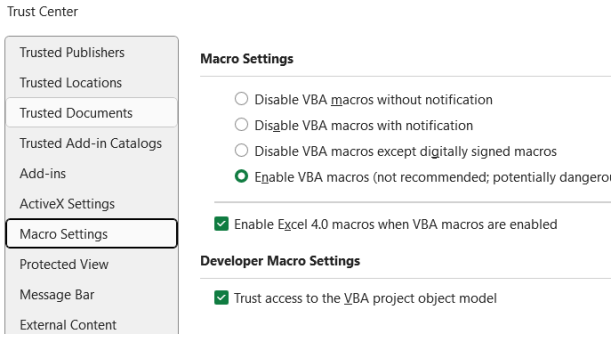
6.3.1 What If Macros Don't Appear?



1. Check Macro Security Settings: Ensure that macro security settings in Excel are not set to disable all macros. Go to the "File" tab, select "Options," choose "Trust Center," and click on "Trust Center Settings." Adjust the settings under "Macro Settings" to enable macros.



2. Verify Macro-Enabled Workbook: Ensure that the Excel workbook you're using is macro-enabled (XLSM format). Macros cannot be stored in standard Excel files (XLSX). If your workbook isn't macro-enabled, save it as a XLSM file format.



3. Review Macro-Containing Workbooks: Verify that you're looking within the correct workbook for the macros. Check all open workbooks by navigating to the "View" tab and selecting "Switch Windows" to switch between open workbooks.

4. Accessing the VBA Editor: If you've confirmed that the workbook is macro-enabled but still can't find the macros, access the Visual Basic for Applications (VBA) editor. Press `Alt + F11` on your keyboard or go to the "Developer" tab and click on "Visual Basic" to open the VBA editor.

5. Navigate to Macro Module: In the VBA editor, locate the project explorer window on the left side. Expand the project for your workbook and navigate to the "Modules" folder. Double-click on each module to view the macro code.

6. Seek External Assistance: If you're still unable to find the macros or encounter difficulties, consider seeking assistance from knowledgeable colleagues, online forums, or professional support services for Excel and VBA programming.

7. Ensure Macro Recording: If you're trying to find recorded macros, ensure that you've recorded the necessary actions using the "Record Macro" feature. Recorded macros won't exist unless they've been explicitly recorded by the user.

6.3.2 Used the Wrong Macro/ Macro Not Doing Anything?

If you find yourself having trouble with a tested macro, it may not be obvious what to do. However, this section will attempt to guide you.

If the problem stems from the macro not doing anything once it had been selected, it may be the result of the macro running but not finding anything to act on. This could the result of hitting the run macro button multiple times, but simply put: If nothing happens, then what it is going to do, it has already done.

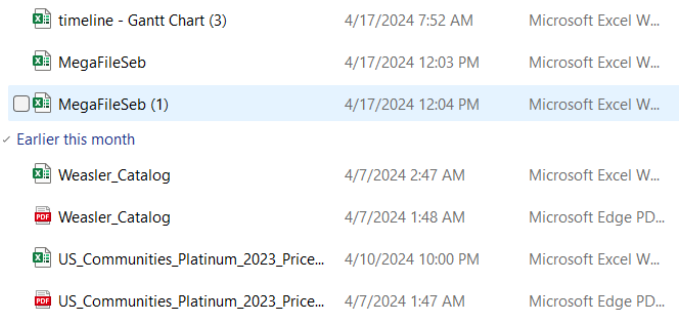
	A	B	C	D
1	Part #	List Price	Navistar and Crane	
2	001 -1033	$189.68	Navistar and Crane	
3	002 -0051	$1,156.60	Navistar and Crane	
4	002 -0064	$186.36	Navistar and Crane	
5	002 -0067	$160.80	Navistar and Crane	
6	002 -0087	$262.68	Navistar and Crane	
7	010.5D	$57.60	Navistar and Crane	
8	014 -0003	$139.52	Navistar and Crane	
9	014 -0091	$123.04	Navistar and Crane	
10	014 -0157	$73.84	Navistar and Crane	
11	014 -0202	$116.80	Navistar and Crane	
12	016 -0003	$183.84	Navistar and Crane	
13	016 -0009	$138.52	Navistar and Crane	
14	016 -0013	$261.16	Navistar and Crane	
15	016 -0016	$291.16	Navistar and Crane	
16	016 -0029	$97.08	Navistar and Crane	
17	016 -0035	$162.52	Navistar and Crane	
18	016 -0036	$240.12	Navistar and Crane	
19	016 -0041	$193.60	Navistar and Crane	
20	016 -0043	$335.52	Navistar and Crane	
21	016 -0044	$331.88	Navistar and Crane	
22	020 -0220	$30.40	Navistar and Crane	

Let's say the data in this table gets corrupted and it needs to be replaced.

26	025 -0576	$23.32	Navistar and Crane	
27	025 -0577	$24.40	Navistar and Crane	
28	025 -0607	$6.88	Navistar and Crane	
29	025 -0801	$23.76	Navistar and Crane	
30	025C -0001	$44.00	Navistar and Crane	

If the problems stem from using an incorrect macro, and it ends up acting in a way that is detrimental to the formatting of the Excel sheet, then there is a mix of good and bad news. Unfortunately, macros seem to be irreversible. However, all is not lost, if you have the files in different places.

So long as you have access to the original file, nothing will be permanently gone, but you may end up reformatting the file so you can put it back.



Point is: If you fear a mistake, make a spare save.

6.4 Chat GPT 4.0 Troubleshooting

6.4.1 What If Chat GPT 4.0 Can't Analyze the Contracts?

There are several reasons that Chat GPT may not be able to analyze the contract files. The most common was found during the process of figuring out how to format the compilation file.

Thankfully, it will tell you what the problem might be, so fixing it is an option.

If it's from the question itself, it could mean that either the question that Chat GPT was asked confused it, where it will tell you that it doesn't understand – easy fix, and it should guide you.

The recommendation here is to go to the subsection 'How You Should Format Your Questions' (pg. 40) to check what kind of information should be included in your question.

For any mishaps in Excel, The section 'Cleaning Contract' (pg. 22) contains a lot of information to guide for how to

ChatGPT

Error analyzing

I apologize once again for the difficulties. It appears that I am unable to process the request to find rows containing the keyword 'ELBOW' in Column G of your Excel file due to persistent technical issues.

If there's any other way I can assist you, or if you have different questions or tasks, please let me know. I'm here to help with any other information or analysis you might need.

It appears there was an issue with the data structure. The column name 'Description' might be different or not aligned with the expected format. I'll re-examine the data to identify the correct column names and then proceed with finding the cheapest ball stud. Let's take a closer look at the data structure.

The structure of the data appears to be slightly different than initially assumed. The first few rows seem to be headers or title rows. The actual data, including part numbers, descriptions, and prices, starts from the third row onwards. I will adjust the data processing to account for this and then search for the cheapest ball stud. Let's proceed with the corrected approach.

ChatGPT

Error analyzing

It seems there was an error in processing the request. Let me try again to retrieve the part numbers for ball studs that are priced between $100 and $150, considering the updated column order.

The part number for the ball stud that is priced between $100 and $150 is "016 -0009".

A sample of the various problems while learning the ideal format for the contract files.

You

MegaFile.xlsm
File

So if I was to give you this file, can you analyze it?

ChatGPT

I can't directly open or analyze files like your Excel workbook. However, you can describe the data or specific issues you're encountering, and I can help you figure out solutions or analyze data conceptually. If you can share details about the contents or any specific calculations or macros you need help with, I'll do my best to assist you!

Here shows that the type of file is important. When you save the file, it needs to be a copy in XLSX, or Chat GPT 4.0 won't recognize it.

6.4.2 Does the Chat Log Save?

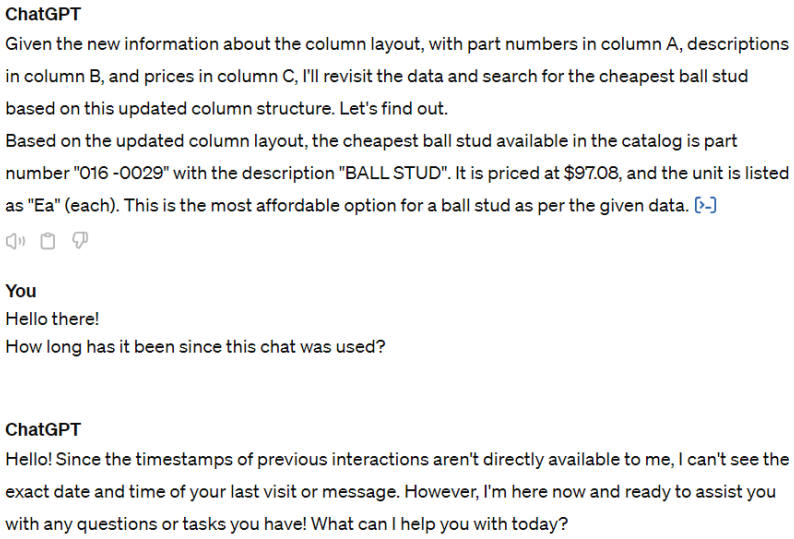
. Save Your Progress: If you're in the middle of a task or discussion, make sure to save any relevant information or notes you have. You can save the conversation transcript or take screenshots if needed.

2. Try Reconnecting: Wait a few moments and try refreshing the page or reconnecting to the internet. In some cases, the interruption may be temporary, and you can resume our conversation once the connection is restored.

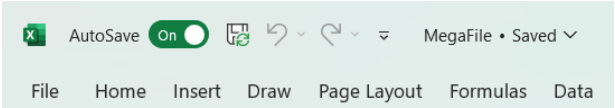
3. Check Back Later: If the interruption persists in the short term, you can check back later to continue your conversation.

4. Explore Resources: In the meantime, you can explore other resources or documentation related to your topic of interest. You might find helpful information or solutions that could address your questions or concerns.

5. Seek Assistance Elsewhere: If you have an urgent issue or need immediate help, consider reaching out to relevant support channels or forums where you can receive assistance from other users or experts in the field.



Chat GPT 4.0 may not have timestamps, but this chat was started in March 2024, and can still be used later (This image is from April 2024).



Automatic and manual saving is critical to safely keeping changes logged in a file.

--End of Document--

Appendix B: Gantt Chart

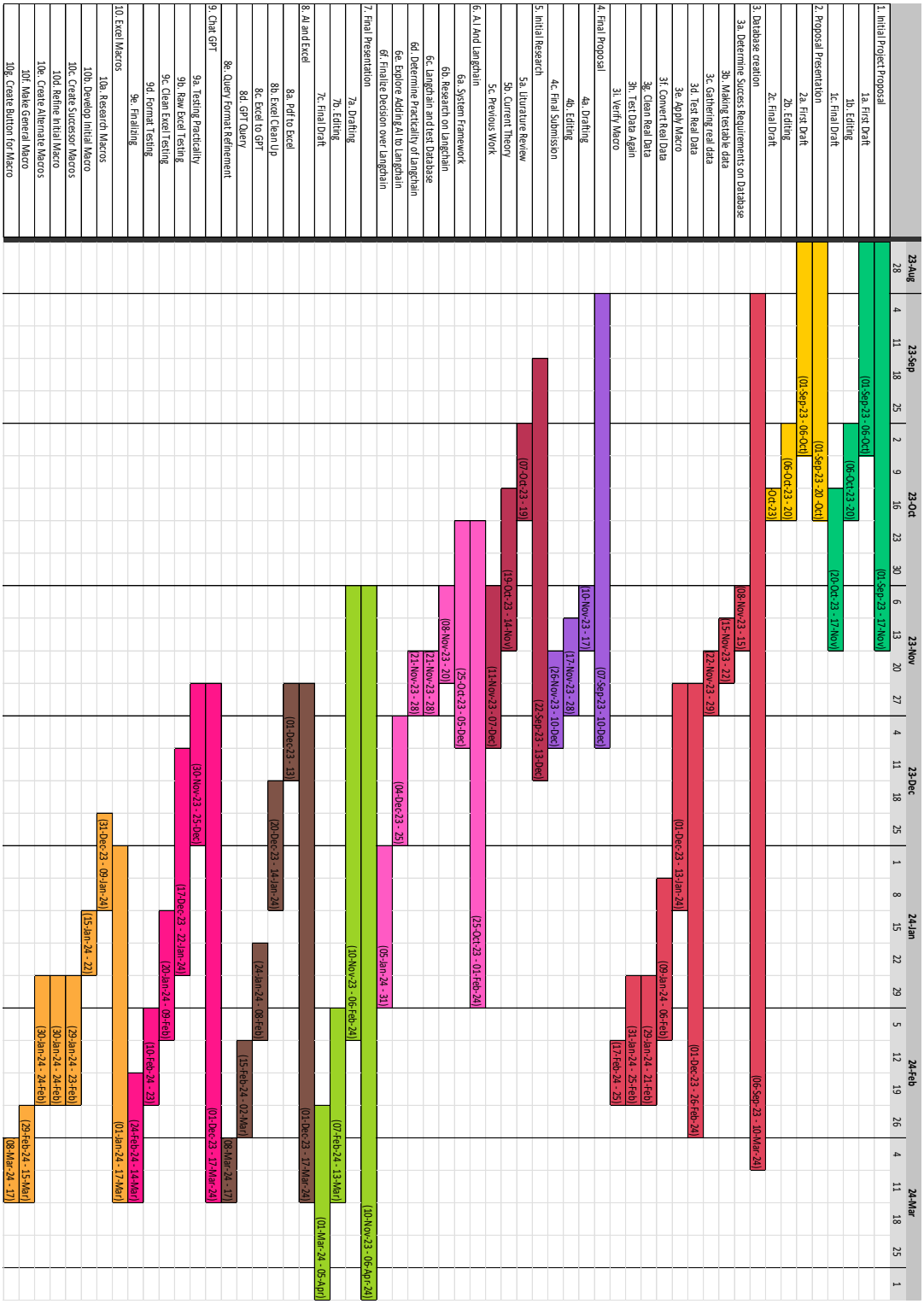


Figure 1: Gantt Chart estimating the workflow of the project.

The figure above is the most recent Gantt chart that was in use by the senior design team. Each part of the project was broken up into smaller sections that would have their own individual deadlines and timetables. This allowed a certain flexibility and reliability in that the team could always refer to what the next step was in the project, but also what the next step should be inside each part of the project.

Appendix C: Team Roles

The role of both Anthony Sadler and Sebastian Blackburn as the Senior design team members for the San Antonio International Airport procurement project was to provide a web-based methodology the procurement team can use for finding SKUs in their contracts, acting with Vroozi, their contract aggregation system, to function as a shortcut. The Senior design team assigned tasks, as displayed on the Gantt Chart (Figure 1.), to either Anthony or Sebastian based on the relevant skills, interest, and time constraints found during the project.

Appendix D: Weekly Logs

The weekly logs included here were a form of summary that allowed us to display the documentation of our accomplishments and stumbling blocks throughout the two-semester project.

11-13-23 I had a meeting to see where we were at and what matter was most urgent to address. Neither one of us had made considerable progress on setting up Lang Chain so we agreed to go to Dr. Aktunc's office on Tuesday to see if he could point us in the right direction.

11-14-23 Sebastian and I went to Dr. Aktunc's office and asked if he was familiar with the LangChain programming setup. He was not. We still looked at the installation process, he helped us, but we did not get it installed. He said that Dr. Torre or Dr. Luo would be more of a help than he was.

11-15-23 We had a meeting with Dr. Torre. I asked if he was familiar with LangChain and if he knew how to use it. He was not familiar with it and did not know how to use it. We were going through the installation documentation, it was complex. We asked Chat GPT 4.0 how to feed it Excel files that it can look through. It told us a method and we tried it on my account but to no avail as I do not have the "advance data analysis" feature on my account (I had the free version GPT 3.5). The paid version of Chat GPT 4.0 was GPT 4.0, which Dr. Torre had. We tried the process on his account, and it worked. Chat GPT 4.0 was able to query information from the Excel file. This was an enormous success. We talked about how Sebastian and I were just using Chat GPT 4.0 for means of communication and not actually training anything. This was a concern I also had. The project seemed quite simple, especially after this meeting with Dr. Torre. I asked if there was a way to make the project more complex. He looked at real contracts the airport uses and said something would need to be predicted from compiling all the data.

11-15-23 We had a meeting with Dr. Mehdi. We told him about the progress we made with making these contracts GPT readable. He was glad to hear it. I explained that the project seemed simple. Especially after we found that there was not as much of a need for the use of LangChain as we had previously thought. I believed that he was receptive to us saying that there might have been a need for an extra layer of complexity for the project. I asked him if there was

something that could be predicted. He said that was a very good question and we would need to think about it. Afterward, he said that this would be a great conversation to have with the rest of the procurement team at the airport on the Thursday we came back from Thanksgiving break.

11-20-23 We explored how to convert PDF to XLS files. There were many ways to do it that were also free such as IlovePDF.com and PDFtoExcel.com format the XLS file in different ways. Both of which were not uniform. What we needed was a uniform result so that when read the machine gave accurate information.

11-21-23 We used the free version of adobe acrobat to convert the PDF version of the "Heavy Truck Contract" (as attached to this submission) to XLM. This process had been streamlined unlike the free versions and provided a uniform version of the contracts. We had to adjust a few cells by hand.

11-27-23 We were supposed to have a meeting on 11-28-23 but it got canceled as the scheduled time did not work for Mr. Mehdi and the Senior design team alike. Mr. Mehdi reiterated that he would have his team convert the PDF contract to XLS and have them ready for the next in person meeting.

11-28-23 Sebastian and I communicated the next meeting between the team and Mr. Mehdi. Due to unforeseen scheduling conflicts, the original meeting for this week was canceled and we had prepared to meet over the break.

12-15-23 Anthony and Sebastian had our first brainstorming session on how to improve macro editability. To clean the Big Truck contract, it took five macros. It would be ideal to cut this down.

1-3-24 Anthony and Sebastian met to discuss how to streamline the process of converting the PDF files to an Excel format. It was determined that Sebastian would convert then using PDFtoExcell.com. The converted contracts would then be uploaded to a central database that is shared between Anthony and Sebastian.

1-5-24 Anthony and Sebastian convened to talk about what is the next step in doing work for the macros as well as the poster that Dr. Moras requested. Sebastian decided to pick this up for the team.

1-15-24 Anthony is working on the macros such that the 'LargeTruck' contract can be edited by Excel alone.

1-24-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

1-31-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

2-3-24 Anthony talked to Mr. Syed to find out which macros would be the best to clean the contracts. It was determined that the macros that required the least number of clicks would be the best. A button would be ideal.

2-5-24 Sebastian made button for macros.

2-7-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

2-14-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

2-21-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

2-28-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

3-06-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

3-13-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

3-20-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

3-27-24 Had a meeting with Dr. Moras and Dr. Bob to discuss the progress of the project.

4-10-24 Team had an internal meeting to discuss revisions left by Dr. Moras

4-11-24 Team had a meeting with Mr. Mehdi to discuss current developments.

4-12-24 Team had a meeting with Dr. Moras and Dr. Bob

Personal Reflections

Sebastian Blackburn

When I think of my time at St. Mary's University, I can't help but think of the Hero's Journey, the cyclic nature of maturing that has been used in story telling for aeons, and exemplified in stories like the Odyssey, Hamlet, and Star Wars. Before I started school, I was directionless, looking for a path to follow but not sure what to do. I got that call to adventure when my SAT scores came in and my grades were enough to get into a good college, all I needed to do was apply. But I refused that call, instead trying to put the cart before the horse and figure out how my life should go before knowing what my life even was. Despite that, I had applied, in the nick of time, to St. Mary's University, and I was committed to the path of the Mechanical Engineer. I had many mentors to help me along my path, the most present being God and my family. My parents kept me on the straight and narrow, and God kept me moving, but they weren't all the help I had. One of my old schoolteachers came by and wished me luck when I graduated, someone who had been my primary educator for years, and it had hurt to have to say goodbye. My old scoutmaster even sent me off, and he had helped me gain my

Eagle Scout Rank; an achievement I carry proudly still in my wallet, a token of 8 years of dedication.

I would earn many small things through the years, from tokens to prizes, to treasures. They would motivate me, restore my strength, calm me, teach me. One of the bigger treasures I have today is my 3D Printer. I've long since desired one, but I had only just gotten it this past year (2023), and when I did, it was a mark; a sign of things to come. I would place long prints on it, and my day would be seemingly determined by my actions and the behavior of the printer, as if God was watching and influencing me through it. If I was having a good day or behaving well, the print went well too, and vice versa. I crossed into the unknown, onto campus, in my sophomore year, not my freshman year due to the school being locked down. I moved onto campus, joined clubs, and was surprised at the school experience. St. Mary's was like this whole other world to me, fantastical yet down to earth. She wasn't scared to challenge me, and each time she did I rose to it, though with more than several close shaves.

I went into the Belly of the Whale, so to speak, in my junior year. It was the year I tried to be practical, logical, and I ended up falling on my face more than I care to admit. School became scary, I failed my first, and only, class ever. It wasn't the end of the world, but it made school real, that illusion of certainty and security of becoming an engineer was gone. I needed guidance more than ever, and I ended up having to find it on my own. I eventually did, of course, as all stories tend to go, and I came into senior year not quite ready, but willing, to face my road of trials.

Having just gotten out of junior year, I had to face the idea that I didn't know everything, that failure was merely another path to learn despite the punishing nature placed on it, and to

have patience for things to come. I had to face myself, who I was, and who I was becoming. I had to let parts of my past fade, even if it was by my own hand; I couldn't stay in school and do many of the things I used to do. And some of it was ugly, I had to face my own dark reflection, my own failings, and in a sense, I had to face my own trauma that I never really understood or known, it just never really came to light until it blocked my path, a stone I set in front of me and now I have to remove. I faced 2 major trials so far in senior year, pulling myself out of the dumps that were so severe, it wasn't certain if I would recover, and fighting myself to keep pushing and try new things and new perspectives instead of stagnating and then failing permanently. I don't imagine that I've faced my third, or even my last, for a long while, but I am determined to keep going, always.

God kept me going, and He helped me out of my barriers, my comfort zone. Fundamentally changing how I interact with the world, and how I interact with others, seem to be a reflection of how I interact with my faith, and it was no small feat. Through the past 4 years, I've had to rediscover my faith. It was never lost, but more so in a stagnant sense of security. I thought I knew all I needed to know and didn't want to know more. But, in a lackadaisical sort of way, I became more exposed to different perspectives, different faiths, and all of a sudden, I wasn't sure. I couldn't point to a specific verse to act as evidence in an argument, I would consider skipping mass with more seriousness over time, though I never did with the intention of skipping. Rather, it was as if somehow, I was losing touch with the roots of my faith and allowing myself to get swept up in the needs of today. So, I began to read, and try to compensate. If I couldn't read, I'd pray, and if I couldn't pray, I'd listen. I can't say whether or not I became more in tune with my faith, but I certainly learned more about it, and just became

more sure that I was in the right direction. The concept of skipping mass became foreign to me, I would listen to music genres I've never listened to before, I had a different understanding of the Word, and I began to appreciate the Homily in mass.

As my faith grew, I had to interact with the world in a different way. I couldn't just detach and not care like I used to, but I couldn't allow myself to become so sensitive to the world that I couldn't act within in it. So, I reached out and grabbed whatever I could. Join and staff a Rattler's Awakening? Yes please; What about the Knights of Columbus? Of course; Or even, would I go to mass even if it prevented me from doing something else? In the end, yes, and with no regrets. And this meant I had to work with others differently than I used to.

To say that I was bad at socializing would be an understatement, but that is also the least interesting thing you can say about me, I would argue that I could pick up social skills rather easily and did. I slowly stood straighter, became more confident, had to fight ego and passion plenty of times. I had to grow mature, patient, wise, and that took time. I'm not going to sit here and say I'm wise, but I did have to learn some things that I can certainly share in doses here and there. But really, that's not where things seem to matter in terms of others, that lies in how I talk, how I walk, how I carry. I had to learn via trial by fire, how to remove ego from projects, and how to walk with my back straight, not in search of a fight, but rather to exude competence, so that I can better be a person that is worthy to wear the ring at graduation.

Now, when I look in the mirror, I can see someone who can talk to strangers, who's growing the ability to make small talk, to be ok with silence, to articulate the small but interesting part of daily life that people don't really seem to notice. I would compensate early on with music, and so music would keep me company. But you can't touch music, you can't convince it, and

you certainly won't win a fight against it. So, I had to learn to not force things, and to let go in small matters, which is an interesting experience and skill for me to learn, and one I am still attempting to master.

The world is such a vague concept, and I don't think many realize that. Because picture what you think the world is: It could be the planet, it could be your home, maybe your life or your work. I could even just be as simple as your town, because to any single person, what is the world? Is it something we can interact with? Something we can touch? Or is it something we're aware of, and in that case, is it crazy small, only what we notice in our day-to-day? And that's not something I can simply answer, but I can say for me, that my world can't afford to care about maybe's, things that are so far removed that it is merely a maybe that I can have an impact on it. My world was a very small place, it consisted mainly of my surroundings, and so I had no appreciation for things outside of it. But I learned that my world was smaller than my reach, and that my reach would only grow longer.

I decided my world should too. I began to explore what it could mean to broaden my horizons, to expand my world. To me, it meant to be more aware of my place in it, and that of the people around me. To me, my world became the thing that I interacted with, the people I came across. But it takes more than that to grow the size of one's world, I had to actually figure out how big of a world could I even handle. I ended up learning that the larger my world became, the more time I had to dedicate to it. Despite all the propaganda or just 'opening one's eye' or journeying 'beyond what you see,' I ended up learning to appreciate the small things in my world, and the smallness of my world. It's a lot bigger than it was, there is no doubt about

it, and it'll continue to grow, but I learned how important it is to stop and smell the roses, and just appreciate the things around me and the people that I'm on a journey with.

Anthony Sadler

Not everything is so serious, being lighthearted was especially hard for me to learn but though ups, downs, sideways and struggle I realized that one can only take life so seriously. In not too long, what I accomplished in this life will not matter, what I do on Earth is temporary and will not carry over to the next life.

Coming into St. Mary's as a quiet and timid freshman I was not sure what to expect. I did not want to ruffle the waters, not wanting to express my opinions in fear that others would not agree. I suppose being timid was many freshmen. But subconsciously, I genuinely wondered if I would ever break the timid mold. Was this behavior conducive to what I wanted out of life? This is what I would ask myself as an eighteen-year-old. Growing up in a single parent household times could be tough. But my mom always found a way to make it work. Always! Her perseverance to make sure that my brother and I always had the best opportunities is not something that I noticed as a child, and even my first couple years of college I failed to appreciate it. Today, I realize how much my mom sacrificed to make sure that my brother and I were out of trouble, fed, sheltered, and most importantly loved. Realizing the sacrifice my mom made me have a deep sense of obligation to pay back the favor. God has given me many blessings: a family, a STEM degree, a car, the capability to complete an engineering degree; it would be an absolute waste to not take full advantage of these gifts. Everyone bears hard times

but even in the most difficult of times there are blessings. I think there is always something to be learned from every situation, and especially in difficult times. I mention what I did about how much my mom sacrificed to say that my mission in life is to firstly, to provide for my family, and not just for my soon to be children and wife but also my parents, grandparents, and brother.

The struggle that my mom went through to provide is a debt that can never be repaid but I will make the rest of her life as comfortable as I can. A passive and timid personality, as I had as a freshman, would not allow me to achieve these goals. St. Mary's helped me break this mold. Though interactions with many different religions, languages, and cultures I have grown to see the similarities all humans have. We are not all so complex and different as I once thought; generally speaking, we all want to be safe and loved. My ability to relate to others has been greatly improved. I am more comfortable in the world than I was ever before. Allowing me to achieve at different heights than I would have with a timid personality.

Understanding the world more because of the diverse community at St. Mary's has given me a better insight to the human condition, which I was first introduced to in my sophomore year through our St. Mary's core curriculum (SMC) class - Self. Even with SMC Self the human condition was not something that could wrap my head around for a long while. Which is probably why did not do so well in SMC Self. I could not understand that while people suffer in different ways there is a fundamental or "baseline" suffering all humans experience. I guess this can be called "baseline" suffering, all humans no matter status, net worth, or color go through it. It is a suffering that as a human we are obliged to. It was only though my setback that I got a better understanding of the human condition.

My junior year I ran into health struggles that set me back a year. It was a very difficult couple of years but in retrospect, I would not trade it for anything (as if I had the option). Going though health troubles at a young age bothered me for a while. I could not understand why this would happen to me. In fact, I would think about it in terms of odds: what are the odds that out of all the people to be sidelined for a year due to health struggles it would be me? While unlikely, it was indeed me to be sidelined. This time sidelined gave me a lot of time to think about where I am going in life and once again made me think about what I want out of life, the same question I was asking myself when I was a freshman. This time of my life also came with a lot of physical and psychological pain, furthermore it felt like the world was passing on as I stood still. These two years of my life showed me what the human condition is. These years made me significantly more empathetic. I feel for those going through a tough time, especially a physically. The best way to relate to someone is if they can relate to you.

It was not until a few months ago from writing this, January '24, that the hook in the popular song "Love Yourz" by J. Cole truly resonated with me. The hook is as follows "beauty in the struggle and ugliness in the success." Deep in my soul I believe that there was a very specific reason that God allowed me to go through what I went through. There is a plan, maybe to make me mentally tough, change my mindset, sideline me from life to avoid trouble? I do not know and I may never know but there was a reason.

But with certainty I can say that Dr. Karlin, Dr. Mitchell, Dr. Turner and Mr. Cortina were supporting me though this difficult time. I failed a couple of classes and barely passed the others during this time although I always felt like they were rooting for me. The first day I was

at school not using crutches I remember Dr. Mitchell and Dr. Turner looking at me in a way that I find hard to describe. Without then saying anything specific I knew that they were happy for me. Dr. Turner said “Hey, Anthony” with a specific tone that I will not forget. It was a moment that will forever be with me, a moment I am thankful for. A combination of difficult times, the right people, and understanding the human condition have allowed me to better understand others and what really matters. I am forever thankful for St. Mary’s support through difficult times.

\St. Mary's changed how I see the world. Along with my experiences, the University taught me that life is a mix of good and bad, just like the hook "beauty in the struggle and ugliness in the success." This applies to all; it is absolute there is no such thing as a perfectly good life. In fact, it would be an absolute lame place to be without struggle. The world is like a big quilt with everyone's unique stories woven together. St. Mary's exposed me to different cultures, religions, and languages, showing me that despite our differences, we all want the same things - safety, love, and understanding. My time at St. Mary's helped me understand that everyone faces challenges, no matter their background. My own setbacks, like health issues in junior year, made me realize that suffering is something we all go through, no matter who we are.

Life has struggles, and success isn't perfect. I believe there's a reason for everything, even tough times. The support from St. Mary's, especially from the aforementioned faculty during my tough times, made me appreciate the connection between people. Their encouragement taught me what really matters in life. In the end, St. Mary's transformed my view of the world. I now appreciate the beauty in struggles and understand that success isn't flawless. I'm thankful for the lessons, friendships, and the new way of looking at the world that St. Mary's gave me.

Life is a mix of moments, each adding to the story called life. There is nothing I would change about my time at St. Mary's. This may seem like lazy thinking but I think not. Anthony would not be Anthony as he is known today if not having gone through the exact ups, sideways and downs that I did. Maybe not having gone through health problems would not have made me as resilient as I am today.

In the darkest corners of life where shadows linger and uncertainty reigns, I hold on to three core sentiments that I aspire to carry with me through the twists and turns for my future. The first is the belief that even within challenging situations, there exists a hidden beauty waiting to be discovered. Life will throw storms my way but I aim to find the blessing in the storm, to see the resilience and strength that emerge in adversity. The second sentiment is a strong belief in the enduring presence of hope. Regardless of how rough the seas of life may become, I choose to anchor myself in the belief that hope is a constant, a guiding light that persists even in the bleakest moments. It serves as a beacon, steering me through the darkest nights and pushing me forward with the assurance that better days are to come. The third sentiment that I hold dear is the unwavering assurance of God's love. In moments of doubt, when thoughts of uncertainty loom, I find peace in the knowledge that God's love is a constant force in my life. It provides me with the strength to weather storms, to stand resolute in my convictions, and to navigate the complex waters of life with grace. As I stand on the brink of graduation basking in accomplishment, I acknowledge that these sentiments are not mere joys associated with success. Instead, they are my deep-rooted beliefs that carried in my heart, convictions that I trust will endure even as life's journey takes me through highs and lows. I humbly implore God to be my anchor, to guide me through moments of doubt, and to fortify my spirit, enabling me

to uphold these sentiments with unwavering strength and conviction. I thank everyone that has helped me get this far, which includes those that may not think they had a part. It takes a village to get a personal to graduation, and a large village it took