**A Guide to Invoice Parsing and Analysis Using Pixtral-12B Model for OCR and RAG**

Invoice parsing and processing is a significant challenge for businesses of all sizes. In most cases, invoices lack a standardized format, which means that companies looking to streamline invoice handling from numerous vendors must build automated systems capable of interpreting a wide variety of layouts.

An effective invoice parsing system should reliably extract key details such as payment terms, totals, and item descriptions. Traditionally, businesses have relied on OCR (Optical Character Recognition) models to accomplish this, but these systems often struggle with inconsistent formatting, complex tables, and handwritten elements. Additionally, OCR models are prone to errors when handling poor image quality or non-text elements, which results in inaccuracies and requires manual correction.

This is where large vision models (LVMs), like OpenAI’s 4o, have shown a lot of promise. LVMs work by combining image recognition capabilities with natural language understanding, allowing them to process both visual and textual data in the same model. These models are trained on internet-scale datasets, which enables them to handle various invoice formats, including complex layouts that traditional OCR models struggle with.

Among the LVMs that have been released recently, Pixtral-12B model by Mistral AI stands out. It is an open model that excels in multimodal tasks, making it highly effective for invoice parsing scenarios. The model, which is around 24GB in size, builds on Mistral's text-focused Nemo 12B and integrates a vision adapter, which allows it to handle complex visual layouts such as tables, graphs, and embedded images within documents. It has been trained using a diverse range of image and text data, enabling it to generalize well across various document types and formats.

In this tutorial, we will walk you through the process of deploying and using Pixtral-12B, and apply it for invoice parsing tasks. We will also build a chat-based invoice analysis system, that can allow you to query multiple invoices at the same time.

Let’s get started!

## Understanding Pixtral-12B

Before we get started with the steps, let’s take a quick look at Pixtral-12B.

* **Multimodal Capabilities**: Pixtral-12B can process both text and images simultaneously, making it highly effective for tasks such as invoice parsing, document processing, and more.
* **12 Billion Parameters**: The model boasts 12 billion parameters. Its size allows it to handle complex and large-scale tasks, and offer superior performance compared to smaller models. However, it is small enough to be deployed on a single A100 GPU.
* **High-Resolution Image Processing**: It can process high-resolution images (up to 1024 x 1024) with a deep understanding of spatial relationships between elements such as tables, graphs, and embedded images.
* **Contextual Understanding**: Pixtral-12B is capable of understanding both textual and visual contexts within documents, enabling more accurate information extraction and parsing. This is what makes it a powerful candidate for invoice parsing.
* **Open-Source**: Available on platforms like GitHub and Hugging Face, the model can be fine-tuned and used for various purposes, with different licensing options for research and commercial applications.

These features make Pixtral-12B a robust solution for automating document workflows and complex multimodal tasks. In our tutorial, we will use it to process both computer-generated and hand-written invoices.

## **Step by Step Guide to Parse Invoices Using Pixtral-12B**

Let’s get started. Our stack will be:

* Pixtral-12B by Mistral AI
* Qdrant Vector Store
* Langchain framework

### **Step 1 - Prerequisites**

Our first step is to create a virtual environment, and then install the required libraries. We will assume that you have done so, and launched a Jupyter Notebook on your chosen cloud or your laptop.

| !pip install vllm !pip install --upgrade mistral\_common |
| --- |

Pixtral requires the mistral\_common library, so let’s install that.

Next, let’s import the modules.

| from vllm import LLM from vllm.sampling\_params import SamplingParams from dotenv import load\_dotenv import os import gradio as gr |
| --- |

What are the use of following imports ?

* from vllm import LLM  
  *Imports the LLM class for interacting with the VLLM language model.*
* from vllm.sampling\_params import SamplingParams  
  *Imports SamplingParams for configuring sampling options when generating text.*
* from dotenv import load\_dotenv  
  *Imports the load\_dotenv function to load environment variables from a .env file.*
* import os  
  *Imports the os module for interacting with the operating system, such as handling file paths.*
* import gradio as gr  
  *Imports Gradio as gr for creating user interfaces for machine learning models.*

Now let’s load the environment variables for each use case

| load\_dotenv() |
| --- |

To install Pixtral-12B locally, we will use vLLM. Also,Now let’s import the libraries.

| from vllm import LLM from vllm.sampling\_params import SamplingParams |
| --- |

### 

You will need an access token from Hugging Face (https://huggingface.co). Get that first, and then download the model in the following way:

| from huggingface\_hub import notebook\_login notebook\_login() llm=LLM(  model="mistral-community/pixtral-12b-240910",  tokenizer\_mode="mistral",  max\_model\_len=4000 ) |
| --- |

### **Step 2 - Context Extraction from the given image url**

Now let’s write a function that will invoke the Pixtral-12B model, with a prompt where we pass the image url. Yes, you can either directly pass the image url, or encode your image in Base64 format. Let’s do the former.

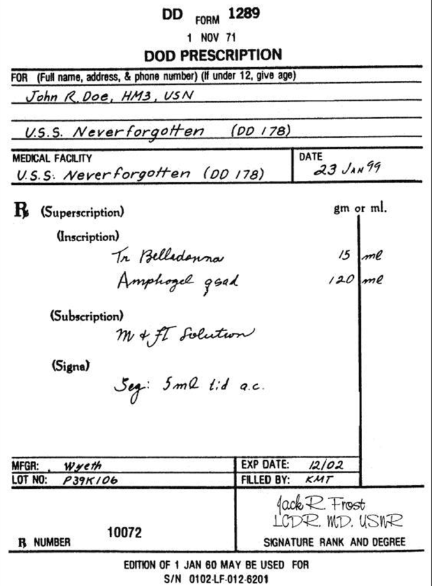
| def generate\_context(url):  model = "pixtral-12b-2409"    messages = [  {  "role": "user",  "content": [  {  "type": "text",  "text": "Extract the text from the image precisely, extract every text."  },  {  "type": "image\_url",  "image\_url": url  }  ]  }  ]    # Get the chat response  chat\_response = client.chat.complete(  model=model,  messages=messages  )    # return the context  return chat\_response.choices[0].message.content |
| --- |

That’s all that’s needed!

Now, let’s try this out with a few bill images.

### **Step 3 - Bill Parsing**

Here’s the first bill image we tried with.



**This is the extracted text using Pixtral-12B.**

| DOD FORM 1289 1 NOV 71  DOD PRESCRIPTION  FOR (Full name, address, & phone number) (If under 12, give age) John R Doe, HM3, USN  U.S.S. Neverforgotten (DD 178)  MEDICAL FACILITY U.S.S. Neverforgotten (DD 178) DATE 23 Jan 99  R (Superscription)  (Inscription)  Tm Belladonna 15 ml Amphogel goat 120 ml  (Subscription)  M & FT Solution  (Signs)  Sig: 5 ml t.d. a.c.  MFGR: Wyeth EXP DATE: 12/02 LOT NO: P39X186 FILLED BY: RWT Jack R Frost LCDR MD USNR  B NUMBER 10072 SIGNATURE RANK AND DEGREE  EDITION OF 1 JAN 60 MAY BE USED FOR S/N 0102 LF 012 8201 |
| --- |

**Extracted Json using Pixtral-12B:**

| {  "DD": "1289",  "Form": "1289",  "Date": "1 NOV 71",  "Title": "DOD PRESCRIPTION",  "Patient": {  "Name": "John R Doe",  "Military Rank": "HM3",  "Military Service": "USN"  },  "Facility": {  "Name": "U.S.S. Neverforgotten",  "Code": "DD 178"  },  "Medical Facility": {  "Name": "U.S.S. Neverforgotten",  "Code": "OO 178",  "Date": "23 Jan 99"  },  "Prescription": {  "Superscription": "B",  "Incription": [  "Tm Belledonna - 15 ml",  "Amphogel hors 120ml"  ],  "Subscription": "M + FT Solution",  "Signa": "Seq. 5 ml t.i.d a.c."  },  "Pharmacy Additional Info": {  "MFGR": "Wyeth",  "Lot No": "P39X106",  "Exp Date": "12/02",  "Filled By": "RMT"  },  "Prescriber": {  "Name": "Jack R Frost",  "Military Rank": "LCDR",  "Medical Degree": "MD",  "Military Service": "USNR",  "BN": "10072"  } } |
| --- |

As you can see, the extraction is quite accurate.

### **Step 4 - Analysis of Bills / Invoices Using LLM and build a QnA over the image**

We will now use the Pixtral-12B model to analyze the parsed data from the invoice under a JSON SCHEMA.

**What are we going to do?**

* **Build a content extractor model using Pixtral-12b which extracts the text from the given image and give the response in JSON format.**
* **On top of that JSON response, we will try to build a Query LLM with Pixtral-12b multimodal capabilities.**
* **Design an Gradio Interface over it best user experience**  .

**Why JSON formatting ?**

* JSON format of parsing helps the data to be structured in a appropriate machine understandable format and will help us somewhat skip the part of building a Vector database over it, and thus we can directly use that JSON data to query over the Image

| def generate\_context(image\_url, prompt = "Extract text from the image and give the response in JSON format"):  messages = [  {  "role": "user",  "content": [  {"type": "text", "text": prompt},  {"type": "image\_url", "image\_url": {"url": image\_url}}  ]  }  ]   outputs = llm.chat(  messages,  sampling\_params=SamplingParams(max\_tokens=8192)  )   return outputs[0].outputs[0].text |
| --- |

Function **generate\_context** deals with the part of extracting the text from the given image after parsing it and make it suitable for for the use case of querying over it.

It already has a default prompt given in case user doesn’t have the requirement of changing the prompt multiple times,

We provide a max token of 8192, which can serve our use case, if you want you can go for many other use max\_token size( in such cases do get your model run throughout the max\_prompt size given in max\_model\_len.

In the prompt we are gonna provide it our prompt and image url on which we are gonna run our extraction.

| def query\_llm(context,query):  messages = [  {  "role": "user",  "content": [  {"type": "text", "text": "You are an answer generation agent, you'll be given context and query, generate answer in human readable form"},  {"type": "text", "text": f"here is the question {query} and here is the context {context}"}  ]  }  ]   outputs = llm.chat(  messages,  sampling\_params=SamplingParams(max\_tokens=8192)  )   return outputs[0].outputs[0].text |
| --- |

Now we’ll use the multimodal capability of Pixtral-12b, i.e, it’s easy to use function with images and text,

We’ll be providing the JSON format context we got from the previous generate\_context function and the query provided by the user.

Pixtral-12b multimodal capabilities are gonna take care of all other things and provide us with the answer we are looking for in a very human readable format.

| import gradio as gr  def process\_query(url, query):  context = generate\_context(url)  response = query\_llm(context, query)  return response  if \_\_name\_\_ == "\_\_main\_\_":  # Create the Gradio interface  interface = gr.Interface(  fn=process\_query,  inputs=[  gr.Textbox(label="Enter the URL", placeholder="Enter image URL here"),  gr.Textbox(label="Enter your query", placeholder="Ask a question about the content")  ],  outputs=gr.Textbox(label="Response"),  title="Pixtral-12b RAG Application",  description="Provide an image URL and ask questions based on the context generated from it."  )   # Launch the interface  interface.launch(share = True) |
| --- |

Now about the gradio interface we are gonna build over it, firstly let’s install gradio

| !pip install -q gradio |
| --- |

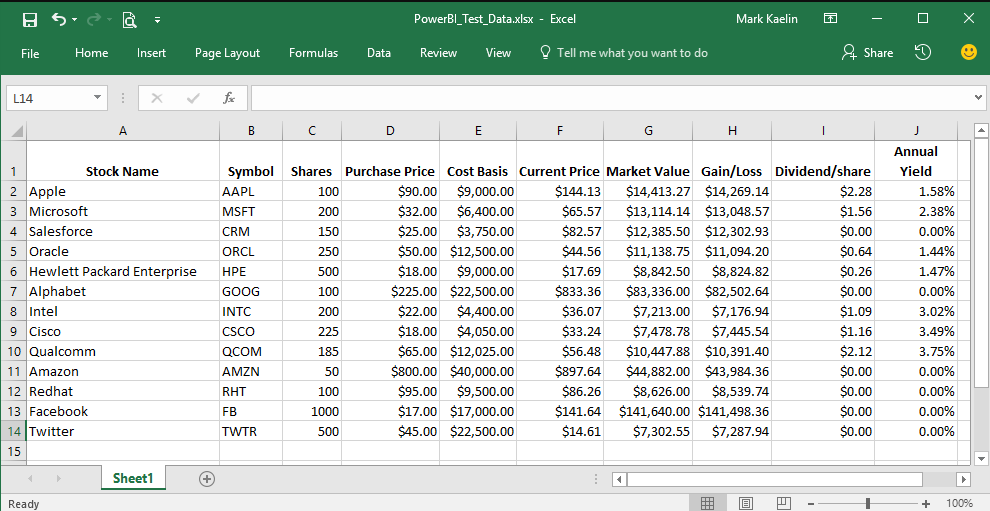
Tips 👍always use -q while installing something to prevent your screen getting filled will all the information displayed on your screen of the installation process.

Major component in Gradio Interface building 👍

* fn → refers to the function our input and output will refer for backend processing.
* Inputs →receives all the input we want and it’s customizable according to our needs, it’s values will be used in fn(backend) function.
* Outputs → we can customise it and response from fn will be the value of this component.
* Title →Title of the gradio interface
* Description →description of the interface

**Step 5 - Outputs :**

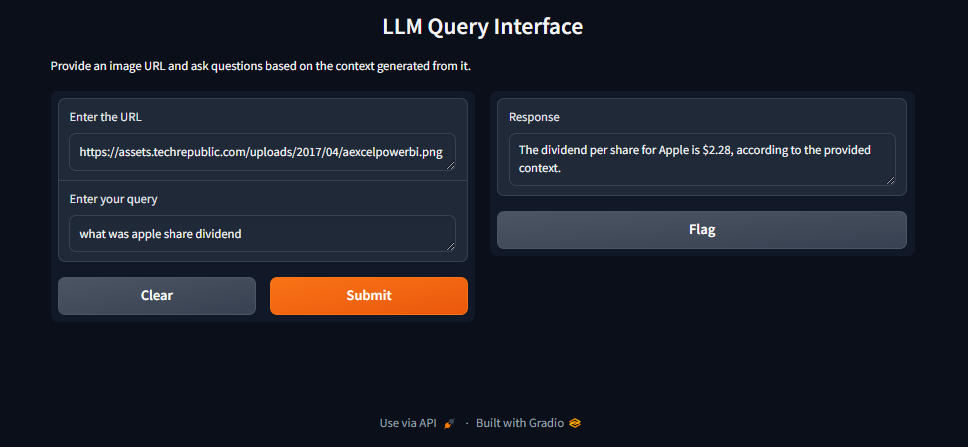
**BILL 1:**

****

**Json Text Extracted 👍**

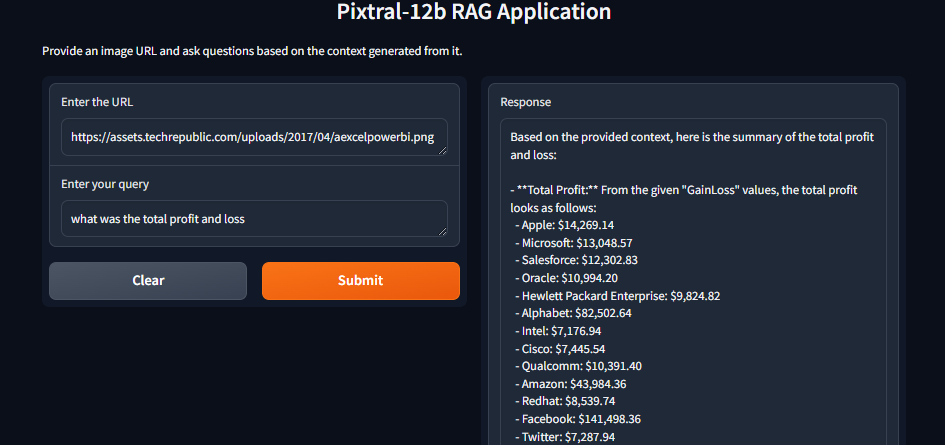
| {  "table": {  "header": [  "Stock Name",  "Symbol",  "Shares",  "Purchase Price",  "Cost Basis",  "Current Price",  "Market Value",  "Gain/Loss",  "Dividend/share",  "Yield"  ],  "rows": [  {  "Stock Name": "Apple",  "Symbol": "AAPL",  "Shares": 100,  "Purchase Price": "$90.00",  "Cost Basis": "$9,000.00",  "Current Price": "$144.13",  "Market Value": "$14,413.27",  "Gain/Loss": "$14,269.14",  "Dividend/share": "$2.28",  "Yield": "1.58%"  },  {  "Stock Name": "Microsoft",  "Symbol": "MSFT",  "Shares": 200,  "Purchase Price": "$62.00",  "Cost Basis": "$12,400.00",  "Current Price": "$64.57",  "Market Value": "$13,114.14",  "Gain/Loss": "$13,048.57",  "Dividend/share": "$1.56",  "Yield": "2.38%"  },  {  "Stock Name": "Salesforce",  "Symbol": "CRM",  "Shares": 150,  "Purchase Price": "$25.00",  "Cost Basis": "$3,750.00",  "Current Price": "$82.57",  "Market Value": "$12,385.50",  "Gain/Loss": "$12,302.83",  "Dividend/share": "$0.00",  "Yield": "0.00%"  },  {  "Stock Name": "Oracle",  "Symbol": "ORCL",  "Shares": 250,  "Purchase Price": "$50.00",  "Cost Basis": "$12,500.00",  "Current Price": "$44.56",  "Market Value": "$11,138.75",  "Gain/Loss": "$11,094.20",  "Dividend/share": "$0.64",  "Yield": "1.44%"  },  {  "Stock Name": "Hewlett Packard Enterprise",  "Symbol": "HPE",  "Shares": 500,  "Purchase Price": "$18.00",  "Cost Basis": "$9,000.00",  "Current Price": "$17.69",  "Market Value": "$8,842.50",  "Gain/Loss": "$8,824.82",  "Dividend/share": "$0.26",  "Yield": "1.47%"  },  {  "Stock Name": "Alphabet",  "Symbol": "GOOG",  "Shares": 100,  "Purchase Price": "$225.00",  "Cost Basis": "$22,500.00",  "Current Price": "$833.36",  "Market Value": "$83,336.00",  "Gain/Loss": "$82,502.64",  "Dividend/share": "$0.00",  "Yield": "0.00%"  },  {  "Stock Name": "Intel",  "Symbol": "INTC",  "Shares": 200,  "Purchase Price": "$22.00",  "Cost Basis": "$4,400.00",  "Current Price": "$36.07",  "Market Value": "$7,213.00",  "Gain/Loss": "$7,176.94",  "Dividend/share": "$1.09",  "Yield": "3.02%"  },  {  "Stock Name": "Cisco",  "Symbol": "CSCO",  "Shares": 225,  "Purchase Price": "$18.00",  "Cost Basis": "$4,050.00",  "Current Price": "$33.24",  "Market Value": "$7,478.78",  "Gain/Loss": "$7,445.54",  "Dividend/share": "$1.16",  "Yield": "3.49%"  },  {  "Stock Name": "Qualcomm",  "Symbol": "QCOM",  "Shares": 185,  "Purchase Price": "$65.00",  "Cost Basis": "$12,025.00",  "Current Price": "$56.48",  "Market Value": "$10,447.88",  "Gain/Loss": "$10,391.40",  "Dividend/share": "$2.12",  "Yield": "3.75%"  },  {  "Stock Name": "Amazon",  "Symbol": "AMZN",  "Shares": 50,  "Purchase Price": "$800.00",  "Cost Basis": "$40,000.00",  "Current Price": "$897.64",  "Market Value": "$44,882.00",  "Gain/Loss": "$43,984.36",  "Dividend/share": "$0.00",  "Yield": "0.00%"  },  {  "Stock Name": "Redhat",  "Symbol": "RHT",  "Shares": 100,  "Purchase Price": "$95.00",  "Cost Basis": "$9,500.00",  "Current Price": "$86.26",  "Market Value": "$8,626.00",  "Gain/Loss": "$8,539.74",  "Dividend/share": "$0.00",  "Yield": "0.00%"  },  {  "Stock Name": "Facebook",  "Symbol": "FB",  "Shares": 1000,  "Purchase Price": "$17.00",  "Cost Basis": "$17,000.00",  "Current Price": "$141.64",  "Market Value": "$141,640.00",  "Gain/Loss": "$141,498.36",  "Dividend/share": "$0.00",  "Yield": "0.00%"  },  {  "Stock Name": "Twitter",  "Symbol": "TWTR",  "Shares": 500,  "Purchase Price": "$45.00",  "Cost Basis": "$22,500.00",  "Current Price": "$14.61",  "Market Value": "$7,302.55",  "Gain/Loss": "$7,287.94",  "Dividend/share": "$0.00",  "Yield": "0.00%"  }  ]  } } |
| --- |

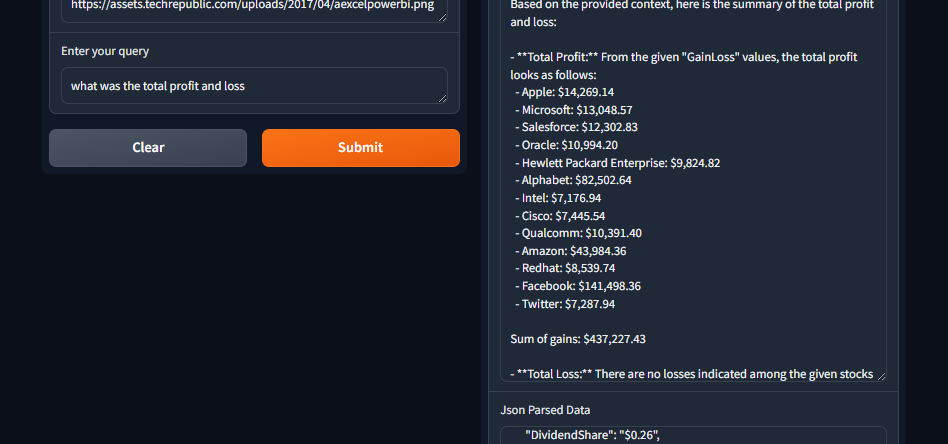
**QnA over Image with Pixtral-12b LLM**

****

**Response:**

| The dividend per share for Apple is $2.28. This means that for each share of Apple stock you own, you will receive $2.28 as a dividend. |
| --- |

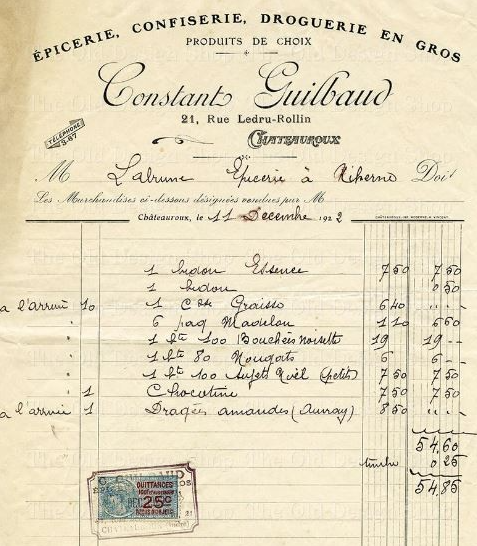
****



**Response:**

| Based on the provided context, here is the summary of the total profit and loss:  - \*\*Total Profit:\*\* From the given "GainLoss" values, the total profit looks as follows:  - Apple: $14,269.14  - Microsoft: $13,048.57  - Salesforce: $12,302.83  - Oracle: $10,994.20  - Hewlett Packard Enterprise: $9,824.82  - Alphabet: $82,502.64  - Intel: $7,176.94  - Cisco: $7,445.54  - Qualcomm: $10,391.40  - Amazon: $43,984.36  - Redhat: $8,539.74  - Facebook: $141,498.36  - Twitter: $7,287.94  Sum of gains: $437,227.43  - \*\*Total Loss:\*\* There are no losses indicated among the given stocks (none of the "GainLoss" values are negative).  Therefore, the total profit from the listed stocks is $437,227.43, and there is no total loss. |
| --- |

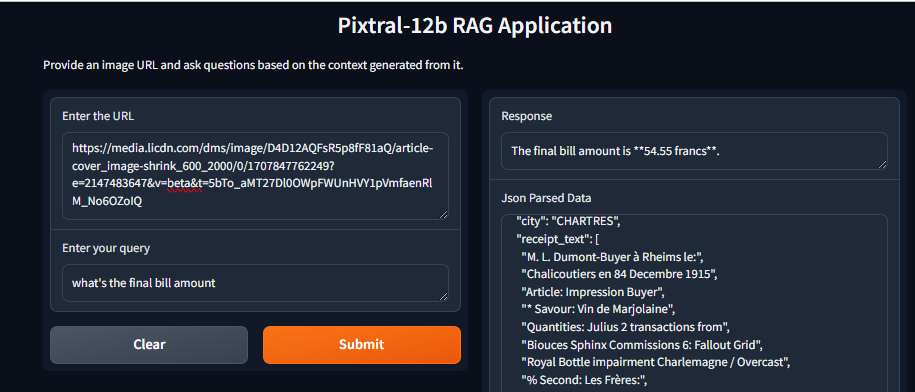
**BILL 2:**

****

**Json Text Extracted 👍**

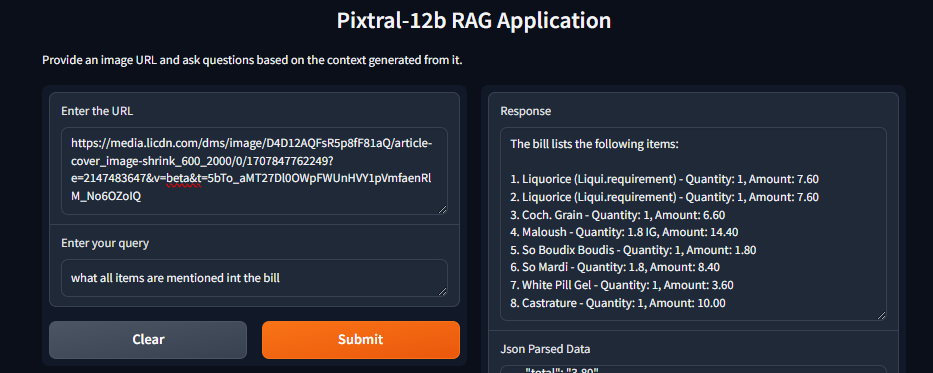
| {  "title": "ÉPICERIE, CONFISERIE, DROGUERIE EN GROS",  "subtitle": "PRODUITS DE GROIX",  "header": "Constances Guilbaud",  "address": "2.1, Rue Ledru-Rollin",  "signature": "M. Ladumaie-Sieux à Richelieu",  "digit": "Les Marchandises ci-dessus désignées ont été reçues ce 95",  "date": "Chantiers, le 4.12.Decembre.1919.",  "items": [  {  "quantity": "1",  "description": "Liquorice (Liqui.requirement)",  "quantity\_unit": "",  "amount": "7.60",  "currency": "",  "total": "7.60"  },  {  "quantity": "1",  "description": "Liquorice (Liqui.requirement)",  "quantity\_unit": "",  "amount": "7.60",  "currency": "",  "total": "7.60"  },  {  "quantity": "1",  "description": "Coch. Grain",  "quantity\_unit": "",  "amount": "6.60",  "currency": "",  "total": "6.60"  },  {  "quantity": "1.8IG",  "description": "Maloush",  "quantity\_unit": "IG",  "amount": "14.40",  "currency": "",  "total": "14.40"  },  {  "quantity": "1",  "description": "So Boudix Boudis",  "quantity\_unit": "",  "amount": "1.80",  "currency": "",  "total": "1.80"  },  {  "quantity": "1.8",  "description": "So Mardi",  "quantity\_unit": "",  "amount": "8.40",  "currency": "",  "total": "8.40"  },  {  "quantity": "1",  "description": "White Pill Gel",  "quantity\_unit": "",  "amount": "3.60",  "currency": "",  "total": "3.80"  },  {  "quantity": "1",  "description": "Castrature ",  "quantity\_unit": "",  "amount": "10.00",  "currency": "",  "total": "10.00"  }  ],  " cultivated": " tot.",  "total": "54.98",  "stamp": {  "text": "Cahier",  "timestamp": "40"  } } |
| --- |

**QnA over image with Pixtral-12b**

****

**Response:**

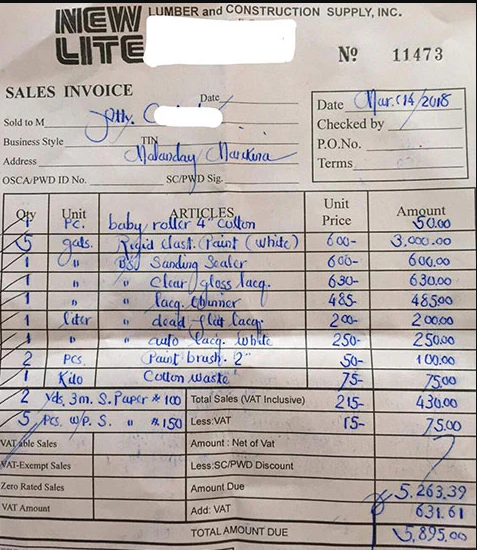
| The final bill amount is \*\*54.55 francs\*\*. |
| --- |

****

**Response:**

| The bill lists the following items:  1. Liquorice (Liqui.requirement) - Quantity: 1, Amount: 7.60 2. Liquorice (Liqui.requirement) - Quantity: 1, Amount: 7.60 3. Coch. Grain - Quantity: 1, Amount: 6.60 4. Maloush - Quantity: 1.8 IG, Amount: 14.40 5. So Boudix Boudis - Quantity: 1, Amount: 1.80 6. So Mardi - Quantity: 1.8, Amount: 8.40 7. White Pill Gel - Quantity: 1, Amount: 3.60 8. Castrature - Quantity: 1, Amount: 10.00 |
| --- |

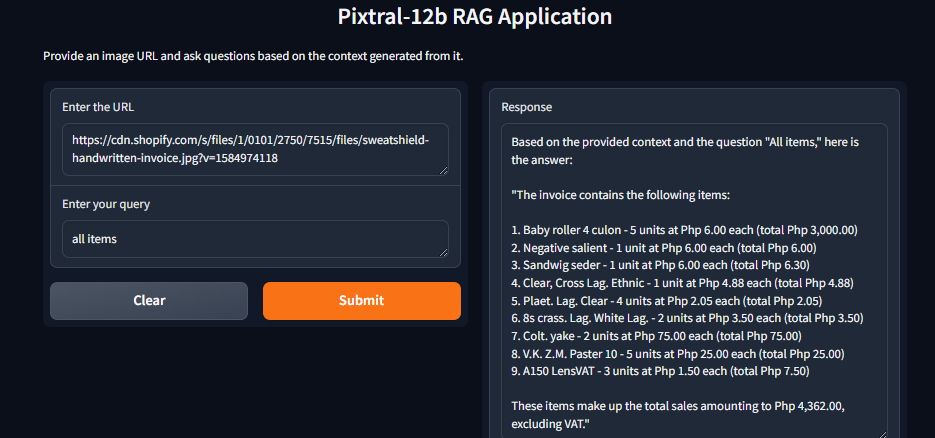
**BILL 3:**

****

**Json Text Extracted 👍**

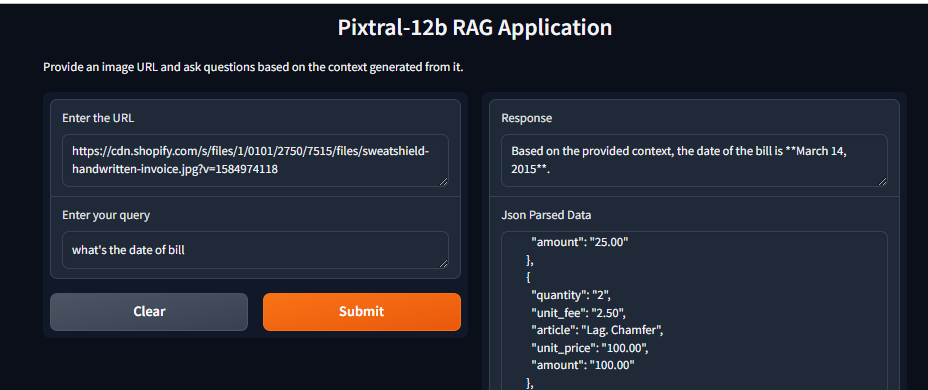
| {  "invoice\_number": "11473",  "invoice\_date": "Mar 14, 2018",  "issued\_to": "THE WEDDING ARRANGER",  "items": [  {  "item\_code": "7005",  "description": "baby roller 4 pcs",  "quantity": "3",  "unit\_price": "6.00",  "total\_price": "18.00"  },  {  "item\_code": "410",  "description": "bed pad grind size inchesX24 inches X150",  "quantity": "n",  "unit\_price": "n",  "total\_price": "n/a"  },  {  "item\_code": "53",  "description": "share grinder 450 Watt",  "quantity": "n",  "unit\_price": "6.30",  "total\_price": "6.30"  },  {  "item\_code": "n",  "description": "tag cloth for cloth",  "quantity": "n",  "unit\_price": "48.50",  "total\_price": "48.50"  },  {  "item\_code": "n",  "description": " coalated expect Fogg Paper for steel",  "quantity": "n",  "unit\_price": "n",  "total\_price": "n/a"  },  {  "item\_code": "500",  "description": "4 inch cloth for fogging the size of new bottles",  "quantity": "3",  "unit\_price": "8.00",  "total\_price": "24.00"  },  {  "item\_code": "102",  "description": "brass brush roller with ring",  "quantity": "4",  "unit\_price": "3.00",  "total\_price": "12.00"  },  {  "item\_code": "00",  "description": "cotton cloth",  "quantity": "2",  "unit\_price": "7.50",  "total\_price": "7.50"  },  {  "item\_code": "20",  "description": "mesh NAC 4 meter",  "quantity": "4",  "unit\_price": "4.150",  "total\_price": "16.60"  },  {  "item\_code": "00",  "description": "cotton cloth wash as per m",  "quantity": "2",  "unit\_price": "2.75",  "total\_price": "5.50"  }  ],  "tax": "631.61",  "total": "5895.00" } |
| --- |

**QnA over image with Pixtral-12b**

****

**Response:**

| Based on the provided context and the question "All items," here is the answer:  "The invoice contains the following items:  1. Baby roller 4 culon - 5 units at Php 6.00 each (total Php 3,000.00) 2. Negative salient - 1 unit at Php 6.00 each (total Php 6.00) 3. Sandwig seder - 1 unit at Php 6.00 each (total Php 6.30) 4. Clear, Cross Lag. Ethnic - 1 unit at Php 4.88 each (total Php 4.88) 5. Plaet. Lag. Clear - 4 units at Php 2.05 each (total Php 2.05) 6. 8s crass. Lag. White Lag. - 2 units at Php 3.50 each (total Php 3.50) 7. Colt. yake - 2 units at Php 75.00 each (total Php 75.00) 8. V.K. Z.M. Paster 10 - 5 units at Php 25.00 each (total Php 25.00) 9. A150 LensVAT - 3 units at Php 1.50 each (total Php 7.50)  These items make up the total sales amounting to Php 4,362.00, excluding VAT." |
| --- |

****

**Response:**

| Based on the provided context, the date of the bill is \*\*March 14, 2015\*\*. |
| --- |

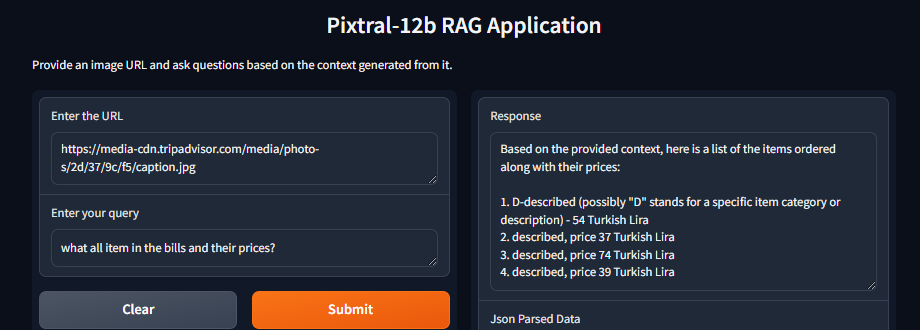
**Bill 4 :**

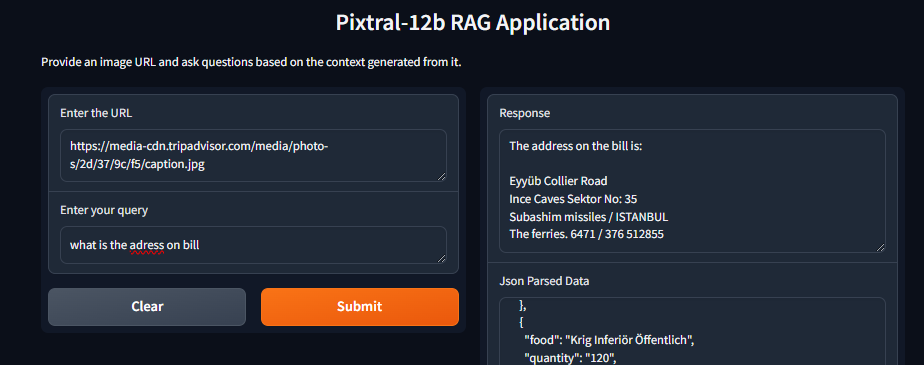
****

**Json Extracted Text 👍**

| {  "menu": {  "title": "PALMIYE RESTAURANT & CAFE",  "location": "Eyüpacolı Evler K starter 8",  "address": "İncirli Caddesi Sok. No:70, Sudanşhrie'stanbul - ISTANBUL",  "contact": "Telefon +90 212 641 76 76 - Faks: +90 212 641 76 77",  "stamp": "TURKISH CUISINE"  },  "admission": {  "name": "ADİSYON",  "ref no": " ballet 174528",  "date": "12.10.2019",  "Registration no": "SEN- A- 6723",  "commital date": "Görüşмой",  "no": "Il Koddu 34 - No: 115059"  },  "order": {  "cinsi": [  "D-described: 50",  "described: 80",  "described: 50",  "described: 30"  ],  "mik": [  "",  "",  "",  ""  ],  "fiyati": [  "54",  "37",  "74",  "39"  ],  "tutar": [  "27",  "26",  "70",  "19"  ]  },  "notes": [  {  "title": "D-site",  "text": "Günaydın Ramazan",  "subtext": "Barkod"  },  {  "title": "Site",  "text": "KFrontwire",  "subtext": "Belusage"  }  ] } |
| --- |

**QnA over image with Pixtral-12b**

****

****

| The address on the bill is:  Eyyüb Collier Road Ince Caves Sektor No: 35 Subashim missiles / ISTANBUL The ferries. 6471 / 376 512855 |
| --- |