HANDS-ON MACHINE LEARNING (HOME)

Lecture/Lab – 3

An Introduction to Machine Learning Using LLMs as an Example – Part III

Goal: Extract Discrete Variables from TCGA Pathology Reports Using Local LLMs

Ghulam Rasool, PhD

Department of Machine Learning, Moffitt Cancer Center

Overview

Running LLMs on Local Machines Using **Ollama**

- Python
  - **Miniconda**
  - **VSCode**
- Prompting and Retrieval Augmented Generation (RAG)
  - **LangChain**
- Graphical User Interface
  - **Open WebUI**

ollama

Windows

https://ollama.com/

- Download and install
- Running LLMs, locally
- http://localhost:11434

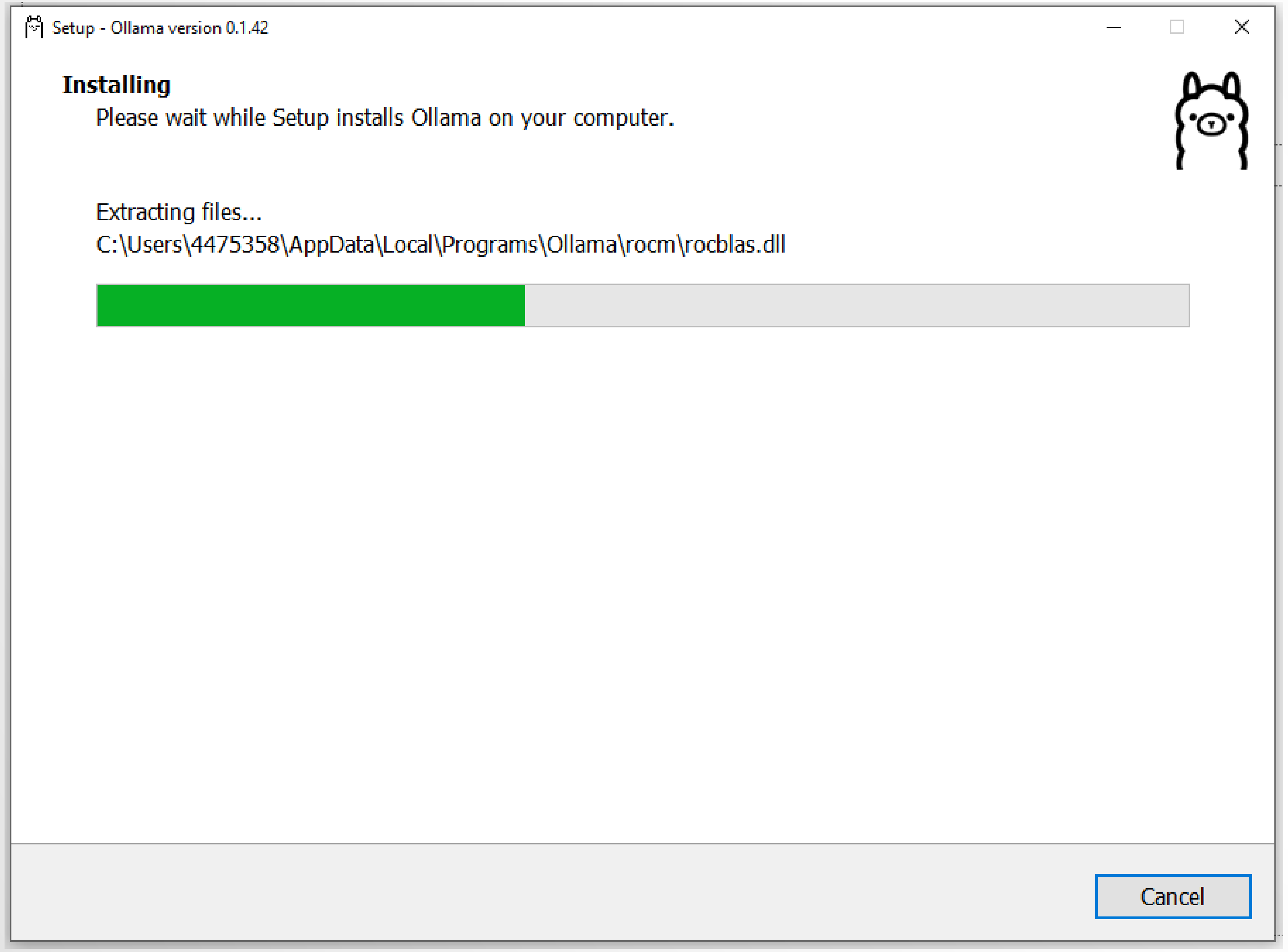
Linux

- Download ollama binary

```
curl -L https://ollama.com/download/ollama-linux-amd64 -o ollama

ollama server&
```

- http://localhost:11434

Download and install ollama


Setup - Ollama version 0.1.42

Installing

Please wait while Setup installs Ollama on your computer.

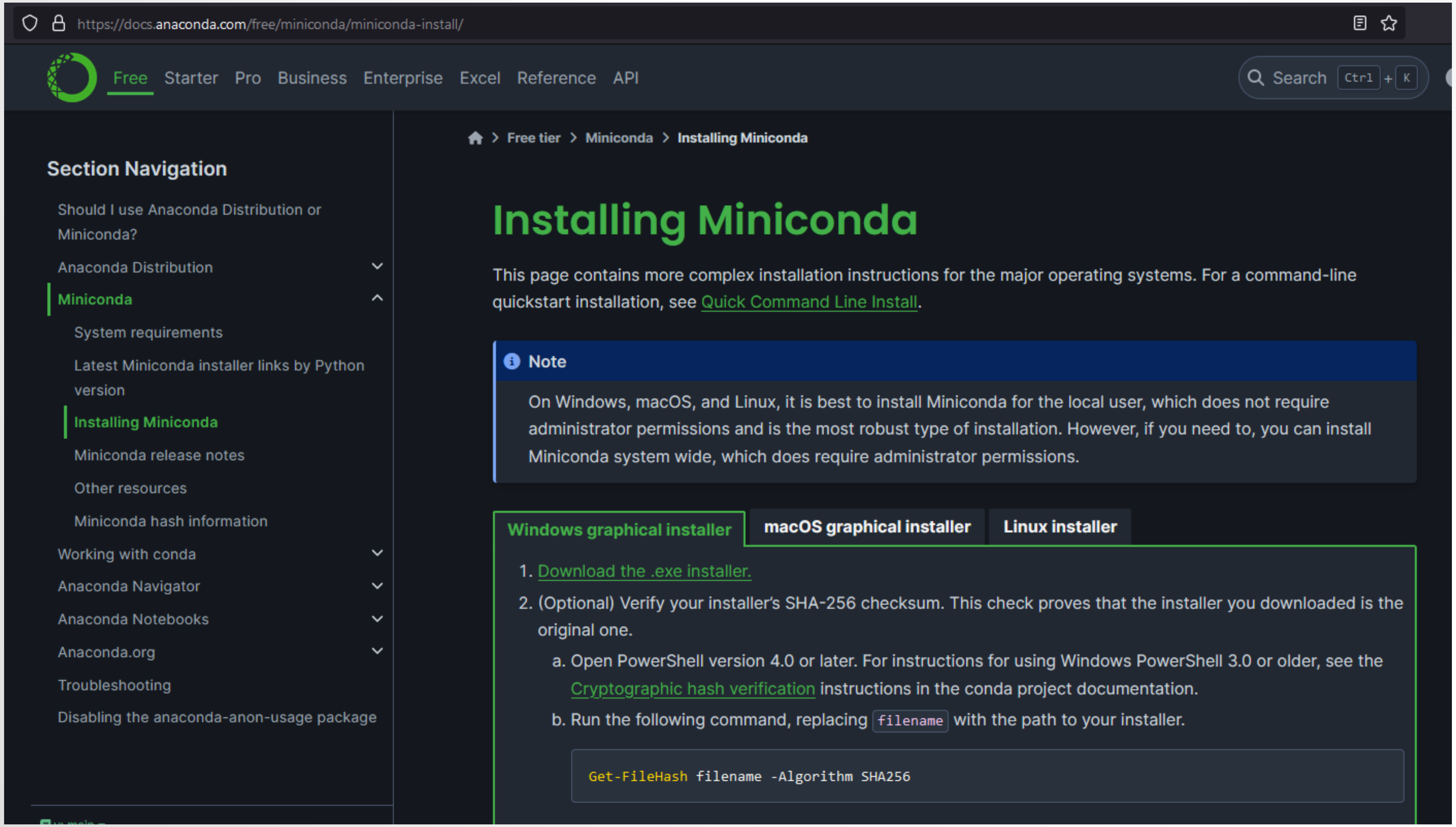
Extracting files...

C:\Users\4475358\AppData\Local\Programs\Ollama\rocm\rocblas.dll

Graphical User Interface - Open Webui

- https://github.com/open-webui/open-webui

Download and install Miniconda



https://docs.anaconda.com/free/miniconda/miniconda-install/

Free Starter Pro Business Enterprise Excel Reference API Search Ctrl + K

Free tier > Miniconda > Installing Miniconda

Section Navigation

- Should I use Anaconda Distribution or Miniconda?
- Anaconda Distribution
- Miniconda
  - System requirements
  - Latest Miniconda installer links by Python version
  - Installing Miniconda
  - Miniconda release notes
  - Other resources
  - Miniconda hash information
- Working with conda
- Anaconda Navigator
- Anaconda Notebooks
- Anaconda.org
- Troubleshooting
- Disabling the anaconda-anon-usage package

Installing Miniconda

This page contains more complex installation instructions for the major operating systems. For a command-line quickstart installation, see Quick Command Line Install.

Note

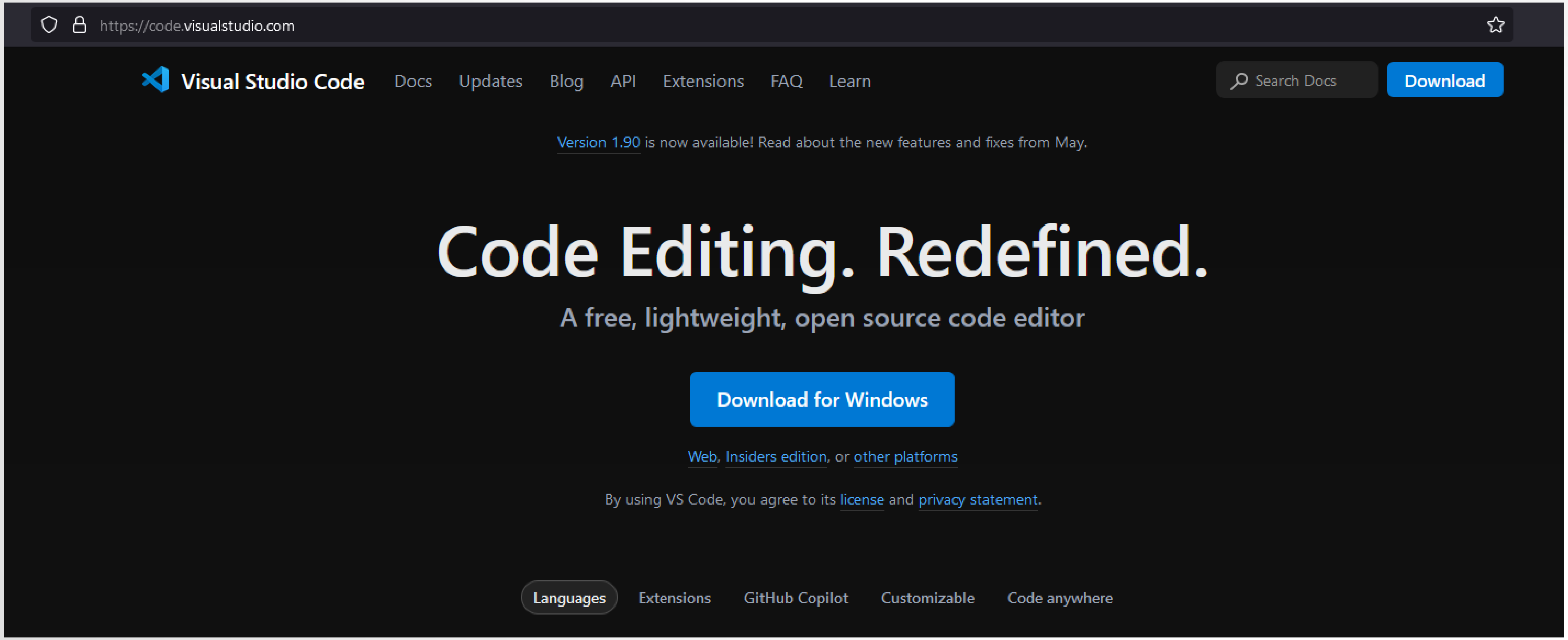
On Windows, macOS, and Linux, it is best to install Miniconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, if you need to, you can install Miniconda system wide, which does require administrator permissions.

Windows graphical installer | macOS graphical installer | Linux installer

1. Download the .exe installer.
2. (Optional) Verify your installer's SHA-256 checksum. This check proves that the installer you downloaded is the original one.
   a. Open PowerShell version 4.0 or later. For instructions for using Windows PowerShell 3.0 or older, see the Cryptographic hash verification instructions in the conda project documentation.
   b. Run the following command, replacing `filename` with the path to your installer.

```
Get-FileHash filename -Algorithm SHA256
```

Install VSCode


https://code.visualstudio.com

Version 1.90 is now available! Read about the new features and fixes from May.

Code Editing. Redefined.

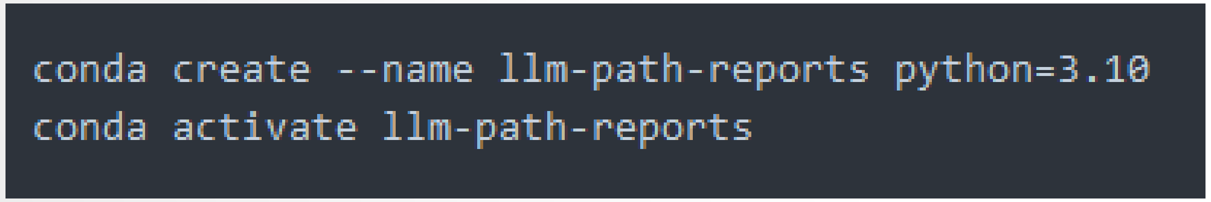
A free, lightweight, open source code editor

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

Create a virtual environment and activate it

- Open VSCode
- Open a Terminal
- Create a virtual environment, activate it



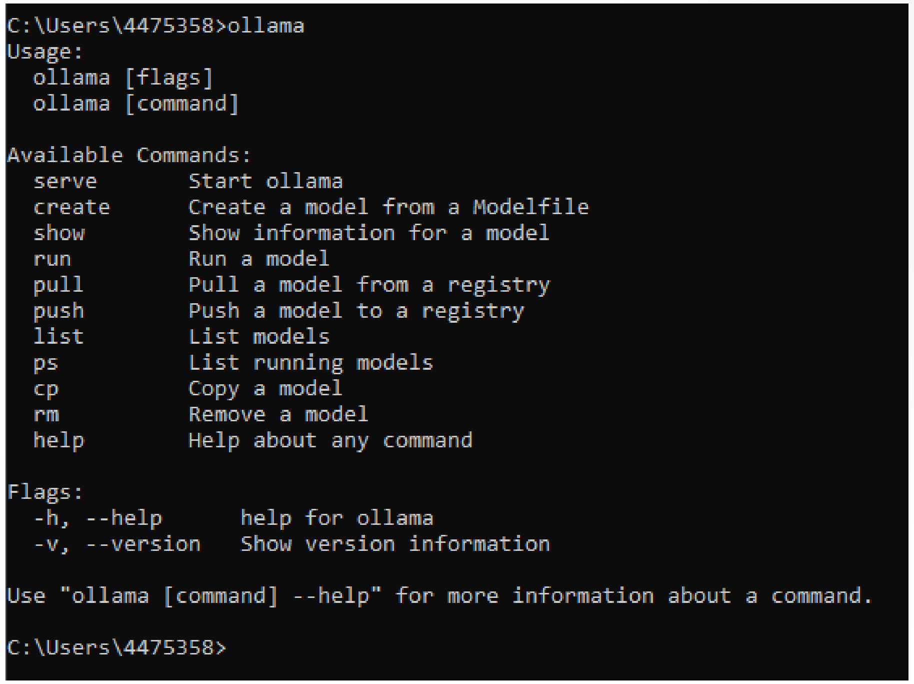
```
conda create --name llm-path-reports python=3.10
conda activate llm-path-reports
```

Clone GitHub Repo and Install requirments

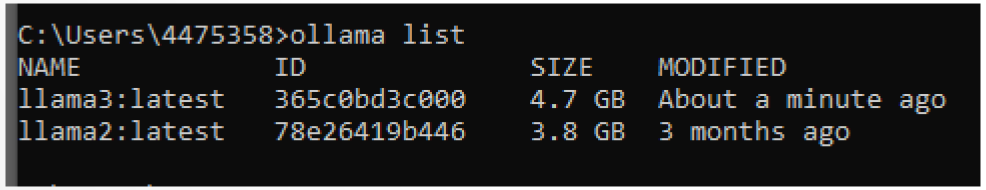
- *(llm-path-reports) > git clone https://github.com/grasool/HOME*
- *(llm-path-reports) > pip install -r .\requirements.txt*

Run Ollama

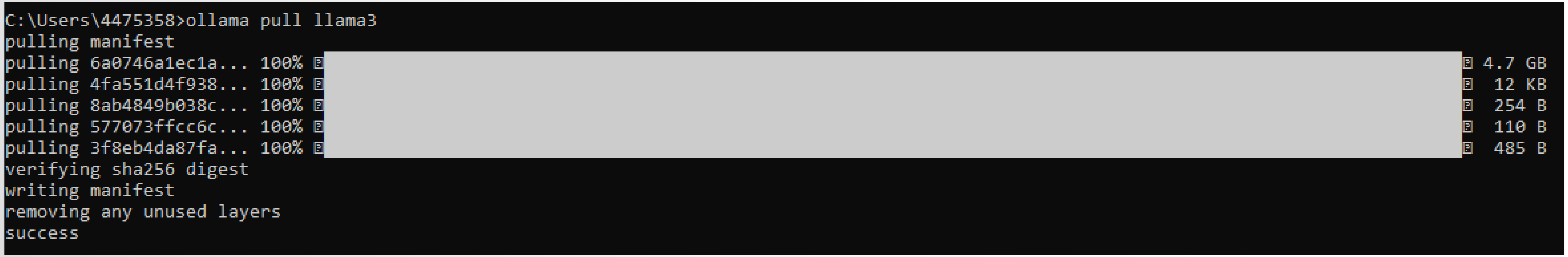
- ollama
- Ollama list
- Ollama pull llama 3



```
C:\Users\4475358>ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version   Show version information

Use "ollama [command] --help" for more information about a command.

C:\Users\4475358>
```

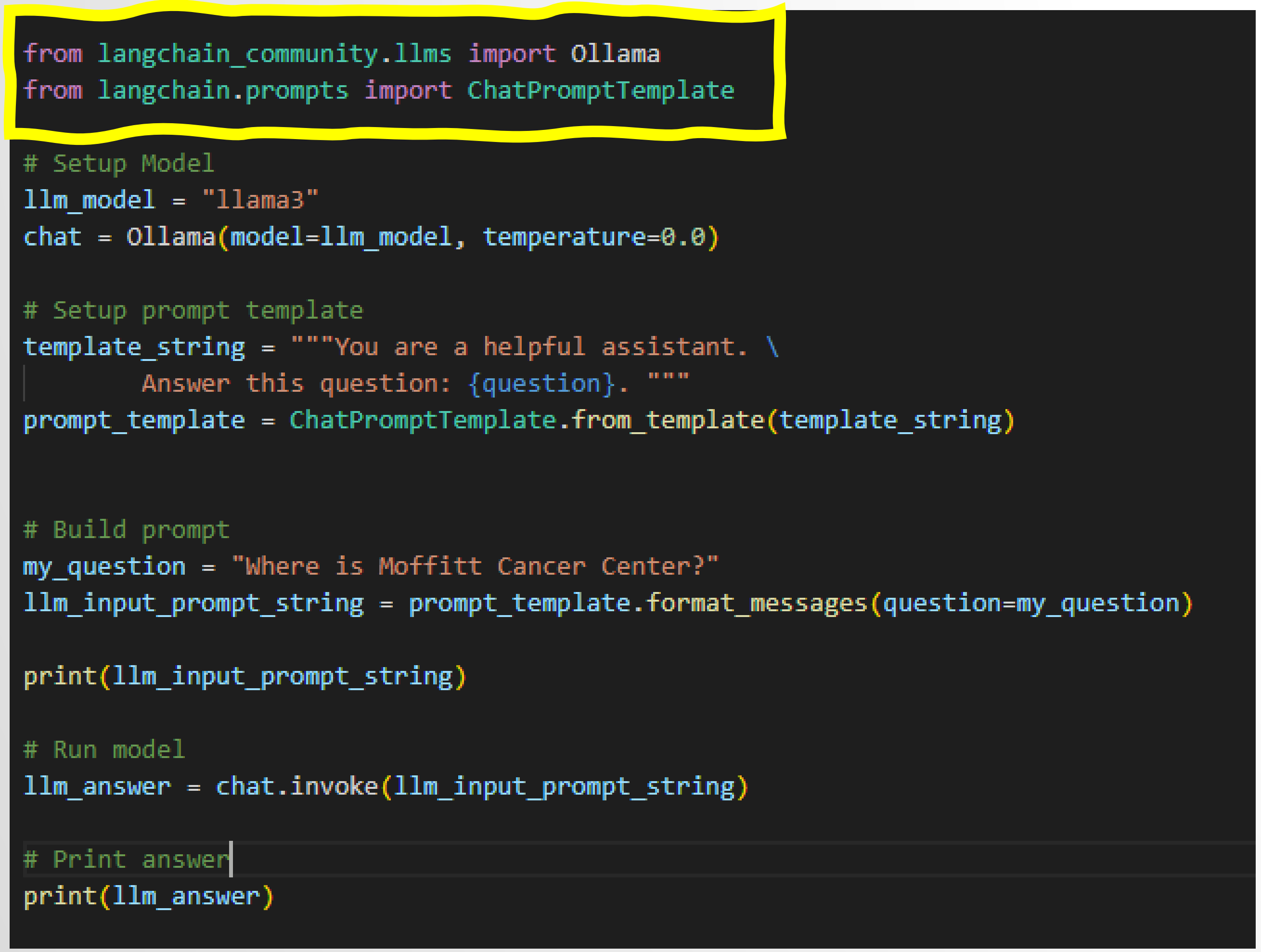


```
C:\Users\4475358>ollama list
NAME            ID              SIZE    MODIFIED
llama3:latest   365c0bd3c000    4.7 GB  About a minute ago
llama2:latest   78e26419b446    3.8 GB  3 months ago
```



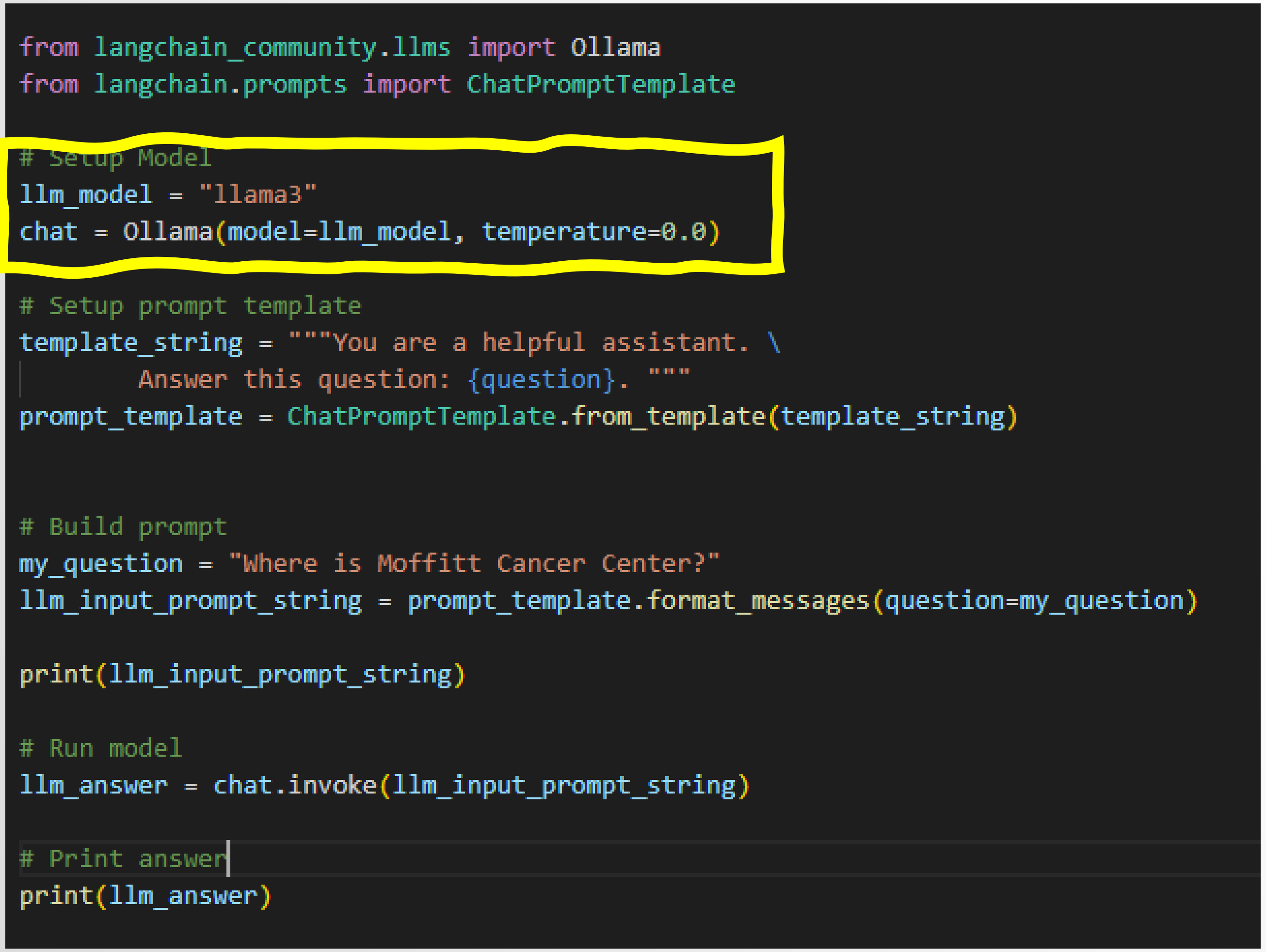
```
C:\Users\4475358>ollama pull llama3
pulling manifest
pulling 6a0746a1ec1a... 100% ▕████████████████▏ 4.7 GB
pulling 4fa551d4f938... 100% ▕████████████████▏  12 KB
pulling 8ab4849b038c... 100% ▕████████████████▏  254 B
pulling 577073ffcc6c... 100% ▕████████████████▏  110 B
pulling 3f8eb4da87fa... 100% ▕████████████████▏  485 B
verifying sha256 digest
writing manifest
removing any unused layers
success
```

First Python Code



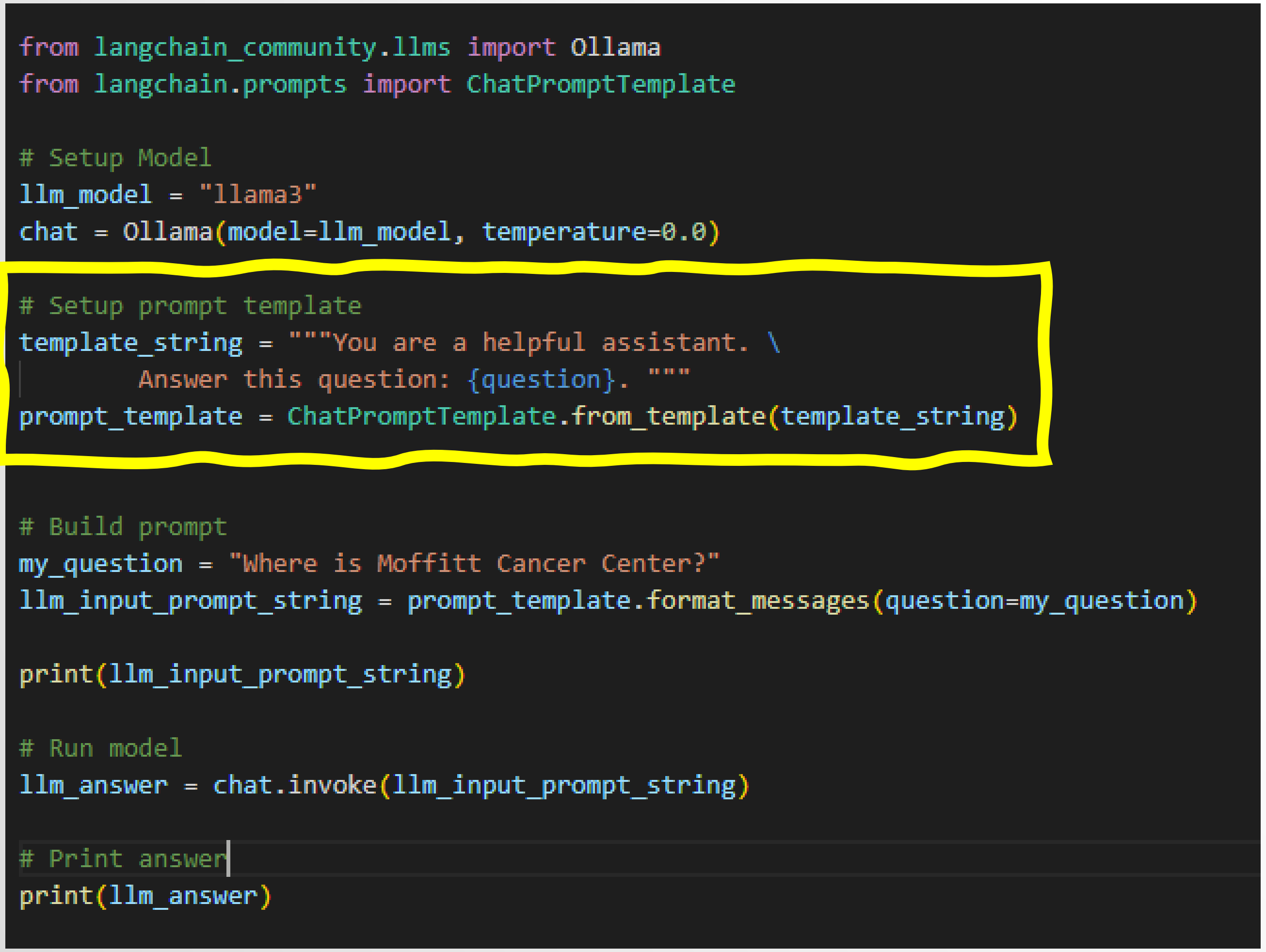
```python
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

# Setup Model
llm_model = "llama3"
chat = Ollama(model=llm_model, temperature=0.0)

# Setup prompt template
template_string = """You are a helpful assistant. \
        Answer this question: {question}. """
prompt_template = ChatPromptTemplate.from_template(template_string)


# Build prompt
my_question = "Where is Moffitt Cancer Center?"
llm_input_prompt_string = prompt_template.format_messages(question=my_question)

print(llm_input_prompt_string)

# Run model
llm_answer = chat.invoke(llm_input_prompt_string)

# Print answer
print(llm_answer)
```

First Python Code



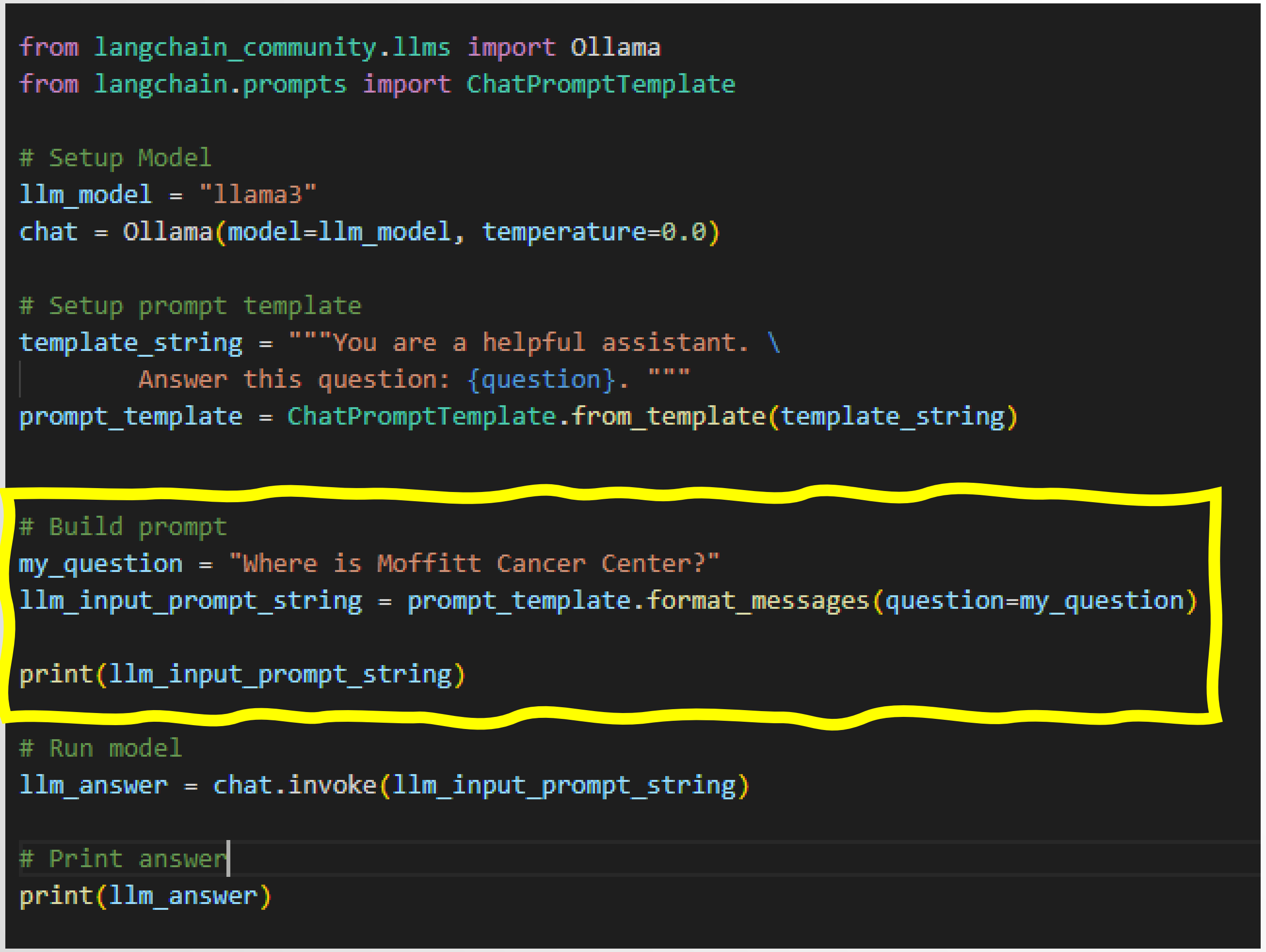
```python
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

# Setup Model
llm_model = "llama3"
chat = Ollama(model=llm_model, temperature=0.0)

# Setup prompt template
template_string = """You are a helpful assistant. \
        Answer this question: {question}. """
prompt_template = ChatPromptTemplate.from_template(template_string)


# Build prompt
my_question = "Where is Moffitt Cancer Center?"
llm_input_prompt_string = prompt_template.format_messages(question=my_question)

print(llm_input_prompt_string)

# Run model
llm_answer = chat.invoke(llm_input_prompt_string)

# Print answer
print(llm_answer)
```

First Python Code

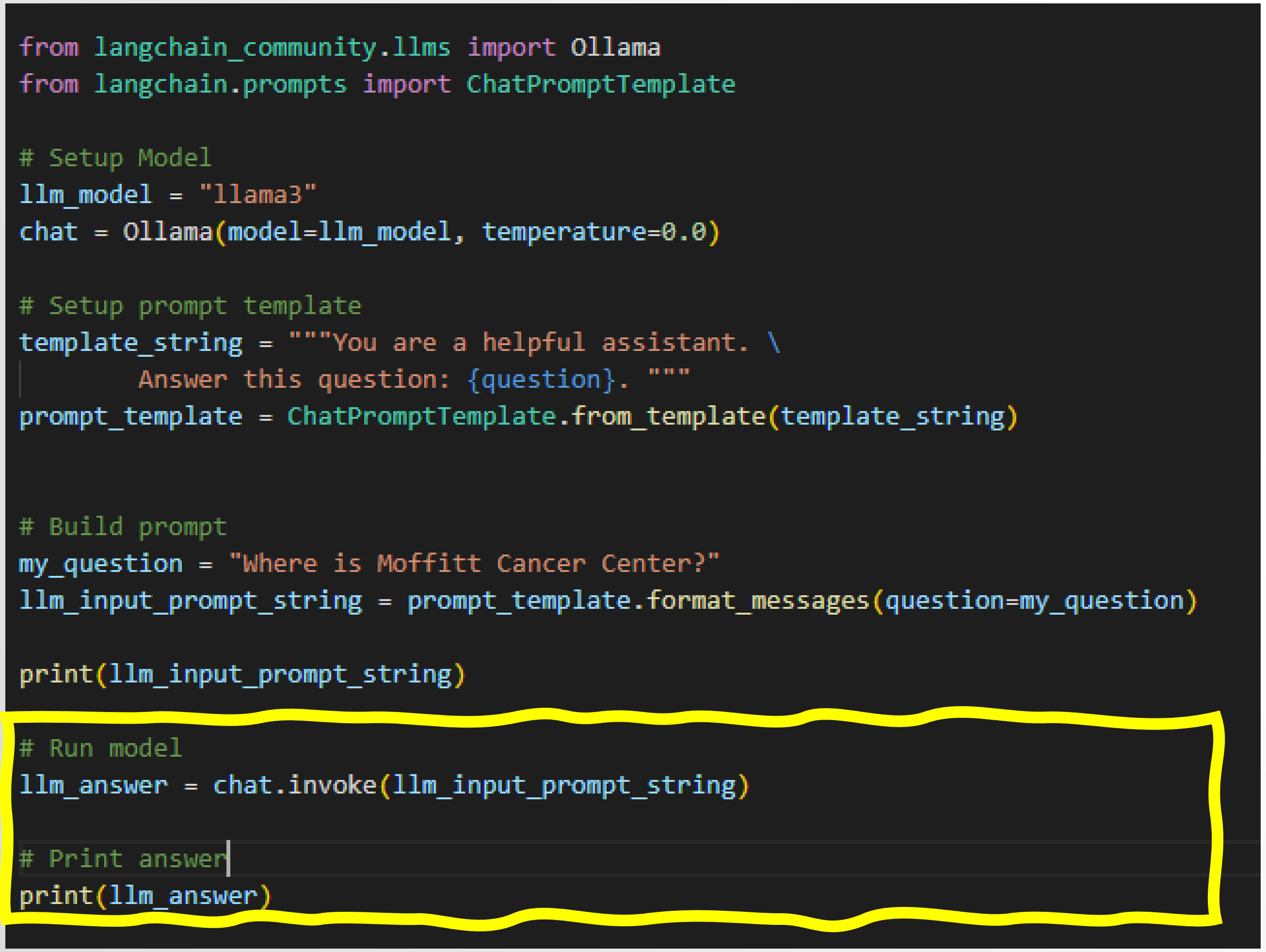


```python
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

# Setup Model
llm_model = "llama3"
chat = Ollama(model=llm_model, temperature=0.0)

# Setup prompt template
template_string = """You are a helpful assistant. \
        Answer this question: {question}. """
prompt_template = ChatPromptTemplate.from_template(template_string)


# Build prompt
my_question = "Where is Moffitt Cancer Center?"
llm_input_prompt_string = prompt_template.format_messages(question=my_question)

print(llm_input_prompt_string)

# Run model
llm_answer = chat.invoke(llm_input_prompt_string)

# Print answer
print(llm_answer)
```

First Python Code

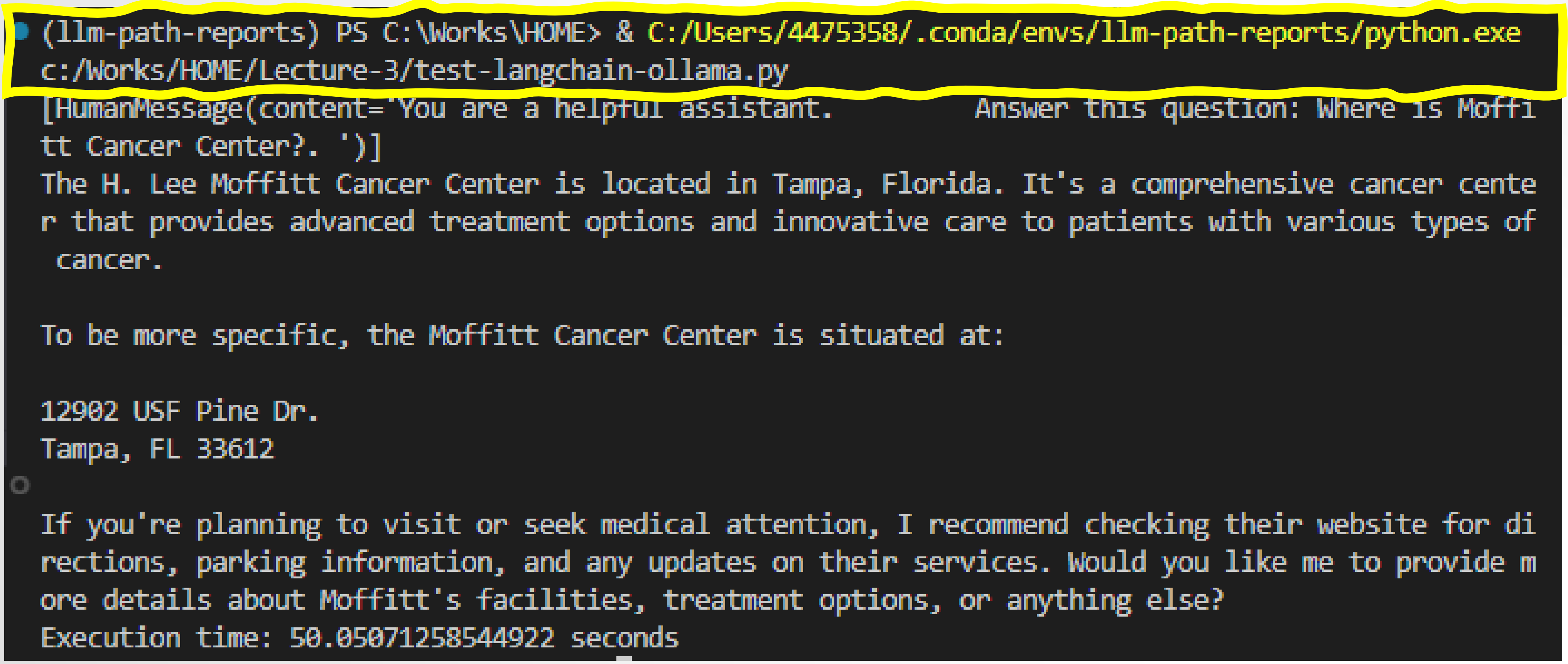

```
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

# Setup Model
llm_model = "llama3"
chat = Ollama(model=llm_model, temperature=0.0)

# Setup prompt template
template_string = """You are a helpful assistant. \
        Answer this question: {question}. """
prompt_template = ChatPromptTemplate.from_template(template_string)


# Build prompt
my_question = "Where is Moffitt Cancer Center?"
llm_input_prompt_string = prompt_template.format_messages(question=my_question)

print(llm_input_prompt_string)

# Run model
llm_answer = chat.invoke(llm_input_prompt_string)

# Print answer
print(llm_answer)
```

First Python Code



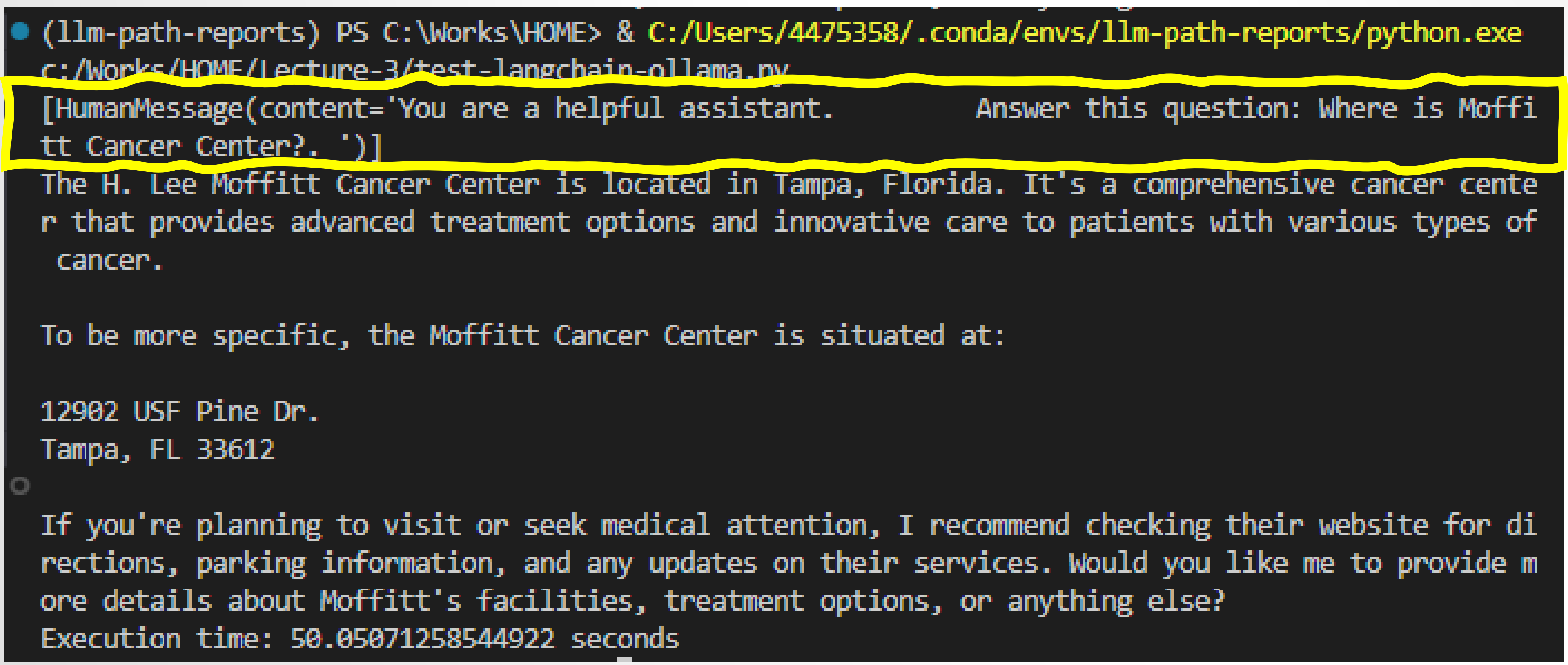
```python
from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

# Setup Model
llm_model = "llama3"
chat = Ollama(model=llm_model, temperature=0.0)

# Setup prompt template
template_string = """You are a helpful assistant. \
        Answer this question: {question}. """
prompt_template = ChatPromptTemplate.from_template(template_string)


# Build prompt
my_question = "Where is Moffitt Cancer Center?"
llm_input_prompt_string = prompt_template.format_messages(question=my_question)

print(llm_input_prompt_string)

# Run model
llm_answer = chat.invoke(llm_input_prompt_string)

# Print answer
print(llm_answer)
```

First Python Code



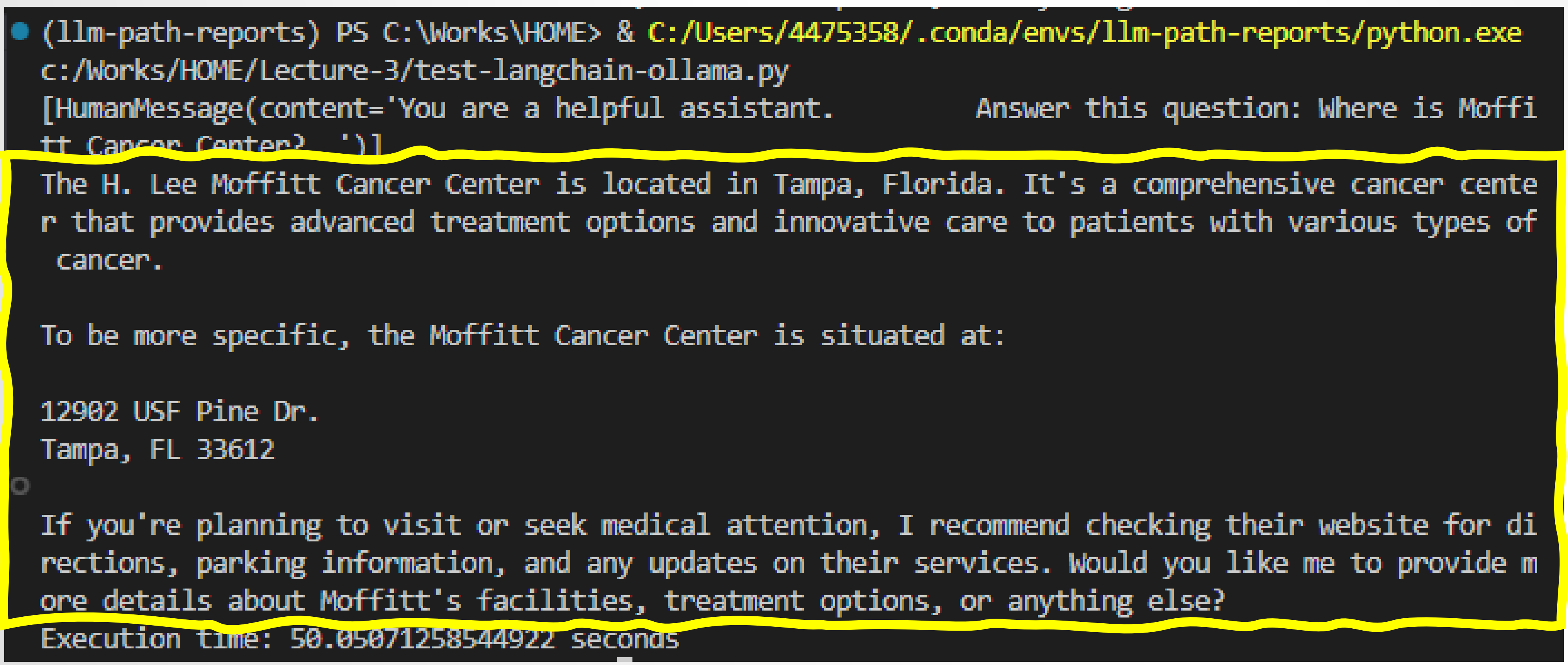
```
(llm-path-reports) PS C:\Works\HOME> & C:/Users/4475358/.conda/envs/llm-path-reports/python.exe
c:/Works/HOME/Lecture-3/test-langchain-ollama.py
[HumanMessage(content='You are a helpful assistant.         Answer this question: Where is Moffi
tt Cancer Center?. ')]
The H. Lee Moffitt Cancer Center is located in Tampa, Florida. It's a comprehensive cancer cente
r that provides advanced treatment options and innovative care to patients with various types of
 cancer.

To be more specific, the Moffitt Cancer Center is situated at:

12902 USF Pine Dr.
Tampa, FL 33612

If you're planning to visit or seek medical attention, I recommend checking their website for di
rections, parking information, and any updates on their services. Would you like me to provide m
ore details about Moffitt's facilities, treatment options, or anything else?
Execution time: 50.05071258544922 seconds
```

First Python Code



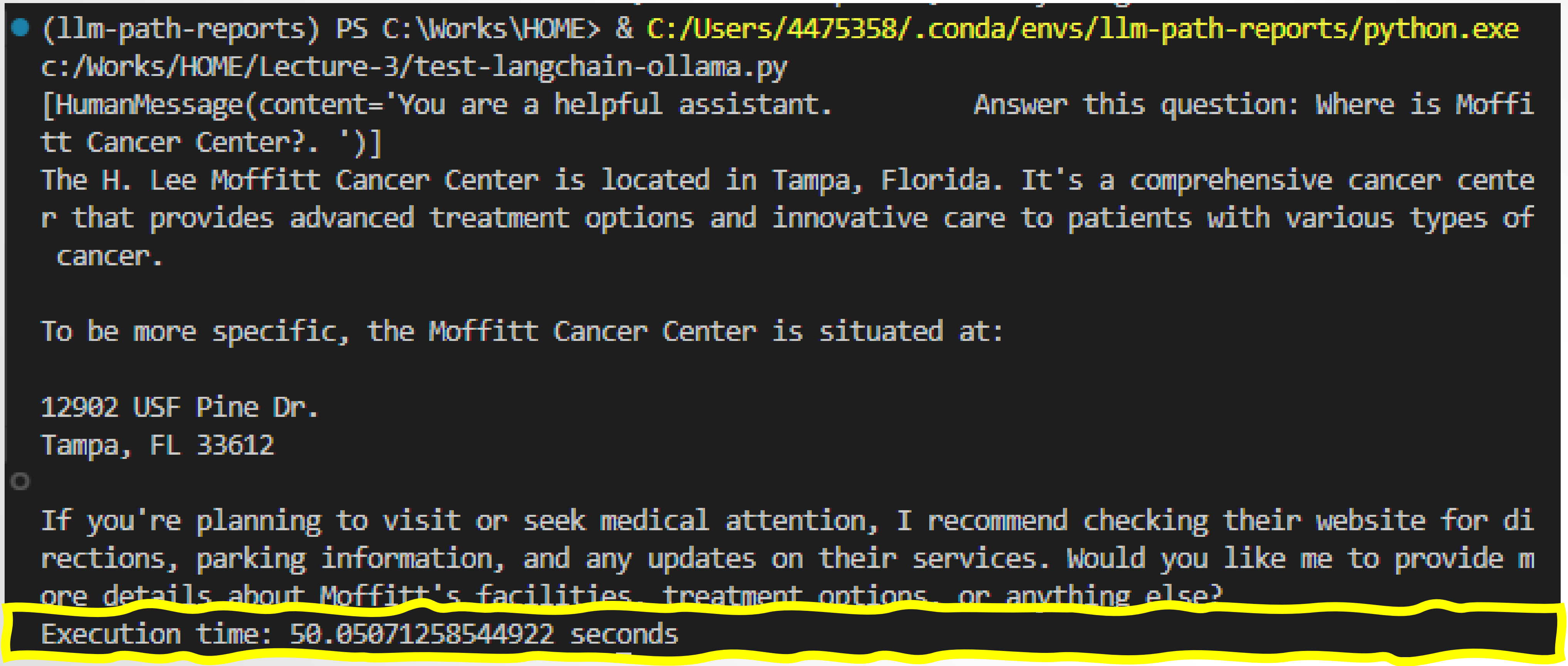
```
(llm-path-reports) PS C:\Works\HOME> & C:/Users/4475358/.conda/envs/llm-path-reports/python.exe
c:/Works/HOME/Lecture-3/test-langchain-ollama.py
[HumanMessage(content='You are a helpful assistant.         Answer this question: Where is Moffi
tt Cancer Center?. ')]
The H. Lee Moffitt Cancer Center is located in Tampa, Florida. It's a comprehensive cancer cente
r that provides advanced treatment options and innovative care to patients with various types of
 cancer.

To be more specific, the Moffitt Cancer Center is situated at:

12902 USF Pine Dr.
Tampa, FL 33612

If you're planning to visit or seek medical attention, I recommend checking their website for di
rections, parking information, and any updates on their services. Would you like me to provide m
ore details about Moffitt's facilities, treatment options, or anything else?
Execution time: 50.05071258544922 seconds
```

First Python Code



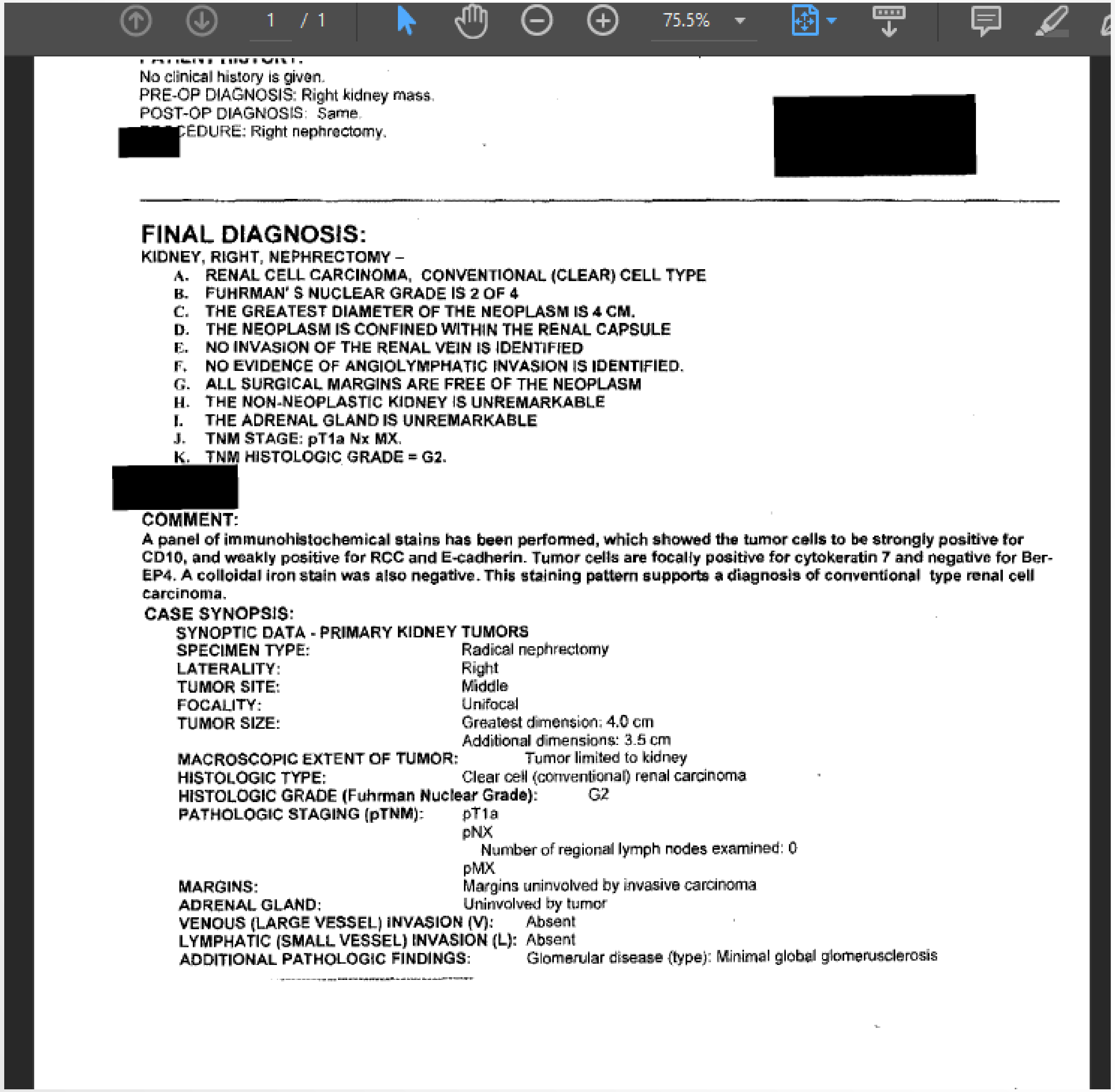
```
(llm-path-reports) PS C:\Works\HOME> & C:/Users/4475358/.conda/envs/llm-path-reports/python.exe
c:/Works/HOME/Lecture-3/test-langchain-ollama.py
[HumanMessage(content='You are a helpful assistant.        Answer this question: Where is Moffi
tt Cancer Center?  ')]
The H. Lee Moffitt Cancer Center is located in Tampa, Florida. It's a comprehensive cancer cente
r that provides advanced treatment options and innovative care to patients with various types of
 cancer.

To be more specific, the Moffitt Cancer Center is situated at:

12902 USF Pine Dr.
Tampa, FL 33612

If you're planning to visit or seek medical attention, I recommend checking their website for di
rections, parking information, and any updates on their services. Would you like me to provide m
ore details about Moffitt's facilities, treatment options, or anything else?
Execution time: 50.05071258544922 seconds
```

First Python Code



```
(llm-path-reports) PS C:\Works\HOME> & C:/Users/4475358/.conda/envs/llm-path-reports/python.exe
c:/Works/HOME/Lecture-3/test-langchain-ollama.py
[HumanMessage(content='You are a helpful assistant.         Answer this question: Where is Moffi
tt Cancer Center?. ')]
The H. Lee Moffitt Cancer Center is located in Tampa, Florida. It's a comprehensive cancer cente
r that provides advanced treatment options and innovative care to patients with various types of
 cancer.

To be more specific, the Moffitt Cancer Center is situated at:

12902 USF Pine Dr.
Tampa, FL 33612

If you're planning to visit or seek medical attention, I recommend checking their website for di
rections, parking information, and any updates on their services. Would you like me to provide m
ore details about Moffitt's facilities, treatment options, or anything else?
Execution time: 50.05071258544922 seconds
```

Path Report



No clinical history is given.
PRE-OP DIAGNOSIS: Right kidney mass.
POST-OP DIAGNOSIS: Same.
[redacted]CEDURE: Right nephrectomy.

[redacted]

FINAL DIAGNOSIS:

KIDNEY, RIGHT, NEPHRECTOMY –

A. RENAL CELL CARCINOMA, CONVENTIONAL (CLEAR) CELL TYPE
B. FUHRMAN' S NUCLEAR GRADE IS 2 OF 4
C. THE GREATEST DIAMETER OF THE NEOPLASM IS 4 CM.
D. THE NEOPLASM IS CONFINED WITHIN THE RENAL CAPSULE
E. NO INVASION OF THE RENAL VEIN IS IDENTIFIED
F. NO EVIDENCE OF ANGIOLYMPHATIC INVASION IS IDENTIFIED.
G. ALL SURGICAL MARGINS ARE FREE OF THE NEOPLASM
H. THE NON-NEOPLASTIC KIDNEY IS UNREMARKABLE
I. THE ADRENAL GLAND IS UNREMARKABLE
J. TNM STAGE: pT1a Nx MX.
K. TNM HISTOLOGIC GRADE = G2.

[redacted]

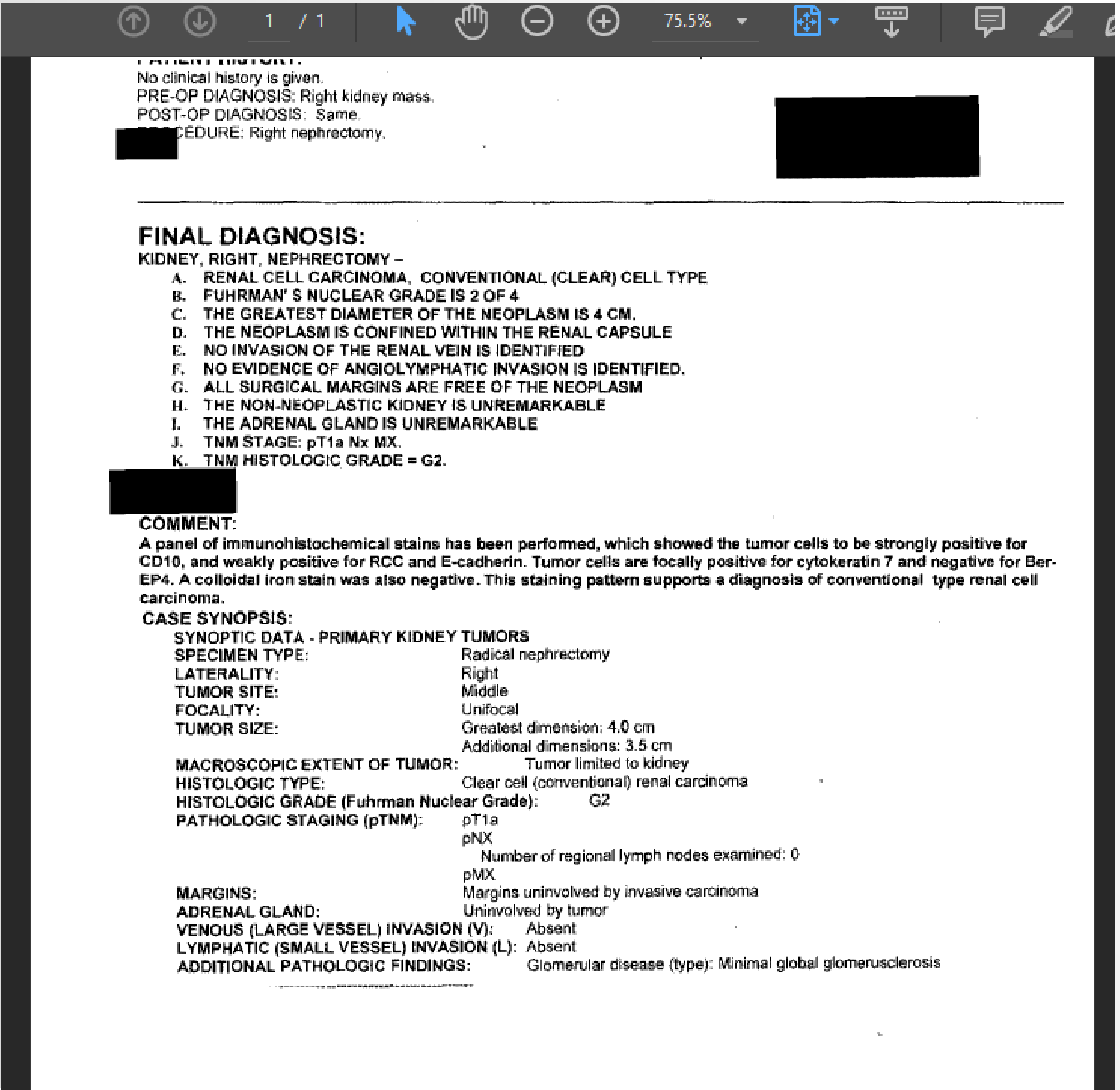
COMMENT:
A panel of immunohistochemical stains has been performed, which showed the tumor cells to be strongly positive for CD10, and weakly positive for RCC and E-cadherin. Tumor cells are focally positive for cytokeratin 7 and negative for Ber-EP4. A colloidal iron stain was also negative. This staining pattern supports a diagnosis of conventional type renal cell carcinoma.

CASE SYNOPSIS:

SYNOPTIC DATA - PRIMARY KIDNEY TUMORS

SPECIMEN TYPE:	Radical nephrectomy
LATERALITY:	Right
TUMOR SITE:	Middle
FOCALITY:	Unifocal
TUMOR SIZE:	Greatest dimension: 4.0 cm Additional dimensions: 3.5 cm
MACROSCOPIC EXTENT OF TUMOR:	Tumor limited to kidney
HISTOLOGIC TYPE:	Clear cell (conventional) renal carcinoma
HISTOLOGIC GRADE (Fuhrman Nuclear Grade):	G2
PATHOLOGIC STAGING (pTNM):	pT1a pNX Number of regional lymph nodes examined: 0 pMX
MARGINS:	Margins uninvolved by invasive carcinoma
ADRENAL GLAND:	Uninvolved by tumor
VENOUS (LARGE VESSEL) INVASION (V):	Absent
LYMPHATIC (SMALL VESSEL) INVASION (L):	Absent
ADDITIONAL PATHOLOGIC FINDINGS:	Glomerular disease (type): Minimal global glomerusclerosis

Stage – 0 (OCR)



No clinical history is given.
PRE-OP DIAGNOSIS: Right kidney mass.
POST-OP DIAGNOSIS: Same.
[redacted]CEDURE: Right nephrectomy.

[redacted]

FINAL DIAGNOSIS:
KIDNEY, RIGHT, NEPHRECTOMY –

A. RENAL CELL CARCINOMA, CONVENTIONAL (CLEAR) CELL TYPE
B. FUHRMAN' S NUCLEAR GRADE IS 2 OF 4
C. THE GREATEST DIAMETER OF THE NEOPLASM IS 4 CM.
D. THE NEOPLASM IS CONFINED WITHIN THE RENAL CAPSULE
E. NO INVASION OF THE RENAL VEIN IS IDENTIFIED
F. NO EVIDENCE OF ANGIOLYMPHATIC INVASION IS IDENTIFIED.
G. ALL SURGICAL MARGINS ARE FREE OF THE NEOPLASM
H. THE NON-NEOPLASTIC KIDNEY IS UNREMARKABLE
I. THE ADRENAL GLAND IS UNREMARKABLE
J. TNM STAGE: pT1a Nx MX.
K. TNM HISTOLOGIC GRADE = G2.

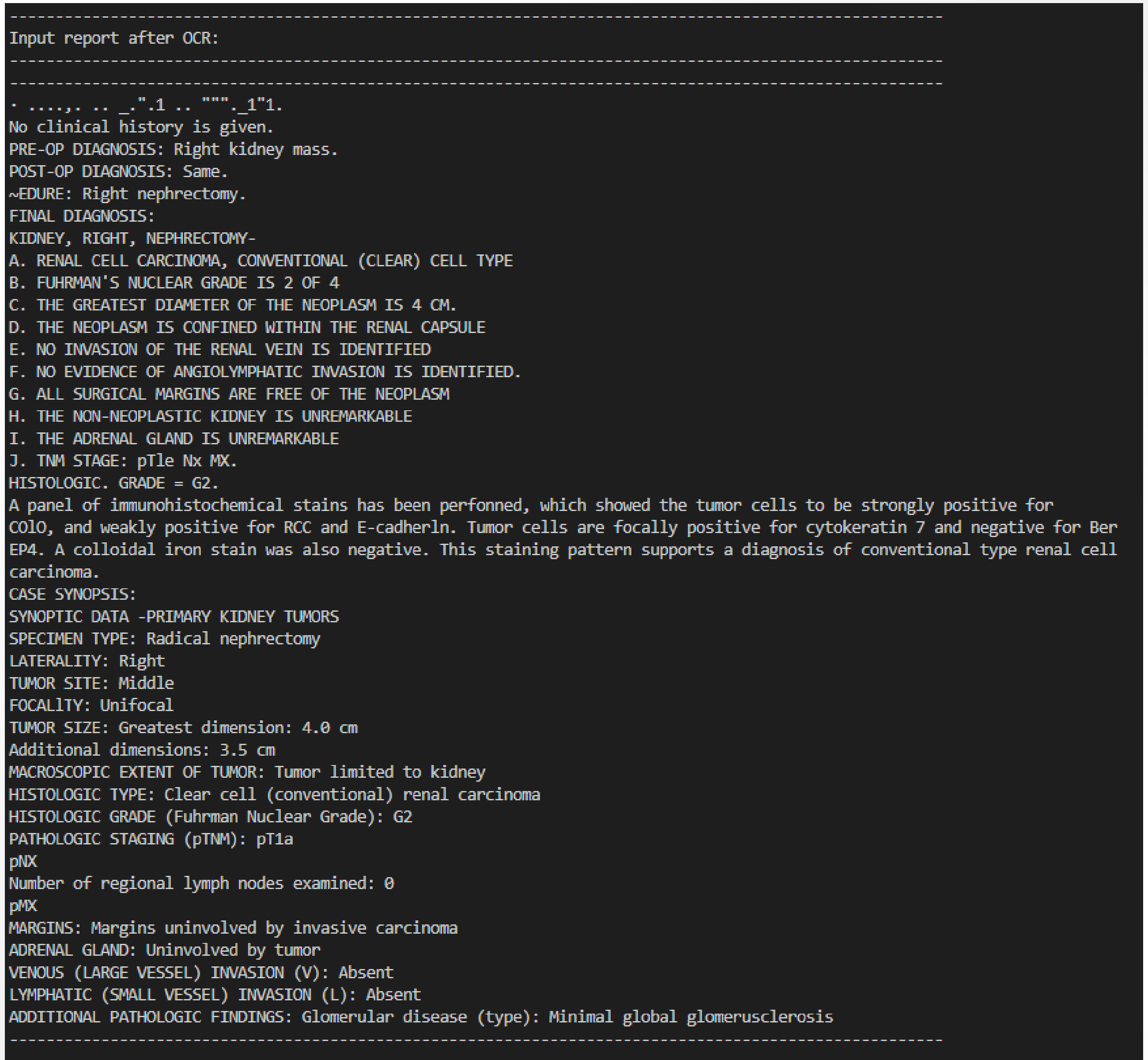
[redacted]

COMMENT:
A panel of immunohistochemical stains has been perfomed, which showed the tumor cells to be strongly positive for CD10, and weakly positive for RCC and E-cadherin. Tumor cells are focally positive for cytokeratin 7 and negative for Ber-EP4. A colloidal iron stain was also negative. This staining pattern supports a diagnosis of conventional type renal cell carcinoma.

CASE SYNOPSIS:
SYNOPTIC DATA - PRIMARY KIDNEY TUMORS

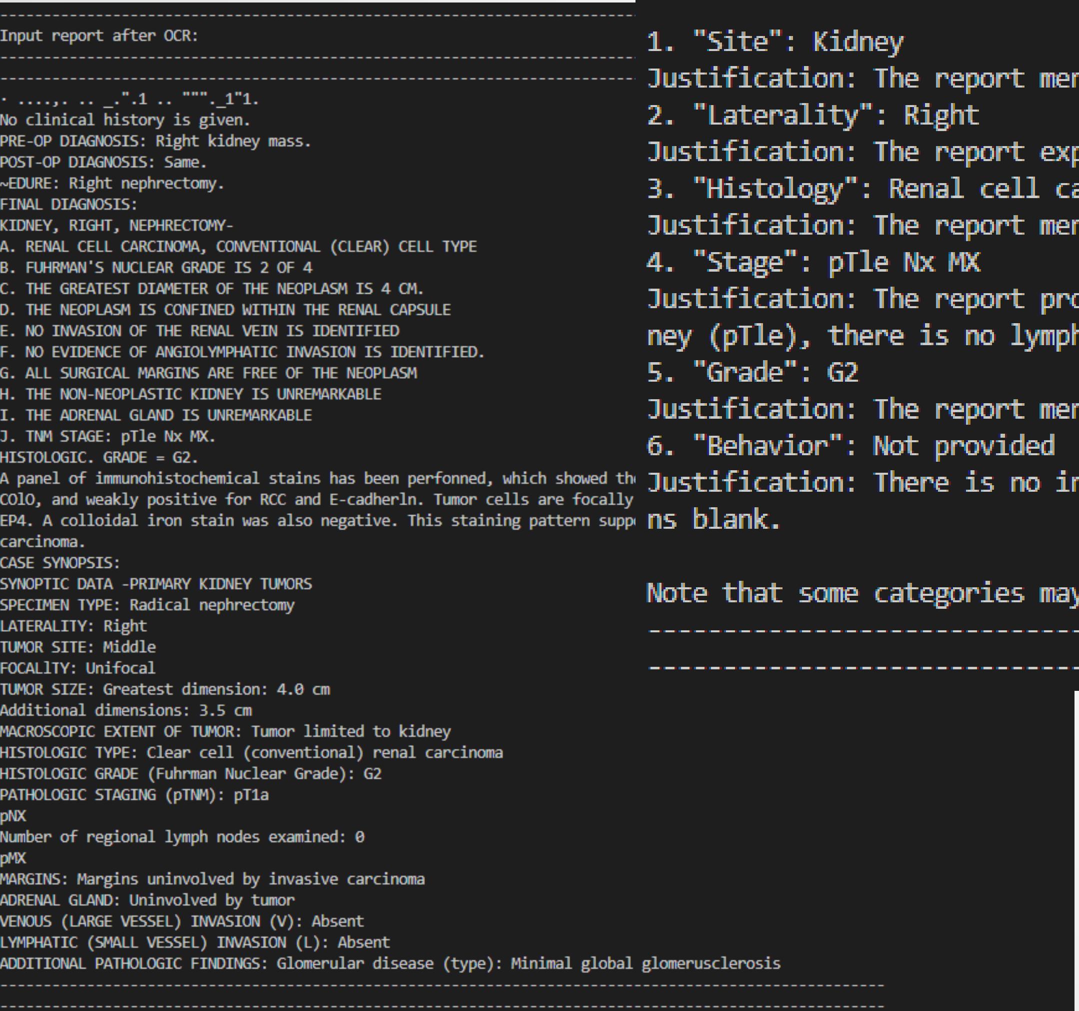
SPECIMEN TYPE:	Radical nephrectomy
LATERALITY:	Right
TUMOR SITE:	Middle
FOCALITY:	Unifocal
TUMOR SIZE:	Greatest dimension: 4.0 cm
	Additional dimensions: 3.5 cm
MACROSCOPIC EXTENT OF TUMOR:	Tumor limited to kidney
HISTOLOGIC TYPE:	Clear cell (conventional) renal carcinoma
HISTOLOGIC GRADE (Fuhrman Nuclear Grade):	G2
PATHOLOGIC STAGING (pTNM):	pT1a
	pNX
	Number of regional lymph nodes examined: 0
	pMX
MARGINS:	Margins uninvolved by invasive carcinoma
ADRENAL GLAND:	Uninvolved by tumor
VENOUS (LARGE VESSEL) INVASION (V):	Absent
LYMPHATIC (SMALL VESSEL) INVASION (L):	Absent
ADDITIONAL PATHOLOGIC FINDINGS:	Glomerular disease (type): Minimal global glomerusclerosis

[redacted]

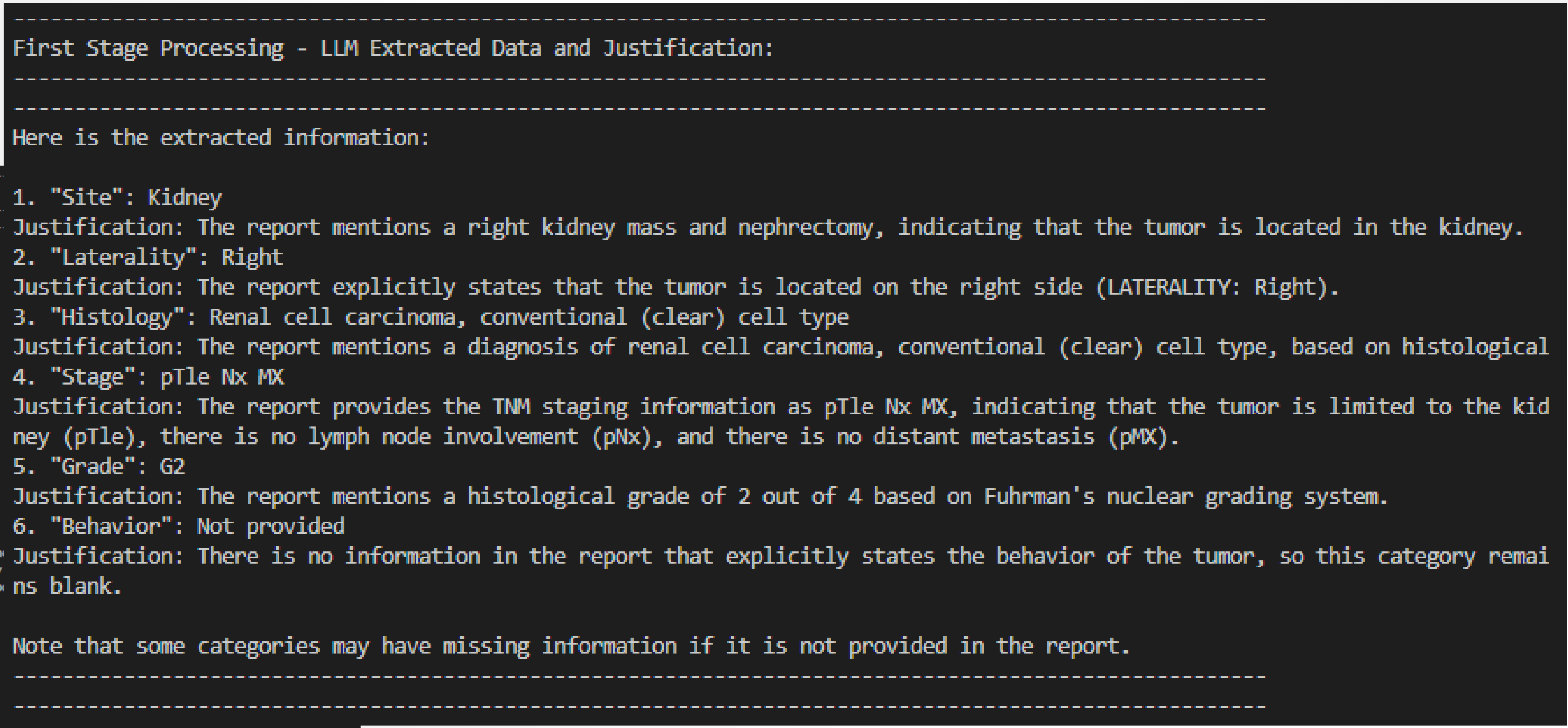


```
-----------------------------------------------------------------------------------------------------------
Input report after OCR:
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
· ....,. .. _.".1 .. """._1"1.
No clinical history is given.
PRE-OP DIAGNOSIS: Right kidney mass.
POST-OP DIAGNOSIS: Same.
~EDURE: Right nephrectomy.
FINAL DIAGNOSIS:
KIDNEY, RIGHT, NEPHRECTOMY-
A. RENAL CELL CARCINOMA, CONVENTIONAL (CLEAR) CELL TYPE
B. FUHRMAN'S NUCLEAR GRADE IS 2 OF 4
C. THE GREATEST DIAMETER OF THE NEOPLASM IS 4 CM.
D. THE NEOPLASM IS CONFINED WITHIN THE RENAL CAPSULE
E. NO INVASION OF THE RENAL VEIN IS IDENTIFIED
F. NO EVIDENCE OF ANGIOLYMPHATIC INVASION IS IDENTIFIED.
G. ALL SURGICAL MARGINS ARE FREE OF THE NEOPLASM
H. THE NON-NEOPLASTIC KIDNEY IS UNREMARKABLE
I. THE ADRENAL GLAND IS UNREMARKABLE
J. TNM STAGE: pTle Nx MX.
HISTOLOGIC. GRADE = G2.
A panel of immunohistochemical stains has been perfonned, which showed the tumor cells to be strongly positive for
C010, and weakly positive for RCC and E-cadherln. Tumor cells are focally positive for cytokeratin 7 and negative for Ber
EP4. A colloidal iron stain was also negative. This staining pattern supports a diagnosis of conventional type renal cell
carcinoma.
CASE SYNOPSIS:
SYNOPTIC DATA -PRIMARY KIDNEY TUMORS
SPECIMEN TYPE: Radical nephrectomy
LATERALITY: Right
TUMOR SITE: Middle
FOCALlTY: Unifocal
TUMOR SIZE: Greatest dimension: 4.0 cm
Additional dimensions: 3.5 cm
MACROSCOPIC EXTENT OF TUMOR: Tumor limited to kidney
HISTOLOGIC TYPE: Clear cell (conventional) renal carcinoma
HISTOLOGIC GRADE (Fuhrman Nuclear Grade): G2
PATHOLOGIC STAGING (pTNM): pT1a
pNX
Number of regional lymph nodes examined: 0
pMX
MARGINS: Margins uninvolved by invasive carcinoma
ADRENAL GLAND: Uninvolved by tumor
VENOUS (LARGE VESSEL) INVASION (V): Absent
LYMPHATIC (SMALL VESSEL) INVASION (L): Absent
ADDITIONAL PATHOLOGIC FINDINGS: Glomerular disease (type): Minimal global glomerusclerosis
-----------------------------------------------------------------------------------------------------------
```

Stage – I (Unstructured Data Extraction)

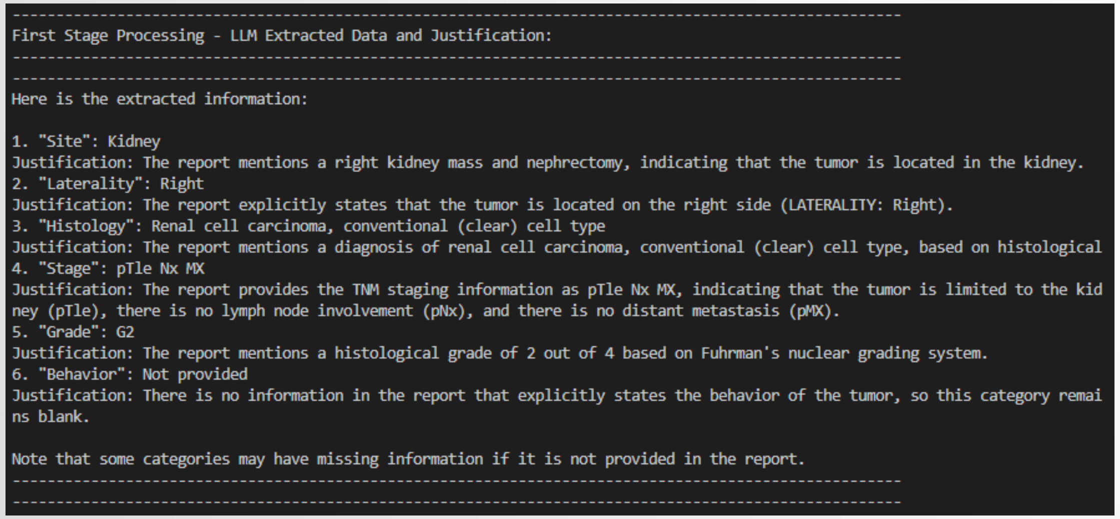


```
--------------------------------------------------------------------------
Input report after OCR:
--------------------------------------------------------------------------
--------------------------------------------------------------------------
· ....,. .. _.".1 .. """._1"1.
No clinical history is given.
PRE-OP DIAGNOSIS: Right kidney mass.
POST-OP DIAGNOSIS: Same.
~EDURE: Right nephrectomy.
FINAL DIAGNOSIS:
KIDNEY, RIGHT, NEPHRECTOMY-
A. RENAL CELL CARCINOMA, CONVENTIONAL (CLEAR) CELL TYPE
B. FUHRMAN'S NUCLEAR GRADE IS 2 OF 4
C. THE GREATEST DIAMETER OF THE NEOPLASM IS 4 CM.
D. THE NEOPLASM IS CONFINED WITHIN THE RENAL CAPSULE
E. NO INVASION OF THE RENAL VEIN IS IDENTIFIED
F. NO EVIDENCE OF ANGIOLYMPHATIC INVASION IS IDENTIFIED.
G. ALL SURGICAL MARGINS ARE FREE OF THE NEOPLASM
H. THE NON-NEOPLASTIC KIDNEY IS UNREMARKABLE
I. THE ADRENAL GLAND IS UNREMARKABLE
J. TNM STAGE: pTle Nx MX.
HISTOLOGIC. GRADE = G2.
A panel of immunohistochemical stains has been perfonned, which showed th
COlO, and weakly positive for RCC and E-cadherln. Tumor cells are focally
EP4. A colloidal iron stain was also negative. This staining pattern supp
carcinoma.
CASE SYNOPSIS:
SYNOPTIC DATA -PRIMARY KIDNEY TUMORS
SPECIMEN TYPE: Radical nephrectomy
LATERALITY: Right
TUMOR SITE: Middle
FOCALlTY: Unifocal
TUMOR SIZE: Greatest dimension: 4.0 cm
Additional dimensions: 3.5 cm
MACROSCOPIC EXTENT OF TUMOR: Tumor limited to kidney
HISTOLOGIC TYPE: Clear cell (conventional) renal carcinoma
HISTOLOGIC GRADE (Fuhrman Nuclear Grade): G2
PATHOLOGIC STAGING (pTNM): pT1a
pNX
Number of regional lymph nodes examined: 0
pMX
MARGINS: Margins uninvolved by invasive carcinoma
ADRENAL GLAND: Uninvolved by tumor
VENOUS (LARGE VESSEL) INVASION (V): Absent
LYMPHATIC (SMALL VESSEL) INVASION (L): Absent
ADDITIONAL PATHOLOGIC FINDINGS: Glomerular disease (type): Minimal global glomerusclerosis
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
```

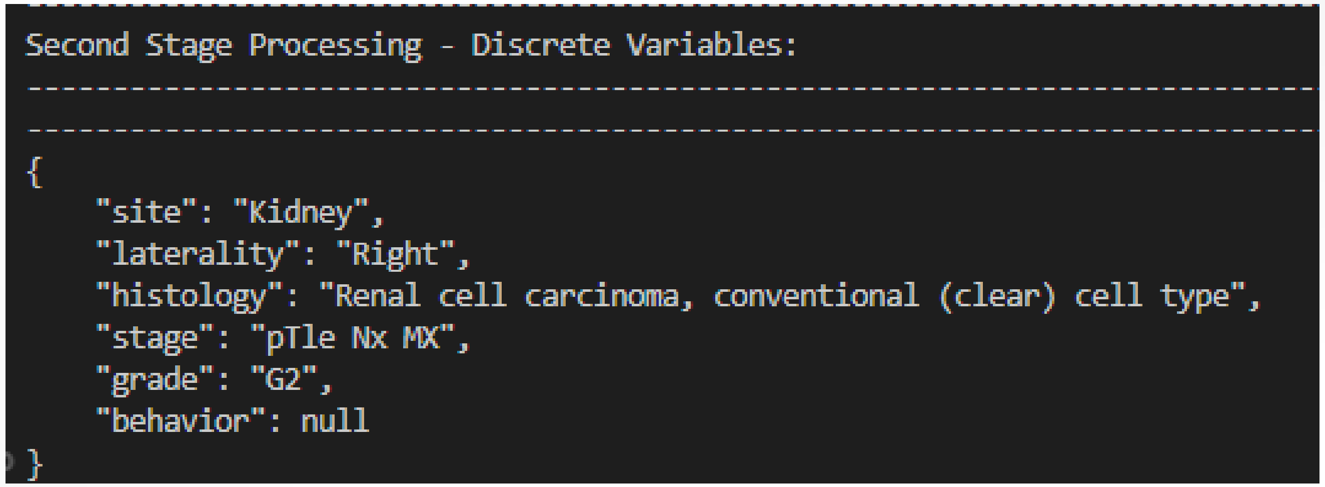


```
--------------------------------------------------------------------------------
First Stage Processing - LLM Extracted Data and Justification:
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Here is the extracted information:

1. "Site": Kidney
Justification: The report mentions a right kidney mass and nephrectomy, indicating that the tumor is located in the kidney.
2. "Laterality": Right
Justification: The report explicitly states that the tumor is located on the right side (LATERALITY: Right).
3. "Histology": Renal cell carcinoma, conventional (clear) cell type
Justification: The report mentions a diagnosis of renal cell carcinoma, conventional (clear) cell type, based on histological
4. "Stage": pTle Nx MX
Justification: The report provides the TNM staging information as pTle Nx MX, indicating that the tumor is limited to the kid
ney (pTle), there is no lymph node involvement (pNx), and there is no distant metastasis (pMX).
5. "Grade": G2
Justification: The report mentions a histological grade of 2 out of 4 based on Fuhrman's nuclear grading system.
6. "Behavior": Not provided
Justification: There is no information in the report that explicitly states the behavior of the tumor, so this category remai
ns blank.

Note that some categories may have missing information if it is not provided in the report.
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

Stage-II Processing (JSON Object)

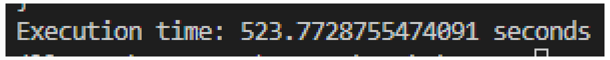


```
------------------------------------------------------------------------------------------
First Stage Processing - LLM Extracted Data and Justification:
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
Here is the extracted information:

1. "Site": Kidney
Justification: The report mentions a right kidney mass and nephrectomy, indicating that the tumor is located in the kidney.
2. "Laterality": Right
Justification: The report explicitly states that the tumor is located on the right side (LATERALITY: Right).
3. "Histology": Renal cell carcinoma, conventional (clear) cell type
Justification: The report mentions a diagnosis of renal cell carcinoma, conventional (clear) cell type, based on histological
4. "Stage": pTle Nx MX
Justification: The report provides the TNM staging information as pTle Nx MX, indicating that the tumor is limited to the kid
ney (pTle), there is no lymph node involvement (pNx), and there is no distant metastasis (pMX).
5. "Grade": G2
Justification: The report mentions a histological grade of 2 out of 4 based on Fuhrman's nuclear grading system.
6. "Behavior": Not provided
Justification: There is no information in the report that explicitly states the behavior of the tumor, so this category remai
ns blank.

Note that some categories may have missing information if it is not provided in the report.
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
```



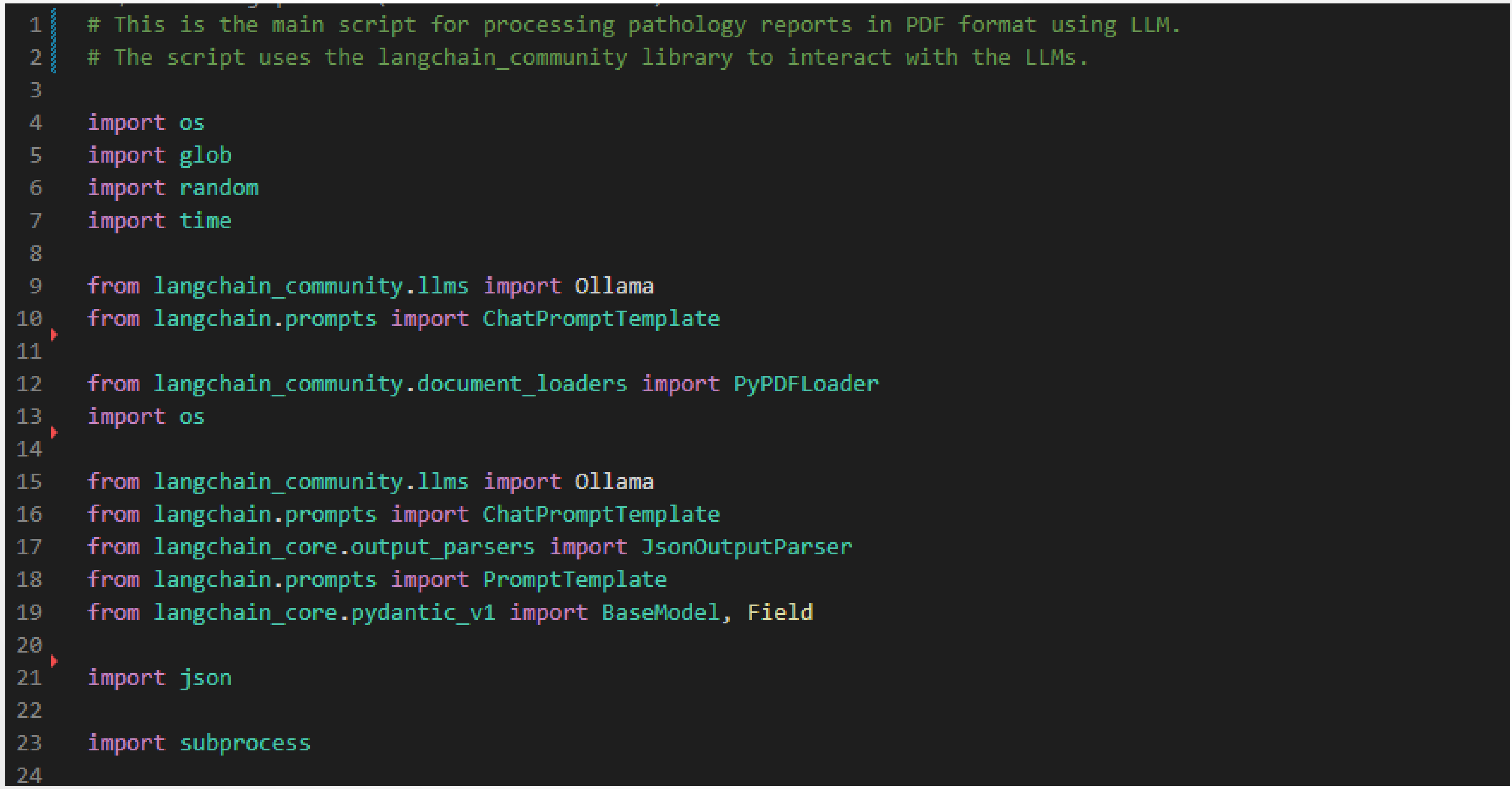
```
Second Stage Processing - Discrete Variables:
-----------------------------------------------------------------
-----------------------------------------------------------------
{
    "site": "Kidney",
    "laterality": "Right",
    "histology": "Renal cell carcinoma, conventional (clear) cell type",
    "stage": "pTle Nx MX",
    "grade": "G2",
    "behavior": null
}
```

- Laptop CPU execution time: < 9 minutes



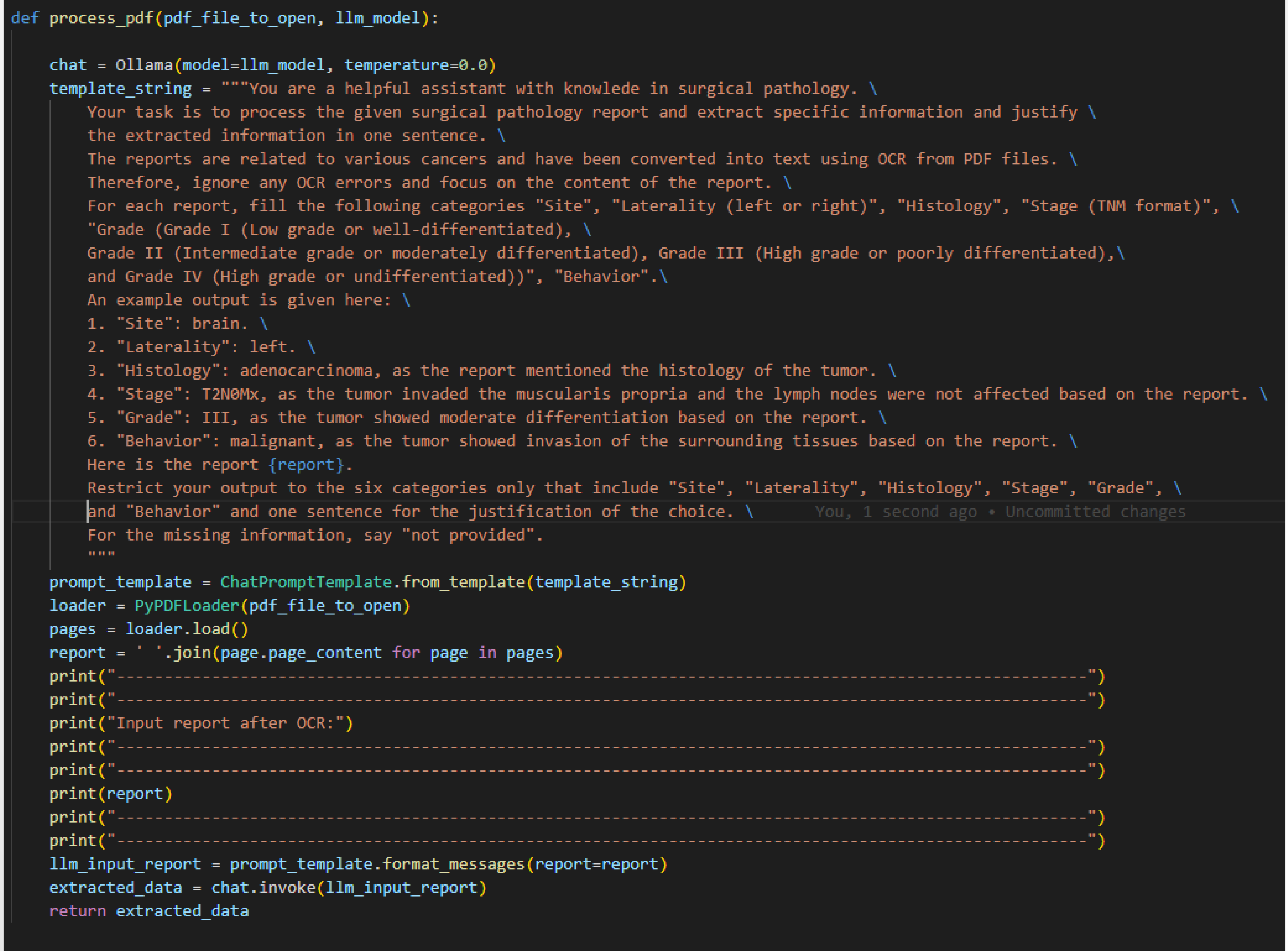
```
Execution time: 523.7728755474091 seconds
```

Imports



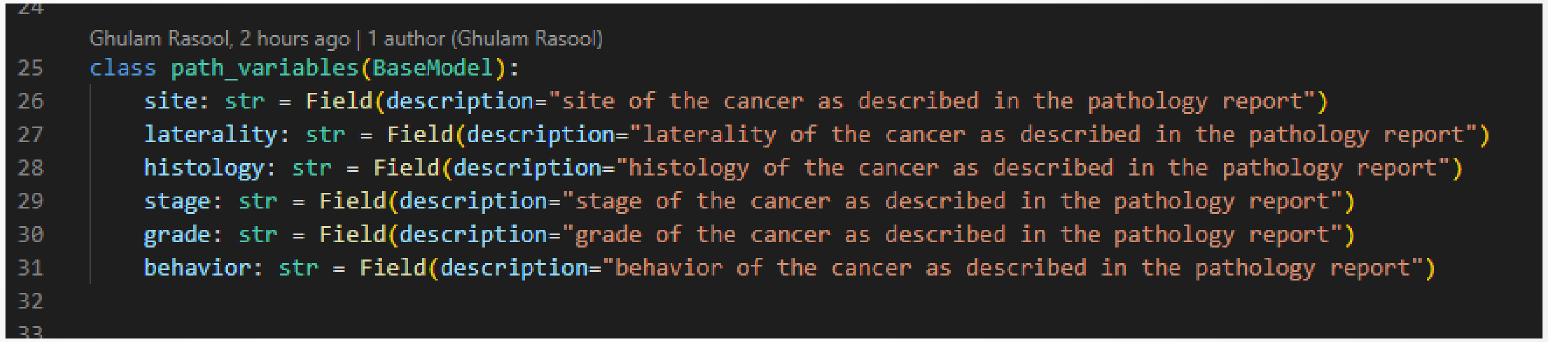
```python
# This is the main script for processing pathology reports in PDF format using LLM.
# The script uses the langchain_community library to interact with the LLMs.

import os
import glob
import random
import time

from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate

from langchain_community.document_loaders import PyPDFLoader
import os

from langchain_community.llms import Ollama
from langchain.prompts import ChatPromptTemplate
from langchain_core.output_parsers import JsonOutputParser
from langchain.prompts import PromptTemplate
from langchain_core.pydantic_v1 import BaseModel, Field

import json

import subprocess

```

Stage-I

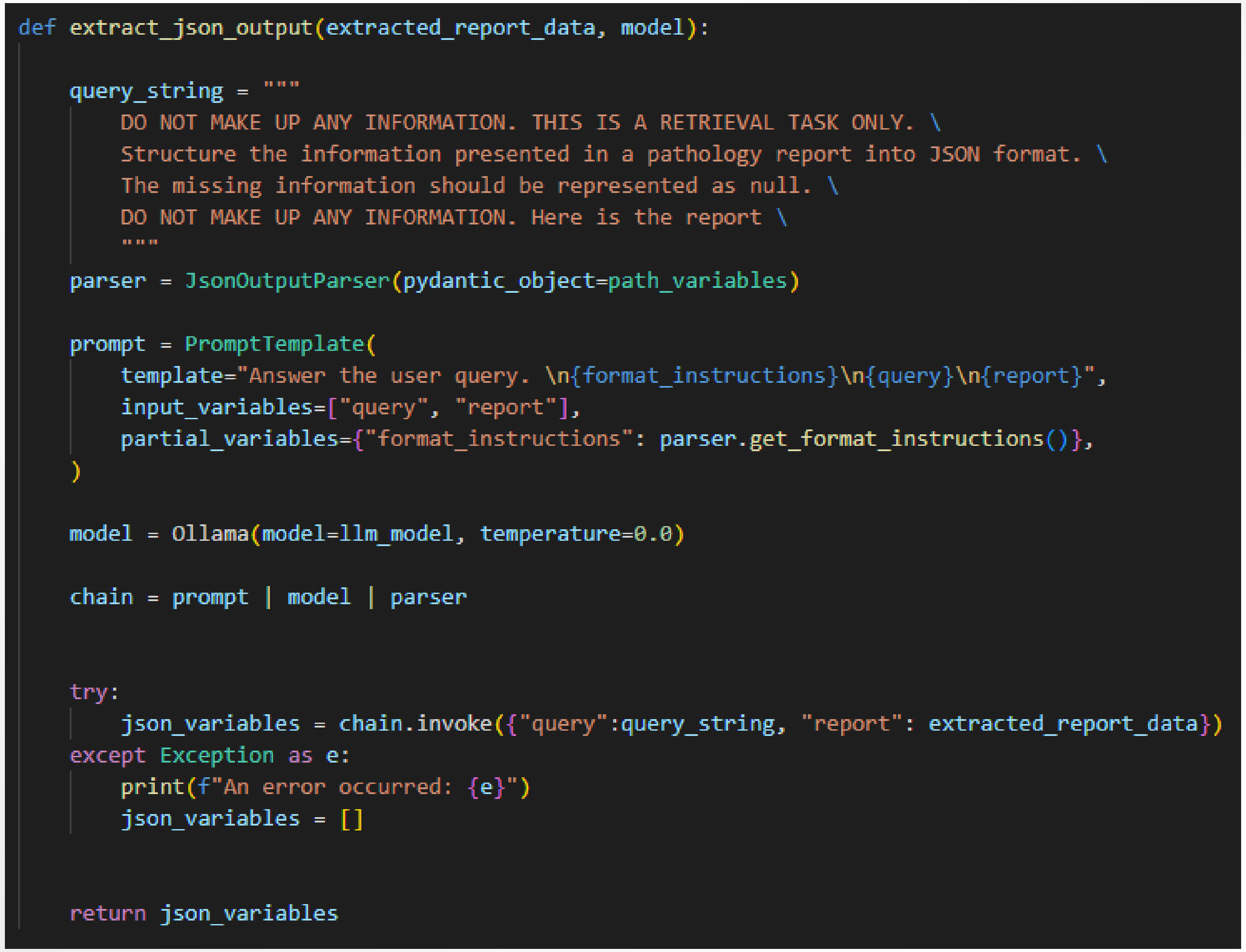


```
def process_pdf(pdf_file_to_open, llm_model):

    chat = Ollama(model=llm_model, temperature=0.0)
    template_string = """You are a helpful assistant with knowlede in surgical pathology. \
        Your task is to process the given surgical pathology report and extract specific information and justify \
        the extracted information in one sentence. \
        The reports are related to various cancers and have been converted into text using OCR from PDF files. \
        Therefore, ignore any OCR errors and focus on the content of the report. \
        For each report, fill the following categories "Site", "Laterality (left or right)", "Histology", "Stage (TNM format)", \
        "Grade (Grade I (Low grade or well-differentiated), \
        Grade II (Intermediate grade or moderately differentiated), Grade III (High grade or poorly differentiated),\
        and Grade IV (High grade or undifferentiated))", "Behavior".\
        An example output is given here: \
        1. "Site": brain. \
        2. "Laterality": left. \
        3. "Histology": adenocarcinoma, as the report mentioned the histology of the tumor. \
        4. "Stage": T2N0Mx, as the tumor invaded the muscularis propria and the lymph nodes were not affected based on the report. \
        5. "Grade": III, as the tumor showed moderate differentiation based on the report. \
        6. "Behavior": malignant, as the tumor showed invasion of the surrounding tissues based on the report. \
        Here is the report {report}.
        Restrict your output to the six categories only that include "Site", "Laterality", "Histology", "Stage", "Grade", \
        and "Behavior" and one sentence for the justification of the choice. \      You, 1 second ago • Uncommitted changes
        For the missing information, say "not provided".
        """
    prompt_template = ChatPromptTemplate.from_template(template_string)
    loader = PyPDFLoader(pdf_file_to_open)
    pages = loader.load()
    report = ' '.join(page.page_content for page in pages)
    print("-------------------------------------------------------------------------------------------------------")
    print("-------------------------------------------------------------------------------------------------------")
    print("Input report after OCR:")
    print("-------------------------------------------------------------------------------------------------------")
    print("-------------------------------------------------------------------------------------------------------")
    print(report)
    print("-------------------------------------------------------------------------------------------------------")
    print("-------------------------------------------------------------------------------------------------------")
    llm_input_report = prompt_template.format_messages(report=report)
    extracted_data = chat.invoke(llm_input_report)
    return extracted_data
```

Stage-II

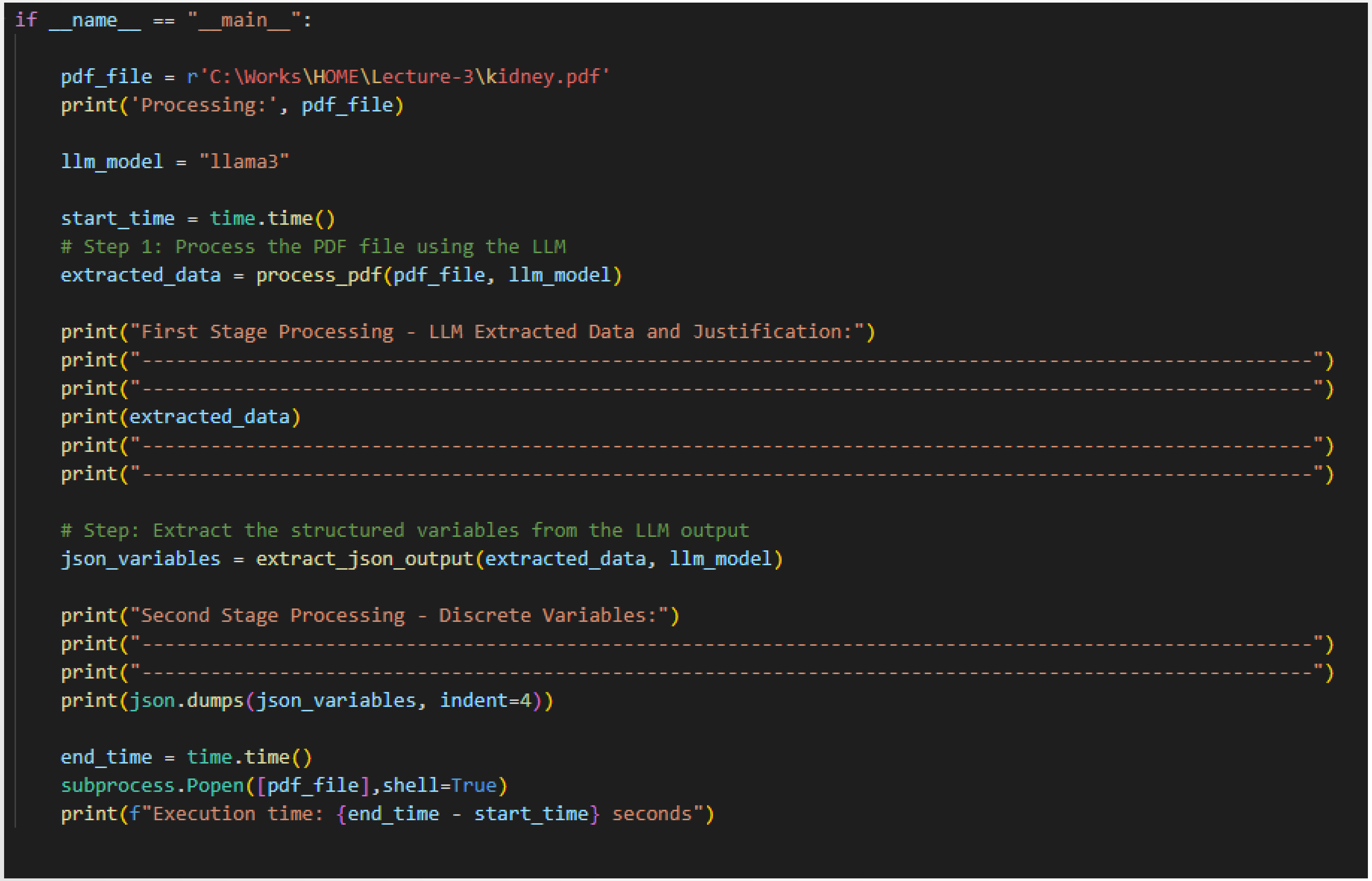


```
Ghulam Rasool, 2 hours ago | 1 author (Ghulam Rasool)
class path_variables(BaseModel):
    site: str = Field(description="site of the cancer as described in the pathology report")
    laterality: str = Field(description="laterality of the cancer as described in the pathology report")
    histology: str = Field(description="histology of the cancer as described in the pathology report")
    stage: str = Field(description="stage of the cancer as described in the pathology report")
    grade: str = Field(description="grade of the cancer as described in the pathology report")
    behavior: str = Field(description="behavior of the cancer as described in the pathology report")
```



```
def extract_json_output(extracted_report_data, model):

    query_string = """
        DO NOT MAKE UP ANY INFORMATION. THIS IS A RETRIEVAL TASK ONLY. \
        Structure the information presented in a pathology report into JSON format. \
        The missing information should be represented as null. \
        DO NOT MAKE UP ANY INFORMATION. Here is the report \
        """
    parser = JsonOutputParser(pydantic_object=path_variables)

    prompt = PromptTemplate(
        template="Answer the user query. \n{format_instructions}\n{query}\n{report}",
        input_variables=["query", "report"],
        partial_variables={"format_instructions": parser.get_format_instructions()},
    )

    model = Ollama(model=llm_model, temperature=0.0)

    chain = prompt | model | parser


    try:
        json_variables = chain.invoke({"query":query_string, "report": extracted_report_data})
    except Exception as e:
        print(f"An error occurred: {e}")
        json_variables = []


    return json_variables
```

Main Function



```python
if __name__ == "__main__":

    pdf_file = r'C:\Works\HOME\Lecture-3\kidney.pdf'
    print('Processing:', pdf_file)

    llm_model = "llama3"

    start_time = time.time()
    # Step 1: Process the PDF file using the LLM
    extracted_data = process_pdf(pdf_file, llm_model)

    print("First Stage Processing - LLM Extracted Data and Justification:")
    print("------------------------------------------------------------------------------------------------------")
    print("------------------------------------------------------------------------------------------------------")
    print(extracted_data)
    print("------------------------------------------------------------------------------------------------------")
    print("------------------------------------------------------------------------------------------------------")

    # Step: Extract the structured variables from the LLM output
    json_variables = extract_json_output(extracted_data, llm_model)

    print("Second Stage Processing - Discrete Variables:")
    print("------------------------------------------------------------------------------------------------------")
    print("------------------------------------------------------------------------------------------------------")
    print(json.dumps(json_variables, indent=4))

    end_time = time.time()
    subprocess.Popen([pdf_file],shell=True)
    print(f"Execution time: {end_time - start_time} seconds")
```

Questions