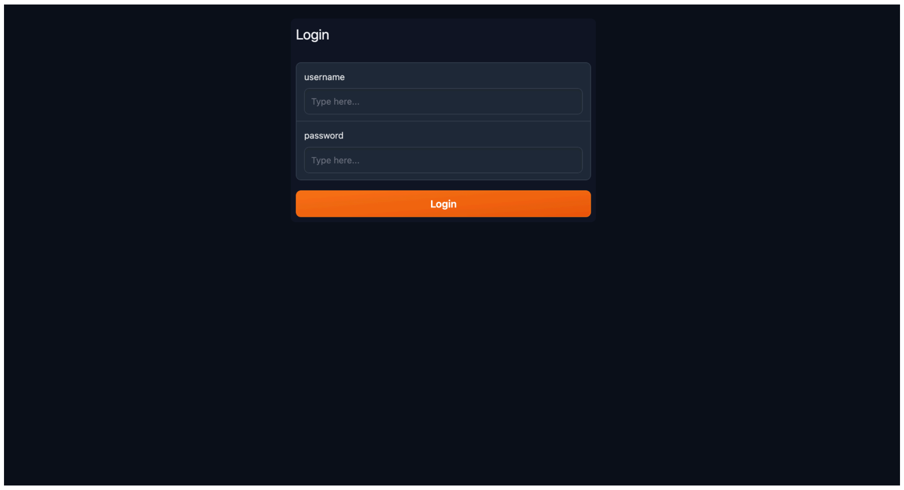
QNA Chatbot Guides

Name: Ketut Toto Suryahadinata
Role: AI Engineer

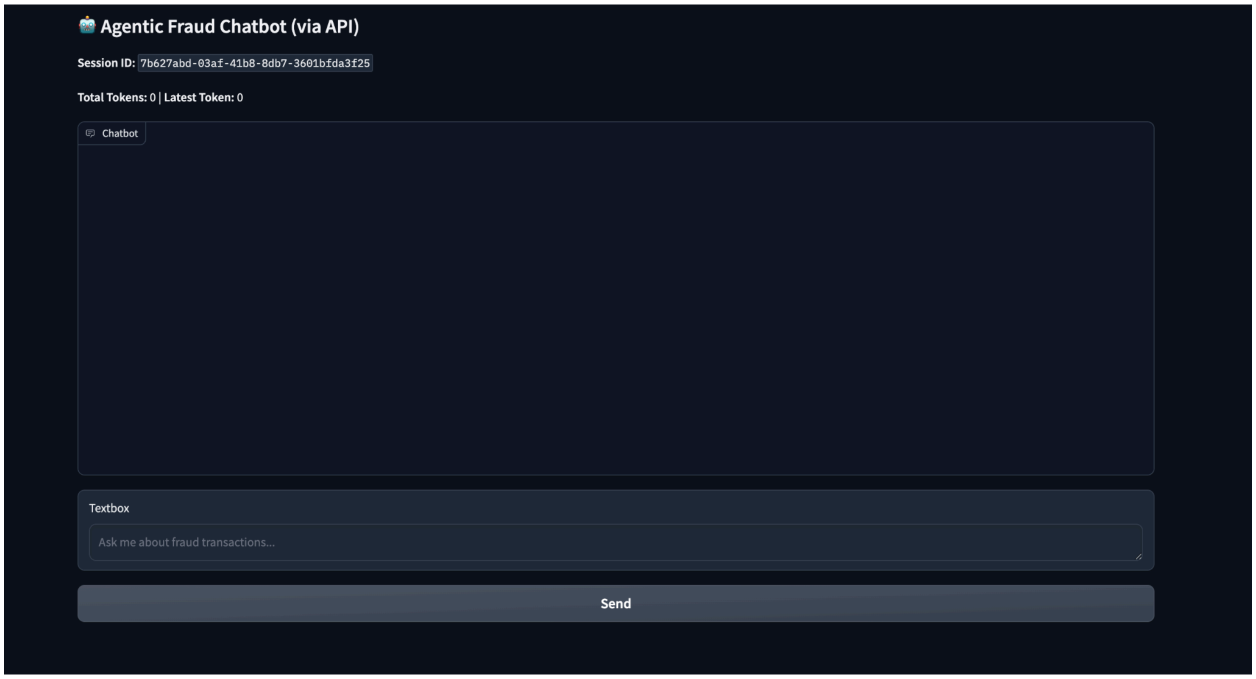
Application Credentials

Web

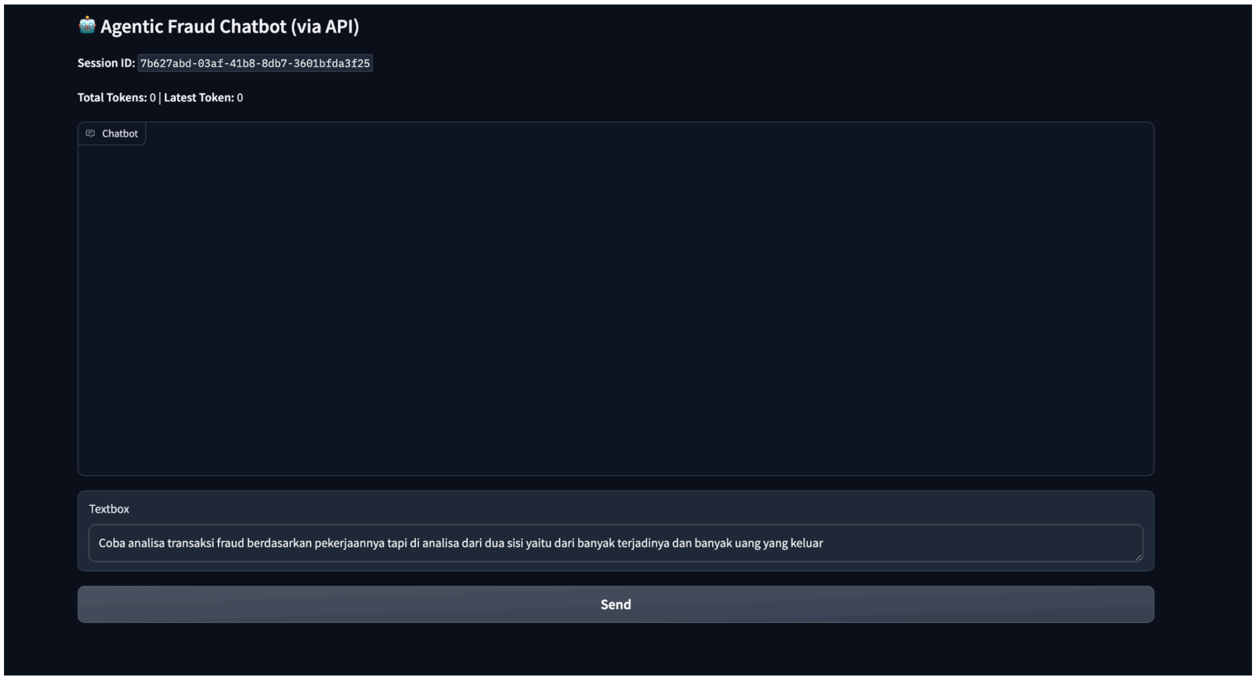
1. Go to: http://18.142.237.65:7860/ (i am sorry i dont have domain and ssl :( )



2. Username: mekari_test
   Password: kmzwa8awaa
3. Empty page


Agentic Fraud Chatbot (via API)
Session ID: 7b627abd-03af-41b8-8db7-3601bfda3f25
Total Tokens: 0 | Latest Token: 0
Chatbot
Textbox
Ask me about fraud transactions...
Send

4. Start chatting (maximum history 5, my openai balance only 3$:(   )


Agentic Fraud Chatbot (via API)
Session ID: 7b627abd-03af-41b8-8db7-3601bfda3f25
Total Tokens: 0 | Latest Token: 0
Chatbot
Textbox
Send

Coba analisa transaksi fraud berdasarkan pekerjaannya tapi di analisa dari dua sisi yaitu dari banyak terjadinya dan banyak uang yang keluar

5. Starter Question:
   a. “Coba analisa transaksi fraud berdasarkan pekerjaannya tapi di analisa dari dua sisi yaitu dari banyak terjadinya dan banyak uang yang keluar”
   b. “Biasanya penipuan kartu kredit itu modusnya gimana?”
   c. “Bisakah kamu memberi ku resep udang keju”
6. Language is following the question, but it is either Indonesia or English

Services
You can check all available api documentation in http://18.142.237.65:8000/docs

Technologies Used

Services

- Tools: FastAPI
- Reason: Fastest way to develop API and the documentation in swagger mode is really good
- Best Choice: Rust is better and faster but in the pdf, user prefer python

Web

- Tools: Gradio
- Reason: Fastest way to develop simple UI + have many templates
- Best Choice: React can be really good choice but the development will take much longer

VectorDB

- Tools: ChromaDB
- Reason: Only one document used so i don’t really need dedicated vector / self hosted vector,
- Best Choice: ChromaDB still the best choice for single document

TableDB

- Tools: Supabase
- Reason: Free
- Best Choice: AWS RDS or Azure Data still better in supabase but its not free

Session and Caching

- Tools: In Memory Fast API
- Reason: Fast
- Best Choice: Using redis will be great but it will be too complex for job entrance testing

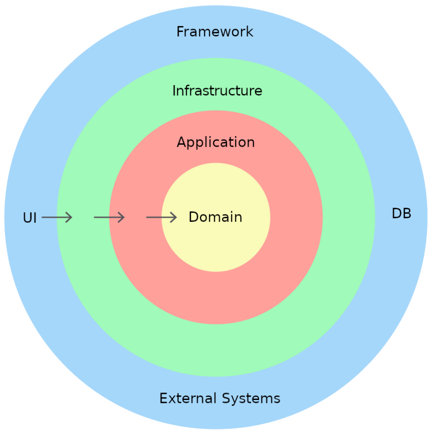
Folder Structure Guide

General Structure

Main folders consist of 4 folders:

1. instruction -> Consist file of instruction sent by email by HR
2. raw_dataset -> Bhatia.pdf is there and the csv (excluded due to large size)
3. Services -> The backend of the apps
4. Web -> The Gradio App

Services Structure



Services folder structure following DDD (Domain Driven Design) consist of

1. Cmd: for command line code
2. Contract: Consist of global data type like response for api and error type for api
3. Handler: Consist of handler logic like for logging or routing the api (rest)
4. Business: Consist of business logic
    a. Domain: Code of connector (like supabase connector, vectordb connector, aws connector, etc)
    b. Model: Consist of data type that later used in handler/rest
    c. Usecase: Business logic
5. Middleware: Like its name, middlewares are put in here
6. Extra: Code for storing csv to supabase and storing pdf into vector db
7. Chroma_data: embedded data
8. Transformed_data: Pdf data that is already transformed into different meaningful form (page index form)
9. Agentic: Haystack logic code

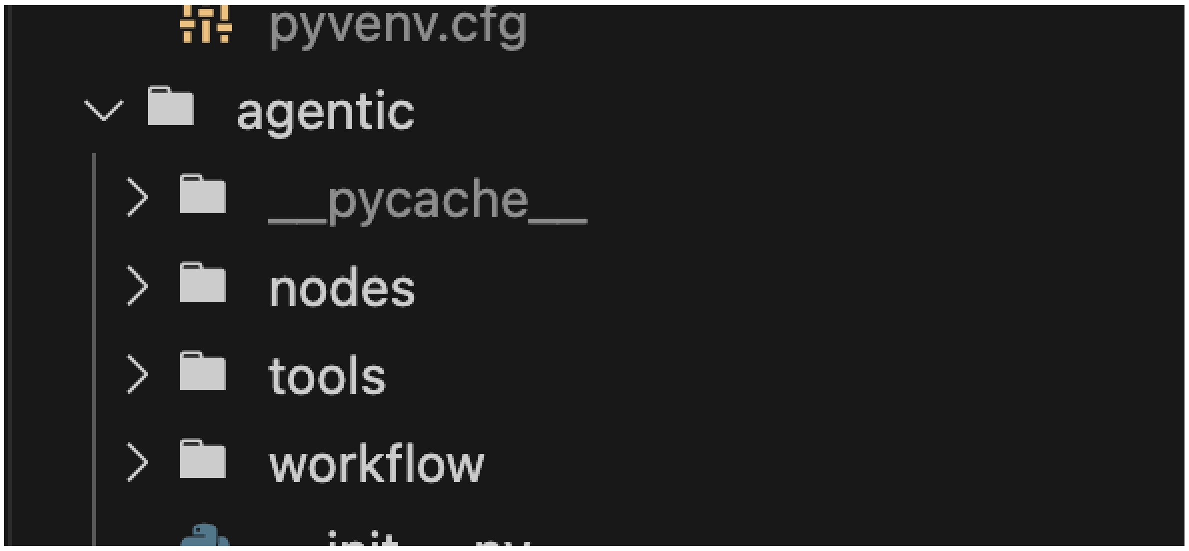
Algorithm Guide

Agentic Guide

- Tools: Haystack AI
- Reason: I am getting used to use it + i don't like langchain because of the spaghetti architecture
- Best Choice: If i am in work, 100% using RAG Flow or N8N or Diffy will pretty much easier and faster, but due to testing code competency in here, I will code the agent myself

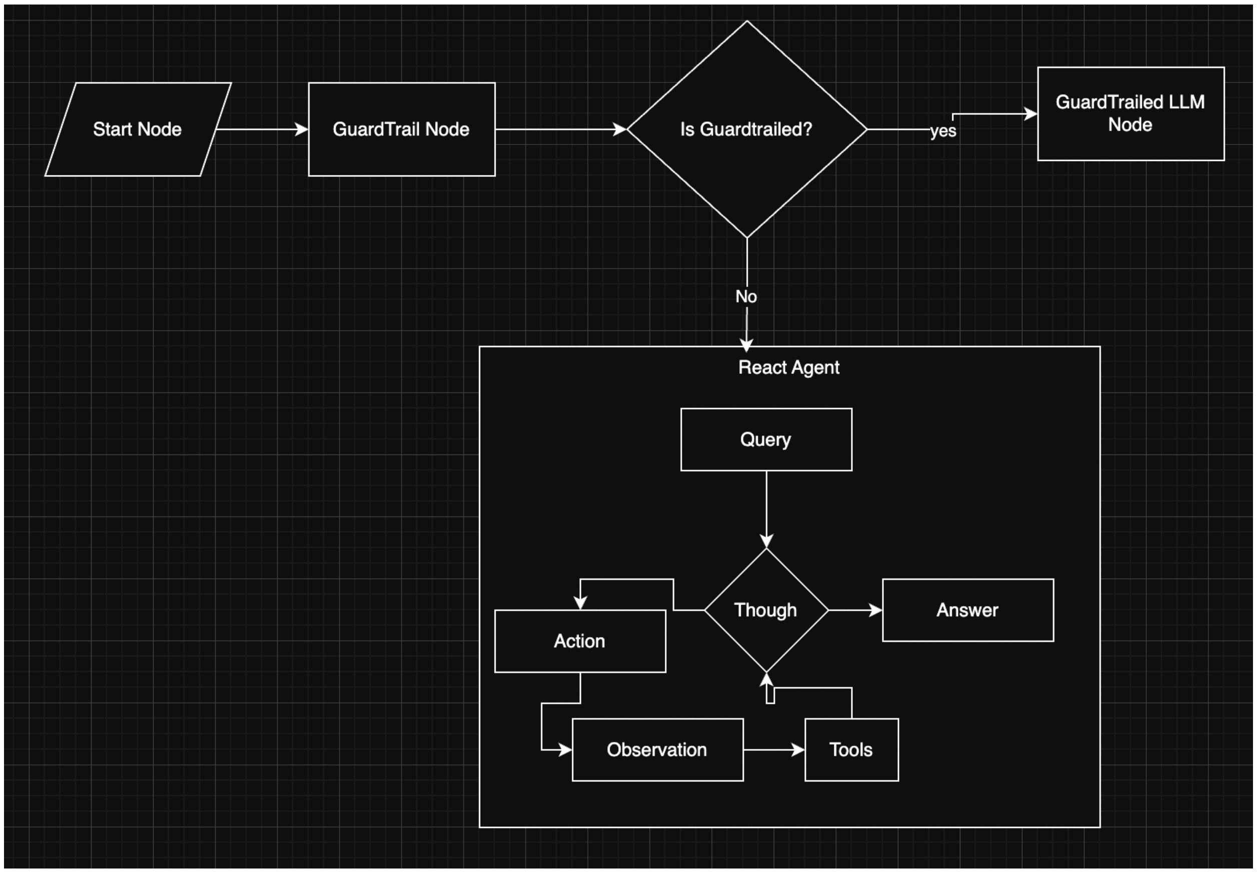
Structure Guide

Agent can be found inside *Services/agentic* , the concept was inspired by N8N and Diffy which using Node Base, Pipeline, and Workflow



```
    pyvenv.cfg
  agentic
    __pycache__
    nodes
    tools
    workflow
```

The agent pipeline should look like this:



The baseline of retrieving data is by using react agent that can iterate to maximum N iteration

Nodes
I develop nodes into 6 pieces (can be more):

- Start_nodes -> all initiation input goes here
- Llm_nodes -> in charge of answering question
- Guradtrail_nodes -> actually the llm_nodes but special for guardrail
- routing _nodes -> for branching and conditioning
- Converstion_nodes -> Nodes for converting output of A node into output that can be accepted for B Node
- Answer_nodes -> All the print output is in here

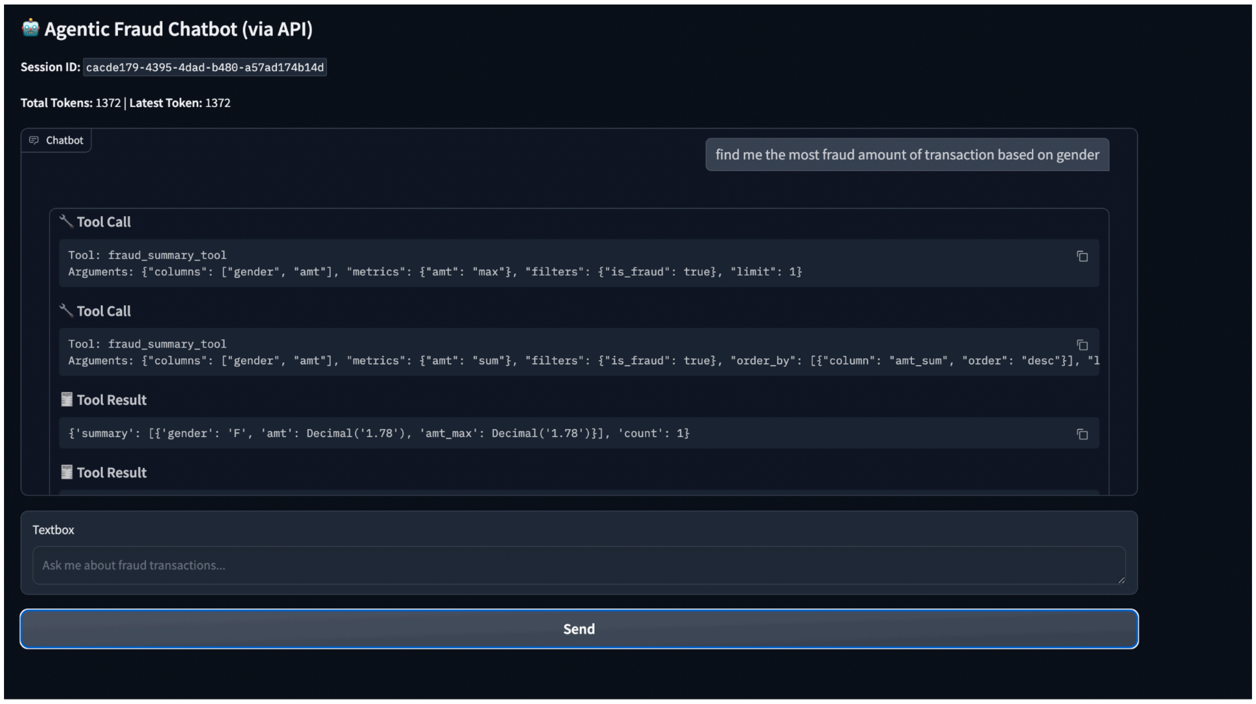
Workflow
I develop only one workflow (QnA workflow) (see above image of the flow)

Retrieving methods

1. Table data

There are so many ways to retrieving data from table data, one of the most popular is text to sql and tabular RAG, but in this case i create 2 tools in purpose of **getting data and getting statistics**. The reason why i was not developing text to sql actually because there are so many error cases and it is hard to direct LLM based on query to generate good quality of sql codes.

The maestro of utilization of this tools is react agent that can iterate ad thinking about result of tools provided.



Agentic Fraud Chatbot (via API)

Session ID: `cacde179-4395-4dad-b480-a57ad174b14d`

Total Tokens: 1372 | **Latest Token:** 1372

Chatbot

find me the most fraud amount of transaction based on gender

Tool Call

```
Tool: fraud_summary_tool
Arguments: {"columns": ["gender", "amt"], "metrics": {"amt": "max"}, "filters": {"is_fraud": true}, "limit": 1}
```

Tool Call

```
Tool: fraud_summary_tool
Arguments: {"columns": ["gender", "amt"], "metrics": {"amt": "sum"}, "filters": {"is_fraud": true}, "order_by": [{"column": "amt_sum", "order": "desc"}], "l
```

Tool Result

```
{'summary': [{'gender': 'F', 'amt': Decimal('1.78'), 'amt_max': Decimal('1.78')}], 'count': 1}
```

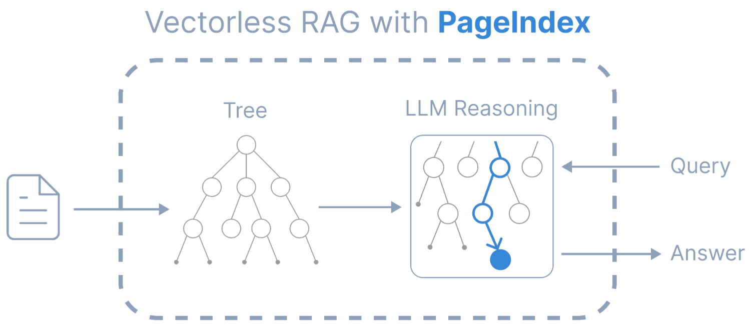
Tool Result

Textbox

Ask me about fraud transactions...

Send

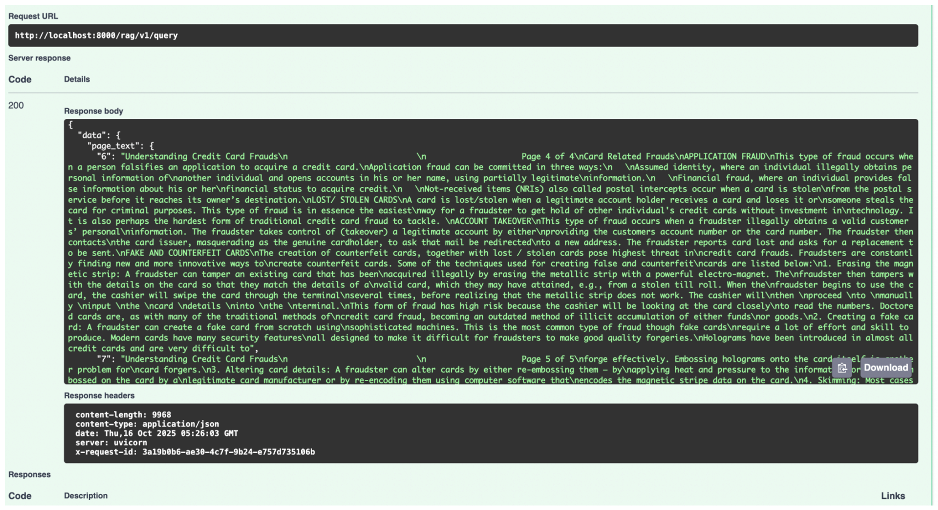
2. Pdf content
   For pdf content i use conventional RAG with cosine similarity with OpenAI Embedder large. The most important algorithm in here is not how do I do RAG, but how do I embed the pdf and chunking itself. I was using a technique called tree-based chunking



The flow is like

1. Transform pdf into json format with node and page index format (see https://github.com/Verietoto/mekari-mle-test/blob/main/services/transformed_data/Bhatla_structure.json)
2. Embed either summary or title for maximum level = 3 (maximum depth is 3)

3. Do similarity on level 3 and pickup the parent nodes
4. Parse PDF based on index from retrieval (it will much faster if we have nosql or s3 for saving parsed pdf based on pages, but due to lack of resource we just parsing the pdf everytime we need the tool). The result should like below image



Request URL

```
http://localhost:8000/rag/v1/query
```

Server response

Code Details

200 Response body

```
{
  "data": {
    "page_text": {
      "6": "Understanding Credit Card Frauds\n                              \n                    Page 4 of 4\nCard Related Frauds\nAPPLICATION FRAUD\nThis type of fraud occurs whe
n a person falsifies an application to acquire a credit card.\nApplication fraud can be committed in three ways:\n   \nAssumed identity, where an individual illegally obtains pe
rsonal information of\nanother individual and opens accounts in his or her name, using partially legitimate\ninformation.\n   \nFinancial fraud, where an individual provides fal
se information about his or her\nfinancial status to acquire credit.\n   \nNot-received items (NRIs) also called postal intercepts occur when a card is stolen\nfrom the postal s
ervice before it reaches its owner’s destination.\nLOST/ STOLEN CARDS\nA card is lost/stolen when a legitimate account holder receives a card and loses it or\nsomeone steals the
card for criminal purposes. This type of fraud is in essence the easiest\nway for a fraudster to get hold of other individual's credit cards without investment in\ntechnology. I
t is also perhaps the hardest form of traditional credit card fraud to tackle. \nACCOUNT TAKEOVER\nThis type of fraud occurs when a fraudster illegally obtains a valid customer
s’ personal\ninformation. The fraudster takes control of (takeover) a legitimate account by either\nproviding the customers account number or the card number. The fraudster then
contacts\nthe card issuer, masquerading as the genuine cardholder, to ask that mail be redirected\nto a new address. The fraudster reports card lost and asks for a replacement t
o be sent.\nFAKE AND COUNTERFEIT CARDS\nThe creation of counterfeit cards, together with lost / stolen cards pose highest threat in\ncredit card frauds. Fraudsters are constantl
y finding new and more innovative ways to\ncreate counterfeit cards. Some of the techniques used for creating false and counterfeit\ncards are listed below:\n1. Erasing the magn
etic strip: A fraudster can tamper an existing card that has been\nacquired illegally by erasing the metallic strip with a powerful electro-magnet. The\nfraudster then tampers w
ith the details on the card so that they match the details of a\nvalid card, which they may have attained, e.g., from a stolen till roll. When the\nfraudster begins to use the c
ard, the cashier will swipe the card through the terminal\nseveral times, before realizing that the metallic strip does not work. The cashier will\nthen \nproceed \nto \nmanuall
y \ninput \nthe \ncard \ndetails \ninto \nthe \nterminal.\nThis form of fraud has high risk because the cashier will be looking at the card closely\nto read the numbers. Doctore
d cards are, as with many of the traditional methods of\ncredit card fraud, becoming an outdated method of illicit accumulation of either funds\nor goods.\n2. Creating a fake ca
rd: A fraudster can create a fake card from scratch using\nsophisticated machines. This is the most common type of fraud though fake cards\nrequire a lot of effort and skill to
produce. Modern cards have many security features\nall designed to make it difficult for fraudsters to make good quality forgeries.\nHolograms have been introduced in almost all
credit cards and are very difficult to",
      "7": "Understanding Credit Card Frauds\n                              \n                    Page 5 of 5\nforge effectively. Embossing holograms onto the card itself is anothe
r problem for\ncard forgers.\n3. Altering card details: A fraudster can alter cards by either re-embossing them – by\napplying heat and pressure to the informat[illegible]
bossed on the card by a\nlegitimate card manufacturer or by re-encoding them using computer software that\nencodes the magnetic stripe data on the card.\n4. Skimming: Most cases
```

Response headers

```
content-length: 9968
content-type: application/json
date: Thu,16 Oct 2025 05:26:03 GMT
server: uvicorn
x-request-id: 3a19b0b6-ae30-4c7f-9b24-e757d735106b
```

Responses

Code Description Links

5. Feed the rag result into LLM

Caveat

1. Tokens calculated is input and output token (since i am using react agent, passing tools must cost some tokens but Haystack do not account for that, so I just account for that , there is a limitation of using open source library though)
2. LLM instability is really hard to deal with, but I am sure can make this better if have more time and more team to brainstorm idea
3. Server will be shut down in Oct 20 2025

Looking forward to work with you guys, i hope my work is sufficient to join in Mekari Team :)