Problem Statement:

https://github.com/codingo/Ransomware-Json-Dataset/blob/master/ransomware_overview.json

Given this JSON as a sample, how will you write a short script/program to process this json and store into databases

Explain selection of coding language and DB of your choice.

In the case of large set of data, how will you ensure that no records are missed when writing into the DB

In the case of duplicate entries, how will you handle them?

Design and Implement a REST API for front end consumption

Design a RESTful API for the system that allows users to store and retrieve data. Implement the endpoints in the language of your choice. Ensure that your API supports the basic CRUD operations on the data.

Design UI for the system for reporting dashboard

Solution:

Part1: Explain selection of coding language and DB of your choice.

Ans: Ideally, we would use Python + MongoDB as Python has good JSON processing libraries and integration with MongoDB. We use a non-relational database like MongoDB as the dataset is regular and has sub-types like Arrays embedded within the primary JSON object. Non-relational database offers better performance for large data sets.

Since, I do not have working experience in Python or MongoDB, I am going with the next best alternative JAVA springboot application with Elasticsearch as the preferred DB of choice due to my familiarity with it. Elastic search is also a NoSQL database that stores data as documents and is optimized for searching.

Part2: In the case of large set of data, how will you ensure that no records are missed when writing into the DB

Ans: For importing large datasets, we have the bulk indexing option provided by Elastic Search this will ensure that documents are written in an optimized manner without missing any and will gracefully handle errors.

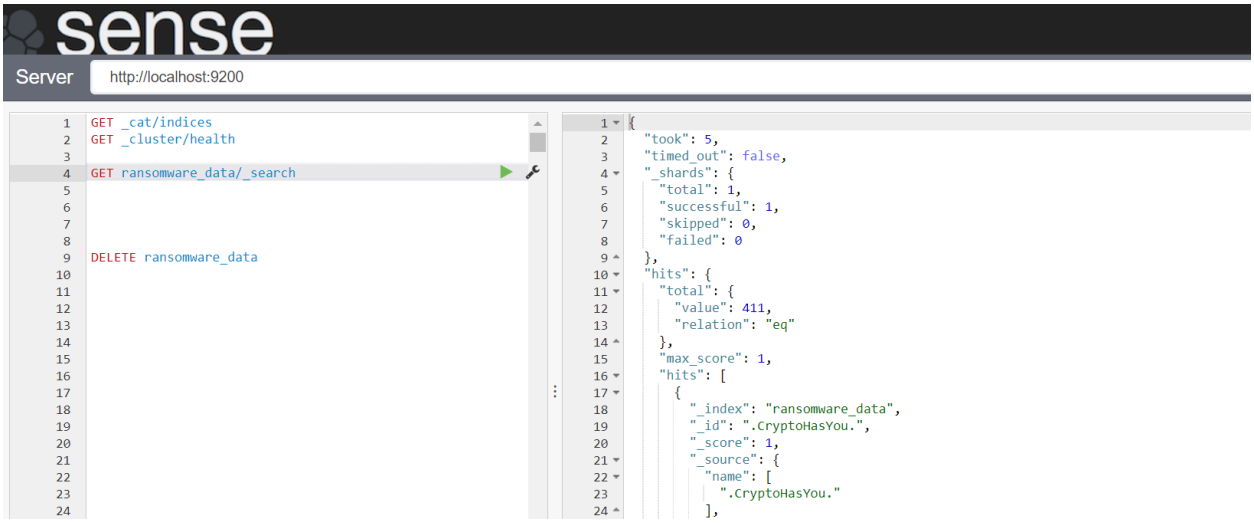
Part3: In the case of duplicate entries, how will you handle them?

Ans: Duplicate entries can be avoided in Elastic search by using a unique identifier as the id of the document. In this case, I am going to use the name property in the JSON object. If there are multiple names, the id will be a combination of the names in the array separated by a "~" symbol. In the below example, the id will be 777~Sevleg.


```
{
    "name": [
      "777",
      "Sevleg"
    ],
    "extensions": ".777",
    "extensionPattern": "._[timestamp]_$[email]$.777\ne.g. ._14-05-2016-11-59-36_$ninja.gaiver@aol.com$.777",
    "ransomNoteFilenames": "read_this_file.txt",
    "comment": "",
    "encryptionAlgorithm": "XOR",
    "decryptor": "",
    "resources": [
      "https://decrypter.emsisoft.com/777"
    ],
    "screenshots": "",
    "microsoftDetectionName": "Ransom:Win32/Empercrypt.A",
    "microsoftInfo": "https://www.microsoft.com/security/portal/threat/Encyclopedia/Entry.aspx?Name=Ransom:Win32/Empercrypt.A",
    "sandbox": "https://www.hybrid-analysis.com/sample/2955d081ed9bca764f5037728125a7487f29925956f3095c58035919d50290b5?environmentId=4",
```

```
    "iocs": "https://otx.alienvault.com/pulse/573b02701116a040ceccdd85/",
    "snort": ""
}
```


The Code solution for Part1-3 is found in the repository under ransomware-data folder. Please find the screenshot below for the data getting successfully inserted in the database:

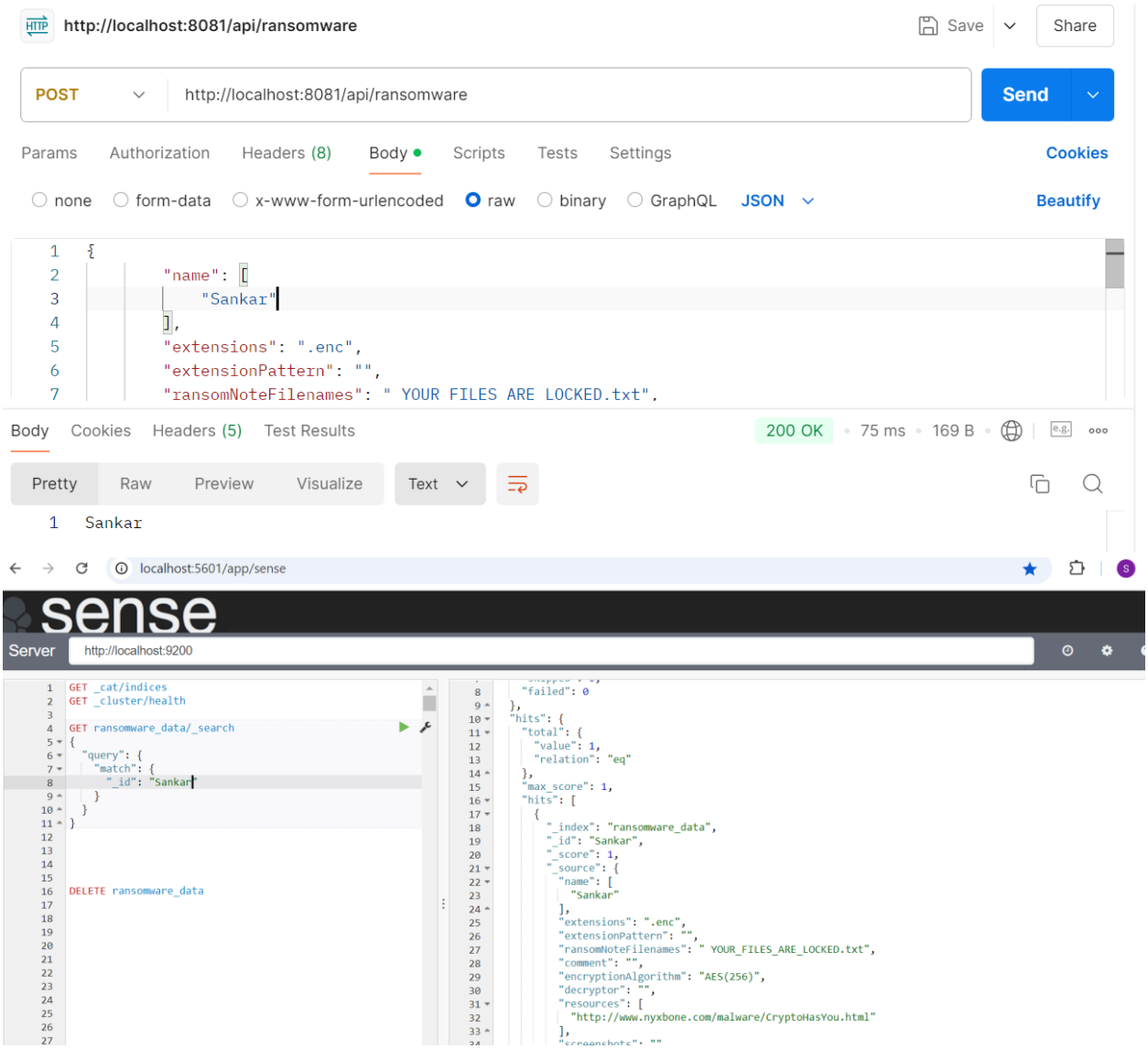


Part4: Design and Implement a REST API for front end consumption

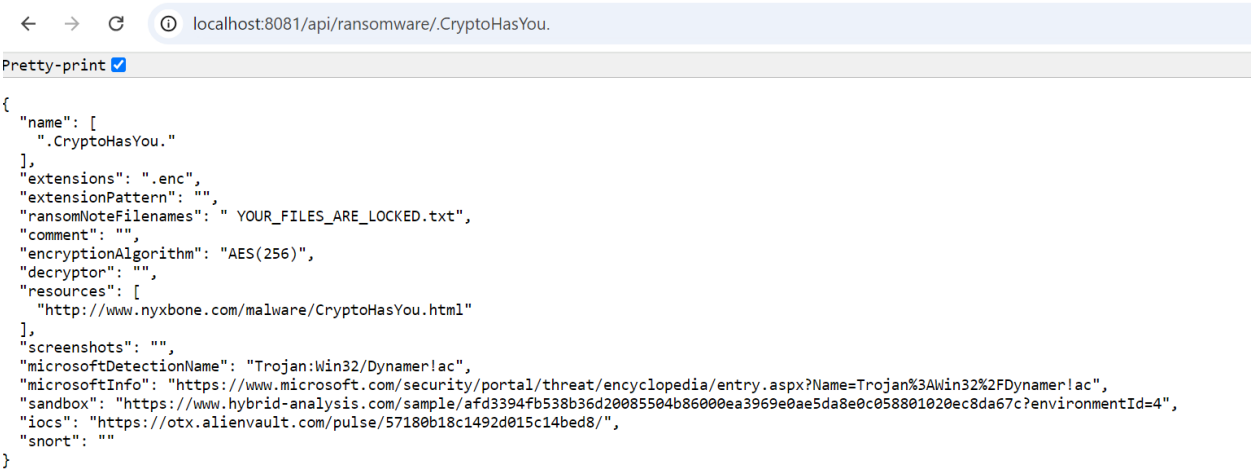
Design a RESTful API for the system that allows users to store and retrieve data. Implement the endpoints in the language of your choice. Ensure that your API supports the basic CRUD operations on the data.

Ans: I am using a spring boot application to design the REST API and it supports all the CRUD operations. The code for this can be found under **ransomware** folder in the repository.

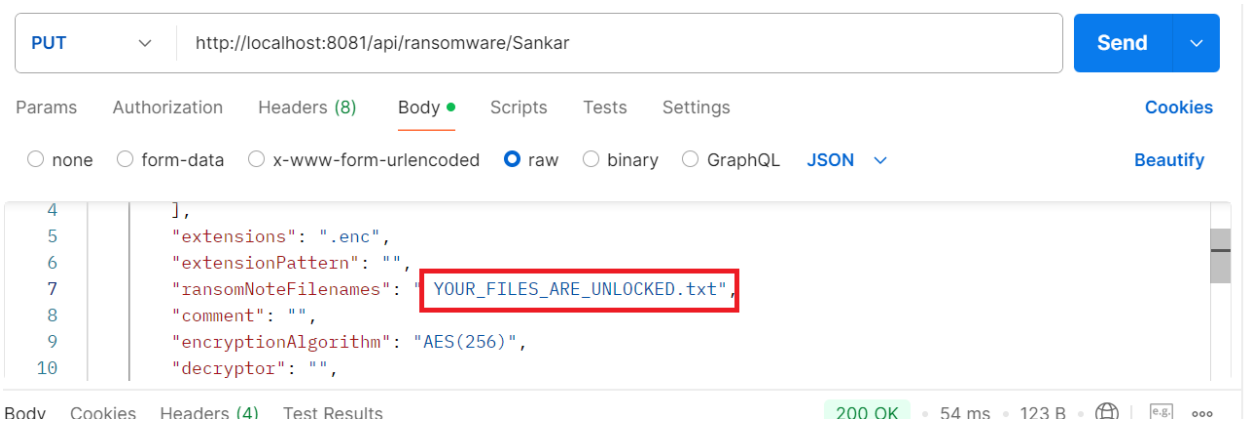
Create:

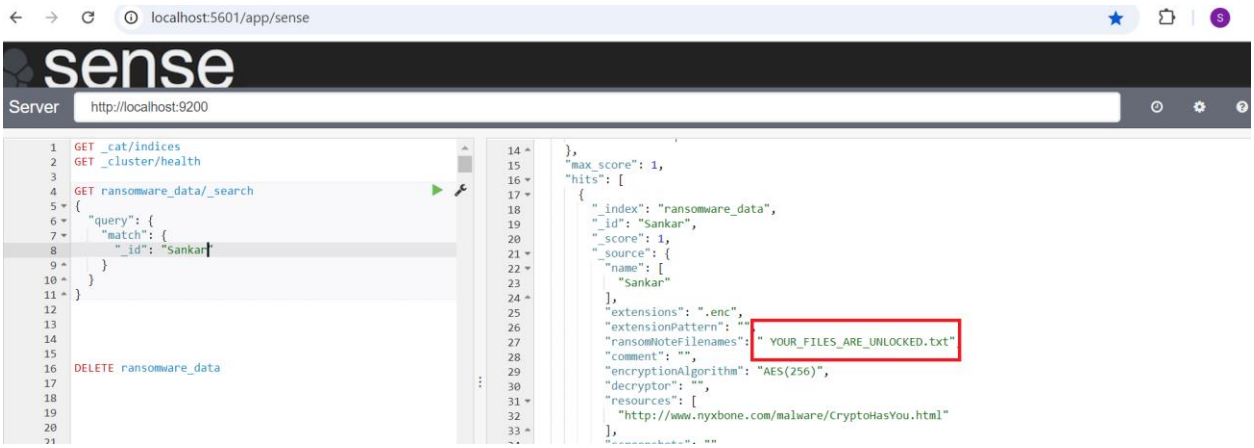

http://localhost:8081/api/ransomware
Save
Share
POST
http://localhost:8081/api/ransomware
Send
Params
Authorization
Headers (8)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name": [
"Sankar"
],
"extensions": ".enc",
"extensionPattern": "",
"ransomNoteFilenames": " YOUR FILES ARE LOCKED.txt",
Body
Cookies
Headers (5)
Test Results
200 OK
75 ms
169 B
Pretty
Raw
Preview
Visualize
Text
Sankar
localhost:5601/app/sense
sense
Server
http://localhost:9200
GET _cat/indices
GET _cluster/health
GET ransomware_data/_search
{
"query": {
"match": {
"_id": "Sankar"
}
}
}
DELETE ransomware_data
"failed": 0
},
"hits": {
"total": {
"value": 1,
"relation": "eq"
},
"max_score": 1,
"hits": [
{
"_index": "ransomware_data",
"_id": "Sankar",
"_score": 1,
"_source": {
"name": [
"Sankar"
],
"extensions": ".enc",
"extensionPattern": "",
"ransomNoteFilenames": " YOUR_FILES_ARE_LOCKED.txt",
"comment": "",
"encryptionAlgorithm": "AES(256)",
"decryptor": "",
"resources": [
"http://www.nyxbone.com/malware/CryptoHasYou.html"
],

Read:

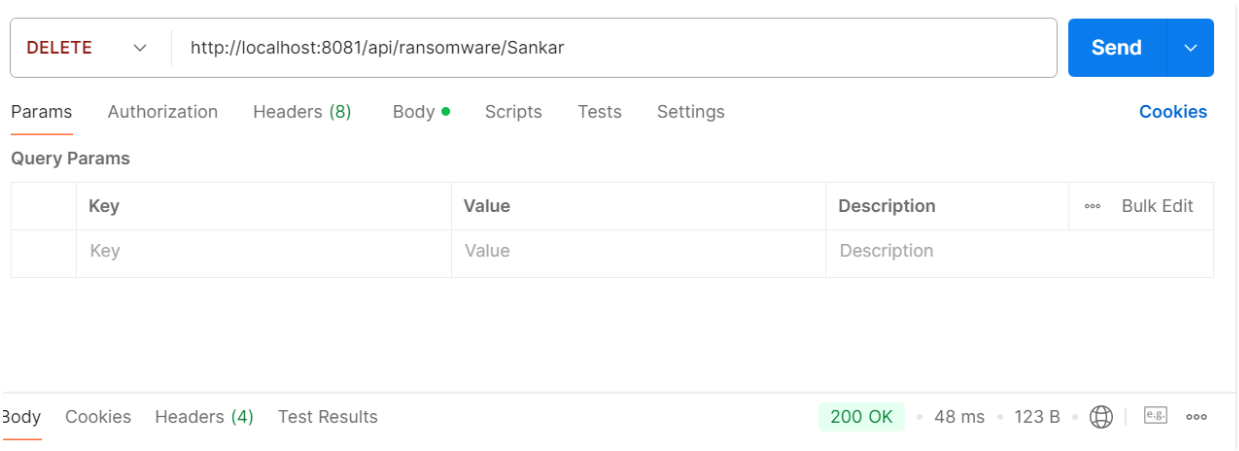


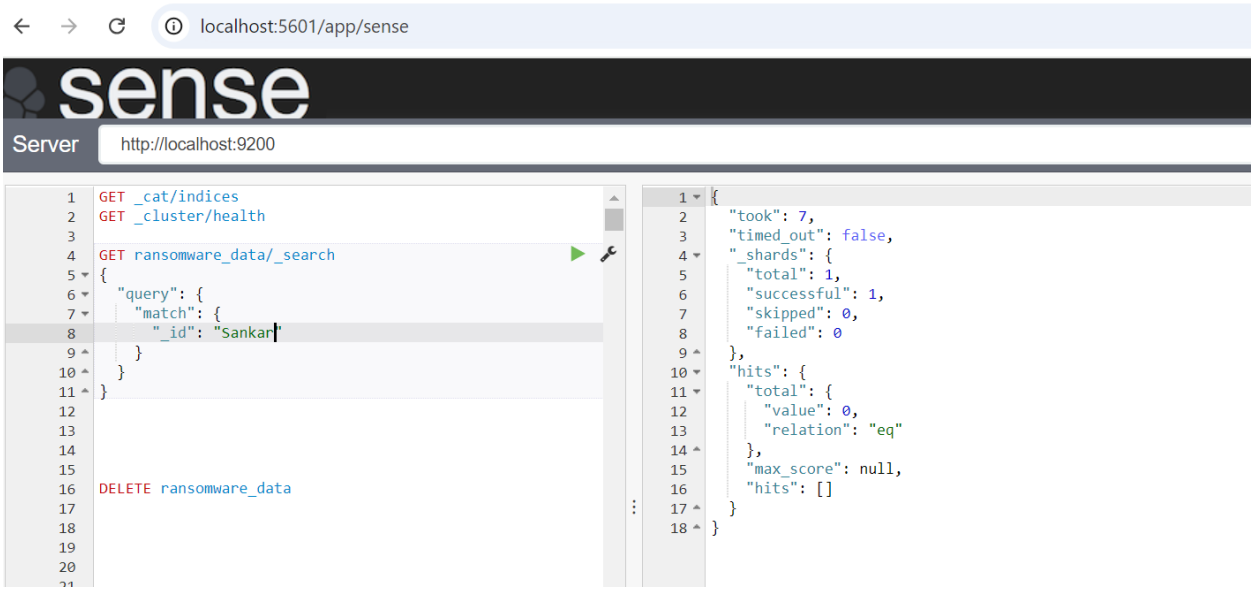
Update:



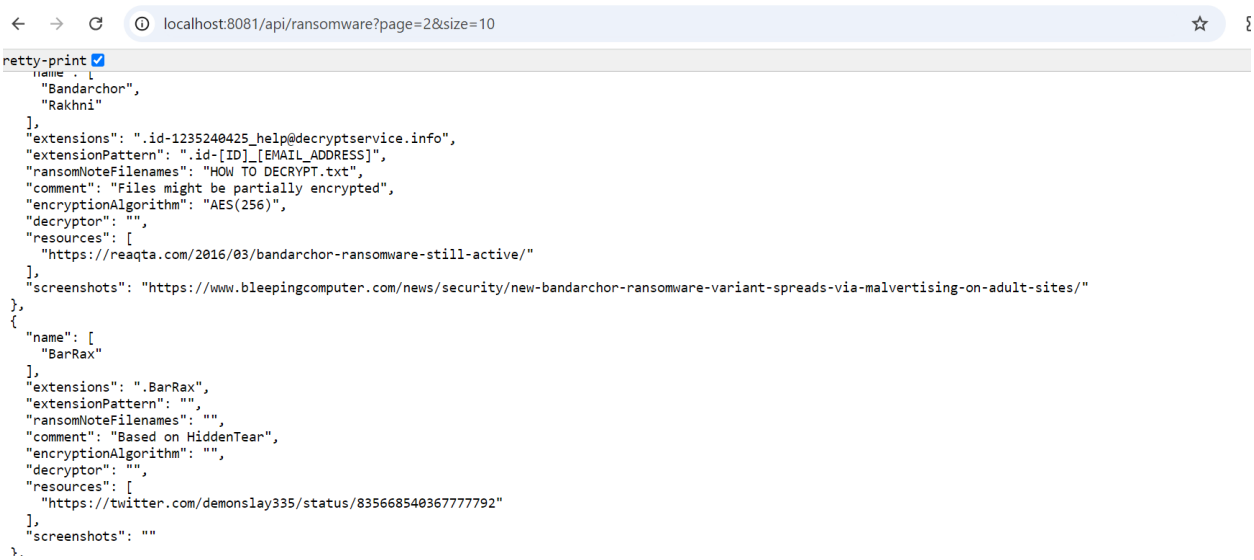


Delete:



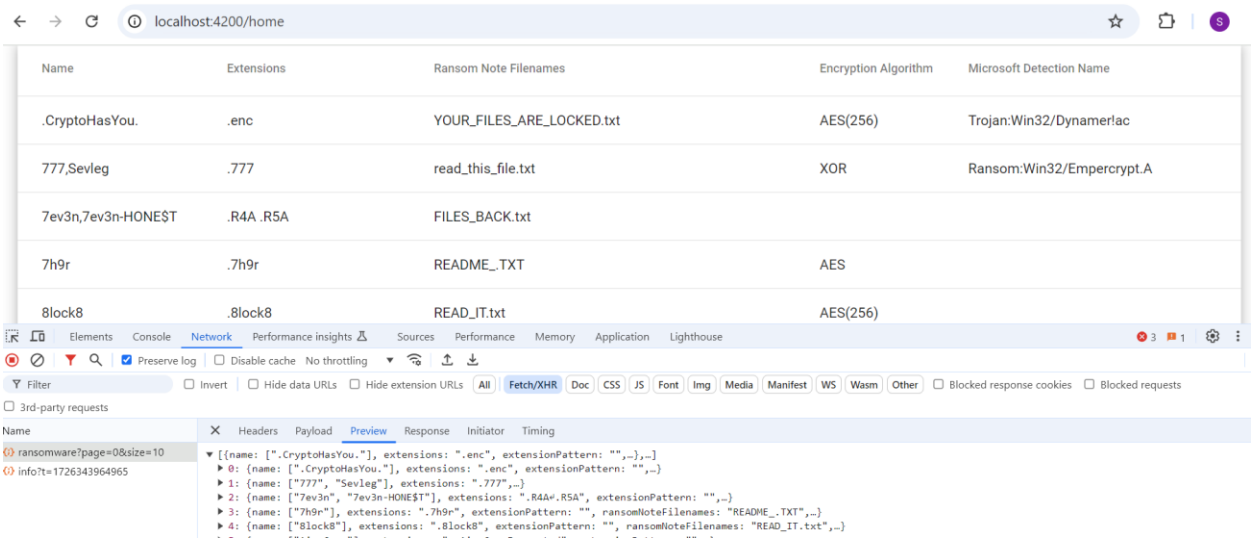


Also, implemented the API with pagination:



Part5: Design UI for the system for reporting dashboard

Ans: Basic UI is implemented with Angular 12 to display a few selected columns from the database. The code for this can be found in **ransomware-ui** folder in the repository. Screenshot below:



Notes/Considerations:

1. No security or authentications mechanisms like JWT has been implemented.
2. No data validations done from UI or Backend side.
3. Exhaustive exception handling or error code handling has not been implemented.