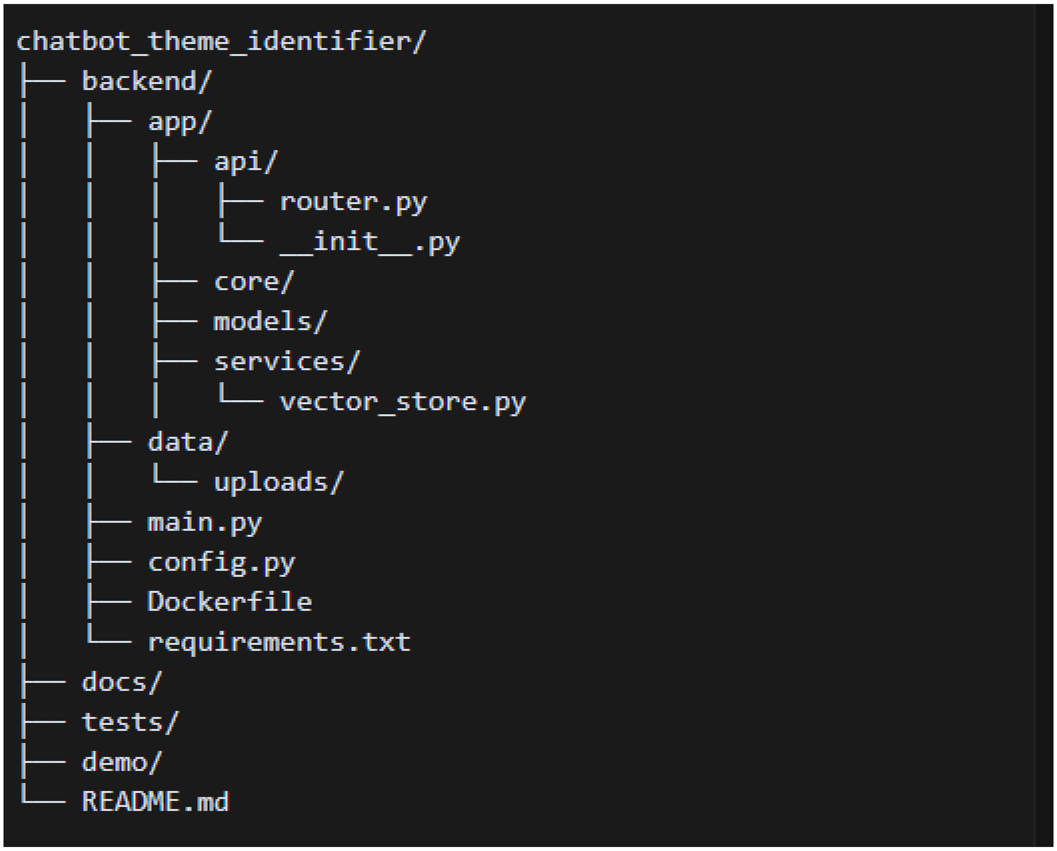
Project Report: Document Research & Theme Identification Chatbot

~ Heerak kashyap

1. Project Overview : The goal was to build a backend system that allows users to upload documents (PDFs, images), extract their text using OCR, store and index the content for semantic search, and provide an API for querying documents by meaning.

2. Folder Structure :



```
chatbot_theme_identifier/
├── backend/
│   ├── app/
│   │   ├── api/
│   │   │   ├── router.py
│   │   │   └── __init__.py
│   │   ├── core/
│   │   ├── models/
│   │   ├── services/
│   │   │   └── vector_store.py
│   ├── data/
│   │   └── uploads/
│   ├── main.py
│   ├── config.py
│   ├── Dockerfile
│   └── requirements.txt
├── docs/
├── tests/
├── demo/
└── README.md
```

3. Technology Stack :

- **Backend Framework:** FastAPI (Python)
- **OCR:** Tesseract (via pytesseract, Pillow)
- **PDF to Image:** pdf2image (requires Poppler)

- **Vector Database:** ChromaDB
- **Embeddings:** sentence-transformers (all-MiniLM-L6-v2)
- **Other:** Uvicorn (ASGI server)

4. Key Features Implemented :

A. Document Upload & OCR

- Users can upload PDFs or images via the /upload-document endpoint.
- Uploaded files are saved in backend/data/uploads/.
- OCR is performed:
- Images: Directly via Tesseract.
- PDFs: Each page is converted to an image (Poppler), then OCR is run.
- Extracted text is saved as a .txt file in the same folder.

B. Vector Store & Semantic Search

- Extracted text is embedded using a sentence transformer.
- Embeddings and metadata are stored in ChromaDB.
- /search endpoint allows users to query documents by semantic similarity, not just keywords.

C. API Endpoints

- GET /ping: Health check.
- POST /upload-document: Upload a document, extract text, store in vector DB.
- GET /search: Query documents by meaning.

5. Installation & Setup :

A. Python Dependencies

Install all dependencies:

```
pip install -r chatbot_theme_identifier/backend/requirements.txt
```

B. Tesseract OCR **-** Extract and add the bin folder to the PATH in env variables

C. Poppler (for PDF support) **-** Extract and add the bin folder to the PATH in env variables

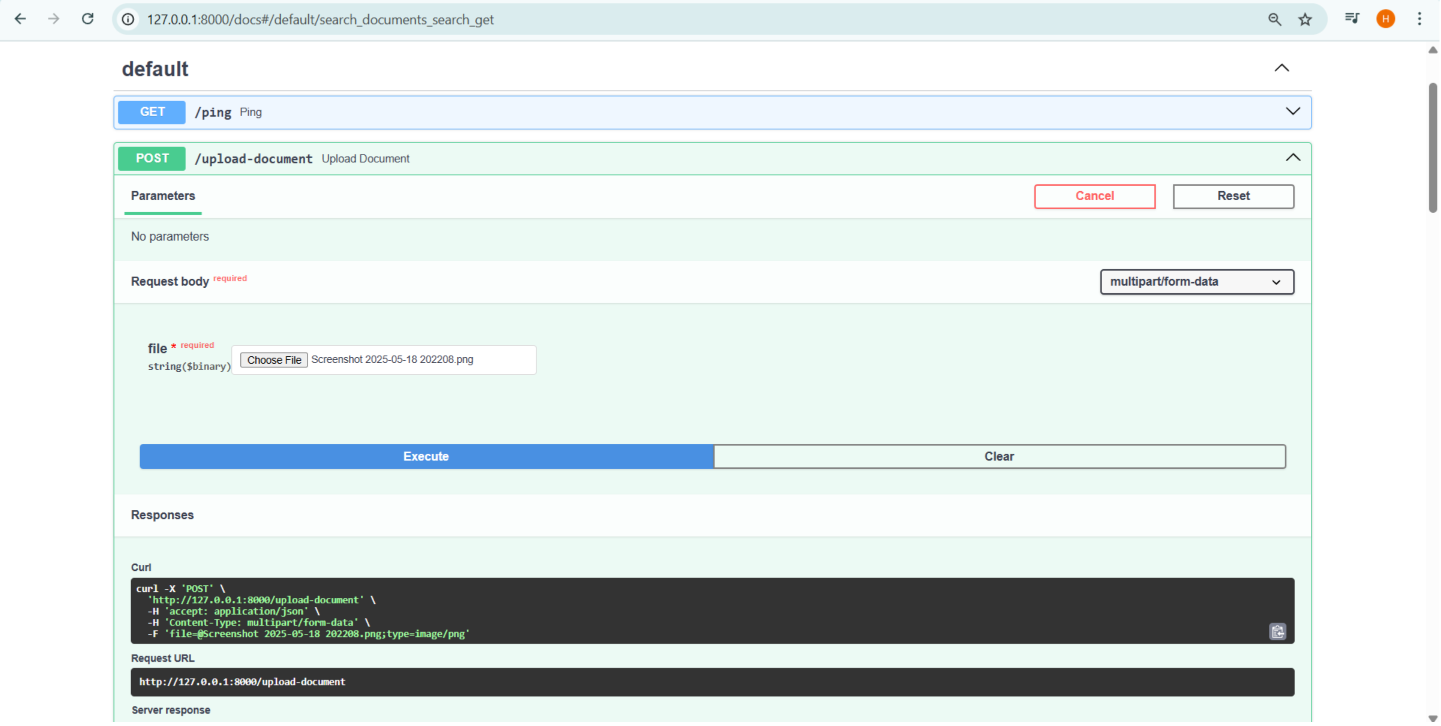
D. Run the Server

```
cd chatbot_theme_identifier/backend
```

uvicorn main:app –reload - Access the API docs at http://127.0.0.1:8000/docs.

6. Testing & Results :

- **Upload:** Used /upload-document to upload PDFs and images.
- **OCR Output:** Verified .txt files were created with extracted text.
- **Semantic Search:** Used /search endpoint to query for relevant documents by meaning.
- **Poppler & Tesseract:** Verified installations using pdfinfo -v and tesseract --version.



127.0.0.1:8000/docs#/default/search_documents_search_get

default

/ping Ping

/upload-document Upload Document

Parameters

No parameters

Request body required: multipart/form-data

file * required
string($binary)
Screenshot 2025-05-18 202208.png

Responses

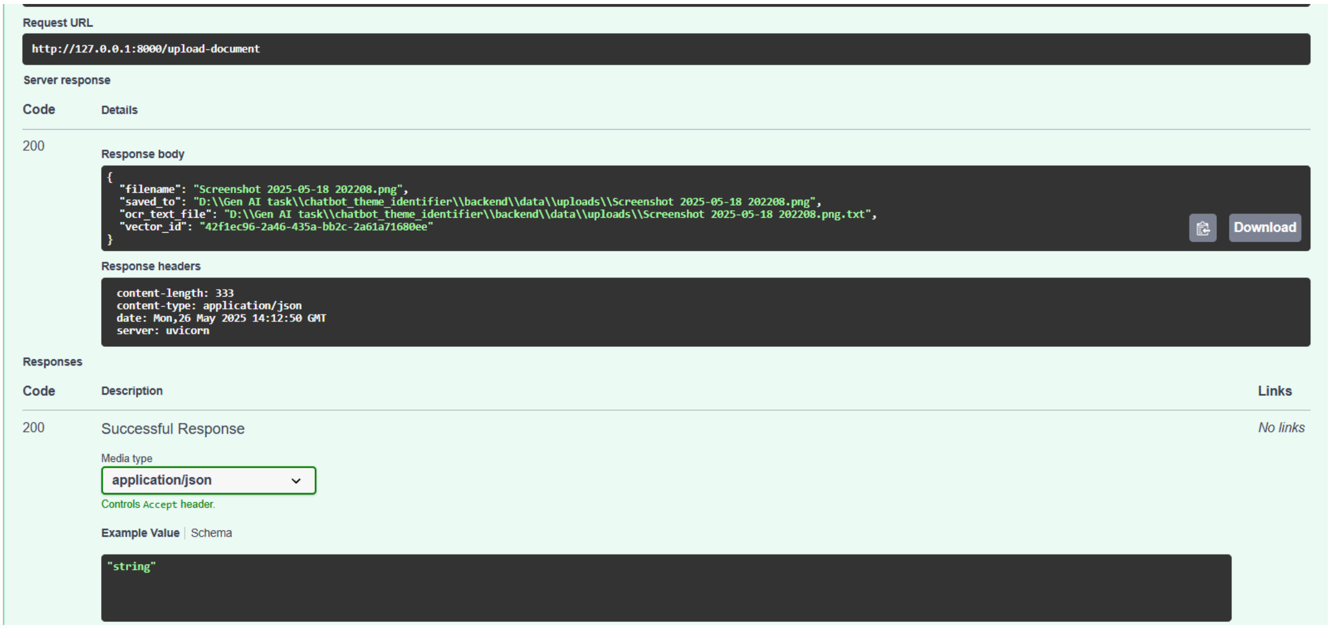
Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/upload-document' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@Screenshot 2025-05-18 202208.png;type=image/png'
```

Request URL

```
http://127.0.0.1:8000/upload-document
```

Server response



Request URL

```
http://127.0.0.1:8000/upload-document
```

Server response

Code | Details

200

Response body

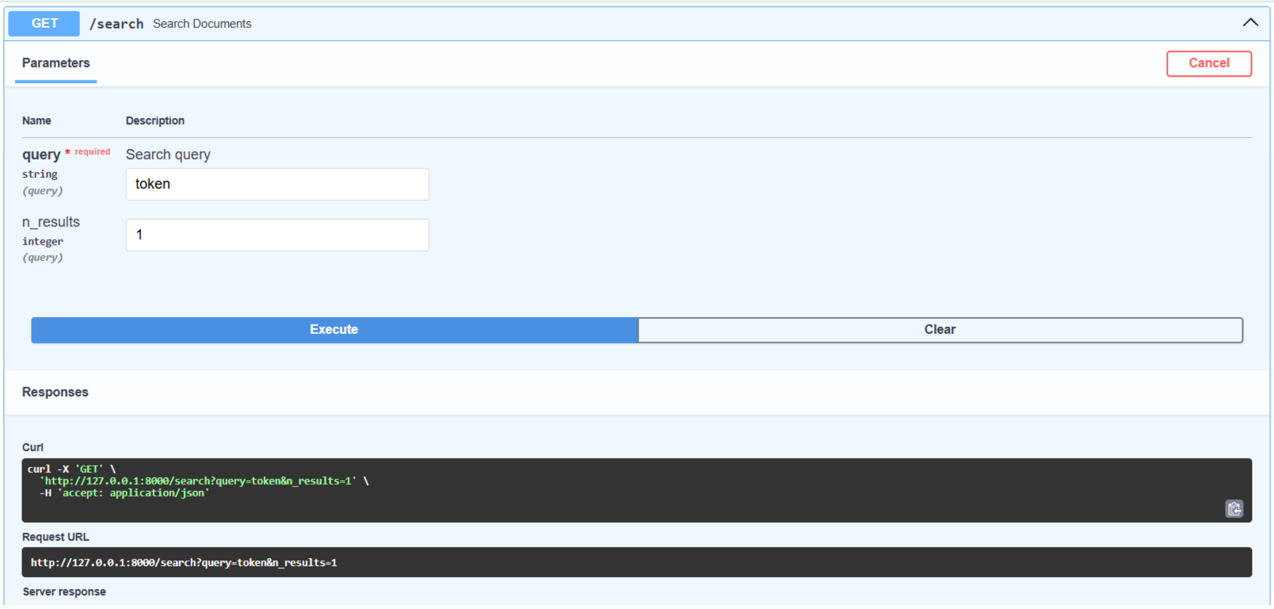
```
{
  "filename": "Screenshot 2025-05-18 202208.png",
  "saved_to": "D:\\Gen AI task\\chatbot_theme_identifier\\backend\\data\\uploads\\Screenshot 2025-05-18 202208.png",
  "ocr_text_file": "D:\\Gen AI task\\chatbot_theme_identifier\\backend\\data\\uploads\\Screenshot 2025-05-18 202208.png.txt",
  "vector_id": "42f1ec96-2a46-435a-bb2c-2a61a71680ee"
}
```

Response headers

```
 content-length: 333
 content-type: application/json
 date: Mon,26 May 2025 14:12:50 GMT
 server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response Media type: application/json Controls `Accept` header. Example Value | Schema "string"	*No links*



query * required
string
(query)
Search query
token

n_results
integer
(query)
1

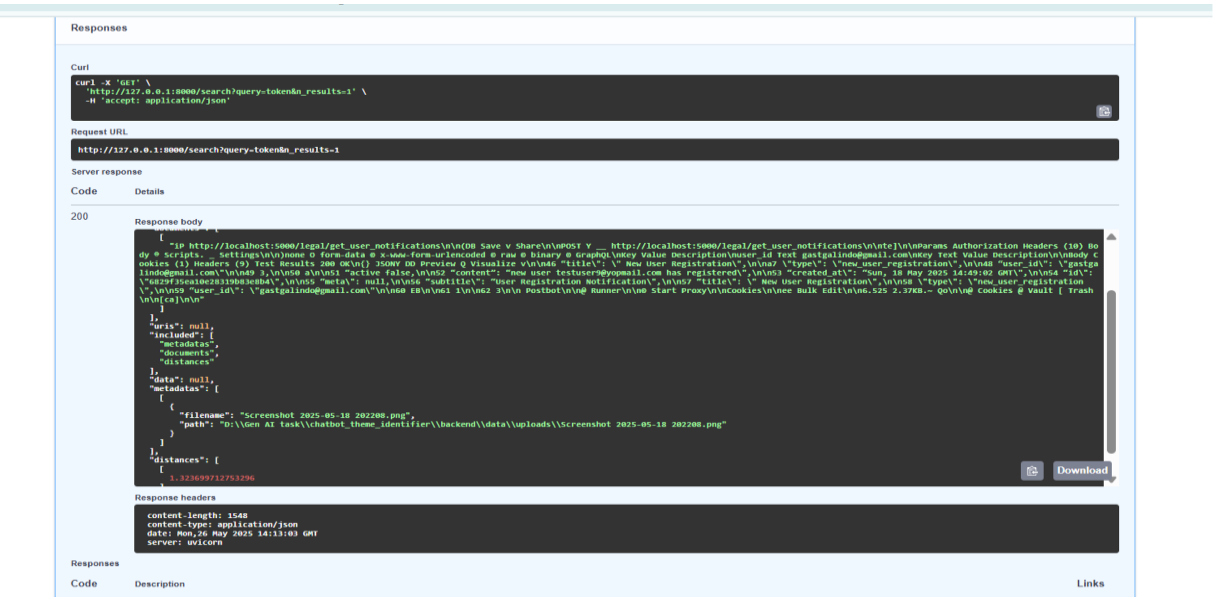
Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/search?query=token&n_results=1' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/search?query=token&n_results=1
```

Server response



Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/search?query=token&n_results=1' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/search?query=token&n_results=1
```

Server response

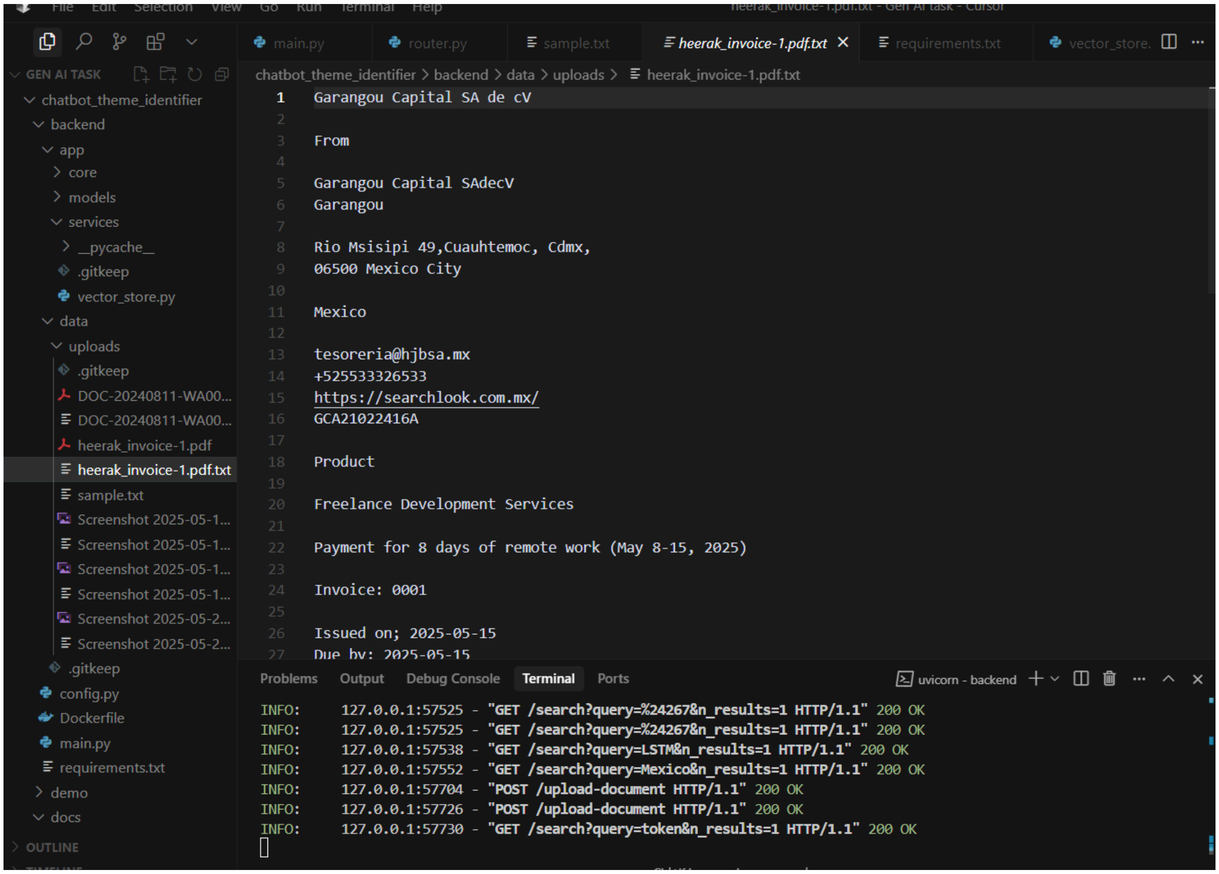
200

Response body

```
    [
      "iP http://localhost:5000/legal/get_user_notifications\n\n(DB Save v Share\n\nPOST Y __ http://localhost:5000/legal/get_user_notifications\n\nte]\n\nParams Authorization Headers (10) Bo
dy ® Scripts. _ Settings\n\n)none O form-data ® x-www-form-urlencoded ® raw ® binary ® GraphQL\nKey Value Description\nuser_id Text gastgalindo@gmail.com\nKey Text Value Description\n\nBody C
ookies (1) Headers (9) Test Results 200 OK\n{} JSONY DD Preview Q Visualize v\n\n46 “title\": \" New User Registration\",\n\na7 \"type\": \"new_user_registration\",\n\n48 “user_id\": \"gastga
lindo@gmail.com\"\n\n49 3,\n\n50 a\n\n51 “active false,\n\n52 “content”: “new user testuser9@yopmail.com has registered\",\n\n53 “created_at\": “Sun, 18 May 2025 14:49:02 GMT\",\n\n54 “id\":
\"6829f35ea10e28319b83e8b4\",\n\n55 “meta\": null,\n\n56 “subtitle\": “User Registration Notification\",\n\n57 “title\": \" New User Registration\",\n\n58 \"type\": \"new_user_registration
\",\n\n59 “user_id\": \"gastgalindo@gmail.com\"\n\n60 EB\n\n61 1\n\n62 3\n\n Postbot\n\n@ Runner\n\n® Start Proxy\n\nCookies\n\nee Bulk Edit\n\n6.525 2.37KB.~ Qo\n\n@ Cookies @ Vault [ Trash
\n\n[ca]\n\n"
    ]
  ],
  "uris": null,
  "included": [
    "metadatas",
    "documents",
    "distances"
  ],
  "data": null,
  "metadatas": [
    [
      {
        "filename": "Screenshot 2025-05-18 202208.png",
        "path": "D:\\Gen AI task\\chatbot_theme_identifier\\backend\\data\\uploads\\Screenshot 2025-05-18 202208.png"
      }
    ]
  ],
  "distances": [
    [
      1.323699712753296
```

Response headers

```
content-length: 1548
content-type: application/json
date: Mon,26 May 2025 14:13:03 GMT
server: uvicorn
```



```
Garangou Capital SA de cV

From

Garangou Capital SAdecV
Garangou

Rio Msisipi 49,Cuauhtemoc, Cdmx,
06500 Mexico City

Mexico

tesoreria@hjbsa.mx
+525533326533
https://searchlook.com.mx/
GCA21022416A

Product

Freelance Development Services

Payment for 8 days of remote work (May 8-15, 2025)

Invoice: 0001

Issued on; 2025-05-15
Due by: 2025-05-15
```

```
INFO:     127.0.0.1:57525 - "GET /search?query=%24267&n_results=1 HTTP/1.1" 200 OK
INFO:     127.0.0.1:57525 - "GET /search?query=%24267&n_results=1 HTTP/1.1" 200 OK
INFO:     127.0.0.1:57538 - "GET /search?query=LSTM&n_results=1 HTTP/1.1" 200 OK
INFO:     127.0.0.1:57552 - "GET /search?query=Mexico&n_results=1 HTTP/1.1" 200 OK
INFO:     127.0.0.1:57704 - "POST /upload-document HTTP/1.1" 200 OK
INFO:     127.0.0.1:57726 - "POST /upload-document HTTP/1.1" 200 OK
INFO:     127.0.0.1:57730 - "GET /search?query=token&n_results=1 HTTP/1.1" 200 OK
```

- FastAPI docs UI with endpoints.
- .txt file with extracted text.

10. Conclusion :

The backend for the Document Research &Theme Identification Chatbot is fully functional, supporting document upload, OCR, semantic indexing, and search. The system is modular, extensible, and ready for further development or deployment.