Doc Search AI

Using AI to Revolutionizes Documents Retrieval for Highly Relevant and Accurate Results

We are the Builders!!!

Kean Seng Tan

Data Scientist

Wai Yan

Full Stack Software Engineer

Khant Sithu Zaw

Full Stack Software Engineer

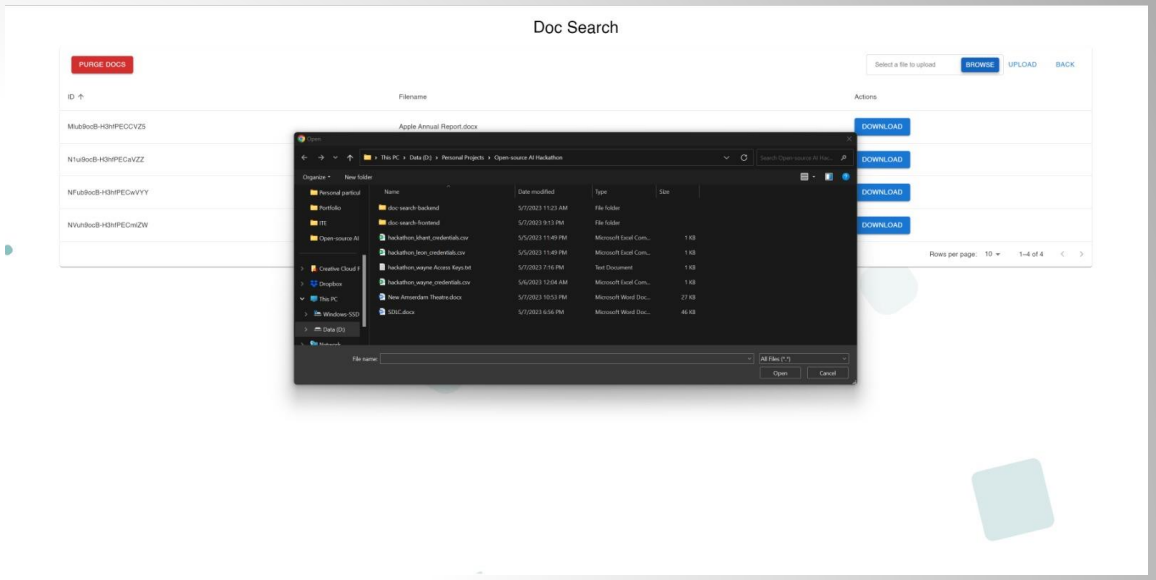
Problem: Difficulty in Finding Relevant Documents

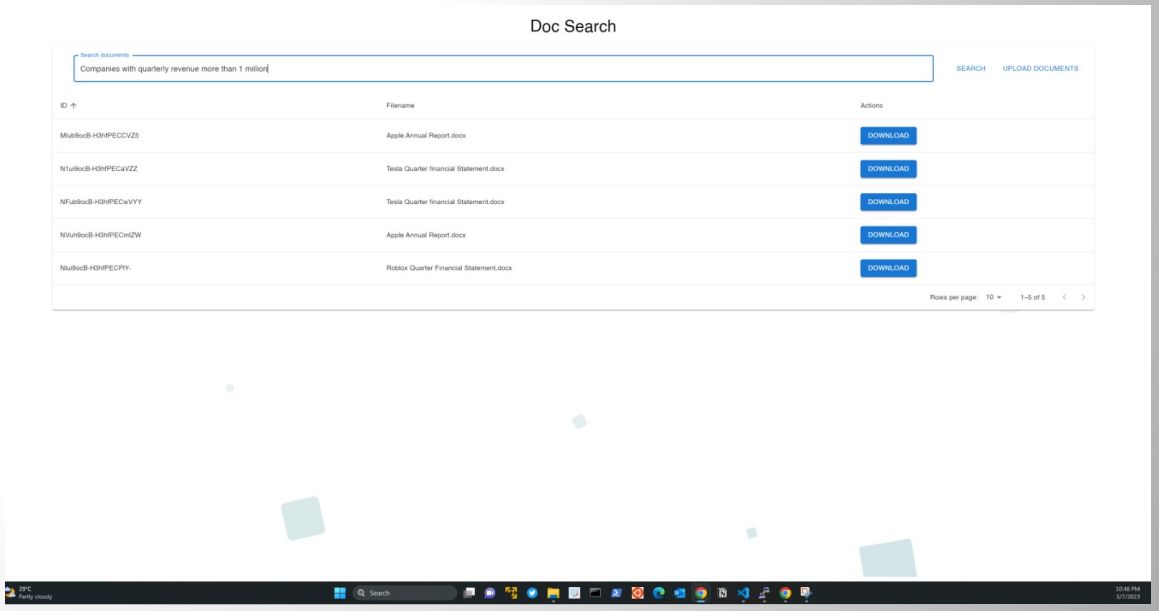
- Users often struggle to find the most relevant and useful documents related to their needs due to the following reasons, which can be time-consuming and frustrating.
    i. Inefficient keyword-based search: Traditional search methods often rely on users inputting specific keywords, which can lead to irrelevant results or missed documents that may still be valuable.
    ii. Lack of query context: Users may not always provide sufficient context in their queries, leading to less accurate search results.

Solution: Doc Search AI

An AI Document Searching tool that addresses the shortcomings of traditional search methods and offers users a more effective and efficient way to find relevant documents by leveraging advanced AI techniques.

1. Upload documents to Doc Search AI data storage.
2. Insert the type of information that you are interested to know in natural way of asking question. For example, user can try to search for “Companies with annual revenue more than $1 millions”.
3. Click to -> Return the documents with the companies that has annual revenue of more than $1 millions.





How we made it

- Frontend: ReactJS.
- Backend:
  - DocSearch AI's backend is a REST API application that allows users to upload documents and search for documents that match a specified query. It is built using Python and FastAPI
  - Elasticsearch is used as the document search engine.
- Open Source AI model:
  - GPT-2 from Hugging Face (https://huggingface.co/gpt2).
  - This AI model is used to conduct semantic expansion on users' document retrieval query so that we can get a search query that better represent the search intents of the users.
- Codes:
  - Frontend: https://github.com/WayneMyo/doc-search-frontend
  - Backend: https://github.com/WayneMyo/doc-search-backend
  - Ai Model Search Codes: https://github.com/WayneMyo/doc-search-llm
- DocSearch AI App URL: http://doc-search-frontend.s3-website-ap-southeast-1.amazonaws.com/

Next Steps: Improvements and Monetization

Improvements:

- **Upgrade to a more advanced language model and hosted the language model with GPU servers**: For the next step, we plan to power the Doc Search AI with larger Open Source language model like Alpaca or Vicuna 13B and hosted the AI model with GPU server so as to improve the quality and speed of getting the results.
- **Personalization**: Integrate user behavior analysis and preferences to provide personalized search results, tailoring the experience to individual users' needs and interests.
- **Integration with databases and support for tabular data search**: Allow users to connect "Doc Search AI" to various databases so that the tool is able to allow users to search for relevant tabular data in databases in natural way of asking question.

Monetization:

- **Subscription-based model**: Offer different subscription tiers with varying levels of access, features, and support. This can include a free basic tier with which only support for document search, as well as premium tiers for more type of data storage integration.
- **White-label solution**: Allow companies to license and rebrand "Doc Search AI" as their own product, offering a fully customizable solution for their customers.