

A REPORT
ON
(RAG (Retrieval-Augmented Generation) pipeline which assist users with
their shopping needs)



BY
SHRISH KUMAR **2022A7PS1295H**
AT
PARALLELDOTS AI, GURUGRAM
A Practice School-I Station of
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
(May-July, 2024)

A REPORT ON
RAG (Retrieval-Augmented Generation) pipeline which assist users with
their shopping needs

BY

SHRISH KUMAR

2022A7PS1295H

CSE

Prepared in partial fulfilment of the
Practice School-I Course Nos.
BITS C221/BITS C231/BITS C241AT

PARALLELDOTS AI, GURUGRAM

A Practice School-I Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(May-July, 2024)

ACKNOWLEDGEMENTS

First, I would like to thank our college Birla Institute of Technology and Science, Hyderabad, for providing us with this opportunity to gain work experience in the form of Practice School-1. We are also grateful to Paralleldots AI, Gurugram for accepting us into their esteemed company to work on the AI chatbot project. We have been doing our best to learn and apply our knowledge of python libraries to develop a number of projects and learn to make real-time applications. I would also like to thank our mentor, Ms. Megha Naithani for her guidance and patience towards us. We would like to thank Mr. Mukhtabh Srivastava for providing us the opportunity to learn and apply our learnings to this important project. I would like to express my gratitude towards our faculty mentor Prof. Ankur Pachauri, who has been our guide throughout working in the industry. He has explained to us the importance of this PS-1 project and the deliverables we must extract to get the most out of this program. He has prepared us to face the industry and the hardships that we would face while trying to do our work.

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
(RAJASTHAN)
Practice School Division**

Station: Paralleldots AI

Centre: Gurugram

Duration: 28th May - 23rd July, 2024

Date of Start: 28th May

Date of Submission: 22nd June, 2024

Title of the Project: Building a RAG (Retrieval-Augmented Generation) pipeline which assist users with their shopping needs

ID No./Name(s)/ 2022A7PS1295H /SHRISH KUMAR / CSE

**Discipline(s)/of
the student(s)**

**Name(s) and
designation(s) of the
expert(s):** Megha Naithani, Data Scientist

**Name(s) of the
PS
Faculty:** Prof Ankur Pachauri

Key Words: LLMs , NLP , Prompt Engineering ,Python

Project Areas: AIML, Information Retrieval, Prompt Engineering, LLMs

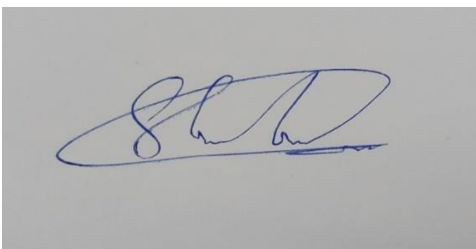
ABSTRACT

Creating a shopping assistant to find suitable products, given a dataset and a query by the user. This system uses a data file in any format as the context and provides accurate and relevant information based on user queries.

Key components of the project include the Ollama local language model (LLM) for generating natural text responses, Llamaindex as the framework, BGE for embedding the data, and ChromaDB for storing the vector database. Python libraries like Pandas and Simpliedirectoryreader(a llama index module) were used to convert a given file into a suitable format.

The process begins with scraping all the useless data, focusing on product titles, descriptions, and details. This data is then preprocessed and stored in a suitable format (compatible with our framework) using pandas and Simpliedirectoryreader, which are then loaded and split into chunks using Llamaindex. Further, this data is indexed using the VectorStoreIndex module of Llamaindex and embedded using BGE embeddings. Then, the data is stored in ChromaDB. The RAG pipeline retrieves the most relevant data/document based on user queries. For this, I used the query engine Llamaindex, which does both the task of choosing the relevant document and the relevant data from the document. I currently use Gemma and Llama3 as the LLMs to answer these queries.

This project demonstrates the creation of a robust RAG pipeline and highlights the integration of modern NLP and machine-learning tools to address real-world problems. The result is an intelligent assistant capable of effectively comprehending and responding to detailed queries.



Signature(s) of Student(s)
21 July

Signature of PS Faculty
Date:

TABLE OF CONTENTS

1.Cover Page	1
2.Title Page	2
3.Acknowledgements	3
4.Abstract sheet	5
5.Table of Contents	6
6.Introduction	7
7.Project Workflow	8
8.Conclusion	16
9.Success Story	17
10. Glossary	18

INTRODUCTION

The advent of artificial intelligence has revolutionized various aspects of daily life, including shopping. Traditional e-commerce platforms have evolved from static catalogs to dynamic, personalized shopping experiences. A significant development in this transformation is the implementation of Retrieval-Augmented Generation (RAG) pipelines. RAG leverages the power of retrieval systems and generative models to provide accurate, context-aware, and user-friendly responses to queries. This report outlines the development and implementation of a RAG pipeline designed to assist users with their shopping needs, providing real-time recommendations, detailed product information, and personalized shopping advice.

ParallelDots AI, a pioneering company in the field of artificial intelligence, is making significant strides in the AI domain by developing cutting-edge solutions for various industries. In this project, ParallelDots AI is focused on revolutionizing the e-commerce sector by building an advanced chatbot designed to assist users with their shopping needs. The chatbot leverages a Retrieval-Augmented Generation (RAG) pipeline to provide real-time product recommendations, detailed information, and personalized shopping advice.

By integrating sophisticated AI technologies, ParallelDots AI aims to enhance user experience, streamline the shopping process, and set new standards in e-commerce.

PROJECT PROGRESS

CONTEXT:

Machine learning and artificial intelligence have come a long way very quickly. This has led to the creation of very complex large language models (LLMs). These models are trained and fine-tuned on a huge number of tasks. They can better process and understand data and come up with appropriate responses and do well at a wide range of tasks because they have so many parameters. However, it takes a lot of time and effort to train and fine-tune these huge models. The development of LLMs, such as Gemma, Llama3, and Mistral, has revolutionized natural language processing (NLP), enabling machines to understand and generate human language with unprecedented accuracy. LLMs have come a long way thanks to the huge amounts of data that are available and the exponential growth of computing power. To train these models, huge datasets with a wide range of languages, contexts, and domains are fed to them. The models get a lot of training, which helps them learn complex patterns and subtleties in human language.

The main aim of this project is to create a sophisticated AI shopping assistant that could be used to answer user queries based on e commerce products. The assistant wants to automate and improve the shopping experience by using the knowledge of AI and natural language processing (NLP). The main goal of this project was to make a strong system that records important information like product names, product descriptions and details. The assistant uses advanced NLP techniques to make sure that the output is accurate with as little hallucination as possible.

This introduction gives a brief outline of the project's goals, methods, main findings, and suggestions based on the scheduling assistant's analysis. This report outlines the development process, insights gained, and problems faced while building the project along with mentioning the next steps that need to be taken in the second half of the project to complete it.

Collaborative teamwork further fueled our efforts, fostering an environment where diverse skills in prompt engineering, Generative AI , Large Language Models (LLMs) converged seamlessly. Working together as a team was very important to the project's success because it let us use our combined knowledge and experience to make the AI assistant better all the time. This synergy made it easier to keep coming up with new ideas and solving problems, so we could deal with new problems quickly and improve the assistant's features one step at a time.

PROMPT ENGINEERING:

Prompt engineering is an important part of making AI-powered apps because it makes sure that language models give correct and useful answers to questions. In our project to make a shopping assistant, getting users to give us concise and precise information and preferences depended on how well the prompts were designed.

There are several steps involved in crafting good prompts:

1. Clearly Stated Goals: To give the language model direction and purpose, make it clear what you want it to do.
2. Make it clear and specific: Make sure your question or request is clear and to the point. Don't leave any room for confusion.
3. Give Context: Give all the background information and specific instructions that the model needs to respond correctly.
4. Iterate and Test: Run your prompts through the model, look at the results, and change the prompts based on how well they work and preventing hallucination by the model.
5. Review and Improve: Always go back and improve your prompts to make communication clearer, faster, and more accurate.

INITIAL EXPERIMENTATIONS

Since organizations cannot use OpenAI for client's data, our team practiced writing prompts that would get clear and useful answers. This first phase was very helpful for learning because it helped us understand how to structure queries in a way that made the most of the model's abilities.

Once the prompt template was finalized, next step was to figure out how to deploy the model on a dataset through a text file then an excel file.

SCRIPTING:

the task was to write a script using LlamaIndex framework, ollama embeddings to convert the data file into chunks of data of fixed size. As Gemma as the base model, a successful script was designed.

To implement the same on a larger dataset, a vector database was needed.

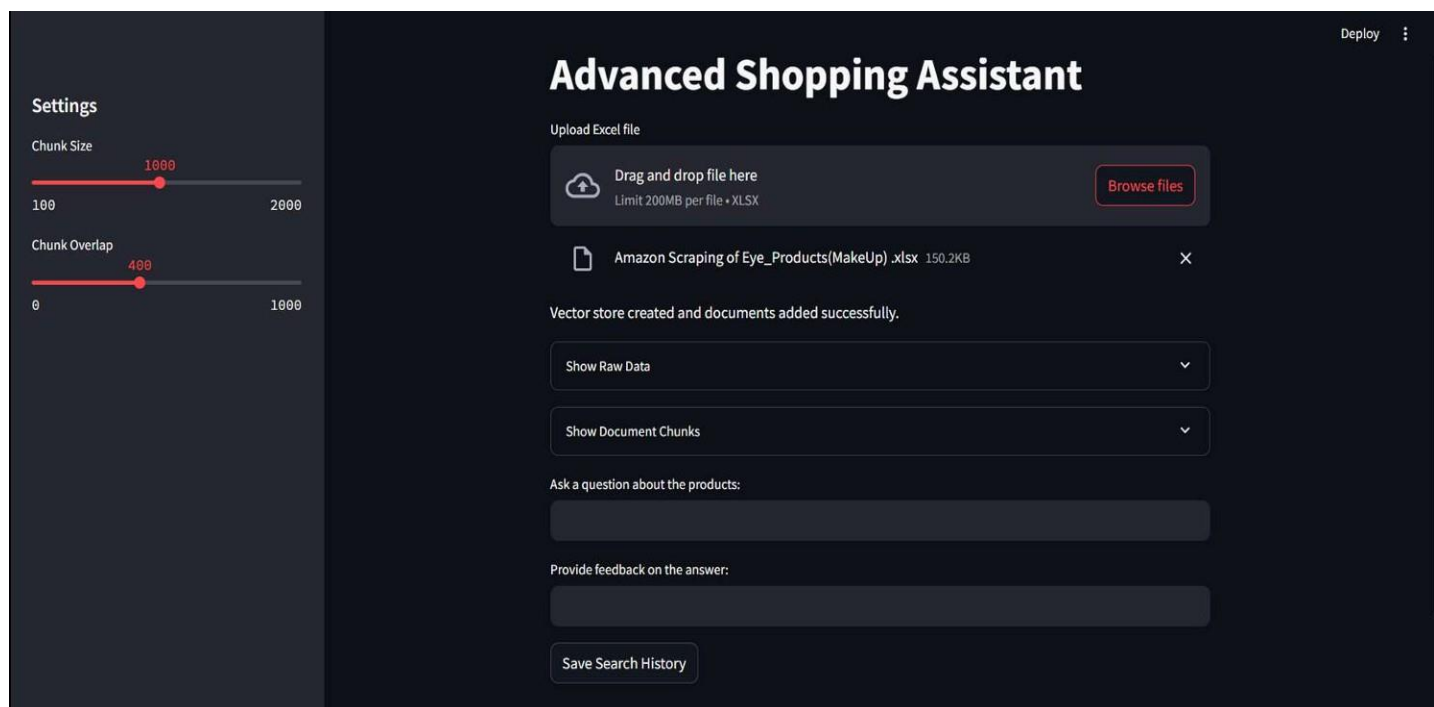
FINDING A SUITABLE VECTOR DATABASE:

Initially Fiass vector database was experimented on, but due to inaccuracies in the outputs, Chromadb was finalized which always give satisfactory output.

DESIGNING A FRONT END FOR THE CHATBOT:

For front end of the python script, Streamlit Application was used to provide a platform. To integrate streamlit with the script, changes had to be made in the code. Many features were added to the front end site, like the display of raw data and data chunks procured to make the debugging process easier, a dynamic slider to set the chunk size and chunk overlap parameters straight from the site.

Site Snippet:



ADDING CATEGORIES AND SUBCATEGORIES:

To test a larger dataset in the future, which will consist of all products of different types the prompt template has to be implemented with all the possible categories and their sub categories:

```
# Categorizing the query into category and sub-category
def categorizeProduct(question):
    Settings.llm = Ollama(model="gemma", request_timeout=1000, temperature=0.5)

    categories_dict = {
        "Makeup": ["Body", "Eyes", "Face", "Lips", "Makeup Palettes", "Makeup Remover", "Makeup Sets"],
        "Skin Care": ["Body", "Eyes", "Face", "Lip Care", "Maternity", "Sets & Kits", "Sunscreens & Tanning Products"],
        "Hair Care": ["Detanglers", "Hair Accessories", "Hair Coloring Products", "Hair Cutting Tools", "Hair Extensions",
                     "Wigs & Accessories", "Hair Fragrances", "Hair Loss Products", "Hair Masks", "Hair Perms, Relaxers & Texturizers",
                     "Hair Treatment Oils", "Scalp Treatments", "Shampoo & Conditioner", "Styling Products"],
        "Fragrance": ["Children's", "Dusting Powders", "Men's", "Sets", "Women's"],
        "Tools & Accessories": ["Bags & Cases", "Bathing Accessories", "Cotton Balls & Swabs", "Makeup Brushes & Tools",
                               "Mirrors", "Refillable Containers", "Shave & Hair Removal"],
        "Shave & hair removal": ["Men's", "Women's"],
        "Personal Care": ["Bath & Bathing Accessories", "Deodorants & Antiperspirants", "Lip Care", "Oral Care",
                          "Piercing & Tattoo Supplies", "Scrubs & Body Treatments", "Shave & Hair Removal"],
        "Salon & Spa Equipment": ["Hair Drying Hoods", "Galvanic Facial Machine", "Handheld Mirrors", "High-Frequency Facial Machines",
                                 "Manicure Tables", "Professional Massage Equipment", "Salon & Spa Stools", "Spa Beds & Tables", "Salon & Spa Chairs",
                                 "Spa Storage Systems", "Spa Hot Towel Warmers"],
        "Foot, Hand & Nail Care": []
    }
}
```

Select a category

Makeup

Makeup

Skin Care

Hair Care

Fragrance

Tools & Accessories

Shave & hair removal

Personal Care

Salon & Spa Equipment

Select a subcategory

Body

Body

Eyes

Face

Lips

Makeup Palettes

Makeup Remover

Makeup Sets

PROVIDING AMAZON LINKS AND DESCRIPTION:

```
# Output
if submit:
    category_subcategory = categorizeProduct(question)
    st.write(category_subcategory)

    names_dict = productSearch(vectorstore, question)

    if names_dict:
        for key, value in names_dict.items():
            parts = value.split("\n")
            link, description = parts[0], parts[1] if len(parts) > 1 else ""

            st.success(key)
            st.write("Buy: ", link)
            st.write(":green[Description]: ", description)
    else:
        st.error("No results found")
```

Enter your query to search product !

Enter your query

Find a waterproof mascara

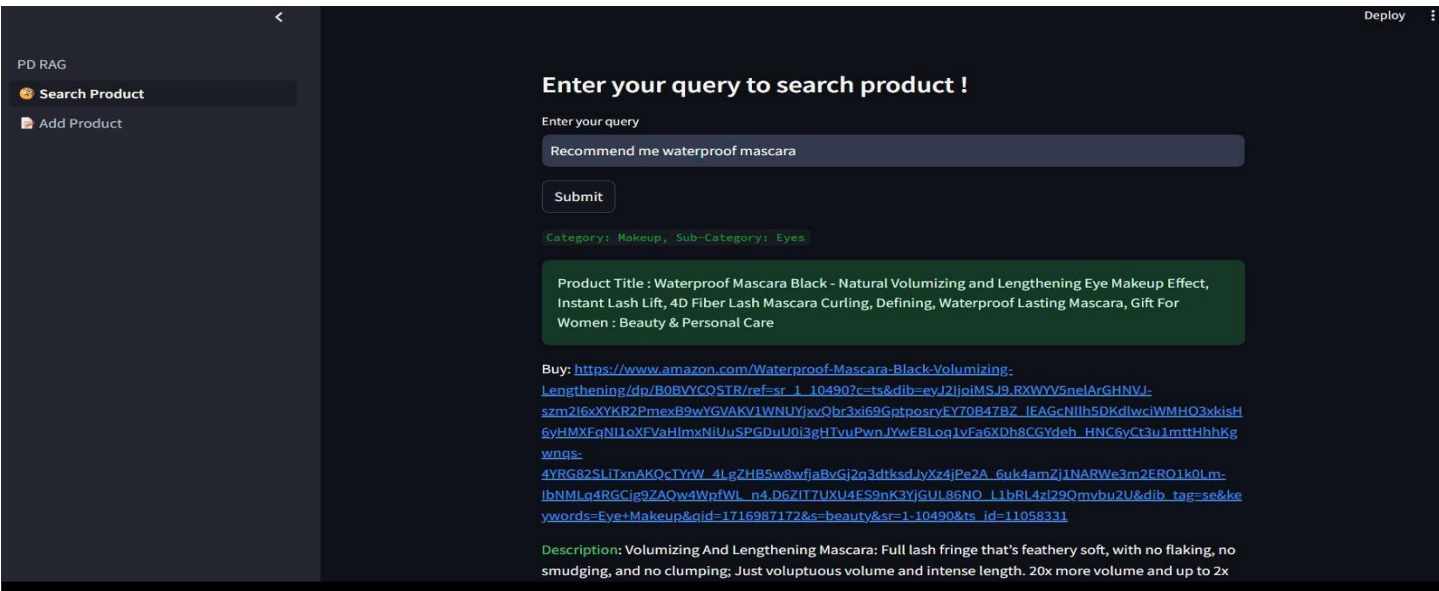
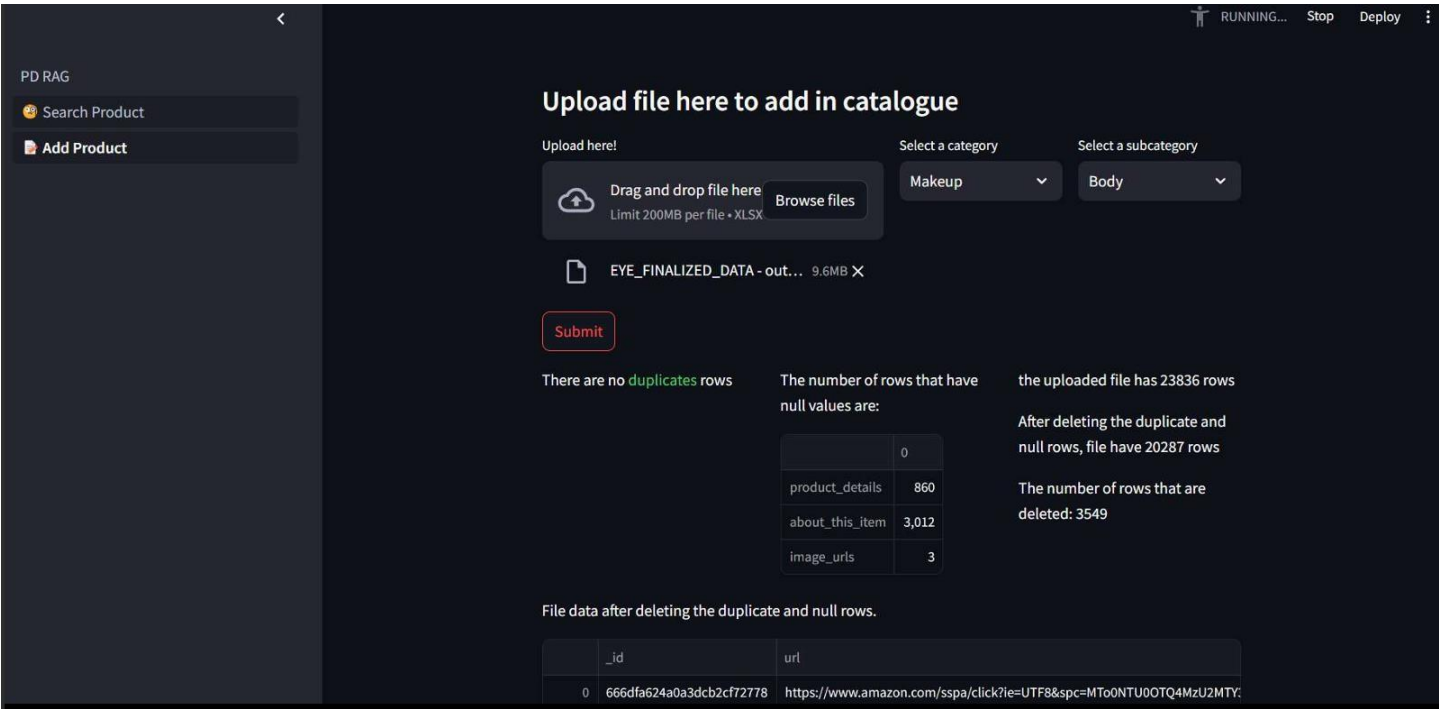
Submit

Category: Makeup, Sub-Category: Eyes

Product Title : essence | Lash Princess False Lash Waterproof Mascara For lengthening,volumizing,moisturizing,separating,long lasting | Vegan & Cruelty Free | Free From Parabens & Microplastic Particles (Pack of 1) : Beauty & Personal Care

ADDING PAGES:

To enhance the user experience on our website, we have introduced two dedicated pages: one for adding data files and another for submitting queries. The "Add Data Files" page offers a straightforward interface where users can easily upload and manage their data files, ensuring a seamless process for integrating their information into our system. On the other hand the "Submit Queries" page is designed to help users seek assistance or ask questions efficiently. This page features a user-friendly form that guides users through submitting their inquiries, ensuring that they receive prompt and accurate responses



SWITCHING TO METIS:

After testing our script for 200 products, we had to now test it on a larger dataset; this was not feasible in our local machine, so switched to a remote machine metis by setting up WSL and installing ubuntu.

SHELVE INTEGRATION:

Now was the time to prompt the llm to return only the names of the products and then use Shelve which is a file-based search library to search for the item and print it's details.

```
logger.debug(f"Response: {response}")
shelv = shelve.open("/home/bitsintern/Raghav/ShelveData/Makeup_Eyes")
answer = str(response)
names = answer.split(",")
template = f"""You'll be given a list a categories with their respective subcategories in the given context
=====
Context: {context}
=====

Based on your prior knowledge, you'll have to tell under which category and subcategory does the product given in the below query fall under
Respond only with "category", "subcategory" where <category> is the category of product in the query and <subcategory> is the subcategory of pro

=====
Question: {query}
=====
"""

st.success(Settings.llm.complete(template))
st.write(names)

# Storing the retrieved data
result = {}
shelv_dict = dict(shelv)
for name in names:
    for key, value in shelv_dict.items():
        parts = value.split("\n")
        link, description = parts[0], parts[1] if len(parts) > 1 else ""
        if name.lower() in key.lower() or name.lower() in description.lower():
            if key not in result.keys():
                result[key] = shelv_dict[key]

if result:
    for key, value in result.items():
        parts = value.split("\n")
        link, description = parts[0], parts[1] if len(parts) > 1 else ""

        st.success(key)
        st.write("Buy: ", link)
        st.write("Description: ", description)
```

ADDING FUZZYWUZZY LIBRARY TO THE SHELVE:

For better product matching we decided to implement the fuzzywuzzy library. It uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package.

CODE SNIPPET:

```
st.write(names)

# Shelve database for retrieval
shelv = shelve.open("/home/bitsintern/Raghav/ShelveData/Makeup_Eyes")
names = names.split(",")

# Storing the retrieved data
result = {}
shelv_dict = dict(shelv)
for name in names:
    for key in shelv_dict:
        if fuzz.ratio(name.lower(), key.lower()) > 70: # Using a threshold of 70 for better accuracy
            if key not in result.keys():
                result[key] = shelv_dict[key]

# Showing the description and URL along with the product title
for key, value in result.items():
    parts = value.split("\n")
    link, description = parts[0], parts[1] if len(parts) > 1 else ""

    st.success(key)
    st.write("Buy: ", link)
    st.write(":green[Description]: ", description)
```

CONCLUSION

During my Practice School-1 internship at Paralleldots, Gurugram i gained a comprehensive understanding of various Artificial Intelligence skills. With the help of the free online courses recommended by my mentors, I educated myself with the information retrieval methods, designing a RAG pipeline and also developing the front end of a chatbot.

To put all of our learnt skills to the test, we will build a shopping assistant to enhance user shopping experience in ecommerce sites. This internship has already taught me to be collaborative, work in sync with team members and upskill myself with working under the expertise of my mentors. I've gained insight into the workings of a complex work ecosystem and environment via my interaction with the industry experts. The various technical and soft skills gained during the course of this internship will surely serve as a strong foundation in my future career. I look forward to applying these skills to innovate and build products that positively impact the world.

SUCCESS STORY

Station Name: Paralleldots - IT

Project Domain: AI, Information Retrieval

Project Title: Building a RAG (Retrieval-Augmented Generation) pipeline which assist users with their shopping needs.

Benefits to PS station:

A practice school gives students the opportunity to work on genuine projects in a professional atmosphere, giving them invaluable real-world experience. They may use their theoretical knowledge and build both hard and soft skills—like communication, problem-solving, teamwork, and time management—with the aid of this practical method. It gives students the opportunity to explore many professional paths, enabling them to make educated decisions about their future academic paths. Numerous opportunities exist for networking with businesses and industry professionals, which can be highly beneficial for upcoming job searches. Students are more competitive in the job market and feel more confident about their professional talents when they have practice school experience on their resume. Students who attend practice school are better able to comprehend the dynamics and culture of the industry, which helps them transition into full-time positions following graduation. Employers also use practice schools as a common recruiting tactic, and many of them lead to job offers. Furthermore, by integrating practice schools into the academic curriculum, some courses enable students to earn academic credit. Practice schools can help students manage their educational expenses by offering financial rewards if they are paid for. While exposure to the most recent technological advancements and industry trends guarantees that students stay knowledgeable and flexible professionals, constructive criticism from supervisors promotes both personal and professional progress.

GLOSSARY

1. LLM - Large Language Models are machine learning models that can comprehend and generate human language text.
2. NLP - Natural Language Processing (NLP) is a field of study focused on making computers understand and generate human language.
3. Ollama - Ollama exposes a local API, allowing developers to seamlessly integrate LLMs into their applications and workflows
4. RAG - A technique to provide LLMs with additional information from an external knowledge source, to generate more accurate and contextual answers.
5. Prompt - a specific input or instruction given to an AI or computer program to generate a desired output or response. 6. Gemma7B - a language model developed by Ollama
6. Streamlit- Streamlit turns data scripts to web apps in minutes.
7. Gen AI - generative AI, the aspiration for AI systems to achieve human-like or general intelligence, capable of understanding and solving diverse problems.