MODULE: EBA5004 PRACTICAL LANGUAGE PROCESSING

PROJECT: ALUMCONNECT - INTELLIGENT ALUMNI DISCOVERY & INSIGHTS USING RAG AND LLM

Team: Group18

Team Members:

ESWAR RAJ RAJENDRAN(A0123045W)

SUDHARSHAN MURUGESAN(A0299028R)

SWETA PATTNAIK(A0296537M)

INTRODUCTION:

In today's fast-evolving job market, resumes often lag behind industry expectations, leaving students and early professionals struggling to identify the skills they truly need. Traditional resumes increasingly fail to reflect current industry demands, leaving both students and early professionals at a disadvantage as they struggle to identify which competencies truly matter in their chosen fields. Through extensive research, we've identified three fundamental pain points in modern career navigation:

1. First and foremost, the breakneck pace of technological advancement has created unprecedented challenges. Job descriptions now evolve at a rate that far outpaces traditional education cycles, with new tools, methodologies, and specialized roles emerging constantly. This creates a widening chasm between what candidates present in their resumes and what employers actually seek, resulting in frustrating mismatches during the hiring process.

2. Compounding this issue is the underutilization of valuable alumni networks. While graduates accumulate priceless real-world experience and insights, this wealth of knowledge remains fragmented across various platforms and personal networks. Current career guidance systems fail to systematically harness these authentic success stories, missing crucial opportunities to demonstrate how specific academic foundations translate into viable career pathways.

3. Lastly, the limitations of existing solutions exacerbate these problems. Outdated search mechanisms force users to rely on manual, time-intensive methods that frequently yield irrelevant or superficial results. These platforms lack the sophisticated contextual understanding required to interpret nuanced career queries and match them with appropriate opportunities or role models.

This is where AlumConnect comes in. Our solution is an AI-powered alumni chatbot that transforms scattered data into actionable insights, helping students navigate their career paths.

SOLUTION:

Our solution to the problem is to develop an ai-powered alumni search chatbot that efficiently retrieves and presents alumni profiles based on job titles, companies, industries, and skills. By leveraging advanced natural language processing and machine learning, our solution goes beyond basic keyword searches to deeply understand queries about job titles, companies, industries, and skill combinations. The system intelligently analyses career trajectories, interpreting nuanced requests like 'alumni who transitioned from engineering to product management' or 'marketing leaders in healthcare startups.' through its conversational interface, users can progressively refine searches, compare career paths, and extract actionable advice - all while the underlying rag architecture ensures responses are grounded in accurate, up-to-date alumni data. This creates a dynamic bridge between academic preparation and real-world career success, helping users discover relevant role models, identify skill gaps, and make informed decisions about their professional futures

ARCHITECTURE :

AlumConnect operates through three integrated pipelines that ensure continuous learning and real-time responsiveness.

Training-pipeline

Whenever a new dump of alumni data arrives, a batch job is triggered: raw PDFs/CSVs are parsed, cleaned and stored in Firestore. A data-prep script then auto-generates question–answer pairs from each profile, tokenises them, and writes a consolidated JSONL file to Cloud Storage. We fine-tune our base Llama-3 checkpoint (or Gemini via adapter tuning) on this file with LoRA and mixed-precision training, log metrics in Weights & Biases, and push the resulting model artefact to a private Hugging Face repo for later deployment.

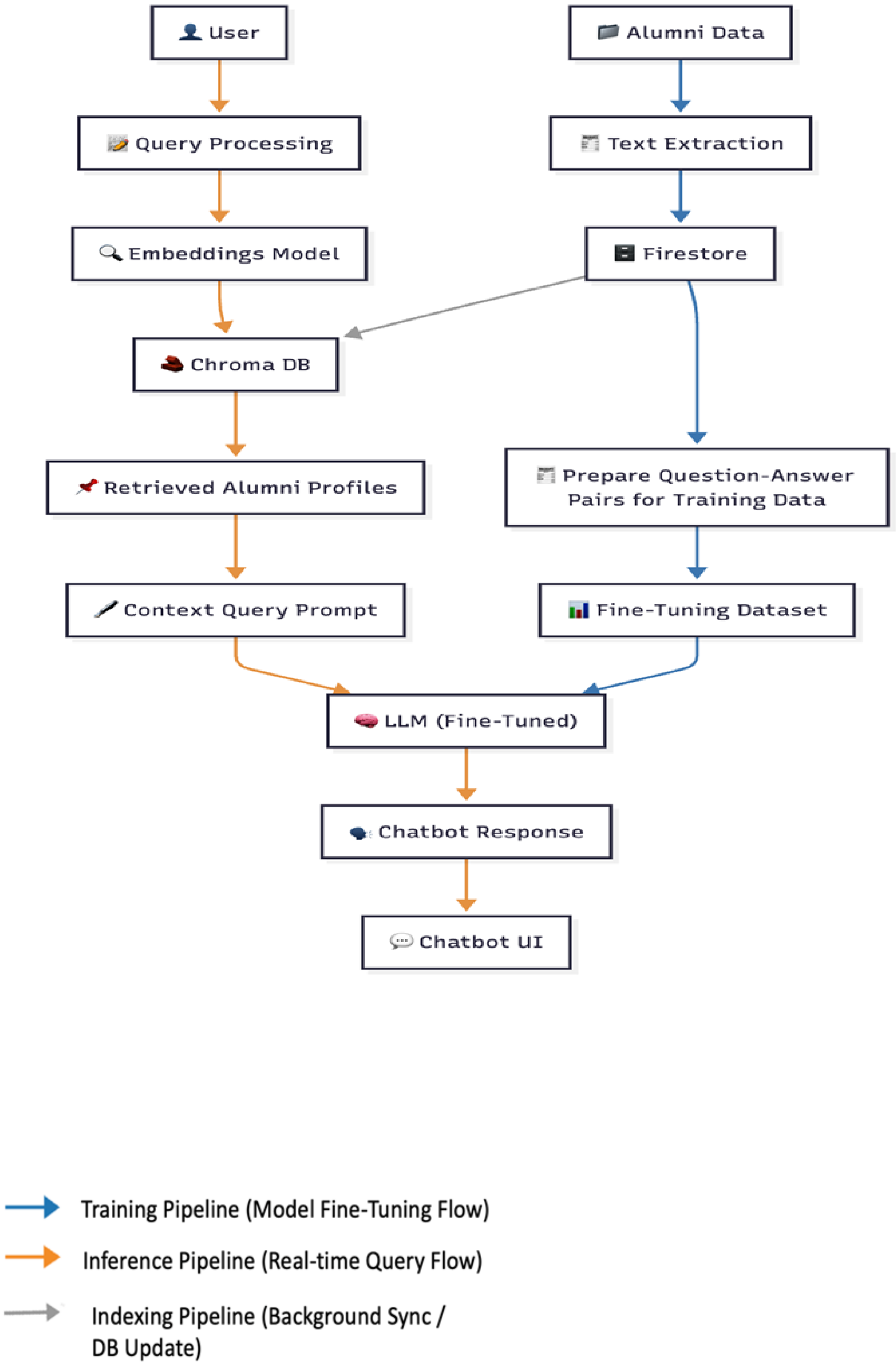
Inference-pipeline

A user query enters through the Gradio UI, is appended to ConversationBufferMemory, embedded with Google embedding-001, and passed to the Chroma vector index. The top-k profile chunks are returned, merged with the question inside a prompt template, and sent to the fine-tuned LLM served behind a lightweight FastAPI endpoint. The model responds at temperature 0.2; the answer is streamed back to the UI and stored with the query for audit and future re-ranking experiments.

Indexing-pipeline

Running in the background, a Cloud Function listens to Firestore document updates. On any change it re-extracts the text, re-embeds the affected chunks, and upserts them into Chroma. This keeps the vector store in sync without blocking either training or live inference.

Together, the three pipelines let us (1) continuously improve the model, (2) serve real-time, context-grounded answers, and (3) ensure the retrieval layer always reflects the latest alumni records.



MODELS USED :

AlumConnect leverages two state-of-the-art language models to deliver intelligent alumni insights. For our primary system, we fine-tuned Meta's **Llama-3-8B-Instruct** model using QLoRA on our custom alumni dataset, enabling it to develop specialized understanding of career trajectories and profile patterns. This fine-tuned model is integrated with our RAG (Retrieval-Augmented Generation) pipeline during inference, combining its tailored knowledge with real-time data retrieval for accurate, context-aware responses.

For comparison and benchmarking, we incorporated Google's **Gemini-1.5-Flash** in its pretrained form, utilizing its strong out-of-the-box capabilities to evaluate performance differences between generic and domain-adapted LLMs. This dual-model approach allows us to balance the precision of a customized solution with the generalizability of a powerful foundation model, while providing valuable insights into the trade-offs of fine-tuning versus retrieval augmentation.

LLM	Version	Comment
Llama	Meta-Llama-3-8B-Instruct	We have fine-tuned this model with our dataset and integrated with RAG during inference.
Gemini	gemini-1.5-flash	We have used this pretrained model for comparison with the finetuned model.

TOOLS & FRAMEWORKS :

Coding Environment	Google Collab
Database Tools	Firestore
Vector Stores	Qdrant, ChromaDB
Embedding Models	Sentence Transformers (all-MiniLM-L6-v2), Google Generative AI Embeddings
LLM Integration Tools	LangChain (chains, memory, prompt templates)
	Google Generative AI (via langchain-google-genai)
	Ngrok (for local-to-web LLM API exposure in early prototype)
Configuration	python-dotenv (for API keys & environment variables)
Utility Libraries	requests, json, tqdm, shutil, traceback, re
UI Framework	Gradio

DATASETS USED:

For our development phase, we utilized a synthetic dataset comprising approximately 3,500 professionally-generated resumes sourced from Kaggle.

The data processing pipeline begins with our dedicated Jupyter notebook (1_Resume_Text_Extraction_N_To_DB_Firestore.ipynb), which systematically ingests resume records from both JSON and plaintext sources. Using a combination of string manipulation techniques and regular expression patterns, the extraction process identifies and categorizes key profile elements including:

- Personal identifiers (name, email, contact information)
- Professional details (job titles, work experience)
- Competency markers (skills, education, languages)
- Geographic information (location)

Each parsed resume undergoes standardization, being transformed into a structured Python dictionary with uniform field keys to ensure consistency. The processed records are then securely stored in Google Firestore under the 'alumni_profiles' collection, where each profile is assigned a unique Firestore document ID for efficient retrieval and management. This structured approach to data handling enables reliable performance in both our training and inference pipelines while maintaining data integrity across the system.

TRAINING AND INFERENCE FOR BOTH MODELS :

LLAMA :

LLM Fine-Tuning with QLoRA (LLaMA-3 8B)

Meta-Llama-3-8B-Instruct is adapted to the alumni domain with the QLoRA method, allowing the entire run to fit on a single Google Colab T4 GPU. This adaptation allows the system to understand domain-specific queries while operating within constrained computational resources.

- **Data preparation:** Alumni Q-A pairs are exported to JSONL and tokenised with the native Llama tokenizer.
- **LoRA settings**: rank 32, α 64, dropout 0.05, applied to the projection layers (q / k / v / o). Quantisation is 4-bit via bitsandbytes.
- **Training loop**: AdamW at 2×10^{-5}, physical batch 4 with gradient-accumulation 4 (effective 16). Training runs in fp16 with gradient checkpointing; checkpoints are written every 500 steps over 20 230 total steps.
- **Tooling**: Hugging Face transformers, peft, and bitsandbytes orchestrate the run; model artefacts are pushed to a private HF repository.

Retrieval-Augmented Generation (RAG) with the Fine-Tuned Model

Embeddings & index – text segments are embedded using all-MiniLM-L6-v2 and stored in a remote Qdrant vector DB.

Front-end

A Gradio chat UI captures user questions and displays streamed answers.

Inference Flow :

- Query capture – question enters via Gradio.
- Similarity search – the query is embedded and sent to Qdrant; the top-k matching documents are returned.
- Generation – query + retrieved passages are posted to the fine-tuned LLaMA endpoint (exposed through an Ngrok tunnel). The model produces a context-grounded answer, which is streamed back to the user.

This design combines a compact, domain-tuned language model with an external vector store, delivering accurate responses under tight GPU constraints while remaining easy to scale.

GEMINI:

Retrieval-Augmented Generation Pipeline – Gemini Variant

Data ingestion and storage

Alumni résumés are extracted from Google Firestore, segmented into coherent text chunks, and embedded with *Google Generative AI embedding-001*. The dense vectors are written to a local **ChromaDB** collection, which therefore acts as the semantic index while Firestore remains the source-of-truth documents store.

Retrieval strategy

LangChain exposes the Chroma collection through .as_retriever(). At query time the user question is embedded, scored for cosine similarity, optionally filtered on metadata (e.g. cohort, skill tags), and re-ranked. The top-k passages (default 10) form the contextual evidence set.

Inference workflow

The evidence set, the current question, and the running chat transcript are passed to LangChain's ConversationalRetrievalChain.

- *Memory*: ConversationBufferMemory maintains dialogue continuity.
- *LLM*: responses are produced by the Gemini model via ChatGoogleGenerativeAI at temperature 0.2.
- *Prompting*: a structured template injects the retrieved passages, the question, and style / safety instructions before the LLM call.

User interface

A lightweight Gradio front-end presents a chatbot window, a text-input box, and *Send / Clear* controls. The handlers below wire UI events to back-end logic.

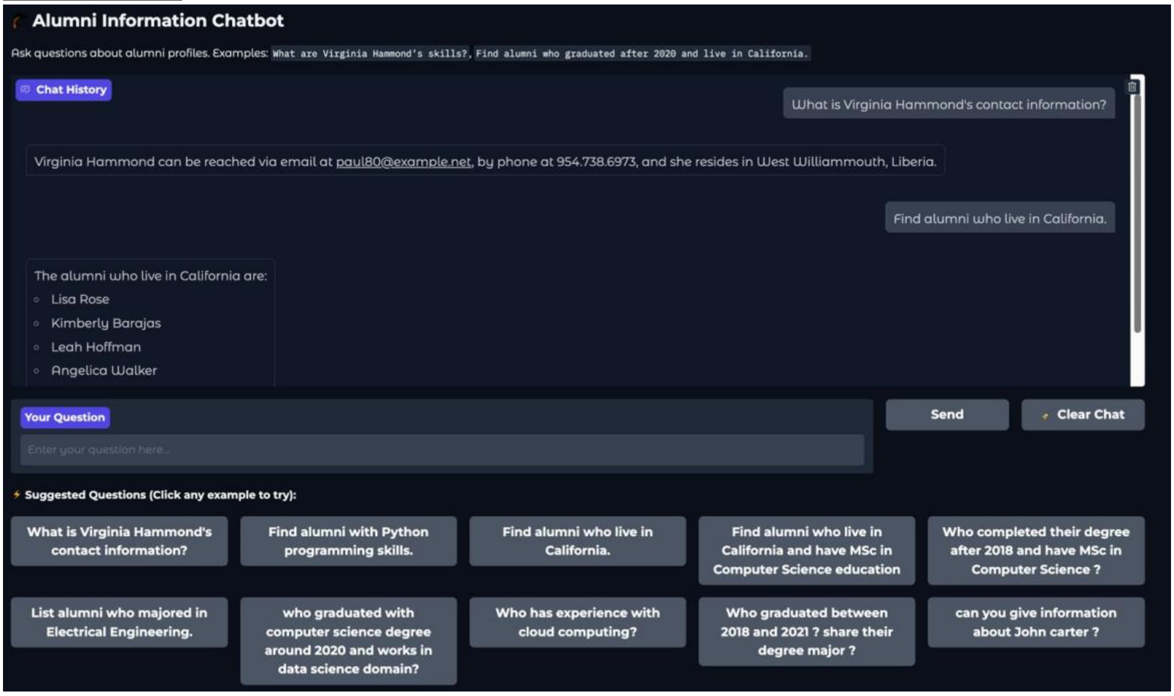
Pseudo code :

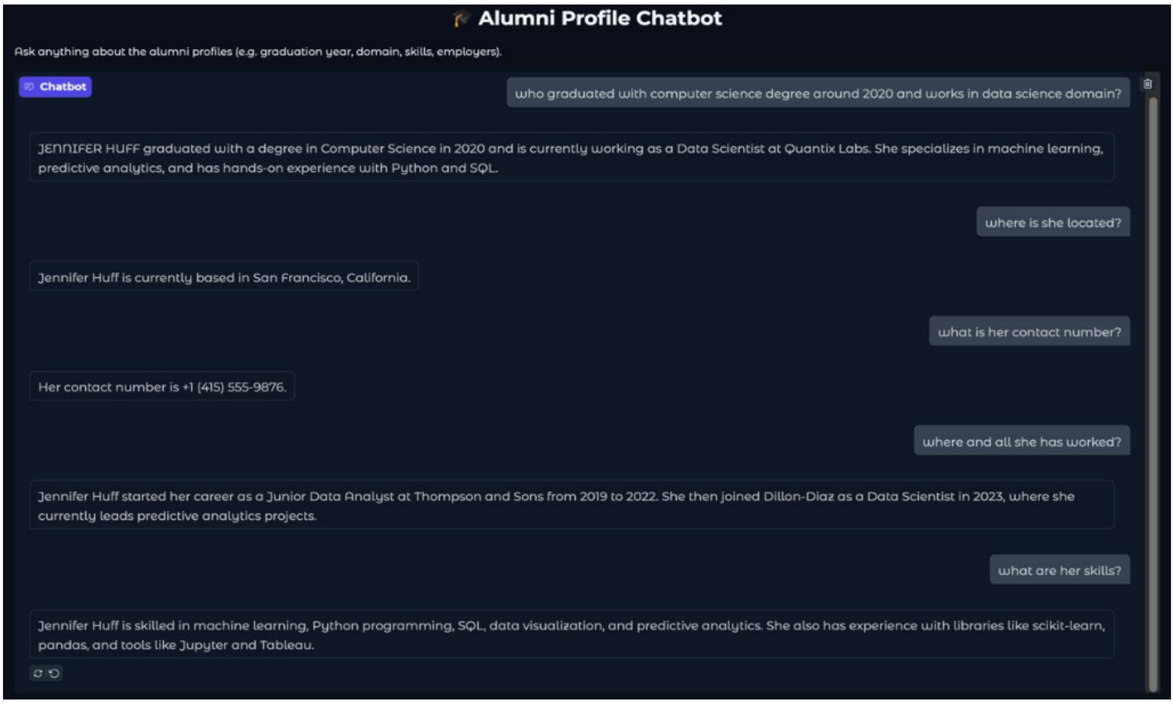
```
# Initialisation

set_api_keys()

chroma = create_vector_db()

embedder = create_embedder()


# Query loop

def answer_query(question: str) -> str:

    q_vec = embedder.encode(question)

    hits  = chroma.similarity_search(q_vec, k=5)

    ctx   = format_profiles(hits) if hits else "No relevant profiles found."
```

```
    prompt = build_prompt(ctx, question)
    return gemini_generate(prompt)


def respond(message, history):
    reply = answer_query(message)
    history.append((message, reply))
    return history


def clear():
    return []
```

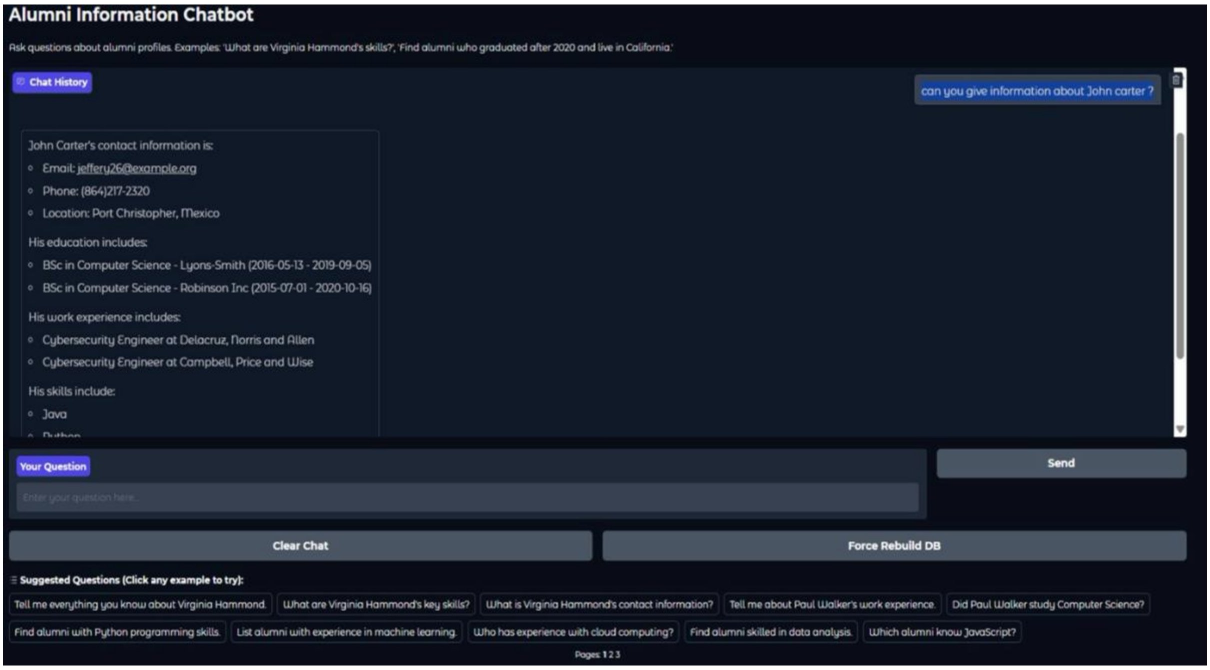
CHATBOT UI SCREENSHOTS:

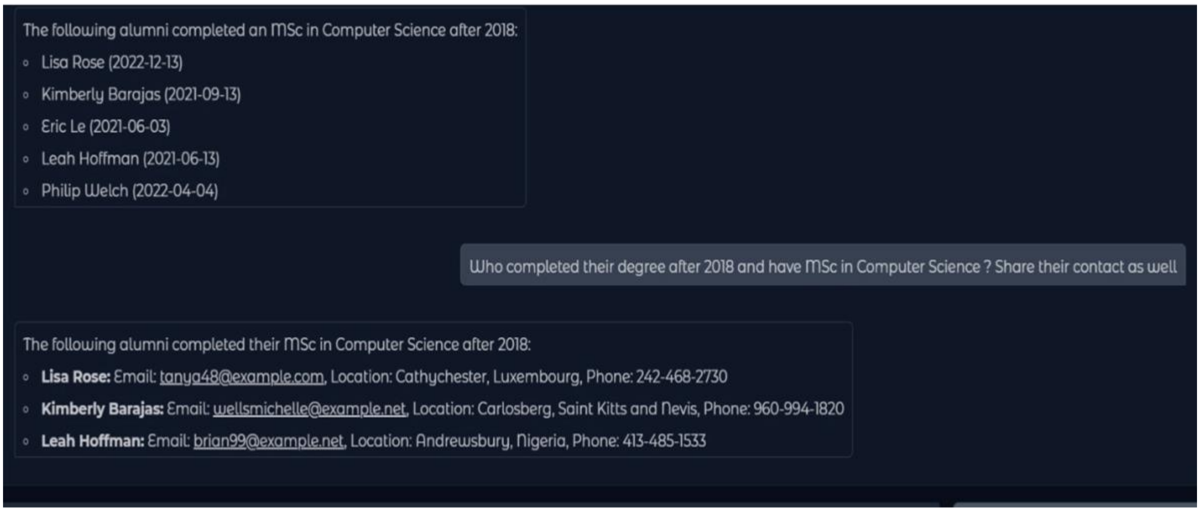
<u>LLAma model :</u>





Gemini model :





EVALUATIONS:

We are using Rouge metric for our evaluation. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a metric used to evaluate text generation quality by comparing model responses to reference answers. It is widely adopted in tasks like summarization and question answering.

ROUGE-1 F1:

•Measures word-level overlap (unigrams) between the generated answer and the reference.

•The **F1 score** balances precision (accuracy of words used) and recall (coverage of key words).

ROUGE-L F1:

•Based on the **Longest Common Subsequence (LCS)** — measures how well the sequence and structure of the generated text matches the reference, rewarding coherence.

QUESTION	RESPONSE LLAMA	RESPONSE GEMINI	ROUGE-1 F1	ROUGE-L F1
What is Virginia Hammond's contact information?	Virginia Hammond can be reached via email at paul80@example.net, by phone at 954.738.6973, and	Virginia Hammond's contact information is: Email: paul80@example.net	0.62	0.62
Find alumni with Python programming skills.	Alumni skilled in Python for data analysis and visualization include Virginia Hammond, John Smith, Alice Johnson, and Bob Williams.	Virginia Hammond: "Perfected data analysis and data visualization using Python and Tableau." John Smith: "Perfected data analysis and data	0.42	0.33
Find alumni who live in California.	The alumni who live in California and have an MSc in Computer Science are: •Lisa Rose	The following alumni studied Computer Science: Michael Bartlett: BSc in Computer Science (2017-08-18 - 2022-11-28), MSc in Computer Science (2018-09-09 -	0.18	0.14
Find alumni who live in California and have MSc in Computer Science education ?	The following alumni live in California and have an MSc in Computer Science: •Lisa Rose	The following alumni have an MSc in Computer Science: Lisa Rose: Jones, Villegas and Williamson (2018-08-29 - 2022-12-13)	0.32	0.31
Who completed their degree after 2018?	The following alumni completed their degree after 2018: •Mathew Campbell – MSc in Computer Science (2020-	The following alumni completed degrees after 2018: Mathew Campbell: MSc in Computer Science (2020-09-11)	1.00	1.00
Who completed their degree after 2018 and have MSc in Computer Science ?	The following alumni completed their degree after 2018 and have an MSc in Computer Science:	Lisa Rose (2022-12-13) Kimberly Barajas (2021-09-13)	0.56	0.56
List alumni who majored in Electrical Engineering.	The following alumni majored in Electrical Engineering: •Mathew Campbell	Mathew Campbell Suzanne Lee	0.84	0.84
who graduated with computer science degree around 2020	The following alumni graduated with a Computer Science degree around 2020: •Jonathan Walker – MSc in Computer Science (2020-	The following alumni completed a Computer Science degree around 2020: Jonathan Walker: MSc in Computer Science (2020-02-	0.95	0.95
who graduated with computer science degree around 2020 and works in data science domain?	This question cannot be answered definitively with the provided data. While we know that: •Jonathan Walker graduated with an MSc in	I cannot answer this question. The provided text mentions alumni who graduated around 2020 with Computer Science degrees, and it mentions alumni with	0.33	0.23
Who graduated between 2018 and 2021 ? share their degree major ?	The following alumni graduated between 2018 and 2021, along with their degree majors: •Amber Holt – BSc in Computer Science (Graduated:	The following alumni graduated between 2018 and 2021 with the following majors: Amber Holt: BSc in Computer Science (2021-10-21)	0.90	0.90
can you give information about John carter ?	Yes, here is the information about John Carter: •Email: jeffery26@example.org •Phone: (864) 217-2320 •Location: Port Christopher, Mexico	Email: jeffery26@example.org Phone: (864)217-2320 Location: Port Christopher, Mexico His education includes:	0.88	0.88

Responses from finetuned Llama model and Gemini model :

QUESTION	RESPONSE LLAMA	RESPONSE GEMINI
What is Virginia Hammond's contact information?	Virginia Hammond can be reached via email at paul80@example.net, by phone at 954.738.6973, and she resides in West Williammouth, Liberia.	Virginia Hammond's contact information is: Email: paul80@example.net Phone: 954.738.6973 Location: West Williammouth, Liberia
Find alumni with Python programming skills.	Alumni skilled in Python for data analysis and visualization include Virginia Hammond, John Smith, Alice Johnson, and Bob Williams.	Virginia Hammond: "Perfected data analysis and data visualization using Python and Tableau." John Smith: "Perfected data analysis and data visualization using Python and Tableau." Alice Johnson: "Perfected data analysis and data visualization using Python and Tableau." Bob Williams: "Perfected data analysis and data visualization using Python and Tableau."

Find alumni who live in California.	The alumni who live in California and have an MSc in Computer Science are: • Lisa Rose • Kimberly Barajas • Leah Hoffman • Angelica Walker • Jacqueline Ford • Lisa Williamson	The following alumni studied Computer Science: Michael Bartlett: BSc in Computer Science (2017-08-18 - 2022-11-28), MSc in Computer Science (2018-09-09 - 2021-07-04) Angelica Walker: BSc in Computer Science (2017-08-12 - 2023-02-03), MSc in Computer Science (2016-12-05 - 2019-07-17) Mathew Campbell: BSc in Computer Science (2015-04-07 - 2021-10-01), MSc in Computer Science (2017-10-10 - 2020-09-11) Lisa Williamson: MSc in Computer Science (2017-03-14 - 2019-08-13), MSc in Computer Science (2015-04-02 - 2022-03-23) Chris Allen: BSc in Computer Science (2016-07-04 - 2022-01-05), BSc in Computer Science (2016-09-09 - 2019-10-12) Lisa Rose: MSc in Computer Science (2018-08-29 - 2022-12-13) Kimberly Barajas: MSc in Computer Science (2017-05-15 - 2021-09-13) Leah Hoffman: MSc in Computer Science (2018-04-17 - 2021-06-13) Jacqueline Ford: MSc in Computer Science (2015-09-28 - 2019-03-17) Christopher Lopez: BSc in Computer Science (2018-11-17 - 2020-11-12), BSc in Computer Science (2015-08-21 - 2021-11-09)

From the scores, what we see is:

High Scores (ROUGE > 0.9)

- Example: *“Who completed their degree after 2018?”*
- Both finetuned LLaMA and Gemini scored ~0.995 on ROUGE-1 and ROUGE-L, indicating highly accurate and well-aligned responses.
- Example: *“Graduated between 2018 and 2021?”*
- Gemini scored 0.90+, reflecting strong understanding of date-based queries.

Mid Scores (ROUGE 0.5–0.8)

- Example: *“Majored in Electrical”* — Gemini reached ~0.83
- Shows partial match with correct names and degrees, but possible variation in wording.

Low Scores (ROUGE < 0.4)

- Example: *"Alumni in California"* — finetuned LLaMA scored 0.17, Gemini slightly better.
- Indicates vague, incomplete, or hallucinated responses likely due to insufficient or noisy context.
- *"Graduated around 2020 and works in data"* — Poor scores reflect inability to handle compound filters in a single query.

CHALLENGES & LIMITATIONS:

- **GPU Constraints:** Local / free-Colab GPUs hit memory ceilings for embeddings and fine-tuned LLMs; real-time tasks often need costlier cloud APIs, adding latency.
- **LLM Hallucinations:** Sparse or messy alumni data lets models invent details; stronger retrieval grounding and tighter prompts are essential.
- **ngrok Free Tier:** Free tunnels throttle bandwidth and uptime, limiting reliable backend access.
- **Prompt Fragility:** Minor template tweaks ripple through answers; over-stuffed guidelines blur together, hurting consistency and forcing exhaustive regression checks.
- **Data Privacy & Scraping:** Platforms like LinkedIn block large-scale scraping; without clear consent we relied on synthetic data, constraining realism and scale.

CONCLUSIONS:

- AlumConnect bridges the critical gap between static resumes and the evolving job market by harnessing the power of underutilized alumni data.
- Through a Retrieval-Augmented Generation (RAG) framework and fine-tuned LLMs, the system tries to intelligently match user queries with relevant alumni career paths, offering actionable, personalized insights.
- Key accomplishments include the development of workflows—retrieval and fine-tuning, integration of advanced models like LLaMA 3 and Gemini, and the creation of an interactive chatbot.
- Looking ahead, the project aims to scale by incorporating real university alumni datasets, deploying to the cloud for wider accessibility, and adding real-time feedback mechanisms to iteratively enhance LLM accuracy and relevance.

REFERENCES:

- LangChain RAG Overview: https://python.langchain.com/docs/use_cases/question_answering/
- Hugging Face Fine-Tuning Guide : https://huggingface.co/docs/transformers/training
- FAISS (Facebook AI Similarity Search) : https://faiss.ai/
- FastAPI for AI Chatbot APIs : https://fastapi.tiangolo.com/
- Meta AI's Llama 3: https://huggingface.co/docs/transformers/main/model_doc/llama3
- https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset
- https://ai.google.dev/gemini-api/docs/models#gemini-1.5-flash-8b