For this challenge, we were asked to build a chatbot that helps guests during Marbet's incentive trips by answering questions about schedules, packing, ship services, and travel instructions. The idea was to use AI to make event info easy to access without having to dig through PDFs or ask a staff member. We were really excited to work on this because it felt like something that could actually be used in a real setting. Plus, it was a chance to practice using newer AI tools like LangChain and Ollama, which we hadn't worked with before.

The data we received came in the form of official documents-pdfs-from Marbet. There were things like a daily activities overview, a packing checklist, a ship guide with policies and services. Some documents were clean and nicely formatted, while others were kind of messy, long, scanned pages or styled in a way that made it hard to extract text. So, a big part of the challenge was figuring out how to organize and standardize all this content into something the AI could understand.

Once we understood the kinds of documents we were working with, we moved into data preparation. We started by converting everything into plain text. Then, we broke the text into smaller chunks so the AI wouldn't be overwhelmed by giant walls of text. Chunking was really important because the AI retrieves info based on these small pieces, and smaller chunks give it a better chance of finding the exact answer. After that, we turned the text chunks into vector embeddings using Ollama's llama3.2 model, and stored them in a FAISS vector database. This step basically made our documents searchable in a smart way.

For modeling, we used LangChain's RetrievalQA pipeline and connected it to the Ollama model. This setup allowed the chatbot to take a question, search through the vectorized chunks of the documents, and generate a useful answer based on what it found. We also created a system prompt that tells the AI it's a helpful assistant, and that it should only answer based on the documents. This was important because we didn't want the AI to guess or make things up. The responses needed to be grounded in what Marbet actually provided. The tone we aimed for was clear, polite, and professional, but not robotic.

We evaluated the chatbot by asking realistic questions that guests might have, like "What activities are planned for Day 2?", "What's on the packing list?", "Where is the spa located?", or "Do I need a visa for the trip to Canada?" For the most part, the chatbot gave really solid answers, especially when the document chunks were clean and specific. If the question was vague or the info was hidden in a complicated paragraph, the chatbot sometimes struggled or gave general answers. But overall, it was accurate, fast, and able to handle different ways of phrasing the same question.

Right now, the chatbot runs locally from a notebook or script, and it connects to the BUas Ollama server. So, it's not a polished app yet, but it works. We also documented

the whole process so that anyone could reuse the pipeline with new documents for other events. The chatbot is flexible enough to be adapted for future use.

In terms of next steps, there's definitely a lot we'd like to add. A proper user interface would make it more usable for non-tech people. Multilingual support would make it more inclusive, since Marbet operates in both Germany and Spain. We'd also love to improve document parsing, especially for scanned or image-heavy PDFs. And maybe even add features like voice input, so guests could ask questions out loud and get spoken responses. But for now, we're happy with what we built, and proud of how far we got within the time we had.

Overall, this challenge taught us a lot, not just about RAG and chatbots, but about working with real-world documents, cleaning messy data, structuring a system from scratch, and thinking about the user experience. We feel like this project helped us connect AI theory to something practical and meaningful.