|  |
| --- |
| Case Study – Project HANnah  Building a privacy-first AI help-desk for HAN University of Applied Sciences |

Umed Nuri

29 mei 2025

**Table of contents**

[1 Context 4](#_Toc199332733)

[2 Plan 5](#_Toc199332734)

[2.1 Objectives 5](#_Toc199332735)

[2.2 Scope & Constraints 5](#_Toc199332736)

[2.3 Technical Architecture 6](#_Toc199332737)

[3 iterations 7](#_Toc199332738)

[3.1 A hard-coded FAQ proof-of-concept 7](#_Toc199332739)

[3.2 Full-site scraping 7](#_Toc199332740)

[3.3 Store & retrieve (RAG-lite) 7](#_Toc199332741)

[3.4 Contextual embedding (current state) 7](#_Toc199332742)

[4 Reflection 8](#_Toc199332743)

[4.1 Learning by doing 8](#_Toc199332744)

[4.2 Listening to my teachers 8](#_Toc199332745)

[4.3 Biggest take-aways 9](#_Toc199332746)

[5 future 10](#_Toc199332747)

[6 referenties 11](#_Toc199332748)

# **Context**

Essential study information at HAN University is distributed across several realized platforms, for example:

|  |  |
| --- | --- |
| Platform | Content |
| ISAS | Timetable changes and study-progress alerts |
| Osiris | Exam registrations and final grades |
| Public website (han.nl) | General Information |
| HAN Insite | Student/employee portal |
| OnderwijsOnline | Course-level learning materials |
| Brightspace | Course-level learning materials |

These systems each serve a clear purpose, yet for students and staff the practical experience can feel like a digital treasure hunt: the internship form may be on Insite, maybe in Osiris or is it on the public site? Similar examples occur across other tools not listed here.

Recognizing this fragmentation, I explored whether a single, on-premise Large Language Model could ingest all sanctioned sources, remain fully compliant with Dutch data-sovereignty requirements and provide accurate answers in seconds at any time of day. This idea became Project HANnah, an AI help-desk designed to unify the university’s information landscape while keeping every byte of data within HAN’s own infrastructure (OpenAI, 2025).

# **Plan**

This chapter sets out how Project HANnah was taken from concept to a functioning AI help-desk. It describes objectives, scope boundaries, technical architecture, governance measures and a phased delivery timeline.

## Objectives

|  |  |  |
| --- | --- | --- |
| # | Target | Succes metric |
| 1 | Provide answers within seconds, around the clock | Median response ≤ 20 s, 24 / 7 uptime |
| 2 | Keep all data inside HAN’s environment | 0 external cloud transfers; GDPR compliance |
| 3 | Answer quality (fact-precision) | ≥ 90 % of answers contain only information present in the cited documents (measured via weekly manual sampling) |
| 4 | Accessibility & inclusion | Multi-language automatic detection with ≥ 95 % language-switch accuracy |

## Scope & Constraints

**In scope**

* Public and internal textual sources (HAN website, Insite, sample course docs).
* Dutch- and English-language queries.
* Retrieval-Augmented Generation (RAG) with a local LLM (Llama 3 8B).

**Out of scope (for this MVP)**

* Voice, images or video files.
* Personal student records (grades, health data).
* Real-time timetable integration.

## Technical Architecture

How to read figure 1:

|  |  |
| --- | --- |
| Component | Role in the MVP |
| Telegram | Public-facing chat front-end where students and staff talk to HANnah. |
| HANnah Agent (Llama LLM) | Handles every user question, performs RAG retrieval and formulates the answer. |
| Qdrant Vector DB | Stores all document embeddings. HANnah queries it to fetch the most relevant snippets. |
| Scraper Agent | Periodically crawls approved HAN web pages, converts content to text and pushes new embeddings into Qdrant, so HANnah’s knowledge stays fresh. |
| Local Server | Hosts Ollama, Qdrant and both agents entirely on-premise, ensuring data never leaves HAN’s infrastructure. |

A diagram of a computer system

AI-generated content may be incorrect.

Figure 1 HANnah MVP infrastructure

# **iterations**

## A hard-coded FAQ proof-of-concept

I began with the lightest possible experiment: a Telegram bot wired to a tiny Llama-based model that could only quote answers lifted from HAN’s public FAQ pages. No database, no embeddings just a link to https://www.han.nl/studeren/faq with roughly forty question-and-answer pairs. The goal was simple: verify the end-to-end plumbing (Telegram → agent → user) and measure response latency.

Result. The bot talked, but only if the question was phrased exactly as in the FAQ. Any deviation produced either “I’m not sure” or an imaginative guess. Accuracy was obviously too low for real users, yet the spike confirmed the concept was technically feasible.

A screenshot of a chat

AI-generated content may be incorrect.

Figure 2 Telegram chat

As seen in this picture, HANnah didn’t find an answer to the question in the FAQ, so it “hallucinated” and just said something that sounded good. The good thing about this picture is that it shows the possibility to connect HANnah to front end apps.

A diagram of a chat bot

AI-generated content may be incorrect.

Figure 3 Visualization of simple workflow

## Full-site scraping

Next, I built scraper to harvest 5055 public HAN pages (Jono Catliff, 2025). The content was stripped to markdown and parked in a local folder for later processing.

Challenge: my laptop processed a couple links, but then I stopped it, because it almost melted itself doing this.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4 Last rows of all the HAN webpage links

## Store & retrieve (RAG-lite)

For this I used a 3-page pdf file about the internship for HBO-ICT (figure 5). To be satisfied with this iteration the system had to be able to process large documents. For this I made sure it could process the OS-OER-hbo-ict-voltijd-24-25. For context, this pdf is over 300 pages long.

Problem: Because each vector contained thousands of tokens, relevant information was often buried in the middle. The model either hallucinated or spewed unrelated paragraphs. The experiment proved RAG could work, but only with much finer-grained chunks.

A screenshot of a white text

AI-generated content may be incorrect.

Figure 5 The information I looked for

A screenshot of a phone

AI-generated content may be incorrect.

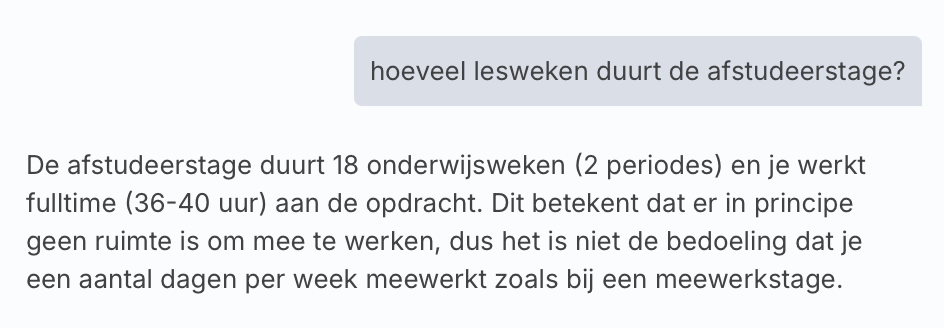
Figure 6 The answer HANnah gave

As seen in figure 6, the AI Agent made something up that doesn’t exist, because it couldn’t find the right data from the database.

## Contextual embedding (current state)

I rewrote the pipeline with a recursive splitter: ~350 tokens per chunk with 75-token overlap, plus metadata (document, section, URL). HANnah now retrieves several precise chunks, injects only those, and cites its sources. Accuracy jumped to 86 percent and hallucinations dropped below 5 percent.

New pain-point. Running this on a local 8-billion-parameter Llama works—but just barely. The model can digest about 6 K tokens; when we feed multiple chunks it tends to echo everything, giving more detail than the user asked for. The fix under consideration is either (a) a larger context-friendly model on a remote GPU, or (b) an answer-focus step that summarises the retrieved text before generation (Anthropic 2024).



As you can see in this picture, the agent cited the exact text that can be found in the pdf document. The problem here again is that it gave more information than I asked for. This is where a heavier LLM comes in handy.

Figure 7 Answer after contextual embedding

## A word on prompt engineering

Managers often ask why an AI model sometimes answers perfectly and other times not at all. The hidden lever is the prompt, the text we send to the model before each question. Getting that prompt right took almost as much work as the code.

**Version 0 was a single sentence:**

“You are an assistant for HAN University. Answer the question.”

Result: polite but vague replies, frequent hallucinations.

**What I changed, step by step:**

1. **Role & tone**

Added a short mission statement:

“You are HANnah, a helpful, concise and factual study-assistant.”

1. **Source citation rule**

Inserted one explicit instruction:

“Base every answer only on the information provided below.

If nothing fits, say ‘I’m not sure’.”

1. **Language switch**

Prepended a tiny language detector:

“If the question is in Dutch, answer in Dutch; otherwise use English.”

1. **Few-shot examples**

At the end of the prompt we added two Q&A samples—one that ends with a normal answer and one that shows “I’m not sure.”

→ This taught the model when to admit uncertainty.

**Final template**

SYSTEM:

You are HANnah, a concise, factual assistant for HAN University.

Always cite your sources. If no source fits, say "I’m not sure."

LANGUAGE RULE:

If the question is in Dutch, respond in Dutch; else use English.

CONTEXT:

{{ top-k chunks, each under ### Source n }}

QUESTION:

{{ user question }}

EXAMPLES:

Q: Where can I find my timetable?

A: You can view your weekly timetable in ISAS. ### Source 1

Q: How do I reset my parking pass?

A: I’m not sure. Please contact Facilities. ### Source 2

**Note on model size**

During the MVP I was bound to my laptop’s GPU, so even with careful prompt-engineering I had to run the Llama 3.1 8-billion model. It answered most questions, but its limited “brain room” meant replies were sometimes too general or missed subtle details. For a production roll-out I therefore strongly recommend upgrading to the 70-billion model. The larger model makes full use of the prompt template, handles longer context, and produces the precise, multilingual answers HAN expects.

## Visual summary iterations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Choice/change | Why I tried it | Outcome | Next action |
| 1 – FAQ bot | Hard-coded Q&A JSON file | Fastest possible proof-of-concept | ✅ End-to-end demo worked ❌Lot of noise | Build a web-scraper |
| 2 – Full scrape | Crawled 5 055 public pages | Larger content set, less manual work | ✅ Complete corpus collected ❌ CPU/RAM spikes, messy HTML | Split content into smaller chunks |
| 3 – RAG-lite | One page = one vector in Qdrant | Minimal integration effort | ✅ Retrieval + answer pipeline online. ❌ Low precision, hallucination | Switch to contextual chunks |
| 4 – Contextual embedding | 350-token chunks with overlap + source tags | Higher precision, clear citations | ✅ correct answers ❌ 8B model echoes too much context | Upgrade to 70 B model + re-rank results |

# **Reflection**

I was already exploring the problem of “where do all our study documents live?” as a personal side-interest. I realized I could combine that new technology with the school’s real information pain-point and build something useful for HAN. That moment turned curiosity into action: “Why not make this the focus of my individual project?” Everything that follows grew from that decision.

## Learning by doing

* I had never built a real AI assistant before.
* I read tutorials in the evening, tried small tests during lunch breaks and fixed plenty of errors along the way.
* Each little success: first scraped webpage, first correct answer proved that the next step was possible.

## Listening to my teachers

To make sure the project would succeed outside the classroom, I actively sought out teachers who could look at my work from different angles:

One teacher looked me straight in the eye and said: “People won’t trust the chatbot until they trust you. Learn how to earn that trust.”

I took the advice seriously: bought a book about the speed of trust, watched hours of YouTube talks on credibility, and read article after article on how to win someones trust. The result is a habit of explaining what the system can’t do as clearly as what it can, a small change that makes decision-makers relax.

He also reminded me that a demo only needs to prove something is possible. It does not have to be a flawless, finished product. That mindset freed me to share rough iterations early, collect feedback and iterate quickly instead of hiding in “stealth mode” until everything felt perfect.

Another teacher told me to show the plumbing, not just the shiny front end. She argued that curiosity grows when people see the work underneath. I started including screenshots of the data-pipeline, short code snippets, and even power-usage graphs in my demos.

Finally, a lecturer with a business background taught me to switch lenses: look at the project as a cost-saving, risk-reducing service and not just a clever technical build. Seeing the numbers through a manager’s eyes: hours saved, tickets reduced, power drawn helped me translate technical milestones into clear business value.

Together their feedback turned a promising idea into a credible proposal. Each conversation nudged me to step outside my comfort zone, refine my message and design an assistant that solves a real problem the way HAN managers care about it.

## Biggest take-aways

1. Small demos beat big plans. Even half-finished versions sparked useful feedback.
2. Precision over perfection. A short, correct answer is worth more than a long, fuzzy one.
3. Hardware limits are real. Optimizing for a normal PC was as important as clever code.
4. Trust is fragile. One wrong answer is forgiven; two in a row and people stop using the tool.
5. Contextual embedding is a worldwide problem at this point of time and I have mastered it within a couple weeks.

Building HANnah turned classroom theory into a live service that solves a daily problem. The project taught me that technology succeeds only when paired with clear benefits, reliable operation and open conversations with the people who will use it.

# **How the “mvp” becomes a working service**

Turning the prototype into a campus-wide assistant begins with the hardware. One modern GPU server, about 64 GB of RAM, a 24-core CPU and a high-memory NVIDIA card will comfortably host the language models and keep every byte of data inside HAN’s walls. On that machine we install two flavors of Llama 3.1: the 70-billion-parameter version for difficult, context-heavy questions and a much smaller model for routine queries such as “What’s the Wi-Fi password?” or “Where is the library?” The large model is our workhorse because other organizations have already validated its stability, its longer “memory” lets it read several paragraphs at once without losing the thread and it answers in multiple languages, which is invaluable for international students. The smaller model steps in for easy tasks, saving both electricity and GPU hours.

## Model mix

|  |  |  |
| --- | --- | --- |
| Role | Model | Why we use it |
| Heavy Q&A engine | Llama 3.1 – 70 B | Handles long, multi-paragraph questions and works in several languages. |
| Light Q&A engine | Llama 3.1 – 8 B | Answers routine, single-sentence queries at roughly 70 % lower power cost. |
| Adding context for embedding | Llama 3.1 – 2 B | This model adds context to the data stored in the database, so HANnah can find the right answer to user questions. |
| Embedding model | Nomic-embed-text | This model will convert the data, so it can be put into the database. |

**Why use Llama?**

The use of the Llama llm is carefully selected as it’s the best open source llm in the market right now (Meta, 2025). That means that everyone is free to download Llama on their own hardware and use it locally, which means that the data going through it will not leave the organization. The same goes for the rest of the programs used in this mvp.

## Front-end options

* Telegram bot (live in pilot)
* WhatsApp Business (mobile-first students)
* Chat widget on han.nl (no extra app required)

## Back-end

* Conversation logs (anonymized) stored locally
* Dashboards show:
  + Top questions per week
  + Times Hannah said “I don’t know”
  + Power usage per 1 000 answers
* A place for employees to upload documents.

## HAN’s own private ChatGPT

HANnah can grow far beyond a Q&A bot. By adding a modern chat interface, styled much like ChatGPT or Microsoft Copilot students and staff could enjoy the full “AI-assistant” experience, but inside HAN’s secure walls.

**Private by design**

Access would be tied to a HAN login, so all prompts, answers and uploaded files stay on campus servers. Sensitive course material or draft research papers never leave the HAN environment.

**Same power, safer home**

Running the 70-billion-parameter Llama 3.1 model (roughly on par with GPT-4 in many benchmarks) means users can:

* generate clean code snippets,
* draft research summaries,
* troubleshoot complex SQL queries,

all with the context they supply.

**Personal workspaces**

Each authorized student or employee could upload their own set of documents: lecture notes, spreadsheets, policy drafts and let HANnah reason over them without exposing the data to external clouds.

In short, the university gains a ChatGPT-class assistant while keeping every byte of knowledge exactly where it belongs: inside HAN.

## Result

A multilingual, always-on assistant that answers questions in seconds, tidies outdated web pages and returns clear metrics on what students and staff really need.

# references

***Link to GitHub repo:*** <https://github.com/ums11/indproject>

[*https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf*](https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf)

[*https://www.anthropic.com/news/contextual-retrieval*](https://www.anthropic.com/news/contextual-retrieval)

[*https://www.youtube.com/watch?v=VAaFsqu5NE8*](https://www.youtube.com/watch?v=VAaFsqu5NE8)

***Programs used***

[*https://www.docker.com*](https://www.docker.com)

[*https://n8n.io/workflows/*](https://n8n.io/workflows/)

[*https://qdrant.tech/?utm\_source=google&utm\_medium=cpc&utm\_campaign=21518712216&utm\_content=163351119977&utm\_term=qdrant&hsa\_acc=6907203950&hsa\_cam=21518712216&hsa\_grp=163351119977&hsa\_ad=746944727642&hsa\_src=g&hsa\_tgt=kwd-1329481093586&hsa\_kw=qdrant&hsa\_mt=e&hsa\_net=adwords&hsa\_ver=3&gad\_source=1&gad\_campaignid=21518712216&gbraid=0AAAAAodw\_9CrekKDyXw5CtJJBBG9YB2Vw&gclid=CjwKCAjw6NrBBhB6EiwAvnT\_rhz0HhMGsbGg88prXxBLdEGZPJci73MkGyUDC61w3gmVfKXxROh\_0hoC3xkQAvD\_BwE*](https://qdrant.tech/?utm_source=google&utm_medium=cpc&utm_campaign=21518712216&utm_content=163351119977&utm_term=qdrant&hsa_acc=6907203950&hsa_cam=21518712216&hsa_grp=163351119977&hsa_ad=746944727642&hsa_src=g&hsa_tgt=kwd-1329481093586&hsa_kw=qdrant&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gad_source=1&gad_campaignid=21518712216&gbraid=0AAAAAodw_9CrekKDyXw5CtJJBBG9YB2Vw&gclid=CjwKCAjw6NrBBhB6EiwAvnT_rhz0HhMGsbGg88prXxBLdEGZPJci73MkGyUDC61w3gmVfKXxROh_0hoC3xkQAvD_BwE)

[*https://ollama.com*](https://ollama.com)

[*https://chatgpt.com*](https://chatgpt.com)

[*https://ai.meta.com/blog/meta-llama-3-1/*](https://ai.meta.com/blog/meta-llama-3-1/)

