

Les 15 — Huiswerk

Eigen PDF + UI + RAG-tool in agent + analyse

Vak: AI-Assisted Development
Deadline: Voor Les 16 — Multimodal AI
Inleveren: GitHub repo + RAG.md in root
Tijd: ~2 uur

Doel

Bouwt voort op de lesopdracht. Echte PDF van min 20 pagina's indexeren, chat-UI bouwen, RAG-tool in ToolLoopAgent integreren, analyse documenteren.

A

Echte PDF + UI (verplicht)

Kies PDF van min 20 pagina's — eigen interesse:

- Studieboek-hoofdstuk
- Open-source whitepaper (LLaMA, RAG-papers)
- Bedrijfsjaarverslag
- Wikipedia-artikel als PDF
- Game manual
- (geen scan-PDFs — moet text-based zijn)

UI:

- Upload-form met file input
- Chat-interface met useChat (zoals Les 12)
- Streaming antwoorden
- Tool-invocations zichtbaar (zoals Les 12 huiswerk)

Eisen:

- Upload werkt
- Geïndexeerde docs zichtbaar (count of lijst)
- Chat met streaming

-
- Antwoorden duidelijk op PDF gebaseerd

B RAG-tool in agent (verplicht)

```
// lib/agent.ts
const ragSearch = tool({
  description:
    "Zoek in geuploade documenten op basis van semantic similarity.",
  inputSchema: z.object({ query: z.string() }),
  execute: async ({ query }) => {
    const embedding = await embedOne(query);
    const { data } = await supabase.rpc("match_chunks", {
      query_embedding: embedding, match_count: 5,
    });
    return data;
  },
});

export const docAgent = new ToolLoopAgent({
  model: openai("gpt-4o-mini"),
  system: `Zoek met ragSearch, lees, eventueel 2e search, antwoord met sources.`,
  tools: { ragSearch },
  stopWhen: stepCountIs(10),
});
```

Eisen:

- ragSearch tool gedefinieerd + werkend
- ToolLoopAgent gebruikt deze tool
- In UI: tool-invocations zichtbaar
- Voor complexe vragen: 2-3 searches

C

RAG.md (verplicht)

Sectie 1 — Document info

- Bestandsnaam + onderwerp
- Aantal pagina's
- Aantal chunks
- Chunk-size + overlap

Sectie 2 — 5 succesvolle vragen

Voor elk: vraag, antwoord-samenvatting, top 3 chunks met similarity scores.

Sectie 3 — 1 fail-case

Vraag waar RAG het verkeerd had. Analyse waarom + wat zou helpen.

Sectie 4 — Chunking experiment

Default vs 1 andere strategie (andere size, overlap, of split-per-page). Documenteer met chunks-count + kwaliteit-oordeel.

Sectie 5 — Eén observatie

- Welke vragen werken het best?
- Hoe lang indexing van X pagina's?
- Effect van top-k (3 vs 5 vs 10)?
- Verschil simple RAG vs RAG-in-agent?

Vorm: max 800 woorden, concrete cijfers + voorbeelden.

D Bonus (optioneel)

- **Streaming + citations** — toon source-naam naast claims
- **Hybrid search** — semantic + Postgres full-text
- **Per-document filter** — UI dropdown, filter op source
- **Re-ranking** — top-20 -> Cohere/LLM -> top-5
- **Multi-document** — 3-5 PDFs, vraag spans meerdere

E Inleveren + Beoordeling

1. GitHub repo URL in Brightspace
2. RAG.md in repo-root (5 secties)
3. Updated lib/agent.ts met RAG-tool
4. Updated app/api/chat/route.ts met agent integration
5. Working app lokaal demonstreerbaar

Beoordeling (10 pt — voldoende 6+)

Criterium	Punten
A — UI + indexing + chat werkend	3
B — RAG-tool in ToolLoopAgent werkend	2
C — RAG.md compleet met 5 secties	3
Fail-case analyse concreet	1
Chunking-experiment + reflectie	1
Totaal	10

Veelvoorkomende valkuilen

- Scan-PDF — unpdf werkt niet op image-PDFs
- Vector dimension mismatch — index + query zelfde model
- Top-1 altijd irrelevant — chunks te klein of vraag te abstract
- Agent doet 1 search en stopt — system prompt expliciet
- RAG hallucineert — prompt strikter

Tips

- Test met simpele vragen eerst
- Log similarity scores — onder 0.4 meestal niet relevant
- Source-citations zijn goud
- Chunk-experiment is verplicht — verschil groter dan je denkt
- Eén goede fail-case is meer waard dan 10 succesvolle