

# Les 12 — Huiswerk

Write-tool + UI visualisatie + reflectie

|                   |                                     |
|-------------------|-------------------------------------|
| <b>Vak:</b>       | AI-Assisted Development             |
| <b>Opleiding:</b> | NOVI Hogeschool Utrecht             |
| <b>Deadline:</b>  | Voor volgende les (Les 13 — Agents) |
| <b>Inleveren:</b> | GitHub repo + TOOLS.md in root      |

## Doel

Bouwt voort op de **lesopdracht** (jouw thema-app met 3 read-tools). Voeg **write-tool** toe, visualiseer tool-calls in UI, schrijf `TOOLS.md`.

### Eerst lesopdracht afmaken

Niet klaar met lesopdracht? Eerst die — het huiswerk heeft die 3 read-tools als startpunt.

A

## Write-tool toevoegen (verplicht)

Voeg tool toe waarmee gebruiker iets kan opslaan.

## Stappen

1. Nieuwe tabel in Supabase (`user_favorites` / `user_notes` / `user_votes` / ...)
2. Write-tool schrijven (`addToFavorites`)
3. List-tool er bij (`listFavorites`)
4. Test in chat — 'voeg X toe' en 'wat staat in mijn lijst?'

## SQL-voorbeeld `user_favorites`

```
create table user_favorites (  
  id          bigserial primary key,  
  user_email  text not null,  
  item_id     bigint not null references items(id) on delete cascade,  
  created_at  timestamp default now(),  
  unique(user_email, item_id)  
);
```

---

## Eisen

- Nieuwe tabel + RLS policies (open voor demo)
- Write-tool werkt — favorieten worden opgeslagen
- List-tool werkt — favorieten worden teruggehaald
- System prompt: 'alleen toevoegen op expliciete request'

**B**

## Tool-calls in UI tonen (verplicht)

`useChat` retournt messages met **parts** — tekst-parts én tool-invocations.

```
{messages.map((m) => (
  <div key={m.id}>
    {m.parts?.map((part, i) => {
      if (part.type === "text") {
        return <div key={i}>{part.text}</div>;
      }
      // v6: tool-parts heten `tool-<toolName>`
      if (part.type?.startsWith("tool-")) {
        const toolName = part.type.replace("tool-", "");
        return (
          <div key={i} className="bg-yellow-50 p-2 rounded">
            ■ <strong>{toolName}</strong>({JSON.stringify(part.input)})
            {part.state === "output-available" && (
              <details>
                <summary>Toon resultaat</summary>
                <pre>{JSON.stringify(
                  part.output, null, 2
                )}</pre>
              </details>
            )}
          </div>
        );
      }
    })}
    </div>
  )}
```

### Eisen

- Tool-invocations zichtbaar als chip / badge / box
- Tool-naam + args getoond
- Resultaat collapsed zichtbaar
- Multi-step vragen tonen meerdere chips

**C**

## TOOLS.md (verplicht)

Schrijf `TOOLS.md` in repo-root.

### Sectie 1 — Mijn tools

Tabel met al je tools (tool, wat doet 't, read/write).

### Sectie 2 — Voorbeeld-vragen

- 3 vragen die 1 tool gebruiken

- 
- 1 vraag die 2+ tools combineert (multi-step)
  - Per vraag: tool die triggert + samenvatting antwoord

### **Sectie 3 — Edge case die AI goed afhandelde**

- Wat gebeurde er
- Welke tool
- Hoe handelde AI 't af

Max 500 woorden totaal. Concreet, geen vage reflectie.

## Bonus (optioneel)

- **Loading indicator** per tool-execute
- **Mooie cards** voor tool-results i.p.v. JSON-dump
- **Confirmation UI** voor write-tools
- **Tool failure recovery** — als tool faalt, AI probeert andere

## Inleveren

1. **GitHub repo URL** in Brightspace
2. **TOOLS.md** in repo-root
3. **Schema-update** in schema.sql (nieuwe tabel)
4. **Updated route.ts** met write-tool
5. **Updated chat/page.tsx** met tool-call rendering

## Beoordeling

| Criterium                         | Punten |
|-----------------------------------|--------|
| A — Write-tool + list-tool werken | 3      |
| B — Tool-calls in UI zichtbaar    | 2      |
| C — TOOLS.md met 3 secties        | 3      |
| Chat werkt end-to-end             | 1      |
| Documentatie sectie 3 is concreet | 1      |
| Totaal                            | 10     |

Voldoende = 6+. Bonus telt mee bij twijfelgevallen.

## Tijd-indicatie

| Onderdeel                            | Tijd     |
|--------------------------------------|----------|
| A — Write-tool (tabel + tool + test) | 40 min   |
| B — UI tool-call rendering           | 30 min   |
| C — TOOLS.md schrijven               | 20 min   |
| Totaal                               | ~1,5 uur |

### Veelvoorkomende valkuilen

- Write-tool zonder permissies — voor demo open, productie streng
- Confusion AI tussen read/write — sterke description
- UI part-types niet ondersteund — check AI SDK versie (v3+)
- `useChat` parts undefined — npm update ai

### Tips

- Schrijf TOOLS.md gaandeweg — niet aan eind
- Test write-tool eerst los voordat je via chat triggert
- UI parts is nieuw — check [ai-sdk.dev/docs](https://ai-sdk.dev/docs) voor actuele syntax
- Volgende les: Agents. Met tools werkend zien we 30+ tool-calls per request.