

Les 15 — Lesopdracht

Polderfest live via drie chats — GitHub + Polderfest + Vercel MCP

Vak: AI-Assisted Development **Opleiding:** NOVI Hogeschool Utrecht **Werkvorm:** in-class na de pauze (±1 uur basis + ±30 min extra) **Geen deliverables.** Gewoon doen tot het werkt.

Doel

Je doorloopt de hele flow die Tim in de demo deed. Vier MCPs koppelen, lokaal draaien, en daarna drie chats waarmee je via Cursor + GitHub MCP + Vercel MCP je Polderfest live krijgt — zonder zelf `git`, `gh` of `vercel` commando's te tikken.

Extra werk: breid je Polderfest MCP uit met een eigen feature via Cursor + Supabase MCP.

Wat je nodig hebt

- **Supabase project** met bands-tabel (uit Les 11/12)
- `SUPABASE_URL` + `SUPABASE_SERVICE_ROLE_KEY` (project Settings → API)
- **Supabase Personal Access Token** (`sbp_...`) + `project_ref`
- **GitHub Personal Access Token** (fine-grained, repo permissies)
- **Vercel account** (geen token vooraf — OAuth)
- **Cursor** ingelogd
- **Node** ≥ 18

Basis — ±1 uur

Stap 1 — Finished zip uitpakken + dev-server

```
cd ~/web
unzip ~/Downloads/polderfest-mcp-finished.zip
cd polderfest-mcp-finished

npm install
cp .env.example .env.local
```

Vul in `.env.local`:

```
SUPABASE_URL=https://xxxxx.supabase.co
SUPABASE_SERVICE_ROLE_KEY=eyJhbGc...
```

(Beide uit Supabase → Settings → API.)

```
npm run dev # → localhost:3000
```

Stap 2 — `~/.cursor/mcp.json` met vier MCPs

```
mkdir -p ~/.cursor
nano ~/.cursor/mcp.json
```

Plak (zie ook template in `lesbestanden/cursor-mcp-config/`):

```
{
  "mcpServers": {
    "supabase": {
      "type": "http",
      "url": "https://mcp.supabase.com/mcp?project_ref=<JOUW_REF>&read_only=false",
      "headers": { "Authorization": "Bearer sbp_..." }
    },
    "github": {
      "type": "http",
      "url": "https://api.githubcopilot.com/mcp/",
      "headers": { "Authorization": "Bearer ghp_..." }
    },
    "vercel": {
      "type": "http",
      "url": "https://mcp.vercel.com"
    },
    "polderfest": {
      "type": "http",
      "url": "http://localhost:3000/api/mcp"
    }
  }
}
```

Tip: zet `read\_only=false` voor Supabase als je in Chat 2 een tabel wilt aanmaken. Voor pure read-only demo's kan `true`.

Stap 3 — Cursor herstart + check

Cmd+Q, opnieuw openen.

In een chat onderaan de MCP-indicator: **4 servers groen** (github, supabase, vercel, polderfest).

Voor Vercel: klik "Needs login" → OAuth-flow in browser → autoriseer.

Stap 4 — Eerst commit lokaal

Een eerste commit klaarzetten voor Chat 1:

```
git init
git add .
git commit -m "polderfest mcp server"
git branch -M main
```

(Géén `gh repo create` — dat doet Chat 1 zo via MCP.)

Chat 1 — GitHub MCP: repo + remote

Open een **nieuwe** Cursor chat (Cmd+L). Plak:

```
Maak een nieuwe public GitHub repo "polderfest-mcp" aan via GitHub MCP.
Voeg toe als remote `origin` aan deze folder.
Push de main branch.
```

Wat je ziet gebeuren:

- GitHub MCP roept `create_repository` aan → URL terug
- Cursor draait `git remote add origin` in terminal
- Cursor draait `git push -u origin main`

■ **Check:** open `github.com` → repo staat er met je code.

Chat 2 — Polderfest + Supabase MCP: nieuwe feature

Open een **nieuwe** chat. Plak:

```
Ik wil een review-systeem voor de Polderfest MCP.

Stap 1: vraag Supabase MCP om een tabel band_reviews te maken
in mijn Polderfest project:
- id (uuid, pk, default uuid_generate_v4())
- band_id (uuid, FK naar bands.id)
- score (smallint, check tussen 1 en 10)
- comment (text)
- created_at (timestamp default now())

Stap 2: voeg twee tools toe aan @file:app/api/mcp/route.ts:
- addReview(bandName, score, comment) — zoekt band op naam, insert in band_reviews
- getReviews(bandName) — geeft reviews voor één band

Gebruik dezelfde patroon als bestaande tools (Zod, content-array).
```

Supabase MCP draait migration → bevestigen. Cursor genereert tools in `route.ts` → Accept all → save.

Commit + push:

```
git add . && git commit -m "feat: review tools"
git push
```

Test in dezelfde chat:

```
Geef Maud and The Cathedrals een review: 9, episch concert.
```

```
Toon alle reviews voor Maud and The Cathedrals.
```

■ **Check:** beide vragen geven antwoord, en in Supabase Table Editor staat een nieuwe rij in `band_reviews`.

Chat 3 — Vercel MCP: project + deploy

Open een **nieuwe** chat. Plak:

```
Maak via Vercel MCP een nieuw Vercel-project aan voor mijn
GitHub repo `polderfest-mcp`.

Voeg toe als env vars (production):
- SUPABASE_URL = <plak hier je URL>
- SUPABASE_SERVICE_ROLE_KEY = <plak hier je key>

Daarna: deploy naar productie en geef me de URL terug.
```

Vercel MCP doet:

- `create_project` met GitHub-koppeling
- env vars toevoegen
- deploy triggeren (±45s)
- productie-URL terug

Test in browser: open `https:///api/mcp` → MCP-handshake response.

Stap 5 — Cursor naar productie

Update `~/.cursor/mcp.json`: vervang de polderfest URL:

```
"polderfest": {
  "type": "http",
  "url": "https://polderfest-mcp-<scope>.vercel.app/api/mcp"
}
```

Cmd+Q Cursor, opnieuw openen.

Stop `npm run dev` lokaal. Stel opnieuw een van de test-vragen in chat — werkt nu zonder lokale dev-server. Alles op productie.

■ **Klaar als:** je vragen aan Polderfest werken met `npm run dev` gestopt.

Extra werk — Polderfest uitbreiden

Voor wie sneller is. Doe Chat 2 nog een keer, voor een eigen feature.

Stappen (zelfde patroon als de demo)

1. Bedenk een feature die nieuwe data nodig heeft
2. **Supabase MCP** via Cursor → maak nieuwe tabel + seed-data
3. **Cursor** → genereer MCP tool(s) in `route.ts`
4. Push naar GitHub — Vercel deploy't automatisch (je project is al gekoppeld via Chat 3)
5. Test in Cursor chat

Ideeën

Feature	Tabel	Tools
Reviews (demo)	band_reviews	addReview + getReviews
Wishlist	wishlist (user_email, band_id)	addToWishlist + getMyWishlist
Stage ratings	stage_ratings (stage, score)	rateStage + getStageRatings
Notes per band	band_notes (band_id, note)	addNote + getNotes
Eigen idee	—	—

Geen deliverables. Werkt het — top.

Vastgelopen?

Probleem	Oplossing
npm install faalt	Check Node ≥ 18, npm cache clean --force
Supabase 401 in app	Service-role key opnieuw kopiëren (let op witregels)
MCP-indicator komt niet	Cursor VOLLEDIG sluiten (Cmd+Q) en opnieuw openen
mcp.json syntax error	jsonlint.com — één komma fout = ALLE servers vallen uit
Vercel "Needs login" werkt niet	Browser pop-up geblokkeerd? Ga manueel naar mcp.vercel.com

Probleem	Oplossing
Supabase MCP weigert migration	read_only=false in mcp.json zetten
Productie-URL geeft 500	Env vars niet (goed) gezet in Vercel project
GitHub MCP geeft 403	PAT scope te beperkt — geef Contents + Issues + PR write-rechten
Tim loopt rond — vragen gewoon stellen.	