**06 - Plan voor Ontwikkelomgeving & Tooling  
Status:** IN PROGRESS (Lokale Dev Env OK, Tooling Gekozen)  
**Inhoudsopgave:**

1. Code Repository
2. Lokale Ontwikkelomgeving
3. Kerntechnologieën (Herhaling)
4. Code Editor & Extensies
5. Code Kwaliteit & Consistentie
6. Testing Frameworks
7. CI/CD
8. Project Management

**1. Code Repository**

* **Status: [✓ Ingericht & In Gebruik]**
* Versiebeheer: Git. [Status: ✓ Gebruikt]
* Hosting: GitHub (of GitLab/Bitbucket). [Status: ✓ Gekozen (GitHub aanname)]
* Repository Structuur: Monorepo beheerd met PNPM workspaces. [Status: ✓ Geïmplementeerd]

/nexa-platform/

├── apps/

│ ├── frontend/ # React app

│ └── supabase/ # Supabase project (migrations, functions, seed)

├── packages/ # (Nog niet in gebruik)

├── docs/ # Documentatie

├── pnpm-workspace.yaml

├── package.json

* Branching Strategie: Gitflow (of variant) aanbevolen. [Status: Concept OK, Implementatie TBD]
  + main: Productie.
  + develop: Integratie/Staging.
  + feature/xxx: Nieuwe features (van develop).
  + bugfix/xxx: Fixes op develop.
  + hotfix/xxx: Fixes op main.
* Pull Requests: Aanbevolen voor merges naar develop en main. [Status: Proces TODO]
* **2. Lokale Ontwikkelomgeving**
  + **Status: [✓ Operationeel]**
  + Vereiste Software:
    - Node.js (LTS versie) [Status: ✓ Geïnstalleerd]
    - PNPM [Status: ✓ Geïnstalleerd]
    - Git [Status: ✓ Geïnstalleerd]
    - Docker Desktop [Status: ✓ Geïnstalleerd]
    - Supabase CLI [Status: ✓ Geïnstalleerd]
  + Setup Stappen (per ontwikkelaar):
    - Clone de Git repository. [Status: ✓ Gedaan]
    - Installeer project dependencies (pnpm install in root). [Status: ✓ Gedaan]
    - Zet lokale Supabase stack op:
      * Login bij Supabase CLI (supabase login). [Status: ✓ Gedaan]
      * Start Supabase services (supabase start vanuit apps/supabase). [Status: ✓ Werkend]
      * Pas migraties toe en seed data (supabase db reset). [Status: ✓ Werkend]
    - Zet lokale Neo4j op:
      * Start Neo4j container via docker-compose up -d. [Status: ✓ Werkend]
      * Voer initiele Cypher CREATE CONSTRAINT/INDEX commando's uit in Neo4j Browser (eenmalig per verse DB). [Status: [ ] TODO]
    - Configureer environment variabelen: Maak .env bestanden aan op basis van .env.example. [Status: ✓ Basis OK (voor lokaal)]
    - Start Frontend dev server (pnpm --filter frontend dev). [Status: ✓ Werkend]
    - Lokale Supabase functions (Deno runtime) worden automatisch gestart via supabase start. [Status: ✓ Werkend (runtime, nog geen functies)]
* **3. Kerntechnologieën (Herhaling & Status)**
  + **Status: [✓ Keuzes Gemaakt, Status Toegevoegd]**
  + **Frontend:**
    - React [Status: ✓ Gebruikt]
    - TypeScript [Status: ✓ Gebruikt]
    - Tailwind CSS [Status: ✓ Gebruikt]
    - Zustand [Status: ✓ Geïnstalleerd, Basis Store OK]
    - React Router [Status: ✓ Geïnstalleerd, Basis Routes OK]
    - TanStack Query (useQuery) [Status: ✓ Geïnstalleerd, Gebruikt]
    - Cytoscape.js [Status: [ ] Nog niet geïmplementeerd]
  + **Backend (API):**
    - **Deno Runtime** [Status: ✓ Gekozen]
    - TypeScript [Status: ✓ Gekozen]
    - **Standaard Deno http/server** (of vergelijkbaar Deno-native) [Status: ✓ Gekozen]
    - Supabase JS Client [Status: ✓ Compatibel, Nog niet gebruikt in backend]
    - Neo4j JS Driver [Status: [ ] Nog niet getest/geïmplementeerd in backend]
    - JSforce (of alternatief) [Status: [ ] Nog niet getest/geïmplementeerd in backend]
  + **Database:**
    - PostgreSQL (via Supabase) [Status: ✓ Lokaal Actief & Schema OK]
    - Neo4j [Status: ✓ Lokaal Actief, Model OK, Setup TODO]
* **4. Code Editor & Extensies**
  + **Status: [✓ Gekozen & Deels Ingesteld]**
  + Editor: VS Code [Status: ✓ Gebruikt]
  + Essentiële Extensies:
    - ESLint [Status: ✓ Geïnstalleerd (Configuratie TODO)]
    - Prettier - Code formatter [Status: ✓ Geïnstalleerd (Configuratie TODO)]
    - Tailwind CSS IntelliSense [Status: ✓ Geïnstalleerd]
    - Docker [Status: ✓ Geïnstalleerd]
    - Supabase (Officiële VS Code extensie) [Status: ✓ Geïnstalleerd (Aanbevolen)]
    - GitLens — Git supercharged [Status: ✓ Geïnstalleerd (Aanbevolen)]
    - **Deno** (Officiële VS Code extensie) [Status: ✓ Geïnstalleerd (Aanbevolen voor Edge Functions)]
    - Import Cost [Status: Optioneel]
    - YAML [Status: Optioneel (Handig voor bijv. GitHub Actions)]
    - Markdown All in One [Status: Optioneel (Handig voor documentatie)]
* **5. Code Kwaliteit & Consistentie**
  + **Status: [Concepten Gekozen, Implementatie TODO]**
  + Linter: ESLint met aanbevolen configuraties voor TypeScript, React, Deno/Node.js (afhankelijk van context). Zorgt voor het afdwingen van codeerstijl en het vinden van potentiële bugs. [Status: ✓ Geïnstalleerd, Configuratie TODO]
  + Formatter: Prettier met een gedeelde configuratie (.prettierrc). Zorgt voor consistente code opmaak. [Status: ✓ Geïnstalleerd, Configuratie TODO]
  + Git Hooks: Gebruik van husky en lint-staged wordt aanbevolen om automatisch linting en formatting uit te voeren vóór elke commit, zodat slechte code de repository niet bereikt. [Status: [ ] Niet Geïmplementeerd]
* **6. Testing Frameworks**
  + **Status: [Keuzes Gedefinieerd, Implementatie TODO]**
  + Unit Tests: **Vitest** (aanbevolen vanwege snelheid en moderne API, compatibel met Vite voor frontend) of Jest. Voor het testen van individuele functies (React componenten, utility functies, Deno Edge Function helpers). React Testing Library voor frontend component tests. [Status: [ ] Niet Geïmplementeerd]
  + Integration Tests: Vitest/Jest kunnen ook gebruikt worden voor tests die de interactie tussen meerdere modules binnen de backend testen (bijv. API functie -> service -> DB call met een test database). [Status: [ ] Niet Geïmplementeerd]
  + End-to-End (E2E) Tests: **Playwright** (aanbevolen vanwege snelheid en betrouwbaarheid) of Cypress. Voor het testen van volledige user flows via de browser op een draaiende (staging) omgeving. [Status: [ ] Niet Geïmplementeerd]
* **7. CI/CD**
  + **Status: [✓ Tooling Gekozen, Implementatie TODO]**
  + Platform: **GitHub Actions** (aanbevolen). [Status: ✓ Gekozen]
  + Configuratie: Workflow bestanden (.github/workflows/\*.yaml) in de repository definiëren de build, test en deploy stappen voor verschillende branches/events. [Status: [ ] Niet Geïmplementeerd]
    - *Zie TAD Sectie 7.2 voor gedetailleerde pipeline stappen (moeten worden aangepast voor Deno).*
* **8. Project Management**
  + **Status: [✓ Tools Gekozen/In Gebruik]**
  + Issue Tracking: GitHub Issues (aanbevolen vanwege integratie) of externe tool (Jira, Linear, Trello). [Status: ✓ Keuze te maken/implementeren]
  + Communicatie: Slack of Microsoft Teams. [Status: ✓ Keuze gemaakt (impliciet)]
  + Documentatie: Google Drive / Gedeelde Map (voor deze documenten) en de docs/ map in de repository (voor technische documentatie). [Status: ✓ In gebruik]