NOVI Hogeschool — AI Leerlijn

Les 8

Supabase × Next.js + Auth

Database koppelen + authenticatie toevoegen

Les	8 van 18
Duur	3 uur (09:00 - 12:00)
Onderwerp	Supabase koppelen + Auth implementeren
Vereisten	Werkend QuickPoll + Supabase project

Inhoudsopgave

Deel 1: Supabase koppelen aan Next.js

Live Coding

Stap 1.1 — Supabase client library installeren

Installeer de Supabase JavaScript client library in je project:

```
npm install @supabase/supabase-js
```

Stap 1.2 — Environment variables instellen

Voeg je Supabase credentials toe in **.env.local**:

```
NEXT_PUBLIC_SUPABASE_URL=https://your-project.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-key-here
```

> *Vervang de URL en key met je eigen Supabase project credentials.*

Stap 1.3 — Supabase client aanmaken

Maak **lib/supabase.ts** aan:

```
import { createClient } from "@supabase/supabase-js";

const supabaseUrl = process.env.NEXT_PUBLIC_SUPABASE_URL!;
const supabaseAnonKey = process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!;

export const supabase = createClient(supabaseUrl, supabaseAnonKey);
```

Stap 1.4 — Types updaten

Update **types/index.ts** met Poll en Option interfaces:

```
export interface Poll {
    id: string;
    question: string;
    created_at: string;
    options: Option[];
}

export interface Option {
    id: string;
    poll_id: string;
    text: string;
    votes: number;
}
```

Stap 1.5 — Supabase queries in data.ts

Herschrijf **lib/data.ts** met Supabase queries:

```
import { supabase } from "./supabase";
import { Poll, Option } from "@/types";

export async function getPolls(): Promise<Poll[]> {
    const { data: polls, error } = await supabase
        .from("polls")
        .select("*, options(*)")
        .order("created_at", { ascending: false });

    if (error) {
        console.error("Error fetching polls:", error);
        return [];
    }

    return polls || [];
}
```

```
export async function getPollById(id: string): Promise<Poll | null> {
    const { data: poll, error } = await supabase
        .from("polls")
        .select("*, options(*)")
        .eq("id", id)
        .single();

    if (error) return null;
    return poll;
}
```

```
export async function votePoll(pollId: string, optionId: string): Promise<Option |
null> {
    const { data: option } = await supabase
        .from("options")
        .select("votes")
        .eq("id", optionId)
        .single();

    if (!option) return null;
    const { data: updated } = await supabase
        .from("options")
        .update({ votes: option.votes + 1 })
        .eq("id", optionId)
        .select().single();
    return updated || null;
}
```

Stap 1.6 — Homepage server component

Update **app/page.tsx** als server component:

```
import { getPolls } from "@/lib/data";
import { PollItem } from "@/components/PollItem";
import Link from "next/link";

export default async function Home() {
    const polls = await getPolls();
    return (
        <div className="w-full p-4">
            <h2 className="text-2xl font-bold mb-4">Onze polls</h2>
            {polls.map((poll) => (
                <Link key={poll.id} href={`/poll/${poll.id}`}>
                    <PollItem poll={poll} />
                </Link>
            ))}
        </div>
    );
}
```

Stap 1.7 — PollItem Component

Maak **components/PollItem.tsx** (client component) met rendering logic voor opties met percentage bars.

Stap 1.8 — VoteForm Component

Maak **components/VoteForm.tsx** (client component) dat het PollItem component weergeeft en stemmen afhandelt via een POST request naar /api/polls/[id].

Stap 1.9 — Poll Detail Pagina

Maak **app/poll/[id]/page.tsx** (server component) dat de poll ophaalt en de VoteForm component weergeeft.

Stap 1.10 — API Route

Maak **app/api/polls/[id]/route.ts** met GET en POST handlers. POST verwerkt stemmingen en retourneert de bijgewerkte poll.

Deel 2: Supabase Auth implementeren

Uitleg + Zelf Doen

Uitleg: Authentication concepten

In dit gedeelte gaan we authenticatie toevoegen aan QuickPoll. Dit betekent dat gebruikers zich moeten registreren en inloggen voordat ze kunnen stemmen. Supabase biedt een volledig auth-systeem dat cookies gebruikt om sessies te beheren.

Kernconcepten:
• **Server-side client:** Voor code die alleen op de server draait (Server Components, API routes)
• **Browser-side client:** Voor code die in de browser draait (Client Components)
• **Middleware:** Voor het automatisch vernieuwen van sessies

Stap 2.1 — @supabase/ssr installeren

```
npm install @supabase/ssr
```

Stap 2.2 — Server-side client

Maak **lib/supabase-server.ts** aan:

```
import { createServerClient } from "@supabase/ssr";
import { cookies } from "next/headers";

export async function createSupabaseServerClient() {
    const cookieStore = await cookies();
    return createServerClient(
        process.env.NEXT_PUBLIC_SUPABASE_URL!,
        process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
        { cookies: { getAll() { return cookieStore.getAll(); },
            setAll(cookiesToSet) {
                try { cookiesToSet.forEach(({ name, value, options }) =>
                    cookieStore.set(name, value, options)); } catch { }
            } } }
    );
}
```

Stap 2.3 — Browser-side client

Maak **lib/supabase-browser.ts** aan:

```
import { createBrowserClient } from "@supabase/ssr";

export function createSupabaseBrowserClient() {
    return createBrowserClient(
        process.env.NEXT_PUBLIC_SUPABASE_URL!,
        process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!
    );
}
```

Stap 2.4 — Middleware

Update **middleware.ts** voor automatische session vernieuwing met createServerClient. Voeg export const config = { matcher: [...] } toe.

Stap 2.5 — Auth callback route

Maak **app/auth/callback/route.ts** aan voor OAuth flow met exchangeCodeForSession().

Stap 2.6 — Signup pagina

Maak **app/signup/page.tsx** (client component) met form voor email/password, roept supabase.auth.signUp() aan.

Stap 2.7 — Login pagina

Maak **app/login/page.tsx** (client component) met form voor email/password, roept supabase.auth.signInWithPassword() aan.

Stap 2.8 — LogoutButton + Navbar

Maak **components/LogoutButton.tsx** (client) en **components/Navbar.tsx** (server). Navbar checkt user status en toont login/signup links of email + logout button.

Stap 2.9 — Layout updaten

Update **app/layout.tsx** om de Navbar component toe te voegen boven de children.

Huiswerk: Create pagina

Bouw een **/create** pagina waarmee authenticated users nieuwe polls kunnen aanmaken.
Vereisten:

• RLS INSERT policy in Supabase voor authenticated users
• Server Component pagina met form
• Form accepteert vraag + minimaal 2 opties
• INSERT in polls + options tabellen
• Redirect naar homepage na succesvolle creatie

Tips:
• Gebruik server-side Supabase client voor database operaties
• Check user status in server component (alleen ingelogde users)
• Valide input voordat je naar Supabase stuurt
• Zorg voor atomaire INSERT van poll + options