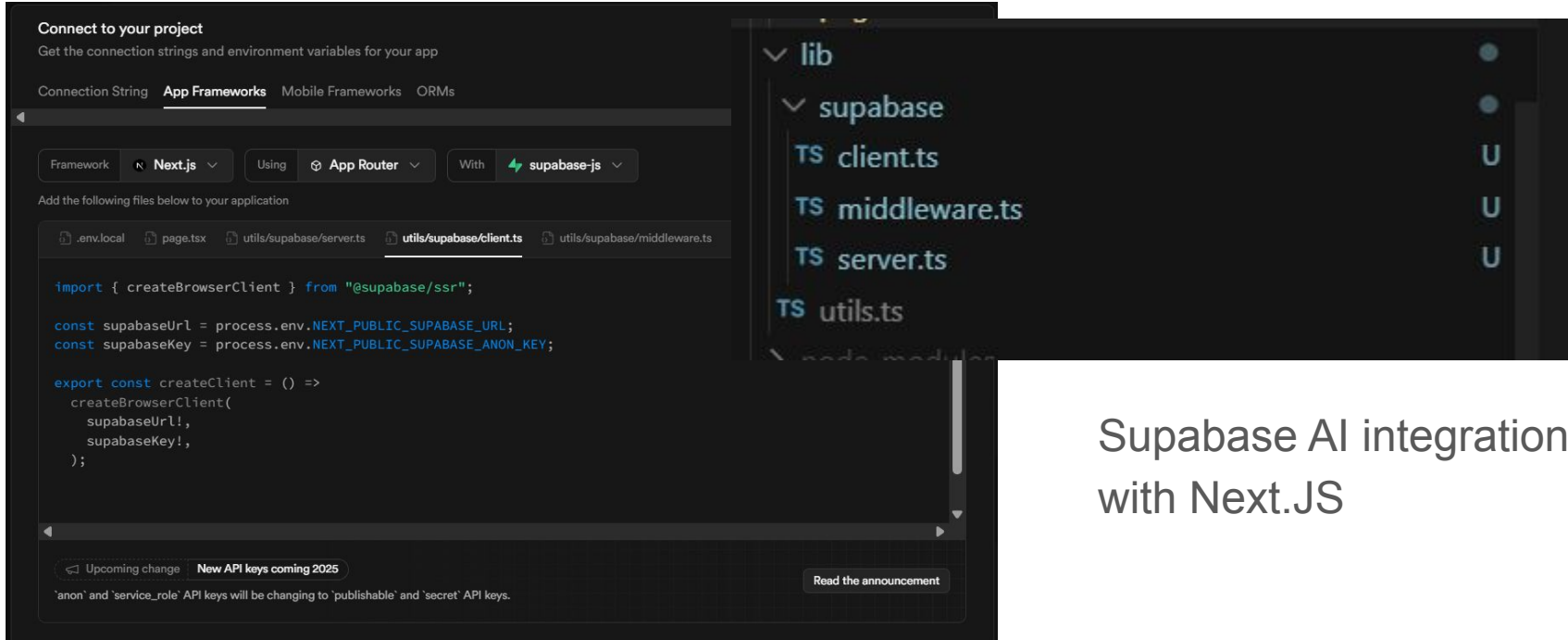


**ALX\_AIDEV**

#doinghardthings

# Building Features with AI - Authentication Feature



The screenshot displays the Supabase AI integration interface. At the top, it says "Connect to your project" and "Get the connection strings and environment variables for your app". Below this, there are tabs for "Connection String", "App Frameworks", "Mobile Frameworks", and "ORMs". The "App Frameworks" tab is selected, showing a configuration for "Next.js" using "App Router" with "supabase-js".

Below the configuration, it says "Add the following files below to your application" and lists several files: ".env.local", "page.tsx", "utils/supabase/server.ts", "utils/supabase/client.ts" (which is selected), and "utils/supabase/middleware.ts".

The selected file, "utils/supabase/client.ts", contains the following code:

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

const supabaseUrl = process.env.NEXT_PUBLIC_SUPABASE_URL;
const supabaseKey = process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY;

export const createClient = () => {
  createBrowserClient(
    supabaseUrl!,
    supabaseKey!,
  );
};
```

On the right side, there is a file explorer showing the project structure:

- lib
  - supabase
    - TS client.ts
    - TS middleware.ts
    - TS server.ts
    - TS utils.ts

At the bottom, there is a notification about "Upcoming change: New API keys coming 2025" and a link to "Read the announcement".

Supabase AI integration  
with Next.JS

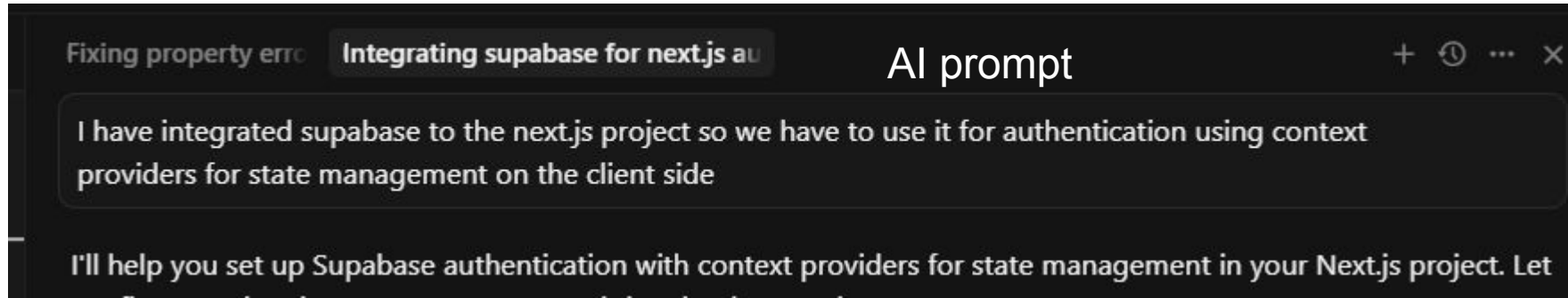
# Building Features with AI - Authentication Feature

✓ So, to summarize:

AI prompt response on use of  
`@supabase/ssr` and state management

- `@supabase/ssr` handles session persistence + sharing (cookies).
- You handle UI state management (context, hooks, etc.) on top of it.

# Building Features with AI - Authentication Feature



# Building Features with AI - Authentication Feature

The screenshot displays a VS Code editor with a TypeScript file named `index.ts` containing the following code:

```
1 import { User as SupabaseUser } from '@supabase/supabase-js';
2
3 // User types
4 export interface User {
5   id: string
6   name: string
7   email: string
8   createdAt: Date
9   updatedAt: Date
10 }
11
12 // Supabase User type for authentication
13 export type AuthUser = SupabaseUser;
14
15 // Authentication context types
16 export interface AuthContextType {
17   user: AuthUser | null
18   loading: boolean
19   signIn: (email: string, password: string) => Promise<{ error: any }>
20   signUp: (email: string, password: string, name: string) => Promise<{ error: any }>
21   signOut: () => Promise<void>
22   resetPassword: (email: string) => Promise<{ error: any }>
23 }
24
25 // Poll types
26 export interface PollType {
27   id: string
28   text: string
29 }
```

On the right side of the editor, a list of AI prompts is shown, each followed by a status bar indicating the number of files changed:

- Fixing property error: Integrating supabase for next.js au**  
I have integrated supabase to the next.js project so we have to use it for authentication using context providers for state management on the client side
- Read server.ts**  
Read index.ts
- Now I'll create the authentication context provider and update the types to work with Supabase. Let me start by creating the authentication context:**  
TS index.ts +16
- Now let me create the authentication context provider:**  
auth-context.tsx (new) +115
- Now let me update the root layout to include the AuthProvider:**  
Read layout.tsx  
layout.tsx +5 -2
- Now let me update the login form to use the authentication context:**  
Read login-form.tsx  
login-form.tsx +35 -5
- Now let me update the register form as well:**  
Read register-form.tsx  
register-form.tsx +53 -6
- Now let me update the navigation component to show authentication links:**  
> 10 Files  
Undo All Keep All

A "Focus period..." timer is visible in the bottom right corner, showing 14 minutes.

# Building Features with AI - Authentication Feature

The image shows a VS Code editor with two panes. The left pane displays the code for `auth-context.tsx`, and the right pane shows the AI chat interface.

**Left Pane (auth-context.tsx):**

```
1 'use client';
2
3 import React, { createContext, useContext, useEffect, useState } from 'react'
4 import { createClient } from '@lib/supabase/client';
5 import { AuthContextType, AuthUser } from '@types';
6 import { User } from '@supabase/supabase-js';
7
8 const AuthContext = createContext<AuthContextType | undefined>(undefined);
9
10 export const useAuth = () => {
11   const context = useContext(AuthContext);
12   if (context === undefined) {
13     throw new Error('useAuth must be used within an AuthProvider');
14   }
15   return context;
16 };
17
18 interface AuthProviderProps {
19   children: React.ReactNode;
20 }
21
22 export const AuthProvider: React.FC<AuthProviderProps> = ({ children }) => {
23   const [user, setUser] = useState<AuthUser | null>(null);
24   const [loading, setLoading] = useState(true);
25   const supabase = createClient();
26
27   useEffect(() => {
28     // Get initial session
29   }, []);
30 }
```

**Right Pane (AI Chat):**

**Fixing property error: Integrating supabase for next.js au**

I have integrated supabase to the next.js project so we have to use it for authentication using context providers for state management on the client side

Read server.ts

Read index.ts

Now I'll create the authentication context provider and update the types to work with Supabase. Let me start by creating the authentication context.

TS index.ts +16 ✓

Now let me create the authentication context provider:

auth-context.tsx (new) +10 ✓

Now let me update the root layout to include the AuthProvider:

Read layout.tsx

layout.tsx +5 -2 •

Now let me update the login form to use the authentication context:

Read login-form.tsx

login-form.tsx +35 -5 •

Now let me update the register form as well:

Read register-form.tsx

register-form.tsx +53 -6 •

Now let me update the navigation component to show authentication status and sign out functionality:

> 10 Files

Undo All Keep All ^

# Building Features with AI - Authentication Feature

The screenshot displays a code editor with the following content:

**Left Pane (layout.tsx):**

```
1 import type { Metadata } from "next";
2 import { Geist, Geist_Mono } from "next/font/google";
3 import "./globals.css";
4 import { Navigation } from "@app/components/layout/navigation";
5 import { AuthProvider } from "@app/contexts/auth-context";
6
7 const geistSans = Geist({
8   variable: "--font-geist-sans",
9   subsets: ["latin"],
10 });
11
12 const geistMono = Geist_Mono({
13   variable: "--font-geist-mono",
14   subsets: ["latin"],
15 });
16
17 export const metadata: Metadata = {
18   title: "ALX Polly - Create and Vote on Polls",
19   description: "A modern polling application built with Next.js",
20 };
21
22 export default function RootLayout({
23   children,
24 }: Readonly<{
25   children: React.ReactNode;
26 }>) {
27   return (
28     <html lang="en">
```

**Right Pane (AI response - step III):**

**Integrating supabase for next.js au**

I have integrated supabase to the next.js project so we have to use it for authentication using context providers for state management on the client side

Read server.ts

Read index.ts

Now I'll create the authentication context provider and update the layout to work with Supabase. Let me start by creating the authentication context:

TS index.ts +16 ✓

Now let me create the authentication context provider:

auth-context.tsx +115 ✓

Now let me update the root layout to include the AuthProvider:

Read layout.tsx

layout.tsx +5 -2 ✓

Now let me update the login form to use the authentication context:

Read login-form.tsx

login-form.tsx +35 -5 ✓

Now let me update the register form as well:

Read register-form.tsx

register-form.tsx +53 -6 ✓

Now let me update the navigation component to show authentication status and sign out functionality:

# Building Features with AI - Authentication Feature

AI response - step IV  
updates auth forms for  
registration and signing  
in

```
TS index.ts M layout.tsx M register-form.tsx M login-form.tsx M X
alk-polly>app>components\auth\login-form.tsx Modified
1 "use client"
2
3 import { useState } from "react"
4 import { Button } from "@app/components/ui/button"
5 import { useAuth } from "@app/contexts/auth-context"
6 import { useRouter } from "next/navigation"
7 import { getAuthErrorMessage } from "@lib/auth-utils"
8
9 export function LoginForm() {
10   const [email, setEmail] = useState("")
11   const [password, setPassword] = useState("")
12   const [isLoading, setIsLoading] = useState(false)
13   const [error, setError] = useState("")
14   const { signIn } = useAuth()
15   const router = useRouter()
16
17   const handleSubmit = (e: React.FormEvent) => {
18     e.preventDefault()
19     // TODO: Implement login logic
20     console.log("Login attempt:", { email, password })
21     setIsLoading(true)
22     setError("")
23
24     try {
25       const { error: signInError } = await signIn(email, password)
26       if (signInError) {
27         setError(getAuthErrorMessage(signInError))
28       }
29     } catch {
30       // Handle network errors
31     }
32   }
33 }

Files M register-form.tsx M login-form.tsx M protected-route.tsx U
alk-polly > app > components C:\Users\ogech\Documents\ALX_AIDEV\alk-polly\app\components\auth\register-form.tsx Modified
1 use client
2
3 import { useState } from "react"
4 import { Button } from "@app/components/ui/button"
5 import { useAuth } from "@app/contexts/auth-context"
6 import { useRouter } from "next/navigation"
7 import { getAuthErrorMessage } from "@lib/auth-utils"
8
9 export function RegisterForm() {
10   const [formData, setFormData] = useState({
11     name: "",
12     email: "",
13     password: "",
14     confirmPassword: ""
15   })
16   const [isLoading, setIsLoading] = useState(false)
17   const [error, setError] = useState("")
18   const { signUp } = useAuth()
19   const router = useRouter()
20
21   const handleSubmit = (e: React.FormEvent) => {
22     e.preventDefault()
23     // TODO: Implement registration logic
24     console.log("Registration attempt:", formData)
25
26     if (formData.password !== formData.confirmPassword) {
27       setError("Passwords do not match")
28     }
29   }
30 }
```



# Building Features with AI - Authentication Feature

The image shows a code editor with a file explorer on the left and a chat window on the right. The file explorer shows a project structure with files like `register-form.tsx`, `login-form.tsx`, and `protected-route.tsx`. The chat window displays a series of prompts and AI responses, detailing the integration of Supabase for authentication.

**AI response - step V**  
created a file for  
handling protected  
routes

**Code Editor Content:**

```
1 | Modified
2 |
3 | import { useEffect } from 'react';
4 | import { useRouter } from 'next/navigation';
5 | import { useAuth } from '@app/contexts/auth-context';
6 |
7 | interface ProtectedRouteProps {
8 |   children: React.ReactNode;
9 |   redirectTo?: string;
10 | }
11 |
12 | export const ProtectedRoute: React.FC<ProtectedRouteProps> = ({
13 |   children,
14 |   redirectTo = '/auth/login'
15 | }) => {
16 |   const { user, loading } = useAuth();
17 |   const router = useRouter();
18 |
19 |   useEffect(() => {
20 |     if (!loading && !user) {
21 |       router.push(redirectTo);
22 |     }
23 |   }, [user, loading, router, redirectTo]);
24 |
25 |   if (loading) {
26 |     return (
27 |       <div className="flex min-h-screen flex-col items-center justify-center">
28 |         <div className="text-center">
```

**Chat Window Content:**

Fixing property error Integrating supabase for next.js au

I have integrated supabase to the next.js project so we have to use it for authentication using context providers for state management on the client side

Read login-form.tsx

login-form.tsx +35 -5

Now let me update the register form as well:

Read register-form.tsx

register-form.tsx +53 -6

Now let me update the navigation component to show authentication status and sign out functionality:

Read navigation.tsx

navigation.tsx +38 -12 ✓

Now let me create a protected route component to handle authentication guards:

protected-route.tsx (new) +41

Now let me update the create poll page to use the protected route:

Read page.tsx

page.tsx +9 -5 ✓

Now let me create a hook for managing authentication state that can be used throughout the app:

use-auth-state.ts (new) +21

Now let me create a utility function for handling authentication errors:

# Building Features with AI - Authentication Feature

The image shows a code editor with a file explorer on the left and a chat interface on the right. The code editor displays the following TypeScript code in `use-auth-state.ts`:

```
1 import { useAuth } from '@app/contexts/auth-context';
2
3 export const useAuthState = () => {
4   const { user, loading, signIn, signUp, signOut, resetPassword } = useAuth()
5
6   const isAuthenticated = !!user;
7   const isEmailVerified = user?.email_confirmed_at ? true : false;
8   const userDisplayName = user?.user_metadata?.name || user?.email || 'User';
9
10  return {
11    user,
12    loading,
13    isAuthenticated,
14    isEmailVerified,
15    userDisplayName,
16    signIn,
17    signUp,
18    signOut,
19    resetPassword,
20  };
21 };
22
```

The chat interface on the right contains the following text and a list of prompts:

Fixing property error Integrating supabase for next.js au

I have integrated supabase to the next.js project so we have to use it for authentication using context providers for state management on the client side

Now let me update the create page to use the protected route.

Read page.tsx

page.tsx +9 -5 ✓

Now let me create a hook for managing authentication state that can be used throughout the app:

TS use-auth-state.ts (new) +21 •

Now let me create a utility function for handling authentication errors:

TS auth-utils.ts (new) +34 •

Now let me update the login form to use the better error handling:

login-form.tsx +26 -3 ✓

I need to fix the login form properly. Let me reapply the complete update:

login-form.tsx +10 -2 ✓

Now let me update the register form to use the better error handling as well:

register-form.tsx +38 -4 ✓

I need to fix the register form properly as well. Let me reapply the complete update:

register-form.tsx +10 ✓

**AI response - step VI created an auth hook**

# Building Features with AI - Authentication Feature

## Testing App Auth feature

P

ALX Polly

Browse Polls

Create an account

Join our polling community

Passwords do not match

Full Name

Ogechukwu Kanu

Email

ogechukwukanu@gmail.com

Password

\*\*\*\*\*

Confirm Password

\*\*\*\*\*

Create Account

Already have an account? [Sign In](#)

P

ALX Polly

Browse Polls

Create Poll

Welcome, Ogechukwu Kanu

✓

Account Created Successfully!

Welcome to ALX Polly! We're excited to have you join our polling community.

✉

Check Your Email

We've sent you a confirmation email. Please check your inbox and click the verification link to activate your account.

Important: You won't be able to sign in until you verify your email address.

Already verified your email?

☰

Gmail

Search mail

Compose

Inbox

24,871

Starred

Snoozed

Sent

Drafts

1

More

Labels

+

Confirm Your Signup

Supabase Auth <noreply@mail.app.supabase.io> to me

Confirm your signup

Follow this link to confirm your user:

[Confirm your mail](#)

You're receiving this email because you signed up for an app

Get out of these emails

P

ALX Polly

Browse Polls

Create Poll

Welcome, Ogechukwu Kanu

✓

Email Verified Successfully!

Your email has been verified. You can now sign in to your account and start creating polls!

Sign In to Your Account

Or go back to home page