**Podsumowanie: Przygotowanie strony do logowania (Next.js + Flask + MongoDB)**

**🔢 1. Backend Flask + MongoDB**

**✅ Plik .env (dla backendu)**

Dodaliśmy konfigurację bazy danych i maila:

MONGO\_URI=mongodb://localhost:27017/trading\_db

SECRET\_KEY=... # klucz JWT

TOKEN\_EXPIRATION=3600

MAIL\_SERVER=smtp.gmail.com

MAIL\_PORT=587

MAIL\_USERNAME=twoj@email.pl

MAIL\_PASSWORD=haslo\_aplikacji

MAIL\_DEFAULT\_SENDER=twoj@email.pl

**📅 Plik app.py**

Dodaliśmy:

* import i konfigurację Flask-Mail
* funkcję send\_welcome\_email(email)
* wywołanie tej funkcji w endpointzie rejestracji:

from flask\_mail import Mail, Message

mail = Mail(app)

def send\_welcome\_email(email):

msg = Message(

subject="Witaj!",

recipients=[email],

body="Dziękujemy za rejestrację.",

charset="utf-8"

)

mail.send(msg)

# w funkcji register:

send\_welcome\_email(data["email"])

**🚫 Ręczna aktywacja (tymczasowo)**

W MongoDB (np. Compass lub mongosh):

db.users.updateOne(

{ \_id: ObjectId("...") },

{ $set: { is\_active: true } }

)

**🚀 2. Frontend Next.js (strona logowania)**

**✅ Plik LoginForm.tsx (w features/auth/components/)**

Dodaliśmy:

* react-hook-form i zod do walidacji:

const loginSchema = z.object({

email: z.string().email("Nieprawidłowy email"),

password: z.string().min(1, "Hasło nie zostało wpisane")

});

* funkcję onSubmit z pobieraniem tokena z ciasteczka:

const match = document.cookie.match(/access\_token=([^;]+)/);

const token = match?.[1];

if (token) {

localStorage.setItem("token", token);

router.push("/dashboard");

}

* obsługę błędu:

catch (error: any) {

if (error?.data?.message) alert("Błąd: " + error.data.message);

else alert("Nie udało się zalogować.");

}

**🔒 3. Ochrona strony /dashboard**

**✅ Plik app/dashboard/page.tsx**

Dodaliśmy przekierowanie, jeśli brak tokena:

"use client";

import { useEffect } from "react";

import { useRouter } from "next/navigation";

export default function DashboardPage() {

const router = useRouter();

useEffect(() => {

const token = localStorage.getItem("token");

if (!token) router.push("/login");

}, []);

return <div>Witaj na dashboardzie!</div>;

}

**📃 4. Testowanie i uruchamianie**

**Backend:**

flask --app app --debug run

**Frontend:**

npm run dev

**Testowanie:**

1. Przejdź do http://localhost:3000/login
2. Wprowadź dane i kliknij "Zaloguj się"
3. Sprawdź:
   * czy token pojawił się w ciasteczkach
   * czy token jest w localStorage
   * czy nastąpiło przekierowanie

✅ Gotowe! Strona logowania i backend są w pełni skonfigurowane i działają razem. Możesz kontynuować od:

* dodania wylogowania
* tworzenia linku aktywacyjnego
* pobierania danych użytkownika przez /me
* refresh tokena