"use client";

import React, { useState } from "react";

import { Input } from "@/components/ui/input";

import { Button } from "@/components/ui/button";

import Image from "next/image";

import { LoaderIcon } from "lucide-react";

import GlobalApi from "@/app/\_utils/GlobalApi";

import { Label } from "@radix-ui/react-label";

import toast, { Toaster } from "react-hot-toast";

import ReCAPTCHA from "react-google-recaptcha";

import { useRouter } from "next/navigation";

import { useAuth } from "@/app/AuthContext";

// Helper function to handle errors and show appropriate toast

const handleError = (error) => {

const errorMessage =

error?.response?.data?.message || "Terjadi kesalahan, coba lagi.";

toast.error(errorMessage);

};

// Validation for login form

const validateLoginForm = (npaPgri, tanggalLahir, isVerified) => {

if (npaPgri.length < 6 || tanggalLahir.length !== 8) {

throw new Error("NPA PGRI harus 6 digit dan Tanggal Lahir harus 8 digit.");

}

if (!isVerified) {

throw new Error("Harap verifikasi reCAPTCHA.");

}

};

// Validation for admin login form

const validateAdminForm = (npaPgri, password, isVerified) => {

if (!isVerified) {

throw new Error("Harap verifikasi reCAPTCHA.");

}

if (!npaPgri || !password) {

throw new Error("NPA PGRI dan Password tidak boleh kosong.");

}

};

function SignIn() {

const [npaPgri, setNpaPgri] = useState("");

const [tanggalLahir, setTanggalLahir] = useState("");

const [password, setPassword] = useState("");

const [loader, setLoader] = useState(false);

const [error, setError] = useState("");

const [isVerified, setIsVerified] = useState(false);

const [activeTab, setActiveTab] = useState("login");

const router = useRouter();

const { setToken, setUserId } = useAuth();

// Function to handle user login

const onSignIn = async () => {

setLoader(true);

setError("");

try {

validateLoginForm(npaPgri, tanggalLahir, isVerified);

const loginData = { npaPgri, tanggalLahir };

const response = await GlobalApi.login(loginData);

// Store token securely in HttpOnly cookie

document.cookie = `token=${response.token}; HttpOnly; Secure; SameSite=Strict`;

// Store session data securely

sessionStorage.setItem("userId", response.id);

sessionStorage.setItem("unitKerja", response.unitKerja);

sessionStorage.setItem("nama", response.namaLengkap);

sessionStorage.setItem("role", response.role);

toast.success(`Selamat Datang ${response.namaLengkap}`);

setTimeout(() => {

router.push("/home");

}, 2000);

} catch (error) {

handleError(error);

} finally {

setLoader(false);

}

};

// Function to handle admin login

const onSignInAdmin = async () => {

setLoader(true);

setError("");

try {

validateAdminForm(npaPgri, password, isVerified);

const response = await GlobalApi.loginAdmin(npaPgri, password);

// Store token securely in HttpOnly cookie

document.cookie = `token=${response.token}; HttpOnly; Secure; SameSite=Strict`;

// Store session data securely

sessionStorage.setItem("userId", response.id);

sessionStorage.setItem("nama", response.namaLengkap);

sessionStorage.setItem("role", response.role);

toast.success(`Selamat Datang ${response.namaLengkap}`);

setTimeout(() => {

router.push("/home");

}, 2000);

} catch (error) {

handleError(error);

} finally {

setLoader(false);

}

};

// reCAPTCHA change handler

const onChange = (value) => {

setIsVerified(!!value);

};

return (

<div className="flex items-baseline justify-center my-9">

<Toaster

toastOptions={{

style: {

fontSize: "1.25rem",

padding: "16px",

},

success: {

style: {

background: "white",

color: "black",

},

},

error: {

style: {

background: "#f44336",

color: "#fff",

},

},

}}

/>

<div className="flex flex-col items-center justify-center p-4 sm:p-6 md:p-10 bg-gray-100 border border-gray-200 rounded-lg shadow-md w-full sm:w-[80%] md:w-[60%] lg:w-[40%] mx-auto">

<Image src="/sanduka.png" width={150} height={150} alt="logo" />

<h2 className="font-bold text-2xl sm:text-3xl mt-4">Masuk ke Akun</h2>

<h2 className="text-gray-500 mt-2 text-center">

Masukkan NPA PGRI dan{" "}

{activeTab === "login" ? "Tanggal Lahir" : "Password"} Anda untuk

Masuk

</h2>

{/\* Tab Navigation \*/}

<div className="flex space-x-2 sm:space-x-4 mt-4">

<button

className={`rounded-md p-2 ${

activeTab === "login"

? "bg-teal-500 text-white"

: "bg-white text-teal-500"

}`}

onClick={() => setActiveTab("login")}

>

Anggota

</button>

<button

className={`rounded-md p-2 ${

activeTab === "password"

? "bg-teal-500 text-white"

: "bg-white text-teal-500"

}`}

onClick={() => setActiveTab("password")}

>

Admin & Super Admin

</button>

</div>

<div className="w-full max-w-md mt-6">

{error && <p className="text-red-500 mb-4 text-center">{error}</p>}

{/\* Form for Login or Admin \*/}

{activeTab === "login" && (

<div>

<div className="mb-6">

<Label htmlFor="npaPgri" className="block text-sm">

NPA PGRI

</Label>

<Input

id="npaPgri"

placeholder="123456"

value={npaPgri}

onChange={(e) => setNpaPgri(e.target.value)}

className="mt-2 p-3 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500 w-full"

/>

</div>

<div className="mb-6">

<Label htmlFor="tanggalLahir" className="block text-sm">

Tanggal Lahir

</Label>

<Input

id="tanggalLahir"

placeholder="21082024"

value={tanggalLahir}

onChange={(e) => setTanggalLahir(e.target.value)}

className="mt-2 p-3 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500 w-full"

/>

</div>

</div>

)}

{activeTab === "password" && (

<div>

<div className="mb-6">

<Label htmlFor="npaPgri" className="block text-sm">

NPA PGRI

</Label>

<Input

id="npaPgri"

placeholder="123456"

value={npaPgri}

onChange={(e) => setNpaPgri(e.target.value)}

className="mt-2 p-3 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500 w-full"

/>

</div>

<div className="mb-6">

<Label htmlFor="password" className="block text-sm">

Password

</Label>

<Input

id="password"

placeholder="\*\*\*\*\*\*\*\*"

value={password}

onChange={(e) => setPassword(e.target.value)}

type="password"

className="mt-2 p-3 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500 w