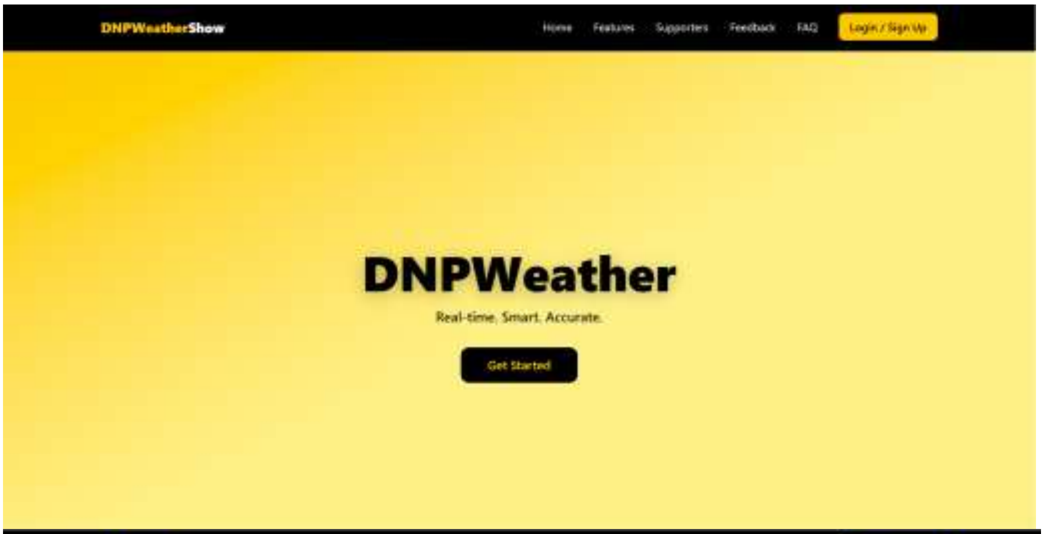
Task2



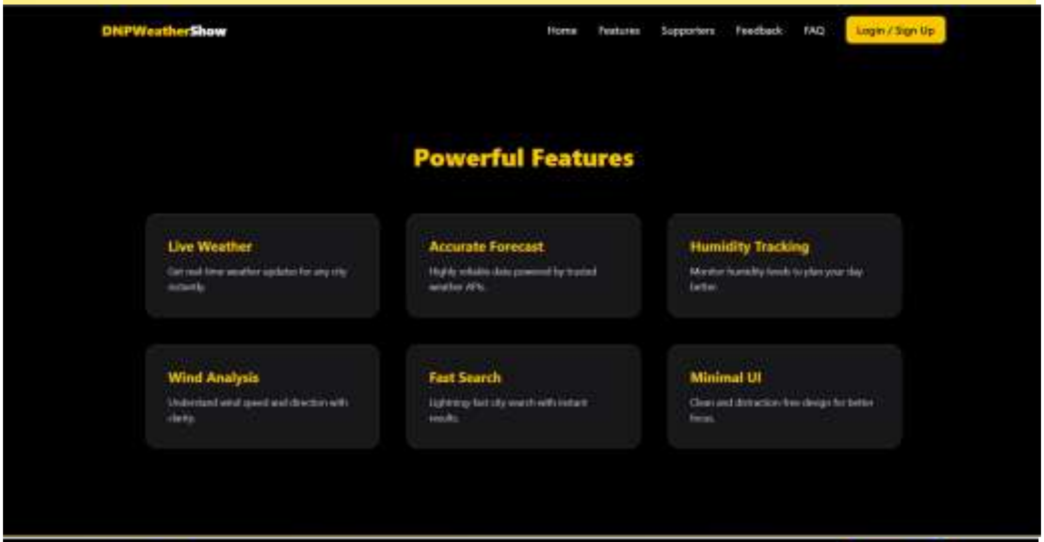
DNPWeatherShow

Home Features Supporters Feedback FAQ Login / Sign Up

DNPWeather

Real-time. Smart. Accurate.

Get Started



DNPWeatherShow

Home Features Supporters Feedback FAQ Login / Sign Up

Powerful Features

Live Weather
Get real-time weather updates for any city instantly.

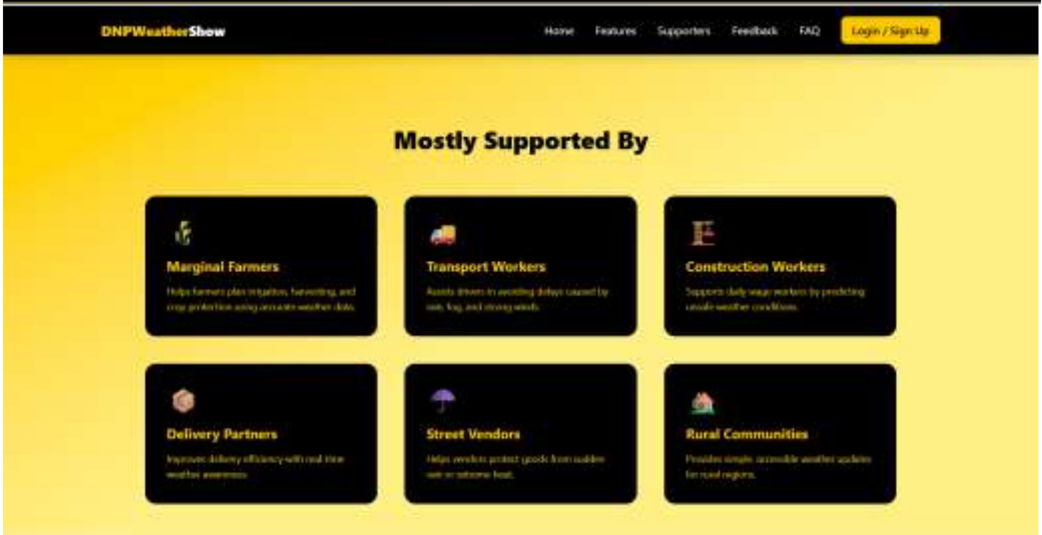
Accurate Forecast
Highly reliable data powered by trusted weather APIs.

Humidity Tracking
Monitor humidity levels to plan your day better.

Wind Analysis
Understand wind speed and direction with clarity.

Fast Search
Lightning-fast city search with instant results.

Minimal UI
Clean and distraction-free design for better focus.



DNPWeatherShow

Home Features Supporters Feedback FAQ Login / Sign Up

Mostly Supported By

Marginal Farmers
Helps farmers plan irrigation, harvesting, and crop protection using accurate weather data.

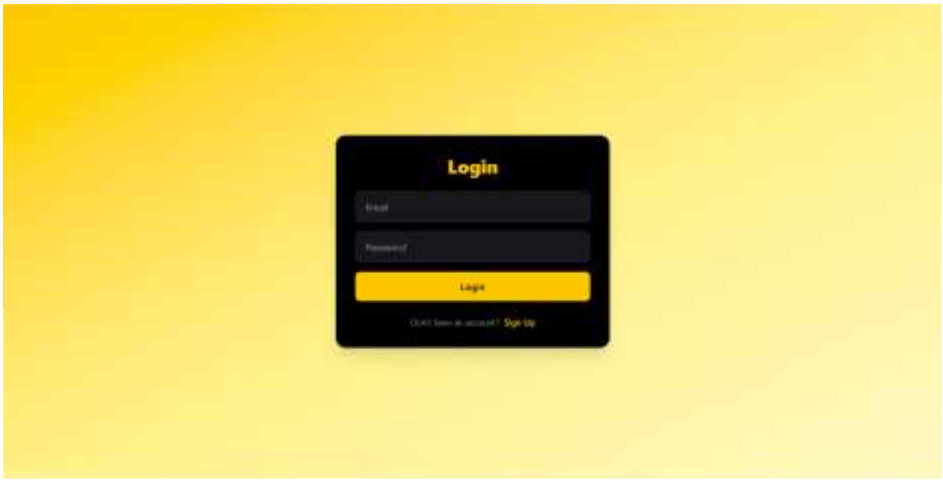
Transport Workers
Assists drivers in avoiding delays caused by rain, fog, and strong winds.

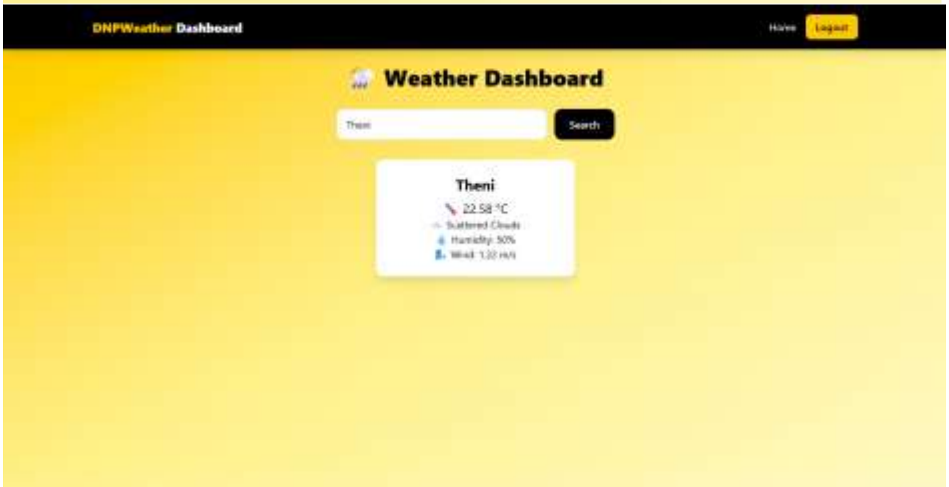
Construction Workers
Supports daily wage workers by predicting unsafe weather conditions.

Delivery Partners
Improves delivery efficiency with real-time weather awareness.

Street Vendors
Helps vendors protect goods from sudden rain or extreme heat.

Rural Communities
Provides simple, accessible weather updates for rural regions.





Theni
🌡️ 22.58 °C
☁️ Scattered Clouds
💧 Humidity: 50%
🌬️ Wind: 1.22 m/s

Live Hosted link: https://dnpwheathershow-if7f.vercel.app/
Github repo: https://github.com/pranitha080511/dnpwheathershow

```
        const WeatherCard = ({ data }) => {
  return (
    <div className="bg-white shadow-lg rounded-xl p-6 w-80 text-center">
      <h2 className="text-2xl font-bold mb-2">{data.name}</h2>
      <p className="text-xl">🌡️ {data.main.temp} °C</p>
      <p className="capitalize">☁️ {data.weather[0].description}</p>
      <p>💧 Humidity: {data.main.humidity}%</p>
      <p>🌬️ Wind: {data.wind.speed} m/s</p>
    </div>
  );
};

export default WeatherCard;

const SearchBar = ({ city, setCity, fetchWeather }) => {
  return (
    <div className="w-full max-w-md flex flex-col sm:flex-row gap-3 mb-8">

      <input
        type="text"
        placeholder="Enter city name"
        value={city}
        onChange={(e) => setCity(e.target.value)}
        className="flex-1 px-4 py-3 rounded-xl
                   bg-white text-black
                   border border-gray-300
                   focus:outline-none focus:ring-2 focus:ring-yellow-400
                   transition"
      />

      <button
        onClick={fetchWeather}
        className="px-6 py-3 rounded-xl
                   bg-black text-white font-semibold
                   active:scale-95
                   transition"
      >
        Search
      </button>
    </div>
  );
};

export default SearchBar;

import { useState, useEffect } from "react";

const Navbar = () => {
  const [open, setOpen] = useState(false);
  const [scrolled, setScrolled] = useState(false);

  // Change background on scroll
  useEffect(() => {
    const onScroll = () => setScrolled(window.scrollY > 20);
    window.addEventListener("scroll", onScroll);
    return () => window.removeEventListener("scroll", onScroll);
  }, []);

  const links = [
    { name: "Home", href: "#home" },
    { name: "Features", href: "#features" },
    { name: "Supporters", href: "#support" },
    { name: "Feedback", href: "#feedback" },
    { name: "FAQ", href: "#faq" },
  ];

  return (
    <>
      {/* NAVBAR */}
      <nav
        className={`fixed top-0 left-0 w-full z-50 transition-colors duration-300
        ${scrolled ? "bg-black shadow-lg" : "bg-black"}`}
      >
        <div className="max-w-7xl mx-auto px-6 h-[72px] flex items-center justify-between">
          {/* BRAND */}
          <div className="text-yellow-400 font-extrabold text-xl">
            DNPWeather<span className="text-white">Show</span>
          </div>

          {/* DESKTOP MENU */}
          <div className="hidden md:flex items-center gap-8 text-white">
            {links.map((link) => (
              <a
                key={link.name}
                href={link.href}
                className="hover:text-yellow-400 transition"
              >
                {link.name}
              </a>
            ))}


           <a
  href="/auth"
  className="px-4 py-2 rounded-lg bg-yellow-400 text-black
             font-semibold hover:opacity-90 transition"
>
  Login / Sign Up
</a>

          </div>

        </div>
      </nav>

      {/* MOBILE MENU */}
      {open && (
        <div className="fixed inset-0 bg-black z-50 md:hidden">
          <div className="p-6 flex flex-col gap-6 text-white">
            <button
              className="self-end text-yellow-400 text-3xl"
              onClick={() => setOpen(false)}
            >
              ✕
            </button>

            {links.map((link) => (
              <a
                key={link.name}
                href={link.href}
                onClick={() => setOpen(false)}
                className="text-lg hover:text-yellow-400"
              >
                {link.name}
              </a>
            ))}

            <a
              href="#login"
              onClick={() => setOpen(false)}
              className="mt-4 px-4 py-2 rounded-lg border border-yellow-400
                         text-yellow-400 text-center"
            >
              Login
            </a>
            <a
              href="#signup"
              onClick={() => setOpen(false)}
              className="px-4 py-2 rounded-lg bg-yellow-400 text-black
                        font-semibold text-center"
            >
              Sign Up
            </a>
          </div>
        </div>
      )}
    </>
  );
};

export default Navbar;

const Loader = () => {
  return (
    <p className="mt-4 text-lg font-medium text-black">
      Loading weather...
    </p>
  );
};

export default Loader;

import { useState } from "react";
import { getWeather } from "../services/weatherApi";
import SearchBar from "../components/SearchBar";
import WeatherCard from "../components/WeatherCard";
import Loader from "../components/Loader";
import Error from "../components/Error";

const Dashboard = () => {
  const [city, setCity] = useState("");
  const [weather, setWeather] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState("");

  const fetchWeather = async () => {
    if (!city) return;
    setLoading(true);
    setError("");

    try {
      const res = await getWeather(city);
      setWeather(res.data);
    } catch {
      setError("City not found");
    } finally {
      setLoading(false);
    }
  };

  return (
    <div
      className="min-h-screen flex flex-col items-center justify-start pt-24 px-4
                bg-gradient-to-br from-yellow-400 via-yellow-200 to-yellow-100"
    >
      <h1 className="text-3xl md:text-4xl font-extrabold mb-8 text-black">
        🌦️ Weather Dashboard
      </h1>

      <SearchBar
        city={city}
        setCity={setCity}
        fetchWeather={fetchWeather}
      />

      {loading && <Loader />}
      {error && <Error message={error} />}
      {weather && <WeatherCard data={weather} />}
    </div>
  );
};

export default Dashboard;

import { useLayoutEffect, useState } from "react";
import gsap from "gsap";
import { ScrollTrigger } from "gsap/ScrollTrigger";

gsap.registerPlugin(ScrollTrigger);

const Home = () => {
  const [index, setIndex] = useState(0);
  const [openFaq, setOpenFaq] = useState(null);

  const feedbacks = [
    { name: "Arun Kumar", msg: "Fast, clean and super accurate weather app." },
    { name: "Priya S", msg: "Animations feel premium. Loved the UI." },
    { name: "Rahul Dev", msg: "Perfect app for daily weather tracking." },
  ];

  useLayoutEffect(() => {
    const ctx = gsap.context(() => {
      gsap.fromTo(
  ".feature-card",
  { y: 40 },
  {
    y: 0,
    stagger: 0.15,
    duration: 0.6,
    ease: "power3.out",
    scrollTrigger: {
      trigger: ".features-section",
      start: "top 80%",
      once: true,
    },
  }
);
// SUPPORTED BY (SAFE GRID ANIMATION)
gsap.fromTo(
  ".support-card",
  { y: 40 },
  {
    y: 0,
    stagger: 0.15,
    duration: 0.6,
    ease: "power3.out",
    scrollTrigger: {
      trigger: ".support-section",
      start: "top 85%",
      once: true,
    },
  }
);

      // FEEDBACK
      gsap.from(".feedback-card", {
        scrollTrigger: {
          trigger: ".feedback-section",
          start: "top 85%",
          once: true,
        },
        autoAlpha: 0,
        y: 30,
        duration: 0.6,
        ease: "power3.out",
      });
    });


    return () => ctx.revert();
  }, []);

  return (
    <div className="w-full overflow-x-hidden pt-[72px]">

      {/* HERO */}
      <section className="h-screen flex items-center justify-center bg-gradient-to-br from-yellow-400 via-
yellow-200 to-yellow-200" id="home">
        <div className="text-center px-6">
          <h1 className="text-5xl md:text-7xl font-extrabold text-black drop-shadow-
[0_4px_12px_rgba(0,0,0,0.25)]">
            DNPWeather
          </h1>
          <p className="mt-4 text-lg md:text-xl text-black/90 font-medium">
            Real-time. Smart. Accurate.
          </p>
          <button
            className="mt-8 px-10 py-3 rounded-xl bg-black text-yellow-400
            font-semibold text-lg hover:scale-105 transition-all"
          >
            Get Started
          </button>
        </div>
      </section>

      {/* POWERFUL FEATURES */}
      <section className="features-section py-32 bg-black text-white"id="features">
        <h2 className="text-center text-3xl md:text-4xl font-extrabold text-yellow-400 mb-16">
          Powerful Features
        </h2>

        <div className="max-w-6xl mx-auto grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-10 px-6">
          {[
            ["Live Weather", "Get real-time weather updates for any city instantly."],
            ["Accurate Forecast", "Highly reliable data powered by trusted weather APIs."],
            ["Humidity Tracking", "Monitor humidity levels to plan your day better."],
            ["Wind Analysis", "Understand wind speed and direction with clarity."],
            ["Fast Search", "Lightning-fast city search with instant results."],
            ["Minimal UI", "Clean and distraction-free design for better focus."],
          ].map(([title, desc], i) => (
            <div
              key={i}
              className="feature-card p-8 rounded-2xl bg-zinc-900
                         border border-yellow-400/20
                         transition-all duration-300
                         hover:-translate-y-2
                         hover:border-yellow-400
                         hover:shadow-[0_0_30px_rgba(250,204,21,0.4)]"
            >
              <h3 className="text-xl font-bold text-yellow-400 mb-3">
                {title}
              </h3>
              <p className="text-sm text-gray-300 leading-relaxed">
                {desc}
              </p>
            </div>
          ))}
        </div>
      </section>
      {/* MOSTLY SUPPORTED BY */}
<section className="support-section py-32 bg-gradient-to-br from-yellow-400 via-yellow-200 to-yellow-200
text-black" id="support">
  <h2 className="text-center text-3xl md:text-4xl font-extrabold mb-16">
    Mostly Supported By
  </h2>

  <div className="max-w-6xl mx-auto grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-10 px-6">
    {[
      {
        title: "Marginal Farmers",
        desc: "Helps farmers plan irrigation, harvesting, and crop protection using accurate weather data.",
        icon: " 🌽 ",
      },
      {
        title: "Transport Workers",
        desc: "Assists drivers in avoiding delays caused by rain, fog, and strong winds.",
        icon: " 🚚 ",
      },
      {
        title: "Construction Workers",
        desc: "Supports daily wage workers by predicting unsafe weather conditions.",
        icon: " 🏗️ ",
      },
      {
        title: "Delivery Partners",
        desc: "Improves delivery efficiency with real-time weather awareness.",
        icon: " 📦 ",
      },
      {
        title: "Street Vendors",
        desc: "Helps vendors protect goods from sudden rain or extreme heat.",
        icon: " ☂️ ",
      },
      {
        title: "Rural Communities",
        desc: "Provides simple, accessible weather updates for rural regions.",
        icon: "🏡",
      },
    ].map((item, i) => (
      <div
        key={i}
        className="support-card p-8 rounded-2xl bg-black text-yellow-400
                   transition-all duration-300
                   hover:-translate-y-2
                   hover:shadow-[0_0_30px_rgba(0,0,0,0.4)]"
      >
        <div className="text-4xl mb-4">{item.icon}</div>

        <h3 className="text-xl font-bold mb-3">
          {item.title}
        </h3>

        <p className="text-sm text-yellow-300/90 leading-relaxed">
          {item.desc}
        </p>
      </div>
    ))}
  </div>
</section>

      {/* USER FEEDBACK */}
<section className="feedback-section py-32 bg-black text-white overflow-hidden"id="feedback">
  <h2 className="text-center text-3xl md:text-4xl font-extrabold text-yellow-400 mb-16">
    User Feedback
  </h2>

  <div className="relative max-w-3xl mx-auto px-6">
    {/* Carousel */}
    <div className="overflow-hidden">
      <div
        className="feedback-track flex"
        style={{ transform: `translateX(-${index * 100}%)` }}
      >
        {feedbacks.map((item, i) => (
          <div
            key={i}
            className="feedback-slide min-w-full flex justify-center"
          >
            <div className="feedback-card bg-zinc-900 p-10 rounded-2xl text-center max-w-xl">
              <p className="text-lg md:text-xl italic leading-relaxed">
                “{item.msg}”
              </p>
              <h4 className="mt-6 text-yellow-400 font-semibold">
                — {item.name}
              </h4>
            </div>
          </div>
        ))}
      </div>
    </div>

    {/* Controls */}
    <div className="flex justify-center gap-6 mt-10">
```

```
      <button
        onClick={() =>
          setIndex(index === 0 ? feedbacks.length - 1 : index - 1)
        }
        className="px-6 py-2 rounded-xl bg-yellow-400 text-black font-semibold
                   hover:scale-105 transition"
      >
        Prev
      </button>

      <button
        onClick={() => setIndex((index + 1) % feedbacks.length)}
        className="px-6 py-2 rounded-xl bg-yellow-400 text-black font-semibold
                   hover:scale-105 transition"
      >
        Next
      </button>
    </div>

    {/* Dots */}
    <div className="flex justify-center gap-3 mt-6">
      {feedbacks.map((_, i) => (
        <span
          key={i}
          className={`h-3 w-3 rounded-full transition
            ${i === index ? "bg-yellow-400" : "bg-zinc-600"}`}
        />
      ))}
    </div>
  </div>
</section>
{/* FAQ */}
<section className="faq-section py-32 bg-gradient-to-br from-yellow-400 via-yellow-200 to-yellow-200 text-
black"id="faq">
  <h2 className="text-center text-3xl md:text-4xl font-extrabold mb-16">
    Frequently Asked Questions
  </h2>

  <div className="max-w-3xl mx-auto px-6 space-y-4">
    {[
      {
        q: "Is DNPWeather free to use?",
        a: "Yes. DNPWeather is completely free and accessible for everyone.",
      },
      {
        q: "How accurate is the weather data?",
        a: "We use trusted real-time weather APIs to provide highly accurate forecasts.",
      },
      {
        q: "Who benefits the most from DNPWeather?",
        a: "Farmers, transport workers, daily wage workers, and rural communities benefit the most.",
      },
      {
        q: "Can I use this app on mobile devices?",
        a: "Yes. DNPWeather is fully responsive and works smoothly on all devices.",
      },
    ].map((item, i) => {
      const isOpen = openFaq === i;

      return (
        <div
          key={i}
          className="bg-black text-white rounded-2xl"
        >
          {/* QUESTION */}
          <button
            onClick={() => setOpenFaq(isOpen ? null : i)}
            className="w-full px-6 py-5 flex justify-between items-center
                       font-semibold text-left"
          >
            <span>{item.q}</span>
            <span
              className={`text-yellow-400 text-2xl transition-transform duration-300 ${
                isOpen ? "rotate-45" : ""
              }`}
            >
              +
            </span>
          </button>

          {/* ANSWER */}
          <div
            className={`overflow-hidden transition-all duration-300 ${
              isOpen ? "max-h-40 px-6 pb-6" : "max-h-0 px-6"
            }`}
          >
            <p className="text-gray-300">
              {item.a}
            </p>
          </div>
        </div>
      );
    })}
  </div>
</section>



    </div>
  );
};

export default Home;

import { useState } from "react";
import { useNavigate } from "react-router-dom";

const Auth = () => {
  const [isLogin, setIsLogin] = useState(true);
  const navigate = useNavigate();

  const handleSubmit = (e) => {
    e.preventDefault();

    // 🔐 later you’ll add real auth here
    navigate("/dashboard");
  };

  return (
    <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-yellow-400 via-
yellow-200 to-yellow-100 px-6">
      <div className="bg-black text-white w-full max-w-md p-8 rounded-2xl shadow-xl">
        <h2 className="text-3xl font-extrabold text-yellow-400 text-center mb-6">
          {isLogin ? "Login" : "Sign Up"}
        </h2>

        <form onSubmit={handleSubmit} className="space-y-4">
          {!isLogin && (
            <input
              type="text"
              placeholder="Full Name"
              className="w-full px-4 py-3 rounded-lg bg-zinc-900 border border-zinc-700"
            />
          )}

          <input
            type="email"
            placeholder="Email"
            className="w-full px-4 py-3 rounded-lg bg-zinc-900 border border-zinc-700"
          />

          <input
            type="password"
            placeholder="Password"
            className="w-full px-4 py-3 rounded-lg bg-zinc-900 border border-zinc-700"
          />

          <button
            type="submit"
            className="w-full py-3 rounded-lg bg-yellow-400 text-black font-semibold"
          >
            {isLogin ? "Login" : "Create Account"}
          </button>
        </form>

        <p className="mt-6 text-center text-sm text-gray-400">
          {isLogin ? "Don't have an account?" : "Already have an account?"}
          <button
            onClick={() => setIsLogin(!isLogin)}
            className="ml-2 text-yellow-400 font-semibold"
          >
            {isLogin ? "Sign Up" : "Login"}
          </button>
        </p>
      </div>
    </div>
  );
};

export default Auth;
```