Build this as my initial prototype

Copy-paste this component to /components/ui folder:
```tsx
sidebar.tsx
"use client";

import { cn } from "@/lib/utils";
import Link, { LinkProps } from "next/link";
import React, { useState, createContext, useContext } from "react";
import { AnimatePresence, motion } from "framer-motion";
import { Menu, X } from "lucide-react";

interface Links {
  label: string;
  href: string;
  icon: React.JSX.Element | React.ReactNode;
}

interface SidebarContextProps {
  open: boolean;
  setOpen: React.Dispatch<React.SetStateAction<boolean>>;
  animate: boolean;
}

const SidebarContext = createContext<SidebarContextProps | undefined>(
  undefined
);

export const useSidebar = () => {
  const context = useContext(SidebarContext);
  if (!context) {
    throw new Error("useSidebar must be used within a SidebarProvider");
  }
  return context;
};

export const SidebarProvider = ({
  children,
  open: openProp,
  setOpen: setOpenProp,
  animate = true,
}: {
  children: React.ReactNode;
  open?: boolean;
  setOpen?: React.Dispatch<React.SetStateAction<boolean>>;
  animate?: boolean;
}) => {
  const [openState, setOpenState] = useState(false);

  const open = openProp !== undefined ? openProp : openState;
  const setOpen = setOpenProp !== undefined ? setOpenProp : setOpenState;

  return (
    <SidebarContext.Provider value={{ open, setOpen, animate }}>
      {children}
    </SidebarContext.Provider>
  );
};

export const Sidebar = ({
  children,
  open,
  setOpen,
  animate,
}: {
  children: React.ReactNode;
  open?: boolean;
  setOpen?: React.Dispatch<React.SetStateAction<boolean>>;
  animate?: boolean;
}) => {
  return (
    <SidebarProvider open={open} setOpen={setOpen} animate={animate}>
      {children}
    </SidebarProvider>
  );
};

export const SidebarBody = (props: React.ComponentProps<typeof motion.div>) => {
  return (
    <>
      <DesktopSidebar {...props} />
      <MobileSidebar {...(props as React.ComponentProps<"div">)} />
    </>
  );
};

export const DesktopSidebar = ({
  className,
  children,
  ...props
}: React.ComponentProps<typeof motion.div>) => {
  const { open, setOpen, animate } = useSidebar();
  return (
    <motion.div
      className={cn(
        "h-full px-4 py-4 hidden md:flex md:flex-col bg-neutral-100 dark:bg-neutral-800 w-[300px]
flex-shrink-0",
        className
      )}
      animate={{
        width: animate ? (open ? "300px" : "60px") : "300px",
      }}
      onMouseEnter={() => setOpen(true)}
      onMouseLeave={() => setOpen(false)}
      {...props}
    >
      {children}
    </motion.div>
  );
};

export const MobileSidebar = ({
  className,
  children,
  ...props
}: React.ComponentProps<"div">) => {
  const { open, setOpen } = useSidebar();
  return (
    <>
      <div
        className={cn(
          "h-10 px-4 py-4 flex flex-row md:hidden items-center justify-between bg-neutral-100
dark:bg-neutral-800 w-full"
        )}
        {...props}
      >
        <div className="flex justify-end z-20 w-full">
          <Menu
            className="text-neutral-800 dark:text-neutral-200 cursor-pointer"
            onClick={() => setOpen(!open)}
          />
        </div>
        <AnimatePresence>
          {open && (
            <motion.div
              initial={{ x: "-100%", opacity: 0 }}
              animate={{ x: 0, opacity: 1 }}
              exit={{ x: "-100%", opacity: 0 }}
              transition={{
                duration: 0.3,
                ease: "easeInOut",
              }}
              className={cn(
                "fixed h-full w-full inset-0 bg-white dark:bg-neutral-900 p-10 z-[100] flex flex-col justify-between",
                className
              )}
            >
              <div
                className="absolute right-10 top-10 z-50 text-neutral-800 dark:text-neutral-200 cursor-pointer"
                onClick={() => setOpen(!open)}
              >
                <X />
              </div>
              {children}
            </motion.div>
          )}
        </AnimatePresence>
      </div>
    </>
  );
};

export const SidebarLink = ({
  link,
  className,
  ...props
}: {
  link: Links;
  className?: string;
  props?: LinkProps;
}) => {
  const { open, animate } = useSidebar();
  return (
    <Link
      href={link.href}
      className={cn(
        "flex items-center justify-start gap-2 group/sidebar py-2",
        className
      )}
      {...props}
    >
      {link.icon}
      <motion.span
        animate={{
          display: animate ? (open ? "inline-block" : "none") : "inline-block",
          opacity: animate ? (open ? 1 : 0) : 1,
        }}
        className="text-neutral-700 dark:text-neutral-200 text-sm
group-hover/sidebar:translate-x-1 transition duration-150 whitespace-pre inline-block !p-0 !m-0"
      >
        {link.label}
      </motion.span>
    </Link>
  );
};
```

demo.tsx

```
"use client";
import React, { useState } from "react";
import { Sidebar, SidebarBody, SidebarLink } from "@/components/ui/sidebar";
import { LayoutDashboard, UserCog, Settings, LogOut } from "lucide-react";
import Link from "next/link";
import { motion } from "framer-motion";
import Image from "next/image";
import { cn } from "@/lib/utils";

export function SidebarDemo() {
  const links = [
    {
      label: "Dashboard",
      href: "#",
      icon: (
        <LayoutDashboard className="text-neutral-700 dark:text-neutral-200 h-5 w-5
flex-shrink-0" />
      ),
    },
    {
      label: "Profile",
      href: "#",
      icon: (
        <UserCog className="text-neutral-700 dark:text-neutral-200 h-5 w-5 flex-shrink-0" />
      ),
    },
    {
      label: "Settings",
      href: "#",
      icon: (
        <Settings className="text-neutral-700 dark:text-neutral-200 h-5 w-5 flex-shrink-0" />
      ),
    },
    {
      label: "Logout",
      href: "#",
      icon: (
        <LogOut className="text-neutral-700 dark:text-neutral-200 h-5 w-5 flex-shrink-0" />
      ),
    },
  ];
  const [open, setOpen] = useState(false);
  return (
    <div
      className={cn(
        "rounded-md flex flex-col md:flex-row bg-gray-100 dark:bg-neutral-800 w-full flex-1
max-w-7xl mx-auto border border-neutral-200 dark:border-neutral-700 overflow-hidden",
        "h-[60vh]" // for your use case, use `h-screen` instead of `h-[60vh]`
      )}
    >
      <Sidebar open={open} setOpen={setOpen}>
        <SidebarBody className="justify-between gap-10">
          <div className="flex flex-col flex-1 overflow-y-auto overflow-x-hidden">
            {open ? <Logo /> : <LogoIcon />}
            <div className="mt-8 flex flex-col gap-2">
              {links.map((link, idx) => (
                <SidebarLink key={idx} link={link} />
              ))}
            </div>
          </div>
          <div>
            <SidebarLink
              link={{
                label: "Manu Arora",
                href: "#",
                icon: (
                  <Image
                    src="https://assets.aceternity.com/manu.png"
                    className="h-7 w-7 flex-shrink-0 rounded-full"
                    width={50}
                    height={50}
                    alt="Avatar"
                  />
                ),
              }}
            />
          </div>
        </SidebarBody>
      </Sidebar>
      <Dashboard />
    </div>
  );
}

export const Logo = () => {
  return (
    <Link
      href="#"
      className="font-normal flex space-x-2 items-center text-sm text-black py-1 relative z-20"
    >
      <div className="h-5 w-6 bg-black dark:bg-white rounded-br-lg rounded-tr-sm rounded-tl-lg rounded-bl-sm flex-shrink-0" />
      <motion.span
        initial={{ opacity: 0 }}
        animate={{ opacity: 1 }}
        className="font-medium text-black dark:text-white whitespace-pre"
      >
        Acet Labs
      </motion.span>
    </Link>
  );
};

export const LogoIcon = () => {
  return (
    <Link
      href="#"
      className="font-normal flex space-x-2 items-center text-sm text-black py-1 relative z-20"
    >
      <div className="h-5 w-6 bg-black dark:bg-white rounded-br-lg rounded-tr-sm rounded-tl-lg rounded-bl-sm flex-shrink-0" />
    </Link>
  );
};

// Dummy dashboard component with content
const Dashboard = () => {
  return (
    <div className="flex flex-1">
      <div className="p-2 md:p-10 rounded-tl-2xl border border-neutral-200 dark:border-neutral-700 bg-white dark:bg-neutral-900 flex flex-col gap-2 flex-1 w-full h-full">
        <div className="flex gap-2">
          {[...new Array(4)].map((i) => (
            <div
              key={"first-array" + i}
              className="h-20 w-full rounded-lg  bg-gray-100 dark:bg-neutral-800 animate-pulse"
            ></div>
          ))}
        </div>
        <div className="flex gap-2 flex-1">
          {[...new Array(2)].map((i) => (
            <div
              key={"second-array" + i}
              className="h-full w-full rounded-lg  bg-gray-100 dark:bg-neutral-800 animate-pulse"
            ></div>
          ))}
        </div>
      </div>
    </div>
  );
};

```

Install these NPM dependencies:

```bash
framer-motion
```