button.tsx

import \* as React from "react"

import { Slot } from "@radix-ui/react-slot"

import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "../../lib/utils"

const buttonVariants = cva(

"inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm font-medium transition-colors focus-visible:outline-none focus-visible:ring-1 focus-visible:ring-ring disabled:pointer-events-none disabled:opacity-50",

{

variants: {

variant: {

default: "bg-primary text-primary-foreground shadow hover:bg-primary/90",

destructive:

"bg-destructive text-destructive-foreground shadow-sm hover:bg-destructive/90",

outline:

"border border-input bg-background shadow-sm hover:bg-accent hover:text-accent-foreground",

secondary:

"bg-secondary text-secondary-foreground shadow-sm hover:bg-secondary/80",

ghost: "hover:bg-accent hover:text-accent-foreground",

link: "text-primary underline-offset-4 hover:underline",

},

size: {

default: "h-9 px-4 py-2",

sm: "h-8 rounded-md px-3 text-xs",

lg: "h-10 rounded-md px-8",

icon: "h-9 w-9",

},

},

defaultVariants: {

variant: "default",

size: "default",

},

}

)

export interface ButtonProps

extends React.ButtonHTMLAttributes<HTMLButtonElement>,

VariantProps<typeof buttonVariants> {

asChild?: boolean

}

const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(

({ className, variant, size, asChild = false, ...props }, ref) => {

const Comp = asChild ? Slot : "button"

return (

<Comp

className={cn(buttonVariants({ variant, size, className }))}

ref={ref}

{...props}

/>

)

}

)

Button.displayName = "Button"

export { Button, buttonVariants }