

Build this as my initial prototype

Copy-paste this component to /components/ui folder:

```
```tsx
```

```
header.tsx
```

```
"use client";
```

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

```
import {
```

```
 NavigationMenu,
```

```
 NavigationMenuContent,
```

```
 NavigationMenuItem,
```

```
 NavigationMenuLink,
```

```
 NavigationMenuList,
```

```
 NavigationMenuTrigger,
```

```
} from "@components/ui/navigation-menu";
```

```
import { Menu, MoveRight, X } from "lucide-react";
```

```
import { useState } from "react";
```

```
import Link from "next/link";
```

```
function Header1() {
```

```
 const navigationItems = [
```

```
 {
```

```
 title: "Home",
```

```
 href: "/",
```

```
 description: "",
```

```
 },
```

```
 {
```

```
 title: "Product",
```

```
 description: "Managing a small business today is already tough.",
```

```
 items: [
```

```
 {
```

```
 title: "Reports",
```

```
 href: "/reports",
```

```
 },
```

```
 {
```

```
 title: "Statistics",
```

```
 href: "/statistics",
```

```
 },
```

```
 {
```

```
 title: "Dashboards",
```

```
 href: "/dashboards",
```

```
 },
```

```
],
```

```
 },
```

```

 title: "Recordings",
 href: "/recordings",
 },
],
 },
 {
 title: "Company",
 description: "Managing a small business today is already tough.",
 items: [
 {
 title: "About us",
 href: "/about",
 },
 {
 title: "Fundraising",
 href: "/fundraising",
 },
 {
 title: "Investors",
 href: "/investors",
 },
 {
 title: "Contact us",
 href: "/contact",
 },
],
 },
],
];

const [isOpen, setOpen] = useState(false);
return (
 <header className="w-full z-40 fixed top-0 left-0 bg-background">
 <div className="container relative mx-auto min-h-20 flex gap-4 flex-row lg:grid
lg:grid-cols-3 items-center">
 <div className="justify-start items-center gap-4 lg:flex hidden flex-row">
 <NavigationMenu className="flex justify-start items-start">
 <NavigationMenuList className="flex justify-start gap-4 flex-row">
 {navigationItems.map((item) => (
 <NavigationMenuItem key={item.title}>
 {item.href ? (
 <>
 <NavigationMenuLink>
 <Button variant="ghost">{item.title}</Button>
 </NavigationMenuLink>

```

```

 </>
) : (
 <>
 <NavigationMenuTrigger className="font-medium text-sm">
 {item.title}
 </NavigationMenuTrigger>
 <NavigationMenuContent className="!w-[450px] p-4">
 <div className="flex flex-col lg:grid grid-cols-2 gap-4">
 <div className="flex flex-col h-full justify-between">
 <div className="flex flex-col">
 <p className="text-base">{item.title}</p>
 <p className="text-muted-foreground text-sm">
 {item.description}
 </p>
 </div>
 <Button size="sm" className="mt-10">
 Book a call today
 </Button>
 </div>
 <div className="flex flex-col text-sm h-full justify-end">
 {item.items?.map((subItem) => (
 <NavigationMenuLink
 href={subItem.href}
 key={subItem.title}
 className="flex flex-row justify-between items-center
hover:bg-muted py-2 px-4 rounded"
 >
 {subItem.title}
 <MoveRight className="w-4 h-4
text-muted-foreground" />
 </NavigationMenuLink>
))}
 </div>
 </div>
 </NavigationMenuContent>
 </>
)}
</NavigationMenuItem>
)}}
</NavigationMenuList>
</NavigationMenu>
</div>
<div className="flex lg:justify-center">
 <p className="font-semibold">TWBlocks</p>

```

```

</div>
<div className="flex justify-end w-full gap-4">
 <Button variant="ghost" className="hidden md:inline">
 Book a demo
 </Button>
 <div className="border-r hidden md:inline"></div>
 <Button variant="outline">Sign in</Button>
 <Button>Get started</Button>
</div>
<div className="flex w-12 shrink lg:hidden items-end justify-end">
 <Button variant="ghost" onClick={() => setOpen(!isOpen)}>
 {isOpen ? <X className="w-5 h-5" /> : <Menu className="w-5 h-5" />}
 </Button>
 {isOpen && (
 <div className="absolute top-20 border-t flex flex-col w-full right-0
bg-background shadow-lg py-4 container gap-8">
 {navigationItems.map((item) => (
 <div key={item.title}>
 <div className="flex flex-col gap-2">
 {item.href ? (
 <Link
 href={item.href}
 className="flex justify-between items-center"
 >
 {item.title}
 <MoveRight className="w-4 h-4 stroke-1
text-muted-foreground" />
 </Link>
) : (
 <p className="text-lg">{item.title}</p>
)}
 {item.items &&
 item.items.map((subItem) => (
 <Link
 key={subItem.title}
 href={subItem.href}
 className="flex justify-between items-center"
 >

 {subItem.title}

 <MoveRight className="w-4 h-4 stroke-1" />
 </Link>
)}}
)}}
)}
 </div>
)}
</div>

```

```

 </div>
 </div>
)}
 </div>
}
</div>
</div>
</header>
);
}

```

```
export { Header1 };
```

demo.tsx

```
import { Header1 } from "@components/ui/header"
```

```
function HeaderDemo() {
 return (
 <div className="block">
 <Header1 />
 </div>
);
}

```

```
export { HeaderDemo };
...

```

Copy-paste these files for dependencies:

```

````tsx
shadcn/button
import * as React from "react"
import { Slot } from "@radix-ui/react-slot"
import { cva, type VariantProps } from "class-variance-authority"

import { cn } from "@lib/utls"

```

```

const buttonVariants = cva(
  "inline-flex items-center justify-center whitespace-nowrap rounded-md text-sm font-medium
  ring-offset-background transition-colors focus-visible:outline-none focus-visible:ring-2
  focus-visible:ring-ring focus-visible:ring-offset-2 disabled:pointer-events-none
  disabled:opacity-50",
  {
    variants: {
      variant: {

```

```

    default: "bg-primary text-primary-foreground hover:bg-primary/90",
    destructive:
      "bg-destructive text-destructive-foreground hover:bg-destructive/90",
    outline:
      "border border-input bg-background hover:bg-accent hover:text-accent-foreground",
    secondary:
      "bg-secondary text-secondary-foreground hover:bg-secondary/80",
    ghost: "hover:bg-accent hover:text-accent-foreground",
    link: "text-primary underline-offset-4 hover:underline",
  },
  size: {
    default: "h-10 px-4 py-2",
    sm: "h-9 rounded-md px-3",
    lg: "h-11 rounded-md px-8",
    icon: "h-10 w-10",
  },
},
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 }

...

```tsx
shadcn/navigation-menu
import * as React from "react"
import { ChevronDownIcon } from "@radix-ui/react-icons"
import * as NavigationMenuPrimitive from "@radix-ui/react-navigation-menu"
import { cva } from "class-variance-authority"

import { cn } from "@lib/utils"

const NavigationMenu = React.forwardRef<
 React.ElementRef<typeof NavigationMenuPrimitive.Root>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Root>
>(({ className, children, ...props }, ref) => (
 <NavigationMenuPrimitive.Root
 ref={ref}
 className={cn(
 "relative z-10 flex max-w-max flex-1 items-center justify-center",
 className
)}
 {...props}
 >
 {children}
 <NavigationMenuViewport />
 </NavigationMenuPrimitive.Root>
))
NavigationMenu.displayName = NavigationMenuPrimitive.Root.displayName

const NavigationMenuList = React.forwardRef<
 React.ElementRef<typeof NavigationMenuPrimitive.List>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.List>
>(({ className, ...props }, ref) => (
 <NavigationMenuPrimitive.List
 ref={ref}
 className={cn(
 "group flex flex-1 list-none items-center justify-center space-x-1",
 className
)}
 {...props}
 />
))
NavigationMenuList.displayName = NavigationMenuPrimitive.List.displayName

```

```
const NavigationMenuItem = NavigationMenuPrimitive.Item
```

```
const navigationMenuTriggerStyle = cva(
 "group inline-flex h-9 w-max items-center justify-center rounded-md bg-background px-4 py-2
 text-sm font-medium transition-colors hover:bg-accent hover:text-accent-foreground
 focus:bg-accent focus:text-accent-foreground focus:outline-none disabled:pointer-events-none
 disabled:opacity-50 data-[active]:bg-accent/50 data-[state=open]:bg-accent/50"
)
```

```
const NavigationMenuTrigger = React.forwardRef(
 React.ElementRef<typeof NavigationMenuPrimitive.Trigger>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Trigger>
>(({ className, children, ...props }, ref) => (
 <NavigationMenuPrimitive.Trigger
 ref={ref}
 className={cn(navigationMenuTriggerStyle(), "group", className)}
 {...props}
 >
 {children}{" "
 <ChevronDownIcon
 className="relative top-[1px] ml-1 h-3 w-3 transition duration-300
group-data-[state=open]:rotate-180"
 aria-hidden="true"
 />
 </NavigationMenuPrimitive.Trigger>
))
NavigationMenuTrigger.displayName = NavigationMenuPrimitive.Trigger.displayName
```

```
const NavigationMenuContent = React.forwardRef(
 React.ElementRef<typeof NavigationMenuPrimitive.Content>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Content>
>(({ className, ...props }, ref) => (
 <NavigationMenuPrimitive.Content
 ref={ref}
 className={cn(
 "left-0 top-0 w-full data-[motion^=from-]:animate-in data-[motion^=to-]:animate-out
data-[motion^=from-]:fade-in data-[motion^=to-]:fade-out
data-[motion=from-end]:slide-in-from-right-52 data-[motion=from-start]:slide-in-from-left-52
data-[motion=to-end]:slide-out-to-right-52 data-[motion=to-start]:slide-out-to-left-52 md:absolute
md:w-auto ",
 className
)}
 {...props}
```



```

 />
))
 NavigationMenuContent.displayName = NavigationMenuPrimitive.Content.displayName

 const NavigationMenuLink = NavigationMenuPrimitive.Link

 const NavigationMenuViewport = React.forwardRef<
 React.ElementRef<typeof NavigationMenuPrimitive.Viewport>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Viewport>
 >(({ className, ...props }, ref) => (
 <div className={cn("absolute left-0 top-full flex justify-center")}>
 <NavigationMenuPrimitive.Viewport
 className={cn(
 "origin-top-center relative mt-1.5 h-[var(--radix-navigation-menu-viewport-height)] w-full
 overflow-hidden rounded-md border bg-popover text-popover-foreground shadow
 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:zoom-out-95
 data-[state=open]:zoom-in-90 md:w-[var(--radix-navigation-menu-viewport-width)]",
 className
)}
 ref={ref}
 {...props}
 />
 </div>
))
 NavigationMenuViewport.displayName =
 NavigationMenuPrimitive.Viewport.displayName

 const NavigationMenuIndicator = React.forwardRef<
 React.ElementRef<typeof NavigationMenuPrimitive.Indicator>,
 React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Indicator>
 >(({ className, ...props }, ref) => (
 <NavigationMenuPrimitive.Indicator
 ref={ref}
 className={cn(
 "top-full z-[1] flex h-1.5 items-end justify-center overflow-hidden
 data-[state=visible]:animate-in data-[state=hidden]:animate-out data-[state=hidden]:fade-out
 data-[state=visible]:fade-in",
 className
)}
 {...props}
 >
 <div className="relative top-[60%] h-2 w-2 rotate-45 rounded-tl-sm bg-border shadow-md"
 />
 </NavigationMenuPrimitive.Indicator>
)

```

```
))
NavigationMenuIndicator.displayName =
 NavigationMenuPrimitive.Indicator.displayName
```

```
export {
 navigationMenuTriggerStyle,
 NavigationMenu,
 NavigationMenuList,
 NavigationMenuItem,
 NavigationMenuContent,
 NavigationMenuTrigger,
 NavigationMenuLink,
 NavigationMenuIndicator,
 NavigationMenuViewport,
}
```

```
...
```

Install these NPM dependencies:

```
```bash
lucide-react, @radix-ui/react-slot, class-variance-authority, @radix-ui/react-icons,
@radix-ui/react-navigation-menu
```
```