

Navbar Components - Complete Source Code & Documentation

Table of Contents

1. Main Navbar
2. Cart Button
3. Dark Mode Toggle
4. Links Dropdown
5. Navigation Search
6. User Menu
7. User Button Wrapper

Main Navbar

components/navbar/Navbar.tsx

Purpose: Main navigation component with responsive design **Location:** `/components/navbar/Navbar.tsx`

```
"use client";

import Link from "next/link";
import { useState } from "react";
import { useAuth } from "@clerk/nextjs";
import { Menu, X, ShoppingBag, Heart, User } from "lucide-react";
import { Button } from "@/components/ui/button";
import { Sheet, SheetContent, SheetTrigger } from "@/components/ui/sheet";
import { Badge } from "@/components/ui/badge";
import { useCart } from "@/utils/actions";
import { useFavorites } from "@/utils/actions";
import { cn } from "@/lib/utils";

const navigation = [
  { name: "Home", href: "/" },
  { name: "Products", href: "/products" },
  { name: "About", href: "/about" },
  { name: "Reviews", href: "/reviews" },
];

const userNavigation = [
  { name: "Profile", href: "/profile" },
  { name: "Orders", href: "/orders" },
  { name: "Favorites", href: "/favorites" },
  { name: "Sign out", href: "/sign-out" },
];

export default function Navbar() {
  const [mobileMenuOpen, setMobileMenuOpen] = useState(false);
  const { isAuthenticated, user } = useAuth();
  const { cartItemsCount } = useCart();
  const { favoritesCount } = useFavorites();

  return (
    <nav className="bg-background/95 backdrop-blur supports-[backdrop-filter]:bg-background/60 border-b border-border sticky top-0 z-50">
      <div className="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8">
        <div className="flex h-16 items-center justify-between">
          {/* Logo */}
          <div className="flex lg:flex-1">
            <Link href="/" className="-m-1.5 p-1.5">
              <span className="sr-only">Wine Store</span>

```

```

        <div className="flex items-center space-x-2">
            <div className="w-8 h-8 bg-gradient-to-br from-purple-600 to-pink-600 rounded-lg flex items-center justify-center">
                <span className="text-white font-bold text-sm">W</span>
            </div>
            <span className="text-xl font-bold bg-gradient-to-r from-purple-600 to-pink-600 bg-clip-text text-transparent">
                Wine Store
            </span>
        </div>
    </Link>
</div>

{/* Desktop Navigation */}
<div className="hidden lg:flex lg:gap-x-12">
    {navigation.map((item) => (
        <Link
            key={item.name}
            href={item.href}
            className="text-sm font-semibold leading-6 text-foreground
            hover:text-primary transition-colors"
            >
                {item.name}
            </Link>
        ))}
</div>

{/* Right side actions */}
<div className="flex items-center gap-4">
    {/* Search */}
    <div className="hidden lg:block">
        <Button
            variant="ghost"
            size="sm"
            className="text-muted-foreground"
            >
            <svg
                className="h-5 w-5"
                fill="none"
                stroke="currentColor"
                viewBox="0 0 24 24"
                >
                <path
                    strokeLinecap="round"
                    strokeLinejoin="round"
                    strokeWidth={2}
                    d="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114 0z"
                />
            </svg>
        </Button>
    </div>

    {/* Favorites */}
    <Link href="/favorites" className="relative">
        <Button
            variant="ghost"
            size="sm"
            className="text-muted-foreground"
            >
            <Heart className="h-5 w-5" />
            {favoritesCount > 0 && (
                <Badge
                    variant="secondary"
                    className="absolute -top-2 -right-2 h-5 w-5 rounded-full p-0 text-xs flex items-center justify-center"
                    >

```

```

        {favoritesCount}
      </Badge>
    )}
  </Button>
</Link>

{/* Cart */}
<Link href="/cart" className="relative">
  <Button
    variant="ghost"
    size="sm"
    className="text-muted-foreground"
  >
    <ShoppingBag className="h-5 w-5" />
    {cartItemsCount > 0 && (
      <Badge
        variant="secondary"
        className="absolute -top-2 -right-2 h-5 w-5 rounded-full p-
0 text-xs flex items-center justify-center"
      >
        {cartItemsCount}
      </Badge>
    )}
  </Button>
</Link>

{/* User Menu */}
{isSignedIn ? (
  <div className="relative">
    <Button
      variant="ghost"
      size="sm"
      className="text-muted-foreground"
    >
      <User className="h-5 w-5" />
    </Button>
    {/* Dropdown menu would go here */}
  </div>
) : (
  <div className="flex items-center gap-2">
    <Link href="/sign-in">
      <Button variant="ghost" size="sm">
        Sign in
      </Button>
    </Link>
    <Link href="/sign-up">
      <Button size="sm">Sign up</Button>
    </Link>
  </div>
)}

{/* Mobile menu button */}
<Sheet>
  <SheetTrigger asChild>
    <Button
      variant="ghost"
      size="sm"
      className="lg:hidden"
      onClick={() => setMobileMenuOpen(true)}
    >
      <span className="sr-only">Open main menu</span>
      <Menu className="h-6 w-6" />
    </Button>
  </SheetTrigger>
  <SheetContent side="right" className="w-[300px] sm:w-[400px]">
    <div className="flex flex-col h-full">

```

```

<div className="flex items-center justify-between mb-6">
  <Link href="/" className="text-xl font-bold">
    Wine Store
  </Link>
  <Button
    variant="ghost"
    size="sm"
    onClick={() => setMobileMenuOpen(false)}
  >
    <X className="h-6 w-6" />
  </Button>
</div>

{/* Mobile Navigation */}
<div className="flex flex-col space-y-4 mb-8">
  {navigation.map((item) => (
    <Link
      key={item.name}
      href={item.href}
      className="text-lg font-medium text-foreground
hover:text-primary transition-colors"
      onClick={() => setMobileMenuOpen(false)}
    >
      {item.name}
    </Link>
  ))}
</div>

{/* Mobile User Actions */}
<div className="flex flex-col space-y-4 mt-auto">
  {isSignedIn ? (
    <>
      <div className="border-t pt-4">
        <p className="text-sm text-muted-foreground mb-2">
          Signed in as
{user?.emailAddresses[0]?.emailAddress}
        </p>
        {userNavigation.map((item) => (
          <Link
            key={item.name}
            href={item.href}
            className="block text-sm text-foreground
hover:text-primary transition-colors py-2"
            onClick={() => setMobileMenuOpen(false)}
          >
            {item.name}
          </Link>
        ))}
      </div>
    </>
  ) : (
    <div className="flex flex-col space-y-2">
      <Link href="/sign-in">
        <Button variant="outline" className="w-full">
          Sign in
        </Button>
      </Link>
      <Link href="/sign-up">
        <Button className="w-full">Sign up</Button>
      </Link>
    </div>
  )}
</div>
</div>
</SheetContent>
</Sheet>

```

```
        </div>
      </div>
    </div>
  </nav>
);
}
```

Navbar Features Explained:

- **Responsive Design:** Desktop and mobile layouts
- **Authentication State:** Shows different content for signed-in users
- **Cart & Favorites:** Badge indicators for item counts
- **Mobile Menu:** Slide-out sheet for mobile navigation
- **Sticky Positioning:** Stays at top during scroll
- **Backdrop Blur:** Modern glass-morphism effect

Cart Button

components/navbar/CartButton.tsx

Purpose: Dedicated cart button component with item count **Location:** </components/navbar/CartButton.tsx>

```
"use client";

import Link from "next/link";
import { ShoppingBag } from "lucide-react";
import { Badge } from "@components/ui/badge";
import { Button } from "@components/ui/button";
import { useCart } from "@utils/actions";

export default function CartButton() {
  const { cartItemsCount } = useCart();

  return (
    <Link href="/cart">
      <Button variant="ghost" size="sm" className="relative">
        <ShoppingBag className="h-5 w-5" />
        {cartItemsCount > 0 && (
          <Badge
            variant="secondary"
            className="absolute -top-2 -right-2 h-5 w-5 rounded-full p-0 text-
xs flex items-center justify-center"
          >
            {cartItemsCount}
          </Badge>
        )}
        <span className="sr-only">Cart ({cartItemsCount} items)</span>
      </Button>
    </Link>
  );
}
```

Cart Button Features:

- **Item Count Badge:** Shows number of items in cart
- **Accessibility:** Screen reader support
- **Link Navigation:** Directs to cart page
- **Visual Feedback:** Badge appears when items exist

Dark Mode Toggle

components/navbar/DarkMode.tsx

Purpose: Theme switching between light and dark modes **Location:** /components/navbar/DarkMode.tsx

```
"use client";

import { Moon, Sun } from "lucide-react";
import { useTheme } from "next-themes";
import { Button } from "@/components/ui/button";
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuTrigger,
} from "@/components/ui/dropdown-menu";

export default function DarkMode() {
  const { setTheme } = useTheme();

  return (
    <DropdownMenu>
      <DropdownMenuTrigger asChild>
        <Button variant="ghost" size="sm">
          <Sun className="h-[1.2rem] w-[1.2rem] rotate-0 scale-100 transition-all dark:-rotate-90 dark:scale-0" />
          <Moon className="absolute h-[1.2rem] w-[1.2rem] rotate-90 scale-0 transition-all dark:rotate-0 dark:scale-100" />
          <span className="sr-only">Toggle theme</span>
        </Button>
      </DropdownMenuTrigger>
      <DropdownMenuContent align="end">
        <DropdownMenuItem onClick={() => setTheme("light")}>
          Light
        </DropdownMenuItem>
        <DropdownMenuItem onClick={() => setTheme("dark")}>
          Dark
        </DropdownMenuItem>
        <DropdownMenuItem onClick={() => setTheme("system")}>
          System
        </DropdownMenuItem>
      </DropdownMenuContent>
    </DropdownMenu>
  );
}
```

Dark Mode Features:

- **Icon Animation:** Smooth transition between sun/moon icons
- **Dropdown Menu:** Three theme options (light, dark, system)
- **System Detection:** Automatically follows OS preference
- **Smooth Transitions:** CSS animations for theme changes

Links Dropdown

components/navbar/LinksDropdown.tsx

Purpose: Dropdown navigation for main site links **Location:** /components/navbar/LinksDropdown.tsx

```
"use client";

import Link from "next/link";
import { ChevronDown } from "lucide-react";
```

```
import { Button } from "@components/ui/button";
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuTrigger,
} from "@components/ui/dropdown-menu";

const navigation = [
  { name: "Home", href: "/" },
  { name: "Products", href: "/products" },
  { name: "About", href: "/about" },
  { name: "Reviews", href: "/reviews" },
];

export default function LinksDropdown() {
  return (
    <DropdownMenu>
      <DropdownMenuTrigger asChild>
        <Button variant="ghost" className="flex items-center gap-1">
          Menu
          <ChevronDown className="h-4 w-4" />
        </Button>
      </DropdownMenuTrigger>
      <DropdownMenuContent align="start" className="w-48">
        {navigation.map((item) => (
          <DropdownMenuItem key={item.name} asChild>
            <Link href={item.href}>{item.name}</Link>
          </DropdownMenuItem>
        ))}
      </DropdownMenuContent>
    </DropdownMenu>
  );
}
```

Links Dropdown Features:

- **Compact Navigation:** Saves space in mobile layouts
- **Consistent Styling:** Matches other dropdown components
- **Accessibility:** Proper ARIA labels and keyboard navigation
- **Responsive:** Works well on smaller screens

Navigation Search

components/navbar/NavSearch.tsx

Purpose: Search functionality in the navigation bar **Location:** /components/navbar/NavSearch.tsx

```
"use client";

import { useState } from "react";
import { Search, X } from "lucide-react";
import { Button } from "@components/ui/button";
import { Input } from "@components/ui/input";

export default function NavSearch() {
  const [isSearchOpen, setIsSearchOpen] = useState(false);
  const [searchQuery, setSearchQuery] = useState("");

  const handleSearch = (e: React.FormEvent) => {
    e.preventDefault();
    if (searchQuery.trim()) {
      // Implement search functionality
      console.log("Searching for:", searchQuery);
    }
  };
}
```

```

        setIsSearchOpen(false);
        setSearchQuery("");
    }
};

if (!isSearchOpen) {
    return (
        <Button
            variant="ghost"
            size="sm"
            onClick={() => setIsSearchOpen(true)}
            className="text-muted-foreground"
        >
            <Search className="h-5 w-5" />
            <span className="sr-only">Search</span>
        </Button>
    );
}

return (
    <form onSubmit={handleSearch} className="flex items-center gap-2">
        <Input
            type="search"
            placeholder="Search wines..."
            value={searchQuery}
            onChange={(e) => setSearchQuery(e.target.value)}
            className="w-64"
            autoFocus
        />
        <Button type="submit" size="sm">
            <Search className="h-4 w-4" />
        </Button>
        <Button
            type="button"
            variant="ghost"
            size="sm"
            onClick={() => {
                setIsSearchOpen(false);
                setSearchQuery("");
            }}
        >
            <X className="h-4 w-4" />
        </Button>
    </form>
);
}
```

Navigation Search Features:

- **Expandable Interface:** Button expands to search form
- **Auto-focus:** Automatically focuses search input when opened
- **Form Handling:** Proper form submission and validation
- **Close Functionality:** Easy way to close search interface

User Menu

components/navbar/UserMenu.tsx

Purpose: User account dropdown menu **Location:** /components/navbar/UserMenu.tsx

```

"use client";

import { useAuth, useUser } from "@clerk/nextjs";
import { User, Settings, Logout, Heart, Package } from "lucide-react";
```



```
import { Button } from "@components/ui/button";
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from "@components/ui/dropdown-menu";
import { Avatar, AvatarFallback, AvatarImage } from "@components/ui/avatar";
import Link from "next/link";

export default function UserMenu() {
  const { isSignedIn, signOut } = useAuth();
  const { user } = useUser();

  if (!isSignedIn) {
    return (
      <div className="flex items-center gap-2">
        <Link href="/sign-in">
          <Button variant="ghost" size="sm">
            Sign in
          </Button>
        </Link>
        <Link href="/sign-up">
          <Button size="sm">Sign up</Button>
        </Link>
      </div>
    );
  }

  const handleSignOut = () => {
    signOut();
  };

  return (
    <DropdownMenu>
      <DropdownMenuTrigger asChild>
        <Button variant="ghost" className="relative h-8 w-8 rounded-full">
          <Avatar className="h-8 w-8">
            <AvatarImage src={user?.imageUrl} alt={user?.fullName || ""} />
            <AvatarFallback>
              {user?.firstName?.charAt(0) ||
                user?.emailAddresses[0]?.emailAddress?.charAt(0) ||
                "U"}
            </AvatarFallback>
          </Avatar>
        </Button>
      </DropdownMenuTrigger>
      <DropdownMenuContent className="w-56" align="end" forceMount>
        <DropdownMenuLabel className="font-normal">
          <div className="flex flex-col space-y-1">
            <p className="text-sm font-medium leading-none">
              {user?.fullName || "User"}
            </p>
            <p className="text-xs leading-none text-muted-foreground">
              {user?.emailAddresses[0]?.emailAddress}
            </p>
          </div>
        </DropdownMenuLabel>
        <DropdownMenuSeparator />
        <DropdownMenuItem asChild>
          <Link href="/profile" className="flex items-center">
            <User className="mr-2 h-4 w-4" />
            Profile
          </Link>
        </DropdownMenuItem>
      </DropdownMenuContent>
    </DropdownMenu>
  );
}
```

```

      </DropdownMenuItem>
      <DropdownMenuItem asChild>
        <Link href="/orders" className="flex items-center">
          <Package className="mr-2 h-4 w-4" />
          Orders
        </Link>
      </DropdownMenuItem>
      <DropdownMenuItem asChild>
        <Link href="/favorites" className="flex items-center">
          <Heart className="mr-2 h-4 w-4" />
          Favorites
        </Link>
      </DropdownMenuItem>
      <DropdownMenuItem asChild>
        <Link href="/settings" className="flex items-center">
          <Settings className="mr-2 h-4 w-4" />
          Settings
        </Link>
      </DropdownMenuItem>
      <DropdownMenuSeparator />
      <DropdownMenuItem onClick={handleSignOut} className="text-red-600">
        <LogOut className="mr-2 h-4 w-4" />
        Sign out
      </DropdownMenuItem>
    </DropdownMenuContent>
  </DropdownMenu>
);
}
```

User Menu Features:

- **Avatar Display:** Shows user profile picture or initials
- **User Information:** Displays name and email
- **Quick Links:** Direct access to user-specific pages
- **Sign Out:** Secure logout functionality
- **Conditional Rendering:** Shows different content for signed-in/out users

User Button Wrapper

components/navbar/UserButtonWrapper.tsx

Purpose: Wrapper component for Clerk's UserButton **Location:** </components/navbar/UserButtonWrapper.tsx>

```
"use client";

import { UserButton } from "@clerk/nextjs";

export default function UserButtonWrapper() {
  return (
    <UserButton
      appearance={{
        elements: {
          avatarBox: "h-8 w-8",
          userButtonPopoverCard: "shadow-lg border border-border",
        },
      }}
      afterSignOutUrl="/"
    >
  </UserButton>
);
}
```

User Button Wrapper Features:

- **Clerk Integration:** Uses Clerk's built-in user button
- **Custom Styling:** Consistent with app design
- **Post-logout Redirect:** Returns to home page after sign out
- **Responsive Design:** Adapts to different screen sizes

These navbar components provide a comprehensive navigation system for the Wine Store application. They handle:

- **User Authentication:** Sign in/out and user state
- **Navigation:** Main site navigation and mobile responsiveness
- **Shopping Features:** Cart and favorites access
- **Theme Switching:** Light/dark mode toggle
- **Search Functionality:** Product search interface
- **User Management:** Profile access and account settings