Getting Started
Introduction
Installation
components.json
Theming
Dark mode
CLI
Monorepo New
Next.js 15 + React 19
Typography
Open in v0
Figma
Changelog

Installation
Next.js
Vite
Remix
Astro
Laravel
Gatsby
Manual

Components
Accordion
Alert
Alert Dialog
Aspect Ratio
Avatar
Badge
Breadcrumb
Button
Calendar
Card
Carousel
Chart

Combobox

Autocomplete input and command palette with a list of suggestions.

Preview Code

Style: New York

Open in v0

Select framework...

Installation

The Combobox is built using a composition of the `<Popover />` and the `<Command />` components.

See installation instructions for the Popover and the Command components.

Usage

```
"use client"

import * as React from "react"
import { Check, ChevronsUpDown } from "lucide-react"

import { cn } from "@/lib/utils"
import { Button } from "@/components/ui/button"
import {
  Command,
  CommandEmpty,
  CommandGroup,
  CommandInput,
  CommandItem,
  CommandList,
} from "@/components/ui/command"
import {
  Popover,
  PopoverContent,
  PopoverTrigger,
} from "@/components/ui/popover"

const frameworks = [
  {
    value: "next.js",
    label: "Next.js",
  },
  {
    value: "sveltekit",
    label: "SvelteKit",
  },
  {
    value: "nuxt.js",
    label: "Nuxt.js",
  },
  {
    value: "remix",
    label: "Remix",
  },
  {
    value: "astro",
    label: "Astro",
  },
]

export function ComboboxDemo() {
  const [open, setOpen] = React.useState(false)
  const [value, setValue] = React.useState("")

  return (
    <Popover open={open} onOpenChange={setOpen}>
      <PopoverTrigger asChild>
        <Button
          variant="outline"
          role="combobox"
          aria-expanded={open}
          className="w-[200px] justify-between"
        >
          {value
            ? frameworks.find((framework) =>
framework.value === value)?.label
            : "Select framework...")
          <ChevronsUpDown className="ml-2 h-4 w-4 shrink-0
opacity-50" />
        </Button>
      </PopoverTrigger>
      <PopoverContent className="w-[200px] p-0">
        <Command>
          <CommandInput placeholder="Search framework...."
/>
          <CommandList>
            <CommandEmpty>No framework found.
</CommandEmpty>
            <CommandGroup>
              {frameworks.map((framework) => (
                <CommandItem
                  key={framework.value}
                  value={framework.value}
                  onSelect={(currentValue) => {
                    setValue(currentValue === value ? "" :
currentValue)
                    setOpen(false)
                  }}
                >
                  <Check
                    className={cn(
                      "mr-2 h-4 w-4",
                      value === framework.value ?
"opacity-100" : "opacity-0"
                    )}
                  />
                  {framework.label}
                </CommandItem>
              ))}
            </CommandGroup>
          </CommandList>
        </Command>
      </PopoverContent>
    </Popover>
  )
}
```

Examples

Combobox

Preview Code

Style: New York

Open in v0

Select framework...

Popover

Preview Code

Style: New York

Status + Set status

Dropdown menu

Preview Code

Style: New York

Open in v0

feature Create a new project ⌘⇧P

Responsive

You can create a responsive combobox by using the `<Popover />` on desktop and the `<Drawer />` components on mobile.

Preview Code

Style: New York

+ Set status

Form

Preview Code

Style: New York

Language

Select language

This is the language that will be used in the dashboard.

Submit

Built by shadcn. The source code is available on GitHub.

Created by BULYAN_KILIC_DAT