```
'use client';

import * as React from 'react';
import * as DropdownMenuPrimitive from '@radix-ui/react-dropdown-menu';
import { Check, ChevronRight, Circle } from 'lucide-react';

import { cn } from '@/lib/utils';

// Root component for the Dropdown Menu
const DropdownMenu = DropdownMenuPrimitive.Root;

// Trigger component for opening the Dropdown Menu
const DropdownMenuTrigger = DropdownMenuPrimitive.Trigger;

// Group component for grouping items within the Dropdown Menu
const DropdownMenuGroup = DropdownMenuPrimitive.Group;

// Portal component for rendering Dropdown Menu content
const DropdownMenuPortal = DropdownMenuPrimitive.Portal;

// Submenu component within the Dropdown Menu
const DropdownMenuSub = DropdownMenuPrimitive.Sub;

// Radio Group component for grouping radio items within the Dropdown Menu
const DropdownMenuRadioGroup = DropdownMenuPrimitive.RadioGroup;
```

```
interface DropdownMenuSubTriggerProps
  extends React.ComponentPropsWithoutRef<
    typeof DropdownMenuPrimitive.SubTrigger
  > {
  className?: string;
  inset?: boolean;
}

/**
 * DropdownMenuSubTrigger component that renders a trigger for a submenu.
 * It includes an optional ChevronRight icon to indicate the submenu.
 */
const DropdownMenuSubTrigger = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.SubTrigger>,
  DropdownMenuSubTriggerProps
>(({ className, inset, children, ...props }, ref) => (
  <DropdownMenuPrimitive.SubTrigger
    ref={ref}
    className={cn(
      'flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-accent data-[state=open]:bg-accent',
      inset && 'pl-8',
      className,
    )}
    {...props}
  >
```

```
    {children}
    <ChevronRight className="ml-auto h-4 w-4" />
  </DropdownMenuPrimitive.SubTrigger>
));
DropdownMenuSubTrigger.displayName =
  DropdownMenuPrimitive.SubTrigger.displayName;

interface DropdownMenuSubContentProps
  extends React.ComponentPropsWithoutRef<
    typeof DropdownMenuPrimitive.SubContent
  > {
  className?: string;
}

/**
 * DropdownMenuSubContent component that renders the content of a submenu.
 */
const DropdownMenuSubContent = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.SubContent>,
  DropdownMenuSubContentProps
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.SubContent
    ref={ref}
    className={cn(
      'z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground
shadow-lg data-[state=open]:animate-in data-[state=closed]:animate-out
data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95
data-[state=open]:zoom-in-95 data-[side=bottom]:slide-in-from-top-2
data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2
data-[side=top]:slide-in-from-bottom-2',
      className,
    )}
    {...props}
  />
));
DropdownMenuSubContent.displayName =
  DropdownMenuPrimitive.SubContent.displayName;

interface DropdownMenuContentProps
  extends React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Content> {
  className?: string;
  sideOffset?: number;
}

/**
 * DropdownMenuContent component that renders the main content of the Dropdown Menu.
 */
const DropdownMenuContent = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Content>,
  DropdownMenuContentProps
>(({ className, sideOffset = 4, ...props }, ref) => (
  <DropdownMenuPrimitive.Portal>
```

```
    <DropdownMenuPrimitive.Content
      ref={ref}
      sideOffset={sideOffset}
      className={cn(
        'z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-md data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2',
        className,
      )}
      {...props}
    />
  </DropdownMenuPrimitive.Portal>
));
DropdownMenuContent.displayName = DropdownMenuPrimitive.Content.displayName;

interface DropdownMenuItemProps
  extends React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Item> {
  className?: string;
  inset?: boolean;
}

/**
 * DropdownMenuItem component that renders an individual item in the Dropdown Menu.
 */
const DropdownMenuItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Item>,
  DropdownMenuItemProps
>(({ className, inset, ...props }, ref) => (
  <DropdownMenuPrimitive.Item
    ref={ref}
    className={cn(
      'relative flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm
outline-none transition-colors focus:bg-accent focus:text-accent-foreground
data-[disabled]:pointer-events-none data-[disabled]:opacity-50',
      inset && 'pl-8',
      className,
    )}
    {...props}
  />
));
DropdownMenuItem.displayName = DropdownMenuPrimitive.Item.displayName;

interface DropdownMenuCheckboxItemProps
  extends React.ComponentPropsWithoutRef<
    typeof DropdownMenuPrimitive.CheckboxItem
  > {
  className?: string;
  checked?: boolean;
}
```

```
/**
 * DropdownMenuCheckboxItem component that renders a checkbox item in the Dropdown Menu.
 */
const DropdownMenuCheckboxItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.CheckboxItem>,
  DropdownMenuCheckboxItemProps
>(({ className, children, checked, ...props }, ref) => (
  <DropdownMenuPrimitive.CheckboxItem
    ref={ref}
    className={cn(
      'relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none transition-colors focus:bg-accent focus:text-accent-foreground data-[disabled]:pointer-events-none data-[disabled]:opacity-50',
      className,
    )}
    checked={checked}
    {...props}
  >
    <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
      <DropdownMenuPrimitive.ItemIndicator>
        <Check className="h-4 w-4" />
      </DropdownMenuPrimitive.ItemIndicator>
    </span>
    {children}
  </DropdownMenuPrimitive.CheckboxItem>
```

```
));
DropdownMenuCheckboxItem.displayName =
  DropdownMenuPrimitive.CheckboxItem.displayName;

interface DropdownMenuRadioItemProps
  extends React.ComponentPropsWithoutRef<
    typeof DropdownMenuPrimitive.RadioItem
  > {
  className?: string;
}

/**
 * DropdownMenuRadioItem component that renders a radio button item in the Dropdown Menu.
 */
const DropdownMenuRadioItem = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.RadioItem>,
  DropdownMenuRadioItemProps
>(({ className, children, ...props }, ref) => (
  <DropdownMenuPrimitive.RadioItem
    ref={ref}
    className={cn(
      'relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none transition-colors focus:bg-accent focus:text-accent-foreground data-[disabled]:pointer-events-none data-[disabled]:opacity-50',
      className,
    )}
    {...props}
  >
    <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
      <DropdownMenuPrimitive.ItemIndicator>
        <Circle className="h-2 w-2 fill-current" />
      </DropdownMenuPrimitive.ItemIndicator>
    </span>
    {children}
  </DropdownMenuPrimitive.RadioItem>
));
DropdownMenuRadioItem.displayName = DropdownMenuPrimitive.RadioItem.displayName;

interface DropdownMenuLabelProps
  extends React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label> {
  className?: string;
  inset?: boolean;
}

/**
 * DropdownMenuLabel component that renders a label within the Dropdown Menu.
 */
const DropdownMenuLabel = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Label>,
  DropdownMenuLabelProps
>(({ className, inset, ...props }, ref) => (
  <DropdownMenuPrimitive.Label
```

```
    ref={ref}
    className={cn(
      'px-2 py-1.5 text-sm font-semibold',
      inset && 'pl-8',
      className,
    )}
    {...props}
  />
));
DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName;

interface DropdownMenuSeparatorProps
  extends React.ComponentPropsWithoutRef<
    typeof DropdownMenuPrimitive.Separator
  > {
  className?: string;
}

/**
  * DropdownMenuSeparator component that renders a separator (horizontal line) within the Dropdown Menu.
 */
const DropdownMenuSeparator = React.forwardRef<
  React.ElementRef<typeof DropdownMenuPrimitive.Separator>,
  DropdownMenuSeparatorProps
>(({ className, ...props }, ref) => (
  <DropdownMenuPrimitive.Separator
    ref={ref}
    className={cn('-mx-1 my-1 h-px bg-muted', className)}
    {...props}
  />
));
DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName;

interface DropdownMenuShortcutProps
  extends React.HTMLAttributes<HTMLSpanElement> {
  className?: string;
}

/**
  * DropdownMenuShortcut component that renders a shortcut (typically for keyboard shortcuts) within the Dropdown Menu.
 */
const DropdownMenuShortcut: React.FC<DropdownMenuShortcutProps> = ({
  className,
  ...props
}) => {
  return (
    <span
      className={cn('ml-auto text-xs tracking-widest opacity-60', className)}
      {...props}
    />
  );
};
DropdownMenuShortcut.displayName = 'DropdownMenuShortcut';

export {
  DropdownMenu,
  DropdownMenuTrigger,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuCheckboxItem,
  DropdownMenuRadioItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuShortcut,
  DropdownMenuGroup,
  DropdownMenuPortal,
  DropdownMenuSub,
  DropdownMenuSubContent,
  DropdownMenuSubTrigger,
  DropdownMenuRadioGroup,
};
```