

```
'use client';
```

```
import * as React from 'react';
```

```
import * as DialogPrimitive from '@radix-ui/react-dialog';
```

```
import { X } from 'lucide-react';
```

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

```
// Root Dialog component
```

```
const Dialog = DialogPrimitive.Root;
```

```
// Trigger component for opening the dialog
```

```
const DialogTrigger = DialogPrimitive.Trigger;
```

```
// Portal component for rendering dialog content
```

```
const DialogPortal = DialogPrimitive.Portal;
```

```
// Component to close the dialog
```

```
const DialogClose = DialogPrimitive.Close;
```

```
interface DialogOverlayProps
```

```
  extends React.ComponentPropsWithoutRef<typeof DialogPrimitive.Overlay> {
```

```
  className?: string;
```

```
}
```

```
/**
```

\* DialogOverlay component that renders a semi-transparent background overlay.

\* It appears behind the dialog content and can be used to focus attention on the dialog.

\*/

```
const DialogOverlay = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Overlay>,
  DialogOverlayProps
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Overlay
    ref={ref}
    className={cn(
      'fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out
data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0',
      className,
    )}
    {...props}
  />
));

DialogOverlay.displayName = DialogPrimitive.Overlay.displayName;
```

```
interface DialogContentProps
  extends React.ComponentPropsWithoutRef<typeof DialogPrimitive.Content> {
  className?: string;
  children: React.ReactNode;
}

/**
```

\* DialogContent component that renders the main content of the dialog.

\* It includes a close button and manages animations for opening and closing the dialog.

\*/

```
const DialogContent = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Content>,
  DialogContentProps
>(({ className, children, ...props }, ref) => (
  <DialogPortal>
    <DialogOverlay />
    <DialogPrimitive.Content
      ref={ref}
      className={cn(
        'fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4
border      bg-background      p-6      shadow-lg      duration-200      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-[state=closed]:slide-out-to-left-1/2      data-[state=closed]:slide-out-to-top-[48%]
data-[state=open]:slide-in-from-left-1/2 data-[state=open]:slide-in-from-top-[48%] sm:rounded-lg',
        className,
      )}
      {...props}
    >
      {children}

      <DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70
ring-offset-background transition-opacity hover:opacity-100 focus:outline-none focus:ring-2
focus:ring-ring focus:ring-offset-0 disabled:pointer-events-none data-[state=open]:bg-accent
```

```
data-[state=open]:text-muted-foreground">
```

```
  <X className="h-4 w-4" />
```

```
  <span className="sr-only">Close</span>
```

```
</DialogPrimitive.Close>
```

```
</DialogPrimitive.Content>
```

```
</DialogPortal>
```

```
));
```

```
DialogContent.displayName = DialogPrimitive.Content.displayName;
```

```
interface DialogHeaderProps extends React.HTMLAttributes<HTMLDivElement> {
```

```
  className?: string;
```

```
}
```

```
/**
```

```
 * DialogHeader component that renders the header section of the dialog.
```

```
 * Typically used to display the title and optional description.
```

```
*/
```

```
const DialogHeader: React.FC<DialogHeaderProps> = ({ className, ...props }) => (
```

```
  <div
```

```
    className={cn(
```

```
      'flex flex-col space-y-1.5 text-center sm:text-left',
```

```
      className,
```

```
    )}
```

```
    {...props}
```

```
  />
```

```
);
```

```
DialogHeader.displayName = 'DialogHeader';
```

```
interface DialogFooterProps extends React.HTMLAttributes<HTMLDivElement> {  
  className?: string;  
}
```

```
/**  
 * DialogFooter component that renders the footer section of the dialog.  
 * Typically used to display action buttons or additional content.  
 */  
const DialogFooter: React.FC<DialogFooterProps> = ({ className, ...props }) => (  
  <div  
    className={cn(  
      'flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2',  
      className,  
    )}  
    {...props}  
  />  
);
```

```
DialogFooter.displayName = 'DialogFooter';
```

```
interface DialogTitleProps  
  extends React.ComponentPropsWithoutRef<typeof DialogPrimitive.Title> {  
  className?: string;  
}
```

```

/**
 * DialogTitle component that renders the title of the dialog.
 * Typically used within the DialogHeader component.
 */

const DialogTitle = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Title>,
  DialogTitleProps
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Title
    ref={ref}
    className={cn(
      'text-lg font-semibold leading-none tracking-tight',
      className,
    )}
    {...props}
  />
));

DialogTitle.displayName = DialogPrimitive.Title.displayName;

interface DialogDescriptionProps
  extends React.ComponentPropsWithoutRef<typeof DialogPrimitive.Description> {
  className?: string;
}

/**
 * DialogDescription component that renders the description of the dialog.

```

\* Typically used within the DialogHeader component to provide additional context or information.

\*/

```
const DialogDescription = React.forwardRef<
  React.ElementRef<typeof DialogPrimitive.Description>,
  DialogDescriptionProps
>(({ className, ...props }, ref) => (
  <DialogPrimitive.Description
    ref={ref}
    className={cn('text-sm text-muted-foreground', className)}
    {...props}
  />
));

DialogDescription.displayName = DialogPrimitive.Description.displayName;
```

```
export {
  Dialog,
  DialogPortal,
  DialogOverlay,
  DialogClose,
  DialogTrigger,
  DialogContent,
  DialogHeader,
  DialogFooter,
  DialogTitle,
  DialogDescription,
};
```