

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

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

```
interface TableProps extends React.TableHTMLAttributes<HTMLTableElement> {  
  className?: string;  
}
```

```
interface TableSectionProps  
  extends React.HTMLAttributes<HTMLTableSectionElement> {  
  className?: string;  
}
```

```
interface TableRowProps extends React.HTMLAttributes<HTMLTableRowElement> {  
  className?: string;  
}
```

```
interface TableCellProps extends React.TdHTMLAttributes<HTMLTableCellElement> {  
  className?: string;  
}
```

```
interface TableCaptionProps  
  extends React.HTMLAttributes<HTMLTableCaptionElement> {  
  className?: string;  
}
```

```
interface TableHeadProps extends React.ThHTMLAttributes<HTMLTableCellElement> {
```

```

    className?: string;
  }

/**
 * Table component that renders a styled HTML table with overflow handling.
 *
 * @param className - Additional class names to apply to the table.
 * @param props - All other props supported by a standard HTML table element.
 * @param ref - A ref object to reference the underlying table element.
 */
const Table = React.forwardRef<HTMLTableElement, TableProps>(
  ({ className, ...props }, ref) => (
    <div className="relative w-full overflow-auto">
      <table
        ref={ref}
        className={cn('w-full caption-bottom text-sm', className)}
        {...props}
      />
    </div>
  ),
);

Table.displayName = 'Table';

/**
 * TableHeader component that renders a styled HTML thead element.
 *

```

* @param className - Additional class names to apply to the thead.

* @param props - All other props supported by a standard HTML thead element.

* @param ref - A ref object to reference the underlying thead element.

*/

```
const TableHeader = React.forwardRef<
  HTMLTableSectionElement,
  TableSectionProps
>(({ className, ...props }, ref) => (
  <thead ref={ref} className={cn(['&_tr:border-b', className])} {...props} />
));

TableHeader.displayName = 'TableHeader';
```

```
/**
 * TableBody component that renders a styled HTML tbody element.
 *
 * @param className - Additional class names to apply to the tbody.
 * @param props - All other props supported by a standard HTML tbody element.
 * @param ref - A ref object to reference the underlying tbody element.
 */
```

```
const TableBody = React.forwardRef<HTMLTableSectionElement, TableSectionProps>(
  ({ className, ...props }, ref) => (
    <tbody
      ref={ref}
      className={cn(['&_tr:last-child:border-0', className])}
      {...props}
    />
```

```

    ),
  );

  TableBody.displayName = 'TableBody';

  /**
   * TableFooter component that renders a styled HTML tfoot element.
   *
   * @param className - Additional class names to apply to the tfoot.
   * @param props - All other props supported by a standard HTML tfoot element.
   * @param ref - A ref object to reference the underlying tfoot element.
   */
  const TableFooter = React.forwardRef<
    HTMLTableSectionElement,
    TableSectionProps
  >(({ className, ...props }, ref) => (
    <tfoot
      ref={ref}
      className={cn(
        'border-t bg-muted/50 font-medium [&>tr]:last:border-b-0',
        className,
      )}
      {...props}
    />
  ));

  TableFooter.displayName = 'TableFooter';

```

```

/**
 * TableRow component that renders a styled HTML tr element.
 *
 * @param className - Additional class names to apply to the tr.
 * @param props - All other props supported by a standard HTML tr element.
 * @param ref - A ref object to reference the underlying tr element.
 */
const TableRow = React.forwardRef<HTMLTableRowElement, TableRowProps>(
  ({ className, ...props }, ref) => (
    <tr
      ref={ref}
      className={cn(
        'border-b transition-colors hover:bg-muted/50 data-[state=selected]:bg-muted',
        className,
      )}
      {...props}
    />
  ),
);
TableRow.displayName = 'TableRow';

```

```

/**
 * TableHead component that renders a styled HTML th element.
 *
 * @param className - Additional class names to apply to the th.
 * @param props - All other props supported by a standard HTML th element.

```

* @param ref - A ref object to reference the underlying th element.

*/

```
const TableHead = React.forwardRef<HTMLTableCellElement, TableHeadProps>(
  ({ className, ...props }, ref) => (
    <th
      ref={ref}
      className={cn(
        'h-12 px-4 text-left align-middle font-medium text-muted-foreground
        [&:has([role=checkbox])]:pr-0',
        className,
      )}
      {...props}
    />
  ),
);

TableHead.displayName = 'TableHead';
```

/**

* TableCell component that renders a styled HTML td element.

*

* @param className - Additional class names to apply to the td.

* @param props - All other props supported by a standard HTML td element.

* @param ref - A ref object to reference the underlying td element.

*/

```
const TableCell = React.forwardRef<HTMLTableCellElement, TableCellProps>(
  ({ className, ...props }, ref) => (
```

```

    <td
      ref={ref}
      className={cn(
        'p-4 align-middle [&:has([role=checkbox])]:pr-0 lg:relative lg:h-[150px]',
        className,
      )}
      {...props}
    />
  ),
);

TableCell.displayName = 'TableCell';

/**
 * TableCell component that renders a styled HTML caption element.
 *
 * @param className - Additional class names to apply to the caption.
 * @param props - All other props supported by a standard HTML caption element.
 * @param ref - A ref object to reference the underlying caption element.
 */
const TableCell = React.forwardRef<
  HTMLTableCaptionElement,
  TableCellProps
>(({ className, ...props }, ref) => (
  <caption
    ref={ref}
    className={cn('mt-4 text-sm text-muted-foreground', className)}

```

```
    {...props}  
  />  
));  
TableCaption.displayName = 'TableCaption';
```

```
export {  
  Table,  
  TableHeader,  
  TableBody,  
  TableFooter,  
  TableHead,  
  TableRow,  
  TableCell,  
  TableCaption,  
};
```