

```
import React, { useMemo } from 'react';
```

```
import {
```

```
  ColumnDef,
```

```
  flexRender,
```

```
  getCoreRowModel,
```

```
  useReactTable,
```

```
} from '@tanstack/react-table';
```

```
import {
```

```
  Table,
```

```
  TableBody,
```

```
  TableCell,
```

```
  TableHead,
```

```
  TableHeader,
```

```
  TableRow,
```

```
} from '@shared/components/ui/table';
```

```
export interface RequirementProps {
```

```
  package: string;
```

```
  installation: string;
```

```
}
```

```
export default function AgentRequirements({
```

```
  requirements,
```

```
}: {
```

```
  requirements: RequirementProps[];
```

```
}) {
```

```
const columns = useMemo(() => {  
  return [  
    {  
      accessorKey: 'package',  
      header: 'Package',  
    },  
    {  
      accessorKey: 'installation',  
      header: 'Installation',  
    },  
  ];  
}, []);
```

```
const table = useReactTable({  
  data: requirements,  
  columns: columns as any,  
  getCoreRowModel: getCoreRowModel(),  
});
```

```
return (  
  <div className="my-6">  
    <h2 className="text-4xl mb-2">Requirements</h2>  
    <Table>  
      <TableHeader>  
        {table  
          ?.getHeaderGroups()        }  
      </TableHeader>  
    </Table>  
  </div>  
)
```

```
?.map((headerGroup) => (
  <TableRow key={headerGroup?.id}>
    {headerGroup?.headers?.map((header) => (
      <TableHead key={header.id}>
        {header?.isPlaceholder
          ? null
          : flexRender(
              header?.column?.columnDef?.header,
              header?.getContext(),
            )}
      </TableHead>
    ))}
  </TableRow>
)})
```

```
</TableHeader>
```

```
<TableBody>
```

```
{table?.getRowModel()?.rows?.length ? (
  table.getRowModel().rows?.map((row, i) => (
    <TableRow key={i} data-state={row?.getIsSelected() && 'selected'}>
      {row
        ?.getVisibleCells()
        ?.map((cell, i) => (
          <TableCell key={i}>
            {flexRender(
              cell?.column?.columnDef?.cell,
              cell?.getContext(),
            )}
          </TableCell>
        ))}
    </TableRow>
  )}
)}
```

```
    }}
```

```
  </TableCell>
```

```
  )}}
```

```
</TableRow>
```

```
)
```

```
): (
```

```
<TableRow>
```

```
  <TableCell colSpan={columns?.length} className="h-24 text-center">
```

```
    No results.
```

```
  </TableCell>
```

```
</TableRow>
```

```
  )}
```

```
</TableBody>
```

```
</Table>
```

```
</div>
```

```
);
```

```
}
```