



**Fig:- ZToolTip Core Component**

### **Code:-**

```
import React, { useState, useEffect, useRef } from "react";
import "./ToolTip.css";

interface SimpleTooltipProps {
  children: React.ReactNode;
  label: string;
  className?: string;
  duration?: number;
}

const SimpleTooltip: React.FC<SimpleTooltipProps> = ({
  children,
  label,
  className,
  duration = 6000,
}) => {
  const [visible, setVisible] = useState(false);
  const timerRef = useRef<NodeJS.Timeout | null>(null);

  const showTooltip = () => {
    setVisible(true);
    if (timerRef.current) {
      clearTimeout(timerRef.current);
    }
    timerRef.current = setTimeout(() => {
      setVisible(false);
    }, duration);
  };
};
```

```

const hideTooltip = () => {
  if (timerRef.current) {
    clearTimeout(timerRef.current);
  }
  setVisible(false);
};

useEffect(() => {
  return () => {
    if (timerRef.current) {
      clearTimeout(timerRef.current);
    }
  };
}, []);

return (
  <div
    className="tooltip"
    onMouseEnter={showTooltip}
    onMouseLeave={hideTooltip}
  >
    {children}
    {visible && <span className={`tooltiptext ${className}`}>{label}</span>}
  </div>
);
};

export default SimpleTooltip;

```

## CSS

```

.tooltip {
  position: relative;
}

.tooltip .tooltiptext {
  visibility: hidden;
}

```

```
width: 100%;
background-color: black;
color: white;
text-align: center;
border-radius: 6px;
padding: 5px 10px;
top: 50px;
left: 25%;
position: absolute;
z-index: 1;
transform: translate(10% , -50%)
}
```

```
.tooltip:hover .tooltiptext {
  visibility: visible;
}
```

```
      <SimpleTooltip
        key='1'
        label='DropDown'
        className="dropdown-tooltip"
      >

        <OrganisationStatus handleSetSelectedItems={handleSetSelectedItems}
      />

    </SimpleTooltip>
```

## Props Required For Tooltip

- 1) ClassName
- 2) key

### 3) label

How to use this component

First Render a Component Wherever it is required then add the following props that implemented in above code where through prop classname we can add our custom css and modify the ZToolTip....

label is prop that gives a name to a table...

Key is used to identify how many zToolTip we Have Used or otherwise we can use it in the map method....

possible ways to improve this component

We Can Modify our ToolTip through passing Props Such as ClassName as per Requirement.....