```
import { UserUsage } from '@/shared/utils/api/usage';
import { getColorForModel } from '../helpers/get-color-model';

interface CustomTooltipProps {
  active: boolean;
  payload: any[];
  label: string;
  usageData: UserUsage | null;
  type?: 'cost' | 'activity';
}

export default function CustomTooltip({
  active,
  payload,
  label,
  usageData,
  type,
}: CustomTooltipProps) {
  const row = payload?.[0]?.payload;
  const modelValue = row?.totalCost || row?.value;

  if (!modelValue) return null;

  if (active && row) {
    return (
      <ul className="bg-white p-4 rounded-lg shadow-md text-xs">
```

```
<li className="flex justify-between font-semibold items-center gap-2 text-gray-800 mb-1">
  <span>{label}</span>
  {type !== 'activity' && <span>${modelValue.toFixed(3)}</span>}
</li>
{Object.entries(row).map(([modelName, cost]) => {
  if (modelName !== 'date' && modelName !== 'totalCost') {
    const value =
      type === 'activity'
        ? Number(cost).toLocaleString()
        : `$${parseFloat(cost as string).toFixed(3)}`;
    return (
      <li
        key={modelName}
        className="flex justify-between items-center gap-2"
      >
        <span
          className="text-gray-800"
          style={{ color: getColorForModel(modelName, usageData) }}
        >
          {modelName.trim()}
        </span>
        <span
          className="text-gray-800"
          style={{ color: getColorForModel(modelName, usageData) }}
        >
          {value}
```

```
                </span>
              </li>
            );
          }
          return null;
        })}
      </ul>
    );
  }
  return null;
}
```