

```
import { Button } from '@shared/components/ui/Button';

import { UserUsage } from '@shared/utls/api/usage';

import React from 'react';

import { PieChart, Pie, Cell, ResponsiveContainer, Tooltip } from 'recharts';

import { getMonthRange } from '../../helpers/get-month-range';
```

```
const COLORS = ['#4BC0C0', '#2D2D2D'];
```

```
const renderCustomizedLabel = ({
  cx,
  cy,
  midAngle,
  innerRadius,
  outerRadius,
  percent,
}: Record<string, number>) => {
  const radius = innerRadius + (outerRadius - innerRadius) * 0.5;
  const x = cx + radius * Math.cos(-midAngle * (Math.PI / 180));
  const y = cy + radius * Math.sin(-midAngle * (Math.PI / 180));

  return (
    <text
      x={cx}
      y={cy}
      fill="white"
      textAnchor="middle"
```

```

    dominantBaseline="central"

    className="font-semibold percentage-label"

    >

    {`${(percent * 100).toFixed(0)}%`}

</text>

);

};

const CustomTooltip = ({

    active,

    payload,

}): {

    active: boolean;

    payload: { name: string; value: number }[];

}) => {

    if (active && payload && payload.length) {

        return (

            <div className="bg-secondary text-foreground p-2 border border-gray-500">

                <p className="text-xs">`${payload[0].name} : ${payload[0].value.toFixed(2)}%`</p>

            </div>

        );

    }

    return null;

};

```

```
const MonthlySpendPieChart = ({
  usageData,
  month,
}: {
  usageData: UserUsage | null;
  month: Date;
}) => {
  const monthlyRange = getMonthRange(month);

  const totalAPICost = usageData ? usageData.totalCost : 0;
  const limit = 200;
  const percentUsed = (totalAPICost / limit) * 100;

  const chartData = [
    { name: 'Used', value: parseFloat(totalAPICost.toFixed(3)) },
    { name: 'Remaining', value: parseFloat((limit - totalAPICost).toFixed(3)) },
  ];

  return (
    <div className="max-xl:mb-6">
      <div className="flex items-center gap-3 text-sm mb-4">
        <span className="text-gray-400">Monthly Usage</span>{' '}
        <span>{monthlyRange}</span>
      </div>
      <div className="flex items-center w-full">
        <ResponsiveContainer width="50%" height={150}>
```

```

<PieChart className="pie-chart">

  <Pie

    data={chartData}

    cx="50%"

    cy="50%"

    innerRadius={50}

    outerRadius={70}

    labelLine={false}

    label={renderCustomizedLabel}

    dataKey="value"

  >

    {chartData.map((entry, index) => (

      <Cell

        key={`cell-${index}`}

        fill={COLORS[index % COLORS.length]}

        opacity={index === 0 ? 1 : 0.5}

      />

    )))}

  </Pie>

  <Tooltip content={<CustomTooltip active={true} payload={[]} />} />

</PieChart>

</ResponsiveContainer>

<div>

  <p className="text-foreground text-xl">${totalAPICost.toFixed(2)}</p>

  <p style={{ color: 'gray' }}>/ ${limit.toFixed(2)} limit</p>

```

```
<Button className="mt-2 bg-teal-900 disabled:opacity-100" disabled>
```

```
<span className="text-sm">{percentUsed.toFixed(1)}% used</span>
```

```
</Button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
);
```

```
};
```

```
export default MonthlySpendPieChart;
```