```
import React, { useMemo } from 'react';
import {
  BarChart,
  Bar,
  XAxis,
  YAxis,
  Tooltip,
  Legend,
  ResponsiveContainer,
} from 'recharts';
import { UserUsage } from '@/shared/utils/api/usage';
import CustomTooltip from '../tooltip';
import { getColorForModel } from '../../helpers/get-color-model';

export default function MonthlySpend({
  usageData,
}: {
  usageData: UserUsage | null;
}) {
  if (!usageData || !usageData?.dailyCosts?.length) {
    return (
      <div className="flex flex-col items-center justify-center border rounded-md px-2 sm:px-4 py-4 sm:py-8 text-card-foreground mt-5 mb-8 gap-2 w-full">
        <h3 className="opacity-60">No monthly usage data available</h3>
      </div>
    );
```

```
}

const daysInMonth = useMemo(() => {
  const firstDate = new Date(usageData.dailyCosts[0].date);
  const year = firstDate.getFullYear();
  const month = firstDate.getMonth();
  return new Date(year, month + 1, 0).getDate();
}, [usageData]);

const chartData = [];
for (let day = 1; day <= daysInMonth; day++) {
  const date = new Date(usageData.dailyCosts[0].date);
  date.setDate(day);
  const dateString = date.toISOString().slice(0, 10);

  const dailyCost = usageData?.dailyCosts.find(
    (cost) => cost.date === dateString,
  );

  const modelCosts: { [key: string]: number } = {};
  if (dailyCost && dailyCost.model) {
    for (const [model, obj] of Object.entries(dailyCost.model)) {
      modelCosts[model.trim()] = obj.costs;
    }
  } else {
    for (const model of Object.keys(
      (usageData as UserUsage)?.dailyCosts?.[0]?.model || {},
    )) {
      modelCosts[model.trim()] = 0;
    }
  }

  chartData.push({
    date: new Date(dateString).toLocaleDateString('en-US', {
      day: '2-digit',
      month: 'short',
    }),
    totalCost: dailyCost?.totalCost,
    ...modelCosts,
  });
}

const allModels: string[] = [];
usageData.dailyCosts.forEach((dailyCost) => {
  if (dailyCost.model) {
    Object.keys(dailyCost.model).forEach((modelName) => {
      allModels.push(modelName.trim());
    });
  }
});

return (
<div className="chart mt-5">
  <ResponsiveContainer width="100%" height={400}>
    <BarChart data={chartData} barSize={20}>
      <XAxis dataKey="date" minTickGap={50} tick={{ fontSize: 12 }} />
      <YAxis
        tickFormatter={(value) => `$${value.toFixed(2)}`}
        tick={{ fontSize: 12 }}
      />
      <Tooltip
        cursor={{ opacity: 0.1, fill: 'white' }}
        content={
          <CustomTooltip
            active={false}
            payload={[]}
            label={''}
            usageData={usageData}
          />
        }
        contentStyle={{
          backgroundColor: 'rgba(255, 255, 255, 0.8)',
          border: '1px solid #e5e7eb',
          color: '#374151',
          fontSize: '12px',
        }}
      />
      {allModels?.map((modelName) => {
            return (
              <Bar
                key={modelName}
                dataKey={modelName}
                stackId="a"
                fill={getColorForModel(modelName, usageData)}
                radius={[2, 2, 0, 0]}
              />
            );
          })}
        </BarChart>
      </ResponsiveContainer>
    </div>
  );
}
```