

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

import {
  BarChart,
  Bar,
  XAxis,
  YAxis,
  Tooltip,
  ResponsiveContainer,
  Legend,
} from 'recharts';

import CustomTooltip from '../tooltip';

import { getColorForModel } from '../../helpers/get-color-model';

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

```
export default function ModelUsage({
  usageData,
}: {
  usageData: UserUsage | null;
}) {
  if (!usageData || !usageData?.dailyCosts?.length) {
    return null;
  }

  const month = usageData?.dailyCosts[0]?.date
    ? new Date(usageData.dailyCosts[0].date).getMonth()
    : null;
```

```
const year = usageData?.dailyCosts[0]?.date
  ? new Date(usageData.dailyCosts[0].date).getFullYear()
  : null;
```

```
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]);
```

```
// Create an object to store costs per model
```

```
const modelData: any = {};
```

```
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,
  );
```

```
  if (dailyCost && dailyCost.model) {
    for (const [model, obj] of Object.entries(dailyCost.model)) {
      if (!modelData[model]) modelData[model] = [];
      modelData[model].push({
```

```

    date: new Date(dateString).toLocaleDateString('en-US', {
      day: '2-digit',
      month: 'short',
    }),
    totalCost: obj.costs,
  });
}
} else {
  for (const model of Object.keys(modelData)) {
    modelData[model].push({
      date: new Date(dateString).toLocaleDateString('en-US', {
        day: '2-digit',
        month: 'short',
      }),
      totalCost: 0,
    });
  }
}
}

return (
  <div className="model-charts mt-16 xl:mb-10 grid md:grid-cols-2">
    {Object.keys(modelData).map((modelName) => (
      <div key={modelName} className="model-chart mb-5">
        <h4 className="mb-5">{modelName}</h4>
        <ResponsiveContainer width="100%" height={250}>

```

```

<BarChart data={modelData[modelName]} barSize={15}>

  <XAxis dataKey="date" tick={{ fontSize: 12 }} minTickGap={50} />

  <YAxis

    tickFormatter={(value) => `$$${value.toFixed(2)}}`

    tick={{ fontSize: 12 }}

  />

  <Tooltip

    cursor={{ opacity: 0.1, fill: 'white' }}

    content={

      <CustomTooltip

        active={true}

        payload={[]}

        label={}

        usageData={usageData}

      />

    }

    contentStyle={{

      backgroundColor: 'rgba(255, 255, 255, 0.8)',

      border: '1px solid #e5e7eb',

      color: '#374151',

      fontSize: '12px',

    }}

  />

  <Bar

    dataKey="totalCost"

```

```
fill={getColorForModel(modelName, usageData)}
```

```
radius={[2, 2, 0, 0]}
```

```
/>
```

```
</BarChart>
```

```
</ResponsiveContainer>
```

```
</div>
```

```
)))}
```

```
</div>
```

```
);
```

```
}
```