

## Contents

<b>SECTION 28: ANALYTICS INTEGRATION (v3.6.0)</b>	<b>1</b>
	1
28.1 Analytics Dashboard Extensions . . . . .	1
	5

## SECTION 28: ANALYTICS INTEGRATION (v3.6.0)

### 28.1 Analytics Dashboard Extensions

```
// apps/admin-dashboard/src/app/admin/analytics/page.tsx

'use client';

import React, { useState } from 'react';
import { useQuery } from '@tanstack/react-query';
import { Card, CardContent, CardHeader, CardTitle } from '@/components/ui/card';
import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';
import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from '@/components/ui/select';
import { LineChart, Line, BarChart, Bar, XAxis, YAxis, CartesianGrid, Tooltip, ResponsiveContainer } from 'recharts';
import { BarChart3, TrendingUp, Users, DollarSign, Cpu, Clock } from 'lucide-react';

export default function AnalyticsPage() {
  const [timeRange, setTimeRange] = useState('7d');
  const [metricType, setMetricType] = useState('usage');

  const { data: metrics } = useQuery({
    queryKey: ['analytics', timeRange, metricType],
    queryFn: async () => {
      const res = await fetch(`api/admin/analytics?range=${timeRange}&type=${metricType}`);
      return res.json();
    }
  });

  const { data: modelStats } = useQuery({
    queryKey: ['model-stats', timeRange],
    queryFn: async () => {
      const res = await fetch(`api/admin/analytics/models?range=${timeRange}`);
      return res.json();
    }
});

const COLORS = ['#0088FE', '#00C49F', '#FFBB28', '#FF8042', '#8884D8'];
```

```

return (
  <div className="space-y-6">
    <div className="flex justify-between items-center">
      <h1 className="text-3xl font-bold">Analytics</h1>
      <div className="flex gap-4">
        <Select value={timeRange} onChange={setTimeRange}>
          <SelectTrigger className="w-32">
            <SelectValue />
          </SelectTrigger>
          <SelectContent>
            <SelectItem value="24h">Last 24h</SelectItem>
            <SelectItem value="7d">Last 7 days</SelectItem>
            <SelectItem value="30d">Last 30 days</SelectItem>
            <SelectItem value="90d">Last 90 days</SelectItem>
          </SelectContent>
        </Select>
      </div>
    </div>

    { /* Summary Cards */ }
    <div className="grid grid-cols-4 gap-4">
      <Card>
        <CardContent className="pt-6">
          <div className="flex items-center justify-between">
            <div>
              <p className="text-sm text-gray-500">Total Requests</p>
              <p className="text-2xl font-bold">{metrics?.totalRequests?.toLocaleString()}</p>
              <p className="text-sm text-green-500">+{metrics?.requestsGrowth}</p>
            </div>
            <BarChart3 className="h-8 w-8 text-blue-500" />
          </div>
        </CardContent>
      </Card>

      <Card>
        <CardContent className="pt-6">
          <div className="flex items-center justify-between">
            <div>
              <p className="text-sm text-gray-500">Total Tokens</p>
              <p className="text-2xl font-bold">{((metrics?.totalTokens / 1000).toFixed(1))}</p>
              <p className="text-sm text-green-500">+{metrics?.tokensGrowth}</p>
            </div>
            <Cpu className="h-8 w-8 text-purple-500" />
          </div>
        </CardContent>
      </Card>
    </div>
  </div>
)

```

```

<Card>
  <CardContent className="pt-6">
    <div className="flex items-center justify-between">
      <div>
        <p className="text-sm text-gray-500">Total Cost</p>
        <p className="text-2xl font-bold">${metrics?.totalCost?.toFixed(2)}</p>
        <p className="text-sm text-green-500">+{metrics?.costGrowth}%
      </div>
      <DollarSign className="h-8 w-8 text-green-500" />
    </div>
  </CardContent>
</Card>

<Card>
  <CardContent className="pt-6">
    <div className="flex items-center justify-between">
      <div>
        <p className="text-sm text-gray-500">Avg Latency</p>
        <p className="text-2xl font-bold">{metrics?.avgLatency}ms</p>
        <p className="text-sm text-green-500">-{metrics?.latencyImprovement}</p>
      </div>
      <Clock className="h-8 w-8 text-orange-500" />
    </div>
  </CardContent>
</Card>
</div>

{/* Charts */}
<div className="grid grid-cols-2 gap-6">
  <Card>
    <CardHeader>
      <CardTitle>Usage Over Time</CardTitle>
    </CardHeader>
    <CardContent>
      <ResponsiveContainer width="100%" height={300}>
        <LineChart data={metrics?.usageTimeSeries || []}>
          <CartesianGrid strokeDasharray="3 3" />
          <XAxis dataKey="date" />
          <YAxis />
          <Tooltip />
          <Line type="monotone" dataKey="requests" stroke="#0088FE" />
          <Line type="monotone" dataKey="tokens" stroke="#00C49F" />
        </LineChart>
      </ResponsiveContainer>
    </CardContent>
  </Card>

  <Card>

```

```

<CardHeader>
    <CardTitle>Model Distribution</CardTitle>
</CardHeader>
<CardContent>
    <ResponsiveContainer width="100%" height={300}>
        <PieChart>
            <Pie
                data={modelStats?.distribution || []}
                dataKey="value"
                nameKey="name"
                cx="50%"
                cy="50%"
                outerRadius={100}
                label
            >
                {((modelStats?.distribution || []).map((entry: any, index: number) => (
                    <Cell key={entry.name} fill={COLORS[index % COLOR.length]}>
                )))
            </Pie>
            <Tooltip />
        </PieChart>
    </ResponsiveContainer>
</CardContent>
</Card>
</div>

{/* Model Performance Table */}
<Card>
    <CardHeader>
        <CardTitle>Model Performance</CardTitle>
    </CardHeader>
    <CardContent>
        <table className="w-full">
            <thead>
                <tr className="border-b">
                    <th className="text-left py-2">Model</th>
                    <th className="text-right">Requests</th>
                    <th className="text-right">Avg Latency</th>
                    <th className="text-right">Success Rate</th>
                    <th className="text-right">Cost</th>
                </tr>
            </thead>
            <tbody>
                {((modelStats?.models || []).map((model: any) => (
                    <tr key={model.id} className="border-b">
                        <td className="py-2">{model.name}</td>
                        <td className="text-right">{model.requests.toLocaleString()}</td>
                        <td className="text-right">{model.avgLatency}ms</td>
                )))
            </tbody>
        </table>
    </CardContent>
</Card>

```

```
        <td className="text-right">{(model.successRate * 100).toFixed(2)}</td>
      </tr>
    ))}
  </tbody>
</table>
</CardContent>
</Card>
</div>
);
}
```