import React from 'react';

import { LineChart, BarChart, AreaChart, Legend, Tooltip, XAxis, YAxis } from 'recharts';

import { Card, CardHeader, CardTitle, CardContent } from '@/components/ui/card';

interface AnalyticsDashboard {

posts: Array<{

id: string;

content: string;

metrics: Array<{

timestamp: Date;

impressions: number;

reactions: number;

comments: number;

shares: number;

clicks: number;

}>;

}>;

}

const AnalyticsDashboard = ({ posts }) => {

return (

<div className="space-y-6">

{/\* Performance Summary Cards \*/}

<div className="grid grid-cols-4 gap-4">

<MetricCard

title="Total Impressions"

value={calculateTotalImpressions(posts)}

trend={calculateTrend('impressions')}

/>

<MetricCard

title="Avg. Engagement Rate"

value={calculateEngagementRate(posts)}

trend={calculateTrend('engagement')}

/>

<MetricCard

title="Best Performing Post"

value={findBestPost(posts)}

type="post"

/>

<MetricCard

title="Weekly Growth"

value={calculateGrowthRate(posts)}

trend={calculateTrend('growth')}

/>

</div>

{/\* Trend Charts \*/}

<div className="grid grid-cols-2 gap-4">

<Card>

<CardHeader>

<CardTitle>Performance Over Time</CardTitle>

</CardHeader>

<CardContent>

<LineChart data={prepareTimeSeriesData(posts)}>

<XAxis dataKey="date" />

<YAxis />

<Tooltip />

<Legend />

<Line type="monotone" dataKey="impressions" stroke="#8884d8" />

<Line type="monotone" dataKey="reactions" stroke="#82ca9d" />

</LineChart>

</CardContent>

</Card>

<Card>

<CardHeader>

<CardTitle>Engagement Distribution</CardTitle>

</CardHeader>

<CardContent>

<BarChart data={prepareEngagementData(posts)}>

<XAxis dataKey="type" />

<YAxis />

<Tooltip />

<Legend />

<Bar dataKey="value" fill="#8884d8" />

</BarChart>

</CardContent>

</Card>

</div>

{/\* Post Performance Comparison \*/}

<Card>

<CardHeader>

<CardTitle>Post Performance Comparison</CardTitle>

</CardHeader>

<CardContent>

<div className="space-y-4">

{posts.map(post => (

<PostPerformanceRow

key={post.id}

post={post}

metrics={calculateMetrics(post)}

/>

))}

</div>

</CardContent>

</Card>

</div>

);

};

// Helper components

const MetricCard = ({ title, value, trend, type = 'number' }) => {

return (

<Card>

<CardContent className="pt-6">

<div className="text-2xl font-bold">

{type === 'number' ? formatNumber(value) : value}

</div>

<div className="text-sm text-gray-500">{title}</div>

{trend && (

<div className={`text-sm ${trend > 0 ? 'text-green-500' : 'text-red-500'}`}>

{trend > 0 ? '↑' : '↓'} {Math.abs(trend)}%

</div>

)}

</CardContent>

</Card>

);

};

const PostPerformanceRow = ({ post, metrics }) => {

return (

<div className="flex items-center space-x-4 p-4 hover:bg-gray-50 rounded-lg">

<div className="flex-1">

<div className="font-medium">{post.content.substring(0, 100)}...</div>

<div className="text-sm text-gray-500">

Posted {formatDate(post.postedAt)}

</div>

</div>

<div className="flex space-x-8">

<MetricPill label="Impressions" value={metrics.impressions} />

<MetricPill label="Reactions" value={metrics.reactions} />

<MetricPill label="Comments" value={metrics.comments} />

<MetricPill label="Shares" value={metrics.shares} />

</div>

</div>

);

};

const MetricPill = ({ label, value }) => (

<div className="text-center">

<div className="text-lg font-semibold">{formatNumber(value)}</div>

<div className="text-xs text-gray-500">{label}</div>

</div>

);

Note: The system includes:

1. **Summary Cards**:
   * Total impressions
   * Average engagement rate
   * Best performing post
   * Growth trends
2. **Time-based Charts**:
   * Performance over time
   * Engagement distribution
   * Post comparison
3. **Post-level Analytics**:
   * Individual post performance
   * Comparative metrics
   * Growth indicators