// Monitoring Service

interface MonitoringMetrics {

userActivity: {

activeUsers: number;

contentGenerated: number;

exercisesCompleted: number;

};

contentQuality: {

averageBeigeScore: number;

personalizationRate: number;

failureRate: number;

};

systemHealth: {

apiLatency: number;

errorRate: number;

resourceUtilization: number;

};

}

class MonitoringService {

async trackMetrics(): Promise<MonitoringMetrics> {

const userActivity = await this.getUserActivityMetrics();

const contentQuality = await this.getContentQualityMetrics();

const systemHealth = await this.getSystemHealthMetrics();

// Log metrics

await this.logMetrics({

userActivity,

contentQuality,

systemHealth

});

// Check alert thresholds

await this.checkAlertThresholds({

userActivity,

contentQuality,

systemHealth

});

return {

userActivity,

contentQuality,

systemHealth

};

}

private async checkAlertThresholds(metrics: MonitoringMetrics): Promise<void> {

// Check error rates

if (metrics.systemHealth.errorRate > 0.05) {

await this.triggerAlert('HIGH\_ERROR\_RATE', metrics.systemHealth);

}

// Check content quality

if (metrics.contentQuality.averageBeigeScore < 40) {

await this.triggerAlert('LOW\_CONTENT\_QUALITY', metrics.contentQuality);

}

// Check API health

if (metrics.systemHealth.apiLatency > 1000) {

await this.triggerAlert('HIGH\_API\_LATENCY', metrics.systemHealth);

}

}

}