# Monitoring & Observability

## Logging Strategy

**Structured Logging with Pino** (optional):

// lib/logger.ts  
import pino from 'pino'  
  
export const logger = pino({  
 level: process.env.NODE\_ENV === 'production' ? 'info' : 'debug',  
 transport: process.env.NODE\_ENV !== 'production'  
 ? { target: 'pino-pretty' }  
 : undefined,  
})  
  
// Usage  
logger.info({ initiativeId, userId }, 'Initiative created successfully')  
logger.error({ error, initiativeId }, 'Failed to create initiative')

## Error Monitoring

**Sentry Integration**:

// lib/sentry.ts  
import \* as Sentry from '@sentry/nextjs'  
  
Sentry.init({  
 dsn: process.env.NEXT\_PUBLIC\_SENTRY\_DSN,  
 tracesSampleRate: 1.0,  
 environment: process.env.NODE\_ENV,  
})  
  
// Capture errors  
try {  
 await createInitiative(data)  
} catch (error) {  
 Sentry.captureException(error, {  
 tags: { action: 'create\_initiative' },  
 extra: { initiativeData: data },  
 })  
 throw error  
}

## Performance Monitoring

**Vercel Analytics** (built-in):

// app/layout.tsx  
import { Analytics } from '@vercel/analytics/react'  
  
export default function RootLayout({ children }) {  
 return (  
 <html>  
 <body>  
 {children}  
 <Analytics />  
 </body>  
 </html>  
 )  
}