**📊 GOOGLE ANALYTICS 4 - COMPLETE IMPLEMENTATION GUIDE**

**For:** Kurzora Platform  
**Platform:** Next.js/React  
**Time Required:** 2 hours  
**Cost:** Free

**🎯 STEP 1: CREATE GOOGLE ANALYTICS ACCOUNT**

**A. Go to Google Analytics**

1. **Visit:** https://analytics.google.com
2. **Sign in** with your Google account (create one if needed)
3. **Click:** "Start measuring"

**B. Set Up Account**

1. **Account Name:** "Kurzora Trading Platform"
2. **Property Name:** "Kurzora.com"
3. **Reporting Time Zone:** "Germany (Berlin)"
4. **Currency:** "Euro (EUR)"
5. **Industry Category:** "Finance & Insurance"
6. **Business Size:** "Small"

**C. Choose Platform**

1. **Select:** "Web"
2. **Website URL:** "https://kurzora.com" (or your domain)
3. **Stream Name:** "Kurzora Web Stream"

**D. Get Measurement ID**

You'll get a **Measurement ID** like: G-XXXXXXXXXX **SAVE THIS** - you'll need it for implementation!

**🛠️ STEP 2: IMPLEMENT IN YOUR NEXT.JS PROJECT**

**A. Install Google Analytics Package**

cd ~/Desktop/kurzora/kurzora-platform/frontend

npm install gtag

**B. Create Analytics Configuration File**

**File:** src/lib/analytics.ts

// Google Analytics 4 Configuration

export const GA\_MEASUREMENT\_ID = process.env.NEXT\_PUBLIC\_GA\_MEASUREMENT\_ID || 'G-XXXXXXXXXX';

// Initialize Google Analytics

export const initGA = () => {

if (typeof window !== 'undefined') {

window.gtag('config', GA\_MEASUREMENT\_ID, {

page\_title: document.title,

page\_location: window.location.href,

});

}

};

// Track page views

export const trackPageView = (url: string) => {

if (typeof window !== 'undefined') {

window.gtag('config', GA\_MEASUREMENT\_ID, {

page\_path: url,

});

}

};

// Track custom events

export const trackEvent = (action: string, category: string, label?: string, value?: number) => {

if (typeof window !== 'undefined') {

window.gtag('event', action, {

event\_category: category,

event\_label: label,

value: value,

});

}

};

// Kurzora-specific tracking events

export const trackSignalView = (signalId: string, signalTicker: string) => {

trackEvent('view\_signal', 'signals', `${signalTicker}-${signalId}`);

};

export const trackTrialStart = (tier: string) => {

trackEvent('trial\_start', 'subscription', tier);

};

export const trackSubscription = (tier: string, amount: number) => {

trackEvent('purchase', 'subscription', tier, amount);

};

export const trackSignalExecute = (signalId: string, signalTicker: string) => {

trackEvent('execute\_trade', 'paper\_trading', `${signalTicker}-${signalId}`);

};

**C. Add to Environment Variables**

**File:** .env.local

NEXT\_PUBLIC\_GA\_MEASUREMENT\_ID=G-XXXXXXXXXX

*(Replace with your actual Measurement ID)*

**D. Add Google Analytics Script to App**

**File:** src/pages/\_app.tsx (or your main app file)

import Script from 'next/script';

import { initGA, trackPageView } from '@/lib/analytics';

import { useRouter } from 'next/router';

import { useEffect } from 'react';

function MyApp({ Component, pageProps }: AppProps) {

const router = useRouter();

useEffect(() => {

// Initialize GA when app loads

initGA();

// Track page changes

const handleRouteChange = (url: string) => {

trackPageView(url);

};

router.events.on('routeChangeComplete', handleRouteChange);

return () => {

router.events.off('routeChangeComplete', handleRouteChange);

};

}, [router.events]);

return (

<>

{/\* Google Analytics Scripts \*/}

<Script

strategy="afterInteractive"

src={`https://www.googletagmanager.com/gtag/js?id=${process.env.NEXT\_PUBLIC\_GA\_MEASUREMENT\_ID}`}

/>

<Script

strategy="afterInteractive"

dangerouslySetInnerHTML={{

\_\_html: `

window.dataLayer = window.dataLayer || [];

function gtag(){dataLayer.push(arguments);}

gtag('js', new Date());

gtag('config', '${process.env.NEXT\_PUBLIC\_GA\_MEASUREMENT\_ID}');

`,

}}

/>

<Component {...pageProps} />

</>

);

}

export default MyApp;

**🎯 STEP 3: ADD KURZORA-SPECIFIC TRACKING**

**A. Track Key User Actions**

Add tracking to your existing components:

**In SignalTable Component:**

import { trackSignalView, trackSignalExecute } from '@/lib/analytics';

// When user views a signal

const handleSignalView = (signal: Signal) => {

trackSignalView(signal.id, signal.ticker);

};

// When user executes a trade

const handleExecuteTrade = (signal: Signal) => {

trackSignalExecute(signal.id, signal.ticker);

// ... rest of your execute logic

};

**In Subscription Components:**

import { trackTrialStart, trackSubscription } from '@/lib/analytics';

// When user starts trial

const handleTrialStart = (tier: string) => {

trackTrialStart(tier);

// ... rest of your trial logic

};

// When user subscribes

const handleSubscription = (tier: string, amount: number) => {

trackSubscription(tier, amount);

// ... rest of your subscription logic

};

**B. Track Important Pages**

**Dashboard visits, Settings changes, etc.**

import { trackEvent } from '@/lib/analytics';

// Track dashboard visits

useEffect(() => {

trackEvent('page\_view', 'dashboard', 'main\_dashboard');

}, []);

// Track filter usage

const handleFilterChange = (filterType: string, filterValue: string) => {

trackEvent('use\_filter', 'signals', `${filterType}-${filterValue}`);

};

**📊 STEP 4: CONFIGURE ANALYTICS GOALS**

**A. Set Up Conversions in GA4**

1. **Go to:** Admin > Events > Create Event
2. **Create these conversion events:**
   * trial\_start (when user starts trial)
   * purchase (when user subscribes)
   * execute\_trade (when user executes paper trade)

**B. Set Up Audiences**

1. **Go to:** Admin > Audiences > Create Audience
2. **Create these audiences:**
   * "Trial Users" (started trial but not subscribed)
   * "Paying Customers" (completed purchase)
   * "Active Traders" (executed trades in last 7 days)

**🎯 STEP 5: CREATE CUSTOM DASHBOARD**

**A. Key Metrics to Track**

1. **Acquisition:**
   * Daily/Weekly visitors
   * Trial signup rate
   * Traffic sources
2. **Engagement:**
   * Pages per session
   * Session duration
   * Bounce rate
3. **Conversion:**
   * Trial-to-paid conversion rate
   * Revenue tracking
   * User journey analysis

**B. Set Up Reports**

1. **Go to:** Reports > Library
2. **Create custom report** with these dimensions:
   * User acquisition source
   * Subscription tier
   * Geographic location
   * Device type

**🚀 STEP 6: VERIFY IMPLEMENTATION**

**A. Test Analytics**

1. **Open your website** in browser
2. **Go to GA4 Real-time reports**
3. **Navigate around your site**
4. **Verify events are firing** in real-time

**B. Test Key Events**

* Start a trial signup (check trial\_start event)
* View signals (check view\_signal event)
* Execute a paper trade (check execute\_trade event)

**C. Debug with Browser Tools**

1. **Open Developer Tools**
2. **Go to Console**
3. **Type:** dataLayer
4. **Should see:** Array with GA events

**📈 STEP 7: WEEKLY REPORTING SETUP**

**A. Create Weekly Report Template**

KURZORA ANALYTICS WEEKLY REPORT

TRAFFIC:

• Total Users: X (+/-% vs last week)

• New Users: X

• Returning Users: X

• Top Traffic Sources: [Organic, Direct, Referral]

CONVERSION:

• Trial Signups: X

• Trial-to-Paid Rate: X%

• Revenue This Week: €X

ENGAGEMENT:

• Avg Session Duration: X minutes

• Pages per Session: X

• Most Viewed Pages: [Dashboard, Signals, etc.]

USER BEHAVIOR:

• Signals Viewed: X

• Trades Executed: X

• Most Popular Filters: [List]

**B. Set Up Automated Reports**

1. **Go to:** Admin > Data Export
2. **Schedule weekly email reports** to your email
3. **Include:** Key metrics summary

**✅ IMPLEMENTATION CHECKLIST**

* [ ] Create GA4 account and get Measurement ID
* [ ] Install gtag package in project
* [ ] Create analytics.ts configuration file
* [ ] Add Measurement ID to .env.local
* [ ] Add GA scripts to \_app.tsx
* [ ] Add tracking to key components (signals, subscriptions)
* [ ] Configure conversion events in GA4
* [ ] Create custom audiences
* [ ] Test implementation with real-time reports
* [ ] Set up weekly reporting template

**Estimated Time:** 2 hours total **Difficulty:** Easy (copy-paste most of the code) **Result:** Complete analytics tracking for your platform!