

Writer Payments Admin Dashboard

A comprehensive system to track writer performance, calculate payments with custom formulas, and gain AI-powered insights.

Sports Mockery - Internal Tool

Overview

The Writer Payments Admin Dashboard is a secure, admin-only portal that automates the tracking and calculation of writer payments. It replaces manual spreadsheet tracking with a real-time system that automatically counts posts, tracks views, and calculates payments based on each writer's unique formula.

- ✓ **Key Benefit:** No more manual data entry or formula errors. The system automatically tracks posts published each month and their view counts, then calculates payments using each writer's custom formula.

Core Features



Automatic Tracking

Posts and views are automatically counted for each writer each month. No manual data entry required.



Custom Payment Formulas

Each writer can have their own unique payment formula. Supports views-only, posts + views, tiered rates, and threshold bonuses.



AI-Powered Insights

Claude AI analyzes writer performance and provides actionable insights about trends, anomalies, and recommendations.



Visual Charts

D3.js-powered charts show posts per writer, views distribution, and payment breakdowns at a glance.



View Counter Protection

Built-in rate limiting prevents artificial view inflation. Views are counted fairly to ensure accurate payments.



Manual Adjustments

Easily add podcast payments, social media bonuses, and other manual adjustments on top of calculated amounts.

Writer Payment Formulas

Each writer has a unique payment formula configured in the system. Here are the current formulas:

WRITER	FORMULA TYPE	CALCULATION
Erik Lambert	Views Only	$(\text{views} / 1000) \times 4$
Dave	Posts + Views + Podcast	$(\text{posts} \times 10) + (\text{views} / 1000) + \text{podcast} + \text{bonus}$
Ficky	Posts + Views + Podcast	$(\text{posts} \times 10) + (\text{views} / 1000) + \text{podcast}$
Ryan Dauterive	Tiered	IF posts < 20: $(\text{posts} \times 7.5)$ ELSE: $(\text{posts} \times 15) + (\text{views} / 1000)$
Aldo Soto	Posts + Views	$(\text{posts} \times 5) + (\text{views} / 1000) \times 3$
DaBearsBlog	Views Only	$(\text{views} / 1000) \times 5$
Mitchell Kaminski	Posts + Views	$(\text{posts} \times 5) + (\text{views} / 1000) \times 3$
Craig Rowland	Views Threshold	$(\text{posts} \times 10) + (\text{views} / 1000) + \text{IF views} > 100\text{K}: (\text{views} / 1000)$
Dhruv Koul	Posts + Views	$(\text{posts} \times 5) + (\text{views} / 1000) \times 3$
ChiSoxFanMike	Posts + Views	$(\text{posts} \times 10) + (\text{views} / 1000)$

WRITER	FORMULA TYPE	CALCULATION
Isaac Nielsen	Views Threshold	$(\text{posts} \times 10) + \text{IF views} > 100\text{K}: (\text{views} / 1000)$



Flexible Formulas: New writers can be added with any of these formula types, or a completely custom formula can be created through the admin interface.

Payment Dashboard Columns

The main payment table shows the following columns for each writer:

COLUMN	SOURCE	DESCRIPTION
Writer	System	Writer's display name
# Posts	Auto-calculated	Number of posts published in the selected month
# Views	Auto-calculated	Total views on posts published that month
\$ Views	Auto-calculated	Payment calculated from writer's formula
Social #	Manual input	Number of social media posts (for tracking)
\$ Podcast	Manual input	Podcast appearance payment
\$ Social	Manual input	Social media payment
\$ Bonus	Manual input	Any additional bonus payment
\$ Total	Auto-calculated	Sum of all payment columns

Visual Analytics

The dashboard includes three D3.js-powered charts for quick visual analysis:



Posts by Writer

Bar chart showing post count per writer for the month



Views by Writer

Bar chart with color gradient based on view count



Payment Distribution

Pie chart showing how total payments are distributed

AI-Powered Insights

The dashboard uses Claude AI (Haiku model) to analyze writer performance and provide actionable insights. The AI reviews current and previous month data to identify:



Top Performers

Identifies writers with the highest views, most posts, and best engagement.



Anomalies

Flags unusual patterns like sudden drops in productivity or unexpected view spikes.



Recommendations

Suggests content strategies based on what's driving views and engagement.



Trend Analysis

Compares month-over-month changes to highlight growth or decline.

View Counter Security

Since payments are based on view counts, the system includes built-in protection against artificial inflation:

- **Rate Limiting:** Maximum 3 views per post per visitor per hour
- **Fingerprinting:** Combines IP address and browser info to identify unique visitors
- **Silent Rejection:** Rate-limited views return success (to prevent gaming) but aren't counted



Fair Payment: This ensures writers are paid based on genuine reader engagement, not artificial view inflation.

Setup Guide

Follow these steps to get the Writer Payments Dashboard running:

1 Configure Supabase

Go to your Supabase dashboard and run the SQL schema file to create all required tables. The schema is located at [supabase/schema.sql](#).

2 Set Environment Variables

Copy `.env.example` to `.env.local` and fill in your Supabase URL, keys, and Anthropic API key.

```
NEXT_PUBLIC_SUPABASE_URL=https://your-project.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-key
SUPABASE_SERVICE_ROLE_KEY=your-service-role-key
ANTHROPIC_API_KEY=your-anthropic-api-key
```

3 Create Admin User

In Supabase Auth, create a user and set `role: 'admin'` in their `user_metadata`. Only users with this role can access the admin dashboard.

4 Run the Application

Start the development server and navigate to `/admin` to access the dashboard.

```
npm run dev
```

Technical Architecture

The feature is built with Next.js 16 (App Router), Supabase, D3.js, and Claude AI. Here's the file structure:

```
src/
├─ app/
│  ├─ admin/ # Admin pages (protected)
│  │  ├─ page.tsx # Main dashboard
│  │  ├─ layout.tsx # Admin layout with nav
│  │  └─ writers/ # Writer management
│  ├─ api/
│  │  ├─ admin/ # Admin API routes
│  │  │  ├─ writers/ # CRUD for writers
│  │  │  ├─ payments/ # Payment data
│  │  │  └─ insights/ # AI insights
│  │  └─ views/ # View counter (public)
│  └─ auth/ # Login/logout
├─ components/admin/ # UI components
│  ├─ WriterPaymentTable.tsx
│  ├─ FormulaEditor.tsx
│  └─ AIInsightsPanel.tsx
├─ charts/ # D3.js charts
├─ lib/
│  ├─ formulas.ts # Payment calculation
│  └─ supabase-server.ts # Database client
└─ types/index.ts # TypeScript types
```

Summary

The Writer Payments Admin Dashboard transforms how we track and pay writers:

Before

- Manual spreadsheet tracking
- Copy/paste view counts from WordPress

After

- Automatic post and view tracking
- Real-time payment calculations
- Custom formulas per writer

- Manual formula calculations
- No insights or trend analysis
- Risk of calculation errors

- AI-powered performance insights
- Visual charts and analytics

Sports Mockery - Writer Payments Admin Dashboard

Built with Next.js, Supabase, D3.js, and Claude AI