

# ***Senna Creative Operations Platform*** ***— Manual***

---

**Version:** SOS FINAL **Product:** TFDWX / Senna **Author:** Chris Ross **Date:** April 2026

---

## **Table of Contents**

---

1. Introduction
  2. Getting Started
  3. Platform Overview
  4. Module Guide — Senna Sync
  5. Module Guide — Jenova
  6. Module Guide — Gaia
  7. Module Guide — Asia
  8. Module Guide — Memory
  9. Site Preview
  10. Account & Billing
  11. New Project Intake
  12. Legal & Compliance
  13. Credit System
  14. Technical Architecture
  15. Component Reference
  16. State Management
  17. Compression & Performance
  18. Deployment Guide
  19. Troubleshooting
-

# 1. Introduction

---

Senna is an AI-native creative operations platform built for independent music artists and creative professionals. It replaces 12+ disconnected tools with a single unified system where every action in one module enriches the others. The platform covers the full creative lifecycle: production, distribution, monetization, analytics, and learning.

The platform is delivered as a single-file React JSX artifact (~130KB) that renders as a fully interactive application. This manual covers both how to use the platform as an end user and how the codebase is structured for developers.

---

## 2. Getting Started

---

### Login

The platform opens to a login screen with a custom waveform-matching CAPTCHA:

1. Enter any password in the password field
2. Click "Start Challenge" to begin the frequency-matching CAPTCHA
3. Drag the slider to match the target waveform (green waveform must align with the fuchsia target)
4. When match accuracy exceeds 80%, the CAPTCHA passes automatically
5. Click "Sign In" to enter the platform

The CAPTCHA uses real-time SVG waveform rendering with sine functions — the target frequency is randomized each attempt. This serves as both a bot-prevention mechanism and a brand-aligned onboarding experience.

### First Screen

After login, you land on the Overview Dashboard, which provides a cross-module health snapshot: credit balance, burn rate, alerts, listener count, and key metrics sourced from every module.

---

## 3. Platform Overview

---

### Navigation Structure

The platform uses a dual-navigation system:

**Top Bar:** Overview | Sync | Jenova | Gaia | Asia | Memory | Site + Credits display, notification bell, Account button, and "+ New Project" button.

**Left Sidebar:** Same module links with color-coded dots, plus a "Site Preview" entry under the "Output" section. A credit gauge sits at the bottom of the sidebar showing balance, burn rate, and a visual health bar (green > 50, amber > 20, red < 20).

### Module Order

The canonical navigation order is: Overview → Senna Sync → Jenova → Gaia → Asia → Memory. This order reflects the creative workflow: manage your catalog (Sync), produce visual content (Jenova), distribute and schedule (Gaia), get strategic intelligence (Asia), and review what Senna has learned (Memory).

### Overview Dashboard

The Overview page contains:

- **Account Health** — 4 KPI cards: Credits, Burn Rate, Alerts, Listeners
  - **Project Performance** — 6 cross-module metric cards sourcing data from Sync, Jenova, Gaia, and Asia
  - **Connected Intelligence** — 6 module cards (Sync, Jenova, Gaia, Memory, Streams, Rights) showing what each module provides to the ecosystem
  - **System Status** — Live module health indicators
-

## 4. Module Guide — Senna Sync

---

### Purpose

Senna Sync is the rights management and pitch tracking hub. It handles licensing deals, split sheets, source file management, and revenue tracking.

### Features

**Revenue Dashboard:** Total revenue, pending deals, and active pitches displayed as KPI cards with delta indicators.

**Pitch Pipeline:** Active pitches shown with partner name, project, status (Reviewing/Countered/Negotiating), and value. Each pitch is a card with color-coded status badges.

**Source File Player:** Each track in the catalog shows: - Track name, BPM, and key - Waveform visualization bar - Play/pause button (colored to match the track's waveform color) - Click-to-seek on the waveform progress bar - File format badges (WAV, MP3, FLAC, AIFF, MOV, etc.)

**Play Button Colors:** Each track's play button uses the track's `wc` (waveform color) property — Midnight Circuit = red, Analog Dreams = fuchsia, Frequencies = amber. This extends to the glow shadow around each button.

**Deal Tracker:** Recent deals with partner, project, format (Film/TV/Ad), value, and status.

**Split Sheet Manager:** Automated split calculations per track with writer, publisher, and PRO registration status.

**Rights Status:** Blockchain-stamped encrypted rights status with clearance percentage.

---

## 5. Module Guide — Jenova

---

### Purpose

Jenova is the AI-powered cinematic production pipeline. It takes a source track and produces visual content (music videos, social clips, visualizers) across multiple platforms through a 25-stage pipeline.

### 25-Stage Pipeline

The pipeline displays 15 key milestone stages in a 3x5 grid:

Row 1	Row 2	Row 3
1. Brief	6. Mood	11. Color
2. Research	7. Assets	12. Motion
3. Script	8. Footage	13. Sound
4. Story	9. Audio	14. Master
5. Vis Dir	10. Rough Cut	15. Review

Each stage shows as done (green), active (white with red background tint), or pending (dim). A progress bar and percentage (e.g., "Stage 8 of 25 — 32% complete") tracks overall pipeline progress.

### Video Theater

A full 16:9 canvas-based video preview with:

- Animated radial gradient visualizer (6 color orbs in motion when playing)
- Frame counter overlay ("JENOVA RENDER // FRAME 00042")
- Play/pause toggle with central button
- Transport controls (rewind, skip, speed)
- Timeline scrub bar with chapter markers
- Quality badges (4K, 24fps, ProRes)

- Timecode display (HH:MM:SS:FF)
- AI Scene detection with BPM sync
- Color palette preview
- Hook score display

## Channel Build Engine

Select from 4 output channels, each with platform-specific specs:

Channel	Spec	Formats
YouTube	16:9 · 4K · Long-form	Music Video, Visualizer, BTS, Premiere
TikTok	9:16 · 1080p · 15-60s	Hook Clip, BTS, Snippet, Challenge
Instagram	9:16 / 1:1 · Reels + Stories	Reel, Story, Carousel, Post
Cinema	2.39:1 · 4K DCI · ProRes	Short Film, Visual Album, Art Film, Festival

Selecting a channel opens a 5-step build engine: Source → Style → Generate → Review → Export. Each step shows real-time status text describing what the AI is doing. Completing the pipeline exports to Gaia for scheduling.

## Export Cut

The "Export Cut →" button sends the current render to Gaia. On click, it turns green and shows "✓ Exported to Gaia" for 2.5 seconds before resetting.

## Active Productions

4 concurrent productions tracked with dot-progress indicators showing stage completion out of total stages.

## Project Switcher

Switch between 3 active projects (Midnight Circuit Vol.3, Neon Sanctuary, Red Velocity II) using tab buttons. Each project displays:

- Source track metadata (BPM, Key, Duration, Feel, Signature, LUFFS)

- Visual direction, cut tempo (BPM-locked), color palette
- Hook target score
- Visibility control (Public/Team Only/Private) with Eye/Users/Lock icons

## Agent Activity

4 AI agents displayed with SVG donut charts showing completion percentage:

- Executive Producer (72%) — Reviewing footage selection
- Visual Director (45%) — Generating moodboard variations
- Sound Engineer (10%) — Waiting on footage lock
- Analytics (5%) — Gaia signals queued

---

## 6. Module Guide — Gaia

---

### Purpose

Gaia handles distribution intelligence, content scheduling, audience analytics, and cross-platform publishing.

### Features

**Content Calendar:** Schedule posts across 6 platforms with drag-to-reschedule. Add custom events with the "+ Add Event" button.

**Publishing Pipeline:** Select a track → choose platforms → step through Content → Schedule → Publish. Each step shows platform-specific formatting and timing optimization.

**Audience Analytics:** Platform reach metrics, engagement rates, and per-track performance. Includes scenario modeling (Social, Streaming, Sync, Full Push) with projected reach and revenue estimates.

**Bio Generator:** AI-generated artist bios with per-platform versions (Spotify, Instagram, Press Kit, LinkedIn). Copy-to-clipboard functionality with visual feedback.

**Automation Rules:** Pre-configured distribution rules that fire on triggers (e.g., "Auto-cross-post Reels to TikTok", "Alert if engagement drops below 2%").

---

## 7. Module Guide — Asia

---

### Purpose

Asia is the strategic intelligence engine. It answers natural-language questions about your creative business using keyword-matched AI responses.

### How to Use

1. Type a question in the query input (e.g., "How is my sync revenue performing?")
2. Asia matches your query against 8 topic categories: sync, hook, distribution, revenue, audience, production, legal, growth
3. Responses include strategic insights, data points, and actionable recommendations
4. Each query costs 1 credit
5. Quick-ask topic buttons provide one-click access to common questions

### Topic Aliases

Asia recognizes natural language variations. For example, "money" or "income" maps to the revenue topic, "platform" or "spotify" maps to distribution, and "viral" or "tiktok" maps to growth.

### Query History

All queries are logged in a history panel showing the question, matched topic, and response for reference.

## Intelligence Cards

Pre-built insight cards cover: Market Position, Content Strategy, Revenue Optimization, Audience Growth, Platform Performance, Trend Analysis, Competitive Intelligence, and Risk Assessment.

---

## 8. Module Guide — Memory

---

### Purpose

Memory is Senna's learning engine. It retains every decision and result across all modules, building pattern recognition over time.

### Features

**Insight Chips:** 8 key learnings displayed as toggleable chips. Click to highlight/filter by insight (e.g., "Dark Electronic — 94 hook avg", "IG Reels 9:16 performs 3x"). Active chips glow with their color.

**Stats Dashboard:** 3 KPI cards — Decisions Logged (847), Projects in Memory (12), Pattern Match Rate (94%).

**Learning Loop:** Visual cycle showing Create → Distribute → Monetize → Learn → (repeat). Represents how Senna's intelligence compounds with every action.

**Credit Timeline:** Live feed of credit transactions showing action, timestamp, and credit impact (positive additions in green, negative spends in red).

---

## 9. Site Preview

---

### Purpose

Preview and manage your public-facing Senna.io creator website.

## Features

**Hero Preview:** Displays the "Go. Fast. Now." brand tagline with the platform description.

**Request Access:** Click to send an access request (shows "✓ Access Requested" confirmation for 3 seconds).

**View Site:** Toggles an expanded preview panel showing: - Live status indicator (green dot + "senna.io") - SSL badge - Platform stats: 3 Active Projects, 142K Listeners, 99.9% Uptime

---

## 10. Account & Billing

---

### Tabs

**Profile:** Displays account fields (Name, Email, Plan, Role, Org, Joined) in a 2-column grid. "Edit Profile" button provides save feedback (shows "✓ Saved" for 2 seconds on click).

**Billing:** Enterprise plan details (\$299/mo), next invoice date, and invoice history (Mar/Feb/Jan 2026) with "Paid" status badges.

**Security:** Two-Factor Auth status, Session Timeout setting, API Keys count, and Last Login timestamp.

**Sign Out:** Confirmation modal with cancel/confirm options. Sign out returns to the login screen.

---

## 11. New Project Intake

---

### Workflow

Click "+ New Project" in the top bar to open the intake wizard:

**Step 1 — Project Info:** - Project name (text input) - Release date (date picker) - Project type selector: Single (♪), EP/Album (□), Music Video (□), Visual (Ⓢ), Campaign (□) - Overview/creative direction textarea

**Step 2 — Upload Assets:** - Drag-and-drop file upload zone - Supports audio, video, images, documents - Uploaded files show name, size, and type badge

**Step 3 — Review & Launch:** - Summary card showing project name, type, release date, and asset count - "□ Launch Project" button: spends 5 credits, logs the project name, and closes the wizard

---

## 12. Legal & Compliance

---

### Accessing Legal Documents

Click the "Legal" button (shield icon) in the bottom-right corner of the platform to open the Legal Modal.

### Available Policies

Policy	Description
Terms of Service	Platform usage, credit terms, IP ownership, liability
Privacy Policy	GDPR + CCPA compliant data handling
Acceptable Use Policy	Content standards, enforcement, three-strike system
Cookie Policy	First-party only, no ad tracking
DMCA & IP Protection	Takedown procedures, counter-notice process
AI Disclosure & DPA	AI transparency, data processing terms

### Viewing Documents

Click "View PDF" on any policy. The button shows "✓ Sent" confirmation indicating the document has been queued for delivery.

## Security Banner

A green banner at the top of the legal modal confirms: "Your data is protected — 256-bit encryption · Industry-standard security."

---

## 13. Credit System

---

### How Credits Work

Senna uses a credit-based AI metering system. Every AI-powered action costs credits:

- **Starting Balance:** 247 credits
- **Burn Rate:** ~18 credits/day
- **Asia Queries:** 1 credit each
- **Project Launch:** 5 credits
- **Jenova Renders:** Variable (based on complexity)
- **Gaia Publishing:** Variable (based on platforms)

### Monitoring

Credits are visible in 3 locations: 1. Top bar (header) — current balance with "credits" label 2. Left sidebar (bottom) — balance, burn rate, and health bar 3. Overview Dashboard — KPI card with burn rate delta

### Credit Timeline

The Memory page shows a live transaction log of all credit spends and additions with timestamps.

### Health Indicators

The sidebar credit bar changes color based on remaining balance: green (> 50), amber (20-50), red (< 20).

---

## 14. Technical Architecture

---

### File Structure

The platform is a single-file React JSX artifact: `Senna-SOS-FINAL.jsx` (~130KB, 1784 lines).

### Technology Stack

- **Framework:** React (functional components, hooks)
- **State:** `useState`, `useRef`, `useEffect`, `useCallback`
- **Rendering:** Inline styles (no external CSS)
- **Graphics:** SVG (donut charts, waveforms, logos), HTML5 Canvas (Jenova video theater)
- **Icons:** `lucide-react` (Play, Upload, Shield, `AlertTriangle`, Users, Lock, Eye, etc.)

### Component Hierarchy

```
SennaApp (root)
├─ LoginPage (auth gate)
├─ Header (top bar nav + credits + notifications)
├─ Sidebar (left nav + credit gauge)
├─ PageComponent (dynamic, one of:)
│   ├─ OverviewPage
│   ├─ SyncPage
│   ├─ JenovaPage
│   ├─ GaiaPage
│   ├─ AsiaPage
│   ├─ MemoryPage
│   ├─ SitePage
│   └─ AccountPage
├─ IntakeWizard (modal)
└─ LegalModal (modal)
```

## Shared Components

Component	Purpose
Glass	Glassmorphic card container with border and background
GlassSm	Smaller glass card variant
GlassTitle	Card header with optional badge
GlassSub	Card subtitle/description
Badge	Status badge with tone (red/lime/amber/blue/fuchsia/muted)
Kpi	KPI metric card with value, label, delta, and color
BarRow	Horizontal bar chart row
Grid2/3/4	CSS Grid layout helpers (2/3/4 columns)
Spark	Sparkline micro-chart
Label	Section label (uppercase, small)
SennaLogo	Brand wordmark SVG

## Color System

All colors are defined in the `C` constant object:

Token	Hex	Usage
C.bg	#050505	Page background
C.red	#EF4444	Jenova, alerts, primary CTA
C.lime	#A3E635	Sync, success, credits
C.fuchsia	#E879F9	Gaia, secondary accent
C.amber	#F59E0B	Asia, warnings
C.blue	#60A5FA	Memory, info
C.white	#FFFFFF	Primary text

Token	Hex	Usage
C.muted	rgba(255,255,255,0.45)	Secondary text
C.border	rgba(255,255,255,0.08)	Card/element borders
C.surface	rgba(255,255,255,0.04)	Card backgrounds
C.surface2	rgba(255,255,255,0.06)	Elevated surfaces

## Style Constants

Constant	Value	Usage
BI	{fontWeight: 900, fontStyle: "italic"}	Bold italic headings (27 instances)
TU	"uppercase"	Text transform shorthand (46 instances)
S.col20	Flex column, gap 20	Page-level layout
S.col16	Flex column, gap 16	Section-level layout
S.h1	20px bold italic lime	Hero metric text
S.h3	13px bold white	Card value text
S.cap	9px uppercase muted	Caption labels
S.label	10px uppercase muted	Form labels

## 15. Component Reference

### Custom Hooks

**useCredits(initialBalance)** - Manages credit balance with

spend(amount, action) method - Maintains transaction history with timestamps -

Returns: { balance, history, spend, burnRate }

**useNotifications()** - Manages notification items with read/unread state - Returns:  
{ items, unread, markAllRead }

## Key State Variables by Page

**JenovaPage:** - `selectedChannel` — active build channel (youtube/tiktok/instagram/cinema) - `buildStep` — current step in 5-step build engine (0-4) - `isPlaying` — video theater play state - `exported` — export confirmation state - `activeProj` — selected project ID (mc3/ns1/rv2) - `projVis` — per-project visibility settings

**GaiaPage:** - `activeTab` — posts/analytics/automation tab - `gaiaPipeline` — active publishing pipeline - `gaiaStep` — pipeline step - `posted` — per-item publish state - `addEvent` — calendar event creation state - `userEvents` — user-added events array

**AsiaPage:** - `asking` — query loading state - `asiaQuery` — current query text - `asiaResponse` — matched response data - `queryHistory` — previous query log

**MemoryPage:** - `activeChips` — array of toggled insight chip labels

**SyncPage:** - `playing` — active track name - `playPct` — playback position percentage - `seekPos` — click-to-seek position

---

## 16. State Management

---

### Data Flow

The platform uses a top-down data flow pattern:

1. SennaApp (root) owns shared state: `page`, `showIntake`, `showLegal`, `loggedIn`, `credits`, `notifications`
2. `credits` and `notifications` are passed as props to all page components
3. Each page manages its own local state for UI interactions
4. The `onLogout` callback is passed to `AccountPage` to enable sign-out

## Cross-Module Communication

Modules communicate through shared data constants rather than real-time events:

- `JENOVA_PROJECTS` — shared project data used by Jenova and referenced in overviews
- `GAIA_TRACKS` — track data shared between Gaia and Sync
- `credits` hook — shared across all modules for spend tracking
- `notifications` hook — shared for alert display

## Constants

Constant	Content
<code>JENOVA_PROJECTS</code>	3 projects with BPM, key, duration, feel, LUFS, visual direction, hook target
<code>VIS_OPTIONS</code>	3 visibility levels (Public/Team/Private) with icons and colors
<code>NAV</code>	6 nav items with ID, label, and dot color
<code>PAGES</code>	Component lookup map for dynamic page rendering
<code>POLICIES</code>	6 legal policy definitions with title, description, icon, and filename

## 17. Compression & Performance

### Size Constraint

The artifact must stay under ~130,200 bytes to render successfully. The SOS FINAL release is 130,194 bytes (6 bytes of headroom).

### Compression Strategies Applied

1. **BI Constant** — `{fontWeight:900,fontStyle:"italic"}` extracted to `const BI` and spread across 27 instances. Saves ~740 bytes.

2. **TU Constant** — `"uppercase"` extracted to `const TU` and referenced in 46 `textTransform` properties. Saves ~414 bytes.
3. **CSS Property Removal** — Transitions, backdrop filters, and non-essential box shadows removed where they don't impact visual quality.
4. **Letter Spacing Shortening** — `"0.1em"` shortened to `".1em"` across all instances.
5. **Text Trimming** — Verbose subtitles shortened (e.g., "Identity and connected accounts." → "Identity and accounts.").

## What NOT to Compress

Per project guidelines, the following must never be shortened: descriptive text visible to users, labels, card content, feature data, module descriptions, or track metadata.

---

## 18. Deployment Guide

---

### As a React Artifact

The JSX file is designed to render in any environment that supports React with lucide-react icons:

1. Ensure React and lucide-react are available
2. Import the default export: `import SennaApp from './Senna-SOS-FINAL'`
3. Render: `<SennaApp />`

### As a Standalone Page

For standalone deployment, wrap in an HTML file with React CDN imports:

```
<script src="https://unpkg.com/react@18/umd/react.production.min.js"></script>
<script src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js"></script>
<script src="https://unpkg.com/lucide-react"></script>
```

## Production Codebase

A full Next.js + TypeScript production application exists in the `senna-production/` directory with Prisma ORM, Zustand stores, Tailwind CSS, and API routes. See `DEPLOY.md` for Vercel deployment instructions.

---

# 19. Troubleshooting

---

## Common Issues

**Blank white page after edits:** The file exceeds the ~130,200 byte rendering limit. Check `wc -c` on the file and compress until under limit.

**Bracket mismatch / render failure:** Count `{`, `(`, `[` vs `}`, `)`, `]` — they must be equal. A balanced file has delta 0.

**Play button not showing color:** Ensure each track object has a `wc` (waveform color) property mapped to a `C.*` color constant.

**Canvas not animating:** The Jenova video theater requires `isPlaying` state to be true. The animation loop checks this flag on each frame.

**Credits not decrementing:** Verify the `credits.spend(amount, action)` call is wired to the relevant button's `onClick` handler.

**Legal PDFs showing dead links:** The SOS FINAL release replaced `<a href>` links with click handlers that show "✓ Sent" feedback. If you see raw links, the fix was not applied.

## Verification Commands

```
# Check file size
wc -c Senna-SOS-FINAL.jsx

# Check bracket balance
python3 -c "
c = open('Senna-SOS-FINAL.jsx').read()
o = c.count('{') + c.count('(') + c.count('[')
cl = c.count('}') + c.count(')') + c.count(']')
print(f'Opens: {o}, Closes: {cl}, Delta: {o-cl}')
"

# Search for specific patterns
grep -c "fontWeight:900" Senna-SOS-FINAL.jsx # Should be 0 (all compressed to ...BI)
grep -c "\.\\.\\.BI" Senna-SOS-FINAL.jsx # Should be ~27
```

## Appendix A — BIG MASTER Folder Structure

```
BIG MASTER/
├── Platform/
│   ├── Senna-SOS-FINAL.jsx          # Creative ops platform (130KB)
│   └── Senna-SSM-FINAL.html        # Sampler-synthesizer plugin (~405KB)
├── Legal/
│   ├── TERMS-OF-SERVICE.md        # Source markdown
│   ├── PRIVACY-POLICY.md
│   ├── ACCEPTABLE-USE-POLICY.md
│   ├── COOKIE-POLICY.md
│   ├── DMCA-IP-POLICY.md
│   ├── AI-DISCLOSURE-DPA.md
│   ├── SECURITY-ARCHITECTURE.md
│   ├── TFDWX-Terms-of-Service.pdf  # Branded PDFs
│   ├── TFDWX-Privacy-Policy.pdf
│   ├── TFDWX-Acceptable-Use-Policy.pdf
│   ├── TFDWX-Cookie-Policy.pdf
│   ├── TFDWX-DMCA-IP-Policy.pdf
│   └── TFDWX-AI-Disclosure-DPA.pdf
```

```

|   |— TFDWX-Security-Architecture-CONFIDENTIAL.pdf
|   |— TFDWX-Mutual-NDA.docx
|   |— TFDWX-IP-Attorney-Checklist.docx
|— Media/
|   |— TFDWX-Senna-Pitch-Deck.pptx # 12-slide pitch deck
|   |— TFDWX-Senna-Pitch-Deck.html # Interactive HTML version
|   |— Senna-MediaKit-General.pdf
|   |— Senna-MediaKit-Press.pdf
|   |— Senna-MediaKit-Investor.pdf
|   |— Senna-MediaKit-Distributor.pdf
|— Docs/
|   |— Senna-SOS-Platform-Manual.md # This manual (source)
|   |— Senna-SOS-Platform-Manual.pdf # This manual (PDF)
|   |— TFDWX-Project-Summary-Resume.md
|   |— TFDWX-Session-Summary TECH.txt
|   |— Senna-SSM-v4-Manual.pdf
|   |— VST-Plugin-Project-Roadmap.md
|— Senna-BIG-MASTER.zip # Complete archive

```

## Appendix B — Module Color Reference

Module	Primary Color	Hex	Dot
Overview	White/20	rgba(255,255,255,0.2)	Dim
Senna Sync	Fuchsia	#E879F9	Fuchsia
Jenova	Red	#EF4444	Red
Gaia	Lime	#A3E635	Lime
Asia	Amber	#F59E0B	Amber
Memory	Blue	#60A5FA	Blue

## Appendix C — Jenova Project Specs

---

Field	Midnight Circuit Vol.3	Neon Sanctuary	Red Velocity II
ID	mc3	ns1	rv2
BPM	128	96	145
Key	Dm	Am	Fm
Duration	4:12	3:44	3:28
Feel	Driven	Ethereal	Aggressive
Signature	4/4	4/4	4/4
LUFS	-8	-10	-6
Visual Dir	Dark Electronic / Cinematic	Ambient / Lo-fi Visual	Hard Electronic / Moto
Cut Tempo	0.47s (BPM-locked)	0.63s (BPM-locked)	0.41s (BPM-locked)
Color	Teal / Dark Red	Purple / Soft White	Red / Black
Hook Target	90+	85+	92+

---

*Senna SOS FINAL — Built by Chris Ross / TFDWX — April 2026*