RACELOGIC: The Motorsport Intelligence Hub

Development Roadmap & Sprint Plan (MVP → v1)
October 1 – December 31, 2025

Tagline: Where History Meets the Checkered Flag
Mission: Deliver real-time and historical motorsport intelligence across F1, MotoGP, WRC, WEC, IndyCar, and NASCAR—with depth, accuracy, and fan-first design.
Audience: Motorsport enthusiasts, journalists, analysts, and fantasy league players.
Initial Focus: Formula 1 + MotoGP (MVP); expand to 6 series by v1.
Compliance: Unofficial site. Not affiliated with FIA, Dorna, NASCAR, or any racing body. All data sourced from open or permissively licensed APIs.

---

1. Product Overview & Philosophy

Core Idea:
Start with historical + simulated live data to build trust, performance, and compliance. Layer true real-time and AI insights in v1+.

MVP Goal:
Ship a revenue-ready, public-facing web app covering:

- Pre-Event: Schedules, history, team/driver bios, vehicle specs
- Live: Simulated race sessions using historical replays
- Post-Event: Results, standings, Hall of Fame

Three Pillars:

1. Accuracy: Source only from vetted, license-compliant APIs (OpenF1, MIT-licensed scrapers).
2. Thematic UX: Sport-specific color schemes, typography, and motion (F1 = aggressive red; MotoGP = sleek blue).
3. Scalable Architecture: Multi-sport data model from Day 1.

Tech Stack:

- Frontend: Next.js 15 (App Router), Tailwind CSS, V0 for rapid UI
- Backend: NestJS (modular per sport), pnpm
- Database: Supabase (PostgreSQL + RLS + Storage)

- Infra: Vercel (FE), Render (BE), GitHub Actions (CI)
- Observability: Custom admin dashboard for pipeline health

---

2. Modules & Scope (MVP vs v1)

A. Core Pages (MVP – F1 + MotoGP Only)

Page	Subpages	Data Source
Schedule	Previous / Current / Future	OpenF1 `meetings` + `sessions`
Live	Simulated session replay	OpenF1 `car_data`, `laps`, `position`
Results	Latest session classification	OpenF1 `session_result`
History	Eras, rule changes, champions	Wiki + curated datasets
Hall of Fame	Winners / Driver Champs / Constructor Champs	OpenF1 + historical CSVs
Teams	Team profiles, colors, history	OpenF1 `drivers` (team_name, team_colour)
Drivers	Bio, stats, career timeline	OpenF1 `drivers` + headshots
Vehicle	Tyres / Specs / Engine	F1 Technical, MotoGP.com, forums

B. Cross-Cutting (MVP)

- Sport-Themed UI: Dynamic CSS variables per sport
- Global Disclaimer: “Unofficial. Not affiliated with F1/MotoGP.”
- Newsletter Signup: Resend + Supabase Edge Function
- Admin Dashboard: Data freshness, error rates, last fetch

C. v1 Enhancements (Q1 2026)

- True live data (paid OpenF1 + MQTT)
- WRC stage times, WEC results, IndyCar/NASCAR schedules
- AI insights: “Tire degradation vs. lap 10”, “Top speed comparison”
- Mobile PWA with offline photo capture
- User favorites & alerts

---

3. Data Strategy (Finalized Sept 26–30, 2025)

Sport	Source	License	Coverage
F1	OpenF1	MIT-like, free (historical)	2018–2025: laps, telemetry, weather, radio
MotoGP	`ParsaD23/MotoGP-API`	MIT	2010–2025: results, riders, teams
WRC	WRC.com Live Timing	Public JSON	2023–2025: stage times, itineraries
IndyCar / NASCAR	Sportradar (free tier)	Trial (1k calls)	Schedules + post-race results only
WEC	Kaggle Historical Datasets	Public domain	2012–2023 lap data

✅ MVP Decision: Only F1 + MotoGP will have deep historical + simulated live. Others = schedule/results only.

---

4. Architecture

Supabase Schema (v1)

All tables include: `id`, `sport_id`, `created_at`, `updated_at`
RLS: Enforce `sport_id = current_sport()` for isolation.

Core Tables:

- `sports` (f1, motogp, wrc…)
- `events` (Grand Prix, Race Weekend)
- `sessions` (Practice, Qualifying, Race)
- `drivers`, `teams`

F1 Extensions:

- `f1_laps`, `f1_car_data`, `f1_weather`, `f1_pit`, `f1_stints`

MotoGP Extensions:

- `motogp_laps`, `motogp_sessions`

NestJS Backend

Next.js Frontend

- Dynamic routes: `/[sport]/schedule`
- Themed context: `useSportTheme()`
- V0-generated base components

---

5. Key Workflows (MVP)

Workflow 1: Simulated Live Session

1. On Oct 1, system replays 2023 Singapore GP in real-time.
2. Fetch lap data, telemetry, position from OpenF1.
3. Display “Live” badge only during session window.
4. Post-session: auto-generate results + Hall of Fame update.

Workflow 2: Driver Profile

1. User visits `/f1/drivers/1`
2. Load Max Verstappen bio, headshot, team, stats.
3. Pull career timeline from historical standings.

Workflow 3: Vehicle Specs

1. User visits `/f1/vehicle/specs`
2. Display curated engine, aero, weight data from F1 Technical.
3. Defer complex telemetry (e.g., DRS maps) to v2.

---

6. Sprint Timetable (13 Weeks)

Sprint	Dates	Focus	Deliverable
Sprint 0	Sept 26–30	Research & Setup	Finalized APIs, Supabase schema, monorepo
Sprint 1	Oct 1–11	F1 Data Pipeline	F1 2023–2025 ingested into Supabase
Sprint 2	Oct 14–25	MotoGP + Unified Schema	MotoGP data + shared `drivers`, `events`
Sprint 3	Oct 28–Nov 8	Schedule + History	`/f1/schedule`, `/motogp/history` live
Sprint 4	Nov 11–22	Drivers, Teams, Vehicle	Rich profiles with sport theming

Sprint 5	Nov 25–Dec 6	Results + Hall of Fame	Champions database, session results
Sprint 6	Dec 9–20	Live Simulation	Replayed sessions, global disclaimer
Sprint 7	Dec 22–31	QA + Launch	Performance audit, SEO, go-live

---

7. Permissions & Security

- Public Site: No auth required. All data read-only.
- RLS: Every query filtered by `sport_id`.
- PII: None stored. All data is public racing info.
- Backups: Daily Supabase snapshots.
- Disclaimer: Visible in footer + meta tags.

---

8. UI/UX Notes

- Design Language: Dark mode default, telemetry-inspired fonts (e.g., `Orbitron`, `Rajdhani`)
- Sport Theming:
  - F1: `#e10600` (red), black background
  - MotoGP: `#0066b3` (blue), silver accents
- Mobile-First: Responsive tables, swipeable session timelines
- V0 Integration: Generate base layouts for Schedule, DriverCard, LiveSimulator

---

9. Definition of Done (MVP)

✅ F1 + MotoGP data fully ingested (2023–2025)
✅ All 8 core pages live with sport theming
✅ Simulated “Live” mode functional

✅ Admin dashboard shows pipeline health
✅ Global disclaimer + no PII exposure
✅ Newsletter signup + Vercel deployment
✅ RLS enforced + backups enabled

---

10. Open Questions for Developers

1. Preferred deployment for NestJS: Render, Fly.io, or Supabase Edge?
2. Should we use Supabase Auth for future user accounts (v2)?
3. Any recommendations for MIT-licensed MotoGP scrapers beyond ParsaD23?
4. Preferred charting lib for lap time comparisons (Chart.js, Recharts)?
5. Should vehicle specs be markdown-based (easy curation) or structured JSON?

Founder's Note:
This isn't just a data dump—it's a tribute to the drama, engineering, and history of motorsport.
Let's build something fans will bookmark, share, and trust.
— *RACELOGIC Founder*