YN.md - Core Knowledge Base & Project Architecture

1. Project Vision (The Vibe)

Core Definition: VYN.md is a Next.js web platform designed to disrupt the Moldovan vehicle history report market. It acts as a premium, AI-driven "Car Buying Assistant" rather than just a raw data aggregator.

The Problem We Are Solving: Currently, Moldovan buyers use outdated websites (like Infocar.md or Vininfo.md) that charge money to generate long, confusing, and technical PDF tables. Users often don't understand what the data means (e.g., "Is a 'Salvage Title' bad?"). Furthermore, the checkout processes are clunky, and the UI looks like it was built in 2015.

The VYN Solution: We don't just show data; we interpret it. VYN aggregates data from multiple APIs (VinAudit, local Gov APIs) and feeds it into an LLM (OpenAI/Gemini). The LLM translates technical jargon into simple, actionable advice for the buyer (e.g., "Do not buy this car, it was flooded in 2021", or "Great car, but the bumper was replaced. Negotiate €500 down.").

UI/UX & Design Language (The "Perspective.co" Standard)

Cursor, when generating frontend components, you must strictly adhere to the following design system. Our benchmark for UI/UX is **https://www.perspective.co/**. We are building a high-converting, highly trusted responsive web platform, NOT a dark/cyberpunk developer tool.

- **Aesthetic & Inspiration:** Clean, modern, trustworthy, and heavily reliant on generous whitespace. It must feel like a premium consumer product (think Apple, Stripe, or Perspective.co).
- **Colors & Contrast:** * Default to a highly polished **Light Mode** (or a very soft, premium Dark Mode only if toggled). Use crisp white backgrounds (`bg-white`) for cards and soft gray (`bg-slate-50`) for the main canvas.
    - Use a single, vibrant, electric accent color (like an electric blue, vivid purple, or trust-inspiring emerald) exclusively for primary Call to Action (CTA) buttons, progress bars, and icons. Do NOT use harsh neon on pitch-black backgrounds.
- **Typography:** Mimic the Perspective.co font stack. Use a premium, geometric sans-serif font that looks incredible in bold weights (e.g., `Plus Jakarta`

`Sans`, `Inter`, or `Manrope`). Headings must be large, high-contrast, and tightly tracked.

- **Layout (Responsive Platform):**
  - The platform must be fully responsive. It must act as a frictionless mobile funnel for the 95% of traffic coming from TikTok/Instagram Reels, but expand elegantly into a full-featured dashboard on desktop monitors for B2B/Dealership users.
  - Use large, full-width, softly rounded components (`rounded-xl` or `rounded-2xl`).
  - Add extremely subtle drop shadows (`shadow-sm` or `shadow-md`) to cards to give depth, completely avoiding flat/boring UI.
- **Interactivity:** Use smooth, spring-like animations (e.g., Framer Motion). Elements should glide in gracefully. Use beautiful skeleton loaders and step-by-step transitions to keep the user engaged while APIs are fetching data.

Business Logic & Development Principles

- **The "Bali Solopreneur" Rule (100% Automation):** The system must run itself. There is zero human intervention. All flows—from VIN input, to Paynet Webhook confirmation, to API fetching, to AI generation, and final UI rendering—must be fully automated and error-handled.
- **Graceful Degradation:** If one API fails (e.g., the Gov API is down), the platform must NOT crash. It should gracefully inform the user ("Local data temporarily unavailable") while still delivering the VinAudit and AI analysis.
- **Instant Gratification (Loading States):** Because fetching from multiple APIs and prompting an LLM takes time (2-5 seconds), the frontend MUST use simulated progress bars and "AI is thinking..." animations to prevent bounce.

Internationalization (i18n) & Localization

The Moldovan market is multilingual. The platform must be built from day one with a robust internationalization framework (e.g., `next-intl` or Next.js middleware routing).

- **Supported Languages:**
  - **Romanian (ro) - PRIMARY/DEFAULT:** This is the main language of the platform and marketing.
  - **English (en) - SECONDARY:** For international users, expats, and the modern tech-savvy demographic.
  - **Russian (ru) - TERTIARY:** Essential for a massive portion of the local Moldovan and Transnistrian auto market.
- **Language Switching:** The UI must include a seamless, visually clean language toggle (e.g., simple text abbreviations like RO | EN | RU) that

updates the URL state and translates the entire funnel instantly without losing the user's progress or their inputted VIN.

2. Tech Stack & Architecture (The Engine Room)

This is your architectural blueprint. Do not deviate from this stack unless explicitly instructed. We are building for extreme speed, high mobile conversion, and zero-maintenance automation.

- **Frontend & Core Framework:** Next.js (App Router). Use Server Components (RSC) by default for lightning-fast page loads. Use Client Components (`"use client"`) ONLY where interactivity is required.
- **Styling & UI Ecosystem:** Tailwind CSS. Use `shadcn/ui` for accessible, unstyled baseline components, customized heavily to fit our Perspective.co clean aesthetic (rounded corners, soft borders, no harsh lines).
- **Animations:** Framer Motion for smooth UI transitions.
- **Icons:** `lucide-react` for clean, minimalist, consistent iconography.
- **Database & Backend:** Supabase (PostgreSQL). We will use the Supabase JS Client.
    - `orders` / `payments`: Tracks Paynet transaction ID, VIN, user contact (WhatsApp), and status (`pending`, `paid`, `abandoned`).
    - `reports`: Stores the VIN, raw JSON data from APIs, and the final generated AI Verdict.
- **Security:** Row Level Security (RLS) must be strictly enforced.
- **APIs & Server-Side Logic:** Next.js API Routes (`/app/api/...`) for Paynet Integration (Webhooks), Data Fetching (VinAudit), and AI Processing (OpenAI/Gemini).
- **Hosting:** Vercel for zero-config edge caching and instant deployments.

3. Data Sources & AI Integration (V1.0 MVP)

Take note: We are currently building Version 1.0 (The MVP). We will rely on a single, robust external data provider combined with an LLM for analysis.

- **3.1 Primary Data Source (VinAudit API):** Endpoint `https://www.vinaudit.com/vehicle-history-api`. It fetches the international history of the vehicle (Specs, Title Records, Salvage/Total Loss, Odometer Readings). The raw JSON is never shown directly to the user.
- **3.2 The "Brain" (AI Processing Layer):** We pass the structured JSON from VinAudit directly into an LLM (OpenAI `gpt-4o-mini` or Gemini) via a strict system prompt. The AI acts as the "Master Mechanic" and returns a clean JSON object containing:

    - **The AI Verdict:** A punchy, 3-sentence summary.
    - **Risk Score:** Green = Safe, Yellow = Caution, Red = Danger.
    - **Negotiation Tip:** A short text the user can use to negotiate the price down.
- **3.3 The Roadmap (V2.0 Scope - DO NOT BUILD IN V1.0):** Structure the code flexibly so we can plug in MConnect/e-Gov API and 999.md Matcher later without a major rewrite.

4. User Flow (The "Easy Go" Experience & Funnel Logic)

Cursor, build the UI state machine to follow this sequence strictly. The goal is zero friction and maximum conversion.

- **Step 1 (The Hook):** User arrives at `vyn.md`. They see a massive, clean input field in the center. They paste their 17-character VIN and tap "Scan Vehicle with AI".
- **Step 2 (The Teaser & Lead Capture):** Smooth transition to a "Scanning" view. The backend extracts basic data (Year, Make, Model) for free to prove it works. A clean modal slides up: *"We found historical records for this vehicle. Enter your WhatsApp/Phone number to securely link this report and proceed to payment."* This creates a `pending` row in Supabase.
- **Step 3 (The Checkout):** User sees a sleek payment summary card: *"Unlock Full AI Report: 350 MDL"*. Tapping "Pay Securely" redirects to the Paynet hosted checkout.
- **Step 4 (The Webhook & AI Engine):** User pays. Paynet fires a Webhook to Next.js. Next.js fetches VinAudit data, sends it to OpenAI, and saves the structured AI JSON to Supabase. Order status updates to `paid`.
- **Step 5 (The Delivery):** User is redirected back to `vyn.md/report/[order_id]` to view the stunning, interactive dashboard.

5. Unique AI Features & Prompt Engineering (The "Moat")

Cursor, this is the most critical logic of the V1.0 MVP.

- **5.1 The LLM Persona:** The AI must act as an expert, unbiased automotive inspector. Tone must be direct and protective. It must output in the language determined by the `locale` parameter (`ro`, `ru`, or `en`).
- **5.2 Strict Structured Output:** You must use OpenAI's Structured Outputs (or `response_format: { type: "json_object" }`).

    - Expected schema: `{ risk_level, verdict_summary, red_flags: [{title, description}], negotiation_script, estimated_ownership_cost }`.
- **5.3 UI Mapping:** * **Hero Section:** Dynamic coloring based on `risk_level` (emerald/amber/rose).
    - **Negotiator Card:** A dedicated clean card with a "Copy to Clipboard" button.
    - **Red Flags:** An accordion mapping through the `red_flags` array.

6. Competition & Our Advantage

We are building a high-converting, tech-forward consumer application. Every line of code must reflect that we are a generation ahead of our competitors.

- **The Local Dinosaurs (Infocar.md & Vininfo.md):** They sell static, boring PDF documents filled with technical tables on cluttered websites.
- **The VYN Unfair Advantages:**
    - **1. The UI/UX Moat:** We sell a beautiful, interactive, Perspective.co-style web platform. This builds instant trust.
    - **2. The AI Interpretation Moat:** We don't force users to interpret data. We give them a clear verdict and negotiation scripts.
    - **3. The Operational Moat:** 100% automated, zero headcount, allowing massive profit margins.
- **Cursor Directives:** Never use raw tables unless absolutely necessary. Use modern card layouts. Prioritize responsive design (perfect on mobile, expansive on desktop). Focus on Next.js Server Components for extreme loading speed.

7. Marketing, Scaling & Automation (The Growth Engine)

- **7.1 The "Abandoned VIN" Retargeting System:** The `orders` table tracks phone numbers and `pending` statuses. Build a Next.js API route (or Supabase Webhook) to identify carts abandoned for >30 mins, enabling future n8n integrations for WhatsApp recovery messages.
- **7.2 Pixel Tracking:** The frontend must include a global Event Tracker triggering `ViewContent`, `Search`, `InitiateCheckout`, and `Purchase` (passing the 350 MDL value).
- **7.3 Content Engine:** The UI must look incredibly clean when screen-recorded on a phone vertically (9:16) for TikTok marketing.
- **7.4 Affiliate B2B:** Include a clean CTA at the bottom of the report to "Hire a physical inspector" via a WhatsApp deep link.

8. The Live Dashboard & PDF Generation (The Delivery)

The moment the Paynet webhook confirms payment, the user sees this Live Dashboard. It must instantly justify the price tag through breathtaking UI.

- **8.1 Interactive Platform UI:** Use a soft background (`bg-slate-50` or `bg-gray-50`) with crisp white cards (`bg-white`) featuring soft drop shadows. No massive tables. Group data into clean `shadcn/ui` Accordions (Ownership History, Mileage, Damage).
- **8.2 The "Negotiator" Action Block:** A visually distinct card featuring the generated script and a large "Copy Message" button.
- **8.3 "Download PDF" Feature:** Crucial for Moldovan buyers. Implement a proper PDF generation flow (e.g., `html2pdf.js` or `puppeteer-core`). The PDF must feature a clean, white layout, the VYN.md logo, the AI Verdict, and a structured breakdown of the data. DO NOT just trigger `window.print()`.
- **8.4 Sticky Navigation:** On mobile, implement a sticky bottom CTA (e.g., "Find Alternatives on 999.md" or "Download PDF") so the user never feels lost.