# **Integrated MVP: Domain Valuation Agent (Core factors embedded)**

## **Overview**

This MVP centers the **valuation engine** (rule-based + AI-assisted) using the 10 core factors you approved. The app will:

* compute a transparent numeric score per factor,
* weight them (configurable in admin),
* map the final score into a price bracket and a suggested retail/investor price,
* surface the factor breakdown in UI and PDF reports,
* keep external dependencies minimal (no SEO backlink APIs).  
  Legal/trademark checks and purchase-history comps are optional integrations (comps allowed; backlink/Ahrefs style dependencies removed per your request).

# **1) Core Valuation Factors → Implementation & Scoring (MVP)**

For each factor I give: **what to measure**, **how to score (0–100)**, and **default weight** (weights sum to 1.0). Legal/trademark is a **gate/multiplier**, not a simple weight.

Default weights (editable in admin):  
Length 0.12, Keywords 0.20, TLD 0.15, Brandability 0.15, Industry 0.10, Comps 0.12, Age 0.06, Traffic 0.04, Liquidity 0.06 — **sum = 1.00**.  
Legal/trademark = gating multiplier (see below).

### **1. Length & Simplicity**

* **Measure:** domain label length (characters), word count, presence of hyphens/numbers.
* **Scoring rule (0–100):**
  + ≤ 6 chars → 100
  + 7–10 chars → 70
  + 10 chars → 40
  + Penalties: hyphen = −20, number = −10 (floor at 0).
* **Default weight:** 0.12

### **2. Keywords & Search Demand (MVP-safe)**

* **Measure:** exact-match vs partial match vs no dictionary words; presence of high-value industry keywords (from internal list).
* **Scoring (0–100):**
  + Exact single keyword .com in high-value industry → 95–100
  + Phrase with strong keywords → 70–90
  + No meaningful keyword → 20–50
* **Default weight:** 0.20
* **Note:** For MVP, search volume is optional user-supplied; external search-volume APIs are excluded.

### **3. TLD (Top-Level Domain)**

* **Measure:** TLD type and market context (is ccTLD aligned with user country?).
* **Scoring (0–100):**
  + .com → 100
  + .net/.org → 70
  + ccTLD aligned to target market → 60–85 (depends on country)
  + new gTLDs (.ai/.io/.xyz) → 40–70 depending on category
* **Default weight:** 0.15

### **4. Brandability (AI-assisted)**

* **Measure:** pronounceability, memorability, uniqueness. Use AI (GPT) to return a **brandability score 0–100** and short commentary.
* **Default weight:** 0.15

### **5. Industry & Market Relevance**

* **Measure:** which industry the domain maps to and whether that industry is high-value (internal industry ranking).
* **Scoring (0–100):**
  + High-value industry (finance, AI, health, travel, e-commerce, crypto) → 80–100
  + Niche/low-revenue categories → 30–60
* **Default weight:** 0.10

### **6. Comparable Sales (Comps) — allowed (purchase-history)**

* **Measure:** look up internal comps dataset or optional NameBio-style query. Compute similarity score to comps (keywords, TLD, length).
* **Scoring (0–100):** similarity-weighted value relative to median of matched comps. If no comps → neutral 50.
* **Default weight:** 0.12

### **7. Age & History**

* **Measure:** domain creation year / WHOIS age (optional lookup or user-supplied). Clean history = positive. Spam/penalty flags = manual review flag.
* **Scoring (0–100):**
  + Age > 8 years → 80–100
  + 3–8 years → 50–80
  + <3 years → 20–50
* **Default weight:** 0.06

### **8. Traffic (Minimal / Optional)**

* **Measure:** type-in traffic or analytics numbers (user-supplied) — MVP accepts user input.
* **Scoring (0–100):** normalized buckets (none → 20, small → 50, moderate → 70, high → 90).
* **Default weight:** 0.04

### **9. Liquidity**

* **Measure:** buyer pool estimate based on domain characteristics (one-word .com, 2-letter .com, numbers, non-dictionary).
* **Scoring (0–100):**
  + High liquidity (one-word .com, LL, LLL, 2-letter) → 90–100
  + Moderate → 50–80
  + Low (ultra-niche phrases) → 20–40
* **Default weight:** 0.06

### **10. Legal / Trademark Risks (Gating)**

* **Measure:** automated substring match vs internal “brand list” + user-declared checks. For MVP, do a lightweight substring match against a curated top-brand list.
* **Action:** If **severe risk detected**, set legalMultiplier = 0 (appraisal invalid / manual legal review). If **possible conflict**, legalMultiplier = 0.5. If clear, legalMultiplier = 1.0.
* **Note:** This is deliberately **conservative** and prompts manual review — we do **not** perform full USPTO searches in MVP.

# **2) Aggregation / Price Estimation Formula**

**Step 1:** compute weighted score:

final\_score = legalMultiplier × sum\_i(weight\_i × score\_i) // score\_i in [0,100]

**Step 2:** map final\_score to price bracket (example piecewise mapping):

* 0–20 → $50 – $500
* 20–40 → $500 – $1,500
* 40–60 → $1,500 – $10,000
* 60–80 → $10,000 – $100,000
* 80–100 → $100,000 – $5,000,000

**Step 3 (comps adjustment):** if a comps median is available, combine log-averages to produce a suggested retail price. (MVP: use a simple heuristic — blend final\_score bracket midpoint with comps median weighted 60/40 favoring comps if many comps found.)

# **3) Worked Example (digit-by-digit arithmetic)**

Domain: **CryptoWallet.com** (hypothetical; assume no legal flags)

Assigned factor scores (example):

* Length score = 40
* Keywords score = 95
* TLD score = 100
* Brandability = 80
* Industry = 95
* Comps similarity = 90
* Age = 60
* Traffic = 50
* Liquidity = 70  
  Legal multiplier = 1.0

Weights (repeated): Length 0.12, Keywords 0.20, TLD 0.15, Brandability 0.15, Industry 0.10, Comps 0.12, Age 0.06, Traffic 0.04, Liquidity 0.06

Compute contribution for each (multiply weight × score):

1. Length: 0.12 × 40 = 4.8
2. Keywords: 0.20 × 95 = 19.0
3. TLD: 0.15 × 100 = 15.0
4. Brandability: 0.15 × 80 = 12.0
5. Industry: 0.10 × 95 = 9.5
6. Comps: 0.12 × 90 = 10.8
7. Age: 0.06 × 60 = 3.6
8. Traffic: 0.04 × 50 = 2.0
9. Liquidity: 0.06 × 70 = 4.2

Now sum them precisely:

* Step A: 4.8 + 19.0 = 23.8
* Step B: 23.8 + 15.0 = 38.8
* Step C: 38.8 + 12.0 = 50.8
* Step D: 50.8 + 9.5 = 60.3
* Step E: 60.3 + 10.8 = 71.1
* Step F: 71.1 + 3.6 = 74.7
* Step G: 74.7 + 2.0 = 76.7
* Step H: 76.7 + 4.2 = 80.9

final\_score = 80.9 (out of 100). With our bracket mapping, this is in **80–100 → $100k – $5M** (ultra-premium bracket).

**Comps example (hypothetical):** suppose CryptoBank.com sold for $250k (median of close comps = $250k). Blend rule (MVP): retailPrice = 0.6 × compsMedian + 0.4 × bracketMidpoint.

* Bracket midpoint for $100k–$5M: compute midpoint approximately: ($100,000 + $5,000,000)/2 = $2,550,000.
* retailPrice = 0.6 × 250,000 + 0.4 × 2,550,000 = 150,000 + 1,020,000 = **$1,170,000** (this shows why comps matter strongly — large bracket ranges lead to wide variance).  
  **MVP note:** That large result reveals the need for more comps or narrower mapping. In practice the app will show: *final\_score, bracket, comps median*, and present a **recommended retail** (conservative) and **investor** price (lower). For this example, a more conservative suggested retail might be **$350k** and investor price **$175k**, if comps are trusted but bracket range is wide. The app should surface both numbers and explain the assumptions.

# **4) UI / UX: how factors appear to users (MVP)**

* **Single results page** shows:
  + Domain + immediate one-line value tag (e.g., Estimate: $175k – $400k)
  + **Score dial** (0–100) and bracket badge.
  + **Factor breakdown**: horizontal bars for each factor with exact score and weight shown (hover shows calculation).
  + **AI commentary**: 2–3 sentences explaining why the domain scored that way.
  + **Comps panel**: list of matched sales (if available) with date & price.
  + **Legal flag**: red/yellow/green and instructions if flagged.
  + **Override** controls (admin only) to tweak weights or mark manual adjustments.
  + **Download report** button (PDF).

# **5) API & Data Model (MVP)**

### **API endpoints (Next.js app-router API routes)**

* POST /api/appraise
  + Request body: { domain: "cryptowallet.com", options: { userTraffic?: number, country?: "NG", useComps?: true } }

Response (example):  
{

"domain":"cryptowallet.com",

"finalScore":80.9,

"bracket":"80-100",

"priceEstimate":{

"investor":"$175,000",

"retail":"$350,000",

"explanation":"See assumptions..."

},

"breakdown":[

{"factor":"Length","score":40,"weight":0.12,"contribution":4.8},

{...}

],

"comps":[ /\* optional comps \*/ ],

"legalFlag":"clear",

"aiComment":"Short explanation..."

}

* GET /api/comps?domain=cryptowallet.com → returns matched sales from internal dataset (MVP: CSV-backed DB).
* POST /api/report → generates downloadable PDF (server-side Puppeteer or headless renderer).

### **DB schema (simplified)**

* Appraisal table:
  + id, domain, createdAt, finalScore (float), breakdown (JSON), priceEstimate (JSON), comps (JSON), userId nullable
* Comps table: domain, soldPrice, soldDate, source
* Settings table: factorWeights JSON, legalBrandList JSON

# **6) Admin controls & configuration**

* Edit factor weights (live or per-tenant).
* Upload comps CSV to seed internal comps DB.
* Manage brand/trademark list used for substring checks.
* Toggle AI usage for brandability scoring (on/off).

# **7) Tests & Acceptance Criteria (MVP QA)**

* Unit tests for each scoring function (length, TLD, hyphen/number penalty).
* Integration test: POST /api/appraise returns timely JSON and includes expected fields.
* Manual validation: spot-check 20 sample domains vs known comps and human broker valuations.
* Legal gating: trigger manual review when legalFlag != clear.

# **8) Developer notes & edge-cases**

* **Wide bracket problem:** bracket ranges are large; rely on comps and admin-tuned weight to narrow outputs. Encourage users to supply comps and traffic when available.
* **Legal risk:** always err on the side of manual review — do not auto-sell names with possible trademark conflicts.
* **AI brandability:** use AI for score + human-readable commentary, but allow admins to override.