

H.A.R.M.O.N.I.™ — Full Algorithm Build Spec & Reference Implementation (Petsales.com.au)

Status: Patent pending • **Use:** Exclusive to Petsales.com.au • **Methodology:** Behavioural science + C-BARQ research • **Scope:** End-to-end specification + reference code to ship today (no omissions)

0) System Overview

- **Goal:** Match a user (Human Vector, 9D) with a pet listing (Pet Vector, 9D). Output a Match %, Risk Rating, and UX badges, apply visibility filters, and sort listings by score.
 - **Pillars:**
 - Deterministic questionnaire → trait mapping (0.0–1.0)
 - Dual similarity (Cosine + inverse Euclidean)
 - Per-trait absolute-difference scoring + weights
 - Risk filters (blocking + warnings)
 - UX rules (promotion thresholds, badges, exports)
 - Full customisation (styling, logos, mapping overrides)
-

1) Traits

1.1 Human Vector (H1–H9)

1. Predictability (H1)
2. Activity Level (H2)
3. Sociability (H3)
4. Reactivity (H4)
5. Sensory Tolerance (H5)
6. Availability (H6)
7. Training Consistency (H7)
8. Emotional Energy (H8)
9. Environmental Factors (H9)

1.2 Pet Vector (P1–P9)

1. Structure Need (P1)
2. Energy Requirement (P2)
3. Social Bonding Need (P3)
4. Sensitivity to Reactivity (P4)
5. Noise/Stimuli Tolerance (P5)
6. Dependence Level (P6)
7. Trainability Expectation (P7)
8. Emotional Demand (P8)

9. Environmental Load (P9)

Scale: All traits in [0.0, 1.0]. Pre-scaled for pets; computed for humans.

2) Questionnaire (DO NOT MODIFY ANY WORDING)

1. What best describes your work situation?

- Work from home / Remote worker
- Work from office
- Hybrid

2. How many hours per day are you away from home?

- Less than 5 hours
- 5-10 hours
- 10-20 hours

3. How would you describe your household?

- Single
- Couple

4. Do you have kids?

- Yes
- No

5. Do you currently have any pets?

- Yes - dog
- Yes - cat
- Yes - other
- No

6. How would you describe your household environment?

- High-energy chaos
- Quiet environment
- Somewhere in between

7. What's your living situation?

- Apartment
- House with small yard
- House with large yard
- Rural / acreage

8. How many hours per week do you work?

- 10-20 hours
- 20-30 hours
- 30-40 hours
- 40-50 hours

- 50+ hours
9. In terms of budgeting and spending, which best describes you?
- I save each pay cycle
 - I live pay cycle to pay cycle
 - I'm living off credit
10. Have you owned a pet before?
- Yes
 - No
11. Does anyone in your household have allergies to pets?
- Yes
 - No
12. What is your love language?
- Words of affirmation
 - Physical touch
 - Acts of service
 - Giving/receiving gifts
 - Quality time
13. Describe your ideal weekend:
- Out with your mates having a weekend bender
 - A casual dinner on Saturday then home to bed, with some exercise on Sunday
 - In bed for the weekend, away from everything
14. When frustrated, how do you communicate?
- I raise my voice
 - I communicate calmly
 - Bit of both - I might raise my voice when things get heated, but I usually calm down fairly quickly
15. Which of the following best describes you in a chaotic household situation?
- I shut down and need to be alone to process
 - I start formulating a plan of attack in my head to get things back to normal
 - dinner needs to be cooked, and I get on with it
 - I'm somewhere in between - I feel overwhelmed but try to keep it together and deal with things step by step
16. How do you approach commitment?
- I find it easy to commit to things I genuinely want
 - I often commit out of obligation because I want to make other people happy
 - Bit of both - depends on the situation and who's involved
 - Honestly? I hate commitment
17. In your interpersonal relationships, when things are continually hard or you keep running into the same issues, how do you respond?

- I do everything I can to make it work - I hang in there, even when it's a struggle
- I tend to shut down or walk away when it feels too hard or draining
- Depends on the relationship - sometimes I fight for it, sometimes I know it's time to move on

18. Pick the one that suits you best - on a daily basis, how would you describe your mood?

- Fairly stable - I'm pretty even most days
- Fluctuates slightly up and down - nothing major, just normal mood shifts
- Highly erratic - I can feel great one minute and suddenly hit a low like the world's ending

19. How would you describe your household noise level?

- Very noisy
- Very quiet
- Somewhere in between

20. Which best describes you when it comes to learning new skills or routines?

- I pick things up quickly and usually get the hang of it without much effort
- I can learn new things but need time, patience, and sometimes reminders
- I struggle with consistency or staying motivated when something feels repetitive or hard

21. How would you describe your social energy when you're at home?

- I love constant company and interaction - I feel best with someone nearby
- I like occasional company, but I also enjoy my personal space
- I value my space and quiet time - I recharge alone

22. Which one best describes where you're at with getting a pet?

- I've done the research and I'm ready to buy - I know what animal suits my lifestyle and personality
- I've only just started looking - still gathering info and figuring out what kind of pet I want
- I'm somewhere in between - I'm browsing, maybe making the occasional enquiry, but I'm not ready to commit just yet
- I'd love to get a pet, but my current circumstances don't allow it right now

3) Response → Human Vector Mapping (Deterministic)

Values are **scaled [0.0-1.0]**. Multiple traits may receive contributions from a single answer.

Q1 Work Situation → Availability (H6)

- Work from home: 1.0

- Hybrid: 0.6
- Work from office: 0.3

Q2 Hours away → Availability (H6)

- <5 hrs: 1.0
- 5–10 hrs: 0.5
- 10–20 hrs: 0.2

Q3 Household → Sociability (H3)

- Single: 0.3
- Couple: 0.7

Q4 Kids? → Environmental Factors (H9), Predictability (H1)

- Yes: H9=1.0, H1=0.3
- No: H9=0.3, H1=1.0

Q5 Current pets → Predictability (H1)

- Yes – dog: 1.0
- Yes – cat: 0.8
- Yes – other: 0.6
- No: 0.3

Q6 Household environment → Environmental Factors (H9), Sensory Tolerance (H5)

- High-energy chaos: H9=1.0, H5=0.2
- Somewhere in between: H9=0.6, H5=0.6
- Quiet environment: H9=0.2, H5=1.0

Q7 Living situation → Environmental Factors (H9)

- Rural/acreage: 1.0
- House with large yard: 0.8
- House with small yard: 0.5
- Apartment: 0.2

Q8 Work hours/week → Availability (H6)

- 10–20: 1.0
- 20–30: 0.8
- 30–40: 0.5
- 40–50: 0.3
- 50+: 0.1

Q9 Budgeting → Predictability (H1)

- Save each pay: 1.0
- Pay-to-pay: 0.5
- Credit: 0.2

Q10 Owned a pet → Training Consistency (H7)

- Yes: 1.0
- No: 0.3

Q11 Allergies → Sensory Tolerance (H5)

- Yes: 1.0
- No: 0.0

Q12 Love language → Sociability (H3), Emotional Energy (H8)

- Quality time: H3=1.0, H8=1.0
- Physical touch: H3=0.9, H8=0.9
- Acts of service: H3=0.8, H8=0.7
- Words of affirmation: H3=0.6, H8=0.8
- Giving/receiving gifts: H3=0.5, H8=0.4

Q13 Ideal weekend → Activity Level (H2), Emotional Energy (H8)

- Weekend bender: H2=1.0, H8=1.0
- Casual dinner + exercise: H2=0.7, H8=0.6
- In bed: H2=0.2, H8=0.3

Q14 When frustrated → Reactivity (H4)

- I communicate calmly: 0.2
- Bit of both: 0.5
- I raise my voice: 1.0

Q15 Chaos response → Reactivity (H4), Training Consistency (H7)

- Shut down: H4=1.0, H7=0.3
- Somewhere in between: H4=0.6, H7=0.6
- Plan of attack: H4=0.2, H7=1.0

Q16 Commitment → Training Consistency (H7)

- Easy to commit: 1.0
- Out of obligation: 0.6
- Depends: 0.4
- Hate commitment: 0.1

Q17 Hard relationships → Emotional Energy (H8)

- Hang in there: 1.0
- Shut down/walk away: 0.2
- Depends: 0.5

Q18 Daily mood → Emotional Energy (H8)

- Fairly stable: 1.0
- Fluctuates: 0.6
- Highly erratic: 0.2

Q19 Household noise → Sensory Tolerance (H5)

- Very quiet: 1.0
- Somewhere in between: 0.5
- Very noisy: 0.2

Q20 Learning style → Training Consistency (H7)

- Pick up quickly: 1.0
- Need reminders: 0.6
- Struggle with consistency: 0.2

Q21 Social energy at home → Sociability (H3)

- Constant company: 1.0
- Occasional company: 0.6
- Value space: 0.2

Q22 Pet readiness → Predictability (H1)

- Ready to buy: 1.0
- Just started: 0.6
- In between: 0.4
- Circumstances don't allow: 0.1

Aggregation rule: If multiple questions write to the same trait, compute the **mean** of contributing values per trait (or allow override to weighted mean via admin mapping overrides).

4) Vector Normalisation

- Ensure H and P are in comparable ranges. Human vector is already [0,1] by mapping; pet vector is pre-scaled [0,1]. Optionally apply L2 normalisation for cosine calculation.
-

5) Similarity Models (Dual layer)

• **Cosine Similarity:** $\cos_sim = \frac{\sum(H_i \times P_i)}{(\|H\| \times \|P\|)}$

• **Inverse Euclidean:**

• $distance = \sqrt{\sum(H_i - P_i)^2}$

• $euclidean_sim = 1 - (distance / \sqrt{9})$

6) Per-Trait Match & Weighting → Final Match %

• $TraitMatch_i = 1 - |H_i - P_i|^2$ (per trait)

• **Weights (sum = 1.0):**

• Activity Level: 0.15

• Reactivity: 0.15

• Predictability: 0.10

• Sociability: 0.10

• Sensory Tolerance: 0.10

• Availability: 0.10

• Training Consistency: 0.10

• Emotional Energy: 0.10

• Environmental Factors: 0.10

• **Final Match %** = $100 \times \sum(TraitMatch_i \times Weight_i)$ (float with 2 decimals)

7) Risk Filters & Visibility Rules

• **Blockers:**

• If H8 (Emotional Energy) < 0.4 → block **high-needs pets** (admin-flagged by high P8, e.g., P8 ≥ 0.7) from recommendations.

• If H6 (Availability) < 0.3 → block **high-dependence pets** (e.g., P6 ≥ 0.7).

• **Thresholds:**

• Match < 60% → listing hidden from **Matched to You**.

• Match < 50% → listing shown with **grey warning banner**.

• **Badges:**

• ≥ 80% → **Promoted + Matched ✓ badge**.

• 50-59.99% → **Warning ❌ banner**.

• **Breeder/Seller view:** anonymised average match metrics across engaged users.

8) Data Schemas (JSON)

8.1 Human Questionnaire Submission

```
{  
    "user_id": "string",  
    "answers": {  
        "q1": "Work from home / Remote worker",  
        "q2": "Less than 5 hours",  
        "q3": "Single",  
        "q4": "No",  
        "q5": "No",  
        "q6": "Quiet environment",  
        "q7": "Apartment",  
        "q8": "30-40 hours",  
        "q9": "I save each pay cycle",  
        "q10": "Yes",  
        "q11": "No",  
        "q12": "Quality time",  
        "q13": "A casual dinner on Saturday then home to bed, with some exercise on Sunday",  
        "q14": "I communicate calmly",  
        "q15": "I'm somewhere in between - I feel overwhelmed but try to keep it together and deal with things step by step",  
        "q16": "I find it easy to commit to things I genuinely want",  
        "q17": "Depends on the relationship - sometimes I fight for it, sometimes I know it's time to move on",  
        "q18": "Fluctuates slightly up and down - nothing major, just normal mood shifts",  
        "q19": "Somewhere in between",  
        "q20": "I can learn new things but need time, patience, and sometimes reminders",  
        "q21": "I like occasional company, but I also enjoy my personal space",  
        "q22": "I've only just started looking - still gathering info and figuring out what kind of pet I want"  
    }  
}
```

8.2 Human Vector

```
{  
    "user_id": "string",  
    "human_vector": {  
        "predictability": 0.0,  
        "activity_level": 0.0,  
    }  
}
```

```

    "sociability": 0.0,
    "reactivity": 0.0,
    "sensory_tolerance": 0.0,
    "availability": 0.0,
    "training_consistency": 0.0,
    "emotional_energy": 0.0,
    "environmental_factors": 0.0
  }
}

```

8.3 Pet Listing

```
{
  "pet_id": "string",
  "name": "Milo",
  "species": "dog",
  "breed": "Border Collie",
  "age_months": 14,
  "sex": "male",
  "location": "VIC",
  "pet_vector": {
    "structure_need": 0.8,
    "energy_requirement": 0.9,
    "social_bonding_need": 0.8,
    "sensitivity_to_reactivity": 0.6,
    "noise_stimuli_tolerance": 0.5,
    "dependence_level": 0.7,
    "trainability_expectation": 0.8,
    "emotional_demand": 0.7,
    "environmental_load": 0.6
  },
  "flags": {
    "high_needs": true,
    "high_dependence": true
  },
  "media": ["https://..."],
  "breeder_id": "string"
}
```

8.4 Match Result

```
{
  "user_id": "string",
  "pet_id": "string",
  "match_percentage": 93.72,
```

```

    "cosine_similarity": 0.94,
    "euclidean_similarity": 0.91,
    "trait_matches": {
        "activity_level": 0.92,
        "reactivity": 0.83,
        "predictability": 0.88,
        "sociability": 0.90,
        "sensory_tolerance": 0.95,
        "availability": 0.97,
        "training_consistency": 0.89,
        "emotional_energy": 0.91,
        "environmental_factors": 0.87
    },
    "risk_flags": ["high_needs_blocked", "high_dependence_blocked"],
    "badges": ["matched"],
    "visible": true,
    "promoted": true
}

```

9) Algorithm Pipeline (Deterministic)

1. **Map answers → trait values** using Section 3 constants.
2. **Aggregate per trait** (mean of contributing answers per trait).
3. **Compose $H = [H_1..H_9]$** and $P = [P_1..P_9]$.
4. **Compute Cosine and Euclidean Similarity.**
5. **Compute TraitMatch_i = $1 - |H_i - P_i|$** for $i \in \{1..9\}$.
6. **Apply Weights** and produce **Match %** to two decimals.
7. **Apply Risk Filters** (Emotional Energy/Availability blockers vs pet flags).
8. **Set Visibility/Badges** based on thresholds.
9. **Return Match Result object.**

10) Reference Implementation (TypeScript)

```

export type HumanVector = {
    predictability: number; // H1
    activity_level: number; // H2
    sociability: number; // H3
    reactivity: number; // H4
    sensory_tolerance: number; // H5
    availability: number; // H6
    training_consistency: number; // H7
    emotional_energy: number; // H8
    environmental_factors: number; // H9
}

```

```

};

export type PetVector = {
  structure_need: number; // P1
  energy_requirement: number; // P2
  social_bonding_need: number; // P3
  sensitivity_to_reactivity: number; // P4
  noise_stimuli_tolerance: number; // P5
  dependence_level: number; // P6
  trainability_expectation: number; // P7
  emotional_demand: number; // P8
  environmental_load: number; // P9
};

export type Pet = {
  pet_id: string;
  pet_vector: PetVector;
  flags?: { high_needs?: boolean; high_dependence?: boolean };
};

const WEIGHTS: Record<keyof HumanVector, number> = {
  activity_level: 0.15,
  reactivity: 0.15,
  predictability: 0.10,
  sociability: 0.10,
  sensory_tolerance: 0.10,
  availability: 0.10,
  training_consistency: 0.10,
  emotional_energy: 0.10,
  environmental_factors: 0.10,
};

function clamp01(x: number) { return Math.max(0, Math.min(1, x)); }

export function l2Norm(v: number[]): number {
  return Math.sqrt(v.reduce((s, x) => s + x * x, 0));
}

export function cosineSimilarity(H: number[], P: number[]): number {
  const dot = H.reduce((s, h, i) => s + h * P[i], 0);
  const nH = l2Norm(H); const nP = l2Norm(P);
  if (nH === 0 || nP === 0) return 0;
  return dot / (nH * nP);
}

export function euclideanSimilarity(H: number[], P: number[]): number {
  const dist = Math.sqrt(H.reduce((s, h, i) => s + (h - P[i]) ** 2, 0));
  return 1 - dist / Math.sqrt(9);
}

```

```

}

export function toArrayH(h: HumanVector): number[] {
  return [
    h.predictability,
    h.activity_level,
    h.sociability,
    h.reactivity,
    h.sensory_tolerance,
    h.availability,
    h.training_consistency,
    h.emotional_energy,
    h.environmental_factors,
  ];
}

export function toArrayP(p: PetVector): number[] {
  return [
    p.structure_need,
    p.energy_requirement,
    p.social_bonding_need,
    p.sensitivity_to_reactivity,
    p.noise_stimuli_tolerance,
    p.dependence_level,
    p.trainability_expectation,
    p.emotional_demand,
    p.environmental_load,
  ];
}

export function traitMatches(h: HumanVector, p: PetVector): Record<keyof HumanVector, number> {
  const map: (keyof HumanVector)[] = [
    'predictability', 'activity_level', 'sociability', 'reactivity', 'sensory_tolerance', 'availability',
    'structure_need', 'energy_requirement', 'social_bonding_need', 'sensitivity_to_reactivity', 'noise_stimuli_tolerance'
  ];
  const petMap: (keyof PetVector)[] = [
    'predictability', 'activity_level', 'sociability', 'reactivity', 'sensory_tolerance', 'availability',
    'structure_need', 'energy_requirement', 'social_bonding_need', 'sensitivity_to_reactivity', 'noise_stimuli_tolerance'
  ];
  const out: any = {};
  for (let i = 0; i < map.length; i++) {
    const ht = h[map[i]]; const pt = p[petMap[i]];
    out[map[i]] = clamp01(1 - Math.abs(ht - pt));
  }
  return out;
}

```

```

export function finalMatchPercent(h: HumanVector, p: PetVector): { match: number; cosine: number; euclid: number; trait: Record<keyof HumanVector, number> } {
  const H = toArrayH(h), P = toArrayP(p);
  const cosine = cosineSimilarity(H, P);
  const euclid = euclideanSimilarity(H, P);
  const trait = traitMatches(h, p);
  const score = (Object.keys(trait) as (keyof HumanVector>[]).reduce((s, k) =>
s + trait[k] * WEIGHTS[k], 0);
  return { match: +(score * 100).toFixed(2), cosine: +cosine.toFixed(4),
euclid: +euclid.toFixed(4), trait };
}

export function riskFilters(h: HumanVector, pet: Pet) {
  const blocks: string[] = [];
  if (h.emotional_energy < 0.4 && pet.flags?.high_needs)
blocks.push('high_needs_blocked');
  if (h.availability < 0.3 && pet.flags?.high_dependence)
blocks.push('high_dependence_blocked');
  return blocks;
}

export function badgesAndVisibility(matchPercent: number): { badges: string[]; visible: boolean; promoted: boolean; banner?: 'warning' } {
  const badges: string[] = [];
  let visible = true; let promoted = false; let banner: 'warning' | undefined;
  if (matchPercent < 60) visible = false;
  if (matchPercent < 50) { visible = true; banner = 'warning'; }
  if (matchPercent >= 80) { promoted = true; badges.push('matched'); }
  return { badges, visible, promoted, banner };
}

export function evaluate(h: HumanVector, pet: Pet) {
  const { match, cosine, euclid, trait } = finalMatchPercent(h, pet.pet_vector);
  const risk_flags = riskFilters(h, pet);
  const { badges, visible, promoted, banner } = badgesAndVisibility(match);
  return { match_percentage: match, cosine_similarity: cosine,
euclidean_similarity: euclid, trait_matches: trait, risk_flags, badges,
visible, promoted, banner };
}

```

11) Admin Customisation

- **Logos/Images:** upload per page; stored with tenant/site settings.
- **Styling:** CSS variables or theme tokens (fonts, colours, spacing).

- **Trait mapping overrides:** JSON config allowing value remaps per answer → trait(s); allow weighted means.
 - **Formula toggles:** enable/disable cosine/euclidean display; choose primary ranking (default: Final Match %).
 - **Export:** Figma-ready frames (JSON → plugin), Webflow embed (script tag + div mount), PDF export for profiles.
-

12) API Endpoints

- POST /api/harmoni/answers → { human_vector }
- POST /api/harmoni/match?user_id=... → [MatchResult*] (across pet inventory)
- POST /api/harmoni/match/:pet_id → MatchResult (single pet)
- GET /api/harmoni/user/:id/vector
- GET /api/harmoni/pets (with filters: species, breed, flags)
- POST /api/harmoni/export (PDF/JSON)

Authentication: JWT with tenant scoping. **Rate limits:** IP + token. **Audit:** store match requests for analytics.

13) UI/UX Rules

- **Result card:** Match %, Cosine, Euclid, per-trait bars, badges, warnings, Save/Share.
 - **Sorting:** by Match % desc; secondary sort by Euclidean Similarity desc.
 - **Visibility:** apply rules before render; grey warning banner if <50%.
 - **Breeder dashboard:** anonymised averages, not raw user vectors.
-

14) Persistence & Privacy

- Store Human Vector separate from PII; link via user_id only.
 - Keep questionnaire answers to allow re-scoring when mappings update.
 - Log versioned mapping/config for reproducibility.
-

15) Example Pet Vector Cookbook (Guidance)

- **High-energy working dog (e.g., Border Collie):** P2≈0.9, P6≈0.7, P7≈0.8, P8≈0.7, P9≈0.6
- **Low-energy cat:** P2≈0.3, P6≈0.3, P7≈0.5, P8≈0.4, P9≈0.2
- **Noise-sensitive small breed:** P5≈0.3, P4≈0.7

These are examples; actual values come from breeder intake + vet notes + behaviour history.

16) QA & Edge Cases

- **Incomplete questionnaire:** block scoring until all required answers present.
 - **Zero vectors:** if any norm = 0 → cosine = 0 safeguard.
 - **Rounding:** final % to 2 decimals; intermediate calcs at double precision.
 - **Ties:** stable sort by pet_id.
 - **Config drift:** re-score on config change; keep `config_version` with results.
-

17) Embedding (Webflow/Figma)

- **Webflow:** load `/harmoni.js`, mount `<div id="harmoni-app"></div>`; theme via CSS variables.
 - **Figma:** export profile cards as JSON; plugin reads trait bars + badges into frames.
-

18) IP & Branding (Do Not Modify)

- Name: **H.A.R.M.O.N.I.™**
 - Status: **Patent pending**
 - Use case: **Exclusive to Petsales.com.au**
 - Methodology: **Behavioural science + C-BARQ research**
 - Positioning: **First-of-its-kind globally**
-

19) Ready-To-SHIP Checklist

-

All required pieces are above with no omissions.

20) Design Integration (Pick Your Track)

You already have the design. Here are **drop-in paths** to wire it up today. Use whichever matches your setup. All three can coexist.

20.1 Webflow (Embed Script + Attributes)

HTML (add to your page):

```
<div id="harmoni-app"
  data-font-family="Inter, system-ui, -apple-system, Segoe UI, Roboto,
  Helvetica, Arial, sans-serif"
```

```

    data-color-bg="#ffffff"
    data-color-card="#0b0b0c"
    data-color-text="#111111"
    data-color-accent="#4865ff"
    data-radius="16"
    data-shadow="0 10px 30px rgba(0,0,0,0.08)">
</div>
<script src="/harmoni.embed.js" defer></script>

```

/public/harmoni.embed.js (vanilla, no deps):

```

(function(){
  const el = document.getElementById('harmoni-app');
  if(!el) return;
  const css = {
    ff: el.dataset.fontFamily || 'Inter, system-ui',
    bg: el.dataset.colorBg || '#fff',
    card: el.dataset.colorCard || '#0b0b0c',
    text: el.dataset.colorText || '#111',
    accent: el.dataset.colorAccent || '#4865ff',
    radius: `${el.dataset.radius} || 16}px`,
    shadow: el.dataset.shadow || '0 10px 30px rgba(0,0,0,0.08)'
  };

  function bar(label, value){
    const wrap = document.createElement('div');
    wrap.style.margin = '8px 0';
    const l = document.createElement('div');
    l.textContent = `${label} ${(value*100).toFixed(0)}%`;
    l.style.font = `500 12px/1 ${css.ff}`;
    l.style.color = css.text;
    const track = document.createElement('div');
    track.style.height = '8px';
    track.style.borderRadius = css.radius;
    track.style.background = 'rgba(0,0,0,0.08)';
    const fill = document.createElement('div');
    fill.style.height = '8px';
    fill.style.width = `${(value*100).toFixed(0)}%`;
    fill.style.borderRadius = css.radius;
    fill.style.background = css.accent;
    track.appendChild(fill);
    wrap.appendChild(l, track);
    return wrap;
  }

  function card(result){

```

```

const c = document.createElement('div');
c.style.background = css.bg;
c.style.color = css.text;
c.style.borderRadius = css.radius;
c.style.boxShadow = css.shadow;
c.style.padding = '20px';
c.style.maxWidth = '720px';
c.style.margin = '20px auto';

const h = document.createElement('div');
h.textContent = `Match ${result.match_percentage}%`;
h.style.font = `700 22px/1.2 ${css.ff}`;

const meta = document.createElement('div');
meta.style.font = `400 12px/1 ${css.ff}`;
meta.style.opacity = '0.7';
meta.textContent = `cos ${result.cosine_similarity} • euclid ${result.euclidean_similarity}`;

const traits = document.createElement('div');
traits.style.marginTop = '12px';

const order =
['activity_level', 'reactivity', 'predictability', 'sociability', 'sensory_tolerance', 'availability',
order.forEach(k => traits.appendChild(bar(k.replace(/_/g, ' '),
result.trait_matches[k])));

const badge = document.createElement('div');
badge.style.marginTop = '12px';
if(result.promoted) badge.textContent = 'Matched ✓';
if(result.banner === 'warning') badge.textContent = '⚠ Low match';

c.append(h, meta, traits, badge);
return c;
}

async function run(){
// Demo: replace with your API call
const demo = await fetch('/api/harmoni/demo').then(r=>r.json());
const node = card(demo);
el.replaceWith(node);
}
run();
})();

```

20.2 Next.js/Node (API + Demo)

/pages/api/harmoni/demo.ts (mock until you wire real data)

```
import type { NextApiRequest, NextApiResponse } from 'next';
export default function handler(req: NextApiRequest, res: NextApiResponse){
  res.status(200).json({
    match_percentage: 86.32,
    cosine_similarity: 0.9321,
    euclidean_similarity: 0.9012,
    promoted: true,
    banner: undefined,
    trait_matches: {
      activity_level: 0.92, reactivity: 0.83, predictability: 0.88,
      sociability: 0.90,
      sensory_tolerance: 0.95, availability: 0.97, training_consistency: 0.89,
      emotional_energy: 0.91, environmental_factors: 0.87
    }
  });
}
```

/pages/api/harmoni/match.ts (evaluate a specific pet)

```
import type { NextApiRequest, NextApiResponse } from 'next';
import { evaluate } from '@/lib/harmoni'; // from Section 10 code
import { getHumanVector, getPetById } from '@/lib/data';

export default function handler(req: NextApiRequest, res: NextApiResponse){
  const { user_id, pet_id } = req.query;
  const h = getHumanVector(String(user_id));
  const pet = getPetById(String(pet_id));
  const result = evaluate(h, pet);
  res.status(200).json(result);
}
```

20.3 React Component (Design-Token Friendly)

```
import * as React from 'react';

export type HarmoniTheme = {
  fontFamily?: string; bg?: string; text?: string; accent?: string;
  radius?: number; shadow?: string;
};
```

```

export function HarmoniCard({ result, theme = {} as HarmoniTheme }){
  const t = {
    fontFamily: 'Inter, system-ui, -apple-system, Segoe UI, Roboto, Helvetica,
    Arial, sans-serif',
    bg: '#fff', text: '#111', accent: '#4865ff', radius: 16, shadow: '0 10px
    30px rgba(0,0,0,0.08)',
    ...theme
  };
  const wrap: React.CSSProperties = {
    fontFamily: t.fontFamily, background: t.bg, color: t.text,
    borderRadius: t.radius, boxShadow: t.shadow, padding: 20, maxWidth: 720,
    margin: '20px auto'
  };
  const row = (label:string, v:number) => (
    <div style={{ margin: '8px 0' }} key={label}>
      <div style={{ fontWeight: 500, fontSize: 12 }}>{label}
      {(v*100).toFixed(0)}%</div>
      <div style={{ height: 8, borderRadius: t.radius, background:
      'rgba(0,0,0,0.08)' }}>
        <div style={{ height: 8, width: `${(v*100).toFixed(0)}%`, borderRadius:
      t.radius, background: t.accent }} />
      </div>
    </div>
  );
  const order =
  ['activity_level', 'reactivity', 'predictability', 'sociability', 'sensory_tolerance', 'availability'];
  return (
    <div style={wrap}>
      <div style={{ fontWeight: 700, fontSize: 22 }}>Match
      {result.match_percentage}%</div>
      <div style={{ opacity: .7, fontSize: 12 }}>cos {result.cosine_similarity}
      euclid {result.euclidean_similarity}</div>
      <div style={{ marginTop: 12 }}>{order.map(k => row(k.replace(/_/g, ' '),
      result.trait_matches[k]))}</div>
      {result.promoted && <div style={{ marginTop: 12 }}>Matched ✓</div>}
      {result.banner === 'warning' && <div style={{ marginTop: 12 }}>⚠️ Low
      match</div>}
    </div>
  );
}

```

How to theme to your design:

```

<HarmoniCard result={result} theme={{
  fontFamily: 'Your Brand Font, Inter, system-ui',
  bg: '#0b0b0c', text: '#f6f7f9', accent: '#00E6A7', radius: 20,

```

```
    shadow: '0 12px 40px rgba(0,0,0,0.25)'  
  }}/>
```

Notes: - All components are token-driven; you can plug in your design's fonts, radii, spacing, and colours without changing logic. - If your design uses Tailwind/shadcn, wrap this with your components or convert styles to classes. - Webflow users: tweak the `data-` attributes to match brand tokens; no code changes needed.