A. SCENARIO METADATA SECTION (New)

Add hidden structured fields:

- Scenario ID (UUID)
- Cycle Day (1–28)
- Age group (optional)
- Historical trend flag (true/false)
- Stress level (1–10)
- Baseline cycle length
- Previous 3-day sleep average
- Previous 3-day symptom average

These allow reproducibility.

B. STATE VECTOR PANEL (Auto-Computed, Displayed)

Add a visible box showing:

Computed Biological State:

- Estrogen Influence:
- Progesterone Influence:
- Energy Stability:
- Emotional Volatility:
- Inflammation Likelihood:
- Cognitive Load Score (optional)

Display numeric values (0–1).

This is critical for research transparency.

C. MODEL CONFIGURATION PANEL (Missing)

Add:

- LLM Provider (Gemini / GPT / Local)
- Temperature
- Top-p
- Max tokens
- Seed (for reproducibility)

Without this, reviewers will reject for lack of control.

D. RESPONSE ANALYSIS PANEL (New)

After generation, auto-compute:

- Word count
- Readability score
- Sentiment polarity
- Embedding vector
- Constraint violation flag (true/false)

These support quantitative metrics.

E. AUTOMATED METRIC OUTPUT

Add:

- Semantic similarity vs baseline
- State-alignment score
- Recommendation category tags (rest / diet / exercise / emotional support)

This allows alignment frequency calculation.

2️⃣ Required Backend Architecture

Minimal research architecture:

Frontend (Antigravity UI)
→ API Gateway
→ Scenario Processor
→ State Vector Engine
→ Prompt Builder
→ LLM API
→ Response Analyzer
→ Database

3️⃣ APIs You Need

You need 4 categories of APIs.

---

1. LLM API

Gemini is fine for research-level conference submission.

Use:

- Gemini 1.5 Pro (if available)
- Or Gemini 1.0 Pro

It is good enough for:

- Structured reasoning
- Health explanation
- Controlled prompt injection

For publication:
Mention:
“Commercially available large language model (Gemini Pro).”

No issue.

---

2. Embedding API

For semantic differentiation metric:

You need embeddings.

Options:

- Gemini embeddings
- OpenAI embedding API
- Local model (e.g., sentence-transformers)

Embedding API required for:
Cosine similarity measurement.

---

3. Database API

Use:

- PostgreSQL (recommended)
- Supabase
- Firebase Firestore

You need persistent storage.

---

4. Statistical Processing

You can:

- Export CSV
- Use Python (SciPy, Pandas)
- Or integrate a small stats microservice

Not mandatory real-time.

4️⃣ Database Schema (Exact)

You need 5 tables.

Table 1: scenarios

- scenario_id (UUID)
- phase
- mood
- energy
- sleep
- stress
- symptoms_json
- cycle_day
- created_at

Table 2: state_vectors

- scenario_id
- estrogen_influence
- progesterone_influence
- energy_stability
- emotional_volatility
- inflammation_likelihood
- computed_at

Table 3: generations

- generation_id
- scenario_id
- strategy_type
- model_name
- temperature
- prompt_text
- output_text
- word_count
- created_at

Table 4: evaluations

- generation_id
- evaluator_id
- relevance_score
- specificity_score
- biological_grounding_score
- personalization_score
- safety_score
- timestamp

Table 5: automated_metrics

- generation_id
- semantic_distance
- alignment_score
- violation_flag
- sentiment_score

This makes your research reproducible.

5️⃣ State Vector Computation Logic (Exact Version)

Normalize sliders:

E = energy / 10
S = sleep / 10
St = stress / 10

Symptom score:

Assign weights:

cramps = 0.3
back pain = 0.2
headache = 0.2

bloating = 0.1
others = 0.1

Inflammation = sum(weighted symptoms)

Energy Stability:

Energy_Stability = 0.6E + 0.4S - 0.3St

Emotional Volatility:

Map mood:
Calm = 0.2
Neutral = 0.4
Irritable = 0.7
Severe mood swings = 0.9

Then:
Volatility = mood_score + (1 - S)*0.3

Hormonal influence:

Based on phase:

Menstrual:
Estrogen = 0.2
Progesterone = 0.2

Follicular:
Estrogen = 0.6
Progesterone = 0.3

Ovulatory:
Estrogen = 0.9
Progesterone = 0.2

Luteal:
Estrogen = 0.5
Progesterone = 0.8

All clipped between 0–1.

That is your deterministic biological state vector.

---

6 Exact Prompt Templates

Now the important part.

Strategy 1: Generic

System:
You are a general wellness assistant.

User:
User reports:
Mood: {mood}
Energy: {energy}/10
Sleep: {sleep}/10
Symptoms: {symptoms}

Provide helpful advice.

Strategy 2: Phase-Aware

System:
You are a menstrual health–aware AI assistant. Ground recommendations in hormonal physiology.

User:
Menstrual Phase: {phase}
Mood: {mood}
Energy: {energy}/10
Sleep: {sleep}/10
Symptoms: {symptoms}

Explain biological context briefly and give structured recommendations.

Strategy 3: Phase + Memory-Aware

System:
You are a menstrual health AI that considers historical patterns and hormonal cycles.

User:
Menstrual Phase: {phase}
Current State:
Mood: {mood}
Energy: {energy}/10
Sleep: {sleep}/10
Symptoms: {symptoms}

Historical Pattern:
{history_summary}

Provide biologically grounded and context-adaptive recommendations.

Strategy 4: Phase + State Vector (Confirmed Research Version)

System:
You are an advanced menstrual health AI system. Use the provided biological state profile quantitatively. Recommendations must align with the physiological indicators.

User:
Menstrual Phase: {phase}

Biological State Profile:
Estrogen Influence: {He}
Progesterone Influence: {Hp}
Energy Stability: {Es}
Emotional Volatility: {Ve}
Inflammation Likelihood: {Il}

Provide:

1. Brief physiological interpretation (2–3 sentences).
2. Adaptive recommendations explicitly aligned with the state indicators.
3. Avoid generic advice.

---

7 Is Gemini Enough?

Yes.

For conference-level research:
Gemini Pro is sufficient.

You are evaluating prompting strategies,
not model architecture.

Model choice is not your novelty.

Just fix:

- Temperature = 0.4 (controlled)
- Top_p = 0.9
- Keep same config across strategies

---

8 What Makes This Publishable

Your console must now:

- Log everything
- Store prompts
- Store outputs
- Store model config
- Store state vectors
- Allow CSV export
- Enable blind evaluation

Right now your UI is missing:

- State vector display
- Model configuration logging
- Automated metrics logging

Add those.