

Technical Evaluation Prompt for Dab Skin Care App

Our Mission and Context

Dab's mission is to democratize expert skincare advice by delivering the best, science-backed guidance—accessible to everyone, regardless of location, language, or income.

We collect real-world data (selfies, before/after photos, routine logs, survey responses, and environmental inputs) to generate personalized “Skin Stories” with a Glow Score and actionable recommendations. Our approach models reality to reduce uncertainty, empower users, and make our service indispensable.

Key Differentiators:

- **Personalized, Data-Driven Guidance:** Aggregates diverse inputs into a “current state” vector that drives a dynamic, evolving Skin Story.
- **Low-Friction, Engaging UX:** Simple product logging (icon taps), before/after photo capture, and an interactive card stack UI.
- **Educational Onboarding & Trial-Only Model:** AI-driven onboarding that educates users, collects essential data, and requires credit card entry (via Stripe) to start a trial.
- **Premium Features:** History Dashboard and Personalization Chat (dermatologist-style Q&A) are premium.
- **Scalable Architecture:** Experience tuples—(Current State, Generated Advice, Observed Glow Score, User Feedback)—are stored in a vector database for continuous, rule-based personalization.
- **Advanced AI Integration:** We expect candidates to leverage AI innovatively. We value the integration of intent resolution, axiomatic reasoning, and adaptive memory. The final submission must include a dedicated section explaining your AI workflow (including which model(s) you used, your prompt strategy, and how AI output was integrated into your design).

Your Task

Select **one shared component** you have built before and reframe its design for our skincare domain (choose from one of the following):

- Authentication
- Payments
- Notifications
- Caching

Your deliverable must include the following sections:

1. **Component Selection & Context:**

- **Original Context:** Briefly describe the component you built, the business problem it solved, and its key requirements.
- **Domain Adaptation:** Explain how you would adapt this design for Dab. Consider skincare-specific needs such as managing before/after photo processes, routine logging (via simple icon taps), and integrating environmental data (UV index, humidity, temperature, pollution). Address how this component would support our educational onboarding and trial-only payment flows.

2. **Architecture Diagram:**

- Provide a high-level architecture diagram showing how your component integrates with:
 - Data aggregation (current state formation)
 - LLM-based narrative generation (Skin Story with Glow Score)
 - The feedback loop (experience tuple storage)
 - Shared components (authentication, payments, etc.)
- Clearly label data flows, API calls, and integration points.

3. **Pseudocode / Code Snippets:**

- Include pseudocode or code snippets outlining the main flow for your component.
- Use comments to explain key steps and demonstrate how data is processed and exchanged with other modules.

4. **Design Considerations, Tradeoffs, & Risks:**

- **Design Considerations:** Describe your architectural choices in the skincare context and why they are optimal.
- **Tradeoffs:** Explain compromises between simplicity and functionality, or performance and scalability.
- **Risks & Bottlenecks:** Identify potential risks (e.g., inconsistent photo quality, variable user engagement, data aggregation challenges) and propose mitigation strategies.
- **Scalability:** Discuss potential scaling bottlenecks and suggest strategies to overcome them.
- **Security & Compliance:** Address security concerns (especially for sensitive user data and payments) and relevant compliance (GDPR/CCPA).

5. **Integration of AI:**

- Describe how you used AI to answer this prompt (which LLM model did you use, what was your prompt strategy, and how did you incorporate the output into your deliverable?).
- Explain how you would integrate AI into your component. For example, using an LLM to analyze user feedback, optimize data caching, or enhance routine logging.
- Emphasize future-proofing—how your solution will adapt to AI advancements.

6. **Assumptions & Ambiguities:**

- List any assumptions you're making regarding the system, data quality, or user behavior.
- Identify ambiguous areas that may require further clarification.

7. Documentation & Presentation:

- Your submission should be a single document (PDF or slide deck) with clearly labeled sections: Architecture Diagram, Pseudocode/Code, Design Considerations, Tradeoffs, Risks, Scalability, Security/Compliance, and AI Integration.
- Responses may be in English or Spanish, but clarity and logical structure are essential.

Time Allotted:

You have 2–4 hours for this challenge. You will be compensated for the time spent.