ChatGPT

Product Requirements Document (PRD) - Updated

Project Overview

We are developing a **mobile-first web app** designed for a one-time use during a customer event. Users will access the site via a **QR code** and engage with a **wizard-like flow** to answer a set of 5 questions. Data from each step will be stored **locally** and also sent via a **Supabase backend**. If Supabase fails, the flow must continue seamlessly.

Following the questions, a **results screen** will display two summary tables: - The first table shows projected patient numbers. - The second table shows monthly revenue and profit. - A third table shows annualized figures based on the monthly data.

Styling should follow the **Sequel Brand Guidelines**.

Goals and Objectives

- Create a seamless, mobile-optimized, user-friendly experience.
- Capture all user inputs at each step, store locally, and submit via POST to Supabase.
- Display dynamic, easy-to-read summary tables with relevant calculations.
- Ensure brand consistency with Sequel's guidelines.
- Allow users to **restart the flow or edit answers at any time**.
- Capture and store **browser, device, OS, IP address, and user agent information** for analytics and traceability.

Functional Requirements

1. User Flow

- **Step 1:** Practice Name (free text, no validation)
- **Step 2:** Monthly Comprehensive Exams (numeric input, max 4 digits)
- **Step 3:** Optical Conversion Rate (numeric, max 100)
- **Step 4:** % Cash Pay in Optical Business (numeric, max 100)
- **Step 5:** MVC Patient Cash Pay Conversion (numeric, max 100)
- **Step 6:** Summary Results:
    - **Header:** "By adding Sequel to your lens offerings alongside Neurolens, you can expect the following:"
    - **First Table:**
        - Columns: Cash Pay Patients, MVC Patients, Monthly Orders
        - Rows: Sequel, Neurolens
    - **Second Table:**
        - Columns: Monthly Revenue, Monthly Profit

        - Rows: Sequel, Neurolens, Total
    - **Third Table:**
        - Same structure as Second Table but Annualized (monthly * 12)

2. Calculations

- **Sequel:**
    - Cash Pay Patients = ((#exams * conversion rate) * 0.6) * cash pay %
    - MVC Patients = ((#exams * conversion rate) * 0.6) * MVC conversion %
    - Monthly Orders = Cash Pay Patients + MVC Patients
- **Neurolens:**
    - Cash Pay Patients = ((#exams * conversion rate) * 0.3) * cash pay %
    - MVC Patients = ((#exams * conversion rate) * 0.3) * MVC conversion %
    - Monthly Orders = Cash Pay Patients + MVC Patients
- **Revenue & Profit:**
    - Sequel: $460 revenue, $247 profit per order
    - Neurolens: $800 revenue, $427 profit per order
    - Multiply revenue & profit per order by Monthly Orders
    - Annualize by multiplying monthly figures by 12

3. Supabase Integration

- Use **Supabase** as the backend to store data.
- Each user input step sends a **POST request** to a Supabase endpoint.
- The app proceeds even if Supabase returns an error.
- **Supabase Table Schema:**
    - `id`: UUID (Primary Key, auto-generated)
    - `practice_name`: Text
    - `comprehensive_exams`: Integer
    - `optical_conversion_rate`: Numeric
    - `cash_pay_percentage`: Numeric
    - `mvc_conversion_percentage`: Numeric
    - `browser`: Text
    - `device`: Text
    - `os`: Text
    - `ip_address`: Text
    - `user_agent`: Text
    - `created_at`: Timestamp (auto-generated)
- Implement a standard JavaScript library (such as UAParser.js) to collect browser, device, OS, and user agent data.
- IP address can be captured using a third-party API (e.g., ipify.org) or server-side logic if needed.

4. Data Storage and Traceability

- Store all answers in local state (e.g., session/local storage or React state).
- Use local state to calculate the summary tables and for displaying results.
- **Traceability:** Implement UUID per session or per practice name (if unique) to track user interactions.

- Maintain historical records in Supabase so that repeated visits or adjustments by a user can be linked to the same practice (via UUID or session ID).
- Use timestamps to record updates to entries for audit trail.

5. UX/UI Requirements

- **Mobile-first layout**
- **One question per screen** to reduce cognitive load
- **Progress indicators** to show users where they are in the flow
- **Next**, **Back**, and **Restart Flow** buttons
- **Summary page** with readable tables and the Sequel brand styling
- **No completion screen**

Non-Functional Requirements

- **Performance:** Fast load times, minimal network delays.
- **Resilience:** Flow must proceed even if Supabase fails.
- **Accessibility:** Follow WCAG 2.1 guidelines.
- **Branding:** Adhere to Sequel Brand Guidelines (fonts, colors, tone of voice).
- **Data Integrity:** Ensure proper management and traceability of data in Supabase.

Deliverables

- A working **mobile-first web app** deployed to a testing environment.
- **Source code** ready for handoff.
- **Documentation** on setting up the Supabase endpoint and capturing device/browser data.

Notes

- **Brand Guidelines:** Sequel Brand Guidelines to be shared with Replit.
- **No additional database needed beyond Supabase**.
- **No authentication required**.

Next Steps

- Review PRD with stakeholders.
- Finalize the Supabase endpoint.
- Implement the browser/device/user agent capture logic.
- Provide Sequel Brand Guidelines to Replit for styling references.
- Develop the app using React or similar technology stack.

Let me know if you'd like me to adjust or expand on any section!