

COURSE PROJECT: BUILDING AI-POWERED APPLICATIONS

Medsplain

AI-Powered Medication Explainer

Understand your medications with trustworthy, on-demand answers.

DEVELOPED BY

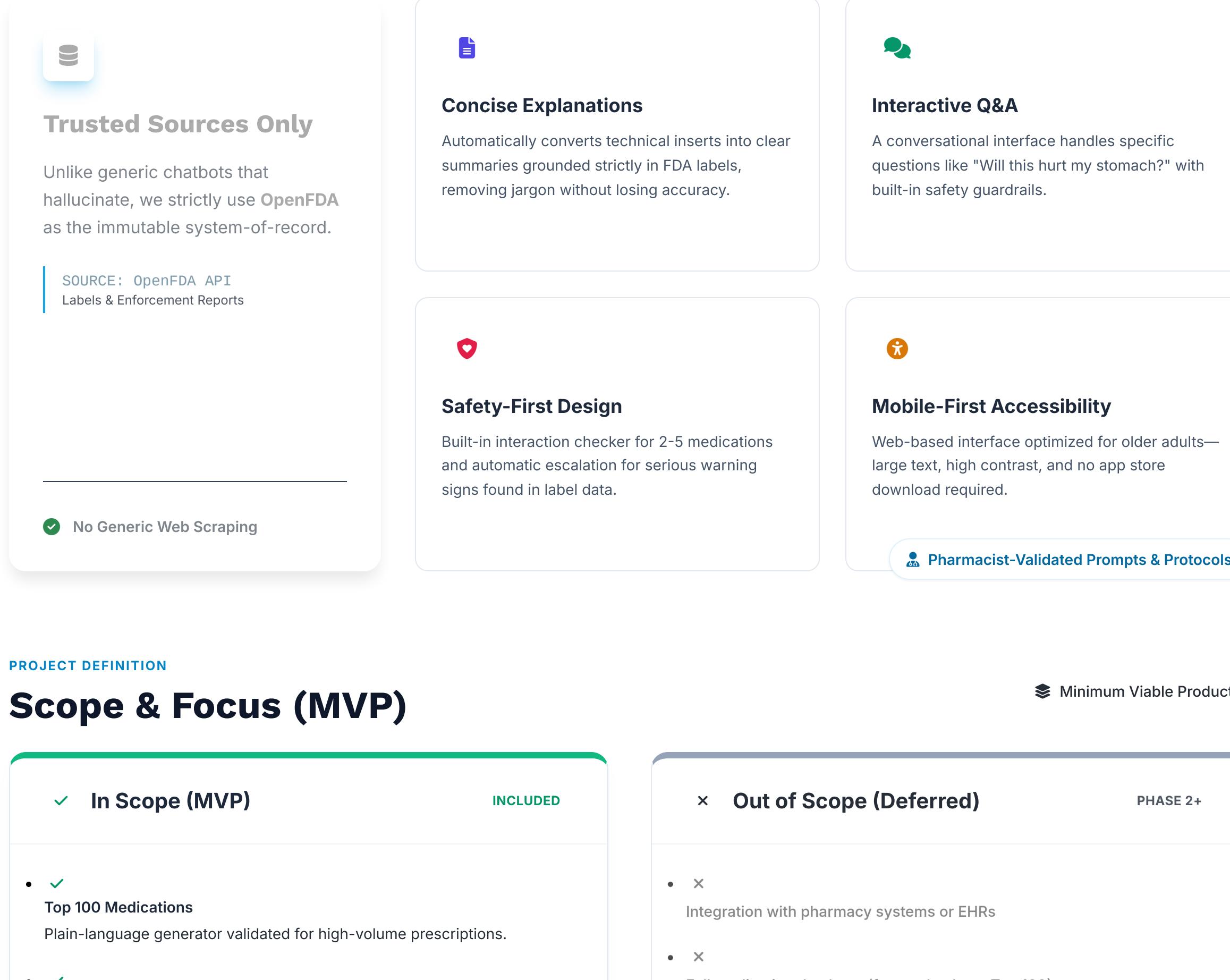
Team Fusion Core

Disclaimer: Educational information only. Not medical advice.

OFFICIAL DATA SOURCE
Sourced from OpenFDA

PROBLEM STATEMENT

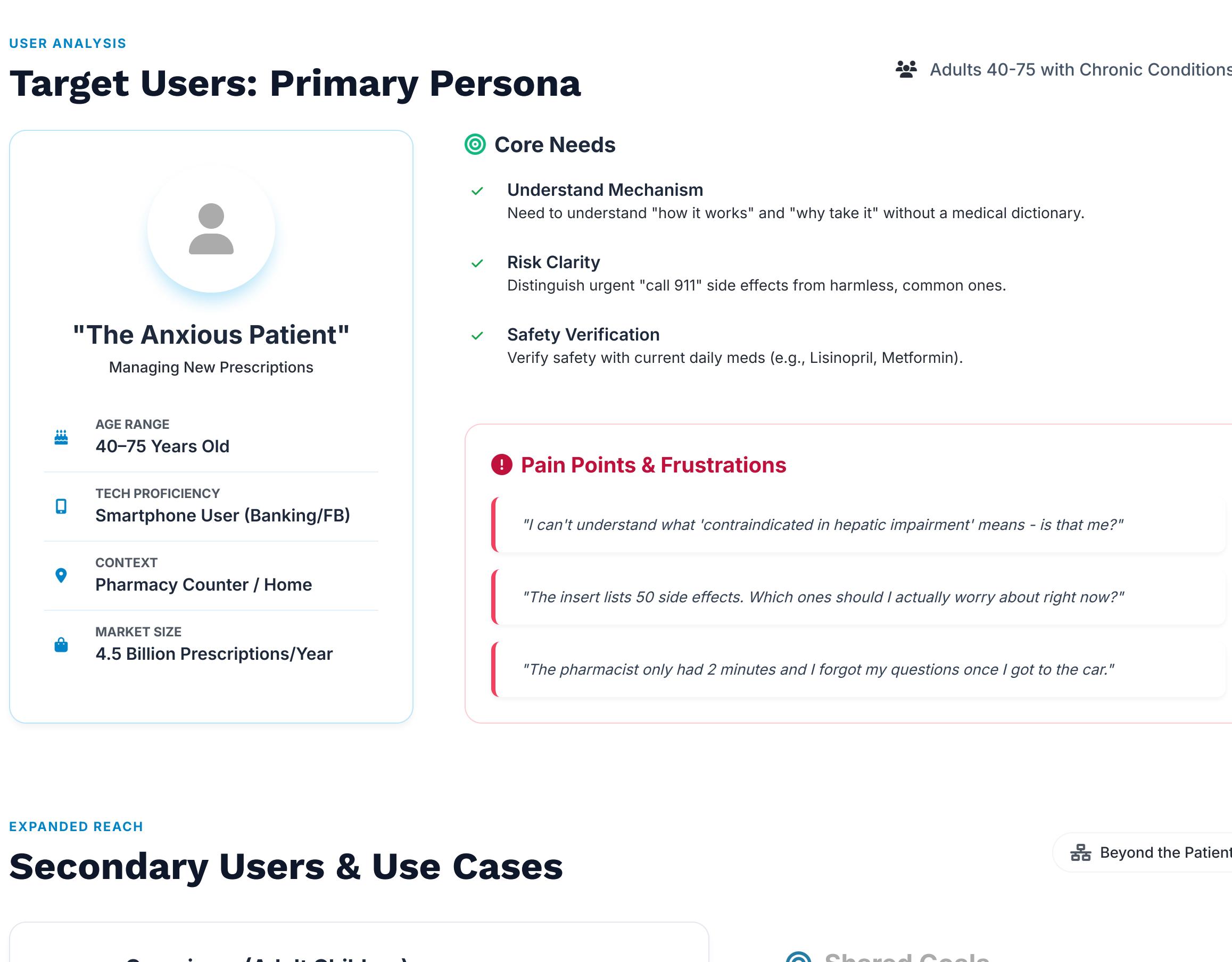
Medication Info Isn't Working



OUR SOLUTION

Medsplain Overview

Translating complex FDA data into personalized, trustworthy explanations with interactive support.



PROJECT DEFINITION

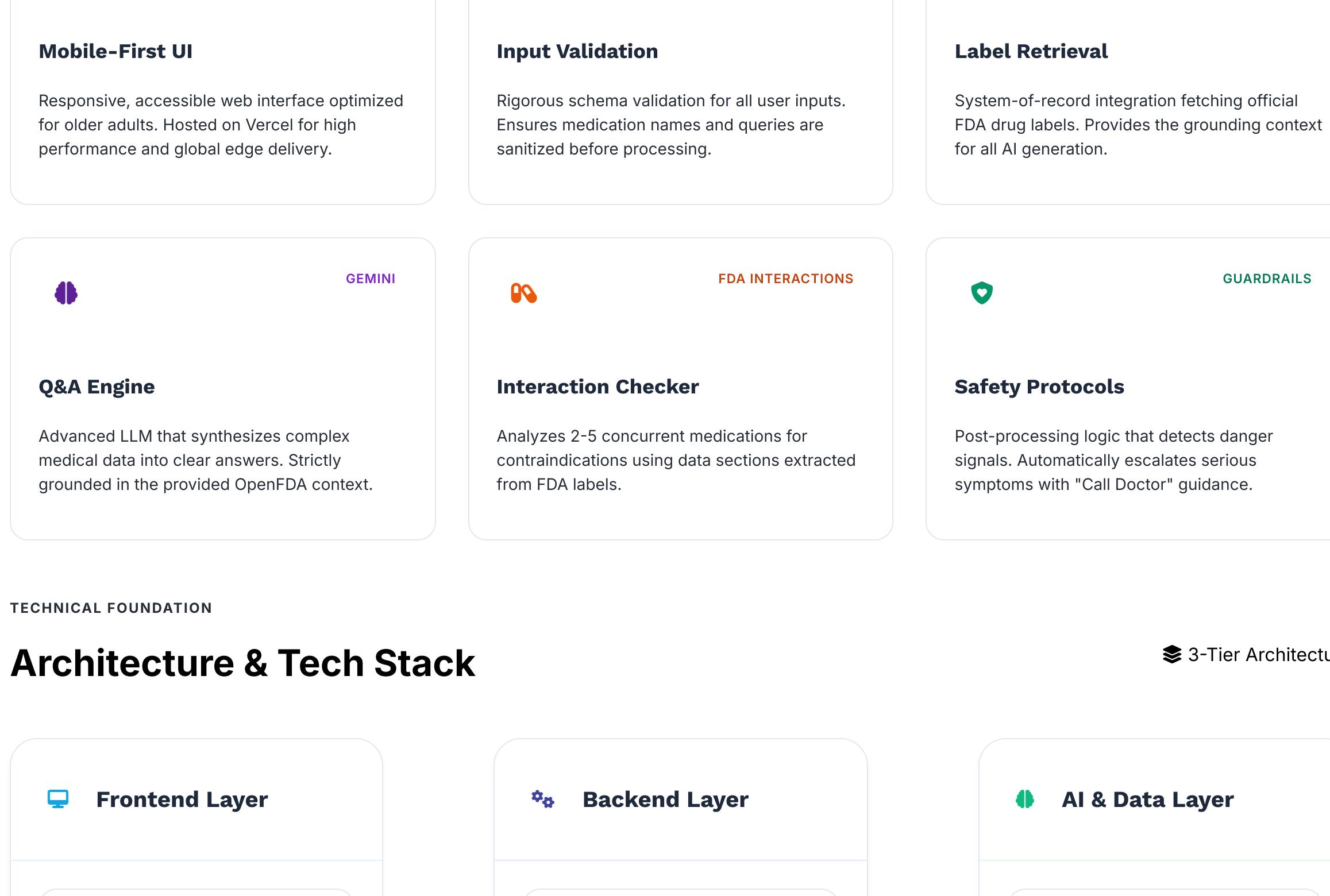
Scope & Focus (MVP)

In Scope (MVP)	INCLUDED	Out of Scope (Deferred)	PHASE 2+
<ul style="list-style-type: none"> Top 100 Medications Plain-language generator validated for high-volume prescriptions. Multi-Medication Checker Analyzes interactions for 2-5 concurrent drugs using OpenFDA data. Safety Guardrails Automatic escalation for "Call Doctor" scenarios and serious warnings. Web-Based Interface Optimized for mobile browsers & accessible to older adults (no app download). 		<ul style="list-style-type: none"> Integration with pharmacy systems or EHRs Full medication database (focused only on Top 100) Symptom checking or diagnosis capabilities Medication shopping or insurance pricing features Native mobile apps (iOS/Android) or multi-language 	Minimum Viable Product

USER ANALYSIS

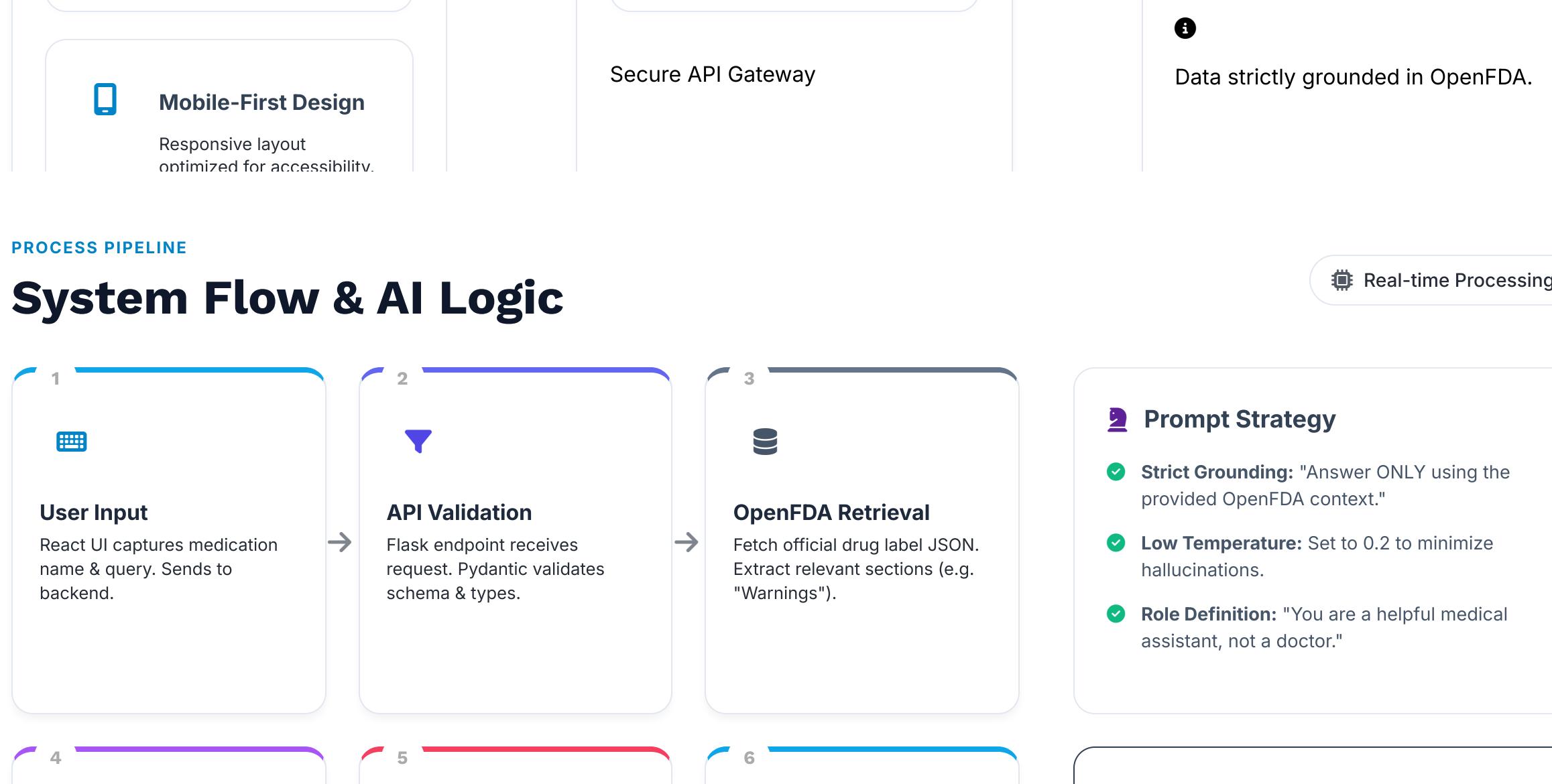
Target Users: Primary Persona

Adults 40-75 with Chronic Conditions



EXPANDED REACH

Secondary Users & Use Cases



SUCCESS FRAMEWORK

Metrics: KPIs & Success Criteria

Product Success Metrics (User Impact)		GOAL	
70%+		3.8/5	

Operational & Technical Targets		GOAL	
3-7 min SESSION EFFICIENCY	Time from input to avail.	20+ users PILOT ADOPTION	Initial users with 30+ sessions

Technical Health		GOAL	
Factual Accuracy	90%+		

TECHNICAL COMPONENTS

System Overview



TECHNICAL FOUNDATION

Architecture & Tech Stack



PROCESS PIPELINE

System Flow & AI Logic

Real-time Processing

User Input		API Validation		OpenFDA Retrieval		Prompt Strategy	
React UI captures medication name & query. Sends to backend.		Flask endpoint receives request. Pydantic validates schema & types.		Fetch official drug label JSON. Extract relevant sections (e.g. "Warnings").		Strict Grounding: Answer ONLY using the provided OpenFDA context.	Low Temperature: Set to 0.2 to minimize hallucinations.

Gemini Generation		Safety Guardrails		UI Delivery		Latency Targets	
Prompt constructed with FDA context. Gemini generates plain answer.		Post-processing checks for danger signals. Appends escalation if needed.		Response streamed via SSE to React frontend on Vercel.		Time to First Byte < 1.5s	Full Response < 3.5s

Optimized via aggressive caching (Redis) and token streaming.

MVP BENCHMARKS

Technical Performance Targets

VERCEL + FLASK		INFRASTRUCTURE		OPENFDA DATA	
< 3s		≥ 90%		90%+	

PYDANTIC		GEMINI API		ZERO TOLERANCE	
< 8%		< \$0.25		ZERO	

Error Rate

Cost Efficiency

Safety Incidents

Contradictions of doctor's orders.

PROJECT RESILIENCE

Risks, Mitigations & Contingency

Deployment Standards

Proactive Management

Gemini Hallucinations

User Over-Reliance

API Rate Limits

Input Malformation

Scope Creep

Gemini Outage

Cost Spike

Low Tester Count

Gemini Halucinations

User Over-Reliance

API Rate Limits

Input Malformation

Scope Creep

Gemini Outage

Cost Spike

Low Tester Count