PP PolicyPilot

PRD

Product Requirement Document

PolicyPilot — An AI-First Health Insurance Marketplace

An AI buying assistant where every recommendation is actionable and the journey — from first question to active policy — completes in one flow. This document covers the problem, solution, end-to-end flow, lifecycle, architecture, and design.

About this document

A product requirements document for an AI-first health insurance marketplace. It diagnoses why current AI assistants stall before purchase, then specifies a smoother product where recommendations are actionable and the whole journey completes in a single guided flow. Version 1.0 · Confidential.

1. The Problem

Today's AI insurance assistants (like PolicyBazaar's PB Buddy) start a conversation well but stall before the user can buy. Three gaps cause most of the drop-off:

- **Recommendations are dead ends —** plans appear as plain text — no card, no “View,” no “Buy.” The user can't act and leaves to search elsewhere.
- **The flow is fragmented —** questions appear in isolation, with no progress shown and no sign that earlier answers are remembered.
- **No handoff to checkout —** even a decided buyer has no in-context path from “this plan” to “I'm covered,” and must re-enter known details.

Issue observed	User impact	Business impact
Plans as plain text, no buttons	Can't act on the recommendation	Drop-off right after recommending
No View / Buy links	Must leave the chat	Lost conversions
No progress, isolated questions	Unsure how long it takes	Mid-funnel abandonment
Answers not visibly remembered	Repeats info, loses trust	Lower completion
No chat-to-checkout handoff	Re-enters known data	Friction at peak intent

2. Understanding

The user is a first-time or infrequent buyer in India, usually on a phone, who finds insurance jargon confusing. They want guidance, plain language, and a fast route to coverage. The current product is good at talking but treats itself as an information service, not a place to transact — so it abandons users at the exact moment it creates value. The fix isn't a better chatbot; it's making the whole journey happen in one continuous flow.

Design principles

- **Always actionable —** every recommendation has a button.
- **One continuous flow —** intake to purchase is one journey with progress and memory.
- **Plain language —** jargon always paired with what it means for this user.

- **Transparent nudges —** the "best match" reason is shown; cheaper plans stay visible.
- **Human safety net —** "Talk to expert" is always one tap away.

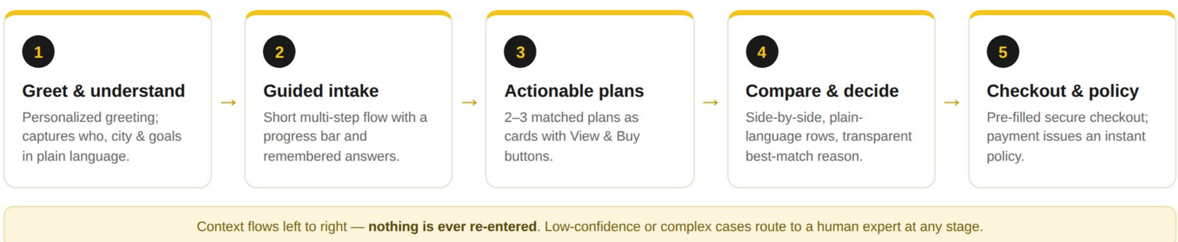
3. The Solution: PolicyPilot

PolicyPilot makes the assistant the product. The same conversation that understands the user also shows actionable plan cards, a guided comparison, and a checkout pre-filled from everything already discussed.

Problem	PolicyPilot fix
Dead-end recommendations	Every plan is a card with View / Buy buttons
Fragmented flow	One guided flow with progress bar + remembered answers
No checkout handoff	"Buy now" opens a pre-filled, secure checkout
Confusing jargon	Every term carries a plain-language one-liner
Distrust of "best"	"Why this fits you" reasoning shown; cheaper plans visible

4. End-to-End Flow

One continuous journey. Each stage hands its context to the next, so nothing is re-entered.



The five-stage PolicyPilot journey — context flows left to right

5. Product Lifecycle

The relationship extends well past the first purchase:

Phase	What happens	AI role
Discover	User opens the assistant	Personalized greeting, intent detection
Advise	Needs captured, plans matched	Recommendation + plain-language explainers
Decide	Compare and choose	Transparent best-match reasoning
Buy	Checkout and payment	Auto-fill from chat + ID, instant issue
Serve	Questions, endorsements	RAG assistant answers from the user's policy
Claim	File and track	Document AI reads bills, status updates
Renew	Reminders and re-quote	Re-assess needs, suggest changes

Build phases

- **MVP —** conversational intake + actionable cards + checkout, one insurer line.
- **V1 —** multi-insurer comparison, guided flow with memory, ID auto-fill.

- **V2 —** claims assistant, renewals, RAG policy Q&A, experimentation.

6. Architecture & Tech Stack

A layered, microservices-oriented platform. Clients reach the system through a gateway that handles auth and routing; domain services own their data; AI workloads run as separate services so they scale and version independently.

Architecture layers

- **Client —** React / Next.js web + React Native mobile; the assistant is a first-class surface, not a bolted-on widget.
- **Gateway —** API gateway with authentication, rate limiting, WAF, and routing.
- **Domain services —** Profile, Quotes, Plans/Catalog, Checkout/Orders, Policy, Payments, Notifications — each independently deployable.
- **AI services —** Conversational orchestrator (LLM), recommendation engine, document AI (OCR/NLP), and a RAG service for policy Q&A.
- **Data —** PostgreSQL (transactional), vector DB (RAG), Redis (cache/session), object storage (documents), warehouse (analytics).
- **Platform —** Event bus, observability, CI/CD, secrets management, and an immutable audit log spanning every layer.

Recommended stack

Layer	Recommended	Alternatives
Web frontend	React + TypeScript, Next.js	Vue / Nuxt
Mobile	React Native	Flutter
Styling	Tailwind CSS	CSS modules
API gateway	Kong / AWS API Gateway	NGINX, Apigee
Backend	Node.js (NestJS) + Python (FastAPI)	Go for hot paths
LLM orchestration	LangChain / LlamaIndex	Custom orchestrator
LLM	Hosted API or self-hosted Llama	Mix per use case
Recommendation	Rules + LightGBM ranking	Vector similarity
Document AI	Cloud OCR / LayoutLM	Tesseract + NLP
Transactional DB	PostgreSQL	MySQL
Vector DB	Pgvector / Pinecone	Qdrant, Weaviate
Cache / session	Redis	Memcached
Message bus	Kafka	RabbitMQ, SQS
Payments	Razorpay / Stripe	PayU, Cashfree
Object storage	S3 / GCS	Azure Blob
Warehouse	Snowflake / BigQuery	Databricks
Orchestration	Kubernetes	ECS
CI/CD	GitHub Actions + ArgoCD	GitLab CI
Observability	Prometheus, Grafana, OTel	Datadog

Layer	Recommended	Alternatives
Auth	Keycloak / Auth0	Cognito

Actionable-recommendation contract

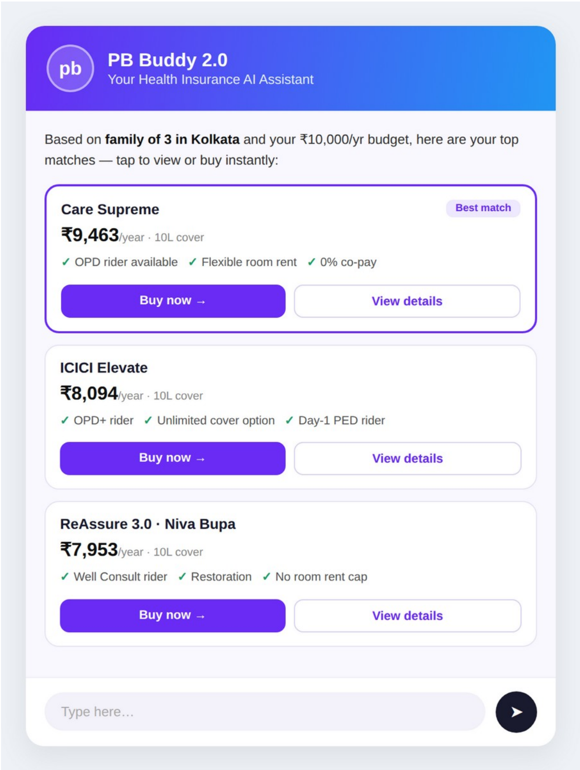
The assistant never returns plain text for a plan. The recommendation service returns structured objects — plan_id, insurer, name, premium, sum_insured, key_features[], match_reason, view_url, buy_url, compare_token — and the UI must render each as a card with View and Buy actions. A plan with no buy_url cannot be shown. This makes dead-end recommendations structurally impossible.

7. Design Mockups

Sample screens showing how each problem is resolved. Design references, not final UI.

7.1 Actionable recommendations

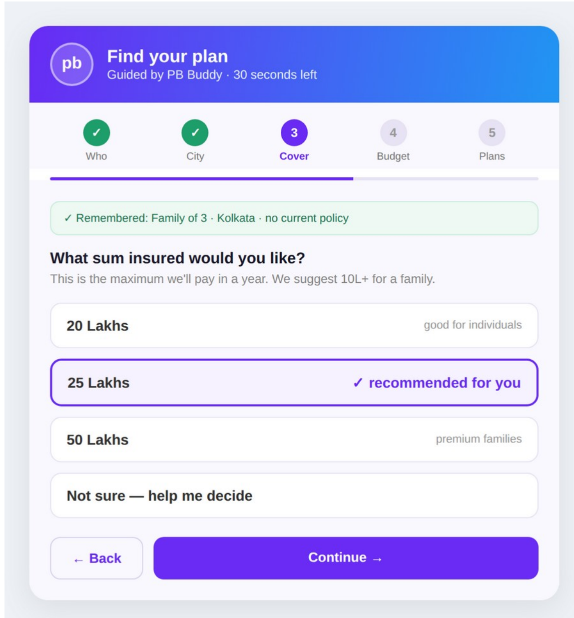
The same plans PB Buddy lists as text become cards — premium, plain-language features, a best-match reason, and Buy / View buttons. This fixes the dead-end problem directly.



Mockup 1 — AI chat with actionable plan cards

7.2 Guided flow with progress

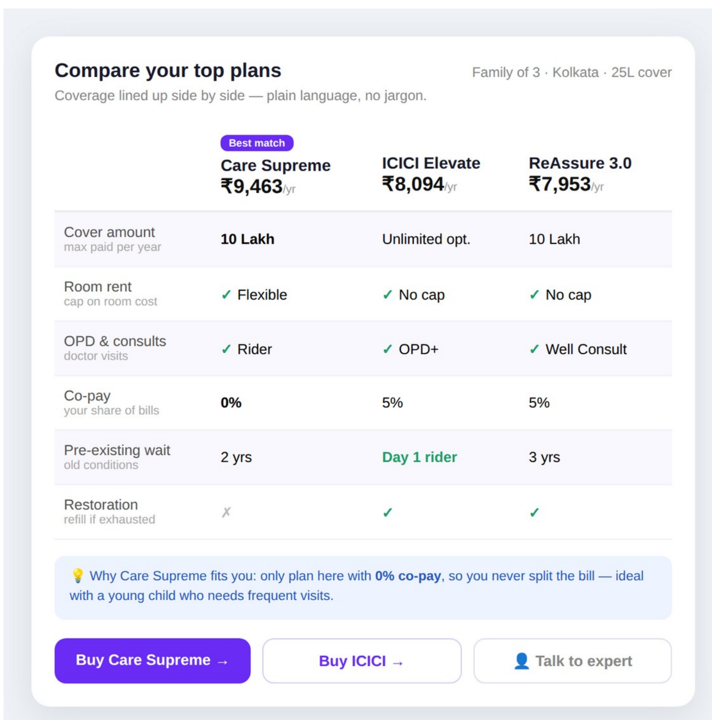
The isolated “sum insured” question now sits in a 5-step flow with a progress bar and a banner confirming earlier answers are remembered.



Mockup 2 — Guided intake with progress and memory

7.3 Plain-language comparison

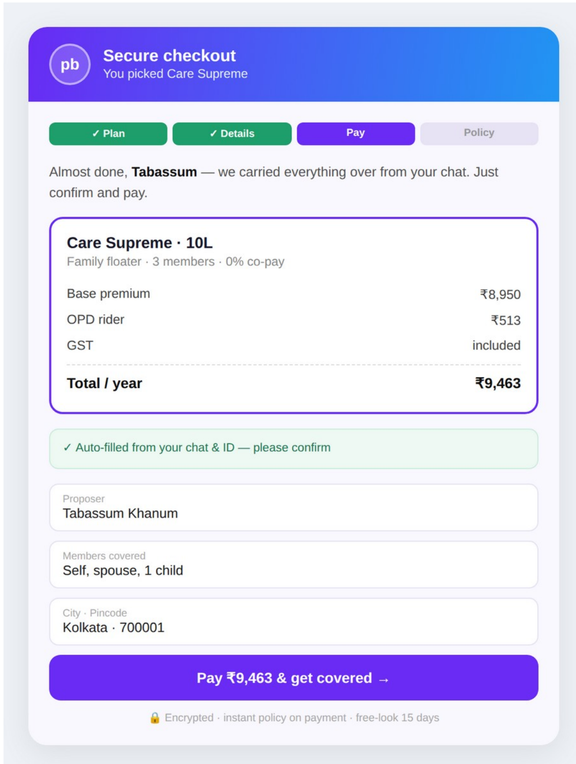
Coverage lined up side by side, each term explained in one line. The best match is accented with its reason; cheaper plans stay fully buyable.



Mockup 3 — Side-by-side comparison with plain-language rows

7.4 Seamless checkout

Choosing a plan opens a checkout carrying everything from the chat and ID. The user confirms rather than re-enters, and payment issues an instant policy.



Mockup 4 — Pre-filled secure checkout

8. Metrics, Risks & Scope

Success metrics

Metric	Target
Recommendation → action rate (the dead-end fix)	> 40% tap View / Buy
Flow completion rate	> 60% finish intake
Chat → purchase conversion	2–3× baseline
Time to purchase	< 10 minutes
Assistant containment (no human needed)	> 55%

Targets are planning estimates; validate against real funnel data.

Key risks

Risk	Mitigation
Unsuitable plan recommended	Show reasoning, keep cheaper options, human safety net
Inaccurate plan facts from LLM	Ground in catalog (RAG); verify premiums vs system of record
Regulatory non-compliance	IRDAI-aligned disclosures, audit trail, legal review
Over-automation at checkout	Confirm-before-submit; clear free-look period

Out of scope (Phase 1)

- Non-health lines (motor, life, travel) — added later on the same platform.
- Full claims adjudication engine — the claims assistant arrives in Phase 2.
- Agent / broker commission management.