Thank you, Maharaj Ji. This PRD is already a strong and comprehensive foundation. Claude has done a solid job, but there are key areas we can **enhance, tighten, or bring more AI-centric depth to** — especially around:

* **Context fusion between screen content + user intent**
* **Model governance, personalization scaffolding, and fine-tuning infrastructure**
* **Error resilience in dynamic content ingestion**
* **Long-term evolution and integrations with other systems (e.g., calendaring, file uploads, referrals)**

Below is the **updated and enhanced version** of your PRD, section-by-section, incorporating best-in-class AI agent design strategies.

🔄 **Enhanced PRD: Cursor AI Agent — LA County Fire Recovery Assistant**

🧠 **Enhancement Themes**

* **Context Fusion**: Merging on-screen content, user location, and session memory to create hyper-relevant answers
* **Data Trust & Quality**: Embed a high-trust document governance model
* **Real-Time Adaptability**: Live sync to source material without agent restarts
* **Scalability**: Robust multi-jurisdictional model with fallback logic
* **Human-AI Teaming**: Escalation-ready for sensitive or ambiguous situations

✅ **Updated Section 1: Core Foundation & Architecture**

🔧 **1.5 Enhancements:**

* **Modular Architecture Blueprint**:
  + Introduce a Knowledge Sync Layer: handles scraping, validation, chunking, and versioning
  + Inference & Routing Layer: routes questions to correct jurisdictional context
  + Conversation Memory Manager: persists session-level user info and intent
* **Model Gateway**:
  + Allow for multi-model fallback (e.g., OpenAI, Claude, Mistral) with runtime switching
  + Use adapters to test responses across providers

🌐 **Updated Section 2: Context Awareness & Location Intelligence**

🚀 **2.5 Enhancements:**

* **DOM Context Injection**: Enable the bot to **ingest the innerText, h1-h6, meta and breadcrumbs** of the current page on load to build a pseudo-contextual "screen awareness"
* **User Journey Tagging**:
  + Classify current user page: e.g., "Debris/Phase II", "Permits/Pasadena", etc.
  + Inject into agent prompt: *“The user is currently browsing the Pasadena debris cleanup guidance page…”*

📚 **Updated Section 3: Multi-Jurisdictional Knowledge Base**

🛠 **3.5 Enhancements:**

* **Multi-Level Context Merging**:
  + Base Layer: LA County fire recovery universal information
  + Mid Layer: City-specific details
  + Top Layer: Page-specific nuance + user input
  + Use a prompt composition strategy with dynamic re-weighting
* **Data Trust Signals**:
  + Add source confidence scores and data freshness flags
  + All responses should embed source URL (invisible or linked text)
* **Jurisdictional Fallback**:
  + If a city doesn’t have custom guidance, fallback to County or FEMA equivalent automatically

🤖 **Updated Section 4: Advanced Query Processing & Intent Recognition**

🧩 **4.5 Enhancements:**

* **Zero-Shot + Prompt-Programmed Classification**:
  + Introduce a multi-step prompt that:
    1. Determines topic category
    2. Infers jurisdiction
    3. Estimates user intent (info-seeking, next-step, emergency, verification)
    4. Asks clarifying questions if ambiguity >30%
* **Conversational Thread Memory**:
  + Store: page context + user responses + last 3 intents
  + Enable: mid-flow course correction (“Actually, I’m asking for Sierra Madre.”)
* **Multi-intent Slot Filling**:
  + Parse multiple parts: “I need to know how to remove debris and if I qualify for insurance help in Malibu”
  + Respond modularly: break into 2-3 digestible replies

🧩 **Updated Section 5: Integration & UX**

🎨 **5.5 Enhancements:**

* **Contextual Appearance**:
  + Chatbot branding subtly adapts to current jurisdiction (logo or color stripe)
* **Contextual Welcome Message**:
  + “Hi! I see you’re viewing [City]’s fire recovery guidance. I can assist with permits, debris removal, and more.”
* **Persistent Sidebar Option**:
  + Let user dock chatbot or expand into a full-screen modal for complex flows
* **Smart UI Actions**:
  + Add suggested action buttons: “Book a Consultation”, “Email the City Planner”, “Download Permit Checklist”

📥 **Updated Section 6: Content Management & Updates**

🔁 **6.5 Enhancements:**

* **Structured Document Format (SDF)**: Internally standardize scraped content with:
  + Jurisdiction
  + Topic category
  + Validity period
  + Version ID
* **Live Diff Engine**:
  + Detect semantic changes (not just text diffs)
  + Trigger partial re-embedding, not full reprocessing
* **Feedback Loop**:
  + All user thumbs-down responses are logged, and trigger a flag for review

📊 **Updated Section 7: Analytics & Monitoring**

📈 **7.5 Enhancements:**

* **Intent-Outcome Mapping**:
  + “User asked about debris removal → downloaded checklist → ended session = ✅ success”
* **Explainability Mode**:
  + Internal toggle (for devs/admins) to see what docs, intents, and context layers contributed to the final answer
* **Bias & Gaps Auditing**:
  + Periodic audit reports to check for jurisdictional imbalance (e.g., Pasadena covered better than Malibu?)

🧠 **NEW SECTION 8: Personalization & User Guidance Memory**

🎯 **Objective:**

Give users a sense of continuity and tailored assistance across sessions

**8.1 Personalized Recovery Assistant**

* Prompt customization based on user progress:
  + “It looks like last time you asked about Phase I debris removal. Want to continue to Phase II?”

**8.2 Guided Recovery Journeys**

* Offer predefined journeys:
  + “Just returned home”
  + “Need to rebuild”
  + “Applying for grants”
  + Each with its own multi-step flow (forms, checklists, etc.)

🧰 **NEW SECTION 9: Developer Tools & Extensibility**

**9.1 Embed SDKs**

* Easy copy-paste script
* Theme builder
* Inline tooltips & API hooks

**9.2 Third-Party Integrations**

* Calendly for appointment scheduling
* Google Drive/Doc sync for new city guidance
* SMS/email follow-up for checklist delivery

📌 **Updates to KPIs (Expanded)**

|  |  |  |
| --- | --- | --- |
| **KPI** | **Goal** | **Owner/Team** |
| Query Resolution Rate | >95% | NLP |
| Content Freshness (auto) | 90% within 6 hrs | Infra |
| Escalation to Human Rate | <3% of sessions | Support |
| Recovery Task Completion | 70% of users complete a next step | UX |
| Content Coverage Gaps | <2 per jurisdiction/month | QA |

📅 **Agile Timeline Enhancements**

Break large “phases” into **Sprint Epics** with review demos:

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Focus Area** | **Key Deliverables** |
| 1–2 | Core chat framework | Widget, vector DB, GPT access |
| 3–4 | Page + location awareness | DOM context injection |
| 5–6 | Multi-jurisdiction data ingestion | All links mapped + tagged |
| 7–8 | Personalization memory | Recovery journey flow |
| 9–10 | Advanced query handling | Intent classifier, threading |
| 11–12 | UI polish + analytics | Embedded dashboards |
| 13–14 | QA, launch prep | Stress tests, source validation |

✅ **Final Enhancements Summary**

* **Better fusion of screen content, user intent, and document knowledge**
* **New personalization and journey-based guidance**
* **Tools for long-term extensibility (SDK, third-party integrations)**
* **Smoother dev/test experience through observability and explainability**
* **Higher QA control with document versioning and audit logs**

Let me know if you'd like me to turn this into a design canvas or break it into task tickets for engineering.