**Product Requirements Document (PRD)**

**Cursor AI Agent - LA County Fire Recovery Assistant**

**Version**: 2.0

**Owner**: Maharaj Ji

**Objective**: Build a context-aware AI chatbot (Cursor AI Agent) that integrates into your site and dynamically answers user queries based on the current page context and a continuously updated set of trusted documents and links.

**Executive Summary**

The Cursor AI Agent will serve as a contextually-aware chatbot integrated into fire recovery assistance websites, providing real-time guidance to residents affected by LA County fires. The agent will dynamically understand user location context, scrape current page content in real-time, and provide relevant information with proper citations from trusted source documents.

**Phase 1: Initialization & Frontend Widget**

**Objective**: Create embedded chatbot widget with real-time page context awareness

**1.1 Frontend Widget Integration**

|  |  |  |
| --- | --- | --- |
| **Task** | **Description** | **Checkpoint** |
| Widget UI | Embedded chat interface with branding and user-friendly layout | ✅ UI renders and opens/closes properly |
| Trigger Conditions | Automatically appear on relevant pages (e.g. after 10s or on scroll) | ✅ Behavior verified via user test |
| Context Tagging | Script scrapes <title>, meta tags, and key headings from current page | ✅ Output shows correct tags in dev console |

**1.2 Page Context Extraction System**

* **Real-time Scraping**: JavaScript scraper extracts semantic info from active webpage
* **Context Elements**:
* Page title and meta descriptions
* H1-H3 headings for topic identification
* URL patterns for location detection
* Key content blocks and form elements
* **Location Detection**: Automatic city/jurisdiction identification from URL and content
* **Topic Classification**: Categorize page content (debris removal, permits, insurance, etc.)

**1.3 Basic Chat Interface**

* **UI Components**:
* Minimizable chat bubble
* Message history with timestamps
* Typing indicators
* Quick action buttons
* Citation display for source references
* **Mobile Optimization**: Responsive design for all screen sizes
* **Accessibility**: WCAG 2.1 AA compliance from start

**1.4 Success Criteria for Phase 1**

* Widget embeds successfully on all target sites
* Real-time context extraction works on 5+ test pages
* UI opens/closes and handles basic interactions
* Context data appears correctly in dev console
* Mobile responsiveness verified across devices

**Deliverable**: Working widget with static responses and real-time context extraction

**Phase 2: Document & Link Integration**

**Objective**: Build semantic knowledge base from provided documents and URLs with continuous updates

**2.1 Backend & Data Layer Architecture**

|  |  |  |
| --- | --- | --- |
| **Module** | **Description** | **Checkpoint** |
| Context Parser | Extracts semantic info from active webpage (location, topic, etc.) | ✅ Parses correctly on 3+ test pages |
| Document Indexer | Caches and semantically indexes contents of provided links/docs | ✅ All URLs/docs indexed without error |
| Retriever Engine | Matches user queries to the most relevant content chunks | ✅ Query returns top 3 semantically relevant results |
| Knowledge Updater | Monitors doc/URL changes & re-ingests periodically or via webhook | ✅ Triggered update adds new info to index |

**2.2 Document Processing Pipeline**

* **Input Sources**:
* HTML pages from government websites
* Google Docs (public links)
* PDF documents
* Dynamic content from recovery sites
* **Processing Steps**:
* Content extraction and cleaning
* Text chunking with overlap
* Semantic embedding generation
* Metadata tagging (source, date, topic, jurisdiction)
* **Storage**: Vector database with PostgreSQL for metadata

**2.3 Semantic Indexing System**

* **Embedding Model**: OpenAI text-embedding-ada-002 or equivalent
* **Chunk Strategy**:
* 500-word chunks with 50-word overlap
* Preserve section headers and context
* Maintain source attribution
* **Metadata Schema**:  
    
  json  
  {  
    "source\_url": "string",  
    "jurisdiction": "string",  
    "topic\_category": "string",  
    "last\_updated": "timestamp",  
    "content\_type": "string",  
    "reliability\_score": "number"  
  }

**2.4 Real-time Content Monitoring**

* **Change Detection**: Monitor source URLs for content updates
* **Update Triggers**:
* Scheduled crawling (daily for critical pages)
* Webhook notifications when available
* Manual refresh via admin panel
* **Version Control**: Track content changes and maintain history

**2.5 Success Criteria for Phase 2**

* All provided URLs successfully indexed
* Semantic search returns relevant results for test queries
* Update system detects and ingests content changes
* Retrieval accuracy >85% for domain-specific queries
* Processing time <5 minutes for new document ingestion

**Deliverable**: Bot can respond using reference material from real documents/URLs with proper citations

**Phase 3: AI Logic & Query Understanding**

**Objective**: Implement sophisticated NLP for intent recognition and response generation

**3.1 AI Logic Layer**

|  |  |  |
| --- | --- | --- |
| **Module** | **Description** | **Checkpoint** |
| NLP Engine | Classifies intent, detects location/context, formats response | ✅ Queries correctly classified in test set |
| Response Synthesizer | Combines extracted data into clear human-like answers | ✅ 80%+ user satisfaction in A/B test |
| Fallback Router | Detects irrelevant queries and triggers fallback (e.g., "We're not sure...") | ✅ 100% fallback on gibberish/edge cases |

**3.2 Intent Classification System**

* **Primary Intent Categories**:
* **Information Seeking**: "How do I apply for debris removal?"
* **Status Checking**: "When can I return to my property?"
* **Process Guidance**: "What's the next step in rebuilding?"
* **Emergency/Urgent**: "I need immediate help"
* **Comparative**: "What's the difference between Phase 1 and Phase 2?"
* **Location-Specific**: "Who do I contact in Pasadena?"

**3.3 Content Domain Taxonomy**

**3.3.1 Topics**

* **Debris Removal** (Phase I & II)
* **Insurance Guidance**
* **Rebuilding & Permits**
* **Property Tax Relief**
* **FEMA/SBA Grants**
* **Tree Inspection & Safety**
* **Contractor Hiring Guidance**
* **Pro Bono Design Help**

**3.3.2 Locations**

* **LA County** (General)
* **City of Pasadena**
* **Sierra Madre**
* **City of Los Angeles**
* **City of Malibu**

**3.4 Query Processing Pipeline**

1. **Context Integration**: Combine user query + page context + conversation history
2. **Intent Classification**: Determine primary and secondary intents
3. **Entity Extraction**: Extract location, dates, specific programs mentioned
4. **Semantic Retrieval**: Find relevant content chunks from knowledge base
5. **Response Synthesis**: Generate coherent answer with proper citations
6. **Fallback Detection**: Identify when confidence is too low for reliable response

**3.5 Response Generation Framework**

* **Structured Responses**: Step-by-step procedures when appropriate
* **Citation Format**: "According to [Source Name], [information]..."
* **Actionable Guidance**: Include specific contact information and next steps
* **Confidence Scoring**: Internal scoring to trigger fallback when needed

**3.6 Fail-Safe Fallback System**

* **Graceful Degradation**: "I don't have specific information about that, but here's who can help..."
* **Contact Routing**: Provide relevant contact information based on detected location/topic
* **Escalation Triggers**: Flag complex queries for human review
* **Learning Integration**: Log fallback cases for knowledge base improvement

**3.7 Success Criteria for Phase 3**

* Intent classification accuracy >95% on test dataset
* Response relevance rating >90% in user testing
* Fallback system catches 100% of off-topic queries
* Citation accuracy rate >98%
* Average response generation time <3 seconds

**Deliverable**: Bot returns accurate, citation-based answers based on current user context and intent

**Phase 4: Testing & Refinement**

**Objective**: Validate system performance through comprehensive testing and user feedback

**4.1 Validation Criteria & Testing Framework**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Goal** | **Testing Method** |
| Response Accuracy | ≥ 90% match to correct doc source | Manual verification against source docs |
| Response Speed | < 2s for average query | Automated performance testing |
| Fallback Accuracy | ≥ 95% detection rate for off-topic queries | Edge case testing with nonsensical queries |
| User Feedback Positivity | ≥ 80% thumbs-up rating | In-app feedback collection |
| Context Awareness | ≥ 90% correct page context detection | Multi-page testing scenarios |

**4.2 Simulated User Journey Testing**

* **Scenario Testing**:
* Pasadena homeowner asking about debris removal
* Malibu resident inquiring about insurance claims
* LA County resident seeking contractor guidance
* Sierra Madre resident asking about tree inspection
* **Multi-turn Conversations**: Test context retention across conversation turns
* **Cross-jurisdictional Queries**: Handle questions spanning multiple cities

**4.3 A/B Testing Framework**

* **Response Formats**: Test different response structures for clarity
* **Citation Styles**: Optimize citation presentation for user trust
* **Fallback Messages**: Test different fallback approaches for effectiveness
* **UI/UX Elements**: Test chat bubble placement and interaction patterns

**4.4 Error Analysis & Improvement**

* **Misclassification Tracking**: Log and analyze incorrect intent classifications
* **Response Quality Review**: Human evaluation of generated responses
* **Content Gap Analysis**: Identify frequently asked questions not covered in knowledge base
* **Performance Optimization**: Identify and resolve bottlenecks

**4.5 Success Criteria for Phase 4**

* 90%+ accuracy in context-appropriate response delivery
* All validation criteria met consistently
* User satisfaction feedback >80% positive
* System handles 100+ concurrent users without degradation
* Comprehensive test coverage across all content domains

**Deliverable**: 90%+ accuracy in context-appropriate response delivery with comprehensive testing validation

**Phase 5: Administration & Content Management**

**Objective**: Comprehensive admin system for content management and system monitoring

**5.1 Administration Tools**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Purpose** | **Features** |
| Admin Panel | Upload/update links, trigger re-index, monitor questions | Link management, manual content updates, system status |
| Training Set Builder | Export common questions/answers for fine-tuning | Query analysis, response optimization, pattern identification |
| Logging Dashboard | Track queries, fallback use, and satisfaction feedback | Real-time analytics, performance metrics, user insights |

**5.2 Content Management System**

* **Link Management Interface**:
* Add/remove source URLs
* Set crawling frequency per source
* Configure content priority levels
* Manual content override capabilities
* **Knowledge Base Editor**:
* Direct content editing for critical information
* Custom response templates
* FAQ management
* Emergency message broadcasting

**5.3 Automated Content Updates**

* **Monitoring System**:
* Track changes on source websites
* Content freshness alerts
* Broken link detection
* Source reliability scoring
* **Update Workflows**:
* Automated re-ingestion triggers
* Content approval workflows for sensitive updates
* Version control and rollback capabilities
* Change notification system

**5.4 Quality Assurance Tools**

* **Response Quality Monitoring**:
* Accuracy scoring system
* Citation verification
* User feedback integration
* Performance trend analysis
* **Content Validation**:
* Fact-checking workflows
* Source authenticity verification
* Legal compliance review
* Multi-language content management (future phase)

**5.5 Success Criteria for Phase 5**

* Admin panel allows complete system management
* Automated updates work reliably within 6 hours
* Content quality metrics tracked and reported
* User feedback integration functional
* System scales to handle content from 50+ sources

**Deliverable**: Fully functional admin system with automated content management and quality assurance

**Phase 6: Analytics & Performance Monitoring**

**Objective**: Comprehensive analytics system for optimization and insights

**6.1 Performance Analytics Dashboard**

* **Real-time Metrics**:
* Active users and sessions
* Query volume and response times
* Error rates and system health
* Content freshness status
* **Historical Analysis**:
* Usage patterns and trends
* Popular queries and topics
* User journey analysis
* Seasonal demand patterns

**6.2 User Behavior Analytics**

* **Engagement Metrics**:
* Session duration and depth
* Query resolution rates
* User satisfaction scores
* Return user patterns
* **Content Performance**:
* Most referenced sources
* Content gap identification
* Citation click-through rates
* Fallback trigger analysis

**6.3 System Health Monitoring**

* **Technical Metrics**:
* API response times
* Database query performance
* Embedding generation speed
* Error logging and alerting
* **Content Quality Metrics**:
* Source reliability scores
* Response accuracy rates
* Citation validity checks
* User correction feedback

**6.4 Reporting & Insights**

* **Automated Reports**:
* Daily system health summaries
* Weekly usage analytics
* Monthly performance reviews
* Quarterly trend analysis
* **Actionable Insights**:
* Content optimization recommendations
* Performance improvement suggestions
* User experience enhancement opportunities
* Scaling and capacity planning

**6.5 Success Criteria for Phase 6**

* Real-time dashboard operational with all key metrics
* Automated alerting for critical issues
* Comprehensive reporting system functional
* Analytics drive continuous improvement recommendations
* Performance optimization based on data insights

**Deliverable**: Complete analytics and monitoring system with actionable insights for continuous improvement

**System Architecture Overview**

**Frontend Integration**

* **Widget Technology**: Vanilla JavaScript widget (framework-agnostic)
* **Embedding**: Single script tag integration
* **Styling**: CSS customization to match site branding
* **Responsive Design**: Mobile-first approach with breakpoint optimization

**Backend Infrastructure**

* **API Gateway**: RESTful API with GraphQL for complex queries
* **Vector Database**: Pinecone or Chroma for semantic search
* **Primary Database**: PostgreSQL for metadata and user sessions
* **Caching Layer**: Redis for performance optimization
* **Message Queue**: For asynchronous content processing

**AI/ML Stack**

* **Language Model**: OpenAI GPT-4 or Anthropic Claude
* **Embeddings**: OpenAI text-embedding-ada-002
* **Intent Classification**: Fine-tuned model on domain-specific data
* **Semantic Search**: Hybrid approach combining vector similarity and keyword matching

**Next Steps for Implementation**

**Immediate Actions Required**

1. **Confirm PRD Scope & Priorities**

* Multi-lingual support requirements?
* SMS/phone integration needed?
* Mobile app vs web-only deployment?
* Integration with existing CRM/ticketing systems?

1. **Technical Environment Setup**

* API access credentials for AI services
* Database hosting preferences (cloud vs on-premise)
* Domain/subdomain for API endpoints
* SSL certificates and security requirements

1. **Content Access Verification**

* Confirm all provided URLs are publicly accessible
* Identify any authentication requirements for Google Docs
* Determine update frequency needs per source
* Privacy/legal considerations for content scraping

**Development Kickoff**

**Phase 1 Start Requirements**:

* Target website URLs for widget integration
* Brand guidelines and UI/UX preferences
* Initial test dataset of questions/answers
* Stakeholder access for testing and feedback

**Questions for Scope Refinement**

1. **Integration Complexity**: Do you need the widget to integrate with existing forms or user authentication systems?
2. **Data Privacy**: Any specific compliance requirements (CCPA, GDPR, HIPAA)?
3. **Scalability**: Expected concurrent user load and geographic distribution?
4. **Customization**: How much visual/functional customization needed per deployment site?
5. **Maintenance**: Preferred approach for ongoing content updates - fully automated vs human oversight?

**Ready to proceed with Phase 1 development once scope is confirmed and technical requirements are clarified.**

**Development Timeline & Milestones**

**Phase 1: Initialization & Frontend Widget (Weeks 1-2)**

* Embedded chat widget with real-time context scraping
* Basic UI/UX with mobile responsiveness
* Page context extraction and display
* **Deliverable**: Working widget with static responses and context extraction

**Phase 2: Document & Link Integration (Weeks 3-5)**

* Backend semantic indexing system
* Content processing pipeline for all provided URLs
* Real-time content monitoring and updates
* **Deliverable**: Bot responds using real document content with citations

**Phase 3: AI Logic & Query Understanding (Weeks 6-8)**

* Intent classification and entity extraction
* Response synthesis with proper citations
* Fallback system for edge cases
* **Deliverable**: Accurate, context-aware responses with high reliability

**Phase 4: Testing & Refinement (Weeks 9-10)**

* Comprehensive user journey testing
* A/B testing of response formats
* Performance optimization and bug fixes
* **Deliverable**: Production-ready system with 90%+ accuracy

**Phase 5: Admin & Content Management (Weeks 11-12)**

* Full admin dashboard with content management
* Automated monitoring and update systems
* Quality assurance tools and workflows
* **Deliverable**: Complete admin system for ongoing management

**Phase 6: Analytics & Launch (Weeks 13-14)**

* Analytics dashboard and reporting
* Performance monitoring and alerting
* Final testing and production deployment
* **Deliverable**: Live system with full monitoring and analytics

**Success Metrics & KPIs**

**Core Performance Indicators**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Target** | **Measurement Method** |
| Response Accuracy | ≥ 90% | Manual verification against source docs |
| Response Speed | < 2s | Automated performance monitoring |
| Fallback Accuracy | ≥ 95% | Edge case testing |
| User Satisfaction | ≥ 80% thumbs up | In-app feedback system |
| System Uptime | > 99.9% | Infrastructure monitoring |
| Context Awareness | ≥ 90% | Multi-page testing scenarios |

**Business Impact Metrics**

* **Support Efficiency**: 40% reduction in manual support requests
* **User Self-Service**: 80%+ query resolution without escalation
* **Information Access**: 50% faster access to relevant resources
* **User Engagement**: Average session >5 minutes with multiple interactions

**Risk Mitigation & Contingency Planning**

**Technical Risks**

* **API Rate Limits**: Implement intelligent caching and request optimization
* **Content Accuracy**: Multi-layer validation with human oversight
* **System Scalability**: Cloud-native architecture with auto-scaling
* **Data Privacy**: GDPR/CCPA compliant design from ground up

**Content Risks**

* **Information Currency**: Real-time monitoring with 6-hour update SLA
* **Source Reliability**: Weighted trust scoring for different sources
* **Legal Compliance**: Regular content review and approval workflows
* **Misinformation Prevention**: Citation requirements and confidence thresholds

**Operational Risks**

* **User Overload**: Graceful degradation and queue management
* **Integration Issues**: Comprehensive testing across deployment environments
* **Maintenance Overhead**: Automated systems with human oversight triggers
* **Cost Management**: Usage monitoring with budget alerts and optimization

This enhanced PRD incorporates the practical, checkpoint-driven approach from ChatGPT while maintaining the comprehensive scope and technical depth needed for a production-ready system. The phased approach ensures each component is thoroughly tested before building the next layer.

* **Privacy Compliance**: CCPA/GDPR adherence
* **Rate Limiting**: API abuse prevention
* **Input Sanitization**: XSS/injection attack prevention
* **Access Control**: Role-based permissions

**Performance Targets**

* **Response Time**: <3 seconds average
* **Uptime**: 99.9% availability
* **Concurrent Users**: Support 1000+ simultaneous users
* **Data Processing**: Handle 100MB+ documents
* **Scalability**: Auto-scaling based on demand

**Development Timeline & Milestones**

**Phase 1: Foundation (Weeks 1-3)**

* Basic chatbot framework
* Document processing pipeline
* Initial URL ingestion
* Basic Q&A functionality

**Phase 2: Context Awareness (Weeks 4-6)**

* Location detection system
* Page content analysis
* Contextual response framework
* Multi-jurisdictional knowledge base

**Phase 3: Advanced Processing (Weeks 7-9)**

* Intent recognition system
* Complex query handling
* Personalization features
* Rich response formatting

**Phase 4: Integration & UX (Weeks 10-12)**

* Website integration components
* Mobile optimization
* Accessibility implementation
* Performance optimization

**Phase 5: Content Management (Weeks 13-15)**

* Automated update system
* Admin dashboard
* Quality assurance tools
* Content workflow management

**Phase 6: Analytics & Launch (Weeks 16-18)**

* Analytics implementation
* Monitoring setup
* Final testing and optimization
* Production deployment

**Success Metrics & KPIs**

**User Engagement**

* **Query Resolution Rate**: >95%
* **User Satisfaction**: >4.5/5 stars
* **Return User Rate**: >60%
* **Average Session Duration**: >5 minutes

**Technical Performance**

* **System Uptime**: >99.9%
* **Average Response Time**: <3 seconds
* **Error Rate**: <0.5%
* **Content Accuracy**: >98%

**Business Impact**

* **Support Ticket Reduction**: 40% decrease
* **User Self-Service Rate**: >80%
* **Information Access Speed**: 50% improvement
* **User Completion Rate**: >70% for multi-step processes

**Risk Mitigation**

**Technical Risks**

* **API Rate Limits**: Implement caching and optimization
* **Data Accuracy**: Multiple validation layers
* **System Scalability**: Cloud-native architecture
* **Security Vulnerabilities**: Regular security audits

**Content Risks**

* **Information Outdated**: Automated update system
* **Misinformation**: Human review process
* **Legal Compliance**: Regular legal review
* **Multi-language Support**: Phase 2 consideration

**User Experience Risks**

* **Complex Queries**: Escalation to human support
* **Accessibility Issues**: WCAG compliance testing
* **Mobile Performance**: Responsive design optimization
* **Integration Problems**: Thorough testing across platforms

This PRD provides a comprehensive roadmap for developing a sophisticated AI chatbot that will serve as an invaluable resource for LA County fire recovery efforts, with built-in flexibility for expansion and continuous improvement.