Insydr AI - Product Requirements Document

Project Overview

Insydr AI is a multi-tenant, embeddable AI chatbot platform that enables businesses to create intelligent conversational agents powered by their own knowledge base. The platform allows users to build, customize, and deploy AI agents that can answer questions, provide support, and engage with website visitors using company-specific information extracted from documents, websites, and manual entries.

The project will be developed as a college MVP with a clear path to commercialization, using a cost-effective tech stack leveraging free LLM APIs (Gemini, open-source models) while maintaining enterprise-grade architecture for future scalability.

Product Vision

To democratize AI-powered customer engagement by providing an accessible, developer-friendly platform where any business can create sophisticated conversational agents without AI expertise. Insydr AI aims to bridge the gap between raw AI capabilities and practical business applications, making intelligent automation available to companies of all sizes.

Long-term Vision: Become the go-to platform for AI-powered business automation, evolving from simple Q&A agents to action-taking operators that can qualify leads, create support tickets, schedule meetings, and integrate with enterprise systems.

Core Value Proposition: Deploy production-ready AI agents in minutes, not months. No AI expertise required, no infrastructure management needed, and pay only for what you use.

Target User

Primary Personas

1. Solo Entrepreneur / Small Business Owner

- Age: 25-45
- Technical skill: Basic to intermediate
- Pain points: Limited resources, needs 24/7 customer support, can't afford dedicated support team
- Goals: Automate customer inquiries, reduce response time, appear more professional
- Budget: $0-$50/month

2. Startup Founder / Product Manager

- Age: 28-40
- Technical skill: Intermediate to advanced

- Pain points: Scaling customer support, repetitive questions drain team time, need data insights
- Goals: Scale support without hiring, improve customer satisfaction, gather user insights
- Budget: $50-$200/month

3. Developer / Technical Lead

- Age: 24-38
- Technical skill: Advanced
- Pain points: Building custom chatbots is time-consuming, maintaining AI infrastructure is complex
- Goals: Quick integration, API access, customization control, reliable performance
- Budget: Values time savings over cost

Secondary Personas (Phase 2)

4. Agency Owner / Consultant

- Manages multiple client websites
- Needs white-label solution
- Budget: $200-$500/month per client

5. Enterprise IT Manager

- Requires compliance, security, SSO
- Budget: Custom enterprise pricing

Core Features

Module 1: User Management & Multi-Tenancy

User Authentication

- Email/password registration with email verification
- Secure password reset flow
- JWT-based session management with refresh token rotation
- Session timeout and security policies

Workspace System

- Each user gets an isolated workspace upon registration
- Workspace represents a complete tenant with separated data, agents, analytics
- Workspace settings: name, logo upload, timezone selection, language preference
- Row-level security ensuring complete data isolation between tenants

API Key Management

- Generate multiple API keys per workspace with descriptive labels
- Domain whitelisting for security (restrict key usage to specific domains)
- Real-time usage tracking per key (requests, messages, bandwidth)
- One-click key rotation and revocation
- Key status indicators (active, inactive, rate-limited)

Module 2: Agent Management

Agent Creation & Configuration

- Create unlimited agents per workspace (plan-based limits)
- Each agent operates as an independent entity with unique configuration
- Agent profiles include name, description, avatar upload, and status toggle

Agent Type Selection

- Predefined templates: Sales Assistant, Customer Support, HR Assistant, Technical Support, General Knowledge, Custom
- Each type comes with optimized default prompts and behavior settings

Behavior & Personality Settings

- Tone adjustment: Friendly, Professional, Formal, Casual, Technical
- Response style: Brief, Detailed, Conversational, Structured
- Personality customization through custom prompt injection
- Temperature control for response creativity vs consistency

Language Configuration

- Auto-detect user language (default)
- Fixed language mode for monolingual applications
- Supported languages: English, Hindi, Spanish, French, German, Portuguese, Japanese, Chinese
- Fallback language when translation unavailable

Response Configuration

- Maximum response length limits (tokens/words)
- Confidence threshold slider (controls when agent says "I don't know")
- Custom fallback messages for low-confidence scenarios
- Source citation toggle (show/hide referenced documents)
- Response format preferences (paragraphs, bullet points, numbered lists)

Conversation Rules & Guardrails

- Allowed topics whitelist
- Blocked words and topics blacklist
- Custom system prompts for fine-tuned behavior
- Configurable greeting messages with dynamic variables
- End-of-conversation messages with call-to-action options
- Maximum conversation length limits

Agent Status & Version Control

- Active/Inactive toggle for quick enable/disable
- Publish/Unpublish workflow for draft agents
- Version management (v1.0, v1.1, v2.0) with rollback capability
- Change history log for audit trail

Module 3: Knowledge Management

Data Ingestion Pipeline

File Upload System

- Supported formats: PDF, DOCX, TXT, CSV, MD
- Drag-and-drop interface with batch upload support

- Maximum file size: 10MB per file (configurable per plan)
- PDF text extraction with structure preservation
- DOCX parsing maintaining formatting context
- CSV processing for structured data and FAQs
- Progress indicators and upload status

Website Crawling

- URL input with automatic sitemap detection
- Configurable crawl depth (1-5 levels)
- Maximum pages per crawl: 50 initially (plan-based scaling)
- Respects robots.txt and meta tags
- Smart content extraction (removes navigation, footer, ads, scripts)
- Duplicate content detection and removal
- Crawl scheduling for periodic updates

Manual Text Entry

- Rich text editor with formatting toolbar
- Markdown support for technical documentation
- FAQ format templates (Question/Answer pairs)
- Bulk import via structured JSON/YAML

Knowledge Processing Engine

Automatic Chunking

- Intelligent text splitting with sentence and paragraph awareness
- Configurable chunk size: 400-600 tokens (default 500)
- Chunk overlap: 50 tokens for context continuity
- Maintains semantic coherence within chunks
- Preserves headings and structure markers

Embedding Generation

- Integration with free embedding models (Gemini Embedding API, sentence-transformers)
- Batch processing for efficiency (process 100+ documents simultaneously)
- Automatic retry on failures with exponential backoff
- Embedding dimension: 768 (balance between quality and storage)
- Store in PostgreSQL with pgvector extension

Metadata Management

- Automatic extraction: document title, upload date, source type, file size
- Language detection using language detection libraries
- Auto-generated tags using keyword extraction
- Manual tag editing and custom categorization
- Document ownership and permissions

Knowledge Versioning & History

- Snapshot-based versioning (v1.0, v1.1, v2.0)
- Version comparison with diff visualization
- Rollback to any previous version instantly
- Archive old versions without deletion

- Version notes and change descriptions

Knowledge Organization

Collections & Categories

- Hierarchical folder structure for organization
- Tag-based filtering and multi-tag support
- Assign specific collections to specific agents
- Collection-level access control
- Smart collections based on automatic rules

Search & Discovery

- Full-text search across knowledge base
- Filter by: type, date range, language, tags, source
- Sort by: relevance, date, size, usage frequency
- Preview document chunks inline
- Highlight search terms in results

Bulk Operations

- Multi-select for batch actions
- Bulk delete with confirmation
- Bulk re-processing of embeddings (when model upgrades)
- Bulk metadata updates
- Export selected documents

Answer Preview & Testing

- Built-in test console to ask questions before publishing
- Shows retrieved source chunks with relevance scores
- Confidence scoring visualization (0-100%)
- Side-by-side comparison of different knowledge base versions
- Suggested improvements based on retrieval quality

Knowledge Gap Detection

- Automatically tracks questions that couldn't be answered
- Identifies patterns in unanswered queries
- Suggests missing content areas
- Flags consistently low-confidence topics
- Generates recommended FAQ questions

Module 4: Widget SDK (Embeddable Chatbot)

Simple Integration

- Single script tag implementation
- Async loading for no performance impact
- Initialization with minimal configuration
- Automatic updates without code changes

Widget Customization

Appearance

- Theme selection: Light, Dark, Auto (follows system preference)
- Primary brand color picker with live preview
- Accent color for highlights and buttons
- Border radius control (0-20px, fully rounded to sharp)
- Shadow style: None, Subtle, Medium, Strong
- Font family selection (system fonts + Google Fonts)
- Font size scaling (80%-120%)

Position & Layout

- Corner positions: Bottom-right (default), Bottom-left, Top-right, Top-left
- Custom offset from edges (X and Y pixels)
- Z-index control for layer management
- Full-screen mode toggle
- Embedded mode (inline on page, not floating)

Avatar & Branding

- Upload custom agent avatar (PNG, JPG, GIF support)
- Agent name display toggle
- Subtitle/tagline below agent name
- "Powered by Insydr" badge (removable on paid plans)
- Custom loader animations

Launcher Button

- Default bubble with icon
- Custom icon upload (SVG, PNG)
- Custom text label option
- Badge notification showing unread message count
- Pulse animation for attention
- Sound notification toggle

Widget Behavior

Greeting & Welcome Experience

- Auto-show on page load with configurable delay (0s, 3s, 5s, 10s)
- Scroll-triggered display (show after X% scroll)
- Time-on-page trigger (show after X seconds)
- Custom greeting message with user name variable
- Suggested questions (3-5 quick reply buttons)
- Context-aware greetings based on page URL

Conversation Flow

- Typing indicators with realistic delays
- Message timestamps (relative and absolute)
- Read receipts
- Persistent message history (session or cross-session)
- Clear chat button
- Download conversation transcript
- Conversation restart option

Multi-Language Support

- Auto-detect browser language from navigator
- Manual language switcher dropdown
- Remember language preference in localStorage
- Real-time translation of UI elements
- Fallback language configuration

Source Citation Display

- Inline source references with footnote numbers
- Click to expand source preview modal
- Full document view in modal
- Multiple sources per answer
- Source credibility indicators

Widget Features

User Interaction

- Text input with Enter-to-send
- Multi-line support with Shift+Enter
- File upload capability (images, PDFs for context)
- Voice input (speech-to-text) using Web Speech API
- Emoji picker
- Markdown rendering in messages
- Code syntax highlighting

Feedback System

- Thumbs up/down on each message
- Optional feedback comment box
- "Report issue" button with categorization
- Star rating for overall conversation
- NPS survey prompt after conversation end

Escalation Flows

- "Talk to human" button with visibility rules
- Automatic escalation trigger on repeated "I don't know"
- Contact form capture (email, phone, name, message)
- Webhook notification to support team
- Integration with live chat tools (Intercom, Zendesk)

Developer Features

Event Hooks

- On widget load, open, close events
- On message sent, received events
- On answer not found event
- On escalation requested event
- On language change event
- On feedback submitted event
- Custom event emission

Programmatic Control

- Open/close widget methods

- Send message programmatically
- Set user context (name, email, custom data)
- Clear conversation history
- Update widget configuration dynamically
- Destroy and reinitialize widget

Custom Styling

- CSS class namespacing for all elements
- Custom stylesheet injection
- CSS variable overrides
- Shadow DOM option for complete isolation

Security & Performance

- Domain whitelisting (restrict widget to approved domains)
- API key validation on every request
- Rate limiting per domain and per visitor
- CORS configuration
- XSS and injection attack prevention
- Content Security Policy support
- Lazy loading for optimal performance
- Message queue for offline support

Module 5: Analytics Dashboard

Overview Dashboard

- Key metrics cards: total conversations, total messages, active agents, knowledge base size
- Date range selector (last 7, 30, 90 days, custom range)
- Trend indicators (up/down percentage vs previous period)
- Average response time tracking
- User satisfaction score (based on feedback)
- Quick action buttons to address issues

Conversation Analytics

Volume Metrics

- Messages per day/week/month with line charts
- Conversation volume trends with forecasting
- Peak usage hours heatmap
- Day-of-week distribution
- Average conversation duration
- Message distribution (user vs agent messages)

Question Analysis

- Top 20 most asked questions with frequency counts
- Trending questions (growing in frequency)
- Question categorization by intent/topic using clustering
- Question complexity analysis (simple vs complex)
- Seasonal patterns in question types

Unanswered Questions Report

- All questions with confidence below threshold
- Questions triggering fallback responses
- Grouped by topic for knowledge gap identification
- Export to CSV for content creation workflow
- Priority ranking based on frequency

User Behavior Insights

- New vs returning visitors ratio
- Conversation abandonment rate and drop-off points
- Average messages per conversation
- Time to first response
- Engagement metrics by page/URL
- Funnel analysis (greeting → question → resolution)

Agent Performance Metrics

- Success rate percentage (answered confidently vs total)
- Average confidence score distribution
- Response accuracy based on user feedback
- Agent comparison when multiple agents exist
- Performance trends over time
- A/B test results when running experiments

Language Analytics

- Language distribution pie chart
- Performance metrics by language (confidence, satisfaction)
- Translation quality indicators
- Untranslated content flagging

Source & Knowledge Analytics

- Most frequently referenced documents
- Least used documents (candidates for archival)
- Source click-through rate
- Knowledge base coverage score
- Document effectiveness ranking
- Content freshness indicators

Feedback Analytics

- Overall thumbs up/down ratio
- Feedback trends over time
- Sentiment analysis of feedback comments
- Issue reports dashboard with categorization
- Feature requests aggregated from feedback
- Correlation between confidence score and feedback

Export & Reporting

- Export any dataset to CSV/JSON
- Scheduled reports via email (daily, weekly, monthly)

- Custom report builder with drag-and-drop
- Automated insights generation
- Shareable dashboard links

Module 6: Admin Dashboard (Frontend)

Dashboard Home

- Quick stats overview with drill-down capability
- Recent activity feed (uploads, agent edits, conversations)
- Quick action shortcuts (Create Agent, Upload Document, View Analytics)
- Notification center for alerts and recommendations
- Workspace health score

Navigation Architecture

- Sidebar navigation with collapsible sections
- Breadcrumb navigation for deep pages
- Global search across all entities (agents, documents, conversations)
- Keyboard shortcuts for power users
- Mobile-responsive hamburger menu

Agents Section

- Grid/list view toggle for agents
- Agent cards with key metrics and status
- Create new agent wizard (step-by-step)
- Edit agent with live preview pane
- Duplicate agent for quick variations
- Archive/delete with confirmation
- Agent-specific analytics dashboard
- Test playground for each agent

Knowledge Section

- File upload zone with drag-and-drop
- URL crawler interface with preview
- Document library with card/list/table views
- Filters and search across documents
- Version timeline visualization
- Document preview modal with full content
- Bulk actions toolbar
- Storage usage indicator with plan limits

Widget Configuration

- Visual customizer with real-time preview
- Split-screen: settings on left, preview on right
- Style presets (Light, Dark, Brand, Minimal)
- Advanced settings in expandable sections
- Integration code snippet with copy button
- Test mode toggle to preview on actual website
- Installation guide with framework-specific examples

Analytics Section

- Interactive charts with zoom and filter

- Date range selectors on all views
- Comparative analysis tools
- Custom dashboard builder
- Saved views and bookmarks
- Data export options

Settings Section

- Workspace settings: name, logo, timezone, language
- API key management with usage graphs
- Team management (invite members, assign roles)
- Billing and subscription management
- Notification preferences
- Integration connections (CRM, webhooks, etc.)
- Security settings (2FA, session timeout)

User Profile

- Personal information and avatar
- Password change
- Email preferences
- Activity log
- Connected accounts

Design Patterns

- Consistent component library (buttons, inputs, cards)
- Loading states and skeletons
- Empty states with helpful CTAs
- Error states with recovery actions
- Success confirmations with undo options
- Tooltips and help text throughout
- Onboarding tour for first-time users

Module 7: API & Developer Tools

REST API

Agent Management

- GET /api/v1/agents (list all agents)
- POST /api/v1/agents (create agent)
- GET /api/v1/agents/:id (get agent details)
- PUT /api/v1/agents/:id (update agent)
- DELETE /api/v1/agents/:id (delete agent)
- POST /api/v1/agents/:id/publish (publish agent)

Knowledge Management

- POST /api/v1/knowledge/upload (upload file)
- POST /api/v1/knowledge/crawl (crawl URL)
- GET /api/v1/knowledge/documents (list documents)
- DELETE /api/v1/knowledge/documents/:id (delete document)
- PUT /api/v1/knowledge/documents/:id (update metadata)
- POST /api/v1/knowledge/documents/:id/reprocess (regenerate embeddings)

Chat & Conversations

- POST /api/v1/chat/message (send message, get response)
- GET /api/v1/conversations (list conversations)
- GET /api/v1/conversations/:id (get conversation history)
- DELETE /api/v1/conversations/:id (delete conversation)
- POST /api/v1/conversations/:id/feedback (submit feedback)

Analytics

- GET /api/v1/analytics/overview (get overview metrics)
- GET /api/v1/analytics/conversations (get conversation stats)
- GET /api/v1/analytics/questions (get top questions)
- GET /api/v1/analytics/export (export data)

Workspace Management

- GET /api/v1/workspace (get workspace info)
- PUT /api/v1/workspace (update workspace)
- GET /api/v1/workspace/usage (get usage statistics)

API Documentation

- Interactive documentation using Swagger UI / ReDoc
- Code examples in multiple languages (cURL, JavaScript, Python, Node.js, PHP, Ruby)
- Authentication guide with examples
- Rate limiting documentation with headers explained
- Error code reference with troubleshooting
- Changelog for API versions
- Sandbox environment for testing

Webhooks

Supported Events

- conversation.started (new conversation initiated)
- conversation.ended (conversation concluded)
- message.received (user message received)
- answer.not_found (agent couldn't answer confidently)
- escalation.requested (user requested human support)
- feedback.submitted (user gave feedback)
- document.processed (document finished processing)
- agent.published (agent went live)

Webhook Configuration

- Add multiple webhook endpoints per event
- Webhook signature verification for security
- Retry mechanism with exponential backoff
- Webhook logs for debugging
- Test webhook button to send sample payload
- Webhook health monitoring

Developer Portal

- API key management with creation/revocation
- API playground for testing endpoints without code
- Request/response inspector
- Real-time API logs with filters
- Usage metrics and quota monitoring
- Sandbox workspace separate from production

- Code generator for quick integration
- SDKs for popular languages

Module 8: Infrastructure & Backend

Authentication & Authorization

- JWT access tokens (short-lived, 15 minutes)
- Refresh token rotation for security
- Role-based access control (Admin, Editor, Viewer)
- API key authentication for widget and API
- Session management with Redis
- Password hashing with bcrypt
- Account lockout after failed attempts
- Email verification required for signup

Multi-Tenancy Architecture

- Row-level security with workspace_id in all tables
- Database connection pooling per workspace
- Resource quotas enforced per workspace (messages, storage, agents)
- Tenant isolation validated on every query
- Cross-tenant access prevention at database level
- Soft delete for data recovery

Vector Search Engine

- PostgreSQL with pgvector extension for similarity search
- Hybrid search combining vector and keyword search
- Relevance ranking with customizable weights
- Semantic caching for frequent queries
- Query optimization with appropriate indexes
- Configurable similarity threshold per agent

LLM Integration

Model Support

- Gemini API (free tier, 15 requests/minute)
- Ollama integration for local models (llama2, mistral, etc.)
- OpenAI API support (for paid plans)
- Model fallback chain (primary → secondary → tertiary)
- Model selection per agent

Prompt Engineering Layer

- Dynamic prompt construction with context injection
- System prompt templates by agent type
- Few-shot examples for better responses
- Conversation history summarization for long chats
- Token counting and truncation
- Temperature and max tokens configuration

Response Handling

- Streaming support for real-time responses
- Response post-processing (formatting, cleanup)

- Safety filtering and content moderation
- Token usage tracking per request
- Cost calculation and quota enforcement
- Response caching for identical questions

File Processing Pipeline

Asynchronous Processing

- Job queue using BullMQ (Node.js) or Celery (Python)
- Worker processes for parallel processing
- Job status tracking (queued, processing, completed, failed)
- Priority queue for urgent processing
- Retry mechanism with exponential backoff
- Dead letter queue for failed jobs

Text Extraction

- PDF parsing using pdf-parse (Node.js) or PyPDF2/pdfplumber (Python)
- DOCX parsing using mammoth.js (Node.js) or python-docx (Python)
- HTML cleaning and text extraction
- Markdown parsing and structure preservation
- CSV parsing with delimiter detection
- Text encoding detection and conversion

Content Processing

- Text normalization (lowercasing, whitespace cleanup)
- Language detection using franc or langdetect
- Tokenization for chunking
- Stop word removal (optional)
- Special character handling
- URL and email pattern extraction

Caching Layer

- Redis for session storage and token blacklisting
- Response caching for frequently asked questions (TTL: 1 hour)
- Embedding cache to avoid recomputation on document re-processing
- Conversation context caching
- API response caching with smart invalidation
- Cache warming for popular queries

Rate Limiting

Multi-Level Rate Limiting

- Per API key: configurable based on plan (100-10,000 req/hour)
- Per workspace: total requests across all keys
- Per IP: widget requests (100 req/hour per visitor)
- Per user: authentication endpoints (10 req/minute)
- Global rate limit for system protection

Implementation

- Redis-based rate limiting with sliding window
- Rate limit headers in responses (X-RateLimit-Limit, X-RateLimit-Remaining)
- 429 status code when exceeded

- Graceful degradation instead of hard blocks

Monitoring & Logging

Application Logging

- Structured logging with JSON format
- Log levels: DEBUG, INFO, WARN, ERROR
- Request/response logging with sanitization
- Performance logging (API latency, database query time)
- User action audit logs

Error Tracking

- Error aggregation and deduplication
- Stack trace capture
- Context information (user, workspace, request details)
- Alert notifications for critical errors
- Error resolution tracking

Performance Monitoring

- API endpoint performance metrics
- Database query performance
- LLM response time tracking
- Vector search latency
- Resource utilization (CPU, memory, disk)

Health Checks

- Liveness probe (application running)
- Readiness probe (ready to accept traffic)
- Database connectivity check
- Redis connectivity check
- External API availability check
- Disk space monitoring

Screen Inventory

Public Screens

1. Landing page
2. Pricing page
3. Documentation site
4. Login page
5. Signup page
6. Password reset request page
7. Password reset confirmation page
8. Email verification page

Dashboard Screens

9. Dashboard home (overview)
10. Agents list page
11. Create agent wizard (multi-step)

12. Edit agent page
13. Agent analytics page
14. Agent test playground
15. Knowledge base home
16. Document upload page
17. URL crawler page
18. Document library (list/grid view)
19. Document preview modal
20. Document version history
21. Widget configuration page
22. Widget preview page
23. Integration code page
24. Analytics overview
25. Conversation analytics page
26. Question analytics page
27. Source analytics page
28. Feedback analytics page
29. Workspace settings
30. API key management
31. Team management page
32. Billing and subscription page
33. User profile page
34. Notification center
35. Onboarding tour screens

Widget Screens

36. Widget launcher button
37. Widget chat window
38. Widget welcome screen
39. Widget conversation interface
40. Widget source preview modal
41. Widget feedback form
42. Widget escalation form
43. Widget settings panel (for users)

API/Developer Screens

44. API documentation page
45. API playground
46. Webhook management page
47. Developer portal home
48. API logs viewer
49. SDK downloads page

Key User Flows

Flow 1: New User Onboarding & First Agent Creation

Goal: Get a user from signup to deployed chatbot in under 10 minutes

1. User lands on homepage, clicks "Start Free"

2. User enters email and password, agrees to terms
3. Email verification sent, user clicks link
4. User redirected to dashboard with welcome modal
5. Onboarding tour begins: "Let's create your first AI agent"
6. Step 1: Name your agent and choose type (e.g., "Customer Support")
7. Step 2: Upload knowledge (drag PDF or enter website URL)
8. System processes document/URL (shows progress)
9. Step 3: Customize appearance (choose colors, position)
10. Step 4: Preview agent in simulated chat
11. User tests agent with sample questions
12. Step 5: Copy integration code
13. System shows installation guide for common platforms
14. User completes setup, sees "Agent Live" confirmation
15. Dashboard shows first agent as active
16. Optional: Tour continues to show analytics and knowledge management

Success Criteria:

- 70% of signups complete first agent creation
- Average time to deployment: 8 minutes
- 50% of users test agent before deploying

Edge Cases:

- Document processing fails → Show error, suggest smaller file or different format
- No valid content extracted → Prompt to add more documents or use manual entry
- Integration code not copied → Send email with integration guide

Flow 2: Uploading and Managing Knowledge Base

Goal: Enable users to build comprehensive knowledge base efficiently

1. User navigates to Knowledge section
2. Dashboard shows current documents and storage usage
3. User clicks "Add Knowledge" button
4. Modal presents three options: Upload Files, Crawl Website, Manual Entry
5. **Option A - File Upload**:
    - User drags multiple PDFs into upload zone
    - Files queue and process one by one
    - Progress bar shows extraction and embedding status
    - Completed files appear in document library
    - User can click to preview extracted content
6. **Option B - Website Crawl**:
    - User enters domain URL
    - System detects sitemap and shows page count
    - User adjusts crawl depth and page limit
    - Crawl initiates with real-time progress
    - Extracted pages appear as individual documents
    - User can exclude specific pages
7. **Option C - Manual Entry**:
    - Rich text editor opens
    - User pastes or types content
    - User adds title and tags
    - Save creates new document

8. User organizes documents into collections (e.g., "Product Docs", "FAQs")
9. User assigns specific collections to specific agents
10. User tests knowledge using "Ask a Question" preview tool
11. System shows which sources were used and confidence score
12. User identifies gaps and adds more content
13. User creates new version of knowledge base
14. User publishes updated version to agents

Success Criteria:

- Average documents per workspace: 20+
- 80% of uploads process successfully
- Users create at least 2 collections
- Knowledge base tested before publishing: 60%

Edge Cases:

- Upload fails → Retry automatically, then show error with suggestion
- Website blocks crawler → Show error, suggest manual robots.txt check
- Duplicate content detected → Prompt to merge or keep separate
- Large file causes timeout → Process in chunks, show detailed progress

Flow 3: Embedding Widget on Website

Goal: Make integration as simple as possible for non-technical users

1. User navigates to Widget section
2. Dashboard shows integration status (Not Installed / Installed)
3. User clicks "Customize Widget"
4. Split screen: settings on left, live preview on right
5. User selects agent to embed
6. **Appearance Customization**:
    - Choose theme (Light/Dark)
    - Pick brand color from color picker
    - See changes instantly in preview
    - Adjust position (bottom-right, etc.)
    - Upload custom avatar
7. **Behavior Configuration**:
    - Set greeting message
    - Add suggested questions
    - Configure auto-show delay
    - Enable/disable features (file upload, sources, feedback)
8. User clicks "Get Integration Code"
9. Code snippet displayed with copy button
10. Installation guide shows platform-specific instructions:
    - WordPress (plugin or manual)
    - Shopify (app or code injection)
    - Webflow (embed code)
    - HTML website (header script)
    - React/Next.js (npm package)
11. User selects their platform
12. Step-by-step instructions with screenshots
13. User copies code and follows guide
14. User pastes code into their website

15. System detects first widget load (webhook or heartbeat)
16. Dashboard updates to "Installed" status
17. User sees first conversation appear in analytics
18. Success confirmation modal with next steps

Success Criteria:

- 60% of users attempt widget installation
- Average time from code copy to first load: 15 minutes
- 40% of widgets go live within 24 hours of account creation

Edge Cases:

- Widget conflicts with existing chat → Show troubleshooting guide
- CORS errors → Explain domain whitelisting
- Widget not appearing → Checklist to debug (API key valid, domain allowed, script loaded)
- User doesn't have website access → Offer "Test Mode" to preview on demo site

Flow 4: Monitoring Conversations & Analytics

Goal: Provide actionable insights to improve agent performance

1. User navigates to Analytics section
2. Overview dashboard loads with key metrics
3. User sees: 127 conversations this week (+23% vs last week)
4. User clicks into Conversation Analytics
5. **Top Questions Analysis**:
    - User sees list of most asked questions
    - User notices "What is your refund policy?" asked 45 times
    - User clicks on question to see all instances
    - User reviews how agent responded
    - User identifies inconsistent answers
6. **Unanswered Questions Review**:
    - User clicks "Unanswered" tab
    - System shows 23 questions with low confidence
    - Questions grouped by topic: "Pricing" (8), "Shipping" (7)
    - User realizes shipping info missing from knowledge base
7. **Taking Action**:
    - User clicks "Add Missing Content" button
    - Redirected to Knowledge section with topic pre-filled
    - User uploads shipping policy document
    - System re-processes and notifies when complete
8. **Feedback Analysis**:
    - User navigates to Feedback tab
    - 87% positive feedback ratio
    - User filters by negative feedback
    - Reads user comments: "Agent couldn't help with order tracking"
    - User notes this as Phase 2 feature need
9. **Source Performance**:
    - User checks which documents are most useful
    - Identifies 3 documents never referenced
    - User archives unused documents to reduce noise
10. **Exporting Data**:
    - User clicks "Export Report"

    - Selects date range and metrics
    - Downloads CSV for stakeholder presentation
11. **Scheduled Reports**:
    - User sets up weekly email report
    - Selects metrics to include
    - Adds team emails to distribution list

Success Criteria:

- 80% of active users check analytics weekly
- Average session duration in analytics: 8+ minutes
- 30% of users export data within first month
- 50% reduction in unanswered questions after content updates

Edge Cases:

- No conversations yet → Show sample data with tutorial
- Data too sparse for insights → Suggest ways to drive traffic
- Negative feedback spike → Alert user with investigation prompt

Flow 5: Iterating and Improving Agent

Goal: Enable continuous improvement of agent quality

1. User receives notification: "Your agent has 15 new unanswered questions"
2. User clicks notification, lands on Questions dashboard
3. User reviews unanswered questions clustered by topic
4. User identifies pattern: customers asking

about custom orders 5. User clicks "Add Content" for this topic 6. **Knowledge Update Flow**:

- User uploads "Custom Orders Policy" PDF
- System processes and embeds content
- User tests agent with previously unanswered questions
- Agent now answers confidently with new source

7. **Agent Configuration Refinement**:
    - User notices agent is too verbose
    - User goes to Agent Settings
    - Changes response style from "Detailed" to "Brief"
    - Adjusts max response length to 150 words
    - Tests change in playground
8. **Tone Adjustment**:
    - User gets feedback that agent seems "robotic"
    - User edits personality settings
    - Changes tone from "Professional" to "Friendly"
    - Adds custom prompt: "Use conversational language and empathy"
    - Saves changes
9. **A/B Testing** (Phase 2):
    - User creates version 2 of agent with different greeting
    - System splits traffic 50/50
    - After 100 conversations, analytics show v2 has 20% higher engagement
    - User promotes v2 to primary
10. **Publishing Update**:
    - User reviews all changes in preview

    - User creates version note: "Added custom orders info, improved tone"
    - User clicks "Publish"
    - Changes deploy instantly
    - Widget on website updates automatically
11. **Monitoring Impact**:
    - User checks analytics next day
    - Unanswered questions decreased from 15 to 3
    - User satisfaction increased from 82% to 89%
    - User notes success in dashboard activity log

Success Criteria:

- 60% of users update knowledge base at least once per month
- Average 3 agent configuration changes per user
- Measurable improvement in satisfaction after iterations
- Version control used by 40% of active users

Edge Cases:

- Changes make agent worse → User rolls back to previous version
- Knowledge conflict (old vs new documents) → System highlights conflicts, user resolves
- Too many versions → System suggests archiving old versions

Success Metrics

North Star Metric

Active Conversations per Workspace: Measures the value users get from the platform. Target: 500 conversations/month per active workspace by Month 6.

Product Metrics

Activation Metrics

- Signup to first agent created: Target 70% completion within 24 hours
- Agent created to widget installed: Target 50% within 1 week
- Widget installed to first conversation: Target 80% within 72 hours
- Time to first value (signup to first conversation answered): Target <24 hours median

Engagement Metrics

- Daily Active Workspaces (DAW): Target 40% of total workspaces
- Weekly Active Workspaces (WAW): Target 70% of total workspaces
- Average conversations per workspace per week: Target 50+
- Average session duration in dashboard: Target 12+ minutes
- Feature adoption rate: Target 60% use knowledge management, 40% use analytics

Retention Metrics

- Week 1 retention: Target 60%
- Week 4 retention: Target 40%
- Month 3 retention: Target 30%
- Workspace churn rate: Target <5% monthly

Quality Metrics

- Agent answer confidence score: Target average 75%+
- User satisfaction (thumbs up ratio): Target 85%+
- Unanswered question rate: Target <10% of total
- Average response time: Target <2 seconds
- Widget uptime: Target 99.5%+

Growth Metrics

- Virality coefficient: Target 0.3 (widget visibility drives signups)
- Referral rate: Target 15% of new signups from referrals
- Workspace growth rate: Target 20% month-over-month
- Knowledge base growth: Target 10+ documents per active workspace

Business Metrics (Phase 2)

Revenue Metrics

- Monthly Recurring Revenue (MRR): Target $10k by Month 12
- Average Revenue Per User (ARPU): Target $50/month
- Conversion rate (Free to Paid): Target 5%
- Revenue retention: Target 95%+ net retention
- Customer Acquisition Cost (CAC): Target <$100
- Lifetime Value (LTV): Target >$600
- LTV:CAC ratio: Target 6:1+

Usage-Based Metrics

- Average messages per paying workspace: Target 5,000/month
- Storage utilization: Target 60% of allocated storage used
- API usage: Target 30% of workspaces use API
- Overage revenue: Target 20% of total revenue

Customer Health Metrics

- Net Promoter Score (NPS): Target 50+
- Customer Satisfaction Score (CSAT): Target 4.5/5
- Support ticket volume: Target <5% of active users/month
- Average ticket resolution time: Target <24 hours

Technical Metrics

Performance Metrics

- API response time (p95): Target <500ms
- Widget load time: Target <1 second
- LLM response time: Target <3 seconds
- Database query time (p95): Target <100ms
- Embedding generation time per document: Target <30 seconds

Reliability Metrics

- System uptime: Target 99.9%
- Error rate: Target <0.1%
- Failed document processing rate: Target <2%
- Webhook delivery success rate: Target >98%

Scalability Metrics

- Concurrent users supported: Target 10,000+
- Peak requests per second: Target 1,000+ rps
- Vector search latency at 1M vectors: Target <200ms
- Database connection pool utilization: Target <70%

Leading Indicators (Early Warning System)

Negative Signals

- 3+ days without login: At-risk user
- 0 documents uploaded after 7 days: Activation failure
- <10 conversations in first month: Low engagement
- 3+ consecutive negative feedbacks: Quality issue
- 50%+ unanswered question rate: Knowledge gap

Positive Signals

- Daily logins in first week: High engagement
- 10+ documents in first week: Power user
- Widget on multiple domains: Expansion usage
- API usage: Developer adoption
- Team member invites: Growth potential

Measurement & Tracking

Analytics Implementation

- Mixpanel/Amplitude for product analytics
- PostHog for session replay and heatmaps
- Custom events for all key actions
- Cohort analysis for retention tracking
- Funnel analysis for conversion optimization

Dashboards

- Real-time operations dashboard (uptime, errors, performance)
- Weekly product dashboard (engagement, retention, feature adoption)
- Monthly growth dashboard (new users, revenue, churn)
- Customer health dashboard (NPS, CSAT, support tickets)

Reporting Cadence

- Daily: Critical metrics review (uptime, errors, signups)
- Weekly: Team review (engagement, feature usage, wins/losses)
- Monthly: Stakeholder review (growth, revenue, roadmap)
- Quarterly: Strategic review (north star, retention, expansion)

Out of Scope

Phase 1 (College Project MVP) Explicitly Excludes:

Advanced AI Features

- Custom fine-tuned models (using pre-trained models only)
- Voice/audio input and output
- Image generation capabilities

- Real-time translation (using pre-translation only)
- Sentiment analysis (basic only, no ML training)
- Intent classification with custom training

Enterprise Features

- Single Sign-On (SSO) integration
- SAML/OAuth enterprise protocols
- SOC 2 compliance certification
- HIPAA compliance
- On-premise deployment options
- Virtual Private Cloud (VPC) hosting
- Service Level Agreements (SLAs)
- Dedicated support team
- Custom contract terms

Advanced Integrations

- CRM integrations (Salesforce, HubSpot)
- Support desk integrations (Zendesk, Freshdesk)
- Calendar booking systems (Calendly, Cal.com)
- Payment processing for lead qualification
- Marketing automation platforms
- Email service providers (beyond basic transactional)
- Slack/Teams native integrations

Action-Taking Capabilities

- Ticket creation in external systems
- Automated email sending on user's behalf
- Calendar event creation
- Database write operations
- File upload to external storage
- Form submission automation
- Meeting scheduling

Advanced Analytics

- Machine learning-based insights
- Predictive analytics
- Custom reporting builder
- Data warehouse integration
- Business intelligence tool exports
- Advanced cohort analysis
- Funnel optimization recommendations

Multi-User Collaboration

- Team workspaces with multiple seats
- Role-based permissions (Admin/Editor/Viewer)
- Collaborative editing
- Comments and annotations
- Approval workflows
- Activity streams per user
- Granular access control

White-Label/Reseller

- Complete white-labeling
- Custom domain hosting (chat.clientdomain.com)
- Reseller/agency mode
- Client workspace management
- Consolidated billing
- Sub-account hierarchies

Advanced Security

- Two-factor authentication (2FA)
- IP whitelisting
- Advanced audit logs
- Compliance reporting
- Data residency options
- Encryption key management
- Security questionnaire responses

Mobile Applications

- Native iOS app
- Native Android app
- Mobile SDK
- Push notifications
- Offline mode

Advanced Widget Features

- Video chat escalation
- Screen sharing
- Co-browsing
- File sharing in chat
- Rich media messages (carousels, cards)
- Payment collection in widget
- Form builder integration

Deliberately Deferred to Phase 2:

Performance Optimizations

- Advanced caching strategies
- CDN for global widget delivery
- Edge computing for low latency
- Database sharding
- Load balancing across regions

Advanced Developer Tools

- GraphQL API
- WebSocket real-time API
- Client SDKs (Python, Ruby, PHP, Go)
- CLI tool for management
- Terraform provider
- GitHub Actions integration

AI/ML Enhancements

- Automatic answer improvement from feedback
- Conversation clustering and insights

- Anomaly detection in usage patterns
- Automated content suggestions
- Question paraphrasing detection

Never in Scope (Platform Limitations):

Compliance & Legal

- Legal advice on chatbot regulations
- GDPR compliance guarantees (provide tools, but user responsible)
- Industry-specific certifications (financial, healthcare)
- Content moderation liability

Content Creation

- Automatic knowledge base generation from websites
- AI-written documentation
- Content translation services
- Plagiarism checking

Hardware/Infrastructure

- Physical server hosting
- Custom hardware requirements
- Dedicated infrastructure per customer (in free/starter tiers)

Third-Party Responsibilities

- LLM model training (using existing APIs only)
- Vector database development (using pgvector)
- Website hosting for customers
- Customer's website performance

Development Phases

Phase 0: Foundation Setup (Week 0, ~40 hours)

Objective: Establish development environment and core architecture

Backend Setup (Node.js + Python)

- Initialize Node.js (Express) repository for API gateway and real-time services
- Initialize Python (FastAPI) repository for AI/ML and vector search services
- Database setup: PostgreSQL with pgvector extension on Neon free tier
- Redis setup for caching and session management (Upstash free tier)
- Environment configuration and secrets management
- Docker compose for local development
- API gateway pattern design (Node.js routes to Python services where needed)

Frontend Setup (Next.js)

- Initialize Next.js 14+ project with App Router
- Install TailwindCSS and shadcn/ui components
- Setup folder structure (app, components, lib, hooks)
- Configure TypeScript and ESLint

- Setup environment variables

Infrastructure

- Domain registration (insydr.ai) - $12/year
- Deploy placeholder on Vercel (frontend)
- Deploy backend on Railway/Render free tier
- Setup GitHub repositories and CI/CD basics
- Configure CORS and security headers

Deliverables:

- Runnable local development environment
- Basic "Hello World" API endpoints
- Deployed landing page with "Coming Soon"
- Database schema V1 design document

Phase 1: Core Foundation (Weeks 1-4, ~160 hours)

Sprint 1: Authentication & Multi-Tenancy (Weeks 1-2)

Backend (Node.js/Express - 40 hours)

- User registration endpoint with email validation
- Login endpoint with JWT token generation
- Password reset flow (request + confirm)
- Refresh token rotation logic
- Session management with Redis
- Email service integration (SendGrid/Resend free tier)

Backend (Python/FastAPI - 20 hours)

- Workspace model and CRUD operations
- Row-level security middleware
- Workspace context injection in all queries

Database (10 hours)

- Users table (id, email, password_hash, verified, created_at)
- Workspaces table (id, name, owner_id, created_at, settings)
- Sessions table (id, user_id, token, expires_at)
- API keys table (id, workspace_id, key, domains, created_at)

Frontend (Next.js - 30 hours)

- Signup page with form validation
- Login page with error handling
- Email verification page
- Password reset flow (request + reset pages)
- Dashboard shell with navigation
- Auth context and protected routes

Testing (20 hours)

- Unit tests for auth endpoints
- Integration tests for signup flow
- Security testing (SQL injection, XSS)

Sprint 2: Agent Management Basics (Weeks 3-4)

Backend (Node.js/Express - 30 hours)

- Agent CRUD API endpoints
- Agent configuration validation
- Agent status management (active/inactive)

Backend (Python/FastAPI - 40 hours)

- LLM integration setup (Gemini API / Ollama)
- Basic prompt template system
- Agent prompt compilation from config
- Simple chat endpoint (no RAG yet)
- Token counting and usage tracking

Database (10 hours)

- Agents table (id, workspace_id, name, type, config, status, created_at)
- Conversations table (id, agent_id, session_id, started_at, ended_at)
- Messages table (id, conversation_id, role, content, confidence, created_at)

Frontend (Next.js - 40 hours)

- Agents list page (card/grid view)
- Create agent modal with wizard
- Agent configuration form (name, type, tone, language)
- Agent detail/edit page
- Simple test playground (text input → response)
- Agent status toggle

Testing (20 hours)

- API endpoint tests
- LLM integration tests with mocks
- Frontend component tests

Deliverables:

- Working authentication system
- Multi-tenant workspace isolation
- Ability to create and test basic agents (without knowledge base)
- User dashboard with agent management

Phase 2: Knowledge Base & RAG (Weeks 5-8, ~160 hours)

Sprint 3: Document Upload & Processing (Weeks 5-6)

Backend (Python/FastAPI - 50 hours)

- File upload endpoint with validation
- PDF text extraction (PyPDF2)
- DOCX parsing (python-docx)
- TXT and CSV handling
- Text chunking algorithm (sentence-aware)
- Embedding generation (Gemini Embedding API)
- Store embeddings in pgvector
- Async job queue setup (Celery with Redis)
- Job status tracking

Database (15 hours)

- Documents table (id, workspace_id, title, type, source, status, created_at)
- Chunks table (id, document_id, content, embedding, position, metadata)
- Jobs table (id, workspace_id, type, status, progress, created_at)

Frontend (Next.js - 35 hours)

- Knowledge base home page
- File upload component (drag-and-drop)
- Upload progress tracking
- Document library (list view)
- Document preview modal
- Delete confirmation modal

Testing (20 hours)

- File processing pipeline tests
- Embedding generation tests
- Storage tests

Sprint 4: Vector Search & RAG Integration (Weeks 7-8)

Backend (Python/FastAPI - 60 hours)

- Vector similarity search implementation
- Hybrid search (vector + keyword BM25)
- Relevance ranking algorithm
- Context retrieval for LLM (top-k chunks)
- RAG pipeline: query → retrieve → augment → generate
- Source citation extraction
- Confidence scoring based on retrieval quality
- Response caching logic

Backend (Node.js/Express - 20 hours)

- Knowledge base CRUD endpoints
- Document metadata updates
- Bulk delete operations

Frontend (Next.js - 30 hours)

- Enhanced test playground with source display
- Answer preview system (test questions)
- Retrieved sources visualization
- Confidence score indicator
- Knowledge gap detection UI (unanswered questions log)

Testing (20 hours)

- Vector search accuracy tests
- RAG pipeline end-to-end tests
- Performance benchmarks

Deliverables:

- Complete knowledge base upload system
- Working RAG pipeline with source citation
- Agents can answer from custom knowledge
- Document management UI

Phase 3: Widget SDK & Integration (Weeks 9-12, ~160 hours)

Sprint 5: Widget Core (Weeks 9-10)

Backend (Node.js/Express - 30 hours)

- Public chat API endpoint (no auth, uses agent ID + API key)
- Domain whitelisting validation
- Rate limiting per domain
- CORS configuration
- Widget analytics events ingestion

Widget SDK (JavaScript - 60 hours)

- Core widget JavaScript bundle
- Chat UI components (messages, input, launcher)
- WebSocket/polling for real-time updates
- Session management (localStorage)
- Event system (custom events)
- Initialization and configuration parsing
- API communication layer
- Message rendering with Markdown support
- Typing indicators
- Error handling and retry logic

Frontend (Next.js - 20 hours)

- Widget configuration page (basic settings)
- Integration code generator
- Test mode toggle

Build System (10 hours)

- Webpack/Rollup setup for widget bundling
- Minification and optimization
- CDN upload automation (Cloudflare)

Testing (20 hours)

- Widget cross-browser testing
- Integration tests
- Performance testing

Sprint 6: Widget Customization (Weeks 11-12)

Widget SDK Enhancements (40 hours)

- Theme system (light/dark/auto)
- Color customization (CSS variables)
- Position configuration
- Custom avatar display
- Greeting message with suggested questions
- Source citation display in chat
- Feedback buttons (thumbs up/down)
- Escalation button ("Talk to human")

Backend (Node.js/Express - 20 hours)

- Feedback submission endpoint
- Escalation webhook trigger
- Widget configuration API

Frontend (Next.js - 40 hours)

- Visual widget customizer (split-screen)
- Live preview iframe
- Appearance settings (colors, position, avatar)
- Behavior settings (greeting, suggested questions)
- Platform-specific installation guides
- Copy code functionality
- Installation status tracking

Widget Documentation (20 hours)

- Integration guide for HTML/WordPress/Shopify/React
- API reference for events and methods
- Customization examples
- Troubleshooting guide

Testing (20 hours)

- Visual regression testing
- Customization option tests
- Installation verification

Deliverables:

- Fully functional embeddable widget
- Customization interface with live preview
- Integration guides for popular platforms
- Working demo on test website

Phase 4: Analytics & Insights (Weeks 13-16, ~160 hours)

Sprint 7: Analytics Backend & Data Collection (Weeks 13-14)

Backend (Python/FastAPI - 40 hours)

- Analytics event schema design
- Event ingestion endpoint (high-volume)
- Analytics aggregation queries
- Conversation metrics calculation
- Question frequency analysis
- Unanswered question detection
- Source usage tracking
- Feedback aggregation

Database (15 hours)

- Analytics_events table (id, workspace_id, event_type, data, created_at)
- Aggregated metrics tables (for performance)
- Indexes for common queries

Backend (Node.js/Express - 20 hours)

- Analytics API endpoints

- Date range filtering
- Export functionality (CSV/JSON)
- Caching for expensive queries

Data Processing (25 hours)

- Background jobs for metric aggregation
- Daily/weekly rollup calculations
- Data retention policies

Testing (20 hours)

- Analytics query performance tests
- Data accuracy validation
- Export functionality tests

Sprint 8: Analytics Dashboard UI (Weeks 15-16)

Frontend (Next.js - 80 hours)

- Overview dashboard with stat cards
- Conversation volume charts (Recharts)
- Top questions list with frequency
- Unanswered questions dashboard
- Source performance view
- Feedback analytics page
- Date range selector component
- Filter and search functionality
- Export button with download
- Responsive design for all views
- Loading states and skeletons
- Empty states with helpful CTAs

Frontend Components Library (20 hours)

- Reusable chart components
- Metric card component
- Table with sorting/filtering
- Loading indicators
- Error boundaries

Testing (20 hours)

- Component tests
- Integration tests with mock data
- Visual testing
- Performance testing with large datasets

Deliverables:

- Complete analytics dashboard
- Real-time conversation monitoring
- Actionable insights (unanswered questions, content gaps)
- Export functionality

Phase 5: Polish & Launch Prep (Weeks 17-20, ~160 hours)

Sprint 9: User Experience & Design (Weeks 17-18)

Frontend (Next.js - 60 hours)

- Design system refinement (consistent spacing, colors, typography)
- Onboarding flow (multi-step wizard for first agent)
- Dashboard home redesign (activity feed, quick actions)
- Improved navigation (breadcrumbs, search)
- Help system (tooltips, contextual help)
- Notification center
- User profile page
- Settings pages (workspace, API keys)
- Dark mode support
- Mobile responsive improvements

UX Improvements (30 hours)

- Error message improvements
- Success confirmations with undo
- Loading state standardization
- Empty state designs
- Form validation improvements
- Accessibility audit and fixes (ARIA labels, keyboard navigation)

Testing (20 hours)

- User acceptance testing
- Usability testing with real users
- Accessibility testing (WCAG 2.1 AA)

Sprint 10: Performance, Security & Documentation (Weeks 19-20)

Performance Optimization (30 hours)

- Database query optimization
- API response time improvements
- Widget load time optimization
- Image optimization
- Code splitting for frontend
- Lazy loading implementation
- Caching strategy refinement

Security Hardening (25 hours)

- Security audit of all endpoints
- Input validation improvements
- Rate limiting tuning
- API key security enhancements
- XSS and CSRF protection verification
- Secrets rotation procedures
- Security headers configuration

Documentation (35 hours)

- User documentation (Getting Started, How-to Guides)
- API documentation (Swagger/OpenAPI)
- Developer guide (SDK usage, webhooks)
- FAQ and troubleshooting
- Video tutorials (screencasts)
- Changelog setup

Infrastructure (20 hours)

- Production deployment setup
- Monitoring and alerting (UptimeRobot, error tracking)
- Backup and disaster recovery
- Database migration procedures
- SSL certificate setup
- CDN configuration

Testing & QA (30 hours)

- End-to-end testing (Playwright/Cypress)
- Load testing (K6 or Artillery)
- Security testing (penetration testing basics)
- Cross-browser testing
- Mobile device testing
- Bug fixing from all testing

Deliverables:

- Production-ready application
- Complete documentation
- Optimized performance (page load <2s, API <500ms)
- Security hardened
- Monitoring and alerting in place

Phase 6: Beta Launch & Iteration (Weeks 21-24, ~160 hours)

Sprint 11: Beta Launch (Weeks 21-22)

Pre-Launch (30 hours)

- Final QA checklist
- Load testing with expected traffic
- Backup verification
- Rollback procedures documented
- Launch announcement content
- Support system setup (help desk)

Launch Activities (20 hours)

- Beta user onboarding (10-20 users)
- User interviews and feedback collection
- Bug triage and prioritization
- Performance monitoring
- Usage analytics review

Marketing Website (30 hours)

- Landing page redesign
- Features page
- Pricing page
- Documentation site
- Blog setup for announcements
- SEO optimization

Support & Community (20 hours)

- Email support setup
- FAQ based on beta feedback
- Community Slack/Discord setup
- User feedback form
- Feature request board

Sprint 12: Iteration & Improvement (Weeks 23-24)

Based on Beta Feedback (80 hours)

- High-priority bug fixes (estimated 30 hours)
- UX improvements based on user feedback (estimated 25 hours)
- Performance improvements (estimated 15 hours)
- Documentation updates (estimated 10 hours)

Feature Refinement (40 hours)

- Agent configuration improvements
- Widget customization enhancements
- Analytics additions based on requests
- Knowledge base workflow improvements

Preparation for Public Launch (20 hours)

- Public launch plan
- Marketing materials
- Press release draft
- Social media content
- Launch day runbook

Final Testing (20 hours)

- Regression testing after all changes
- User acceptance testing round 2
- Performance verification
- Security review

Deliverables:

- Stable beta version with 20+ active users
- Refined product based on real feedback
- Comprehensive support resources
- Ready for public launch

Phase 2 (Post-College, Revenue Phase) - High-Level Roadmap

Phase 2A: Monetization Foundation (Months 6-8)

Billing & Payments

- Stripe integration for subscriptions
- Pricing tier enforcement (message limits, storage)
- Usage tracking and overage billing
- Invoice generation
- Payment method management

Team & Collaboration

- Team workspace support
- Role-based permissions (Admin, Editor, Viewer)
- Invite team members
- Activity logs and audit trail
- Workspace member management

API & Developer Tools

- Full REST API documentation
- API playground
- Webhook system for events
- API rate limiting by plan
- Developer portal

Phase 2B: Action-Taking Agents (Months 9-11)

Integrations Framework

- CRM integrations (Salesforce, HubSpot)
- Support desk integrations (Zendesk, Freshdesk)
- Calendar booking (Calendly, Cal.com)
- Email notifications
- Slack/Teams alerts

Sales Agent Features

- Lead qualification flows
- Contact information capture
- Meeting scheduling
- CRM sync
- Lead scoring

Support Agent Features

- Ticket creation
- File upload handling
- Priority detection
- Status tracking
- SLA monitoring

Phase 2C: Enterprise Features (Months 12+)

Enterprise Security

- SSO (SAML, OAuth)
- Two-factor authentication
- IP whitelisting
- Advanced audit logs
- SOC 2 compliance path

White-Label & Reseller

- Custom domain support
- Complete branding removal
- Agency/reseller mode
- Client workspace management
- Consolidated billing

Advanced Analytics

- Conversation intelligence
- Intent classification
- Sentiment analysis
- Predictive analytics
- A/B testing framework

Total Estimated Timeline

Phase 1 (College Project): 24 weeks (6 months)

- ~800 total development hours
- Assumes 1-2 developers working part-time (20-30 hrs/week)
- Buffer included for learning curve and unexpected issues

Phase 2 (Revenue Ready): 6-12 months

- Post-graduation, full-time development
- Customer feedback driven
- Iterative releases

Privacy and Safety

Data Privacy Principles

Data Minimization

- Collect only essential data required for functionality
- No unnecessary tracking or profiling
- Users can delete their data at any time
- Automatic data cleanup for inactive accounts (after 6 months warning)

User Data Ownership

- Users own all content uploaded to knowledge base
- Users own all conversation data
- Clear data export functionality (download all data as JSON/CSV)
- Data portability in standard formats

Transparency

- Clear privacy policy in plain language
- Explain what data is collected and why
- Notify users of privacy policy changes
- No hidden tracking or analytics without consent

Data Security Measures

Encryption

- All data encrypted in transit (TLS 1.3)
- Sensitive data encrypted at rest (database level)
- API keys hashed using secure algorithms
- Passwords hashed with bcrypt (cost factor 12)

Access Control

- Row-level security for multi-tenancy

- API key authentication for widget
- JWT tokens with short expiration (15 minutes)
- Refresh token rotation on every use
- Role-based access control (RBAC)

Security Best Practices

- Regular security audits
- Dependency vulnerability scanning (npm audit, safety)
- SQL injection prevention (parameterized queries)
- XSS protection (sanitized inputs/outputs)
- CSRF tokens for state-changing operations
- Rate limiting on all endpoints
- API key domain whitelisting

Privacy Compliance

GDPR Compliance (EU Users)

- Right to access (data export)
- Right to deletion (account deletion deletes all data)
- Right to rectification (users can edit their data)
- Right to data portability (JSON/CSV export)
- Consent for data processing (privacy policy acceptance)
- Data processing agreements for sub-processors (LLM providers)

CCPA Compliance (California Users)

- Disclosure of data collection practices
- Opt-out of data sale (we don't sell data, clearly stated)
- Right to deletion
- Non-discrimination for exercising privacy rights

Cookie Policy

- Essential cookies only for authentication (no tracking cookies in Phase 1)
- Cookie consent banner for EU users
- Clear explanation of cookie usage

Content Safety & Moderation

User-Generated Content

- Users responsible for content they upload
- Terms of Service prohibit illegal, harmful, or abusive content
- Content moderation filters for obvious violations
- Abuse reporting mechanism
- Account suspension for ToS violations

AI Safety Measures

- LLM safety filters to prevent harmful outputs
- Refusal to answer dangerous/illegal queries (built into prompts)
- No generation of: hate speech, explicit content, illegal instructions, harmful medical advice
- Confidence thresholds prevent hallucinated misinformation
- Source citation reduces misinformation spread

Child Safety

- No data collection from users under 13 (COPPA compliance in US)
- Age verification on signup (self-reported)
- Additional protections for 13-17 age group (if applicable)

Third-Party Data Processing

LLM Provider Data Sharing

- Clear disclosure that queries are sent to LLM APIs (Gemini, OpenAI)
- Links to LLM provider privacy policies
- Option to use self-hosted models in future (Ollama)
- No training on user data without explicit consent
- Data processing agreements with providers

Sub-Processors Disclosure

- Clear list of all third-party services used:
  - LLM providers (Gemini API)
  - Email service (SendGrid/Resend)
  - Hosting (Vercel, Railway/Render, Neon)
  - Analytics (Mixpanel/Amplitude) - only if user consents
  - CDN (Cloudflare)

User Safety Features

Conversation Safety

- Users can report inappropriate agent responses
- Conversation history deletion at any time
- Option to disable conversation logging
- No PII collection unless explicitly provided by user

Widget Safety

- No data collection beyond conversation content
- No tracking across websites
- Domain whitelisting prevents unauthorized widget usage
- Clear "Powered by Insydr" badge (unless removed on paid plans)

Data Breach Protocol

- Incident response plan documented
- User notification within 72 hours (GDPR requirement)
- Clear communication about what data was affected
- Steps taken to prevent recurrence
- Offer of remediation (password reset, monitoring)

Platform Abuse Prevention

Spam & Abuse

- Rate limiting on all endpoints (100 req/hour per IP for public endpoints)
- CAPTCHA on signup to prevent bot accounts
- Email verification required
- Suspicious activity detection (unusual API usage patterns)
- Account flagging system

Resource Abuse

- Storage quotas per plan
- Message limits per plan
- API rate limits per workspace
- Automatic throttling for excessive usage
- Overage alerts before blocking

Compliance & Legal

Terms of Service

- Clear acceptable use policy
- Prohibited uses (illegal content, abuse, spam)
- Service availability disclaimers (no SLA in free tier)
- Intellectual property rights (users retain ownership)
- Limitation of liability
- Dispute resolution mechanism

Disclaimers

- No warranty on AI accuracy
- Users responsible for verifying AI outputs
- Not a substitute for professional advice (legal, medical, financial)
- Service provided "as-is" in free tier

Legal Requirements

- DMCA compliance (copyright takedown process)
- Abuse reporting mechanism (abuse@insydr.ai)
- Law enforcement cooperation policy
- Data retention for legal purposes

Future Privacy Enhancements (Phase 2)

Advanced Security

- Two-factor authentication (2FA)
- SSO for enterprise users
- SOC 2 Type II compliance
- Annual security audits
- Bug bounty program

Enhanced Privacy

- Zero-knowledge encryption option (encrypt before storing)
- Self-hosted deployment option (on-premise)
- HIPAA compliance for healthcare customers
- Enhanced data residency options (EU-only data centers)

Definition of Scope

In Scope - Must Have for College Project

Core User Journey

1. User can sign up and create an account
2. User can create at least one AI agent

3. User can upload documents (PDF, DOCX, TXT) to knowledge base
4. User can embed a widget on their website
5. Visitors can ask questions and receive answers from knowledge base
6. User can view analytics about conversations

Critical Features (MVP)

- User authentication (signup, login, password reset)
- Single workspace per user
- Agent creation with basic configuration (name, type, tone)
- File upload and processing (PDF, DOCX, TXT, CSV)
- Vector embeddings and similarity search
- RAG pipeline (retrieve context + generate answer)
- Embeddable widget with basic customization
- Basic analytics dashboard (conversation count, top questions)
- Source citation in responses

Technical Requirements

- Multi-tenant architecture (even if single user per workspace initially)
- PostgreSQL with pgvector
- Free LLM API integration (Gemini or Ollama)
- Node.js/Express backend + Python/FastAPI for ML operations
- Next.js frontend
- Deployed and accessible via web (Vercel + Railway/Render)
- Responsive design (desktop + mobile)

Success Criteria for College Demo

- Live demo where instructor can:
  - Sign up as new user
  - Create an agent
  - Upload a sample document (college FAQ)
  - Embed widget on a test page
  - Ask questions and get accurate answers
  - View analytics showing the conversation
- Total cost: <$20/month
- Working project submitted on time

In Scope - Should Have if Time Permits

Enhanced Features (Nice to Have in Phase 1)

- Website crawling for knowledge ingestion
- Agent versioning (v1, v2 rollback)
- Knowledge base collections/categories
- Answer preview/testing before publishing
- Multiple API keys per workspace
- Webhook for conversation events (escalation, feedback)
- Manual text entry for knowledge
- Advanced widget customization (position, colors, avatar)
- Feedback system (thumbs up/down)
- Unanswered question tracking

These are aspirational - include if ahead of schedule

Out of Scope - Phase 2 Only

Post-Graduation Features

- Payment processing and billing (Stripe)
- Paid subscription tiers with limits
- Team workspaces with multiple users
- Role-based permissions
- CRM integrations (Salesforce, HubSpot)
- Support desk integrations (Zendesk, Freshdesk)
- Calendar booking integrations
- Action-taking agents (create tickets, send emails)
- SSO and enterprise authentication
- White-label options
- Custom domains for widget
- Advanced analytics (ML insights, predictions)
- A/B testing for agents
- Fine-tuning custom models

Explicitly Out of Scope - Never

Features We Will Not Build

- Voice/audio chatbot (outside scope of project)
- Video chat (too complex)
- Mobile native apps (web-only)
- Blockchain/crypto integration (unnecessary)
- On-device AI processing (server-side only)
- Custom hardware (software only)
- Legal compliance certifications (SOC 2, HIPAA) in Phase 1
- Human handoff to live agents in Phase 1 (webhook only)
- Multi-language UI (English only for dashboard, agents can detect user language)

Boundary Definitions

What "Agent" Means in Phase 1

- IN SCOPE: Configurable Q&A bot that answers from knowledge base
- OUT OF SCOPE: Autonomous agent that takes actions (creates tickets, sends emails)
- IN SCOPE: Multiple agents with different knowledge bases and personalities
- OUT OF SCOPE: Agents that learn and improve automatically over time

What "Knowledge Base" Means in Phase 1

- IN SCOPE: Documents uploaded by user, chunked and embedded
- OUT OF SCOPE: Automatic web scraping without user initiation
- IN SCOPE: Manual text entry and file uploads
- OUT OF SCOPE: Automatic knowledge graph generation

What "Analytics" Means in Phase 1

- IN SCOPE: Conversation count, message volume, top questions, unanswered questions
- OUT OF SCOPE: AI-powered insights, predictions, recommendations
- IN SCOPE: CSV/JSON export of data
- OUT OF SCOPE: Automated reports with insights emailed to users

What "Customization" Means in Phase 1

- IN SCOPE: Widget appearance (colors, position, avatar)

- OUT OF SCOPE: Custom widget code/templates
- IN SCOPE: Agent behavior settings (tone, response length)
- OUT OF SCOPE: Custom LLM fine-tuning

What "Multi-Tenant" Means in Phase 1

- IN SCOPE: Complete data isolation between workspaces at database level
- IN SCOPE: Separate agents, knowledge bases, API keys per workspace
- OUT OF SCOPE: Team members within a workspace (single user only)
- OUT OF SCOPE: Cross-workspace reporting or management

Dependencies & Constraints

Technical Constraints

- Must use free/freemium services (cost constraint)
- Must use PostgreSQL + pgvector (specific to project goals)
- Must be deployable without ongoing costs >$20/month
- Must work in common browsers (Chrome, Firefox, Safari, Edge)

Time Constraints

- 24 weeks (6 months) for Phase 1 completion
- Average 20-30 hours per week development time
- Final 2 weeks reserved for testing and documentation

Resource Constraints

- 1-2 developers (college project team size)
- No budget for paid services beyond domain ($12/year)
- Limited to free API quotas (Gemini: 15 req/min, 1500 req/day)

Skill Constraints

- Built by college students (intermediate level)
- Learning curve for new technologies factored in
- Possible need for additional learning time on:
  - Vector databases (pgvector)
  - LLM prompt engineering
  - Widget SDK development

Risk Mitigation

Scope Creep Prevention

- Weekly review of progress vs. plan
- Features added only if ahead of schedule
- "Nice to have" features moved to Phase 2 if falling behind
- Focus on core user journey above all else

Technical Risks

- LLM API rate limits → Use caching aggressively, implement queue
- Vector search performance → Start with small datasets, optimize later
- Widget compatibility issues → Test on major platforms early
- Embedding cost → Use smallest model (text-embedding-3-small), batch processing

Timeline Risks

- Falling behind schedule → Cut "should have" features, keep "must have"

- Unexpected technical challenges → Allocate 20% buffer time in each sprint
- External dependencies down → Have fallback for critical services

Validation Checkpoints

Milestone 1 (Week 4): Can a user sign up and create a basic agent? **Milestone 2 (Week 8)**: Can an agent answer questions from uploaded documents? **Milestone 3 (Week 12)**: Can a widget be embedded and function on a website? **Milestone 4 (Week 16)**: Are analytics available and useful? **Milestone 5 (Week 20)**: Is the system ready for beta users? **Milestone 6 (Week 24)**: Is the project ready to demonstrate and submit?

If any milestone is missed by >2 weeks, scope must be reduced.

Design System

Brand Identity

Name: Insydr AI

Tagline: "Intelligence embedded in every conversation"

Brand Personality

- Modern and professional
- Approachable and friendly
- Innovative but reliable
- Technical without being intimidating

Logo Concept (to be designed)

- Incorporates chat bubble or conversation motif
- Clean, minimalist design
- Works in monochrome and color
- Scalable from favicon to billboard

Color System

Primary Colors

Insydr Blue (Brand Primary)
- Primary: #2563EB (blue-600)
- Hover: #1D4ED8 (blue-700)
- Light: #DBEAFE (blue-100)
- Usage: Primary buttons, links, active states

Secondary Colors

Insydr Purple (Accent)
- Secondary: #7C3AED (violet-600)
- Hover: #6D28D9 (violet-700)
- Light: #EDE9FE (violet-100)

- Usage: Secondary actions, highlights

Neutral Colors

Gray Scale
- Gray-50: #F9FAFB (backgrounds, cards)
- Gray-100: #F3F4F6 (hover states)
- Gray-200: #E5E7EB (borders, dividers)
- Gray-300: #D1D5DB (disabled states)
- Gray-400: #9CA3AF (placeholders)
- Gray-500: #6B7280 (secondary text)
- Gray-600: #4B5563 (body text)
- Gray-700: #374151 (headings)
- Gray-800: #1F2937 (dark backgrounds)
- Gray-900: #111827 (primary text)

Semantic Colors

Success
- Green-600: #16A34A (success state)
- Green-100: #DCFCE7 (success background)

Warning
- Yellow-500: #EAB308 (warning state)
- Yellow-100: #FEF9C3 (warning background)

Error
- Red-600: #DC2626 (error state)
- Red-100: #FEE2E2 (error background)

Info
- Blue-600: #2563EB (info state)
- Blue-100: #DBEAFE (info background)

Typography

Font Families

Primary (UI): Inter
- Clean, modern sans-serif
- Excellent readability
- Variable font support
- Weights: 400 (regular), 500 (medium), 600 (semibold), 700 (bold)

Monospace (Code): JetBrains Mono
- For code snippets, API keys, technical content
- Weight: 400 (regular)

Type Scale

Display (Hero Headlines)
- Size: 48px / 3rem
- Line Height: 1.2
- Weight: 700
- Letter Spacing: -0.02em

H1 (Page Titles)
- Size: 36px / 2.25rem
- Line Height: 1.25
- Weight: 700
- Letter Spacing: -0.01em

H2 (Section Titles)
- Size: 30px / 1.875rem
- Line Height: 1.3
- Weight: 600
- Letter Spacing: -0.01em

H3 (Subsection Titles)
- Size: 24px / 1.5rem
- Line Height: 1.4
- Weight: 600

H4 (Card Titles)
- Size: 20px / 1.25rem
- Line Height: 1.4
- Weight: 600

Body Large
- Size: 18px / 1.125rem
- Line Height: 1.6
- Weight: 400

Body (Default)
- Size: 16px / 1rem
- Line Height: 1.6
- Weight: 400

Body Small
- Size: 14px / 0.875rem
- Line Height: 1.5
- Weight: 400

Caption
- Size: 12px / 0.75rem
- Line Height: 1.4
- Weight: 400
- Color: Gray-500

Spacing System

Base Unit: 4px (0.25rem)

Spacing Scale (Tailwind)

0: 0px
1: 4px
2: 8px
3: 12px
4: 16px
5: 20px
6: 24px
8: 32px
10: 40px
12: 48px
16: 64px
20: 80px
24: 96px

Common Patterns

- Card padding: 24px (p-6)
- Section spacing: 48px (gap-12 or mb-12)
- Input padding: 12px vertical, 16px horizontal (py-3 px-4)
- Button padding: 12px vertical, 24px horizontal (py-3 px-6)
- Modal padding: 32px (p-8)

Component Library (shadcn/ui based)

Buttons

Primary Button
- Background: Primary color (#2563EB)
- Text: White
- Padding: 12px 24px
- Border Radius: 8px
- Font Weight: 600
- Hover: Darken 10%, slight lift
- Active: Darken 15%, pressed state
- Disabled: Opacity 50%, no hover

Secondary Button
- Background: White
- Border: 2px solid Gray-300
- Text: Gray-700
- Same padding and radius as primary
- Hover: Background Gray-50
- Active: Background Gray-100

Tertiary Button (Text/Ghost)
- Background: Transparent

- Text: Primary color
- Padding: 8px 16px
- Hover: Background Gray-100
- Active: Background Gray-200

Destructive Button
- Background: Red-600
- Text: White
- Same styling as primary
- Used for delete, remove actions

Icon Button
- Square (40x40px or 32x32px)
- Only icon, no text
- Background transparent or light gray
- Hover: Background change

Input Fields

Text Input
- Height: 44px
- Padding: 12px 16px
- Border: 1px solid Gray-300
- Border Radius: 8px
- Font Size: 16px
- Focus: Border color Primary, outline ring
- Error: Border color Red-600
- Disabled: Background Gray-100, text Gray-400

Textarea
- Same styling as text input
- Min height: 120px
- Resizable vertically

Select Dropdown
- Same styling as text input
- Chevron icon on right
- Dropdown menu with max height, scrollable

Checkbox
- Size: 20x20px
- Border: 2px solid Gray-400
- Border Radius: 4px
- Checked: Background Primary, white checkmark
- Focus: Outline ring

Radio Button
- Size: 20x20px
- Border: 2px solid Gray-400
- Circular
- Selected: Primary color dot inside

- Focus: Outline ring

Toggle Switch
- Width: 44px, Height: 24px
- Background: Gray-300 (off), Primary (on)
- Circle: 20px diameter, white
- Smooth animation on toggle

Cards

Default Card
- Background: White
- Border: 1px solid Gray-200
- Border Radius: 12px
- Padding: 24px
- Shadow: Subtle (0 1px 3px rgba(0,0,0,0.1))
- Hover: Shadow lift (0 4px 12px rgba(0,0,0,0.1))

Stat Card
- Same as default card
- Includes large number (32px), label (14px), trend indicator
- Icon in corner (optional)

Modals & Dialogs

Modal Overlay
- Background: rgba(0, 0, 0, 0.5)
- Backdrop blur: 4px
- Z-index: 100

Modal Content
- Background: White
- Max width: 600px (md size) or 900px (lg size)
- Border Radius: 16px
- Padding: 32px
- Shadow: Large (0 20px 60px rgba(0,0,0,0.2))
- Close button: Top-right corner

Modal Header
- Font Size: 24px
- Font Weight: 600
- Margin Bottom: 16px

Modal Footer
- Margin Top: 24px
- Buttons aligned right
- Primary + Secondary button pattern

Navigation

Top Navigation Bar
- Height: 64px
- Background: White
- Border Bottom: 1px solid Gray-200
- Padding: 0 24px
- Logo on left, user menu on right
- Sticky to top on scroll

Sidebar Navigation
- Width: 260px (collapsible to 64px icon-only)
- Background: Gray-50 or White
- Padding: 16px
- Nav items: 44px height, 12px padding
- Active state: Primary background light, Primary text
- Hover: Background Gray-100
- Icons: 20x20px, left-aligned

Breadcrumbs
- Font Size: 14px
- Color: Gray-600
- Separator: "/" or "›"
- Last item: Gray-900 (current page)
- Hover: Primary color

Tables

Table
- Width: 100%
- Border: 1px solid Gray-200
- Border Radius: 8px
- Overflow: Hidden (for rounded corners)

Table Header
- Background: Gray-50
- Font Weight: 600
- Font Size: 14px
- Text Transform: None
- Padding: 12px 16px
- Border Bottom: 1px solid Gray-200

Table Row
- Padding: 16px
- Border Bottom: 1px solid Gray-100
- Hover: Background Gray-50
- Clickable rows: Cursor pointer

Table Cell
- Padding: 16px
- Font Size: 14px
- Vertical Align: Middle

Badges & Tags

Badge
- Padding: 4px 12px
- Border Radius: 9999px (pill shape)
- Font Size: 12px
- Font Weight: 500

Status Badges
- Active: Green-100 background, Green-700 text
- Inactive: Gray-200 background, Gray-700 text
- Error: Red-100 background, Red-700 text
- Warning: Yellow-100 background, Yellow-700 text

Tags
- Similar to badges
- Padding: 6px 12px
- Border Radius: 6px
- Removable with X icon

Loading States

Spinner
- Size: 24px (small), 40px (medium), 64px (large)
- Border: 3px
- Color: Primary
- Animation: Spin 1s linear infinite

Skeleton Loader
- Background: Gray-200
- Border Radius: 4px
- Animation: Pulse (shimmer effect)
- Match shape of content being loaded

Progress Bar
- Height: 8px
- Background: Gray-200
- Fill: Primary color
- Border Radius: 4px
- Smooth animation

Alerts & Notifications

Alert Box
- Padding: 16px
- Border Radius: 8px
- Border Left: 4px solid (status color)
- Icon on left (20x20px)

- Close button on right

Success Alert
- Background: Green-50
- Border: Green-600
- Icon: Checkmark circle
- Text: Green-800

Error Alert
- Background: Red-50
- Border: Red-600
- Icon: X circle
- Text: Red-800

Warning Alert
- Background: Yellow-50
- Border: Yellow-600
- Icon: Exclamation triangle
- Text: Yellow-900

Info Alert
- Background: Blue-50
- Border: Blue-600
- Icon: Information circle
- Text: Blue-800

Toast Notification
- Similar styling to alerts
- Fixed position: Bottom-right or Top-right
- Width: 350px
- Shadow: Large
- Auto-dismiss after 5 seconds
- Stack multiple toasts vertically

Iconography

Icon System: Lucide React (already integrated with shadcn/ui)

Icon Sizes

- Small: 16x16px (forms, inline)
- Medium: 20x20px (default, buttons)
- Large: 24x24px (headings, prominent actions)
- XL: 32x32px (empty states, illustrations)

Icon Style

- Stroke width: 2px (consistent)
- Rounded corners
- Outlined style (not filled)

Common Icons

- Navigation: Menu, ChevronRight, ArrowLeft, X
- Actions: Plus, Edit, Trash, Download, Upload, Copy

- Status: Check, X, AlertTriangle, Info, Loader
- Media: Image, File, FileText, Link, Eye
- User: User, Users, Settings, LogOut
- Communication: MessageSquare, Send, ThumbsUp, ThumbsDown

Layout System

Grid System

- 12-column grid (Tailwind's grid system)
- Gutter: 24px (gap-6)
- Container max-width: 1280px (max-w-screen-xl)
- Responsive breakpoints:
  - sm: 640px
  - md: 768px
  - lg: 1024px
  - xl: 1280px
  - 2xl: 1536px

Page Layout

```
Dashboard Layout
├── Top Nav (64px height)
├── Sidebar (260px width, collapsible)
└── Main Content
    ├── Page Header (breadcrumbs, title, actions)
    ├── Content Area (cards, tables, forms)
    └── Footer (optional)
```

Spacing Patterns

- Page padding: 32px (p-8)
- Section spacing: 48px (space-y-12)
- Card spacing: 24px (p-6)
- Form spacing: 24px between fields (space-y-6)
- Button group spacing: 12px (gap-3)

Widget Design System

Widget Appearance

```
Launcher Button
- Size: 64x64px (can be configured)
- Border Radius: 50% (circular)
- Background: Primary color
- Icon: MessageSquare (white)
- Shadow: 0 4px 12px rgba(0,0,0,0.15)
- Hover: Slight lift, shadow increase
- Badge: Notification count (top-right, circular, red)

Chat Window
```

- Width: 400px (desktop), 100% (mobile)
- Max Height: 600px (desktop), 100vh (mobile)
- Border Radius: 16px (desktop), 0 (mobile fullscreen)
- Shadow: 0 8px 32px rgba(0,0,0,0.12)
- Background: White

Chat Header
- Height: 64px
- Background: Primary color (customizable)
- Padding: 16px
- Agent avatar + name
- Close button (X icon)
- Color: White text on primary background

Chat Messages Area
- Padding: 16px
- Background: Gray-50
- Scrollable (auto-scroll to bottom)
- User messages: Right-aligned, Primary color background
- Agent messages: Left-aligned, White background
- Timestamp: 12px, Gray-500, below message

Message Bubbles
- Padding: 12px 16px
- Border Radius: 16px
- Max Width: 80%
- User message: Blue-600 background, White text
- Agent message: White background, Gray-900 text
- Shadow: Subtle

Typing Indicator
- Three dots animation
- Gray-400 color
- Height: 40px

Chat Input
- Height: 56px
- Background: White
- Border Top: 1px solid Gray-200
- Padding: 12px 16px
- Text input + Send button

Send Button
- Size: 40x40px
- Background: Primary color
- Icon: Send (white)
- Border Radius: 50%
- Disabled: Gray-300 when input empty

Source Citations
- Font Size: 12px
- Color: Primary color

- Hover: Underline
- Click: Opens modal with full source

Feedback Buttons
- Thumbs up/down icons
- Size: 16x16px
- Color: Gray-400
- Hover: Primary color
- Positioned below agent messages

Responsive Design Principles

Mobile First Approach

- Design for mobile (320px) first
- Progressively enhance for larger screens
- Touch targets: Minimum 44x44px
- Font sizes scale up on larger screens

Breakpoint Strategy

- Mobile: 320px - 767px (single column, full-width)
- Tablet: 768px - 1023px (collapsible sidebar, 2-column grids)
- Desktop: 1024px+ (full sidebar, multi-column grids)

Mobile Adaptations

- Sidebar becomes drawer (hamburger menu)
- Tables become cards on mobile
- Modals become full-screen on mobile
- Widget chat becomes full-screen on mobile
- Reduced padding (16px instead of 32px)
- Larger touch targets

Animation & Transitions

Timing Functions

- Fast: 150ms (hover, focus)
- Default: 250ms (most transitions)
- Slow: 350ms (modals, drawers)
- Easing: cubic-bezier(0.4, 0, 0.2, 1) (default)

Animation Types

- Fade In: Opacity 0 → 1
- Slide In: Transform translateY(10px) → 0, Opacity 0 → 1
- Scale: Transform scale(0.95) → 1
- Spin: Rotate 360deg (for loaders)
- Pulse: Scale 1 → 1.05 → 1 (for emphasis)

Hover States

- Buttons: Background color change + slight lift (translateY(-2px))
- Cards: Shadow increase
- Links: Color change + underline
- Icon buttons: Background color change

Focus States

- Outline: 2px solid Primary color
- Offset: 2px
- Border Radius: Match element
- Remove default browser outline

Dark Mode (Phase 2)

Color Adjustments

Background
- Light: White
- Dark: Gray-900

Surface (Cards)
- Light: White
- Dark: Gray-800

Text
- Light: Gray-900 (primary), Gray-600 (secondary)
- Dark: Gray-100 (primary), Gray-400 (secondary)

Border
- Light: Gray-200
- Dark: Gray-700

Primary remains the same (adjusts automatically with Tailwind)

Dark Mode Toggle

- Positioned in user menu or settings
- Moon/Sun icon
- Instant toggle (no page reload)
- Preference saved to localStorage

Accessibility Standards

WCAG 2.1 Level AA Compliance

- Color contrast: Minimum 4.5:1 for normal text, 3:1 for large text
- Keyboard navigation: All interactive elements accessible via keyboard
- Focus indicators: Visible on all focusable elements
- Alt text: All images and icons have descriptive alt text
- ARIA labels: For screen reader support
- Form labels: Associated with inputs
- Error messages: Clear and descriptive
- Skip links: "Skip to main content" for keyboard users

Semantic HTML

- Use proper heading hierarchy (H1 → H6)
- Use `<button>` for buttons, not `<div>`
- Use `<nav>` for navigation
- Use `<main>` for main content

- Use `<aside>` for sidebars
- Use `<article>` for cards/items

Design Deliverables

For Development Team

1. Component library in Figma (or similar)
2. Style guide document (this section)
3. Icon set with naming conventions
4. Responsive layout templates
5. Widget mockups with all states
6. Empty state illustrations
7. Loading state designs
8. Error state designs

For Testing

1. Accessibility checklist
2. Browser compatibility matrix
3. Responsive design test cases
4. Dark mode comparison screenshots (Phase 2)

This comprehensive PRD provides a complete blueprint for building Insydr AI from concept to launch, with clear boundaries, success metrics, and a scalable design system. The document balances ambition with pragmatism, ensuring the college project is achievable while laying groundwork for future commercialization.