SuryaVani - Technical Documentation

Overview

A comprehensive Retrieval-Augmented Generation (RAG) chatbot system that enables intelligent querying across multiple document sources with advanced memory capabilities and multilingual podcast generation.

Architecture

Core Components

- **Backend Framework**: Python with SQLAlchemy ORM
- **Database**: JevoraDB (PostgreSQL-based with pgvector support)
- **Embedding Model**: OpenAI text-embedding-3-large (3072 dimensions)
- **LLM**: OpenAI GPT-4.1(query and Podcast script),GPT-4.1 mini(IMAGE),GPT-4o-mini(Memory summary),saarika V2.5(speech to text-youtube),saaras V2.5(STT for voice input) and Bulbul TTS(Podcast)
- **Hybrid Retrieval**: BM25 + Vector Search with RRF Fusion & FlashRank Reranking
- **Memory Layer**: Custom Vector-based Memory (Last 3 + Top 3 semantic search)
- **Quality Assurance**: RAGAS Metrics (Relevancy, Context Precision, Context Recall)
- **Podcast Generation**: Bulbul TTS for multilingual audio synthesis

Models & Services Overview

Component	Model/Technology	Type	Access Method
Main LLM	OpenAI GPT-4.1	Closed Source	API Key (OpenAI)
Image Analysis	OpenAI GPT-4.1 mini	Closed Source	API Key (OpenAI)
Memory Summary	OpenAI GPT-4o-mini	Closed Source	API Key (OpenAI)
Podcast Script	OpenAI GPT-4.1	Closed Source	API Key (OpenAI)
Embedding Model	OpenAI text-embedding-3-large	Closed Source	API Key (OpenAI)
STT (YouTube/Audio)	Sarvam AI Saarika v2.5	Closed Source	API Key (Sarvam AI)
STT (Voice Input)	Sarvam AI Saaras v2.5	Closed Source	API Key (Sarvam AI)
TTS (Podcast)	Bulbul TTS	Closed Source	API Key (Bulbul)
Web Scraping	Crawl4AI	Open Source	Free (No API Key)
Reranker	FlashRank	Open Source	Free (No API Key)
Keyword Search	BM25 (Okapi BM25)	Open Source	Free (Local)
Vector Database	PostgreSQL + pgvector	Open Source	Free (Self-hosted)
Browser Automation	Playwright	Open Source	Free (No API Key)
HTML Parsing	BeautifulSoup4	Open Source	Free (No API Key)
Audio Download	Youtube API Key	Open Source	Free (1000 calls per day)

Document Processing Pipeline

Supported Source Types

The system supports multiple input formats across various categories:

Documents

- **PDF** (.pdf) - Standard text-based PDFs with OCR support for scanned documents
- **Word Documents** (.docx) - Microsoft Word files
- **Text Files** (.txt, .md) - Plain text and Markdown files
- **Direct Text Pasting** - Copy-paste text directly into the interface

Images

- **Common Formats**: .png, .jpg, .jpeg, .webp, .gif, .bmp, .avif
- **Professional Formats**: .tif, .tiff, .heic, .heif, .ico, .jp2
- **OCR Analysis**: All images processed with GPT-4.1 mini for text extraction and content analysis

Spreadsheets

- **Excel Files**: .xlsx, .xls, .xlsm
- **CSV Files**: .csv - Comma-separated values

Audio/Video

- **Audio Formats**: .mp3, .wav, .m4a, .aac, .ogg, .flac, .wma, .opus, .aiff, .amr
- **Video Formats**: .mp4, .mov, .avi, .webm (audio extraction)
- **YouTube Links** - Direct URL support with youtube API KEY

Web Content

- **Web Pages** - URL-based content ingestion via Crawl4ai

Document Upload Flow

User Upload → Source Type Detection → Processing Pipeline → Chunking → Embedding → Storage

Processing Details by Type

Standard Documents (PDF/TXT/DOCX/MD)

- Direct text extraction
- Content chunking with configurable parameters
- Metadata preservation (page numbers, character positions)

Images (PNG/JPG/WEBP/HEIC/etc.)

- OCR analysis using OpenAI GPT-4.1 mini
- Text extraction from images
- Visual content understanding and description
- Support for 15+ image formats

Spreadsheets (Excel/CSV)

- Structured data extraction
- Table parsing and conversion to text
- Support for multiple sheets and formats

OCR PDF

- Image extraction using pdf2image library
- Visual content analysis via OpenAI GPT-4.1 mini API
- Text reconstruction from image analysis
- Chunking and embedding of extracted content

Web Content

- **Web Scraper**: Crawl4AI with AsyncPlaywrightCrawlerStrategy
- **Single URL Scraping**: Extract content from individual web pages
- **Recursive Crawling**: Automatically discover and crawl linked pages within same domain

- **Smart Link Filtering**: Excludes external domains, social media, and file downloads
- **Content Extraction**: Markdown-based clean content extraction
- **Metadata Preservation**: Title, description, keywords, language detection
- **Batch Processing**: Support for multiple URLs with rate limiting
- **Preview Mode**: Quick URL preview without full scraping
- **Natural Chunking**: Breaks at paragraph or sentence boundaries
- **URL Validation**: Ensures valid URL format before processing

Audio/Video Processing

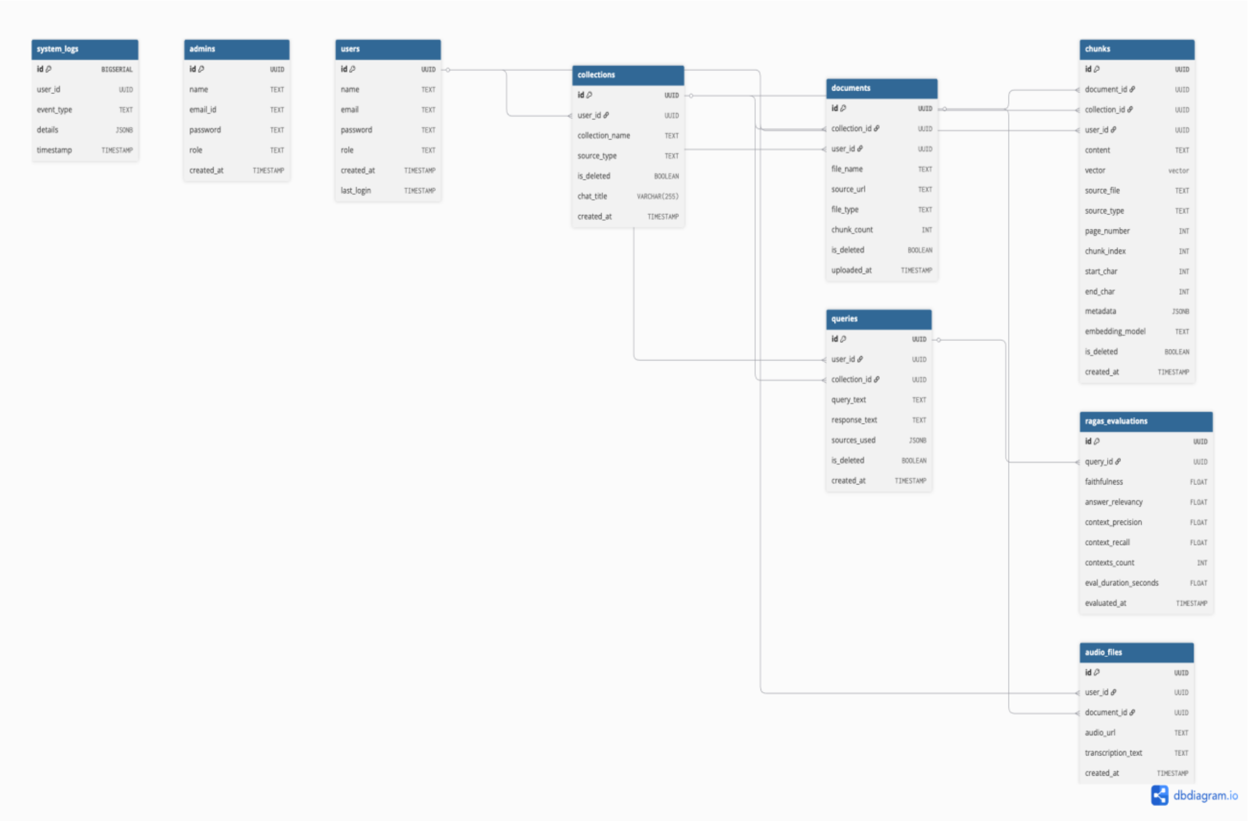
- YouTube link support via youtube API KEY that allows 1000 calls per day.
- Audio extraction from 15+ formats
- Speech-to-text transcription using Sarvam AI Saarika v2.5
- Transcription text chunked and embedded

Direct Text Input

- Paste text directly into the interface
- Immediate processing without file upload
- Supports markdown formatting

Data Storage Architecture

Database Schema



system_logs	
id	BIGSERIAL
user_id	UUID
event_type	TEXT
details	JSONB
timestamp	TIMESTAMP

admins	
id	UUID
name	TEXT
email_id	TEXT
password	TEXT
role	TEXT
created_at	TIMESTAMP

users	
id	UUID
name	TEXT
email	TEXT
password	TEXT
role	TEXT
created_at	TIMESTAMP
last_login	TIMESTAMP

collections	
id	UUID
user_id	UUID
collection_name	TEXT
source_type	TEXT
is_deleted	BOOLEAN
chat_title	VARCHAR(255)
created_at	TIMESTAMP

documents	
id	UUID
collection_id	UUID
user_id	UUID
file_name	TEXT
source_url	TEXT
file_type	TEXT
chunk_count	INT
is_deleted	BOOLEAN
uploaded_at	TIMESTAMP

queries	
id	UUID
user_id	UUID
collection_id	UUID
query_text	TEXT
response_text	TEXT
sources_used	JSONB
is_deleted	BOOLEAN
created_at	TIMESTAMP

chunks	
id	UUID
document_id	UUID
collection_id	UUID
user_id	UUID
content	TEXT
vector	vector
source_file	TEXT
source_type	TEXT
page_number	INT
chunk_index	INT
start_char	INT
end_char	INT
metadata	JSONB
embedding_model	TEXT
is_deleted	BOOLEAN
created_at	TIMESTAMP

ragas_evaluations	
id	UUID
query_id	UUID
faithfulness	FLOAT
answer_relevancy	FLOAT
context_precision	FLOAT
context_recall	FLOAT
contexts_count	INT
eval_duration_seconds	FLOAT
evaluated_at	TIMESTAMP

audio_files	
id	UUID
user_id	UUID
document_id	UUID
audio_url	TEXT
transcription_text	TEXT
created_at	TIMESTAMP

Query Processing

Conversation Configuration

Users can customize their chat experience with the following settings:

Response Length

- **Short** - Concise answers
- **Medium** - Balanced responses (default)
- **Long** - Detailed explanations

Conversation Style

- **Default** - Formal tone

- **Custom**

Response Languages

Support for 9 languages:

- English
- Hindi
- Tamil
- Telugu
- Kannada
- Malayalam
- French
- Spanish
- Arabic

Voice Input Support

- **Voice Queries**: Users can ask questions using voice input
- **Speech-to-Text**: Powered by Sarvam AI Saaras v2.5
- **Seamless Integration**: Voice transcribed and processed through standard RAG pipeline

RAG Pipeline

1. **Query Reception**: User submits natural language question (text or voice)
2. **Voice Transcription**: If voice input, convert to text using Saaras v2.5
3. **Hybrid Retrieval System**:
   a. **BM25 Keyword Search**: Traditional keyword-based search for exact matches
   b. **Vector Semantic Search**: OpenAI embeddings (3072d) for semantic similarity
   c. **Parallel Execution**: Both searches run simultaneously
   d. **RRF Fusion**: Results combined using Reciprocal Rank Fusion
      i. Formula: `(Semantic × 0.6) + (BM25 × 0.4)`
      ii. Semantic search weighted at 60%, BM25 at 40%
   e. **FlashRank Reranking**: Final ranking optimization of merged results
   f. **Performance**: +20-30% keyword accuracy, +12-18% overall accuracy

    g. **Fallback**: Gracefully degrades to semantic-only if BM25 fails

4. **Context Assembly**: Compile top-ranked chunks with source metadata
5. **Memory Integration**:
    a. Retrieve last 3 conversation turns for immediate context
    b. Perform vector search on conversation history to find top 3 relevant past interactions
    c. Inject both recent and semantically relevant history into prompt
6. **LLM Generation**: OpenAI GPT-4.1 generates response with:
    a. Retrieved document context
    b. Conversation history
    c. User's language and style preferences
7. **Follow-up Questions**: System generates 3-5 relevant follow-up questions
8. **RAGAS Evaluation**: Calculate quality metrics for response:
    a. **Answer Relevancy**: Verifies answer addresses the question
    b. **Context Precision**: Checks retrieved context relevance
    c. **Context Recall**: Ensures all necessary information retrieved
9. **Response Tracking**: Store query, response, sources, and RAGAS metrics
10. **Memory Update**: Store conversation summary with embeddings for future retrieval

Response Features

- **Source Attribution**: All responses cite specific documents and chunks
- **Follow-up Questions**: Automatically generated to guide conversation
- **Image Analysis**: If query references images, OCR and visual analysis included
- **Multilingual Output**: Responses in user's selected language

Memory Layer

The system maintains conversational context using a custom vector-based memory approach:

Memory Architecture

Storage Mechanism:

- Each query-response pair is summarized and stored in the database
- Summaries are embedded using OpenAI embeddings (3072d)
- Both summaries and full conversation history maintained

Context Retrieval:

- **Recent Context**: Last 3 conversation turns included in every query
- **Semantic Context**: Vector search retrieves top 3 most relevant past conversations
- Combined context provides both recency and relevance

Memory Flow:

1. User sends query
2. System retrieves:
    a. Last 3 conversation turns
    b. Top 3 semantically similar past interactions (vector search)
3. Both contexts injected into LLM prompt
4. After response generation:
    a. Conversation summary created
    b. Summary embedded and stored
    c. Full conversation history updated

Chat Sharing

Users can share their conversations in multiple formats:

Share as Link

- Generate unique shareable URL for conversation
- Public or private sharing options

- Preserved formatting and sources

Share as PDF

- Export entire conversation to PDF format via reportlab.
- Includes:
  - All messages with timestamps
  - Source citations and references
  - Follow-up questions
  - User preferences (language, style)
- Professional formatting for reports and documentation

Podcast Generation Feature

Overview

Transform collection content into engaging multilingual audio podcasts.

Workflow

1. **Collection Selection**: User selects a collection containing multiple sources
2. **Content Aggregation**: System retrieves all chunks from collection documents
3. **Script Generation**: OpenAI GPT-4.1 creates conversational podcast script
4. **Text-to-Speech**: Bulbul TTS engine converts script to audio
5. **Multilingual Support**: Generate podcasts in multiple languages
6. **Audio Delivery**: Synthesized audio file served to user

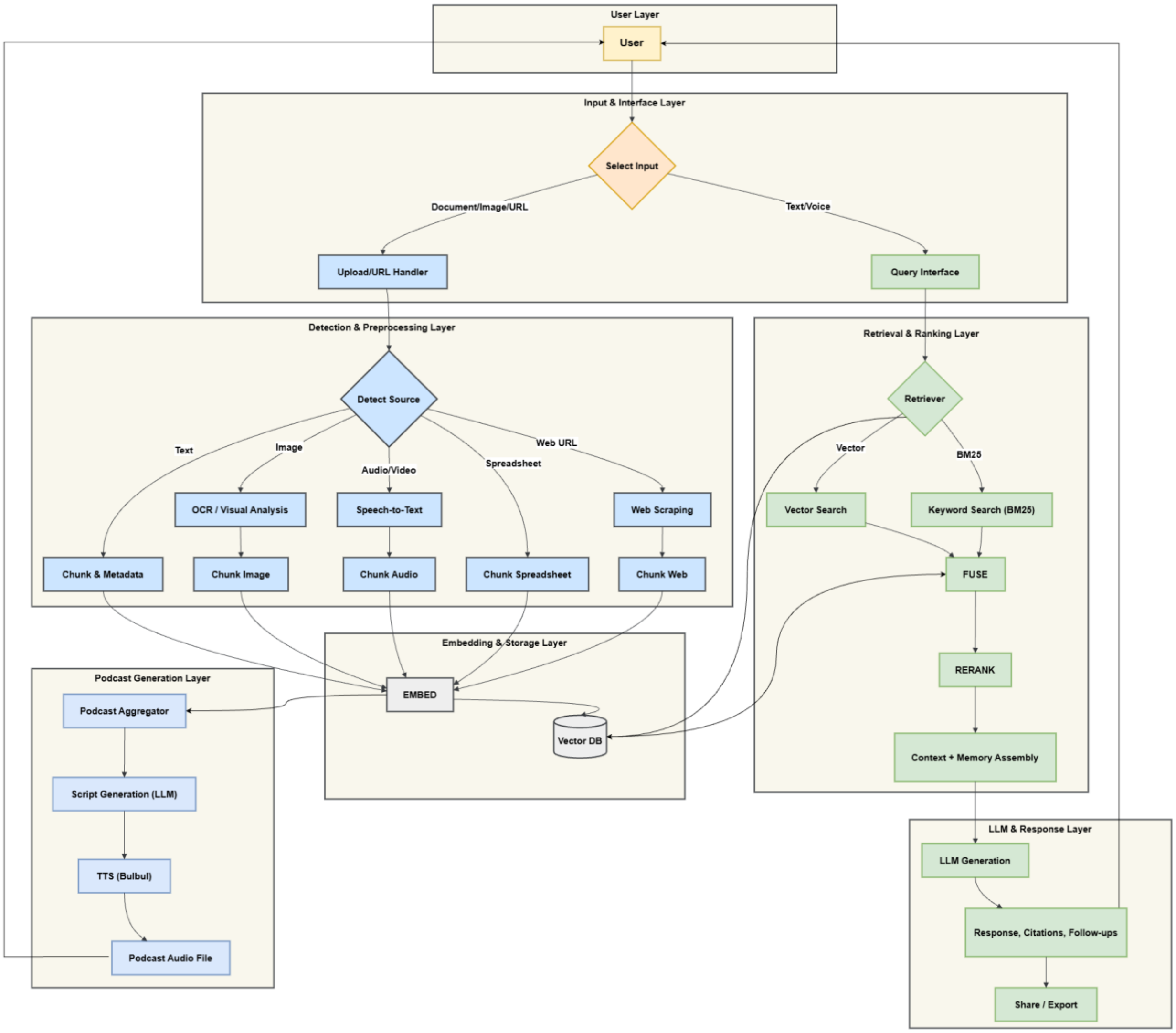
Technology Stack Summary

Component	Technology
Embedding Model	OpenAI text-embedding-3-large (3072d)
LLM	OpenAI GPT-4.1
Keyword Search	BM25 (Okapi BM25)
Hybrid Fusion	Reciprocal Rank Fusion (RRF)
Reranker	FlashRank
Quality Metrics	RAGAS (Faithfulness, Relevancy, Precision, Recall)
OCR Processing	OpenAI GPT-4.1 mini (PDFs & Images)
Image Analysis	OpenAI GPT-4.1 mini (15+ formats)
Web Scraping	Crawl4AI with AsyncPlaywrightCrawlerStrategy
Browser Automation	Playwright (headless)
HTML Parsing	BeautifulSoup4
Audio Download	Youtube API KEY
Speech-to-Text (Upload)	Sarvam AI Saarika v2.5
Speech-to-Text (Voice Query)	Sarvam AI Saaras v2.5
Text-to-Speech	Bulbul TTS
Memory Layer	Custom Vector-based Memory(gpt-4o mini)
Database	JevoraDB
ORM	SQLAlchemy
Vector Storage	pgvector extension (3072 dimensions)

Key Features

- **Extensive format support** - 40+ file formats including documents, images, spreadsheets, audio/video
- **Direct text pasting** - No file upload required for quick queries
- **Image analysis with OCR** - 15+ image formats with visual content understanding
- **Voice input** - Ask questions using voice via Saaras v2.5
- **Hybrid retrieval** - BM25 keyword + vector semantic search with RRF fusion
- **FlashRank reranking** - Final optimization for maximum relevance
- **RAGAS quality metrics** - Real-time evaluation of every response (Relevancy, Precision, Recall)
- **Advanced web scraping** - Single URL or recursive crawling with Crawl4AI
- **Intelligent link discovery** - Automatic same-domain crawling with smart filtering
- **Configurable responses** - Custom length, style, and language preferences
- **Multilingual support** - 9 languages for responses
- **Intelligent chunking** with metadata preservation and natural boundaries
- **Vector similarity search** using OpenAI embeddings
- **Dual-layer memory** - Recent context (last 3 turns) + semantic search (top 3 relevant)
- **Follow-up questions** - Auto-generated to guide conversation
- **Source attribution** in responses
- **Collection-based organization** for multi-document contexts
- **Chat sharing** - Export as link or PDF
- **Multilingual podcast generation** from document collections
- **YouTube video transcription** support
- **Excel/CSV processing** for structured data
- **Batch URL processing** with rate limiting

ARCHITECTURE DIAGRAM



User Layer
User
Input & Interface Layer
Select Input
Document/Image/URL
Text/Voice
Upload/URL Handler
Query Interface
Detection & Preprocessing Layer
Detect Source
Text
Image
Audio/Video
Spreadsheet
Web URL
OCR / Visual Analysis
Speech-to-Text
Web Scraping
Chunk & Metadata
Chunk Image
Chunk Audio
Chunk Spreadsheet
Chunk Web
Retrieval & Ranking Layer
Retriever
Vector
BM25
Vector Search
Keyword Search (BM25)
FUSE
RERANK
Context + Memory Assembly
Embedding & Storage Layer
EMBED
Vector DB
Podcast Generation Layer
Podcast Aggregator
Script Generation (LLM)
TTS (Bulbul)
Podcast Audio File
LLM & Response Layer
LLM Generation
Response, Citations, Follow-ups
Share / Export