

SAKSHI AI: Technical Change Log

Generated: January 3, 2026

Comparison: `Sakshi_updated_Jan2.zip` (Before) vs Current Deployment (After)

Table of Contents

1. [Executive Summary](#)
2. [Modified Files](#)
3. [chat.py - AI Logic & Vector Search](#)
4. [SakshiUnifiedWidget.tsx - Frontend Chatbot](#)
5. [Preloader.tsx - Loading Animation](#)
6. [meetings.py - Re-vectorization Hook](#)
7. [New Files Added](#)
8. [app/services/ Directory](#)

Executive Summary

Category	Changes Made
Backend AI	Vector search (FAISS), Speaker-aware prompts, White-labeled errors
Frontend	Markdown rendering, Custom branding, Debounced @mention search
UX	Multi-stage loading animation
Infrastructure	New <code>services/</code> module with vector store engine

Modified Files

1. `chat.py` - Backend AI Logic

Path: `SMC_MOM_BACKEND/app/routers/chat.py`

Change 1: Department-Based Access Control

Motivation: Non-admin users should only see meetings from their department.

```
-async def search_meetings_in_db(query: str, limit: int = 10) -> list:
+async def search_meetings_in_db(query: str, limit: int = 10, user_department: str = None, user_role: str = 'admin') -> list:
+
+    # Build department filter for non-admin users
+    base_filter = {}
+    if user_role != 'admin' and user_department:
+        base_filter = {
+            '$or': [
+                {'department': user_department},
+                {'departments': user_department},
+                {'created_by': user_department}
+            ]
+        }
+
+    return await search_meetings(query, limit, base_filter)
```

Change 2: Semantic Vector Search Integration

Motivation: Solve token limit errors by retrieving only relevant transcript chunks instead of the full text.

```
-        # Create a friendly, context-aware prompt
+        #
+        # USE SEMANTIC SEARCH FOR CONTEXT (FAISS VECTOR STORE)
+        # Falls back to truncation if vector search unavailable
+        #
+        try:
+            from ..services.vector_store import get_meeting_context, FAISS_AVAILABLE
+
+            if FAISS_AVAILABLE:
+                relevant_context = get_meeting_context(
+                    query=request_obj.question,
+                    meeting_id=request_obj.meeting_id,
+                    departments=departments,
+                    max_chars=4500
+                )
+                transcription_context = relevant_context
+            except Exception as vec_err:
+                transcription_context = context.get('transcription', '')[:4500]
```

Change 3: Speaker-Aware AI Prompt

Motivation: AI was using names mentioned in speech content instead of diarization labels.

BEFORE:

```
Instructions:  
- Answer in a friendly, warm, and conversational tone  
- Base your answer ONLY on the meeting transcription...
```

AFTER:

```
IMPORTANT INSTRUCTIONS:  
1. **Speaker Attribution**: ALWAYS mention who said what.  
   - Example: "Officer Kumar raised the budget concern..."  
  
7. **⚠ CRITICAL - Speaker Identification Rules**:  
   THE SPEAKER NAME IS ONLY WHAT APPEARS **BEFORE THE COLON**.  
  
✗ WRONG: If transcript says "Speaker_0: My name is Dr. Sachin Omwase..."  
   - Do NOT say "Dr. Sachin Omwase said..."  
  
✓ CORRECT: Say "***Speaker_0** said his name is Dr. Sachin Omwase..."  
  
NEVER extract personal names from what people say in their speech.
```

2. `SakshiUnifiedWidget.tsx` - Frontend Chatbot

Path: `frontend/src/components/SakshiUnifiedWidget.tsx`

Change 1: Markdown Renderer Function (NEW)

Motivation: AI responses showed raw `**bold**` text instead of formatted output.

```
+// =  
+// MARKDOWN RENDERER - Renders formatted AI responses  
+// =  
+  
+const renderMarkdown = (text: string): JSX.Element[] => {  
+  // Parse **bold**, *italic*, headers, lists  
+  // ... ~100 lines of parsing logic  
+};
```

Usage Change:

```
-<p className="text-sm whitespace-pre-wrap">{msg.content}</p>  
+{msg.role === 'assistant'  
+ ? <div className="prose prose-sm">{renderMarkdown(msg.content)}</div>  
+ : <p className="text-sm whitespace-pre-wrap">{msg.content}</p>  
+}
```

Change 2: Debounced @Mention Search

Motivation: Typing @ triggered excessive API calls; now debounced to 800ms.

```
-  const loadRecentMeetings = async () => {  
+  const fetchMeetingSuggestions = async (query: string = '') => {  
  
+    // Debounce the search (800ms)  
+    mentionDebounceRef.current = setTimeout(() => {  
+      fetchMeetingSuggestions(afterAt);  
+    }, afterAt ? 800 : 0);
```

Change 3: Custom Brand Logos

Motivation: Replace generic Lucide icons with SAKSHI brand assets.

```
-<MessageSquare className="w-8 h-8 text-white" />  
+  
  
-<MessageSquare className="w-8 h-8 text-indigo-600" />  
+
```

3. `Preloader.tsx` - Loading Animation

Path: `frontend/src/components/Preloader.tsx`

Change: Multi-Stage Loading Logic

Motivation: Single random pause felt artificial; new flow simulates real initialization phases.

BEFORE (Old Logic):

```
if (currentProgress < 65) {  
    currentProgress += 8; // Fast to 65%  
} else if (currentProgress >= 65 && currentProgress < 85 && !isPaused) {  
    // Random pause between 65-85%  
    const pauseDuration = Math.random() * 3000 + 2000;  
}
```

AFTER (New Logic):

```
// Stage 1: 0% to 20% (Slow initialization)  
if (currentProgress < 20) {  
    currentProgress += 1;  
}  
  
// Stage 1.5: Brief pause at 20%  
else if (currentProgress === 20 && !hasPausedAt20) {  
    setTimeout(() => startLoop(), 500);  
}  
  
// Stage 2: 20% to random point (Fast processing)  
else if (currentProgress < pausePoint) {  
    currentProgress += 6;  
}  
  
// Stage 3: Random pause (Finalizing)  
else if (currentProgress === pausePoint) {  
    const pauseDuration = Math.random() * 3000 + 2000;  
    setTimeout(() => finishAnimation(), pauseDuration);  
}
```

4. meetings.py - Re-vectorization Hook

Path: SMC_MOM_BACKEND/app/routers/meetings.py

Change: Auto Re-index on Speaker Name Edit

Motivation: When user edits speaker names in UI, vector store must update to reflect new names.

```
+     # _____  
+     # RE-VECTORIZE MEETING WITH UPDATED SPEAKER NAMES  
+     # _____  
+     try:  
+         from ..services.vector_store import vectorize_meeting, remove_meeting_from_index  
+  
+         if transcript:  
+             # Remove old vectors  
+             remove_meeting_from_index(meeting_id, departments)  
+  
+             # Re-vectorize with updated speaker names  
+             vectorize_meeting(  
+                 meeting_id=meeting_id,  
+                 transcription=transcript,  
+                 departments=departments,  
+                 meeting_title=meeting_title,  
+                 meeting_date=meeting_date  
+             )  
+             logger.info(f"✅ Re-vectorized meeting {meeting_id}")  
+         except Exception as vec_error:  
+             logger.warning(f"⚠️ Re-vectorization failed: {vec_error}")
```

New Files Added

app/services/ Directory

Status: Entirely new directory (did not exist in old version)

File	Size	Purpose
__init__.py	18 bytes	Module initialization
vector_store.py	22,849 bytes	FAISS vector store engine

vector_store.py Key Functions:

Function	Purpose
embed_text()	Generate embeddings using sentence-transformers
vectorize_meeting()	Index a meeting's transcript into FAISS
remove_meeting_from_index()	Remove meeting vectors (for re-indexing)

Function	Purpose
<code>get_meeting_context()</code>	Semantic search for relevant transcript chunks
<code>search_vectors()</code>	Low-level FAISS similarity search

Why Added: Enables semantic search (RAG) to find relevant transcript sections instead of dumping entire transcripts into AI context, solving token limit crashes for long meetings.

Files Summary

File	Status	Lines Changed
<code>chat.py</code>	Modified	+150 lines
<code>SakshiUnifiedWidget.tsx</code>	Modified	+180 lines
<code>Preloader.tsx</code>	Modified	+40 lines
<code>meetings.py</code>	Modified	+40 lines
<code>services/__init__.py</code>	NEW	+1 line
<code>services/vector_store.py</code>	NEW	+450 lines

Document generated from actual `diff -u` command outputs.