BotFlow Knowledge Base Test Document

This is a test document for the BotFlow PDF knowledge base system.

It contains actual text content (not images) so pdf-parse can extract the text successfully.

Key Features of BotFlow:

- AI-powered WhatsApp automation
- Template-based bot creation
- Knowledge base with vector search
- RAG (Retrieval-Augmented Generation)
- Multi-tenant SaaS platform

This paragraph contains information about South African businesses. BotFlow is designed specifically for South African SMEs who want to automate their WhatsApp customer service. The platform supports multiple languages, handles load shedding mentions, and includes local payment integrations like Paystack.

Technical Architecture

Backend Stack:

- Fastify API server with TypeScript
- Supabase PostgreSQL with pgvector
- OpenAI GPT-4 and text-embedding-3-small
- BullMQ message queue with Redis

Knowledge Base Features:

The knowledge base uses vector embeddings to enable semantic search. When a PDF is uploaded, it is chunked into 2000-character segments with 200-character overlap. Each chunk is then embedded using OpenAI text-embedding-3-small model, creating 1536-dimensional vectors. These vectors are stored in PostgreSQL using the pgvector extension for efficient similarity search.

RAG Implementation:

During conversations, the bot retrieves relevant knowledge chunks using cosine similarity search. The top matching chunks are included in the AI context, allowing the bot to answer questions based on the uploaded documents. This retrieval-augmented generation approach ensures accurate, source-based responses.

Testing Guide

To test PDF processing:

- Upload a PDF via /api/bots/:botId/knowledge
- The backend downloads the PDF from Supabase Storage
- pdf-parse extracts the text content
- Text is chunked into overlapping segments
- OpenAI generates embeddings for each chunk
- Embeddings are stored in knowledge_embeddings table
- Article status updates to "indexed"

Expected Results:

This 3-page document should generate approximately 2-3 chunks, depending on the exact character count. The first chunk will contain the introduction and features list. The second chunk will cover the technical architecture. The third chunk will include the testing guide. Each chunk overlaps with adjacent chunks to maintain context continuity.

Success Criteria:

' PDF text extracted successfully
' Multiple chunks created with overlap
' Embeddings generated for all chunks
' Searchable via semantic similarity
' Article marked as indexed