Enterprise AI Assistant - Project Overview

1. Project Introduction

The Enterprise AI Assistant is a next-generation, platform-managed application designed to provide intelligent document retrieval (RAG), data analytics, and strategic recommendations without relying on complex external backend infrastructure or third-party API keys (like OpenAI). It runs entirely within the client's environment, ensuring data privacy and zero latency.

2. Key Features

- Platform-Managed RAG: Built-in document intelligence that indexes PDFs directly in the browser using PDF.js and performs semantic retrieval without external vector databases.
- Agentic Intent Routing: Automatically detects user intent (Document Query, Data Insight, or Action Recommendation) and routes the request to the appropriate specialized agent.
- Multi-Query Processing: Capable of handling complex, compound questions (e.g., 'What is the leave policy? AND Analyze attrition trends') by splitting them into sub-tasks.
- Data Privacy: All data processing happens client-side. Uploaded documents and data never leave the user's session.
- Serverless Architecture: Removed dependency on Python/FastAPI backend, simplifying deployment and maintenance.

3. System Architecture Flow

The system follows a modular agentic flow:

- Input Layer: User provides text input or uploads files (PDFs, CSVs).
- Ingestion Engine: Files are parsed immediately. PDFs are chunked into text segments; CSVs are converted to structured JSON.
- Router Agent: Analyzes the user's query using keyword mapping and pattern recognition to determine the Intent.
- Execution Engines:

  - Knowledge Engine: Performs similarity search on text chunks (TF-IDF/Overlap).
  - Analytics Engine: Executes statistical operations on structured data.
  - Advice Engine: Generates business rules-based recommendations.
- Response Synthesizer: Aggregates results (especially for multi-part queries) and presents them in a unified chat interface.

4. Technical Stack

- Frontend: Next.js (React Framework) with Tailwind CSS for styling.
- Data Processing: PDF.js (PDF parsing), PapaParse (CSV parsing).
- State Management: React Hooks for in-memory vector storage and session history.
- Environment: Node.js / Browser Runtime.

5. Usage Guide

- 1. Launch: Run 'npm run dev' to start the application.
- 2. Upload: Use the sidebar to upload HR Policies (PDF) or Employee Data (CSV). Multiple files supported.
- 3. Interact: Ask questions like 'What is the probation period?' or 'Analyze sales trends'.

- 4. Review: interpretation of documents and data is instant.