Project Title:

DocuQuery: AI-Powered PDF Knowledge Assistant Using Google PALM

Phase-1: Brainstorming & Ideation

Objective:

DoDocuQuery is an AI-powered PDF knowledge assistant that utilizes Google PALM to efficiently extract, analyze, and summarize information from PDF documents. It helps users quickly find relevant insights with precision and ease.

Key Points:

1. **Problem Statement:**
    - Managing and extracting relevant information from large volumes of PDF documents is time – consuming and inefficient.
    - Traditional search methods if often failed to provide precise answers , requiring users to manual shift through extensive text.
2. **Proposed Solution:**
    - **DoDocuQuery is an AI-powered PDF knowledge assistant that leverages Google PALM to intelligently process and analyze PDF documents. It enables users to quickly extract relevant insights, summarize content, and retrieve precise answers using natural language queries. By automating information retrieval and summarization, DoDocuQuery enhances productivity, reduces manual effort, and improves decision-making efficiency.**
3. **Target Users:**
    - **Business & Corporate Users**
      Executives & Managers – Quickly extract insights from business reports, financial statements, and strategic documents.
    - **Legal & Compliance Teams – Analyze contracts, policies, and regulatory documents.**
    - **Market Analysts & Consultants – Summarize industry trends from reports and whitepapers.**

4. **Expected Outcome:**
    - Efficient Information Retrieval: Users can quickly find relevant data within PDFs using AI-powered search.
    - Enhanced Productivity: Reduces manual effort in reading and analyzing large documents.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the do **AI-Powered PDF Knowledge Assistant Using Google PALM**

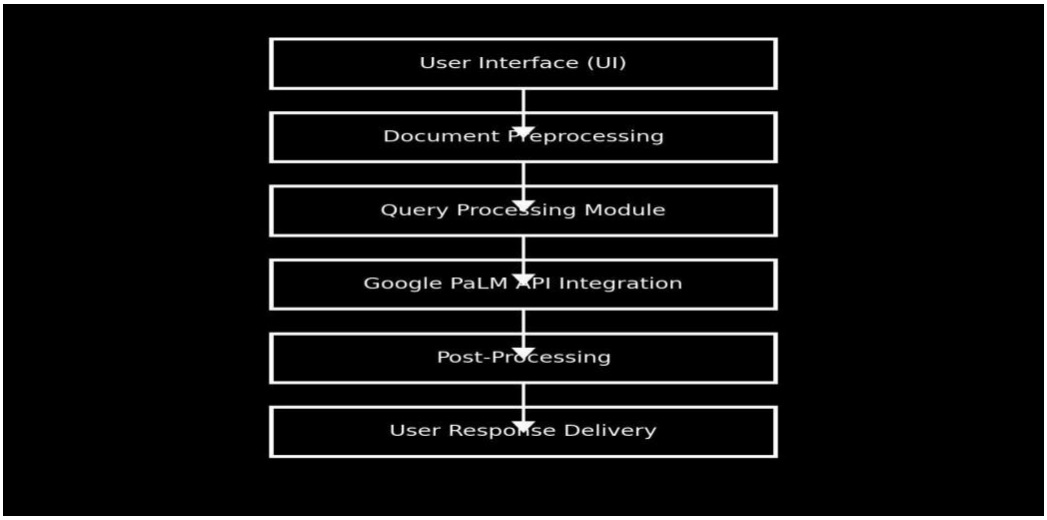
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: **FastAPI /flask/Django for API and services**
    - Frontend: **Angular/React.js/Vue.js**
    - Database: **primary, cache, search engine**
2. **Functional Requirements:**
    - User management
    - PDF processing and text extraction
    - AI Powered query knowledge Retrieval
    - Data export and sharing
3. **Constraints & Challenges:**
    - Dependence on Google PALM API
    - Storage and processing limitations
    - Multilingual Support
    - Data Privacy Regulations

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - User interface
    - Application Layer
    - AI module layer
    - Data storage and Indexing
    - Security and Authentication layer
    - Integration & API Layer
    - Cloud Infrastructure & Development
2. **User Flow:**
    - Step-1: Uploading the document
    - Step-2: Preprocessing and Indexing
    - Step-3: User query and input
    - Step-4: AI powered query processing
    - Step-5: Output Generation
    - Step-6: Query interaction and refinement
    - Step-7: Exporting & sharing
3. **UI/UX Considerations:**
    - Onboard & First time user experience
    - Clean & Minimalist
    - Smooth Query experience

Phase-4: Project Development

Objective:

Implement core features of the AutoSage App.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Google Gemini Flash API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Gemini API integration**.
    - Develop **vehicle comparison and maintenance tips logic**.
    - Optimize **search queries for performance and relevance**.
3. **Challenges & Fixes:**
    - **Challenge:** Delayed API response times.
      Fix: Implement **caching** to store frequently queried results.
    - **Challenge:** Limited API calls per minute.
      Fix: Optimize queries to fetch **only necessary data**.