Project Brief: General Doc Summarisation AI

Project Title: General Doc Summarisation AI

Project Overview:

The **General Document Summarisation AI** is a versatile, user-friendly application designed to help individuals and professionals quickly digest **long-form documents**. Whether it's research papers, reports, legal contracts, or technical manuals, this tool uses advanced AI models to generate **clear, concise summaries** that capture the core ideas and key points.

Built for general use, the app offers a clean interface where users can upload documents in various formats and receive structured summaries in seconds. It’s ideal for students, researchers, analysts, and anyone who regularly works with dense or time-consuming content.

Objectives:

1. Create a **general-purpose AI summarisation tool** for long documents across industries and use cases.
2. Support **multi-format document uploads**, including PDF, DOCX, and TXT.
3. Use **state-of-the-art AI models** to generate accurate, readable summaries.
4. Provide **customisation options** such as summary length, tone (formal/informal), and focus (e.g. executive summary vs. technical overview).
5. Enable **copy and export functionality** for easy sharing and integration into workflows.
6. Ensure the system is **fast, scalable, and intuitive**, with minimal setup required.

Key Features:

1. **Multi-Format Uploads:** Accepts PDFs, Word documents, and plain text files.
2. **AI-Powered Summarisation:** Uses large language models to extract and condense key information.
3. **Custom Summary Settings:** Users can choose summary length, tone, and focus area.
4. **Real-Time Preview:** See the summary instantly before exporting.
5. **Copy & Export Options:** Summaries can be copied to clipboard or downloaded as .txt or .pdf.
6. **Clean, Accessible UI:** Designed for general users with minimal technical knowledge.
7. **Scalable Backend:** Handles large documents efficiently with background processing.
8. **Optional Summary Comparison:** View side-by-side summaries with different settings for comparison.

Technical Specifications:

1. **Programming Language:** Python
2. **Frameworks and Libraries:**

- Flask/Django: Web application framework.
- PyPDF2, python-docx: For document parsing
- Gemini or OpenAI APIs: For summarisation
- NLP libraries (e.g., SpaCy or NLTK): for language understanding and intent recognition.

3. **Database:** SQLlite for storing user preferences and summary logs (optional).
4. **Deployment:** Docker containerization and cloud service deployment (e.g., AWS or Azure).
5. **Version Control:** Git for source code management.

Expected Outcomes:

1. A robust and intelligent **AI-powered calendar assistant** suitable for professional use in executive support roles.
2. Increased **efficiency and responsiveness** in scheduling workflows.
3. Seamless **integration with existing tools**, reducing cognitive load and manual time-management effort.
4. A scalable platform capable of expanding to support teams and organisational needs.

Risks and Mitigations:

1. **Data Privacy:** Implement strong encryption, access control, and audit logging to safeguard calendar and email data.
2. **Integration Challenges:** Perform thorough testing of email and calendar API connections across platforms.
3. **Language Ambiguity:** Use advanced NLP models and customizable parsing rules to handle vague or varied scheduling instructions.
4. **Performance & Sync Reliability:** Optimise calendar syncing and ensure real-time responsiveness with efficient background workers and retry logic