Text for Your PDF

Project Plan: Personal Context-Aware AI Assistant

1. Overview

This document outlines the development plan for a Personal Context-Aware AI Assistant. The primary goal of this project is to create an intelligent agent that goes beyond standard chatbots by leveraging a user's personal context to provide truly personalized and proactive assistance. The system will be designed to learn from user-provided documents, notes, and other data sources to answer questions, summarize information, and recommend actions. The initial version will focus on text-based context, with future iterations incorporating multimodal capabilities.

2. Core Components

The assistant will be built on a foundation of three key technologies:

• Retrieval-Augmented Generation (RAG): This is the core of the system. Instead of relying solely on its pre-trained knowledge, the AI will first retrieve relevant information from a user-specific vector database containing their personal notes and documents. This retrieved context will be provided to a Large Language Model (LLM) to generate accurate, grounded answers. This approach ensures responses are personalized and based on the user's own data.

• Vector Database: We will use a local vector database, such as FAISS or ChromaDB, to store the numerical representations (embeddings) of the user's documents. This enables efficient and fast semantic search to find the most relevant context for any given query.

• Agentic Behavior: The assistant will be designed not just to answer questions, but to take simple actions. The initial action set will include summarizing long documents and drafting emails based on provided notes and a user's prompt.

3. Future Goals

The long-term vision is to expand the assistant's capabilities significantly. Key future goals include:

• Multimodal Integration: Enhance the system to understand context from images (e.g., diagrams, photos of whiteboards) and audio notes.

• Proactive Assistance: Develop logic for the assistant to identify potential needs and offer suggestions without being prompted, such as creating a to-do list from meeting notes.

• Knowledge Graph: Implement a system to build a knowledge graph from the user's data, allowing for more complex reasoning about relationships between different pieces of information.