**POC Document: AI Workspace Assistant**

**1. ✅ Project Title**

**AI Workspace Assistant**  
A Unified Smart Assistant for Document Intelligence, Task Planning, and Knowledge Search

**2. 🎯 Objective**

To build a powerful, LLM-driven assistant that allows users to:

* Upload internal documents and query them in natural language
* Extract summaries, action items, and FAQ-style answers
* Plan high-level goals into actionable tasks
* Retain memory of past interactions to improve over time

This assistant leverages **Multi-Component Pipelines (MCP), RAG, Agents, and Vector Memory** for a deeply intelligent and extensible AI tool.

**3. 🔍 Key Features**

| **Feature** | **Description** |
| --- | --- |
| 📄 **Multi-File Upload** | Upload PDFs, CSVs, plain text, or URLs |
| 💬 **Natural Language Q&A** | Ask questions, get cited answers from uploaded content |
| 📑 **Summarization & Action Items** | Extract summaries and decisions from meeting notes/docs |
| 🧠 **Goal Planner** | Input high-level goals → generate subtasks using agents |
| 🧭 **Smart File Routing Agent** | LLM decides how to handle PDFs vs CSVs vs links |
| 🗂️ **Memory** | Store past conversations and preferences in vector memory |
| 📈 **Feedback Logging** | Track and improve responses via user feedback |

**4. 🏗️ Architecture Overview**

┌────────────────────────────┐

│ User Input │

└────────────┬───────────────┘

↓

[ Multi-Component Agent ]

↓

┌──────┬────────┴──────────┬──────────┐

↓ ↓ ↓

[Task Planner] [RAG Search Engine] [Summarizer]

↓ ↓ ↓

Subtasks Relevant Chunks Summary/Actions

↓

┌────────[Vector DB (Chroma)]────────┐

│ Stores embeddings for docs │

└────────────────────────────────────┘

**5. ⚙️ Technology Stack**

| **Layer** | **Tools** |
| --- | --- |
| 🧠 LLMs | OpenAI (GPT-4) or Gemini Pro |
| 🔗 Orchestration | LangChain (Agents, Chains, Memory) |
| 🧰 Backend | FastAPI |
| 💾 Vector Store | ChromaDB |
| 📚 Loaders | LangChain Document Loaders (PDF, CSV, Web, TXT) |
|  |  |

**6. 🔧 Endpoints (API-first MVP)**

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /upload | POST | Upload and embed files |
| /query | POST | Ask a question, get a contextual answer |
| /plan | POST | Submit a goal, get subtasks |
| /feedback | POST | Save user feedback for training |
| /memory | GET/POST | Access/update user memory |

**7. 📌 Implementation Phases**

| **Phase** | **Features** |
| --- | --- |
| **Phase 1** | File Upload, RAG Search, Basic Q&A |
| **Phase 2** | Summarization, Action Item Extraction |
| **Phase 3** | Goal Planner Agent (Task Breakdown) |
| **Phase 4** | Smart File Type Router (PDF, CSV, etc.) |
| **Phase 5** | Feedback Logger + Memory Store |
|  |  |

**8. 💡 Use Cases**

* Internal knowledge retrieval from policies, product docs
* Meeting minutes summarization with follow-up tasks
* Legal document clause Q&A
* Business goal planning with task breakdown
* CSV reports auto-analysis and question answering

**9. 🏆 Value Proposition**

| **Stakeholder** | **Value** |
| --- | --- |
| 📈 Managers | Task breakdown from goals |
| 🧑‍💼 HR | Query policies/contracts |
| 📄 Legal | Contract clause search |
| 📊 Analysts | Query CSV-based dashboards |
| 🤖 Dev teams | Use as internal AI doc tool or API for other apps |

**10. ✅ Future Scope**

* Fine-tuned models for legal/HR/marketing domains
* Web UI with conversational memory
* Multi-user authentication and history

**11. 📂 Deliverables**

* Full FastAPI backend
* LangChain-powered intelligent chains/agents
* Sample uploaded docs and question-answer demos
* Final presentation + walkthrough demo