Updated Initial Implementation Plan
AI-Driven Document Compliance Analysis System

Project Overview

This project aims to develop a Streamlit-based AI-powered application that allows businesses to upload internal documents, select a compliance standard (e.g., ISO 9001, ISO 27001, GDPR), and receive a detailed compliance report. The report will:

- Identify what aligns with the selected standard
- Highlight gaps with actionable recommendations

To ensure output quality, a **multi-agent architecture** will be implemented, where a team of AI agents generates the initial report, and another team reviews it before final output using an automated **feedback loop**.

Milestones & Timeline

Milestone 1: Frontend Development (Streamlit UI)

Timeline: Day 1 – Day 5

Scope:

- Develop the user interface using Streamlit
- File upload component supporting PDF, DOCX, TXT
- Dropdown menu for compliance standard selection
- Button to trigger analysis
- Loading indicators and result display area

- Basic input validation

Deliverables:

- Fully functional Streamlit-based UI
- Input form with validation and standard selection
- Result panel for displaying compliance insights
- GitHub repository with frontend code

Milestone 2: AI Agents Development & Feedback Loop Integration

Timeline: Day 6 – Day 19 *(includes 5-day extension)*

Scope:

- Design and implement two agent teams:
    - **Report Generation Agents**: Analyze content and generate compliance insights
    - **Review Agents**: Independently verify the generated report
- Establish a **feedback loop**: If inconsistencies are found, report is sent back to generation agents for revision
- Prompt templates aligned with ISO/GDPR requirements
- Modular architecture using **LangChain** or **CrewAI**

Deliverables:

- Modular AI agent system (generation + review teams)
- Feedback loop logic implemented and tested
- Prompt templates for selected standards

- Evaluation mechanism for agents' review quality
- Logs for agent decisions and actions
- GitHub repository with all agent logic and configuration

Milestone 3: Backend Integration

Timeline: Day 20 – Day 24

Scope:

- Document preprocessing: parsing, text extraction, and chunking
- Connect frontend to AI agent backend
- Generate structured, exportable reports (HTML/PDF)
- In-memory file handling for privacy

Deliverables:

- Preprocessing pipeline for uploaded files
- End-to-end integration (upload → analyze → report)
- Downloadable compliance report (PDF or HTML)
- Complete backend scripts in GitHub repository

Milestone 4: Deployment & Testing

Timeline: Day 25 – Day 30 *(updated project duration)*

Scope:

- Deploy app on **Streamlit Community Cloud**
- Perform usability, performance, and functional testing
- Final QA and adjustments based on test feedback
- Deliver technical documentation and usage guide

Deliverables:

- Live deployed app with public access URL
- QA test logs (including edge case tests)
- Final project documentation
- User onboarding guide or instructions
- Deployment link and repository access

---

3. Tools, Frameworks, and APIs

Component	Tools / Libraries
UI	Streamlit
Document Parsing	`pdfminer.six`, `python-docx`, `PyMuPDF`
LLM Integration	OpenAI GPT-4 API
Agent Framework	LangChain or CrewAI
Report Generation	HTML templates, `pdfkit` or `reportlab`
Deployment	Streamlit Community Cloud

---

7. Success Criteria

- **Frontend Usability**: Smooth file upload, clean interface, intuitive standard selection
- **AI Accuracy**: ≥90% alignment to real compliance content (validated by manual reviews)
- **Processing Speed**: ≤90 seconds for a document of up to 10 pages
- **User Feedback**: ≥4.5/5 average rating on report clarity and usefulness
- **Deployment Uptime**: ≥99% availability during test period