Anna AI Coach – Brief Write-Up

1. Architecture Overview

Anna AI Coach is a modular, web-based AI assistant designed to support first-time entrepreneurs. Its architecture consists of:

- **Frontend:** Responsive chat interface with:
  - Conversation window
  - Quick suggestion buttons
  - "New Conversation" reset functionality
- **Backend/API:** Flask handles HTTP requests (/chat, /reset, /health) and manages:
  - Input processing
  - Response generation via the AI engine
  - Interaction logging
- **AI Engine:** OpenAI GPT-4 integration:
  - Generates contextual responses using conversation history and knowledge base content
  - Guided by prompt engineering to provide goal-oriented advice
- **Knowledge Base:** Markdown file (knowledge_base.md) with curated entrepreneurship content.
- **Logging System:** Records timestamped interactions with reasoning for transparency.

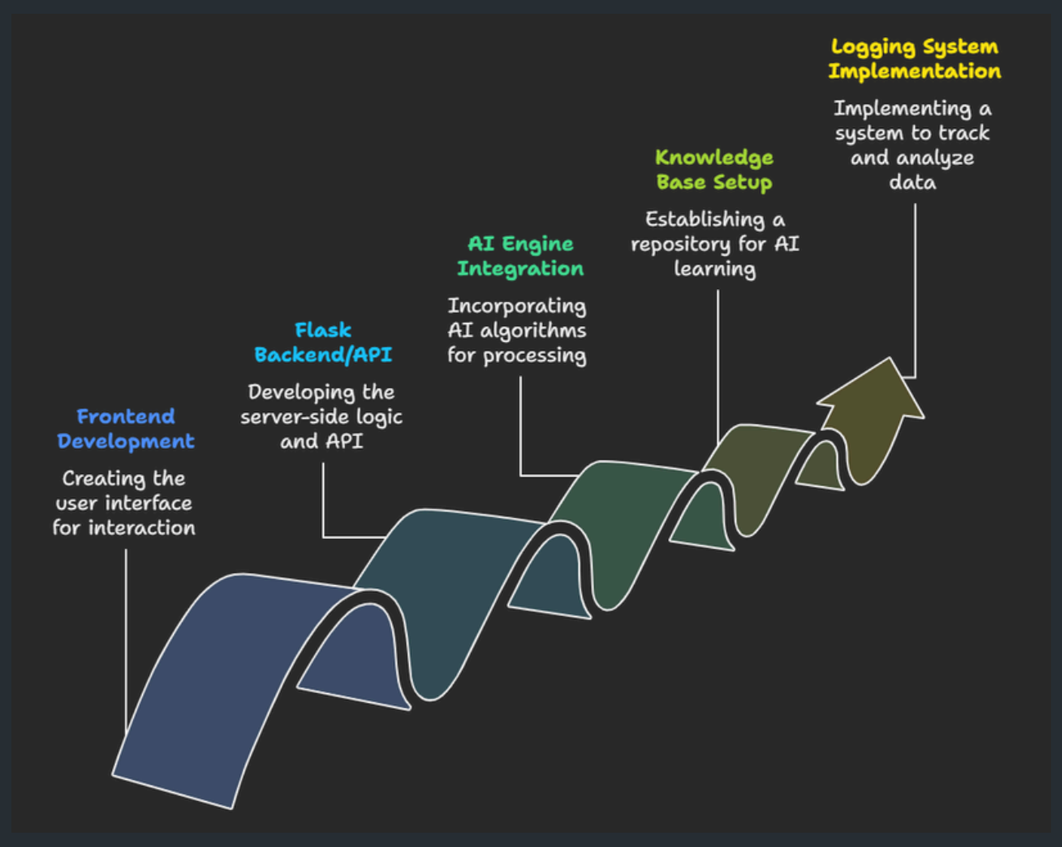

Fig 1: Data Flow Through System Components

2. Design Decisions and Trade-Offs

- **Flask:** Lightweight, easy to deploy. Trade-off: Limited scalability for high-concurrency users.
- **Single Markdown Knowledge Base:** Easy maintenance. Trade-off: Cannot support large-scale dynamic queries without embedding/vector DB.
- **GPT-4o AI Engine:** High-quality, contextual responses. Trade-off: Requires API access and incurs latency/cost.
- **Chat History Context:** Retains last 3 exchanges to balance context vs. prompt length. Trade-off: Long conversations may lose earlier context.
- **Render Deployment (Free Tier):** Quick public availability. Trade-off: Idle spin-down can delay first request after inactivity.

Decision	Pros	Cons
Flask	Lightweight, simple	Limited concurrency
Markdown KB	Easy updates	Limited retrieval
GPT-4o	Context-aware advice	API cost/latency
Chat History	Context retention	Limited to last 3 exchanges
Render Free Tier	Publicly accessible	Idle spin-down delays

Table 1: Pros vs Cons

3. Ideas for Improvement

- **Personalization:** Track user progress and tailor advice per stage or industry.
- **Enhanced Retrieval:** Use embeddings/vector DB for more nuanced answers.
- **Multimedia Integration:** Charts, templates, or voice-based interaction.
- **Scalability:** Move backend to FastAPI or cloud scalable infrastructure.
- **External API Integrations:** CRM, financial tools, market research APIs.
- **Analytics Dashboard:** Display progress metrics, conversation summaries, and recommended actions.