**Project Title:** Domain-Specific AI Assistant

**Project Overview:**

This project aims to develop a domain-specific AI assistant capable of understanding and responding to natural language queries using a combination of modern NLP tools and frameworks. The assistant will be tailored to a particular field (e.g., Law, Medical, Resume-building) and will leverage semantic search, conversational memory, and user project tracking. It will be designed for deployment as a web application, featuring a FastAPI backend and a Next.js frontend, with optional voice interaction and Figma-based UI design.

**Core Technologies:**

* **LangChain:** For managing the conversational flow, memory, and LLM integration.
* **ChromaDB:** To store and retrieve vector embeddings for semantic search.
* **FastAPI:** To build RESTful backend APIs.
* **Next.js:** For creating a dynamic frontend interface.
* **Figma:** Used to design the UI layout (later to be implemented in the frontend).
* **Framer (Optional):** Final integration into a visual website if CMS compatibility is available.
* **Ollama (Mistral-7B):** For running a local instruction-tuned language model (~4.1 GB) capable of chat and reasoning directly on the developer’s machine.
* **HuggingFace Embeddings:** For computing text embeddings without paid APIs.

**Key Features:**

1. **Domain-Specific Query Understanding:**
   * Chatbot will understand and respond accurately to user queries within a selected domain.
2. **Semantic Search Integration:**
   * Uses ChromaDB and embeddings to search across documents, FAQs, or knowledge bases.
3. **Conversational Memory:**
   * Tracks user history and context using LangChain memory modules.
4. **Project Knowledge:**
   * Capable of storing, retrieving, and referring to ongoing user projects, specifically those related to “Arun Jayesh.”
5. **Frontend Interface:**
   * Responsive, clean chat UI with project display dashboard.
   * Optional voice input integration.
6. **File and Reference Logging:**
   * All development actions and logic updates will be recorded in a file called refer.txt to reduce confusion and prevent hallucinations.
7. **Dual-Function Chat Intelligence:**
   * The assistant can handle both project-specific questions (e.g., “What is Arun Jayesh working on?”) and general domain queries (e.g., “How does LangChain manage memory?” or “Explain indemnity in insurance”).
8. **Local Model Deployment:**
   * The assistant uses a local LLM (via Ollama) that runs on the developer’s machine.
   * The current model is **Mistral-7B**, an instruction-tuned model (~4.1 GB), ideal for reasoning and lightweight coding tasks.
   * This ensures cost-free operation but limits public web access unless the local backend is deployed to a server.

**Deployment Note:**

* For users to access the chatbot via a public website (e.g., through Framer), the FastAPI backend and the local LLM must be deployed to a server (e.g., cloud VM or VPS) that can run Ollama and expose the API.
* Without deployment, the chatbot will only be accessible from the local machine where it is hosted.

**Public Access Strategy (Cost-Free):**

* ✅ **Start** with ngrok to temporarily expose the local FastAPI server and connect it to a button/link on the Framer website.
* ✅ **Later**, migrate the backend (FastAPI + chatbot logic) to a free-tier platform such as **Render** or **Railway**.
* ✅ Use a **free hosted model** from HuggingFace Inference API to replace Ollama in production for consistent uptime and low-resource hosting.

**Optional Extensions:**

* Voice-to-text using Web Speech API
* Embeddable widget for Framer-based websites
* Project dashboard for real-time updates

**Target Outcome:**

A production-ready AI chatbot assistant tailored for a specific domain, capable of real-time semantic search and conversation, with integration to track and reference user projects. The assistant will be able to answer both project-related and general domain-specific user queries in a natural, contextual way. The system will be designed to be expandable, modular, and user-friendly, and can later be hosted on any cloud platform or embedded into other systems.