Here’s a well-structured **Product Requirements Document (PRD)** for your **Career Guidance Chatbot** project:

# **📄 Product Requirements Document (PRD)**

**Product Name:** Career Guidance Chatbot  
 **Prepared By:** Abdus Samad  
 **Date:** 30-Aug-2025  
 **Version:** 1.0

## **1. Objective**

The **Career Guidance Chatbot** aims to provide **personalized, context-aware career guidance** for students and professionals. It leverages **LangChain**, **Groq API**, **Gemini embedding model**, **FAISS**, **MongoDB**, and a **Streamlit UI** to deliver meaningful conversations. The chatbot maintains context from the last 15 user-AI exchanges, compresses them into a **300-token summary**, and generates responses tailored to user goals and queries.

## **2. Key Features**

### **2.1 Core Functionality**

* **User Query Handling:**
  + Accepts text-based queries via a Streamlit UI.
* **Context Management:**
  + Stores last 15 user-AI exchanges (30 messages total).
  + Summarizes conversation history using a lightweight model (≤300 tokens).
  + Passes summary + latest query to reasoning model for coherent responses.
* **Response Generation:**
  + Provides **context-aware**, **detailed career advice**.
* **History Persistence:**

MongoDB stores session history as tuples:  
  
 {

"session\_id": "<UUID>",

"chat\_history": [

{"user\_query": "...", "ai\_response": "..."},

...

]

}

### **2.2 Database & Storage**

* **MongoDB**:  
  + Stores user sessions & conversation history.
  + Enables retrieving past conversations for continuity.
* **FAISS Vector Database**:  
  + Stores embeddings of past queries/responses.
  + Uses **Gemini embeddings** for semantic retrieval & better context relevance.

### **2.3 Models & APIs**

* **Embedding Model:**
  + **Gemini Embeddings** for converting text into vectors for FAISS.
* **Summarization Model:**
  + **Lightweight Groq API model** for summarizing last 15 chats into ≤300 tokens.
* **Reasoning Model:**
  + **Advanced Groq API model** for generating detailed, context-rich career guidance responses.

### **2.4 User Interface (UI)**

* **Technology:** Streamlit
* **Features:**
  + Chat window with real-time conversational flow.
  + Option to start **new/existing sessions**.
  + Display AI responses in structured, readable format.
  + Minimalist, professional theme for **accessibility & ease of use**.
* **Future Enhancements:**
  + Dark mode option.
  + Export chat history (PDF/Doc).
  + Voice input & output.

## **3. User Stories**

* **As a student**, I want to ask career-related questions and receive guidance tailored to my interests.
* **As a job seeker**, I want the chatbot to suggest skills and certifications relevant to my career path.
* **As a professional**, I want to explore transition opportunities and upskilling paths.
* **As a user**, I want my previous session history to be available when I return.

## **4. System Workflow**

1. **User inputs a query** in the Streamlit chat window.
2. **Chat history retrieval**: MongoDB fetches last 15 exchanges (if available).
3. **Summarization**: Groq lightweight model compresses them to ≤300 tokens.
4. **Embedding & Retrieval**: Gemini embeddings + FAISS fetch semantically relevant past context.
5. **Reasoning & Response**: Strong Groq model generates final response using summarized + retrieved context.
6. **Storage**: MongoDB updates session with (user\_query, ai\_response) pair.
7. **Display**: Response appears in the Streamlit UI.

## **5. Technical Stack**

* **Frontend/UI:** Streamlit
* **Backend/Logic:** LangChain + Groq API
* **Database:** MongoDB (history), FAISS (embeddings)
* **Models:** Gemini (embeddings), Groq (summarization + reasoning)
* **Hosting:** (To be decided – could use Streamlit Cloud, Render, or local deployment)

## **6. Success Metrics**

* **Accuracy:** % of responses rated as relevant and useful by users.
* **Retention:** % of users who return for multiple sessions.
* **Latency:** Average response time (target < 3s per query).
* **Engagement:** Average number of queries per session.

## **7. Risks & Mitigation**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| High API cost due to frequent calls | Medium | Use summarization model to reduce tokens passed to reasoning model |
| Poor response quality | High | Fine-tune prompt templates & chain structure |
| MongoDB scaling issues | Medium | Optimize schema, add indexing |
| Data privacy concerns | High | Encrypt session IDs & anonymize data |

## **8. Future Enhancements**

* **Multi-language support** for career advice.
* **Voice-based interaction** (speech-to-text + text-to-speech).
* **Analytics Dashboard** for career trends (skills in demand, certifications, etc.).
* **Integration with LinkedIn/Job APIs** for real-time job suggestions.

✅ That’s your refined PRD for the Career Guidance Chatbot.

Do you also want me to **turn this into a visual architecture diagram** (showing data flow between UI → LangChain → Groq → FAISS/MongoDB)?