

Alexa-Powered Personal Learning Assistant (RAG Tutor)

1. Problem Statement

Traditional virtual assistants like Amazon Alexa can answer general knowledge questions but fail to provide contextually grounded and curriculum-specific learning experiences. This project bridges that gap by transforming Alexa into an AI-powered tutor capable of delivering accurate, contextual, and personalized explanations using Retrieval-Augmented Generation (RAG) and a custom educational dataset.

2. Objective

To develop a Retrieval-Augmented Alexa Skill that:

- 1.Retrieves factual answers from a custom knowledge base.
2. Generates clear explanations using an integrated LLM (Llama 3.1 via Groq API).
3. Suggests logical follow-up topics to create a continuous learning journey.

3. Dataset Overview

File	Title	Core Concepts Covered
01	AI Introduction & History	Definition, history, goals, and applications of AI.
02	Data Preprocessing	Cleaning, normalization, feature scaling, and encoding.
03	Supervised Learning	Regression, classification, bias-variance tradeoff.
04	Unsupervised Learning	Clustering,PCA,dimensionality reduction.
05	Evaluation Metrics	Accuracy, precision, recall, F1 score, confusion matrix.
06	Neural Network Basics	Perceptron, activation functions, backpropagation.
07	DeepLearning Architectures	CNNs, RNNs, LSTMs, GRUs.
08	NLP & Word Embeddings	Word2Vec,GloVe,embeddings, attention mechanism.
09	Transformers & LLMs	Self-attention, GPT, BERT models.

10	Generative AI & RAG	Generative models, retrieval-augmented generation.
----	---------------------	--

A curated corpus of 10 educational text modules was used to train the knowledge base.

4. System Architecture

User → Alexa Skill → Flask Webhook (app.py) → RAG Engine (rag_engine.py) → [ChromaDB Retrieval + LLM (Groq API)] → Response + Follow-up → Alexa Output

5. Implementation Phases

Phase 1 – Knowledge Base Creation: Loaded 10 text files, chunked using LangChain, generated embeddings, and stored in ChromaDB.

Phase 2 – RAG Engine Development: Developed rag_engine.py to retrieve relevant data and generate responses with follow-up.

Phase 3 – Alexa Skill Integration: Created Flask webhook, integrated ASK SDK, exposed via ngrok, and connected to Alexa Developer Console.

Phase 4 – Testing: Verified multi-turn flow with Alexa Developer Console test tab.

6. Libraries and Dependencies

Flask, ask-sdk-core, ask-sdk-webservice-support, LangChain, ChromaDB, Sentence-Transformers, Groq API, dotenv, ngrok

7. Conclusion

The Alexa-Powered RAG Tutor successfully demonstrates how voice AI can be turned into a structured, intelligent learning assistant. By integrating Retrieval-Augmented Generation, vector search, and Alexa Skills Kit, the system enables accurate, context-aware, and guided learning experiences.