📘 Prototype Documentation – Multilingual RAG Chatbot (SIH 2025)

1. 🎯 Project Overview

We are building a **multilingual conversational chatbot** for students that:

- Understands queries in **multiple languages** (English, Hindi, Gujarati).
- Uses **Rasa** for intent classification and dialogue management.
- Uses **XLM-R (XLM-RoBERTa)** for multilingual embeddings.
- Uses **LangChain + RAG (Retrieval-Augmented Generation)** for document-based Q&A.
- Returns answers in the **same language as input** (via translation).

This is the **prototype version** — backend-first, minimal but working.

---

2. 🛠️ Tech Stack

- **Rasa** → Intent classification & dialogue management.
- **XLM-R** → Multilingual embeddings for understanding queries.
- **LangChain** → Orchestrates the RAG pipeline.
- **FAISS** → Vector database for fast similarity search.
- **FastAPI/Flask** → REST API to expose chatbot backend to frontend.
- **IndicTrans2 / HuggingFace NLLB** → Translation layer for multilingual answers.

---

3. ⚡ System Architecture (Prototype)

Flow:

1. **User Query (Input)**
   - Text in English/Hindi/Gujarati.
2. **Language Understanding**
   - Query passed to **XLM-R** → generates multilingual embeddings.
   - Rasa uses these embeddings for **intent classification**.
3. **Conversation Manager (Rasa)**
   - Detects intent: FAQ, timetable query, deadline query, fallback.
   - Routes query:
     - FAQ → Predefined responses.
     - Document-based → Send to RAG pipeline.
4. **RAG Pipeline (LangChain + FAISS)**
   - Document loader → split circulars/FAQs into chunks.
   - Store embeddings in FAISS DB.
   - On query: retrieve top-k matching chunks.
   - Return relevant text.
5. **Response Generation**
   - Basic text template or retrieved chunk.
   - If input language ≠ English → translate output back to input language.
6. **Output**
   - Sent back via API → frontend (Web UI / WhatsApp).

---

4. 📌 Implementation Steps

Step 1 – Setup Rasa

- Install Rasa.
- Create intents:
  - greeting, faq, timetable_query, deadline_query, document_query, fallback.
- Add training data (sample queries in English, Hindi, Gujarati).
- Configure Rasa pipeline to use **custom embedding featurizer (XLM-R)**.

Step 2 – Integrate XLM-R

- Use HuggingFace `xlm-roberta-base`.
- Add custom Rasa component that:
  - Takes text input.
  - Generates embeddings using XLM-R.
  - Passes embeddings to Rasa classifier.

Step 3 – Setup LangChain + FAISS

- Load 2–3 sample PDFs (circulars, timetable).
- Use LangChain `DocumentLoader` → split into chunks.
- Convert chunks → embeddings (XLM-R).
- Store in FAISS vector DB.
- Build retrieval function: `query → FAISS → top-k results`.

Step 4 – Connect Rasa & LangChain

- If intent = "document_query":
  - Call LangChain pipeline.
  - Return top result chunk to Rasa.

Step 5 – Translation Layer

- Detect query language (e.g., `langdetect`).
- If non-English:
  - Translate retrieved response into input language (IndicTrans2/NLLB).

Step 6 – Expose API

- Wrap entire flow in FastAPI endpoint:
  - `/chat` → Input: `{ query: "..." }` → Output: `{ response: "..." }`.
- Frontend (web or mobile) calls this API.

---

5. ✅ Prototype Demo Plan

- Upload 2 PDFs:
  1. Exam Timetable.
  2. Academic Circular.
- Test cases:
  1. Input (English): *"Show me the exam timetable"*.
     → Bot retrieves timetable from PDF.

   2. Input (Hindi): “मेरी परीक्षा का टाइमटेबल दिखाओ।”
      → Bot retrieves same timetable, replies in Hindi.
   3. Input (Gujarati): “મારું એક્ઝામ ટાઈમટેબલ બતાવો.”
      → Bot replies in Gujarati.

---

6. 🚀 Deliverables for SIH PPT

1. **Block Diagram** (Pipeline: Input → XLM-R → Rasa → LangChain → FAISS → Response).
2. **Working Prototype**:
   - Multilingual text query.
   - Retrieval from documents.
   - Reply in same language.
3. **Screenshots**: console logs + frontend chat.
4. **Future Scope** (voice input, WhatsApp/Telegram integration, analytics dashboard).

---

7. 🔮 Future Enhancements (Beyond Prototype)

- Add **speech-to-text** + **text-to-speech**.
- Add **dashboard** for admin (upload new docs, monitor usage).
- Expand to **20+ Indic languages**.
- Integrate with **college ERP** for live data (attendance, marks).
- Add **analytics & insights** for student behavior.