GEN AI ENGINEER ASSIGNMENT
FUNNEL DROP CHATBOT

Problem Context

In a fintech product, users often **drop off** during high-friction journeys such as:

- KYC completion
- PAN/Aadhaar verification
- Loan onboarding
- Merchant registration
- Credit limit enhancement
- Payment flow (UPI/AutoPay/Mandate)

Your task is to build a **GenAI-powered Funnel Drop Chatbot** that:

1. Detects *where* the user dropped off.
2. Understands *why* they may have dropped off.
3. Provides *intelligent nudges* and *issue-resolution guidance*.
4. Answers questions conversationally with **LLM + RAG (Retrieval Augmented Generation)**.
5. Generates personalized advice, based on user state + product rules + funnel logic.

This assignment assesses your ability to design **LLM pipelines, retrieval systems, personalization logic, prompt engineering, fine-tuning strategies, caching, guardrails, and end-to-end system thinking**.

High-Level Objective

Build a **production-grade Funnel Drop Chatbot** that:

- **Takes a user's funnel journey state**
- **Diagnoses likely reasons for their drop-off**
- **Answers their queries naturally**
- **Provides nudges to get them back into the flow**
- **Is grounded in a document describing the full funnel (provided as sample)**
- **Avoids hallucinating and must cite sources**

Inputs to the Chatbot

1. User State Object (JSON)

Example:

- `{`
- `  "user_id": "A937272",`
- `  "stage_dropped": "KYC_verification_pending",`
- `  "timestamp": "2025-03-14T10:22:00Z",`
- `  "error_code_history": ["OCR_FAIL", "PAN_NAME_MISMATCH"],`
- `  "device": "Android",`
- `  "language": "en"`
- `}`

2. Funnel Drop Document (PDF or Markdown)

A detailed description of:

- All funnel stages

- What each stage requires

- Common drop-off reasons
- Error codes
- Eligibility rules
- Compliance text
- Example nudges
- Troubleshooting steps

(We have shared the document with you.)

3. User Query

Example:

- *“Why was my KYC rejected?”*
- *“How can I complete my registration?”*
- *“What went wrong earlier?”*

Expected Outputs

Structured JSON Response

- `{`
- `  "predicted_drop_reason": "PAN name mismatch during OCR validation",`
- `  "explanation": "The name extracted from your PAN card did not match the name on your profile.",`
- `  "nudge_message": "Upload a clearer PAN card image or update your profile name to match your PAN.",`
- `  "steps_to_fix": [`
- `    "Hold the PAN card against a contrasting background",`
- `    "Ensure lighting is even",`
- `    "Update profile name if spelling differs"`
- `  ],`

- `  "confidence_score": 0.86,`
- `  "citations": ["funnel_doc_section_3.1", "error_codes_table"]`
- `}`

Conversational Message

“Looks like your KYC couldn’t be verified because the PAN name didn’t match your profile details. This is easy to fix, try re-uploading a clearer image or edit your profile name to match what’s on your PAN.”

Mandatory Requirements

1. RAG Pipeline

You must build a retrieval system to ground all answers:

Must include:

- Semantic chunking of the Funnel Drop Document
- Metadata (stage type, error code, FAQ, troubleshooting)
- Vector embeddings + keyword fallback
- Reranking layer
- Source-citation in final answers

Must prevent hallucination:

- If RAG confidence is low → chatbot must say:
  “I’m not fully sure — can you provide more details?”

2. Drop-Off Reasoning Engine

Develop logic that predicts **why the user dropped** based on:

- Stage they dropped in
- Recent errors
- Device type
- Time of day
- Common funnel rules from the PDF
- LLM reasoning using chain-of-thought (hidden, not exposed)

This module should return:

- Primary reason
- Secondary probable reasons
- Confidence score

3. Nudge Generation Engine

LLM must generate:

- **Personalized nudges**
- **Multi-lingual messages (English + Hindi)**
- **Context-aware variants (short CTA, long reassurance message, compliance-safe version)**
- **Emotionally sensitive text (avoid blame, maintain trust)**

You must provide:

- 3 versions of nudges: explanatory, CTA-focused, and empathetic.

4. Chatbot Orchestration Layer

You must architect a pipeline that contains:

- **Intent classifier LLM**
- **RAG retriever**
- **Drop-off reasoning module**
- **Nudge generator**
- **Final response synthesizer LLM**
- **Guardrail layer** (no financial advice, compliance-friendly text)

Orchestration must be modular so that:

- Any LLM (OpenAI, Claude, Llama) can be swapped
- Any vector DB can be swapped

5. Fine-Tuning or LoRA Adapter

Fine-tune a small model on:

- Funnel troubleshooting examples
- Error code explanations
- Nudge generation tasks
- Multilingual outputs

You should describe:

- Training dataset design
- Hyperparameters
- Evaluation metrics
- Before/after comparison

Actual training is optional but **design is mandatory**.

6. Evaluation Framework

You must measure:

- **RAG relevance (recall@5)**
- **Nudge helpfulness score (LLM-as-a-judge)**
- **Hallucination rate**
- **JSON correctness**
- **Latency & token cost per request**

Provide a small evaluation report.

7. Technical Deliverables

Code

- Clean, modular Python or Node.js code
- API endpoints:
    - `/predict_reason`
    - `/nudge_user`
    - `/chat`

Artifacts

- Vector store folder
- Model weights (if fine-tuned)
- Sample test cases

Documentation

- Architecture diagram
- Data flow
- Design decisions
- Tradeoffs
- Limitations & future extensions

Sample User Query → Expected Chatbot Behavior

User Query:

"Why did I get stuck while uploading my PAN card?"

Bot Behavior:

1. Detect user stage → KYC OCR
2. Retrieve relevant funnel doc chunks
3. Identify likely root cause → glare or angle error
4. Provide human-friendly explanation
5. Provide actionable steps
6. Offer a retry CTA link (dummy)

Submission Requirements

Your submission must include:

- GitHub repository
- Instructions to run
- PDF describing architecture, fine-tuning approach, and evaluation