

## Task Assignment: RAG-based Chatbot with Complaint Creation API

### Objective:

Develop a Retrieval-Augmented Generation (RAG) based chatbot that handles customer complaints by creating complaint records via API calls, collecting follow-up details (name, phone number, email, complaint details), generating a unique complaint ID, and retrieving complaint details when queried with the complaint ID.

### Requirements:

#### 1. RAG-based Chatbot

- **Purpose:** Provide contextual responses based on a knowledge base (e.g., customer service guidelines, FAQs) and handle complaint-related conversations.
- **Functionality:**
  - Engage users in natural language to collect complaint details.
  - Store and retrieve relevant documents for RAG.
  - Maintain conversation context to ask follow-up questions (name, phone number, email, complaint details) if not provided initially.
- **Knowledge Base:**
  - Create a sample knowledge base (e.g., pdf, text files or a small database) with customer service policies, complaint handling procedures, or FAQs. You can use any pdf for demo.

#### 2. Complaint Creation API

- **Purpose:** Create a RESTful API to handle complaint creation and store details in a database.
- **Endpoints:**
  - **POST /complaints:** Create a new complaint.
    - **Input (JSON):**

```
{
  "name": "string",
  "phone_number": "string",
  "email": "string",
  "complaint_details": "string"
}
```
    - **Output (JSON):**

```
{
  "complaint_id": "unique_string",
  "message": "Complaint created successfully"
}
```
  - **GET /complaints/{complaint\_id}:** Retrieve complaint details by complaint ID.

- **Output (JSON):**
- {
- "complaint\_id": "unique\_string",
- "name": "string",
- "phone\_number": "string",
- "email": "string",
- "complaint\_details": "string",
- "Created At": "timestamp"
- }

- **Implementation:**
  - Store complaints in a database.
  - Generate a unique complaint ID for each complaint.
  - Ensure proper input validation (e.g., valid email, phone number format).

### 3. Chatbot Integration with API

- **Complaint Creation:**
  - Collect all required fields (name, phone number, email, complaint details).
  - Prompt for missing fields with follow-up questions.
  - Create the complaint via the POST /complaints endpoint.
  - Respond with the complaint ID and confirmation message.
- **Complaint Retrieval:**
  - Extract complaint ID from user query.
  - Fetch details via the GET /complaints/{complaint\_id} endpoint.
  - Display details in a formatted response.

### 4. Example Interaction:

**User:** I want to file a complaint about a delayed delivery.

**Chatbot:** I'm sorry to hear about the delay. Please provide your name.

**User:** Deepak

**Chatbot:** Thank you, Deepak. What is your phone number?

**User:** 1234567890

**Chatbot:** Got it. Please provide your email address.

**User:** deepak@example.com

**Chatbot:** Thanks. Can you share more details about the delayed delivery?

**User:** My order #12345 was supposed to arrive on May 1 but is still not here.

**Chatbot:** Your complaint has been registered with ID: XYZ123. You'll hear back soon.

**User:** Show details for complaint XYZ123.

**Chatbot:** Complaint ID: XYZ123

Name: deepak

Phone: 1234567890

Email: deepak@example.com

Details: My order #12345 was supposed to arrive on May 1 but is still not here.  
Created At: 2025-05-08 10:00:00

### **Submission**

- Submit code in a GitHub repository.
- Include a live demo link or video walkthrough.