**Methodology for QnA Bot Development Using OCR-Extracted Data**

## 1. Introduction

This document outlines the methodology that will be followed to develop a **Question Answering (QnA) system** using **OCR-extracted text**. The system will consist of two models:

1. **An Open-Source QnA Model** that will retrieve and answer queries from structured OCR data.
2. **A Google Gemini-Based QnA Model** that will process entire PDFs for direct question answering.

The objective is to compare their efficiency, accuracy, and response quality to determine the most effective approach for document-based QnA systems, particularly for large documents.

---

## 2. Data Processing and Preparation

### 2.1 OCR Extraction and Preprocessing

- Text, tables, and images will be extracted from PDFs using **Mistral AI's OCR API**.
- The response will be saved in **JSON format**, containing structured text and metadata.
- **Data cleaning and filtering** will be applied to remove OCR noise, redundant content, and formatting artifacts.

### 2.2 Text Structuring and Chunking

- The extracted text will be **segmented into meaningful chunks (500-1000 tokens each)** to enable efficient retrieval.
- For large documents, **retrieval-augmented generation (RAG) techniques** will be used to manage content effectively.
- Tables will be **properly aligned and converted into structured formats** for better indexing.
- Each chunk will be stored with **metadata (page number, section headers, and keywords)** to improve search accuracy.

---

## 3. QnA Model Development

### 3.1 Open-Source QnA Model

- **Model Selection:**
  - Multiple open-source models will be evaluated, including **Mistral 7B, Phi-2, and Deepseek R1**.
  - The best-performing model will be selected based on **benchmark accuracy, retrieval efficiency, and scalability**.
- **Implementation:**
  - The structured text will be embedded using **Sentence Transformers**.
  - The embeddings will be stored in **FAISS or ChromaDB** for efficient similarity search.
  - A **retrieval-augmented generation (RAG) pipeline** will be implemented to handle large documents:
    1. The user query will be **vectorized and searched** within the indexed database.
    2. The top-ranked text chunks will be retrieved as context.
    3. The **open-source model** will generate an answer based on the retrieved context.

### 3.2 Google Gemini QnA Model

- The **Google Gemini API** will be used to **directly process entire PDFs**.
- User queries will be sent along with **full PDF input** for context-aware responses.
- The answers will be extracted without requiring prior text segmentation or embedding.

---

## 4. Model Comparison and Evaluation

### 4.1 Evaluation Criteria

Both models will be compared based on:

- **Answer Accuracy** – The correctness and relevance of generated answers.
- **Response Time** – The time taken to generate answers.
- **Context Awareness** – The model's ability to understand document structure for better answers.
- **Scalability and Cost** – The efficiency in handling large documents and API cost implications.

### 4.2 Performance Metrics

- **F1-score, Precision, and Recall** will be used to measure answer relevance.
- **Latency in milliseconds** will be recorded for each model.
- **Human evaluation** will be conducted to assess qualitative answer correctness.