

## 5. Technical Architecture

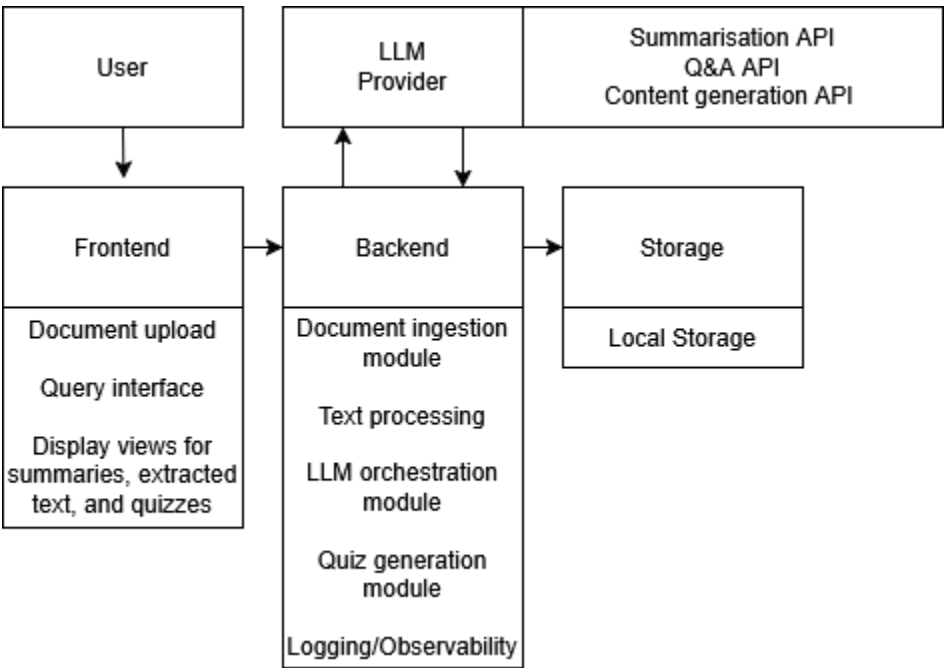
This section describes the end-to-end architecture of **ReadStudyAssist AI**, including system components, data flow, layer separation, and integration strategy. The architecture is designed to support three core capabilities: **Text Summarization**, **Document Q&A**, and **Content Generation for quizzes**.

### 5.1 Architecture Overview

ReadStudyAssist AI uses a modular four-layer architecture:

1. **User Interface Layer** – Web application where learners upload documents, ask questions, and review generated summaries/quizzes.
2. **Service Layer** – Backend API responsible for requesting routing, document-processing pipelines, and communication with AI components.
3. **AI/LLM Layer** – Handles summarization, Q&A and quiz generation through prompt-based interactions and function calling.
4. **Data Layer** – Responsible for storing raw files, processed text chunks and metadata.

### 5.2 System Architecture Diagram



## 5.3 Layer Structure

### (1) User Interface Layer

#### Responsibilities:

- Collect user inputs (upload PDFs, ask questions)
- Present summarized content and generated quizzes

### (2) Service Layer

This layer orchestrates all business logic.

#### Modules:

##### *a. Document Ingestion Module*

- Accepts PDF uploads
- Converts files into plain text
- Splits into semantic chunks for processing and retrieval

##### *b. AI Request Orchestrator*

- Builds structured prompts for summarization, Q&A, and quiz generation
- Handles function calling (e.g., to trigger quiz generation)
- Manages retries/errors

##### *c. Output Formatting Module*

- Formats summaries, Q&A responses, and quizzes into UI-friendly formats

### (3) AI / LLM Layer

The system relies on a single LLM provider - Gemini, accessed through:

- Summarization prompt templates (for full or chapter-level summaries)
- Context-aware Q&A prompts (LLM + retrieved chunks)
- Quiz generation prompts (MCQs, fill-in-the-blank, short answer)

### (4) Data Layer

Includes:

##### *a. Document Storage*

- Stores raw PDFs and extracted text

- Metadata: upload timestamps, source type, etc.

#### ***b. Logs and Observability***

- Captures LLM call metadata
- Stores user interactions for debugging, evaluation, and usage patterns

### **5.4 Data Flow Diagram**

