7. Feature Specification

This section outlines all features planned for ReadStudyAssist AI. Since development has not yet started, all features are currently in the planning phase. Features are grouped into **Implemented**, **Planned**, and **Out of Scope**, as required.

7.1 Implemented Features (as of December 7)

None.
At this stage, the team has not yet implemented any functional components of the application. All features described below remain in the design and planning phase.

7.2 Planned Features (to be completed by December 22)

The following features represent the core functionality required to deliver the MVP version of ReadStudyAssist AI. Each feature includes its description, priority level, and expected implementation complexity.

1. Document Upload & Ingestion

Description:
Allow users to upload PDF or DOCX files, extract their text content, and prepare the document for summarisation or Q&A.

Includes:

- File upload interface (Streamlit)
- PDF text extraction

Priority: Critical
Complexity: Medium

2. Text Summarisation

Description:
Generate concise summaries of entire documents or selected chapters using the Gemini LLM.

Includes:

- Prompt template for summarization
- Full-document and chapter-level summaries
- Display of summaries in the UI

Priority: Important
Complexity: Medium

3. Context-Aware Document Q&A

Description:
Allow users to ask natural-language questions about their uploaded documents. The system retrieves relevant text sections (or uses full text for MVP) and queries Gemini.

Includes:

- Q&A prompt template
- Retrieval of document segments (optional for MVP)
- Structured answer output

Priority: Important
Complexity: High (due to retrieval + LLM pipeline)

4. Quiz Generation (Content Creation)

Description:
Automatically produce study materials such as multiple-choice questions, fill-in-the-blank questions, and short-answer questions based on the content of the uploaded document.

Includes:

- Quiz prompt templates
- Function calling for structured quiz output
- Display of quizzes in UI

Priority: Low
Complexity: Medium–High

5. User Interface (Streamlit Frontend)

Description:
Frontend interface enabling students to interact with the system intuitively.

Includes:

- Document upload section
- Summaries and Q&A pages
- Quiz display page
- Basic styling & layout

Priority: Normal
Complexity: Medium

6. Logging & Observability (Langfuse)

Description:
Track LLM calls, latency, errors, and interactions for debugging and improvement.

Includes:

- Integration with Python backend
- Logging key events (document ingestion, summarization, Q&A calls)
- Dashboard for monitoring

Priority: Important (for the testing purposes)
Complexity: Low–Medium

7. Basic Document Management

Description:
Store the uploaded documents and extracted text, so users do not need to re-upload them within the same session and then delete them as the session ends.

Includes:

- Local file storage
- Session-state management in Streamlit

Priority: Nice-to-have
Complexity: Low