16/9/2025

SRS Document for Quality Education (Mini-Proj)

Problem Statement :

Online learners face information overload due to the abundance of study material from PDFs, notes, and textbooks. The absence of personalized study tools adversely affects engagement and retention rates. Existing e-learning platforms lack integrated features like summarization, flashcards, semantic search, and voice interaction in a single ecosystem, creating a need for an ai-powered assistant that makes learning faster, interactive, and adaptive.

1. Introduction

1.1 Purpose of the Document

This document formally specifies the software requirements for the development of an AI-powered personalized learning platform. It guides developers, faculty, and testers throughout the project life cycle and ensures all stakeholders share a unified understanding of the platform's goals and deliverables

1.2 Scope of the document

~~SRS is~~

- Automatic summarization of uploaded study materials (PDFs, slides, notes)
- Flashcards and MCQ generation for active recall and exam preparation
- Semantic search and LLM-based Q&A for fast discovery of academic concepts
- Voice-enabled interaction and concept mapping for multimodal access and topic visualization

- Admin functions for educators to manage users and monitor student progress

1.3 Overview

The project leverages AI/ML, specifically transformer models (BERT, T5), NER and clustering for quiz generation, and speech-to-text DNNs for accessibility. Features are accessed via a web dashboard and voice UI, optimized for typical college classroom and home study scenarios.

2. General Description

- The platform solves academic overload by automating summarization and recall.
- Integrated tools (summarization, flashcards, quizzes, voice) consolidate disparate learning resources into a single student-friendly portal for BMSCE and similar college environments
- Designed for accessibility, with special consideration for students needing assistive technology

3. Functional Requirements

- Automatic Summarization: Upload PDF/text notes, system automatically generates highlights and brief summaries using models
- Flashcards & MCQ Generation: NLP/NER extracts key facts from notes to create flashcards and quiz questions tailored for exam revision.
- Voice Interaction: Students ask questions or navigate modules using speech-to-text with high accuracy (>90%), accessible via browsers/mobile.

- Concept Mapping : Generate visual links b/w courses, concepts, and chapters for improved comprehension.

- User profiling : Maintains personal study histories and recommends next steps as per learning progress (supports differentiated instruction).

- Admin Dashboard : Faculty can upload class material, view student analytics, and manage study resource

4. Interface Requirements

- Web UI : Dashboard for students to upload files, access summaries/quizzes, and view progress; responsive and compatible with college intranet bandwidth.

- Voice UI : Browser-based voice capture : essential controls available via speech for visually impaired/physically challenged students.

- REST APIs : Integrate ML backends for summarization, quiz generation, and search.

- Educator/Admin Panel : Manage enrollment, resource uploads, and analytics summaries.

5. Performance Requirements

- Summarization/search must return results within 5 seconds on typical PPT/note sizes used in college (≤ 100 pages)

- Quiz / flashcards generation completes in ≤ 10 seconds per document (max size : 50 pages)

- System supports ≥ 500 concurrent student and faculty logins, suitable for small-medium college deployment

- Speech recognization achieves >90% accuracy in quiet environment; average response time < 2 secs

6. Design Constraints:

- Must run as micro-services on standard cloud infrastructure (AWS/ Azure/ GCP or BMSCE data center) for reliability and scalability

- All ML models retrainable periodically with new syllabus/content; supports student feedback and educator updates.

- Voice modules robust to classroom-level ambient noise.

7. Non-Functional Requirements

- Security : All data encypted at rest and in transit; role-based access for faculty/students.
- Privacy : Student's records and submissions access controlled, in line with academic policies.
- Maintainability : Codebase must use industry std frameworks with documentation for maintenance by next year's project team.

• Usability : All UI features visually distinct, keyboard/voice accessible; easy onboarding for first-year students

• Reliability : Cloud uptime SLA ≥ 99.5%; automatic failover for critical ML services.

• Scalability : Modular design; supports additional colleges and added courses with minimal downtime.

8. Preliminary Schedule and Budget

Schedule

Phase	Duration (weeks)
Requirement analysis	2
Design and Architecture	2
Model integration & Development	4
Testing	2
Deployment & User feedback	2

Budget Estimate

item	Amount (INR)
cloud hosting & compute (6months)	264600
ML Model Training & API usage	61740
Development Team (4 members)	352800
Miscellaneous (licenses, marketing)	44100
Total	723240