Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	27 June 2025
Team ID	LTVIP2025TMID30136
Project Name	EduTutor AI: Personalized Learning with Generative AI and LMS Integration
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Chat Functionality	Engage in conversational interactions with Edu-Tutor. Receive AI-generated responses to user queries.
FR-2	Text Summarization & Explanation	Summarize user-provided text into "short" or "detailed" formats. Provide "simple explanations" of user-provided text.
FR-3	Text Refinement	Refine and improve the clarity, grammar, and style of user text.
FR-4	Word Meaning & Usage Lookup	Provide definitions and example sentences for a given word.
FR-5	Sentence Translation	Translate English sentences into selected target languages (Hindi, Spanish, French, German, Japanese, Tamil).
FR-6	Quiz Generation	Generate multiple-choice quizzes based on a user-provided topic. Generate multiple-choice quizzes from an uploaded PDF document. Allow users to specify the number of questions and difficulty (easy, medium, hard).
FR-7	Quiz Submission & Feedback	Enable users to submit their answers to generated quizzes. Display immediate feedback including score, correct answers, and explanations for each question.
FR-8	Performance Tracking & Insights	Record quiz results (topic, difficulty, score) in a local database. Display a historical DataFrame of all quizzes taken. Identify and display "weak areas" based on past quiz performance. Suggest next steps based on performance insights.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The Gradio-based user interface should be intuitive, easy to navigate, and provide clear interactive elements across all functional tabs, ensuring a smooth learning experience for the user.
NFR-2	**Security**	The application is designed for local, single-user operation and relies on the security provided by the underlying operating system and Python environment. Explicit advanced security measures (e.g., user authentication, data encryption at rest/in transit, access controls) are not implemented in the current code.
NFR-3	**Reliability**	The application's reliability is contingent on the stability of the local server environment and the consistent availability and performance of the loaded LLM. Basic error handling is in place for database operations.
NFR-4	**Performance**	Key performance considerations include the speed of LLM inference for text generation, summarization, and quiz creation. The application uses accelerate and torch to optimize LLM operations, allowing for GPU utilization if available, to enhance processing speed.
NFR-5	**Availability**	As a single-instance Gradio application deployed on a local server, its availability is directly tied to the uptime of that specific instance. The current architecture does not include mechanisms for high availability such as load balancing or redundant deployments.
NFR-6	**Scalability**	The current architecture is a monolithic application. For increased user load or to support multiple concurrent users, the application would primarily scale vertically (by increasing server resources). Horizontal scaling would necessitate a significant architectural re-design, potentially involving containerization and orchestration.