

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

Date	4 February 2026
Team ID	LTVIP2026TMIDS91648
Project Name	Gemini Historical Artifact Description
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	User Input Handling	<ul style="list-style-type: none"> • User can enter the Artifact Name • User can enter the Historical Period / Civilization • User can select the word count (100–2000 words)
FR-2	Input Validation	<ul style="list-style-type: none"> • System checks that Artifact Name is not empty • System checks that Historical Period is entered • System ensures word count is within allowed range (100–2000)
FR-3	AI Artifact Description Generation	<ul style="list-style-type: none"> • System creates structured prompt for Gemini API • System sends request to Gemini API • System receives generated artifact description
FR-4	Content Display	<ul style="list-style-type: none"> • System displays generated description clearly • System shows Title, Origin, Material, Historical Context, Cultural Significance

FR-5	Markdown Export	<ul style="list-style-type: none"> System converts content to Markdown format User can download .md file
FR-6	Loading Experience	<ul style="list-style-type: none"> System displays a historical fact or quote during AI processing
FR-7	Error Handling	<ul style="list-style-type: none"> System shows user-friendly error message if API fails System prevents application crash

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system shall provide a clean, intuitive Streamlit interface requiring no technical expertise.
NFR-2	Performance	AI generation should complete within acceptable response time (typically 3-8 seconds depending on API latency).
NFR-3	Reliability	The system shall handle API failures gracefully without crashing.
NFR-4	Availability	The application should be accessible whenever the hosting platform is active.
NFR-5	Scalability	The system should handle multiple concurrent requests depending on Gemini API limits.
NFR-6	Maintainability	The codebase should be modular and easy to update for future enhancements.
NFR-7	Security	API keys must be securely stored using environment variables or Streamlit secrets.