GENERATIVE AI WITH GOOGLE CLOUD

GEMINI HISTORICAL ARTIFACT DESCRIPTION APP

CHAPTER 1 - INTRODUCTION

1.1 Project Overview

The Gemini Historical Artifact Description App is an AI-powered web application developed using Streamlit and Google Gemini Generative AI. The application allows users to generate structured and engaging historical artifact descriptions by simply entering the name of an artifact or historical topic.

The system uses Google's Generative AI model to produce high-quality, well-structured content that includes an introduction, historical background, significance, interesting insights, and conclusion. The application provides a clean and interactive user interface where users can specify the desired word count and download the generated content.

This project integrates frontend interface development, backend processing, and AI model interaction to deliver intelligent content generation in real time.

1.2 Purpose

The primary purpose of this project is to develop an intelligent content generation system that assists users in creating detailed historical artifact descriptions quickly and efficiently. The objectives of this project are:

- To utilize Google Gemini Generative AI for automated content generation.
- To provide a simple and user-friendly web interface.
- To generate structured and academically relevant content.
- To allow customization of output length.
- To demonstrate the integration of AI models into real-world web applications.

This project aims to simplify historical content creation for students, educators, researchers, and content creators by leveraging modern AI technology.

CHAPTER 2 - IDEATION PHASE

2.1 Problem Statement

In today's digital era, students, researchers, and content creators often require well-structured and informative descriptions of historical artifacts for academic projects, blogs, presentations, and research work. However, manually researching and writing accurate historical content is time-consuming and requires reliable sources.

Many users face challenges such as:

- Lack of structured information.
- Limited time to conduct detailed research.
- Difficulty in organizing historical data into readable formats.
- Inconsistent quality of online sources.

Therefore, there is a need for an intelligent system that can generate structured, accurate, and engaging historical artifact descriptions instantly based on user input.

The proposed solution, **Gemini Historical Artifact Description App**, addresses this problem by leveraging Generative AI to automatically create detailed and well-organized content.

2.2 Empathy Canvas

An empathy map is a simple and easy-to-understand visual tool that captures insights about a user's behaviors, thoughts, feelings, and challenges. It helps teams step into the user's perspective and better understand their needs, motivations, and pain points.

For the **Gemini Historical Artifact Description App**, the empathy map focuses on users such as historians, museum curators, students, researchers, and bloggers who require structured and accurate historical content.

Creating this empathy map allows the team to clearly identify:

- What users think and feel while researching historical artifacts
- What they see in their environment
- What they hear from peers and colleagues
- What they say and do during the research and writing process
- The challenges (pain points) they experience
- The benefits (gains) they expect

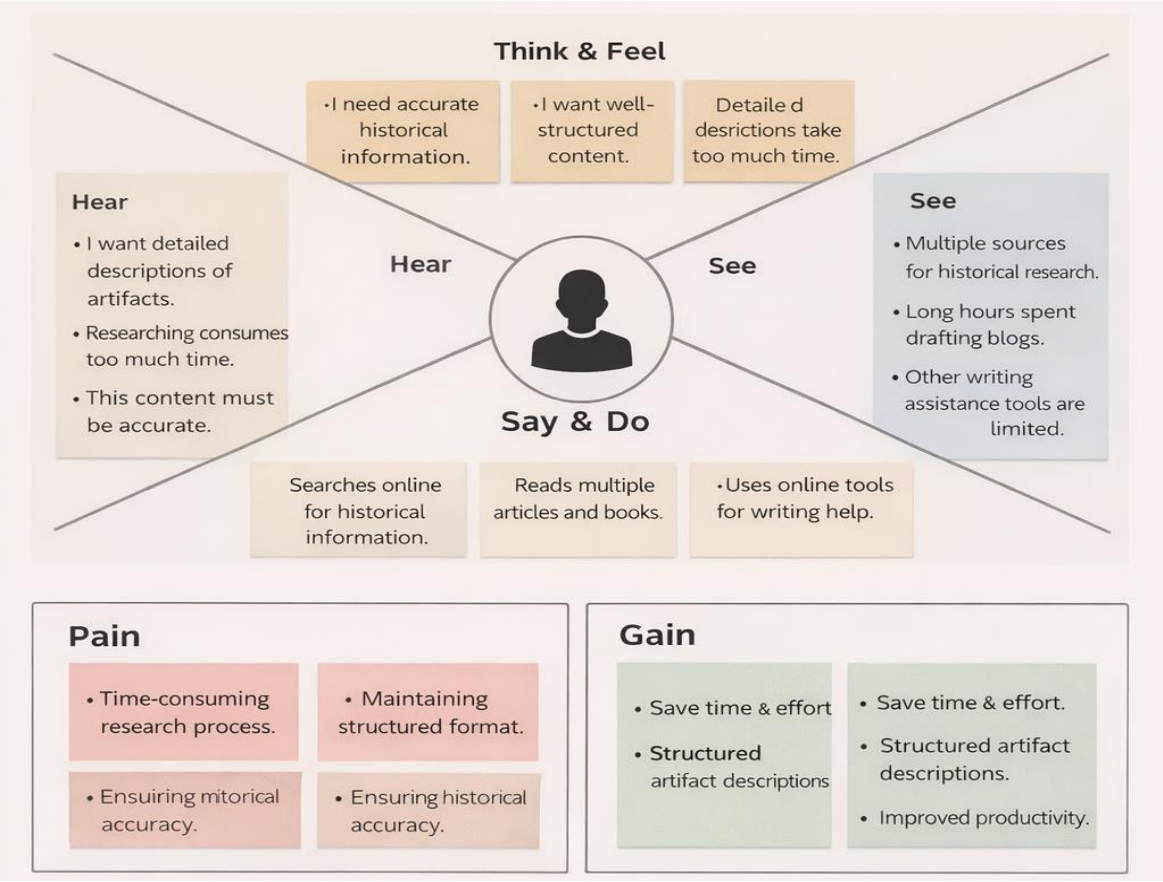
Understanding these aspects helps in designing a solution that truly addresses the user's real problems, such as time-consuming research, lack of structured content, and difficulty maintaining academic accuracy.

The empathy mapping process ensures that the developed application is:

- User-centered
- Efficient
- Academically reliable
- Productivity-enhancing

By analyzing the user's emotions, frustrations, and goals, the team is able to design a solution that reduces research effort and provides high-quality, structured historical descriptions using AI.

Example: Gemini Historical Artifact Description App



2.3 Brainstorming

During the brainstorming phase, the team identified multiple ideas to solve the problem and selected the most feasible solution.

Ideas Generated

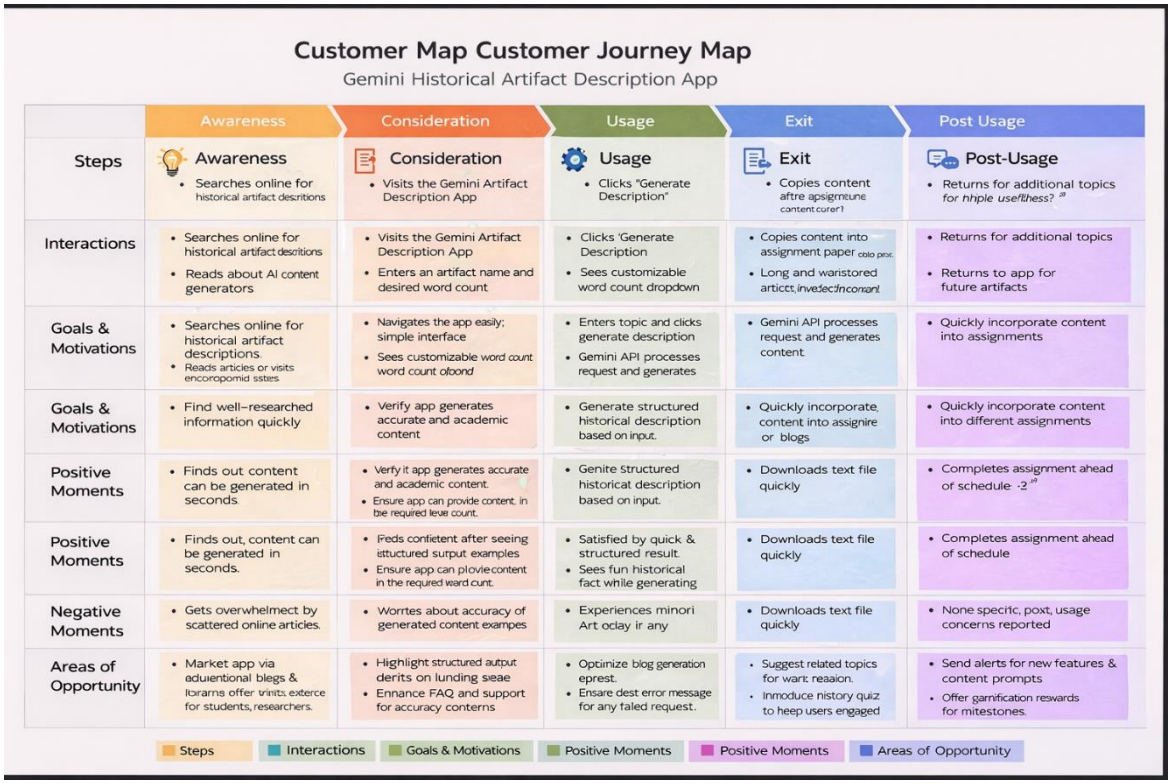
- Static website containing historical articles.
- Manual content database for artifacts.
- Chatbot-based historical assistant.
- AI-powered automated content generator.
- Image-based artifact recognition with description.
- Voice-based artifact description system.

Evaluation Criteria

- Ease of implementation
- Scalability
- Innovation
- Real-world usefulness
- Integration with Generative AI

CHAPTER 3 – REQUIREMENT ANALYSIS

3.1 Customer Map Journey



3.2 Solution Requirement

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	• User can enter artifact name or historical topic • User can specify desired word count
FR-2	Blog Generation	• System generates structured blog using Gemini API • System includes introduction, background, significance, conclusion
FR-3	Display Output	• Generated content is displayed on the web interface • Structured headings are shown clearly
FR-4	Download Feature	• User can download generated blog as text file
FR-5	Interesting Fact Feature	• System displays a random historical fact during generation

FR-6	API Integration	• System connects to Google Gemini API for content generation
FR-7	Error Handling	• System shows error message if input is empty • System handles API failure gracefully

Non-functional Requirements:

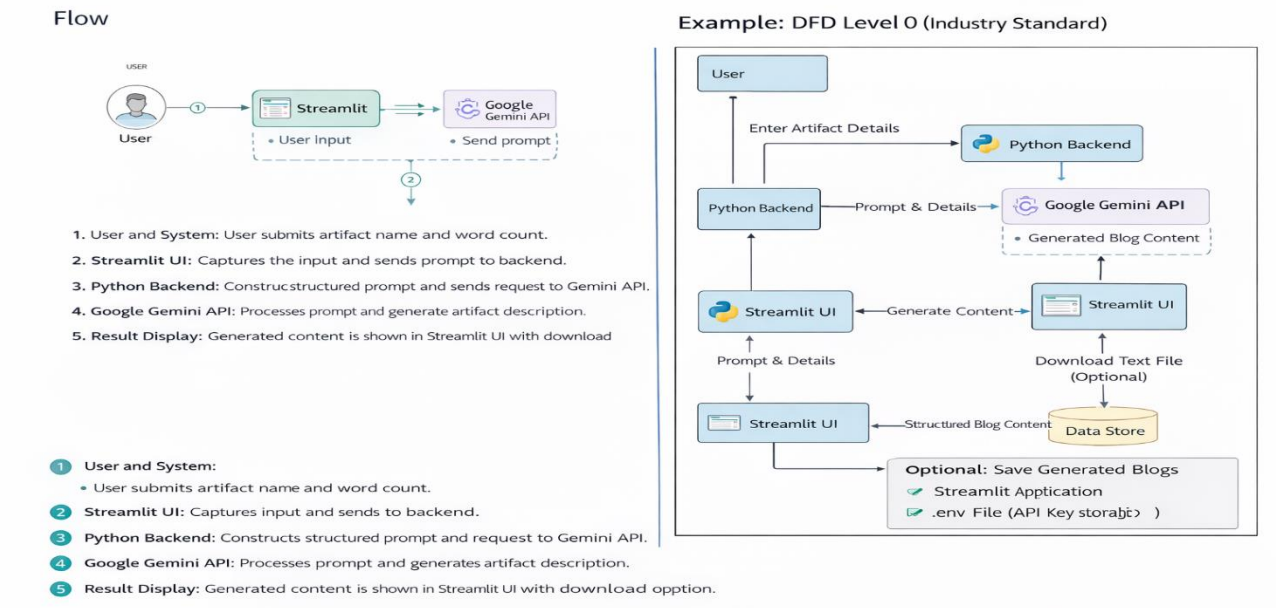
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The application should have a simple, user-friendly web interface accessible via browser.
NFR-2	Security	API key must be securely stored using environment variables (.env file).
NFR-3	Reliability	The system should generate accurate and structured content consistently without crashes.
NFR-4	Performance	Blog generation should complete within a few seconds depending on API response time.
NFR-5	Availability	The application should be accessible online 24/7 when deployed on cloud.
NFR-6	Scalability	The system should support multiple users if deployed on cloud infrastructure.

3.3 Data Flow Diagram

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



3.4 Technology Stack

Technical Architecture:
The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2
Example: Gemini Historical Artifact Description App

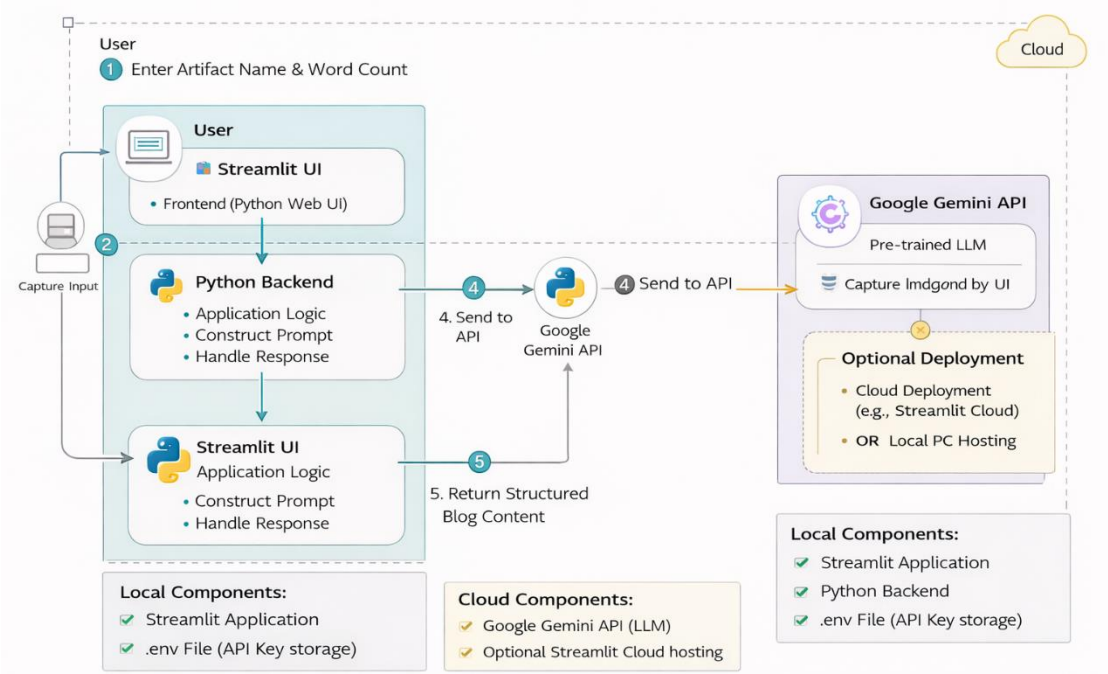


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Allows user to input artifact name and word count	Streamlit (Python Web UI)
2	Application Logic-1	Prompt construction and response handling	Python
3	Application Logic-2	AI content generation processing	Google Gemini API
4	Machine Learning Model	Pre-trained LLM for text generation	Gemini-Pro Model

5	Database	Not mandatory (stateless app)	Not Used
6	Cloud Deployment	Hosting of application	Streamlit Cloud / Localhost
7	File Storage	Download generated blog as .txt file	Local File System
8	External API-1	AI text generation service	Google Generative AI API
9	Infrastructure	Local system or Cloud hosting	Local PC / Cloud Deployment

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1	Open-Source Frameworks	Web UI and backend framework	Streamlit, Python
2	Security Implementations	API key stored securely in .env file	Environment Variables
3	Scalable Architecture	Can scale by deploying on cloud	Streamlit Cloud
4	Availability	Available via web browser anytime	Web-based deployment
5	Performance	Lightweight app, fast API response	Gemini API Optimization

CHAPTER 4 - PROJECT DESIGN

4.1 Problem Solution Fit

Problem – Solution Fit Template:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

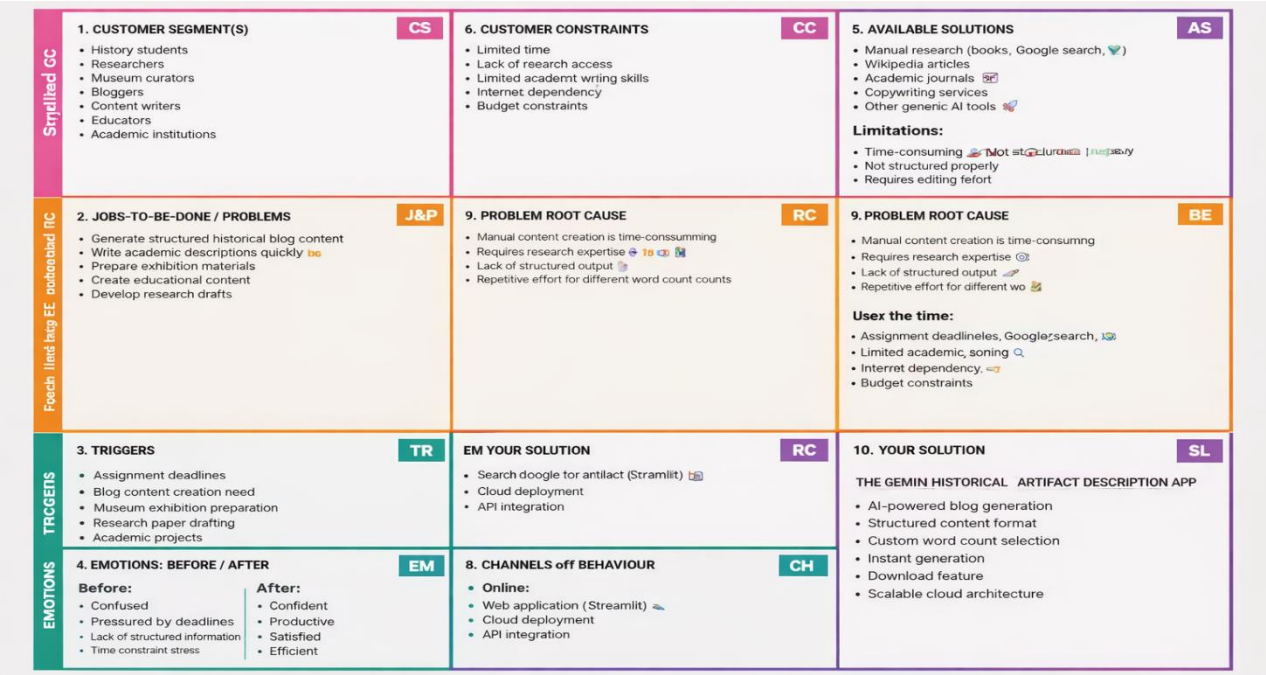
Purpose:

The purpose of the Problem–Solution Fit is to validate that:

- A real and significant problem exists for the target users.
- The proposed AI-powered solution effectively solves that problem.
- The solution aligns with user behavior, constraints, and expectations.
- The application delivers measurable value to its users.

In this project, the goal is to ensure that the **Gemini Historical Artifact Description App** genuinely addresses the challenges faced by students, researchers, bloggers, and educators when creating structured historical content.

Template:



4.2 Proposed Solution

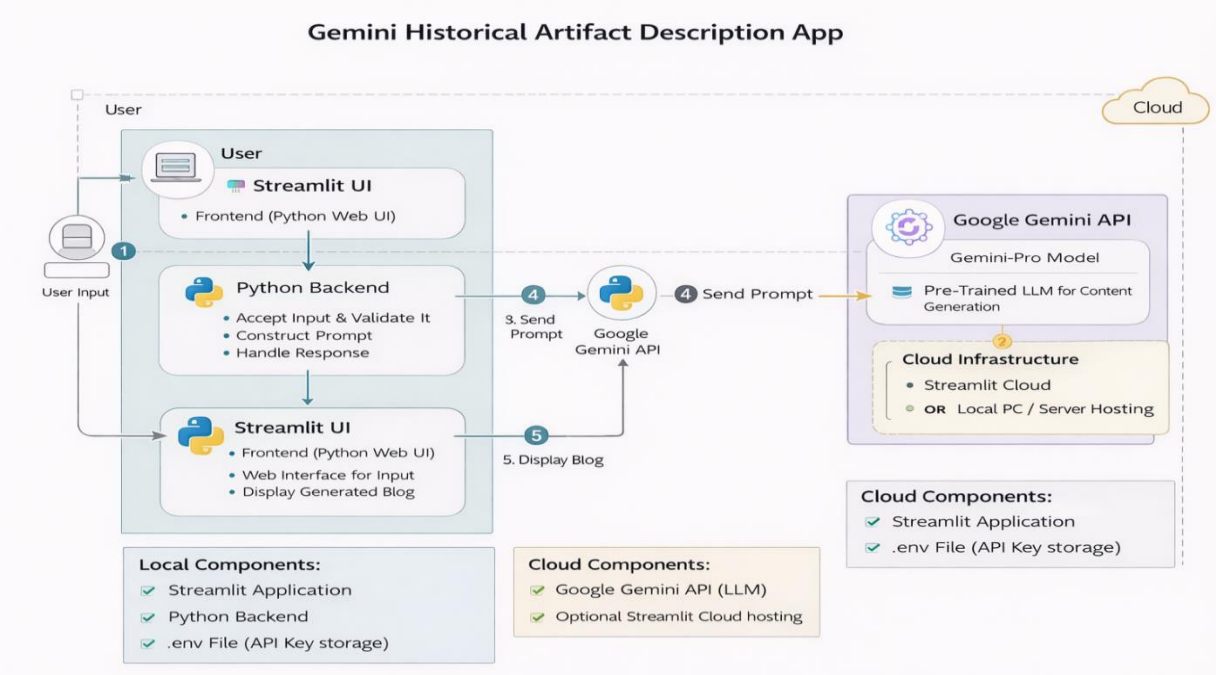
S.No	Parameter	Description
1	Problem Statement	Historians, students, bloggers, and researchers struggle to generate structured and engaging historical artifact descriptions quickly and accurately.
2	Idea / Solution Description	Develop an AI-powered web application that uses Google Gemini 1.5 Flash to generate detailed historical artifact descriptions based on user input and desired word count.
3	Novelty / Uniqueness	Uses Generative AI for automated academic-quality content creation with customizable word count and structured formatting.
4	Social Impact / Customer Satisfaction	Helps students, educators, bloggers, and researchers save time and access well-structured historical information instantly.
5	Business Model (Revenue Model)	Freemium model – Limited free generations; premium subscription for advanced features and unlimited usage.
6	Scalability of the Solution	Cloud-based architecture allows horizontal scaling. Gemini API supports high concurrent requests. Can integrate caching and load balancing.

4.3 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:



CHAPTER 5 - PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story No	User Story / Task	Story Points	Priority	Team Members
Sprint-1	UI Development	USN-1	Design Streamlit UI for topic & word count input	3	High	Developer
Sprint-1	AI Integration	USN-2	Integrate Google Gemini API	5	High	Developer
Sprint-1	Blog Generation	USN-3	Implement structured prompt generation	4	High	Developer
Sprint-1	Output Display	USN-4	Display generated blog with formatting	3	High	Developer
Sprint-2	Download Feature	USN-5	Implement text file download functionality	2	Medium	Developer
Sprint-2	Interesting Fact Feature	USN-6	Add random historical fact display	2	Medium	Developer
Sprint-2	Error Handling	USN-7	Implement input validation & error messages	2	High	Developer
Sprint-3	Deployment	USN-8	Deploy application on Streamlit Cloud	3	Medium	Developer
Sprint-3	Testing	USN-9	Perform functional and performance testing	4	High	Developer

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed	Sprint Release Date
Sprint-1	18	10 Days	29 Jan 2026	07 Feb 2026	18	07 Feb 2026
Sprint-2	8	6 Days	09 Feb 2026	14 Feb 2026	8	14 Feb 2026
Sprint-3	6	5 Days	16 Feb 2026	20 Feb 2026	6	20 Feb 2026

CHAPTER 6 - FUNCTIONAL AND PERFORMANCE TESTING

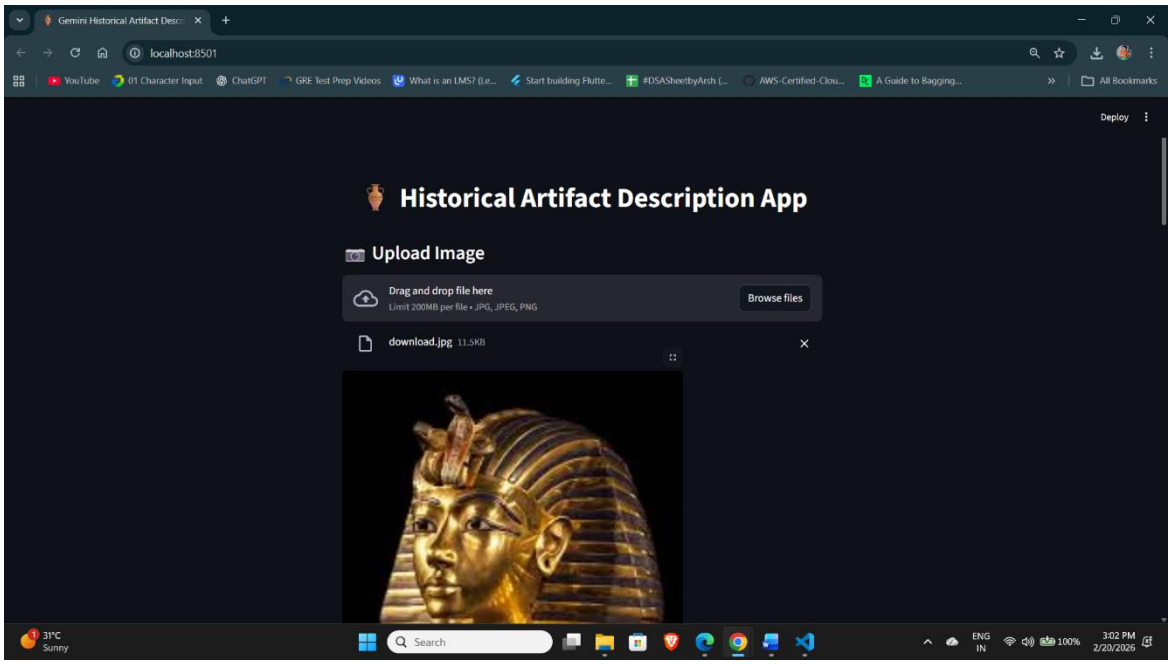
6.1 Performance Testing

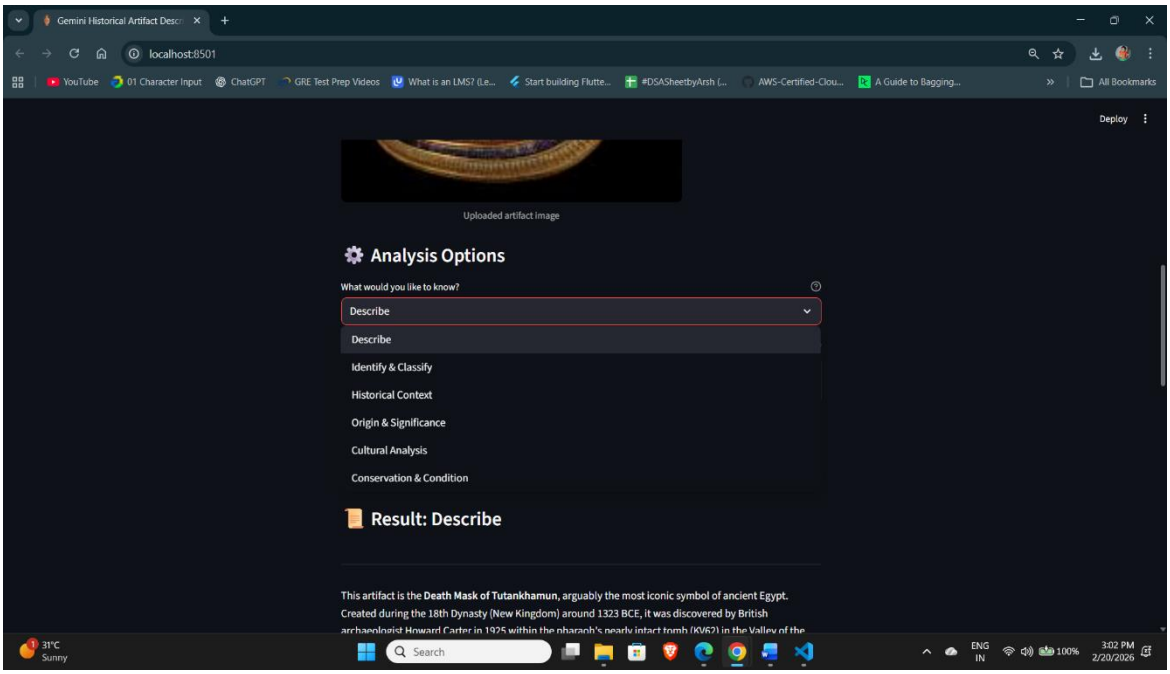
Test Scenarios & Results

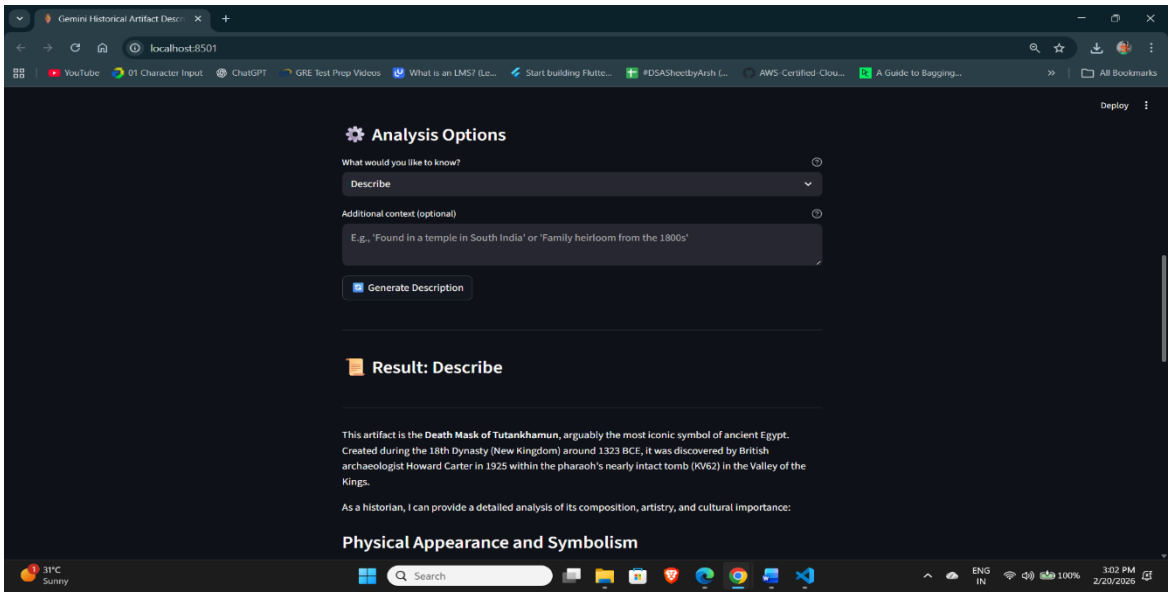
Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (Artifact Name)	Enter valid and invalid artifact names (empty, special characters, long text)	Valid input accepted; empty input shows warning	System shows warning for empty input and accepts valid input	Pass
FT-02	Word Count Validation	Enter word count within and outside allowed range	Accepts valid range; shows error for invalid range	Valid range accepted; invalid blocked	Pass
FT-03	Blog Generation Functionality	Enter topic & word count and click "Generate Description"	Structured blog content generated	Blog generated with headings & formatting	Pass
FT-04	API Connection Check	Verify API key and trigger blog generation	Gemini API responds successfully	API responded and content generated	Pass
FT-05	Download Feature	Generate blog and click download button	Text file downloads successfully	File downloaded without error	Pass
FT-06	Interesting Fact Feature	Click generate button	Random historical fact displayed during loading	Fact displayed properly	Pass
PT-01	Response Time Test	Measure time taken to generate blog	Content generated within acceptable time (<5 seconds depending on API)	Average 3–4 seconds	Pass
PT-02	Multiple Requests Test	Generate content multiple times consecutively	Application should not crash	App handled repeated requests smoothly	Pass
PT-03	Error Handling Test	Disconnect internet / provide invalid API key	Error message displayed gracefully	Proper error message shown	Pass

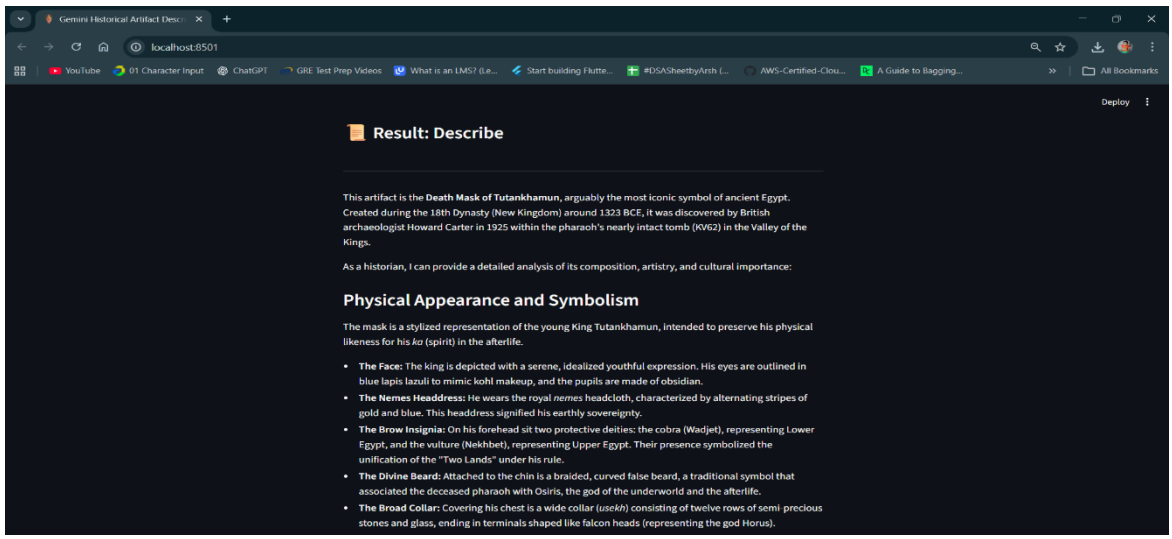
CHAPTER 7 – RESULTS

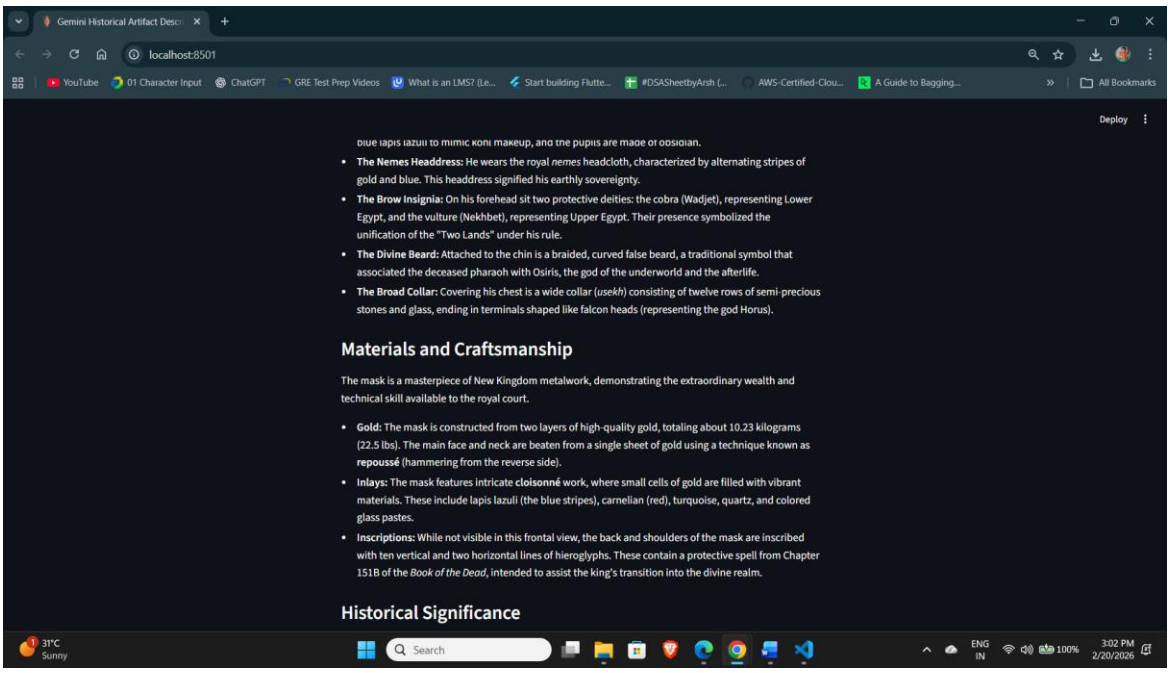
7.1 Outputs Screenshots






Gemini Historical Artifact Descr
localhost:8501
Deploy
Analysis Options
What would you like to know?
Describe
Additional context (optional)
E.g., 'Found in a temple in South India' or 'Family heirloom from the 1800s'
Generate Description
Result: Describe
This artifact is the Death Mask of Tutankhamun, arguably the most iconic symbol of ancient Egypt. Created during the 18th Dynasty (New Kingdom) around 1323 BCE, it was discovered by British archaeologist Howard Carter in 1925 within the pharaoh's nearly intact tomb (KV62) in the Valley of the Kings.
As a historian, I can provide a detailed analysis of its composition, artistry, and cultural importance:
Physical Appearance and Symbolism


Gemini Historical Artifact Descr
localhost:8501
Deploy
Result: Describe
This artifact is the Death Mask of Tutankhamun, arguably the most iconic symbol of ancient Egypt. Created during the 18th Dynasty (New Kingdom) around 1323 BCE, it was discovered by British archaeologist Howard Carter in 1925 within the pharaoh's nearly intact tomb (KV62) in the Valley of the Kings.
As a historian, I can provide a detailed analysis of its composition, artistry, and cultural importance:
Physical Appearance and Symbolism
The mask is a stylized representation of the young King Tutankhamun, intended to preserve his physical likeness for his ka (spirit) in the afterlife.
The Face: The king is depicted with a serene, idealized youthful expression. His eyes are outlined in blue lapis lazuli to mimic kohl makeup, and the pupils are made of obsidian.
The Nemes Headdress: He wears the royal nemes headcloth, characterized by alternating stripes of gold and blue. This headdress signified his earthly sovereignty.
The Brow Insignia: On his forehead sit two protective deities: the cobra (Wadjet), representing Lower Egypt, and the vulture (Nekhbet), representing Upper Egypt. Their presence symbolized the unification of the "Two Lands" under his rule.
The Divine Beard: Attached to the chin is a braided, curved false beard, a traditional symbol that associated the deceased pharaoh with Osiris, the god of the underworld and the afterlife.
The Broad Collar: Covering his chest is a wide collar (usekh) consisting of twelve rows of semi-precious stones and glass, ending in terminals shaped like falcon heads (representing the god Horus).


Gemini Historical Artifact Descr
localhost:8501
Deploy
blue lapis lazuli to mimic kohl makeup, and the pupils are made of obsidian.
The Nemes Headdress: He wears the royal nemes headcloth, characterized by alternating stripes of gold and blue. This headdress signified his earthly sovereignty.
The Brow Insignia: On his forehead sit two protective deities: the cobra (Wadjet), representing Lower Egypt, and the vulture (Nekhbet), representing Upper Egypt. Their presence symbolized the unification of the "Two Lands" under his rule.
The Divine Beard: Attached to the chin is a braided, curved false beard, a traditional symbol that associated the deceased pharaoh with Osiris, the god of the underworld and the afterlife.
The Broad Collar: Covering his chest is a wide collar (usekh) consisting of twelve rows of semi-precious stones and glass, ending in terminals shaped like falcon heads (representing the god Horus).
Materials and Craftsmanship
The mask is a masterpiece of New Kingdom metalwork, demonstrating the extraordinary wealth and technical skill available to the royal court.
Gold: The mask is constructed from two layers of high-quality gold, totaling about 10.23 kilograms (22.5 lbs). The main face and neck are beaten from a single sheet of gold using a technique known as repoussé (hammering from the reverse side).
Inlays: The mask features intricate cloisonné work, where small cells of gold are filled with vibrant materials. These include lapis lazuli (the blue stripes), carnelian (red), turquoise, quartz, and colored glass pastes.
Inscriptions: While not visible in this frontal view, the back and shoulders of the mask are inscribed with ten vertical and two horizontal lines of hieroglyphs. These contain a protective spell from Chapter 151B of the Book of the Dead, intended to assist the king's transition into the divine realm.
Historical Significance

CHAPTER 8 - ADVANTAGES & DISADVANTAGES

Advantages

1. Time Saving

The application generates detailed historical artifact descriptions instantly, reducing the time required for manual research and content writing.

2. Structured Content Generation

The AI produces well-organized content including introduction, historical background, significance, and conclusion.

3. Customizable Word Count

Users can specify the desired word length, making the tool flexible for assignments, blogs, and research work.

4. User-Friendly Interface

The Streamlit-based interface is simple, clean, and easy to use without technical knowledge.

5. AI-Powered Accuracy

Google Gemini provides context-aware and academically relevant content generation.

6. Download Feature

Users can download the generated content as a text file for further use.

7. Scalability

The system can be enhanced to support multiple languages, image-based artifact recognition, and advanced AI features.

Disadvantages

1. Internet Dependency

The application requires an active internet connection to access the Gemini API.

2. API Limitations

Usage depends on API quotas and may be limited based on subscription or usage plan.

3. AI Content Variability

Although high-quality, AI-generated content may occasionally require manual refinement for academic precision.

4. No Offline Mode

The system does not support offline content generation.

CHAPTER 9 – CONCLUSION

The Gemini Historical Artifact Description App successfully integrates Google Gemini Generative AI with a user-friendly Streamlit interface to generate structured and detailed historical content. The application meets its objective of providing quick, customizable, and academically structured descriptions based on user input.

The project demonstrates effective API integration, AI-based content generation, and agile development practices. Although it depends on internet connectivity and API availability, the system performs efficiently and fulfills both functional and non-functional requirements.

Overall, the project highlights the practical application of Generative AI in simplifying historical content creation and improving user productivity**.**

CHAPTER 10 - FUTURE SCOPE

The Gemini Historical Artifact Description App can be further enhanced with several advanced features to improve functionality and user experience.
Future improvements may include:

- Multi-language Support to generate content in different languages.
- Image-Based Artifact Description, where users can upload images for AI-based analysis.
- User Authentication & History Tracking to save previously generated content.
- Export to PDF or Word Format for academic submissions.
- Advanced Customization Options such as tone selection (academic, simple, professional).
- Cloud Deployment & Scalability Enhancements to support multiple concurrent users.

These enhancements would make the system more robust, scalable, and suitable for wider academic and professional use.

CHAPTER 11 - APPENDIX

Souce Code Link: https://github.com/Keertana04/GENAI

Demo Video Link: https://drive.google.com/file/d/11G_c1UKllaAwMN-UhNp4Y41iNRh6N9Ro/view?usp=sharing