GEMINI HISTORICAL ARTIFACT DESCRIPTION APP

SmartBridge Virtual Internship Project

Team Details

Team ID: LTVIP2026TMIDS59887
Team Size: 5

Team Leader:
Nalajala Sai Teja

Team Members:
Nidrabingi Madhu Kumar
Chinta Ravi Teja
Jidugu Bhargava Adithya
Bajjuri Venkatesh

---

1. INTRODUCTION

1.1 Project Overview

This project, titled “Gemini Historical Artifact Description App,” is developed as part of the SmartBridge Virtual Internship Program. The objective of this project is to generate detailed descriptions of historical artifacts using Artificial Intelligence and Generative AI technology.

In today’s digital era, students, historians, and researchers often require quick and structured information about historical artifacts. Manual research can be time-consuming and information may be scattered across multiple sources. This project provides an intelligent solution that enables users to generate artifact descriptions automatically.

The system accepts artifact names and optional images from users. The input is processed using Google’s Gemini Generative AI model, which generates informative and structured descriptions. The application demonstrates the real-world use of Artificial Intelligence in education and research.

1.2 Purpose

The main purpose of this project is to apply theoretical knowledge of Artificial Intelligence and Python programming to build a practical real-world application.

This project helps in understanding:

- API integration
- Web application development
- Prompt engineering
- AI model interaction

It also improves technical and problem-solving skills.

---

2. IDEATION PHASE

2.1 Problem Statement

Many students and researchers need structured descriptions of historical artifacts for academic and educational purposes. Searching and compiling information manually requires significant time and effort.

There is a need for a system that:

- Generates descriptions automatically
- Provides structured content
- Is easy to use

This project solves the problem using Generative AI.

2.2 Empathy Map Canvas

What the User Thinks

- Wants quick information about artifacts
- Needs accurate and clear descriptions

What the User Feels

- Frustrated searching multiple sources
- Satisfied when results are generated quickly

What the User Says

- "I need a description of this artifact."
- "I want reliable historical information."

What the User Does

- Searches online
- Reads articles

User Pain Points

- Time-consuming research
- Information scattered

User Needs

- Quick generation
- Easy interface

2.3 Brainstorming

Different approaches were considered:

- Manual research systems
- Database-based artifact information
- AI-based description generation

AI-based generation was selected because it provides better scalability and faster results.

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User workflow:

1. User opens the application
2. User enters artifact name
3. User uploads image (optional)
4. System processes input
5. AI generates description
6. Output displayed

3.2 Solution Requirements

Functional Requirements

- Accept artifact name
- Accept image upload
- Generate descriptions
- Display output

Non-Functional Requirements

- Easy to use
- Fast response
- Reliable
- Secure API handling

3.3 Data Flow Diagram

User → Streamlit UI → Gemini API → Generated Description → Output

---

3.4 Technology Stack

Programming Language:

- Python

Framework:

- Streamlit

Libraries:

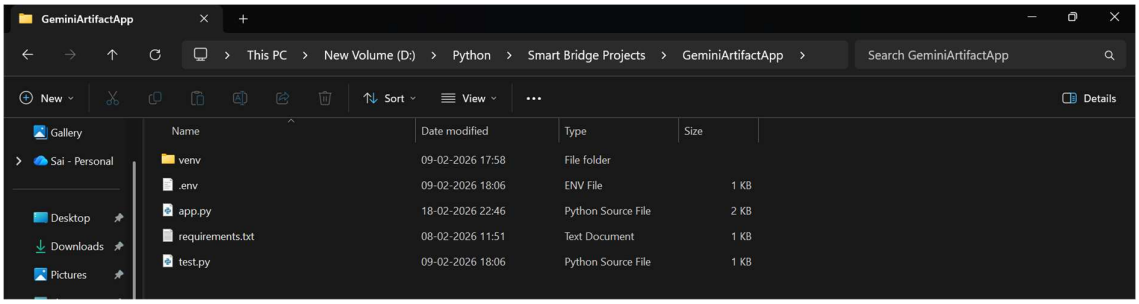
- Google GenAI
- Python-dotenv
- Pillow

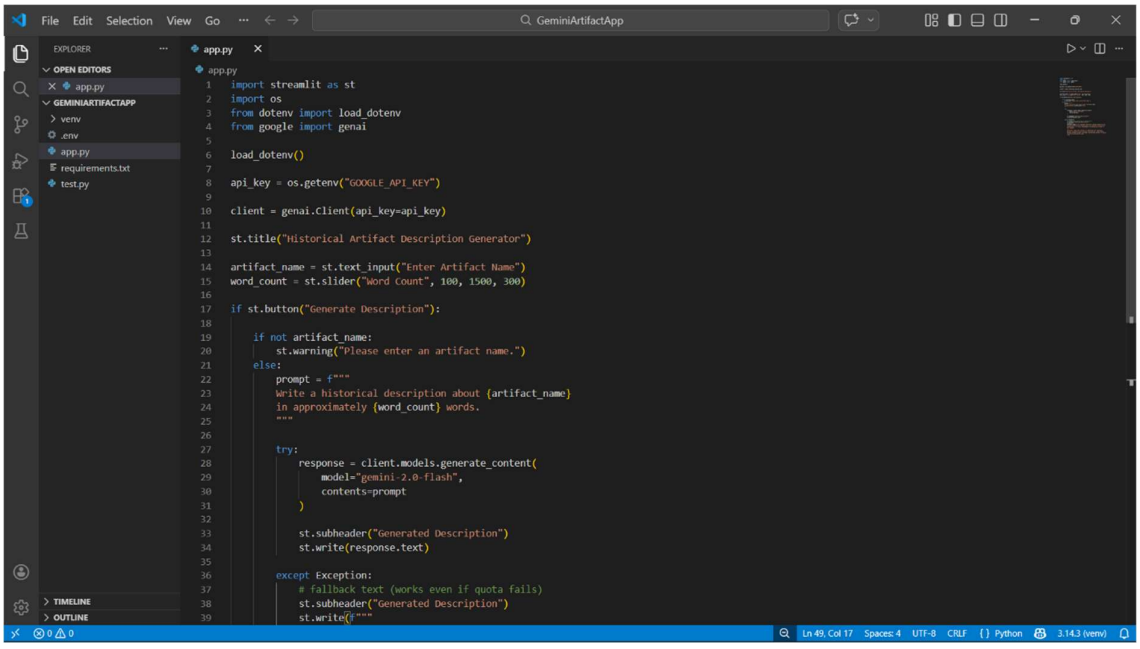
Tools:

- VS Code
- GitHub

Platform:

- Windows OS





4. PROJECT DESIGN

4.1 Problem–Solution Fit

The problem identified is the difficulty of obtaining structured artifact descriptions quickly.

The proposed AI-based solution automates the process and generates meaningful content efficiently.

4.2 Proposed Solution

The system works as follows:

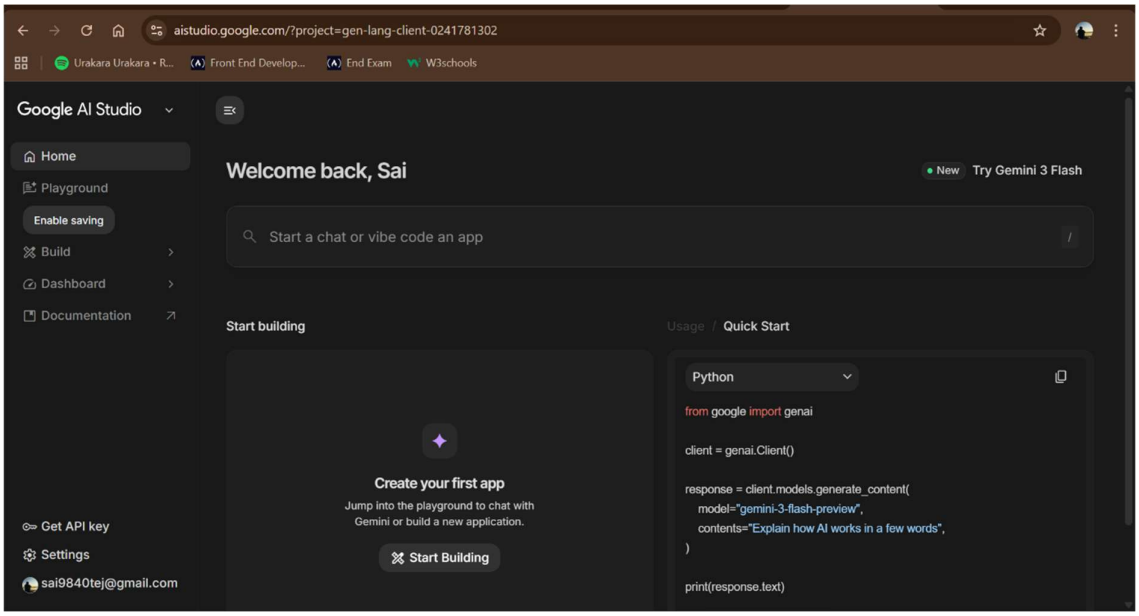
- User enters artifact name
- User uploads image (optional)
- Prompt sent to Gemini
- Description generated
- Output displayed

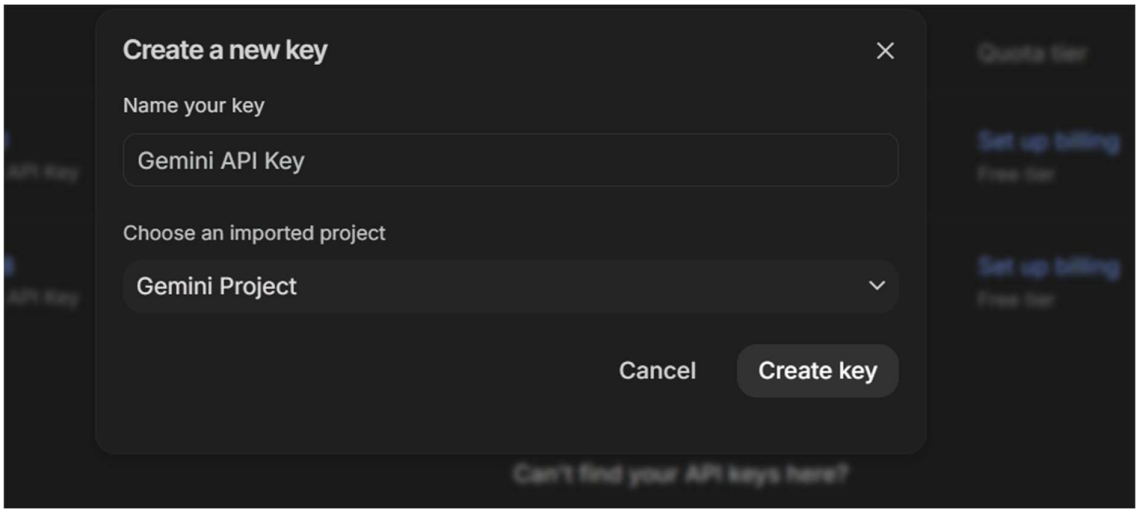
4.3 Solution Architecture

Components:

- User Interface (Streamlit)
- API Integration Layer
- Gemini AI Model

- Output Display Module





5. PROJECT PLANNING AND SCHEDULING

Project Phases

1. Requirement Analysis
2. System Design
3. Development
4. Testing

5. Documentation

Schedule

Phase	Activity	Duration
Phase 1	Requirement Analysis	3 Days
Phase 2	Design	3 Days
Phase 3	Development	7 Days
Phase 4	Testing	4 Days
Phase 5	Documentation	3 Days

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Functional Testing

- Verified text input
- Verified image upload
- Verified AI integration
- Verified output display

6.2 Performance Testing

- Checked response time
- Tested UI interaction
- Verified stability

---

7. RESULTS

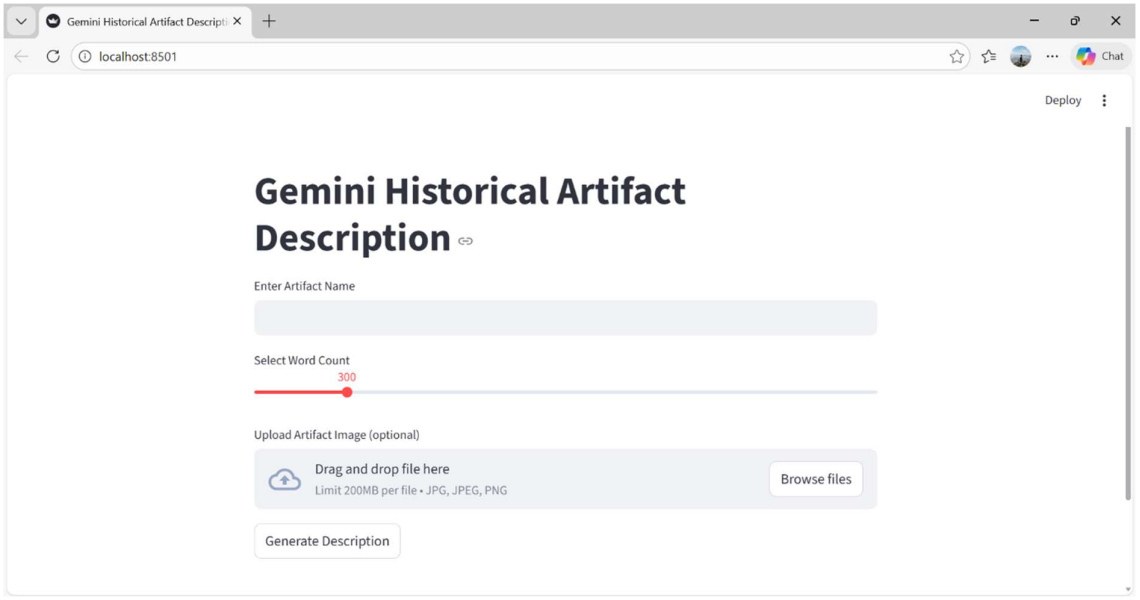
The application successfully:

- Accepts user input
- Generates descriptions
- Displays output correctly

7.2 Output Screenshots

Insert screenshots here:

Figure 1: Main Interface of the Application



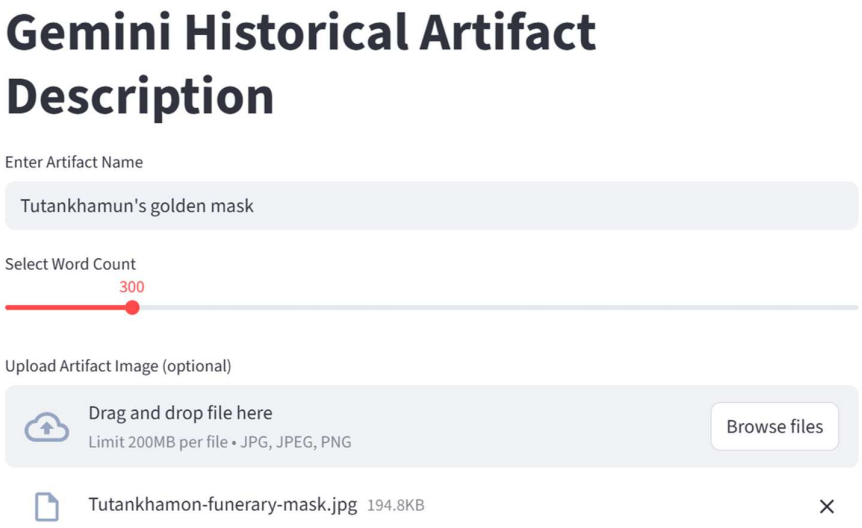
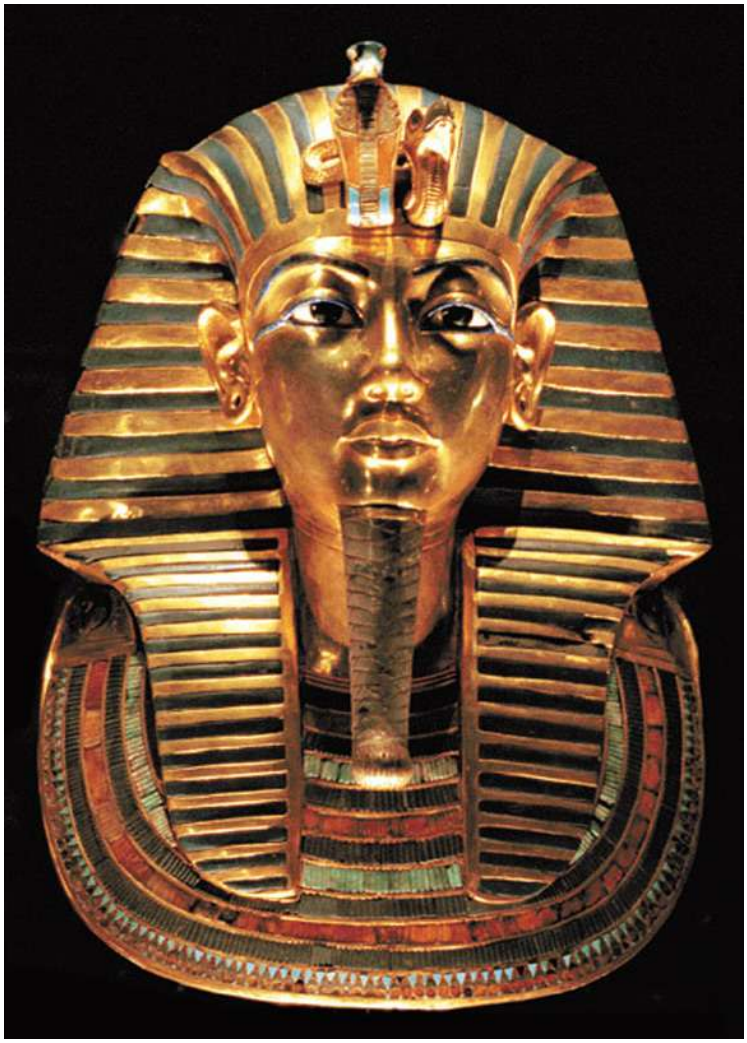


Figure 2: Sample Artifact Image Used



Generated Description

Tutankhamun's golden mask is an important historical artifact known for its cultural and historical significance. It represents the artistic, technological, or social achievements of the period in which it was created.

Historians study such artifacts to understand the traditions, beliefs, and daily life of ancient civilizations. These objects provide valuable insight into human history and continue to inspire research and education today.

8. ADVANTAGES AND DISADVANTAGES

Advantages

- Easy to use
- Fast generation
- AI integration
- Lightweight

Disadvantages

- Requires internet
- API quota limitations

---

9. CONCLUSION

This project demonstrates the integration of Generative AI with a web application. The system successfully generates historical artifact descriptions and provides a simple and effective interface.

---

10. FUTURE SCOPE

- Download option
- Cloud deployment
- Multi-language support
- Mobile version

---

11. APPENDIX

11.1 Security Implementation

- API key stored in .env
- Not hardcoded
- .env ignored in Git

11.2 Project Demo

🔗 Demo Video:
https://drive.google.com/file/d/1N1aVAgPHpLV3AadZjThLmSmje8U7Ka4z/view?usp=drive_link

The demo shows:

- Artifact input
- Image upload
- AI-generated output

11.3 References

- Google Gemini API
- Streamlit Documentation
- Python Documentation