

PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The AI Historical Artifact Description App is a web-based application that analyzes images of historical artifacts and generates detailed historical descriptions using transformer-based artificial intelligence models. The system integrates computer vision and natural language processing to interpret visual inputs and provide structured historical insights.

1.2 Purpose

The purpose of this project is to develop an intelligent system that assists students, researchers, and history enthusiasts in understanding historical artifacts by automatically generating contextual descriptions from images.

2. IDEATION PHASE

2.1 Problem Statement

Understanding historical artifacts requires expert knowledge and manual research. Many individuals lack access to professional archaeologists or historians. There is a need for an automated solution that can analyze artifact images and generate meaningful historical interpretations.

2.2 Empathy Map Canvas

Users:

- Students
- Museum visitors
- History researchers

User Needs:

- Quick artifact understanding
- Easy-to-read descriptions
- Reliable historical context

Pain Points:

- Limited expert access
- Time-consuming research
- Complex historical terminology

2.3 Brainstorming

Multiple solutions were considered:

- Manual description database

- API-based AI integration
- Offline transformer-based solution

The final approach selected was a two-stage transformer pipeline combining image captioning and text generation.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens application
2. Uploads artifact image
3. Clicks “Generate Description”
4. AI processes image
5. Historical description displayed

3.2 Solution Requirement

Functional Requirements:

- Upload image (JPG, PNG)
- Generate caption
- Generate structured historical description
- Display formatted output

Non-Functional Requirements:

- Offline operation
- Fast response time
- Easy-to-use interface

3.3 Data Flow Diagram

User → Upload Image → Image Processing (BLIP Model) → Caption Generation → Text Processing (FLAN-T5 Model) → Historical Description Output → Display in UI

3.4 Technology Stack

- Python
- Streamlit
- Hugging Face Transformers
- PyTorch
- PIL (Image Processing)

4. PROJECT DESIGN

4.1 Problem Solution Fit

The project directly addresses the need for automated historical interpretation by combining visual recognition with contextual language generation.

4.2 Proposed Solution

A two-stage AI pipeline:

1. Vision Model (BLIP) extracts visual caption.
2. Language Model (FLAN-T5-Large) expands caption into structured historical analysis.

4.3 Solution Architecture

User Interface (Streamlit)

↓

Image Upload Module

↓

BLIP Image Captioning Model

↓

FLAN-T5-Large Language Model

↓

Historical Description Output

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was divided into two sprints:

Sprint 1:

- Image upload implementation
- Model integration
- Basic output generation

Sprint 2:

- Historical reasoning enhancement
- UI improvements
- Performance optimization

Velocity achieved: 10 Story Points per Sprint.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Initial model loading time: ~2–3 minutes
 - Subsequent response time: ~5–10 seconds
 - Output length: 150–250 words
 - Minimum 10 sentence generation ensured
 - No critical system failures observed
-

7. RESULTS

7.1 Output Screenshots

- Application Interface
- Uploaded Artifact Image
- Generated Historical Description
- Terminal Execution

(The screenshots are included in the GitHub repository.)

8. ADVANTAGES & DISADVANTAGES

Advantages

- Fully offline operation
- No API dependency
- Detailed structured output
- Educational value
- Scalable architecture

Disadvantages

- Initial model download size is large
 - Performance depends on system hardware
 - Historical accuracy depends on model generalization
-

9. CONCLUSION

The AI Historical Artifact Description App successfully demonstrates the integration of computer vision and natural language processing to generate meaningful historical interpretations from images. The system provides structured, detailed descriptions and operates fully offline using transformer models.

10. FUTURE SCOPE

- Cloud deployment for public access
 - Multilingual support
 - Integration with museum databases
 - Improved fine-tuned historical models
 - PDF export functionality
-

11. APPENDIX

Source Code

Included in GitHub repository.

Dataset Link

Pre-trained models from Hugging Face:

- Salesforce/blip-image-captioning-base
- google/flan-t5-large

GitHub & Project Demo Link

GitHub Repository:

<https://github.com/YourUsername/Gemini-Historical-Artifact-Description>

(Local deployment using Streamlit