

Gemini historical artifact description

Gemini Historical Artifact Description is a web application that leverages Google's Generative AI to create unique and detailed descriptions of historical artifacts. The app allows users to input an artifact name or historical period and specify the desired word count for the description. Using these parameters, the AI generates engaging and informative content. Additionally, the app includes a feature that shares an interesting historical fact to engage users while the AI generates the description.

Scenario 1: Describing an Ancient Egyptian Artifact

A historian specializing in Ancient Egypt opens the Gemini Historical Artifact Description app and inputs "Tutankhamun's Golden Mask" with a 1200-word count. As the app generates the content, it shares a fascinating historical fact. The AI quickly delivers a detailed and insightful description. The historian reviews the well-crafted content and incorporates it into their research or publication, ready to share with their audience.

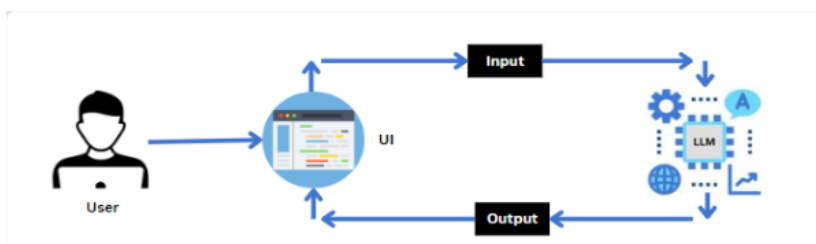
Scenario 2: Crafting a Renaissance Artifact Description

A museum curator looking for detailed descriptions of Renaissance artifacts uses the Gemini Historical Artifact Description app, inputting "Leonardo da Vinci's Notebook" and specifying an 800-word count. The app provides an intriguing historical fact while generating the content. The AI produces a concise yet informative description filled with historical context and artistic significance. The curator finds the description valuable and incorporates it into the museum's exhibition materials.

Scenario 3: Developing a Medieval Artifact Description

A medieval history enthusiast accesses the Gemini Historical Artifact Description app to generate new content for their blog. They enter "The Bayeux Tapestry" as the topic and select a 1500-word count. The app entertains with a historical fact during the content creation process. The AI delivers a comprehensive and well-detailed description. The enthusiast reviews the high-quality content and publishes it on their history blog, confident it will engage and inform their readers.

Architecture



Project Flow

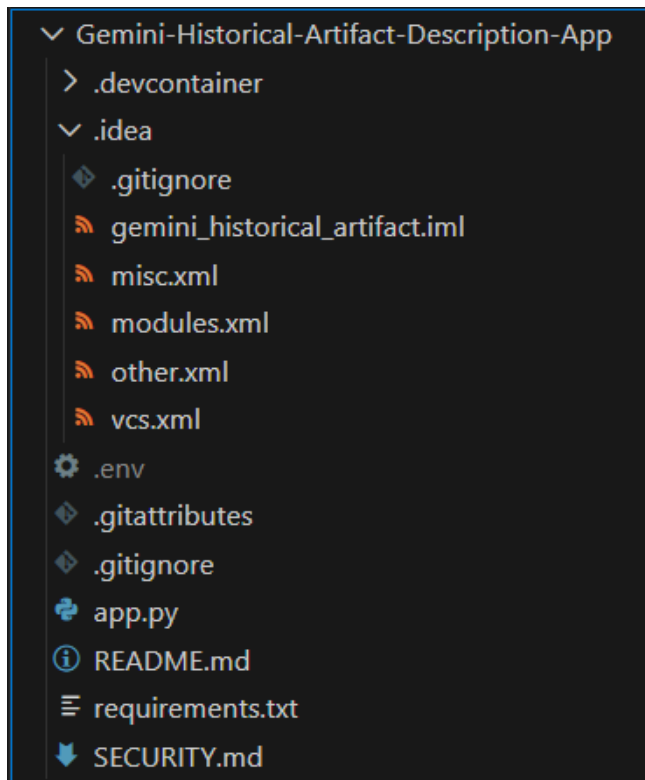
Project Flow

- Users input a topic and specify the desired length of the blog post through the Streamlit UI.
- The input topic and length are sent to the Gemini 1.5 Flash language model, which is integrated into the backend.
- Gemini 1.5 Flash processes the input and generates a blog post based on the user's specifications.
- The model autonomously creates a well-structured, engaging blog post tailored to the specified topic and word count.
- The generated blog post is sent back to the frontend for display on the Streamlit app.
- Users can customize the blog post further if desired and export or copy the content for their use.

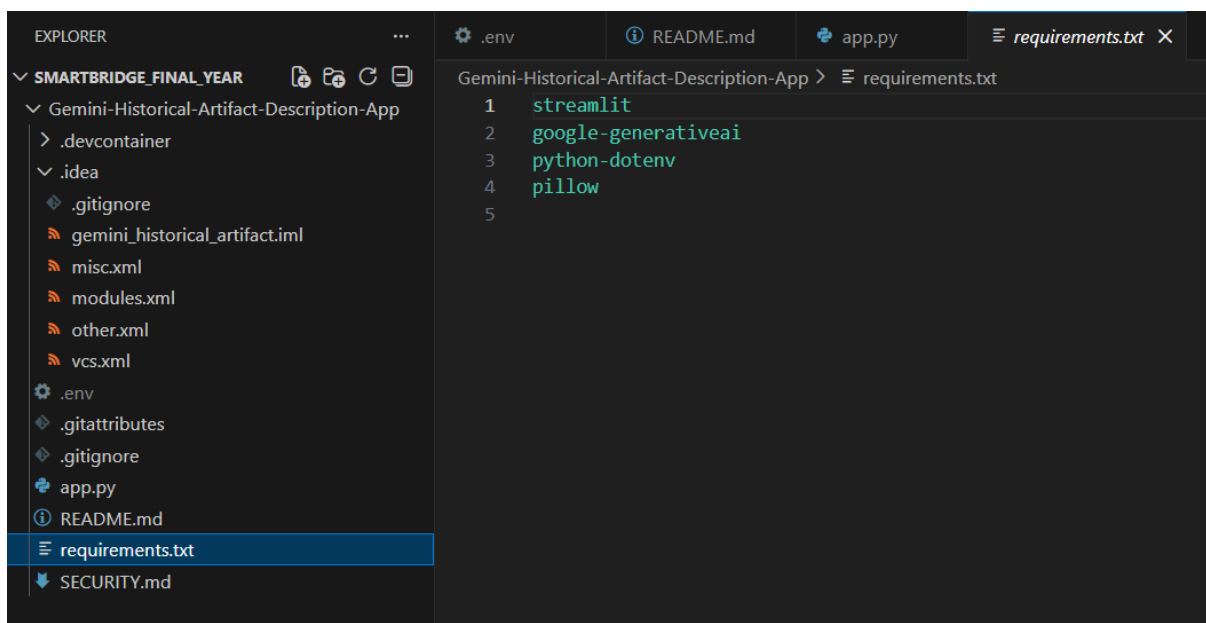
To accomplish this, we have to complete all the activities listed below,

- Initialize Gemini 1.5 Flash:
- Generate Gemini 1.5 Flash API
- Initialize the pre-trained model
- Interfacing with Pre-trained Model
 - Blog Generation
- Model Deployment
 - Deploy the application using Streamlit

Project Structure



Requirements Specification



Function for image setup

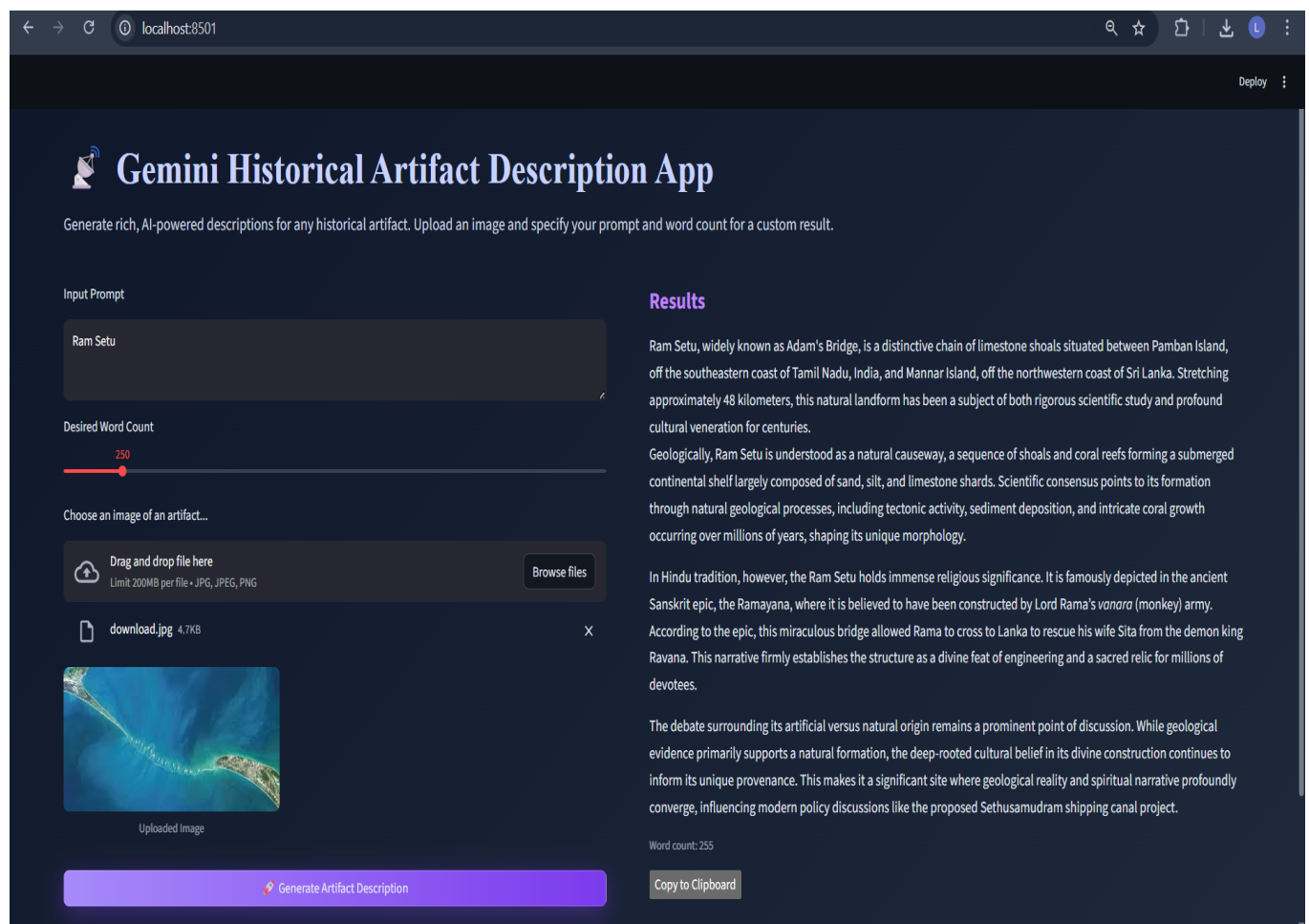
```
# -----
# Image Setup
# -----

def input_image_setup(uploaded_file):
    if uploaded_file is not None:
        bytes_data = uploaded_file.getvalue()
        return [{
            "mime_type": uploaded_file.type,
            "data": bytes_data
        }]
    return None

# -----
# Gemini API Call
# -----

def get_gemini_response(input_text, image_data, prompt, api_key):
    genai.configure(api_key=api_key)
    model_name = os.getenv("GEMINI_MODEL", "gemini-pro")
```

Output:



Demo Video:

https://drive.google.com/file/d/1iJ_tFQAd2gwgdAD7egJvXxdVfOQpZS5G/view?usp=sharing

GitHub Link:

[BolluSnigdha/-Gemini-Historical-Artifact-Description-App](#)