

Hackathon Project Phases Template

10/3/2025

Project Title:

Gemini Landmark Description App

Team Name:

Neural Voyagers

Team Members:

- U. Nityam Kethan (22B81A6627)
- A Siddhartha (22B81A6642)

Phase-1: Brainstorming & Ideation

Objective:

- Identify the problem statement.
- Define the purpose and impact of the project.

Key Points:

1. **Problem Statement:** Tourists lack quick, detailed information about landmarks, reducing their travel experience.
2. **Proposed Solution:** An AI app that generates landmark descriptions from uploaded images, focusing on history and facts.
3. **Target Users:** Travelers, history enthusiasts, and tour guides.
4. **Expected Outcome:** Enriched tourist experiences with instant landmark insights.

Phase-2: Requirement Analysis

Objective:

- Define technical and functional requirements.

Key Points:

1. **Technical Requirements:** Python, Streamlit, Google Generative AI API, Google Vision API, Wikipedia API, gTTS (Text-to-Speech).
2. **Functional Requirements:** Image upload, AI-generated descriptions, text-to-speech conversion, simple web interface.
3. **Constraints & Challenges:** API response time, multilingual translation accuracy, dependency on internet connectivity.

Phase-3: Project Design

Objective:

- Create the architecture and user flow.

Key Points:

1. **System Architecture:** Web app using Streamlit, AI-based image recognition, Wikipedia API for factual data, Google Maps API for location insights.
2. **User Flow:** User uploads an image → AI identifies landmark and generates description → User receives historical data, maps, and audio narration.
3. **UI/UX Considerations:** Minimalistic interface, accessible design, easy navigation.

System Architecture Diagram

Phase-4: Project Planning (Agile Methodologies)

Objective:

- Break down tasks using Agile methodologies.

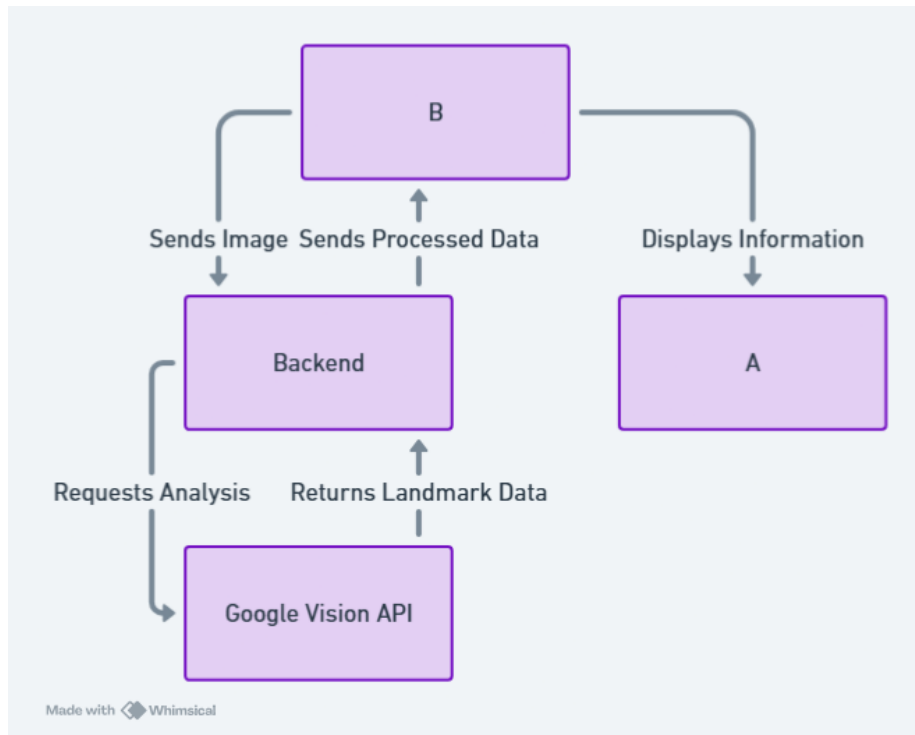


Figure 1: System Architecture of Gemini Landmark Description App

Key Points:

1. Sprint Planning:

- Day 1: Backend setup, Google Vision API integration.
- Day 2: Streamlit UI implementation, testing, and enhancements.

2. Task Allocation:

- U Nityam: AI integration, backend logic in Streamlit.
- A Siddhartha: Streamlit frontend and UI implementation.

3. Timeline & Milestones:

- Day 1 Morning: Google Vision API integration.
- Day 1 Afternoon: Backend logic in Streamlit.
- Day 2 Morning: Streamlit UI implementation.
- Day 2 Afternoon: Integration and testing.

Phase-5: Project Development

Objective:

- Code the project and integrate components.

Key Points:

1. **Technology Stack Used:** Python, Streamlit, Google Generative AI, Google Vision API, Wikipedia API, gTTS.
2. **Development Process:** AI-based landmark detection → Data fetching → Text-to-speech generation → UI rendering.
3. **Challenges & Fixes:** Slow API response handled with caching, improved UI elements for clarity.

Phase-6: Functional & Performance Testing

Objective:

- Ensure the project works as expected.

Key Points:

1. **Test Cases Executed:** Image upload verification, AI response accuracy, Wikipedia link generation, Google Maps integration.
2. **Bug Fixes & Improvements:** Handled edge cases for unidentified landmarks, improved response time with parallel API calls.
3. **Final Validation:** Successfully identifies major landmarks and provides structured descriptions.
4. **Deployment:** Hosted on Streamlit Cloud (or local demo if needed).

Final Submission

1. GitHub/Code Repository Link: https://github.com/Nityam2305/Gemini_Landmark_Descriptor.git