**Project Planning: Gemini Landmark Explorer**

Project Phase: Project Design & Planning (Main Folder 02)

Sub Folder: Project Planning

Date Prepared: June 20, 2025 (Reflecting a plan for June 12 - June 19, 2025)

Project Title: Gemini Landmark Explorer – An AI-Powered Multimodal Landmark Description App

TEAM:

Kamlesh Chowdhary ( [kamlesh.23bce10260@vitbhopal.ac.in](mailto:kamlesh.23bce10260@vitbhopal.ac.in) )

Somil Asati ( [somil.23bce10364@vitbhopal.ac.in](mailto:somil.23bce10364@vitbhopal.ac.in) )

Devansh Tyagi ( [devansh.23bce10247@vitbhopal.ac.in](mailto:devansh.23bce10247@vitbhopal.ac.in) )

Prashasti Joshi ( [prashasti.23bce10893@vitbhopal.ac.in](mailto:prashasti.23bce10893@vitbhopal.ac.in) )

**1. Executive Summary**

This document outlines the project plan for the Gemini Landmark Explorer application for the period of June 12th to June 19th, 2025. It details the scope, deliverables, task breakdown, conceptual timeline, required resources, and key considerations to ensure the successful development of a functional AI-powered web application.

**2. Project Scope Confirmation**

**In Scope:**

* Streamlit web application development.
* Integration with Google Gemini API for image analysis and text generation.
* Scenario-based prompt customization (Traveler, Tour Guide, Educator, Personal Explorer).
* Multilingual output translation.
* Download and session-based history features.
* Secure API key handling.
* In-memory processing for user privacy.

**Out of Scope (for this phase):**

* Real-time camera integration.
* Persistent data storage (images, user profiles).
* External knowledge base integration for factual grounding.
* Advanced features (e.g., voice output, AR/VR).

**3. Key Deliverables (by June 19th)**

1. **Functional Web Application (app.py):** Core features implemented and tested.
2. **Requirements File (requirements.txt):** Listing all project dependencies.
3. **Project Documentation Updates:** Finalized Ideation and Project Report sections relevant to the implemented features.

**4. Work Breakdown Structure (WBS) & Task List**

Below is a breakdown of major project phases and their associated tasks, allocated within the given timeline.

**Phase 1: Setup & Core Integration (June 12 - June 13)**

* **Task 1.1:** Environment setup (Python, virtual env, pip install).
* **Task 1.2:** Install core libraries (streamlit, python-dotenv, Pillow, google-generativeai).
* **Task 1.3:** Configure Google API Key (.env).
* **Task 1.4:** Initialize Streamlit app structure.
* **Task 1.5:** Implement basic Gemini API call for image processing (get\_gemini\_response, input\_image\_setup).

**Phase 2: Feature Development (June 14 - June 16)**

* **Task 2.1:** Develop image upload and display UI.
* **Task 2.2:** Implement scenario selection and dynamic prompt logic.
* **Task 2.3:** Integrate multilingual translation (googletrans, translate\_text).
* **Task 2.4:** Develop language selection UI.
* **Task 2.5:** Implement download button for descriptions.
* **Task 2.6:** Set up session history for past descriptions.

**Phase 3: Refinement & Finalization (June 17 - June 19)**

* **Task 3.1:** Comprehensive functional testing and bug fixing.
* **Task 3.2:** Refine AI prompts for clarity and accuracy across scenarios.
* **Task 3.3:** Enhance UI/UX (styling, error messages, loading indicators).
* **Task 3.4:** Finalize requirements.txt.
* **Task 3.5:** Complete and review relevant project documentation.
* **Task 3.6:** Prepare for project demonstration/submission.

**5. Conceptual Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task ID** | **Task Description** | **Start Date** | **End Date** | **Duration (Days)** |
| 1.0 | **Setup & Core Integration** | Jun 12 | Jun 13 | 2 |
| 1.1 | Environment & API Key Setup | Jun 12 | Jun 12 | 1 |
| 1.2 | Basic Gemini Integration | Jun 13 | Jun 13 | 1 |
| 2.0 | **Feature Development** | Jun 14 | Jun 16 | 3 |
| 2.1 | UI Components (Upload, Display) | Jun 14 | Jun 14 | 1 |
| 2.2 | Scenario/Prompt Logic | Jun 15 | Jun 15 | 1 |
| 2.3 | Translation Integration | Jun 16 | Jun 16 | 1 |
| 2.4 | Download & History Features | Jun 16 | Jun 16 | 0.5 |
| 3.0 | **Refinement & Finalization** | Jun 17 | Jun 19 | 3 |
| 3.1 | Testing & Bug Fixing | Jun 17 | Jun 17 | 1 |
| 3.2 | Prompt Refinement & UI Polish | Jun 18 | Jun 18 | 1 |
| 3.3 | Documentation & Submission Prep | Jun 19 | Jun 19 | 1 |

**6. Resource Allocation**

* **Human Resources:** Project Lead, Developer(s).
* **Technical Resources:** Python 3.9+, Streamlit, Google Gemini API, Pillow, googletrans, python-dotenv.
* **Hardware:** Standard development machine with stable internet access.

**7. Risk Management**

* **API Rate Limits:** Potential for delays if API calls hit limits.
* **Mitigation:** Monitor usage, efficient coding.
* **Translation Quality:** googletrans reliability can vary.
* **Mitigation:** Acknowledge limitations, focus on clear translations.
* **Timeline Pressure:** Tight schedule for feature implementation.
* **Mitigation:** Prioritize core features, minimize scope creep, efficient task management.
* **AI Output Accuracy:** Risk of occasional factual inaccuracies.
* **Mitigation:** Design clear prompts, add user disclaimers.

**8. Communication Plan**

* Daily brief check-ins (15 mins) for progress and blockers.
* Regular code commits and reviews.
* Shared document for task tracking.

**9. Evaluation Metrics**

Project success will be measured by:

* Application Functionality: All planned features working correctly.
* Performance: Response time under 10 seconds.
* Output Quality: Relevance, accuracy, coherence, and detail level of AI descriptions.

**10. Diagrams Required for Planning**

1. **Gantt Chart:** This is the primary diagram recommended for project planning.

