**Hackathon Project for the Gemini Landmark Description**

**PROJECT TITLE:**

Gemini Landmark Description

**Team Name:**

Team Techno

**Team Members:**

KANDULA AKHILA

JELLA PRAVALIKA

SAGANTI SRAVANTHI   
SIDDI MANISHA

**Phase 1: Brainstorming & Ideation**

**Objective:**

To develop an AI-powered landmark recognition system that provides real-time descriptions for users on both web and mobile platforms.

**Key Points:**

1. **Problem Statement:**

Users lack instant and reliable information about landmarks. Traditional methods require manualsearches, which can be time-consuming.

1. **Proposed Solution:**

**A** GenAI-powered web and mobile application that identifies landmarks using real-time image processing and provides detailed descriptions

1. **Target Users:**

Tourists, travelers, history enthusiasts, and students.

1. **Expected Outcome:**

A fully functional, AI-driven landmark recognition app with a user-friendly interface.

**Phase 2: Requirement Analysis**

**Objective:**

To define the technical and functional requirements necessary for the development of the application.

**Key Points:**

1. **Technical Requirements:** 
   * Programming Language: Python, JavaScript (React/Node.js)
   * Backend: Flask/Django (Python) or Node.js
   * Frontend: React.js for web, React Native for mobile
   * Database: Firebase, PostgreSQL, or MongoDB
   * AI/ML Framework: TensorFlow, OpenCV, or Google Vision API
2. **Functional Requirements:** 
   * Real-time image capture and processing.
   * AI-powered landmark recognition.
   * Detailed descriptions with historical and cultural context.
   * Multi-language support.
   * Offline mode for previously recognized landmarks.

1. **Constraints & Challenges:** 
   * Accuracy of AI model in diverse environments.
   * Performance optimization for real-time processing.
   * Handling large-scale landmark data efficiently.

**Phase 3: Project Design**

**Objective:**

To create a scalable system architecture and design a user-friendly interface.

**Key Points:**

1. System Architecture:
   * Client-Server model with cloud-based AI processing.
   * API integration for data retrieval.
   * Scalable database for storing landmark information.
2. **User Flow:**

User captures or uploads an image → AI processes the image → Identifies the landmark → Displays real-time description and additional details.

1. **UI/UX Considerations:** 
   * Intuitive and responsive design for both mobile and web.
   * Clean and engaging UI with interactive features.
   * Dark mode and accessibility options.

**Phase 4: Project Planning**

**Objective:**

Sprint-based development planning with clear milestones.

**Sprint Planning with Priorities**

**Sprint 1 – Setup & Integration**

High Priority: Set up the environment and install dependencies.

High Priority: Integrate Google Vision API for landmark recognition.

Medium Priority: Build a basic UI with input fields.

**Sprint 2 – Core Features & Debugging**

High Priority: Train and fine-tune the landmark recognition model.

High Priority: Optimize API responses for real-time landmark descriptions.

**Sprint 3 – Testing, Enhancements & Submission**

Improve UI design for better user experience.

Low Priority: Final demo preparation & deployment.

**Phase 5: Project Development**

* Develop AI-based landmark recognition system.
* Implement frontend and backend functionalities.
* Integrate real-time data fetching APIs.
* Test for performance and accuracy.
* Deploy and gather user feedback for improvements.

**Final Submission**

**1.Projecr Report Based on templates**

**2.GitHub/Repository Link**