Hackathon Project Phases Template

Project Title: Gemini Landmark Description App: Enhancing Tourist Experience with AI

Team Name:
(Wander AI)

Team Members:

- Deepa Soni
- Prashanth
- Khushi Yadav
- Akash

Phase-1: Brainstorming & Ideation

Objective:
Develop an AI-powered landmark description app using Gemini AI to enhance the tourist experience by providing real-time, AI-generated descriptions and historical insights of landmarks.

Key Points:

1.

Problem Statement:

- Tourists often struggle to find accurate and engaging information about landmarks.
- Traditional guidebooks and static websites may lack real-time updates and interactive experiences.

2.

Proposed Solution:

- An AI-powered application using Gemini AI to provide real-time descriptions of landmarks.
- The app delivers historical insights, fun facts, and cultural significance in multiple languages.
- Integration with AR for an immersive experience.

3.

Target Users:

- Tourists looking for instant landmark descriptions.
- Travel enthusiasts interested in rich, AI-generated content.
- Students and researchers studying historical and cultural sites.

4.

Expected Outcome:

- A functional AI-powered landmark description app that enhances the tourism experience with real-time data and an interactive UI.

Phase-2: Requirement Analysis

Objective:
Define the technical and functional requirements for the Gemini Landmark Description App.

Key Points:

1. **Technical Requirements:**

   - **Programming Language:** Python / JavaScript
   - **Backend:** Google Gemini AI API
   - **Frontend:** React.js / Next.js
   - **Database:** Firebase / MongoDB
   - **AR Integration:** WebAR / AR.js (if applicable)

2. **Functional Requirements:**

   - Ability to fetch landmark details using Gemini AI.
   - Display descriptions, historical facts, and cultural significance.
   - Support multiple languages for a diverse tourist base.
   - Provide a visually engaging UI with images and interactive maps.

3. **Constraints & Challenges:**

- Ensuring accurate and relevant AI-generated descriptions.
- Managing API rate limits and optimizing API calls.
- Delivering a seamless AR experience (if implemented).

Phase-3: Project Design

Objective:
Develop the architecture and user flow of the application.

Key Points:

1. **System Architecture:**

   - User enters a landmark-related query via UI.
   - Query is processed using Google Gemini AI.
   - AI model fetches and processes the data.
   - The frontend displays landmark descriptions, historical insights, and images.

2. **User Flow:**

   - **Step 1:** User enters a landmark name or scans a QR code.

- **Step 2:** The backend calls the Gemini AI API to retrieve information.
- **Step 3:** The app processes the data and displays results in an easy-to-read format.
- **Step 4:** Users can interact with AR features for an immersive experience.

3.

UI/UX Considerations:

- Minimalist, user-friendly interface for seamless navigation.
- Interactive elements for engaging content consumption.
- Light and dark mode for better accessibility.

Phase-4: Project Planning (Agile Methodologies)

Objective:
Break down development tasks for efficient completion.

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

- (🔴 High Priority) Set up the environment & install dependencies.
- (🔴 High Priority) Integrate Google Gemini AI API.
- (🟡 Medium Priority) Build a basic UI with input fields.

Sprint 2 – Core Features & Debugging (Day 2)

- (🔴 High Priority) Implement search & description functionalities.
- (🔴 High Priority) Debug API issues & handle errors in queries.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

- (🟡 Medium Priority) Test API responses, refine UI, & fix UI bugs.
- (🟢 Low Priority) Final demo preparation & deployment.

Phase-5: Project Development

Objective:
Implement core features of the Gemini Landmark Description App.

Key Points:

1. **Technology Stack Used:**

- **Frontend:** React.js / Next.js
- **Backend:** Google Gemini AI API
- **Database:** Firebase / MongoDB
- **AR Integration:** WebAR / AR.js (if applicable)

2. **Development Process:**

- Implement API key authentication and Gemini AI integration.
- Develop AI-based description and historical insights logic.
- Optimize search queries for performance and relevance.

3. **Challenges & Fixes:**

- **Challenge:** Delayed API response times.
  Fix: Implement caching to store frequently queried results.
- **Challenge:** Limited API calls per minute.
  Fix: Optimize queries to fetch only necessary data.

Phase-6: Functional & Performance Testing

Objective:
Ensure that the Gemini Landmark Description App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status
TC-001	Functional Testing	Query "Taj Mahal description"	AI-generated landmark details	✅ Passed
TC-002	Functional Testing	Query "Historical facts about Eiffel Tower"	Relevant historical insights	✅ Passed
TC-003	Performance	API response time under 500ms	Quick data retrieval	❌ Needs Optimization

	Testing			
TC-004	Bug Fixes	Fixed incorrect AI responses	Improved data accuracy	✅ Fixed
TC-005	Final Validation	Ensure UI is responsive across devices	UI should work on mobile & desktop	❌ Failed - UI broken on mobile
TC-006	Deployment Testing	Host the app using Vercel / Firebase Hosting	App should be accessible online	✅ Deployed

---

Final Submission

1. Project Report Based on the templates
2. Demo Video (3-5 Minutes)
3. GitHub/Code Repository Link
4. Presentation

This document is structured to match the hackathon template while being tailored to your project. Let me know if you need modifications or additions! 🚀