Hackathon Project Phases Template Gemini landmark description app enhancing tourist experiencees with AI

Hackathon Project Phases Template

Project Title: Gemini landmark description app enhancing tourist experiencees with AI

Team Name: VistaGuide AI

Team Members:

- Gorile Geetha
- Jeevana Choppadandi
- Divya Bharti
- Shatakshi Dudala

Phase-1: Brainstorming & Ideation

Objective:

To develop an AI-powered landmark description app that enhances tourist experiences by providing real-time, interactive, and multilingual insights on historical and cultural sites, improving accessibility, engagement, and local business growth.

Key Points:

1. **Problem Statement:**
   - Tourists often struggle with accessing accurate, real-time, and engaging information about landmarks they visit. Traditional guidebooks, static information plaques, and generic tour apps fail to provide interactive, personalized, and immersive experiences. Additionally, language barriers, outdated content, and the lack of AI-driven assistance

make it difficult for travelers to fully appreciate historical, cultural, and architectural significance.

- A Gemini-powered landmark description app aims to enhance the tourist experience by leveraging AI to provide real-time, interactive, and personalized landmark descriptions, audio guides, AR overlays, and multilingual support. This solution addresses the limitations of traditional methods, offering a smarter and more engaging way for tourists to explore and learn.

2. **Proposed Solution:**

    - Utilize AI to generate detailed, real-time descriptions of landmarks, including historical facts, cultural significance, and visitor tips**.**
    - Implement AI-driven language translation to cater to a global audience and enhance accessibility.
    - Use AI to suggest nearby attractions, restaurants, and events based on user preferences and location.
    - Collaborate with local businesses and tourism boards to promote attractions and services, creating a sustainable tourism ecosystem.

3. **Target Users:**

    - **Tourists & Travelers** – Individuals exploring new destinations who want detailed, real-time information about landmarks.
    - **Students & Researchers** – Academics studying history, architecture, or cultural heritage.
    - **Local Businesses & Vendors** – Shops, restaurants, and hotels that can be promoted through AI-driven recommendations.
    - □ **Local & International Tourists** – Both domestic and foreign visitors seeking multilingual and AI-powered descriptions.

4. **Expected Outcome:**

- **Better Tourist Experience** – Visitors get real-time AI-powered information about landmarks.
- **Easy Access for Everyone** – Multilingual and voice support make it user-friendly.

Phase-2: Requirement Analysis

Objective:

The objective of the **requirement analysis** for the Gemini Landmark Description App is to define the functional and non-functional needs of the system to enhance tourist experiences using AI..

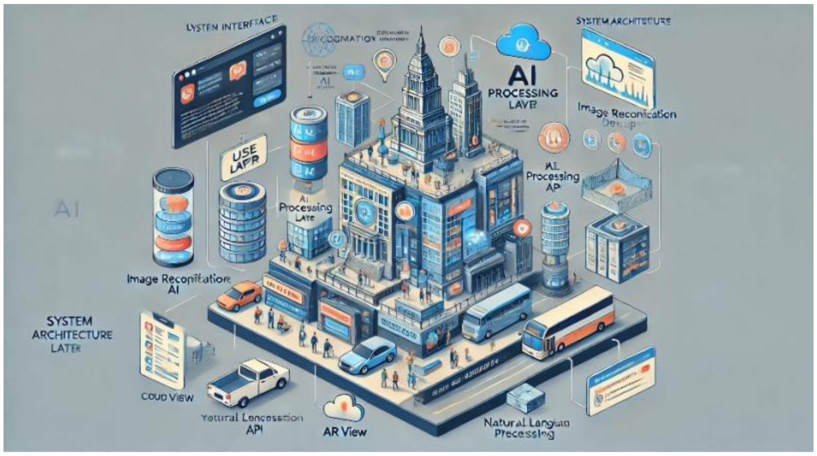
Key Points:

1. **Technical Requirements:**
    - Programming Language: **Python**
    - Backend: **Tkinter**
    - Frontend: **React Native**
    - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
    - **AI-Powered Landmark Recognition** – Users can take a photo or use GPS to identify landmarks.
    - **Real-Time Information** – Provides historical facts, fun trivia, and cultural significance of landmarks.
    - **Multilingual Support** – Offers descriptions in multiple languages for global tourists.
    - **Personalized Recommendations** – Suggests nearby landmarks based on user interests.
3. **Constraints & Challenges:**
    - Ensuring real-time updates from **Gemini API**.
    - Handling **API rate limits** and optimizing API calls.
    - Providing a **smooth UI experience** with Streamlit.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - **Mobile App (iOS & Android)** – Provides an intuitive interface for users.
    - **Natural Language Processing (NLP)** – Generates human-like descriptions of landmarks.
    - **Content Management System (CMS)** – Allows updates to landmark descriptions and user-generated content.
    - **Data Encryption (AES-256)** – Ensures secure storage and transmission of user data.
    - **Crowdsourced Landmark Updates** – Users can contribute missing details, verified by AI.
2. **User Flow:**
    - Step 1: App Launch & Onboarding.
    - Step 2: Landmark Discovery & Exploration
    - Step 3: AI-Powered Landmark Description
    - Step 4: Augmented Reality (AR) Exploration\
    - Step 5: AI Chatbot Assistance
    - Step 6: Social Engagement & Sharing
    - Step 7: Saving & Offline Mode.
    - Step 8: Exit & Feedback
3. **UI/UX Considerations:**
    - **Multi-Language Support** – Allows tourists to access information in their preferred language.
    - **Adaptive UI** – Works seamlessly across different screen sizes (phones, tablets).
    - **User-Generated Content** – Enables reviews, ratings, and photo uploads.
    - .**Data Encryption** – Protects personal and location data.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	🔴 High	6 hours (Day 1)	End of Day 1	G.Geetha	Google API Key, Python, Streamlit setup	API connection established & working

Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	CH.Jeevana	API response format finalized	Basic UI with input fields
Sprint 2	Vehicle Search & Comparison	🔴 High	3 hours (Day 2)	Mid-Day 2	Divya Bharti	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	D.Shatakshi	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	G.Geetha & CH.Jeevana	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Demo-ready project

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

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

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 High Priority) Implement **search & comparison 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 enhancing tourist experiencees with ai.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Google Gemini Flash API
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Gemini API integration**.
    - 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 enhancing tourist experiencees with ai as expected .

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Search for any landmark."	App finds landmark and displays an information.	✅ Passed	Tester 1
TC-002	Functional Testing	Query " what is the history of a landmark "	App provide historical information about the searched landmark	✅ Passed	Tester 2
TC-003	Performance Testing	Time to display description<2sec.	API should return results quickly.	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Tester 2

TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	DevOps

Final Submission

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