Hackathon Project Phases Template

Project Title: Gemini Landmark Description App Enhancing Tourists Experiences with AI

Team Name: TOUR TECHIES

TOUR TECHIES

Bring The Happiness

Team Members:

- CHALLA BHAVYA (24P61A0570/CSE)
- CHILLAR HAMSIKA(24P61A0568/CSE)
- DEVARASHETTY PALLAVI (24P61A0589/CSE)
- DANNINA BHAVYA SREE (24P61A0584/CSE)

Phase-1: Brainstorming & Ideation

Objective:

The Gemini Landmark Explorer is designed to revolutionize tourism by harnessing AI-driven insights, offering travelers immersive, context-rich, and personalized narratives about historical landmarks, cultural treasures, and iconic attractions worldwide.

Key Points:

1.Problem Statement:

AI-Driven Landmark Discovery – Users can upload images or enter prompts to receive AI-curated, immersive descriptions.

Deep Cultural Insights – Unveils historical narratives, architectural brilliance, and hidden stories of landmarks.

Transforming Travel Experiences – A must-have for tourists, guides, and history enthusiasts.

Global Connectivity – Bridges cultures by offering rich knowledge of landmarks across continents.

Multilingual Intelligence – Delivers descriptions in multiple languages for seamless accessibility.

Inclusive Exploration – Designed with accessibility features to cater to users of all abilities.

Instant Knowledge Hub – Provides on-the-go, real-time landmark insights anytime, anywhere.

2. Proposed Solution:

- A cutting-edge AI-powered app leveraging Gemini Flash, offering instant, dynamic landmark descriptions and historical context.

- **Interactive Exploration –** Users can engage with AI-powered storytelling, 3D reconstructions, and augmented reality overlays.

- **Smart Recommendations –** The app personalizes insights based on user preferences and past explorations.

- **Offline Mode –** Enables access to essential details even in low-connectivity areas.

3. **Target Users:**

Curious Travelers – Seeking rich, AI-enhanced exploration of cultural sites.

Tour Guides & Historians – Leveraging AI for detailed, engaging storytelling.

Students & Researchers – Accessing a vast historical and architectural knowledge base.

Globetrotting Enthusiasts – Discovering the world's wonders in an innovative way

4. **Expected Outcome:**

A **next-gen AI-powered travel companion** that transforms landmark visits into immersive, educational, and personalized experiences.

A **global knowledge bridge,** enhancing cultural appreciation through smart AI insights.

A user-friendly, inclusive, and multilingual platform making history and heritage accessible to all.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Gemini landmark description app enhancing tourist experiences with AI.

Key Points:

1. User Requirements

Target Users: Tourists, history enthusiasts, students, and travel bloggers.

User Needs:

Easy access to detailed landmark information.

AI-generated descriptions in multiple languages.

Offline accessibility for remote locations.

Personalized recommendations for travel routes and landmarks.

AR-based interactive experiences.

Social sharing features for trip memories.

2. Functional Requirements

- **User Authentication & Profiles**

Signup/Login via email, Google, or social media.

Profile preferences (language, interests, accessibility needs).

- **Landmark Identification & Description**

AI-powered image and GPS-based landmark recognition.

Text and voice-based descriptions of landmarks.

- **AI Chatbot Assistance**

AI-driven chatbot answering landmark-related queries.

Context-aware suggestions based on user preferences.

- **Augmented Reality (AR) Experience**

AR overlays providing historical and cultural insights.

- **Itinerary & Recommendation System**

Personalized suggestions based on user behavior and location.

Travel routes with estimated time and accessibility information.

- **Offline Mode**

Downloadable landmark descriptions for offline access.

- **Social Features**

Users can share experiences, reviews, and AI-generated content.

- **Accessibility Features**

Voice narration, text-to-speech, adjustable font sizes.

3. Non-Functional Requirements

- **Performance**

Fast AI processing for real-time landmark recognition.

Low-latency responses for chatbot interactions.

- **Scalability**

Cloud-based architecture supporting high user traffic.

AI model optimization for efficient processing.

- **Usability**

Intuitive UI/UX with easy navigation.

Multilingual support for global users.

- **Security**

Secure user authentication (OAuth 2.0, JWT).

End-to-end encryption for user data and chat interactions.

- **Reliability & Availability**

99.9% uptime with cloud-based redundancy.

4.**System Requirements**

Hardware Requirements

User Device: Android/iOS smartphone with AR capability.

Server: Cloud-based infrastructure (Google Cloud, AWS, or Azure).

Software Requirements

Mobile App: React Native / Flutter.

Backend: Node.js / FastAPI (Python).

Database: PostgreSQL / Firebase Firestore.

AI Processing: Google Gemini AI / OpenAI API / Google Vision API.

Mapping: Google Maps API for navigation.

5.Data Requirements

- **Landmark Database**

Historical, cultural, and architectural details of landmarks.

Geospatial data for GPS-based recognition.

- **User Data**

Preferences, travel history, and frequently visited places.

- **AI Training Data**

Large-scale image datasets for accurate landmark recognition.

- **Real-Time Data**

Live weather, opening hours, and local event updates.

6. AI-Specific Requirements

1. AI Model Selection

Image Recognition: Google Vision API / Custom-trained CNN model.

NLP for Chatbot: Google Gemini AI / OpenAI GPT models.

Recommendation System: Collaborative filtering & ML-based ranking.

2. Model Optimization

Cloud-based inference for real-time results.

Edge AI support for offline processing.

3. AI Training & Testing

Continuous learning from user feedback to improve descriptions.

7. Legal & Ethical Considerations

1. Data Privacy

Compliance with GDPR, CCPA, and other regional regulations.

User consent for AI-based data processing.

2. Bias & Fairness

AI-generated content must be historically accurate and unbiased.

Diverse dataset training to avoid regional or cultural biases.

3. User Safety

Warnings for restricted or dangerous areas.

8. Monetization Strategy

- **Freemium Model**

Free access to basic landmark descriptions.

Premium subscription for AR experiences, offline mode, and AI chatbot assistance.

- **In-App Purchases**

Exclusive historical insights or guided AR tours.

- **Advertising & Partnerships**

Sponsored content from local businesses (restaurants, hotels, travel agencies).

Paid promotions for premium travel packages.

- **Affiliate Marketing**

Referral commissions from booking platforms (Expedia, Airbnb, etc.).

9. Testing & Validation

Testing Strategy

- **Unit Testing**

Ensuring API responses, AI model accuracy, and database queries function correctly.

- **Performance Testing**

Load testing for high user traffic.

AI inference speed testing for real-time processing.

- **Usability Testing**

A/B testing for UI/UX improvements.

Feedback collection from beta users.

- **Security Testing**

Data encryption verification.

Penetration testing to identify vulnerabilities.

- **AI Model Evaluation**

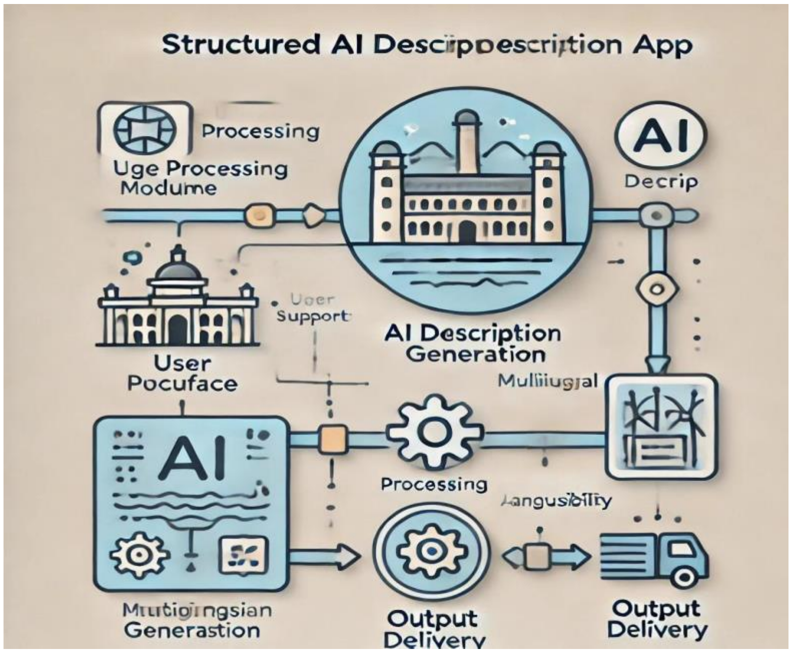
Continuous learning updates based on user feedback.

Validation against real-world landmark datasets.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
   - User enters input as image
   - Query is processed using **Google collab**.
   - AI model fetches and processes the data.
   - The frontend displays **location map,sign up and description space.**
2. **User Flow:**
   - Step 1: User uploads an image (e.g., “any place or destination”).
   - Step 2: The backend **calls the google collab** to retrieve landmark description.
   - Step 3: The app processes the data and **displays results** in an easy-to-read format.
3. **UI/UX Considerations:**
   - **Minimalist, user-friendly interface** for seamless navigation.
   - **Filters for price, mileage, and features**.
   - **Dark & light mode** for better user experience.

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	Bhavya	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	Medium	1 hour (Day 1)	End of Day 1	Pallavi	API response format finalized	Basic UI with input fields
Sprint 2	Search & Comparison	🔴 High	3 hours (Day 2)	Mid-Day 2	Bhavya sree	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	4 hours (Day 2)	Mid-Day 2	Pallavi hamsika	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Bhavya	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 AutoSage App.

Key Points:

1. **Technology Stack Used:**
   - **Frontend:** 10Web
   - **Backend:** google collab
   - **Programming Language:** Python
2. **Development Process:**
   - Implement **API key authentication** and **Gemini API integration**.
   - Develop **landmark locaters and maintenance tips 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 AutoSage App works as expected.

Test	Category	Test Scenario	Expected Outcome	Status	Tester

Case ID					
TC-001	Functional Testing	Query "Best budget cars under ₹10 lakh"	Relevant budget cars should be displayed.	✅ Passed	Tester 1
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	✅ Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	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**