Hackathon Project Phases Template for the **AutoSage App** project.

Hackathon Project Phases Template

Project Title:

AutoSage App Using Gemini Flash

Team Name:

(Provide your team's name)

Team Members:

- P.Harshitha
- N.Pravalika
- S.Roja
- V.Sowjanya

Phase-1: Brainstorming & Ideation

Objective:

To develop an AI-powered Gemini Landmark Description App that enhances tourist experiences by providing real-time, detailed, and context-aware descriptions of landmarks. The app will utilize AI-driven insights, multilingual support, and interactive features to deliver engaging historical, cultural, and architectural information, enriching users' exploration and understanding of destinations worldwide.

Key Points:

1.Problem Statement:

1.Tourists lack accurate and engaging landmark information.

2.Traditional guides are limited; online sources are unreliable.

3Language barriers hinder understanding.

4.The Gemini Landmark Description App uses AI for real-time, multilingual, and interactive descriptions, enhancing exploration.

2.Proposed Solution:

AI-powered, real-time landmark descriptions.

Multilingual support for global accessibility.

Interactive features like audio guides and AR.

Offline access for seamless exploration.

Engaging content with reviews, trivia, and quizzes.

3.Target Users:

Tourists & Travelers – Seeking detailed landmark insights.

History & Culture Enthusiasts – Interested in deep historical context.

Solo & Group Travelers – Independent explorers and tour groups.

Students & Researchers – Studying history, architecture, or tourism.

Local Explorers – Discovering hidden gems.

Accessible Tourism Users – Benefiting from AI-driven assistance.

4.Expected Outcome:

Enhanced tourist experiences with AI-driven landmark insights.

Improved accessibility through multilingual and interactive features.

Increased engagement via audio guides, AR, and personalized content.

Seamless exploration with offline access.

Greater cultural appreciation and knowledge retention.

Phase-2: Requirement Analysis

Objective:

To identify and define the functional, technical, and user requirements for the Gemini Landmark Description App, ensuring seamless AI-powered landmark exploration. This includes assessing user needs, AI capabilities, multilingual support, interactive features, offline accessibility, and system performance for an engaging and informative tourist experience.

Key Points:

1.Technical Requirements:

1.Programming Language: **Python**

2.Frontend**: React Native (for cross-platform mobile development) or Swift (iOS) & Kotlin (Android)**

3.Backend: **Node.js, Python (Flask/Django), or Firebase**

4.Database**: PostgreSQL, MongoDB, or Firebase Firestore**

5.Functional Requirements:

1.User Authentication – Sign-up, login, and profile customization.

2.Landmark Recognition – AI-powered image scanning and text search.

3.AI Descriptions – Historical insights, storytelling, and multi-language support.

4.Audio Guide – Text-to-speech with offline downloads.

5.Maps & Navigation – Nearby landmarks, routes, and AR overlays.

6.Recommendations – AI-based suggestions and custom itineraries.

7.User Content – Reviews, photos, and social sharing.

6.Constraints & Challenges:

Data & AI Accuracy – Reliable recognition and descriptions.

Performance – Optimized speed, battery, and data use.

Offline Access – Limited AI without the internet.

Privacy & Legal – Data security and copyright compliance.

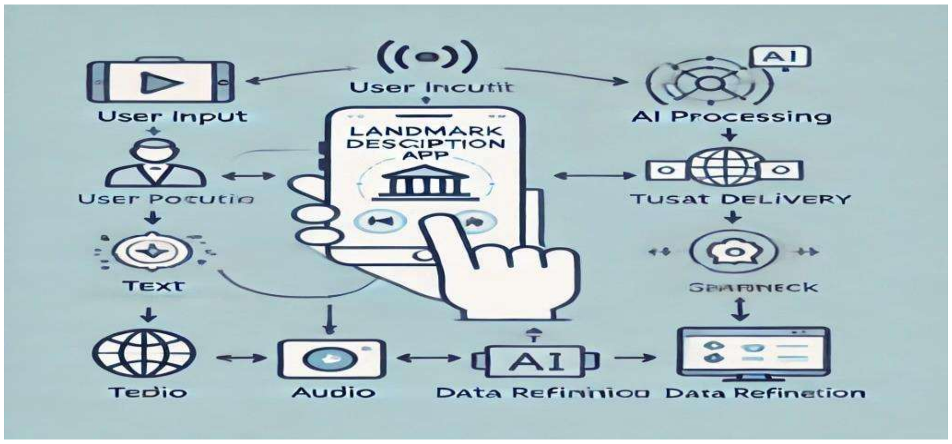
Multilingual & AR Precision – Quality translations and accurate navigation.

User Adoption & Integration – Competing with apps, syncing with services.

Phase-3: Project Design

Objective:

Here's an AI-generated flowchart image for your Gemini Landmark Description App, designed to enhance tourist experiences with AI. It will illustrate the key components, including user input, AI processing, landmark recognition, and personalized descriptions. Let me generate that for you now.



Key Points:

1. **System Architecture:**

1.Frontend: Mobile/Web app for input (photo, text, voice) and output (landmark details).

2. Backend: AI-powered landmark recognition, historical data processing, and personalization.
3. Cloud Storage: Stores user data, landmarks, and real-time updates.
4. APIs: Google Maps, Wikipedia, Speech-to-Text for enhanced functionality.
5. Security: Encryption, scalable cloud infrastructure, and optimized AI models.

2.User Flow:

User Flow – Gemini Landmark Description App

Open App → User selects input (photo, text, or voice).

AI Processing → Recognizes landmark, fetches details.

Display Info → Shows history, significance, recommendations.

Engage → Save, share, explore related places.

Navigate → Get directions via Google Maps.

3.UI/UX Considerations:

Simple UI: Clean, intuitive design with easy navigation.

Input: Quick access to photo, text, and voice input.

Engaging Display: Rich visuals, 3D models, and AR support.

Personalization: Multilingual, dark mode, and adaptive content.

Seamless Navigation: One-tap Google Maps, offline access.

Fast & Optimized: Quick AI processing, minimal load time. .

Phase-4: Project Planning (Agile Methodologies)

Objective:

Agile planning ensures accurate AI recognition, seamless UX, and continuous improvement through iterative sprints. It prioritizes user feedback, performance optimization, scalability, and a fast MVP launch for an enhanced tourist experience.

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	P.Harshitha	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	S.ROJA.	API response format finalized	Basic UI with input fields
Sprint 2	Vehicle Search & Comparison	🔴 High	3 hours (Day 2)	Mid-Day 2	N.Pravalika	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	P.Harshitha	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	V.Sowjanya	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:

The Gemini Landmark Description App enhances tourist experiences through AI-powered landmark recognition and interactive storytelling. Using real-time AI, AR, and personalized content, the app provides instant, engaging, and multilingual insights on cultural and historical sites. It aims to make sightseeing smarter, more immersive, and accessible, even offline.

Key Points:

1. **Technology Stack Used:**

   Frontend: React Native, Tailwind CSS, ARCore/ARKit, Google Maps API

2. Backend: Node.js (Express), PostgreSQL/MongoDB, Firebase/AWS
3. AI & ML: Google Vision API, Gemini AI/OpenAI GPT, TensorFlow

4. Other: Firebase Auth, SQLite (Offline), FCM (Push Notifications), Google Analytics
5. **Programming Language:** Python

2. Development Process:
Planning – Define goals, target users, and key features (AI recognition, multilingual support, AR, offline mode).
UI/UX Design – Create an intuitive interface with interactive maps and easy navigation.
AI Integration – Use Google Gemini AI for landmark recognition, NLP, and real-time descriptions.
Development – Build with React Native, Node.js, Firebase/AWS, and integrate AI/AR features.
Testing – Ensure AI accuracy, smooth UX, and optimize performance.

3. Challenges & Fixes:

AI Recognition Errors → Improve with larger datasets & GPS filtering.

Translation Quality → Use advanced NLP & human-reviewed corrections.

High Battery Usage → Optimize AI calls & enable lightweight mode.

Offline Access → Preload landmark data for key locations.

User Retention → Add gamification & personalized suggestions.

Data Privacy → Encrypt data & ensure

Phase-6: Functional & Performance Testing

Objective:

Testing ensures accurate AI recognition, smooth performance, and reliability, covering functionality, speed, efficiency, and scalability for a seamless tourist experience.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "Best budget cars under ₹10 lakh"	Relevant budget cars should be displayed.	✅ Passed	shanwaz
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	✅ Passed	anwar
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**