Hackathon Project Phases Template for the **Landmark Lens** project.

Hackathon Project Phases Template

Project Title:

Landmark Lens

Team Name:

AI-TISTS

Team Members:

- U.Vasudha
- V.Pranavi
- S.M.Ayesha Banu
- M.Nikitha

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered website that enhances tourist experiences by providing real-time landmark descriptions, historical insights, and cultural significance using image recognition.

Key Points:

1. **Problem Statement:**

   Tourists often struggle to find reliable and detailed information about landmarks, historical sites, and cultural significance while traveling.
   Travelers need a quick and interactive way to access landmark insights without relying on multiple sources.

2. **Proposed Solution:**

An AI-powered web application using Gemini Vision Pro to recognize landmarks from images and provide real-time historical, cultural, and travel-related information.

Integration of an AI-powered vehicle expert tool to help travelers make informed transportation choices, including eco-friendly options.

3. **Target Users:**

Tourists seeking detailed and reliable landmark descriptions.
Travel enthusiasts looking for historical and cultural insights.
Travelers needing guidance on trip.

4. **Expected Outcome:**

A functional AI-powered travel companion that enhances tourism experiences by providing instant landmark descriptions, historical insights, and travel recommendations.

Seamless trip planning support to help travelers explore destinations efficiently and meaningfully.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Landmark Lens website

Key Points:

1. **Technical Requirements:**

Programming Language: **Python**
Backend: **Google Gemini Pro Vision API**
Frontend: **HTML, CSS, JavaScript with Bootstrap**
Database: **Not required initially (API-based queries)**

2. **Functional Requirements:**

Ability to recognize landmarks using the Gemini Vision Pro API and provide detailed descriptions.
Display historical facts, cultural significance, and travel tips in an intuitive UI.
Provide personalized trip recommendations, including nearby attractions and best visiting times.

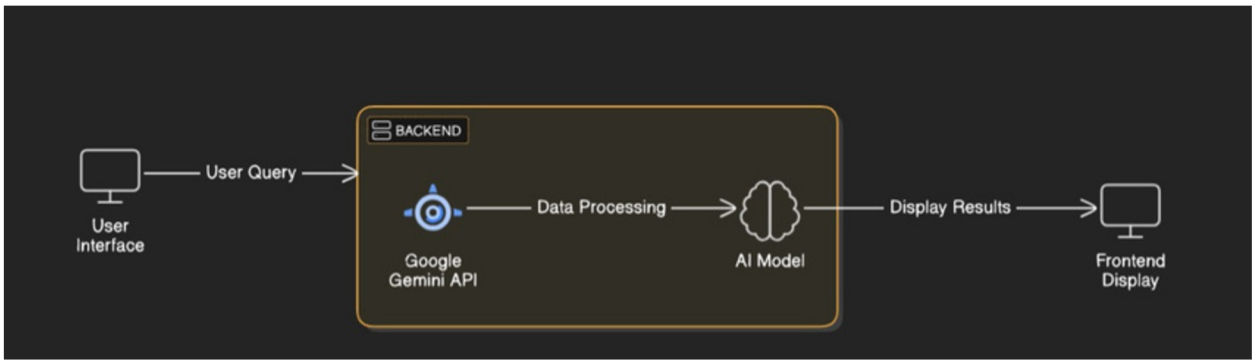
3. **Constraints & Challenges:**

Ensuring real-time updates from **Gemini API**.
Handling **API rate limits** and optimizing API calls.
Providing a smooth UI experience with HTML, CSS, JavaScript, and Bootstrap.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   User uploads an image and enters a landmark-related query via the UI.
   Query is processed using the Google Gemini Vision Pro API.
   AI model fetches and processes landmark details.
   The frontend displays results according to the query asked.

2. **User Flow:**

   Step 1: User uploads an image of a landmark and enters a related query (e.g., "History of this monument").
   Step 2: The backend calls the Gemini Vision Pro API to analyze the image and retrieve relevant landmark data.
   Step 3: The app processes the data and displays responses specifically tailored to the user's query, providing historical facts, cultural significance, travel tips, or other relevant information.

3. **UI/UX Considerations:**

   Minimalist, user-friendly interface for seamless navigation.
   Search options for landmarks based on image, and user query.
   Clear and structured presentation of landmark details for an enhanced 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	Nikitha	Google API Key, Python, setup API	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Pranavi & Ayesha	response format finalized	Basic UI with input fields
Sprint 2	Landmark search & Description	🔴 High	3 hours (Day 2)	Mid-Day 2	Vasudha & Nikitha	API response, UI elements ready	Search functionality with image
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Vasudha	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Entire Team	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 development environment & install dependencies.
🔴 High Priority – Integrate Google Gemini Vision Pro API for landmark recognition.
🟡 Medium Priority – Build a basic UI with input fields for image upload & text queries.

Sprint 2 – Core Features & Debugging (Day 2)

🔴 High Priority – Implement landmark search and comparison functionalities.
🔴 High Priority – Debug API issues & handle errors for invalid queries or unrecognized images.

Sprint 3 – Testing, Enhancements & Deployment (Day 3)

🟡 Medium Priority – Test API responses, refine UI, & fix bugs related to search and display.
🟢 Low Priority – Prepare for the final demo and deploy the app online for user access.

Phase-5: Project Development

Objective:

Implement core features of the Landmark Lens.

Key Points:

1. **Technology Stack Used:**

   Frontend: HTML,CSS ,JavaScript with Bootstrap
   Backend: Google Gemini Vision Pro API
   Programming Language: Python

2. **Development Process:**

   Implement **API key authentication** and **Gemini API integration**.
   Develop **landmark recognition with user related user query**.
   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 Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Query "History of Taj Mahal"	Detailed historical facts should be displayed.	✅ Passed	Tester 1
TC-002	Functional Testing	Query "Best time to visit Eiffel Tower"	Correct seasonal visiting 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 landmark descriptions.	Data accuracy should be improved.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on desktop.	Passed - UI on desktop	Tester 2
TC-006	Deployment Testing	Host the website using web server	Website 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**