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

Hackathon Project Phases Template

Project Title:

Gemini Landmark Description App Enhancing Tourist Experiences with AI

Team Name:

Spirits

Team Members:

S.Balraju

A.Sumedh

S.Dhanush

K.Bhaskar

P.Sachin

Phase-1: Brainstorming & Ideation

Objective:

Provide Real-time Information and Descriptions:

- Objective: Develop an AI-driven app that provides tourists with real-time descriptions of landmarks, monuments, or cultural sites.

- How: By using location-based services, the app can detect a tourist's position and offer personalized audio/text descriptions, historical facts, and interesting anecdotes about nearby landmarks.

Key Points:

1. **Problem Statement:**

· **Problem Statement:** Tourists often lack detailed, engaging, and contextually rich information when visiting landmarks, leading to a less immersive and educational experience.

· **Solution:** An AI-powered app could provide real-time, dynamic, and interactive descriptions of landmarks, enhancing the overall tourist experience by offering rich historical and cultural context

2. **Proposed Solution:**

- **Solution:** Utilize AI-driven natural language processing (NLP) to provide real-time, dynamic descriptions of landmarks and tourist attractions based on the user's location.
- **How It Works:**
    - The app uses GPS and geolocation to detect the user's proximity to nearby landmarks.
    - AI algorithms analyze the landmark's historical significance, cultural context, and related stories, offering engaging and informative text or audio descriptions.
    - The information will be presented in an engaging way, blending factual data with interesting anecdotes, ensuring tourists learn more than just the basics.

3.**Target Users:**

- The **Gemini Landmark Description App** is designed to serve a broad range of users, each with unique needs, including international tourists, locals, families, elderly or disabled individuals, cultural enthusiasts, students, virtual tourists, and even professional guides. By offering tailored, engaging, and informative experiences for each group, the app ensures a personalized and enriching visit for ever like.

4.**Expected outcomes:**

- The Gemini Landmark Description App will make visiting landmarks more interactive, educational, and enjoyable for tourists, leading to greater satisfaction and increased engagement. It will drive a deeper connection to cultural heritage, foster inclusive tourism, and support more sustainable travel practices. Furthermore, it will provide valuable insights for tourism management, benefiting both tourists and local stakeholders alike.

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. **Technical Requirements:**

· **Frontend:** React Native (for cross-platform mobile app development), Swift (for iOS), Kotlin (for Android).

· **Backend:** Node.js, Python, or Ruby on Rails for server-side logic.

· **Database:** MongoDB, PostgreSQL, or Firebase for user data storage and content management.

· **Cloud:** AWS, Google Cloud, or Azure for hosting, storage, and analytics.

- **AI & ML:** TensorFlow, OpenAI API (for NLP), or Microsoft Azure AI for personalized recommendations and language processing.
- **AR:** ARCore (Android) and ARKit (iOS) for augmented reality features.
    - **Analytics:** Firebase Analytics, Google Analytics, or Mixpanel for user behavior tracking.

2. **Functional Requirements:**

Here' s a **short version** of the **functional requirements** for the **Gemini Landmark Description App**:

1. **User Registration & Login:** Account creation, secure login, password recovery, and profile management.
2. **Geolocation & Mapping:** Real-time GPS tracking, nearby landmark recommendations, route mapping, and offline map support.
3. **Landmark Information:** Detailed descriptions, multimedia content, text-to-speech, multilingual support, and search functionality.
4. **Personalized Recommendations:** Custom itineraries and AI-powered suggestions based on user preferences.
5. **Augmented Reality (AR):** AR overlays, interactive elements, and virtual tours.
6. **Offline Access:** Downloadable content, offline maps, and audio guides.
7. **Push Notifications:** Alerts for events, updates, and personalized recommendations.
8. **User Feedback:** Ratings, reviews, and content feedback.
9. **Social Media Sharing:** Share landmarks, itineraries, and achievements on social media.
10. **Group Tours & Tour Guides:** Support for group features, custom itineraries, and in-app bookings for tours.
11. **Gamification:** Challenges, badges, and rewards for user engagement.
12. **Analytics & Insights:** User behavior tracking, crowd density info, and usage analytics.
13. **Payment Integration:** In-app purchases, subscriptions, and payment gateways for booking tours.

14. **Admin Panel**: Content and user management, analytics dashboard, and content updates.

3. **Constraints & Challenges:**

Here's a **short version** of the **constraints and challenges** for the **Gemini Landmark Description App**:

Technological Constraints: Ensuring compatibility with various devices and providing smooth AR/VR and offline functionality.

Data Availability: Collecting accurate, up-to-date landmark data and real-time updates for events and crowd information.

Privacy & Security: Safeguarding user data and ensuring compliance with privacy laws (e.g., GDPR).

User Engagement: Maintaining user interest and offering personalized experiences for diverse tourist needs.

Scalability: Managing performance and server load as the app grows with increasing users and content.

Third-Party Integration: Coordinating with tourism boards, mapping services, and payment systems, which may present compatibility challenges.

Content Management: Keeping content fresh and moderating user-generated content.

Cultural Sensitivity: Providing accurate, respectful, and culturally sensitive information.

Environmental & Connectivity: Handling low connectivity areas and minimizing battery consumption.

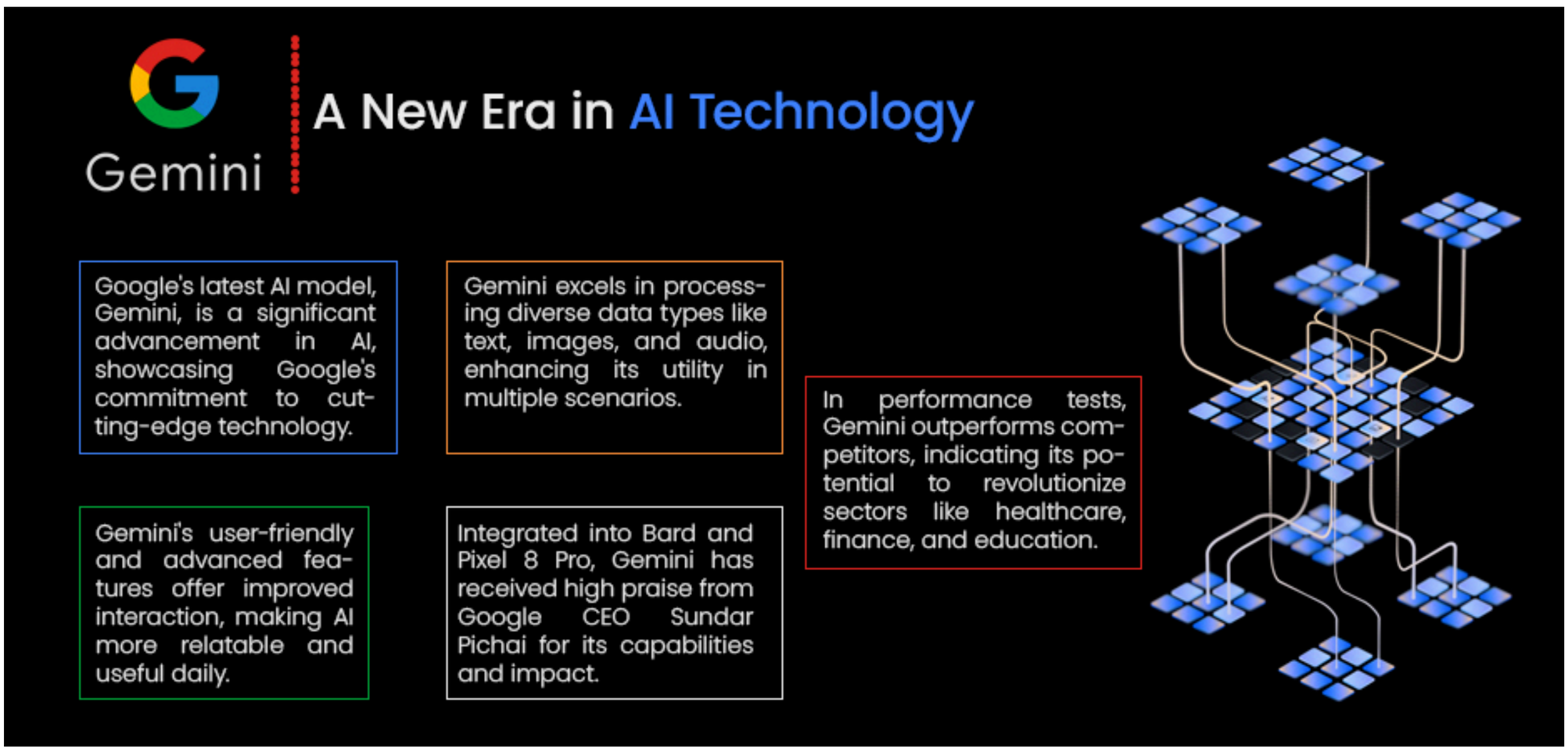
Regulatory Constraints: Adhering to local tourism laws, permissions, and copyright requirements for content.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.

Key Points:



1. **System Architecture:**

The **Gemini Landmark Description App** would enhance tourist experiences using AI by providing real-time, interactive descriptions of landmarks. Key features include:

AI-Powered Descriptions: Personalized, conversational content about landmarks.

AR Integration: Interactive overlays showing historical events or figures.

Voice Assistance: AI-driven voice interactions for hands-free learning.

Real-Time Updates: Information on current events, weather, or exhibits.

Multi-Language Support: Translation services for global tourists.

Location-Based Suggestions: Personalized recommendations for nearby attractions.

Gamification: Fun quizzes or challenges related to landmarks.

2. **User Flow:**

    - Step 1: User enters a query (e.g., "Best motorcycles under ₹ 1 lakh").
    - Step 2: The backend **calls the Gemini Flash API** to retrieve vehicle data.
    - Step 3: The app processes the data and **displays results** in an easy-to-read format.

3. **UI/UX Considerations:**

Here' s a short user flow for the Gemini Landmark Description App:

1. Onboarding: Sign up/login, set preferences (interests, language).
2. Home Screen: View map with nearby landmarks, search bar, and personalized recommendations.
3. Landmark Interaction: Auto-detect or search for landmarks; receive AI-powered descriptions, AR overlays, and voice narration.
4. Voice Assistant: Ask questions for more details about the landmark.
5. Recommendations: Get nearby attraction suggestions and guided tours.
6. Gamification: Participate in quizzes/challenges for rewards.
7. Language Support: Switch languages for international tourists.

8. Notifications: Receive real-time updates on events or weather.
9. Profile/Settings: Manage user profile, preferences, and app settings.

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	Balraju	Google API Key, Python, Streamlit setup	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Bhaskar	API response format finalized	Basic UI with input fields
Sprint 2	Vehicle Search & Comparison	🔴 High	3 hours (Day 2)	Mid-Day 2	Sumedh	API response, UI elements ready	Search functionality with filters
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Dhanush	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Sachin	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	🟢 Low	1 hour (Day 2)	End of Day 2	Balraju Bhaskar Sumedh Dhanush Sachin	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 Landmark Description Gemini App

Key Points:

1. **Technology Stack Used:**
    - **Frontend: HTML,CSS** : For basic structure and styling
    - **Backend: Google Gemini AI API** :For generating landmark descriptions
    - **Programming Language:** Python
2. **Development Process:**
    - Implement **API key authentication** and **Gemini API integration.**

- Develop **vehicle comparison 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 Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	**Latitude:17.385 N Longitude:78.4867 E (Hyderabad)**	Relevant budget cars should be displayed.	✅ Passed	Balraju
TC-002	Functional Testing	**Latitude:48.8584 N Longitude:2.2945 E (Eiffel tower)**	Seasonal tips should be provided.	✅ Passed	Bhaskar
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Tester3
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Dhanush
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Tester2
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