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:

AI AVENGERS

Team Members:

1.Malleshwari
2.Sachitha
3.Sahithya
4.Lirisha

Phase-1: Brainstorming & Ideation

Objective:

By leveraging AI technology to provide interactive, personalized, and real time information about landmarks and enhancing their travel experience and making it more engaging and educational.

Key Points:

1. **Problem Statement:**
    - **Lack of Instant Landmark Information** travelers often struggle to find quick, reliable details about historical and cultural landmarks.
    - **Language & Accessibility Barriers** many tourists face difficulties understanding landmark descriptions due to language differences or accessibility limitations.
    - **Limited Engagement with Cultural Heritage** without engaging and interactive content, visitors may not fully appreciate the historical and architectural significance of landmarks.

2. **Proposed Solution**:
   a) **Instant Landmark Identification & Information**:
      - Users upload an image and input a prompt to receive AI-generated descriptions covering history, architecture, and fun facts.
   b) **Inclusive & Accessible**:
      - The app supports multiple languages and accessibility features, making it easy for all travelers to use.
   c) **Enhanced Travel Experience**:
      - Ideal for tourists, guides, and history enthusiasts, the app deepens cultural appreciation and knowledge of landmarks worldwide.
3. **Target Users:**
   - **Tourists & Travelers** individuals exploring new cities who want quick and detailed information about landmarks.
   - **Tour Guides & History Enthusiasts** professionals and curious learners seeking deeper insights into cultural and historical sites.

4. **Expected Outcome:**
   - **Enhanced Cultural Exploration** users gain instant, detailed insides into landmarks, fostering a deeper appreciation and understanding of global heritage.

Phase-2: Requirement Analysis

Objective:

To define core features, user needs, technical feasibility, accessibility, and security requirements for a seamless and informative tourist experience.

Key Points:

1. **Technical Requirements:**

   - Programming Language: **Python**
   - Backend: **Google Gamini Flash API**
   - Frontend: **Google Collab**
   - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**

   - **Image Upload & Recognition** Users can upload landmark images for AI analysis.

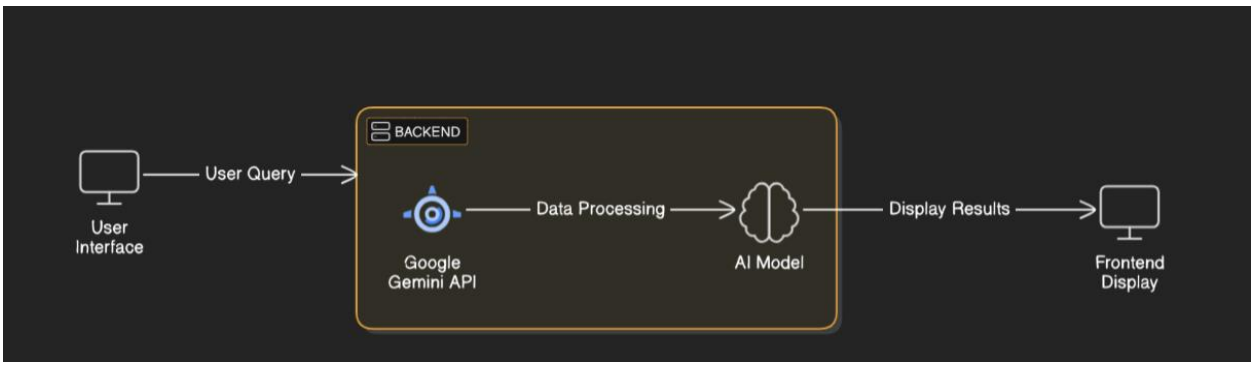
- **AI-generated Descriptions** the app provides historical, architecture, and cultural insights.
- **Real Time Processing** AI quickly generates responses for a seamless user experience.
- **Security & Privacy** Ensures safe image processing and data protection.
- **Sharing & Saving Options** Users can save descriptions or share them via social media.

3. **Constraints & Challenge**
   - Ensuring real-time updates from **Gemini API**.
   - **Privacy** Protecting user data and images.
   - **Device Support** Making sure the app runs smoothly on all devices.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**

   - **User Interface:** A simple app for uploading images, selecting languages, and viewing descriptions.
   - **Application Logic:** AI recognizes landmarks, generates descriptions, and translates them into different languages.
   - **Data Storage:** Stores images, descriptions and landmarks info in the cloud.
   - **External Services**: Uses translation APIs and AI models for recognition and description generation.

2. **User Flow:**

   - Step 1: Open app & upload image.

- Step 2: AI identifies landmark.
- Step 3: View and select language for description.
- Step 4: Save, share, or explore more.

3. **UI/UX Considerations:**

   - **Simplicity** to clean and easy to navigate design.
   - **Accessibility** features like voice input, text to speech, and clear fonts.
   - **Responsiveness** fast performance for quick image processing and description generation.

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	Research & gather landmark data	High	5 hours	End of day 1	Member 1	None	Comprehensive list of landmarks
Sprint 1	Define project scope & objective	High	3 hours	End of day 1	Member 2	None	Clear roadmap for development
Sprint 2	Design database schema	High	4 hours	End of day 1	Member 3	Landmark data collection	Structured database for storing data
Sprint 2	Develop backend API	High	6 hours	End of day 1	Member 4	Data base schema	API for fetching and storing landmark data

Sprint 3	Implement frontend UI	Mediu m	7 hours	Mid-Day 2	Member 1 & 2	API Developmen t	Basic UI for browsing landmarks
Sprint 3	Map integration (Google/ Bing)	High	5 hours	Mid-Day 2	Member 3	API & data base setup	Interactive maps displaying landmarks
Sprint 4	User authentication system	Medium	5 hours	Mid-Day 2	Member 4	Backend & database setup	Secure login & user profiles
Sprint 4	Filter & search functionality	High	6 hours	Mid-Day 2	Member 1 & 3	Landmark database & UI	Users can search for specific landmarks
Sprint 5	Mobile responsivene ss	Medium	4 hours	End of day 2	Member 1	UI implementat ion	Optimized UI for mobile devices
Sprint 5	Performance testing	High	5 hours	End of day 2	Member 2 & 3	Core features	Ensure past & smooth user experience
Sprint 6	User feedback collection	Medium	3 hours	End of day 2	Member 4	Usability testing	Insights for further improvements

Sprint Planning with Priorities

Sprint 1 – Research and project scope (Day 1)

- (⚫ **High Priority)** Research the landmark
- (⚫ **High Priority)** Project scope and objective

Sprint 2 – Design database and develop backend (Day 2)

- (⚫ **High Priority)** Design database schema
- (⚫ **High Priority)** Develop backend API

Sprint 3 – Implement frontend UI & Map integration

(▯ **Medium Priority)** Implement frontend UI
(🔵 **High Priority) ** Map integration

Sprint 4 – User authentication and filter & search

(Medium Priority) User authentication system

(High priority) filter and search functionality

Sprint 5 – Mobile responsiveness and performance testing

(Medium Priority) Mobile responsiveness

(High priority) Performance testing

Sprint 6 – User feedback collection

(Medium Priority) User feedback collection

Phase-5: Project Development

Objective:

Implement core features of the AutoSage App

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Google collab
    - **Backend:** Google Gemini flash
    - **Programming Language:** Python
2. **Development Process:**
    - **Planning & Design** Define objectives, collect data, and design UI/UX.
    - **Development & Integration** Build backend, frontend, and integrate maps & APIs.
    - **Testing & Deployment** Test for performance, fix issues, and launch on the cloud.

3. **Challenges & Fixes:**

    - **Challenge**: Inaccurate or outdated landmark information.
      Fix: Implement regular data
    - **Challenge:** Slow loading time with large datasets.
      Fix: Optimize queries, use coaching, and implement pagination.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Auto Sage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Verify UI elements are responsive.	UI should be interactive and responsive.	Pass/Fail	Tester 1
TC-002	Functional Testing	Install on Windows, Linux .	Installation completes without errors.	Pass/Fail	Tester 2
TC-003	Functional Testing	Run on different OS versions.	Tool should work across all platforms.	Pass/Fail	Tester 3
TC-004	Performance Testing	Increase users to 5000.	Tool should fail gracefully under high load.	Pass/Fail	Tester 4
TC-005	Performance Testing	Transfer 1GB of data.	Throughput should be expected rates.	Pass/Fail	Tester 1
TC-006	Performance Testing	Monitor CPU, memory, disk usage during tests.	Usage should remind within acceptable limits.	Pass/Fail	Tester 3

Final Submission

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