**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:**

DATA CLAN

## **Team Members:**

* Shaik Pharvej
* Shaik Sahil
* P.Bhanu Prakash
* Rohith
* K.Ramu

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

### The goal of the Gemini Landmark Description App is to leverage advanced AI technologies to revolutionize how tourists interact with landmarks and cultural sites. By providing immersive, personalized, and interactive experiences, the app aims to make exploration more engaging, informative, and accessible.

### **Key Points:**

1. **Problem Statement:**

Tourists visiting landmarks and cultural sites often face challenges such as limited access to personalized information, language barriers, and a lack of immersive engagement with historical or cultural contexts.

1. **Proposed Solution:**

The **Gemini Landmark Description App** aims to revolutionize the tourist experience by leveraging cutting-edge AI technologies to provide personalized, immersive, and interactive content about landmarks and cultural sites. By offering real-time descriptions that adapt to user preferences, interests, language, and accessibility needs, the app will enhance engagement and understanding for tourists.

1. **Target Users:**

· **Tourists and Travelers**:

· Individuals visiting historical landmarks, cultural sites, or natural attractions, seeking more in-depth and personalized information about the places they explore.

· **International Visitors**:

· Tourists from different countries who may face language barriers and need multilingual support for seamless and accurate descriptions.

· **History and Culture Enthusiasts**:

· Users who are passionate about history, architecture, and culture and seek detailed, immersive, and interactive experiences at landmarks.

· **Families and Group Travelers**:

· Groups looking for engaging and interactive ways to learn about landmarks, including features that cater to various age groups or interests.

· **Educational Institutions and Students**:

· Schools, universities, or individual students who use the app for educational purposes, utilizing AI-driven descriptions and AR features to better understand historical or cultural sites.

· **Accessibility-Oriented Users**:

· Individuals with specific accessibility needs, such as those requiring visual or auditory aids, voice guides, or personalized content that accommodates disabilities.

· **Tech-Savvy and Experience-Seeking Users**:

· Travelers who enjoy using advanced technology such as augmented reality (AR) and interactive mobile apps to enhance their travel experiences.

1. **Expected Outcome:**The expected outcomes focus on improving both individual tourist experiences and contributing to broader tourism growth and cultural understanding.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the Gemini Landmark Description App.

### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Python**
   * Backend: **open AI , wikipedia**
   * Frontend**: streamlit**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**
   * Ability for Landmak Description using Gemini Flash API.
   * Display History of landmark,images and
   * Provide history and culture of landmark
3. **Constraints & Challenges:**
   * Ensuring real-time updates from **Gemini API**.
   * Handling **API rate limits** and optimizing API calls.
   * Providing a **smooth UI experience** with Streamlit.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the application.



### **Key Points:**

1. **System Architecture:**
   * User uplode image of landmark.
   * Query is processed using **Google Gemini API**.
   * AI model fetches and processes the data.
   * The frontend displays history and culture of landmark.
2. **User Flow:**
   * Step 1: User uplode picture of landmark
   * Step 2: The backend **calls the Gemini Flash API** to retrieve description of landmark.
   * 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 history,culture of landmark**.
   * **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 | Shaik Pharvej | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | P.Bhanu Prakash | API response format finalized | Basic UI with input fields |
| Sprint 2 | Landmark Search & Filtering | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Shaik sahil | API response, UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Shaik Pharvej,P.Bhanu prakash | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | K.Ramu | 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 App.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Streamlit
   * **Backend: open AI , wikipedia**
   * **Programming Language:** Python
2. **Development Process:**
   * Implement **API key authentication** and **Gemini API integration**.
   * Develop history and culture description of landmark.
   * 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 "Historical landmarks in Paris" | Relevant landmarks in Paris should be displayed. | ✅ Passed | Shaik.Sahil |
| TC-002 | Functional Testing | Query "Top cultural landmarks in Japan" | Cultural landmarks in Japan should be displayed. | ✅ Passed | Shaik.Pharvej |
| TC-003 | Performance Testing | API response time under 500ms for landmark details | API should return results quickly. | ⚠ Needs Optimization | P.Bhanu Prakash |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect landmark details in API responses | Data accuracy for landmark descriptions should be improved. | ✅ Fixed | Developer |
| TC-005 | Final Validation | Ensure UI is responsive across devices | UI should work properly on mobile and desktop devices. | ❌ Failed - UI broken on mobile | Rohith |
| TC-006 | Deployment Testing | Host the app using Streamlit Sharing | The app should be accessible online to users. | 🚀 Deployed | DevOps |

## **Final Submission**

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