**Hackathon Project Phases Template**  for the **Triplens** project.

**Hackathon Project Phases Template**

**Project Title:**

**Triplens App Using Gemini Flash**

**Team Name:**

Iconic Insights

**Team Members:**

* Thanniru Manoharini
* Amma Sri Harsha
* Thanniru Tejasree
* Gotti Sai Poojitha

**Phase-1: Brainstorming & Ideation**

**Objective:**

**Enhance tourist’s experiences by providing detailed, AI-generated landmark information through user-uploaded images and prompts, with support for multiple languages and accessibility features.**

**Key Points:**

1. **Problem Statement:**

* **Lack of Immediate Information: Tourists often find it difficult to quickly access detailed and comprehensive information about landmarks, which can make their exploration less enriching.**
* **Inclusivity and Accessibility Challenges: Many current tools do not provide adequate support for multiple languages and accessibility options, making it harder for people from diverse backgrounds and abilities to access and enjoy landmark information.**

1. **Proposed Solution:**

* **AI-Powered Landmark Descriptions: Develop an app that uses AI to generate detailed and comprehensive descriptions of landmarks based on user-uploaded images. This will provide tourists with instant access to information about a landmark's historical significance, architectural features, and interesting facts.**
* **Multilingual and Accessible Features: Integrate multilingual support and accessibility options to ensure that users from diverse backgrounds and abilities can access and appreciate landmark information. This will promote inclusivity and enhance the user experience.**

1. **Target Users:**

* **Tourists and Travelers**: Individuals exploring new cities who want immediate and detailed information about landmarks.
* **Tour Guides**: Professionals leading tours who need quick access to comprehensive landmark descriptions.
* **History and Culture Enthusiasts**: People with a keen interest in learning about the historical and cultural significance of landmarks.

1. **Expected Outcome:**The Gemini Landmark Description App provides users with instant, detailed, and multilingual landmark descriptions, enhancing the tourist experience with accessible and inclusive information.

**Phase-2: Requirement Analysis**

**Objective:**

* **Define the technical and functional requirements for the Triplens App.**

**Key Points:**

1. **Technical Requirements:**
   * Programming Language: **Java,Python**
   * Backend:**FastAPI**
   * Frontend: **Flutter**
   * Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**

* Ability to fetch landmark details using Gemini Landmark API
* Display detailed landmark descriptions in an intuitive and visually appealing UI
* Provide real-time cultural and historical insights based on the landmark's significance.
* Allow users to search landmarks based on geographical location or historical period.

**3.Constraints & Challenges:**

* + **Data Quality and Availability**
  + **Performance**
  + **Image Recognition Accuracy**

**Phase-3: Project Design**

**Objective:**

Develop the architecture and user flow of the application.

A screen shot of a computer

AI-generated content may be incorrect.

**Key Points:**

1. **System Architecture:**
   * **User Interface (UI)**
   * **Processing the Image and Query**
   * **Data Fetching and Processing**
   * **Displaying Results**
2. **User Flow:**
   * Launch the app.
   * Upload an image and enter a prompt.
   * Image and query are processed.
   * AI generates descriptions.
   * Display results.
   * View additional options.
   * Manage profile and settings.
   * Provide feedback or seek support.
   * End session.
3. **UI/UX Considerations:**

* **Easy to Use**: Keep the app simple and easy to navigate.
* **Clear Feedback**: Make sure users know when their actions are successful.
* **Good Design**: Use clear fonts, good color contrast, and nice visuals.
* **Accessible**: Ensure the app works for everyone, including those with disabilities.
* **Consistent**: Keep design elements consistent throughout the app.

**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 | Tejasree | AndroidStudio,  Android toolkit, | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Manoharini and Poojitha | Flutter, | Basic UI with input fields |
| Sprint 2 | Models of Landmarks | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | SriHarsha and poojitha | Figma | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Tejasree and SriHarsha | Chatgpt,Gemini | Improved API stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Manoharini | Hardware mobile,crome,android studio | 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)** integrated based with frontend  
 **(🟡 Medium Priority)** Build a **basic UI with input fields**.

**Sprint 2 – Core Features & Debugging (Day 2)**

**(🔴 High Priority)** Implement **landmark recognition from images**.  
 **(🔴 High Priority)** generate AI-based description using GPT-4

**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 Triplens App.

**Key Points:**

1. **Technology Stack Used:**
   * **Frontend:** Flutter
   * **Backend:** FastAPI
   * **Programming Language:** Python,Java
2. **Development Process:**
   * Implement **API key authentication** and **Gemini API integration**.
   * Develop **UI 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 search.  
      **Fix:** Optimize queries to fetch **only recent 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 "Best Landmark I like” | Relevant budget cars should be displayed. | ✅ Passed | Tejasree |
| TC-002 | Functional Testing | Query "landmark maintenance tips for winter" | Seasonal tips should be provided. | ✅ Passed | Manoharini |
| TC-003 | Performance Testing | Response for the asked landmark | API should return results quickly. | ⚠ Needs Optimization | SriHarsha |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect results. | Data accuracy should be improved. | ✅ Fixed | Poojitha |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | SriHarsha |
| TC-006 | Deployment Testing | Host the app using Flutter | App should be accessible online. | 🚀 Deployed | DevOps |