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

# Hackathon Project Phases Template

**Project Title: SEVAK**

**Audio to visuals: The Art of Sound Visualization: Creating Dynamic Audio-Visual Experiences"**

**Team Name:**

Hackfinity

**Team Members:**

* Bukya. Lokesh
* D. Veera Venkata Satya Ganesh Kumar
* A. Ganesh
* A. Siddu
* Ch. Harshith

## Phase-1: Brainstorming & Ideation

**Objective:**

**Audio2Art: Transforming Voice Prompts into Visual Creations**

**Key Points:**

1. **Problem Statement:**

Despite the rich auditory experiences provided by soundscapes, there is a lack of effective methods to visually represent these sounds, limiting the ability to engage audiences in multi-sensory experiences."

1. **Proposed Solution:**

* Develop a software tool that analyzes soundscapes in real-time and generates corresponding visual representations using techniques such as spectrograms, waveforms, and generative art. This tool could allow users to interactively explore soundscapes through visualizations that change dynamically with the audio input.

1. **Target Users:**

* **Education:** students, teachers, parents.
* **Others:**  Sound artists, musicians, audio engineers, educators, and researchers interested in sound visualization.

1. **Expected Outcome:**

* 1. A functional **AI-powered website** that provides visuals based on voice prompt based on user requirement

## Phase-2: Requirement Analysis

**Objective:**

Define the technical and functional requirements for the AutoSage App.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python**

○ Backend: **Google Gemini Flash API**

○ Frontend: **Streamlit Web Framework**

○ Database: **Not required initially (API-based queries)**

1. **Functional Requirements:**

* 1. Ability to **fetch vehicle details** using Gemini Flash API.

○ Display **specifications, reviews, and comparisons** in an intuitive UI.

○ Provide **real-time vehicle maintenance tips** based on seasons.

○ Allow users to **search eco-friendly vehicles** based on emissions and incentives.

1. **Constraints & Challenges:**

* 1. Ensuring real-time updates from **Gemini API**.

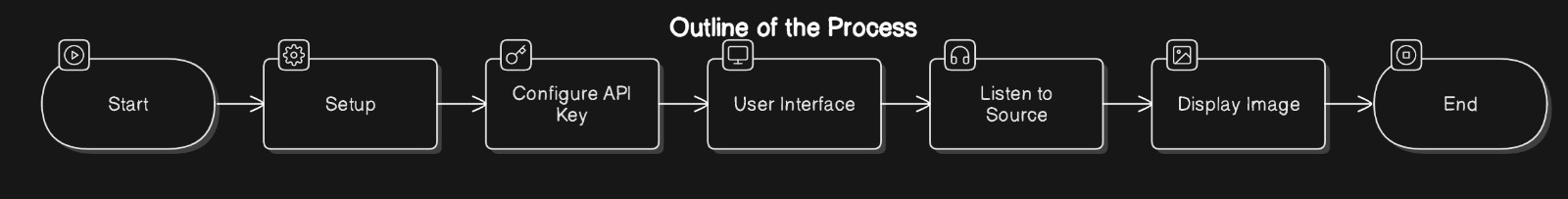
○ Handling **API rate limits** and optimizing API calls.

○ 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:**

* 1. User enters voice-related query via UI.

○ Query is processed using **Google**  **AI STUDIO**.

○ AI model fetches and processes the data.

○ The frontend displays **voice prompt, and results.**

1. **User Flow:**

* 1. Step 1: User enters a query (e.g., "batsmen hitting six").

○ Step 2: The backend **calls the Google AI STUDIO** to retrieve data based on voice prompt.

○ Step 3: The webpage processes the data and **displays results** in an easy-to-read format.

1. **UI/UX Considerations:**

* 1. **Minimalist, user-friendly interface** for seamless navigation.

○ **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 | A.Ganesh  A.siddu  D.veera | Google AI studio,  Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | D.veera | API response format finalized | Basic UI with input fields |
| Sprint 2 | AI prompt | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | A.Siddu | API response, UI elements ready | Create functionality with filters |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | A.Ganesh  A.siddu  D.veera | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | CH.harshith  B.Lokesh | 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 AI STUDIO API**.

**(**🟡 **Medium Priority)** Build a **basic UI with input fields**.

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

**(**🔴 **High Priority)** Implement **voice & create 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 Web page.

**Key Points:**

1. **Technology Stack Used:**

* 1. **Frontend:** Streamlit

○ **Backend:** Google AI Studio API

○ **Programming Language:** Python

1. **Development Process:**

* 1. Implement **API key authentication** and **Open AI API integration**.

○ Develop **voice prompting and visual generation**

○ Optimize **search queries for performance and relevance**.

1. **Challenges & Fixes:**

* 1. **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 Webpage works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Query "a cat" | A cat. | ✅ Passed | A.ganesh |
| TC-002 | Functional  Testing | Query "old men hitting six" | An image of an old man hitting six | ✅ Passed | A.siddu |
| TC-003 | Performance  Testing | API response time under  500ms | API should return results quickly. | ⚠ Needs Optimization | D.veera |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect API responses. | Data accuracy should be improved. | ✅ Fixed | D.veera |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | B.lokesh |
| TC-006 | Deployment  Testing | Host the app using  Streamlit Sharing | webpage should be accessible online. | 🚀 Deployed | team |

## Final Submission

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