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

s

# Hackathon Project Phases Template

**Project Title:**

ProvisionAI: Unleashing the power of Gemini Vision for Image Annotation

**Team Name:**

# New Vision

**Team Members:**

## Gangu Murali Karthik Reddy

## Mohammed Moshaibuddin

## Modem. Naga Kusuma Sri

## Phase-1: Brainstorming & Ideation

# Objective:

To develop ProVision AI, an AI-powered image annotation tool that leverages Google Gemini Vision API to analyze images, detect objects, extract text, and provide insightful annotations for enhanced visual understanding.

## Key Points:

## Problem Statement:

Manual image annotation is time-consuming and inefficient. ProVisionAI aims to solve this by using Google Gemini Vision API to automatically analyze images, detect objects, recognize text, and generate meaningful annotations, making image processing faster and more accurate.

.

## Proposed Solution:

**1. AI-Powered Image Analysis** – Utilize **Google Gemini Vision API** to automatically detect objects, extract text, and generate annotations from images.

**2. User-Friendly Interface**  – Develop a Streamlit-based web app where users can upload images and receive AI-generated insights instantly.

**Target Users:**

1. **Researchers & Developers** – For AI/ML model training and data labeling.
2. **Businesses & Marketers** – To analyze product images and extract insights.
3. **Students & Educators** – For learning and academic projects related to AI and image processing.
4. **Healthcare & Security Professionals** – For analyzing medical images or surveillance footage.
5. **Content Creators & Designers** – To organize and tag visual assets efficiently.

**Expected Outcome:**

1. **Automated Image Annotation** – Users get AI-generated labels, object detection, and text extraction from images.
2. **Faster & Accurate Analysis** – Eliminates manual annotation, making image processing more efficient.
3. **User-Friendly Web App** – A simple interface for uploading images and viewing AI insights.

## Phase-2: Requirement Analysis

**Objective:**

To develop **ProVisionAI**, an AI-driven image annotation tool that utilizes **Google Gemini Vision API** to automate object detection, text extraction, and image analysis, providing fast, accurate, and user-friendly insights for various applications.

**Key Points:**

**1.Technical Requirements:**

* + - * Programming Language: Python
      * Backend: Google Gemini Vision API
      * Frontend: ProVisionAI is built using Streamlit
      * Database: Local storage

**2.Functional Requirements:**

* + - * + **Upload Images** – Users can upload JPG, PNG, or JPEG files.
        + **AI Analysis** – Google Gemini Vision API processes images to detect objects and text.
        + **Show Results** – Display detected objects, extracted text, and annotations.
        + **Download Data** – Users can download the AI results in JSON format.
        + **Easy-to-Use UI** – A simple **Streamlit** interface for interaction.
        + **Error Handling** – Show messages for invalid uploads or API issues.
        + **Fast & Efficient** – Ensure quick processing and smooth performance.

.

**3.Constraints & Challenges:**

**API Limits & Costs** – Google Gemini Vision API has usage limits and may require billing.

**Internet Dependency** – Requires a stable internet connection for image processing.

**Fast Processing** – Ensuring quick response times for multiple image requests.

## Phase-3: Project Design

**Objective:**

The project design aims to build a **fast, accurate, and user-friendly** AI-powered image annotation tool using **Google Gemini Vision API**. It ensures **seamless image processing, an intuitive Streamlit interface, and efficient performance** for handling multiple requests. The goal is to provide a **scalable and accessible solution** for automated image analysis.

**Key Points:**

**AI-Powered Image Annotation**

**Simple & User-Friendly UI**

**Fast & Efficient Processing**

**Scalability & Performance**

**Error Handling & Security**

## Phase-4: Project Planning (Agile Methodologies)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | Murali | Google API Key,  Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡  Medium | 2 hours  (Day 1) | End of Day  1 | Kusuma sri | API response format finalized | Basic UI with input fields |
| Sprint 2 | Provision AI connect to google vision api | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Moshaibuddin | API response, UI elements ready | Linked provision AI to google vision API |
| Sprint 2 | Error Handling &  Debugging | 🔴 High | 1.5 hours  (Day 2) | Mid-Day 2 | Murali and kusuma | API logs, UI inputs | Improved API stability |
| Sprint 3 | Testing & UI  Enhancements | 🟡  Medium | 1.5 hours  (Day 2) | Mid-Day 2 | moshaibuddin | 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 |

## Phase-5: Project Development

**Objective:**

The objective of **ProVisionAI** development is to build a **fast, accurate, and user-friendly** AI-powered image annotation tool using **Google Gemini Vision API**.

**Key Points:**

**AI Integration** – Uses **Google Gemini Vision API** for image analysis.  
**User-Friendly UI** – Built with **Streamlit** for easy interaction.  
**Efficient Processing** – Ensures **fast and accurate** image annotation.  
**Error Handling** – Manages API failures and invalid inputs smoothly.  
**Scalability** – Designed to handle multiple users efficiently.  
**Deployment & Testing** – Ensures a **stable and demo-ready** final product.

## Phase-6: Functional & Performance Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | Image upload testing | image uploaded | ✅ Passed | Murali |
| TC-002 | Functional  Testing | API integration  testing | Seasonal tips should be provided. | ✅ Passed | Kusumasri |
| TC-003 | Performance  Testing | API response time under  500ms | API should return results quickly. | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Ui functionality  testing | Data accuracy should be improved. | ✅ Fixed | moshaibuddin |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed to give response | Entire team |
| TC-006 | Deployment  Testing | Annotation accuracy testing | Not able to  annotate | 🚀 Deployed | Gemini vision API |