Hackathon Project Phases Template for the **AI Image Annotation** project.

Hackathon Project Phases Template

Project Title:

ProVisionAI: Unleashing the Power of Gemini Vision for Image Annotation

Team Name:

Optimize Prime

Team Members:

- K.Veda Swarup Bhushan
- T. Sai Vikhil Reddy
- Mohammed Ayman
- A.Ruthvik
- Mohammed Fahad Ali Siddique

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered image annotation platform using Gemini Vision Pro to generate rich descriptive captions and insightful information for uploaded images.

Key Points:

1. **Problem Statement:**
    - Many users struggle with manual image annotation, which is time-consuming and lacks consistency.
    - There is a need for an AI-driven solution that can automatically generate precise descriptions and relevant facts about images.
2. **Proposed Solution:**
    - An AI-powered platform leveraging Google's Gemini Vision Pro to analyze and annotate images.

    - Provides users with detailed image descriptions and contextual insights.
    - Enhances accessibility and automation in various domains, including education, e-commerce, and media
3. **Target Users:**

    - **Graphic designers** and **content creators** needing automated image descriptions.
    - **Researchers** and **archivists** requiring precise image metadata.
    - **Businesses** using AI-driven tagging for media libraries and e-commerce**.**
4. **Expected Outcome:**

    - A functional **AI-powered image annotation system** that can generate meaningful insights from uploaded images.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for AI-powered image annotation system.

Key Points:

1. **Technical Requirements:**

    - Programming Languages: **Express.js, Node.js**
    - Backend: **Google Gemini Flash API, Express.js**
    - Frontend: **Html, Css, Java script**
    - Database: **Not required initially (API-based queries)**
2. **Functional Requirements:**

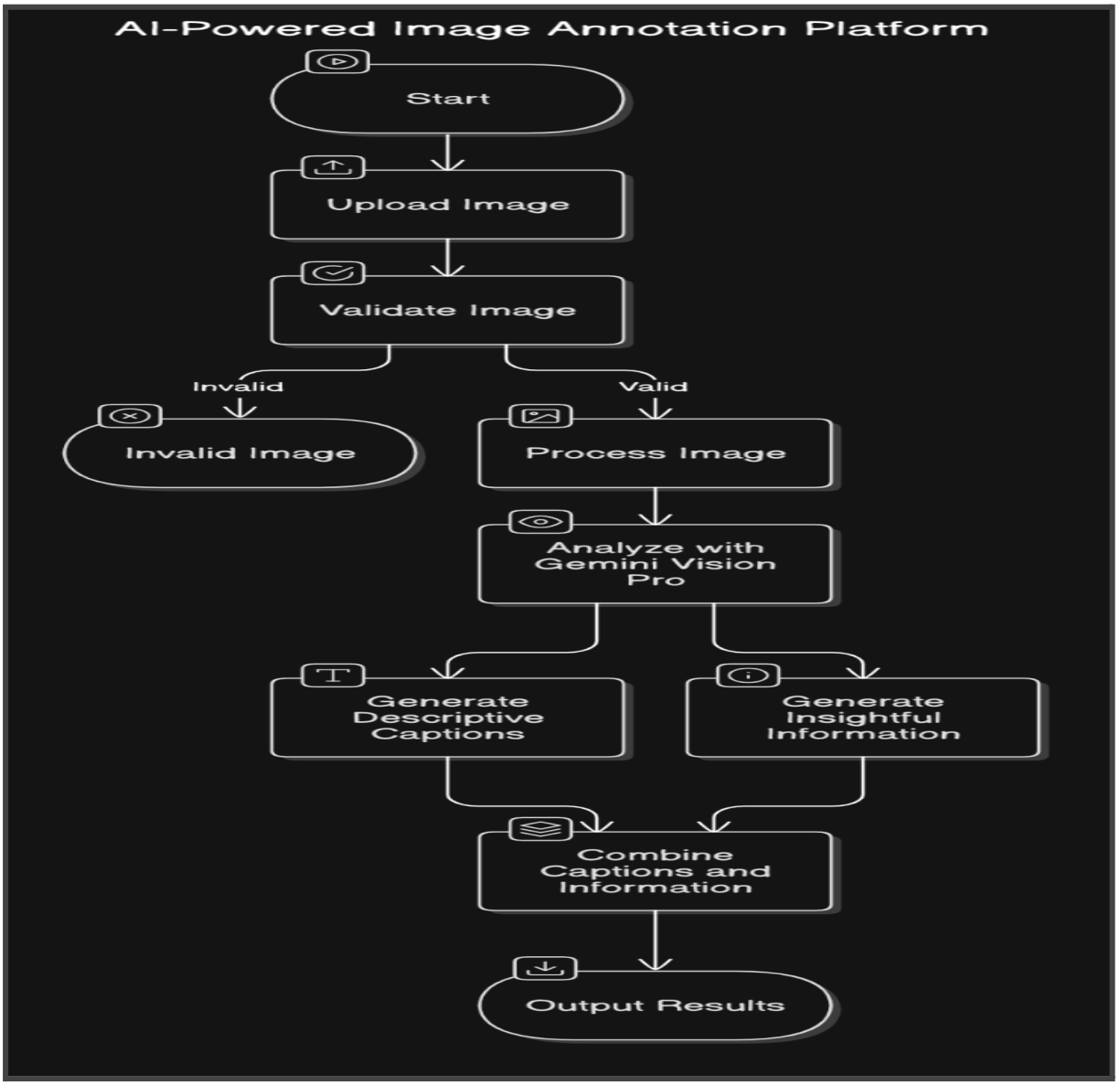
    - Upload images and receive AI-generated annotations.
    - Provide contextual insights and interesting facts about the image content.
    - Display results in a user-friendly interface.
3. **Constraints & Challenges:**

    - Ensuring high accuracy in AI-generated captions.
    - Handling API rate limits and optimizing API calls.
    - Delivering a smooth and intuitive user experience.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - User uploads an image via UI.
    - The backend processes the image using Gemini Vision Pro API.
    - AI model analyzes the image and generates annotations.
    - The frontend displays the annotated results.
2. **User Flow:**
    - **Step 1:** User uploads an image.

- **Step 2:** Backend calls Gemini Vision Pro API for analysis.
- **Step 3:** The AI model generates and returns annotations.
- **Step 4:** The annotated image and insights are displayed.

3. **UI/UX Considerations:**

- Clean, minimalistic interface for a seamless experience.
- Drag-and-drop image upload functionality.
- Dark & light mode options for accessibility.

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	Veda Swarup, Vikhil Reddy Ayman	Google API Key, Express.js	API connection established & working
Sprint 1	Frontend UI Development	🟡 Medium	2 hours (Day 1)	End of Day 1	Fahad A. Ruthvik	API response format finalized	Basic UI with upload fields
Sprint 2	Image Upload & Processing	🔴 High	3 hours (Day 2)	Mid-Day 2	Vikhil Reddy Ayman	API response, UI elements ready	Image Process working
Sprint 2	Error Handling & Debugging	🔴 High	1.5 hours (Day 2)	Mid-Day 2	Veda Swarup Fahad	API logs, UI inputs	Stable API responses
Sprint 3	Testing & UI Enhancements	🟡 Medium	1.5 hours (Day 2)	Mid-Day 2	Veda Swarup Fahad A. Ruthvik	API response, UI layout completed	Responsive UI, Enhanced UX
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 AI-powered image annotation system.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Html, Css, Java script
    - **Backend:** Google Gemini Flash API, Express.js
    - **Programming Language:** Express js, Node js
2. **Development Process:**
    - Implement API key authentication and integration.
    - Develop image upload and annotation processing logic.
    - Optimize query handling for performance and accuracy.
3. **Challenges & Fixes:**
    - **Challenge:** Slow API response times.
      Fix: Implement caching for frequently queried results.
    - **Challenge:** Limited API calls per minute.
      Fix: Optimize requests to minimize redundant queries.

Phase-6: Functional & Performance Testing

Objective:

Ensure the AI-powered image annotation system functions as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Upload an image of a landmark	Relevant budget cars should be displayed.	✅ Passed	Veda Swarup, Vikhil Reddy
TC-002	Functional Testing	Query "Motorcycle maintenance tips for winter"	Seasonal tips should be provided.	✅ Passed	Ayman Fahad
TC-003	Performance Testing	API response time under 500ms	API should return results quickly.	⚠ Needs Optimization	Ruthvik fahad
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Veda Swarup, Fahad
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Entire team
TC-006	Deployment Testing	Host the app using Streamlit Sharing	App should be accessible online.	🚀 Deployed	DevOps

---

Final Submission

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