Hackathon Project Phases Template for the **Gesture – Based Human Computer Interaction System** project.

Hackathon Project Phases Template

Project Title:

Gesture – Based Human Computer Interaction System using OpenCV, MediaPipe and Palm’s text-bison-001

Team Name:

Gesture Grid

Team Members:

- Amulya. T
- S. Akshaya
- S. Sai Priya
- Y. Laxmi

Phase-1: Brainstorming & Ideation

Objective:

Develop a Gesture-Based Human-Computer Interaction System that enables touchless digital interactions using hand gesture recognition. The system leverages OpenCV, MediaPipe, and Palm’s text-bison-001 to detect, interpret, and describe user gestures in real time. This innovative solution enhances hygiene, accessibility, and user experience in public kiosks, retail, education, and healthcare environments.

Key Points:

1. **Problem Statement:**

    - People often face difficulties using public touchscreens due to hygiene concerns and lack of accessibility for individuals with disabilities.
    - Frequent maintenance, wear and tear, and inefficient interactions make traditional touch interfaces less reliable in high-traffic environments.

2. **Proposed Solution:**

    - The system uses **OpenCV** and **MediaPipe** to recognize hand gestures for touch-free control of public kiosks, reducing physical contact and promoting hygiene, especially in high-traffic areas during health crises.
    - By leveraging standard webcams and open-source libraries, the system offers an affordable, customizable, and scalable solution for various industries, including retail, healthcare, and public services.

3. **Target Users:**

    - **Public Kiosk Users:** Individuals using touchless kiosks in high-traffic areas.
    - **Business Owners:** Companies implementing cost-effective, touch-free interaction systems.

4. **Expected Outcome:**

    - **A Touchless Interaction & Hygiene:** Reduced physical contact in public spaces through gesture-based control, promoting a hygienic environment.

    - **Enhanced User Experience & Accessibility:** Intuitive, real-time gesture recognition for seamless kiosk navigation and interaction in high-traffic areas.

Phase-2: Requirement Analysis

Objective:

Define and prioritize technical and functional requirements for Gesture based HCI.

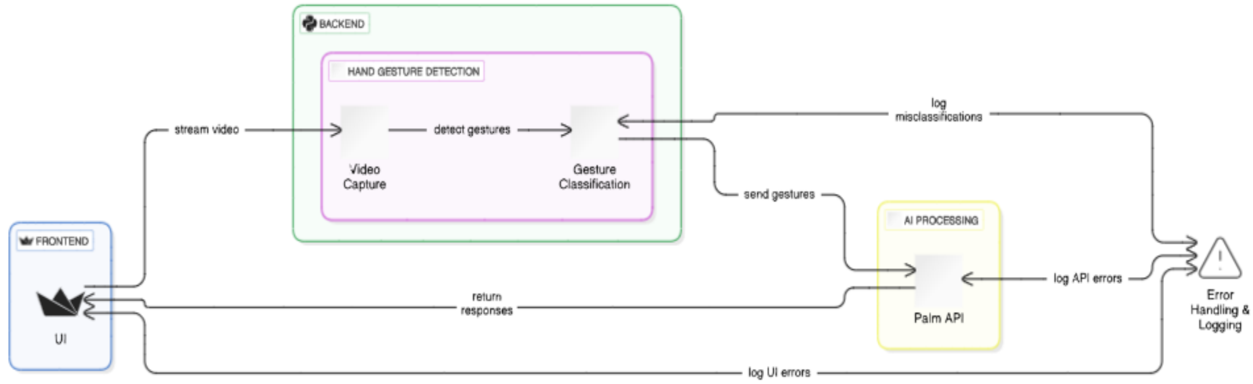
Key Points:

1. **Technical Requirements:**
   - Programming Language: Python
   - Frontend: Streamlit Web Framework
   - Database: Not required initially (API-based queries via Google Palm API for AI-based descriptions)
2. **Functional Requirements:**
   - Real-time hand gesture recognition using MediaPipe.
   - AI-based description generation for recognized gestures using Google Palm API.
   - Streamlit interface to display real-time video feed and recognized gestures.
   - User-friendly visualization of gestures and their corresponding descriptions.
3. **Constraints & Challenges:**
   - Ensuring fast and accurate gesture recognition.
   - Handling real-time video processing without significant latency.
   - Managing the API response time and ensuring smooth interaction.
   - 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:**

   ○ **Frontend:**
   Streamlit: Provides the user interface to display real-time video, recognized gestures, and AI descriptions.

   ○ **Core Processing:**
   OpenCV: Captures the video feed.
   MediaPipe: Detects hand gestures by tracking landmarks.

   ○ **AI Integration:**
   Google Palm API: Generates descriptions for recognized gestures.

   ○ **Data Flow:**
   Video is captured by OpenCV, processed by MediaPipe for gesture recognition, and descriptions are fetched from Palm API. The results are displayed on the Streamlit interface.

2. **User Flow:**

   - **Step 1: Launch Application:**
     The user opens the Streamlit web interface.
   - **Step 2: Start Video Capture:**
     The webcam feed is captured by OpenCV and displayed in real-time on the interface.
   - **Step 3: Hand Gesture Detection:**
     MediaPipe processes the video frames to detect hand landmarks and classify gestures (e.g., thumbs up, fist, open hand).
   - **Step 4: Gesture Recognition:**
     When a gesture is recognized, it is classified (e.g., “thumbs_up”, “fist”).
   - **Step 5: AI Description:**
     For recognized gestures, a prompt is sent to the Google Palm API to generate a description.
   - **Step 6: Display Results:**
     The recognized gesture and its AI-generated description are displayed on the interface in real-time.
   - **Step 7: User Interaction:**
     The user can continue interacting by performing different gestures to trigger corresponding actions or descriptions.

3. **UI/UX Considerations:**

   - **Simple Design:** Clean and intuitive layout with clear labels for gestures and descriptions.

   - **Real-Time Feedback:** Instant visual updates on gesture recognition and AI descriptions.

   - **Performance:** Ensure smooth, fast gesture detection with minimal delay.

   - **Accessibility:** High contrast colors for visibility and legible text.

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 and API integration	🔴 High	1.5 hours (Day 1)	End of Day 1	Laxmi and Amulya	Google API Key, Python	Functional Setup
Sprint 1	Basic UI using Streamlit	🟡 Medium	1 hour (Day 1)	End of Day 1	Akshaya	Streamlit setup	Interactive UI
Sprint 1	Implement hand gesture detection and AI-based gesture interpretation	🔴 High	2 hours (Day 1)	Mid-Day 2	Akshaya and Amulya	MediaPipe, OpenCV, Palm API	Gesture recognition working and meaningful responses
Sprint 2	Error Handling & Debugging	🔴 High	1 hour (Day 2)	Mid-Day 2	Sai Priya and Laxmi	API logs, UI inputs	Improved API stability
Sprint 3	Testing & UI Enhancements	🟡 Medium	1 hour (Day 2)	Mid-Day 2	Sai Priya and Akshaya	API response, UI layout completed	Responsive UI, better user experience
Sprint 3	Final 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**, API Integration & install dependencies.
(🔴 High Priority) Implement Hand Gesture Detection & AI .

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

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 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 Gesture based HCI application.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** Palm API
    - **Programming Language:** Python
2. **Development Process:**
    - **Setup & Integration** – Configure environment, integrate APIs (Google API, Palm API, MediaPipe), and set up Streamlit for UI.
    - **Implementation & Debugging** – Develop gesture detection, AI-based responses, and handle errors through API logs and UI validation.
    - **Testing & UI Enhancements –** Refine UI, improve responsiveness, and ensure stable performance.
3. **Challenges & Fixes:**
    - **Gesture Detection Accuracy**

      Challenge: Misclassification of gestures due to varying hand positions.

      Fix: Adjust landmark detection logic and fine-tune threshold conditions for better recognition.

    - **Package Version Incompatibility**

      Challenge: Errors while installing and using dependencies due to version conflicts.

      Fix: Use a virtual environment, specify compatible versions in requirements.txt, and test installations systematically.

Phase-6: Functional & Performance Testing

Objective:

Ensure that all features, including gesture detection, AI-based responses, and UI interactions, work as expected without errors.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Verify API integration with Palm AI	API should return meaningful responses	✅ Passed	Akshaya
TC-002	Functional Testing	Test hand gesture detection	System should correctly recognize gestures	✅ Passed	Amulya
TC-003	Performance Testing	Measure response time of AI-based interpretation	API should return results quickly.	⚠ Needs Optimization	Sai Priya
TC-004	Bug Fixes & Improvements	Fixed incorrect API responses.	Data accuracy should be improved.	✅ Fixed	Laxmi
TC-005	Final Validation	Ensure UI is responsive across devices.	UI should be responsive and display results	✅ Passed	Akshaya

---

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
   https://github.com/AmulyaThammineni/Gesture-Based-Human-Computer-Interaction-System-using-OpenCV-MediaPipe-and-Palm-s-text-bison.git

4. **Presentation**