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

Project Title:

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

Team Name:

WaveTech

Team Members:

- Akhila Patha
- Spandana Pola
- Keerthana Yalla
- Harshitha Thangella

Phase-1: Brainstorming & Ideation

Objective:

Develop a real-time hand gesture recognition system that enables users to interact with computers using intuitive hand movements. The system uses OpenCV and MediaPipe for gesture detection and integrates a generative AI model to provide descriptive narratives for recognized gestures. A user-friendly Streamlit interface facilitates seamless interaction.

Key Points:

Problem Statement:

- Public kiosks, such as those in airports and museums, require touchless interactions to enhance hygiene and accessibility.
- Users need a convenient way to interact with digital interfaces using hand gestures, reducing reliance on physical touch.
- The lack of intuitive touchless interfaces limits accessibility for users with disabilities.

Proposed Solution:

- Implement a computer vision-based gesture recognition system using OpenCV and MediaPipe.
- Utilize Palm's text-bison-001 generative AI model to generate real-time gesture descriptions.
- Develop a user-friendly Streamlit interface for interactive gesture-based controls.

Target Users:

- Visitors using kiosks at public places (airports, museums, malls, etc.).
- Individuals with physical disabilities who benefit from touch less interaction.
- Gamers and technology enthusiasts interested in innovative control mechanisms.

Expected Outcome:

- A functional gesture-based interaction system that improves user experience.
- Enhanced accessibility and hygiene at public interfaces.
- Seamless integration with existing kiosk software and interactive applications.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the gesture-based system.

Key Points:

Technical Requirements:

- **Programming Language:** Python
- **Computer Vision Tools:** OpenCV, MediaPipe
- **AI Model:** Palm's text-bison-001
- **Frontend:** Streamlit
- **Hardware:** Standard webcam for gesture recognition

Functional Requirements:

- Detect and classify hand gestures in real time.
- Provide AI-generated descriptive narratives for recognized gestures.
- Enable touchless navigation and selection in a user interface.

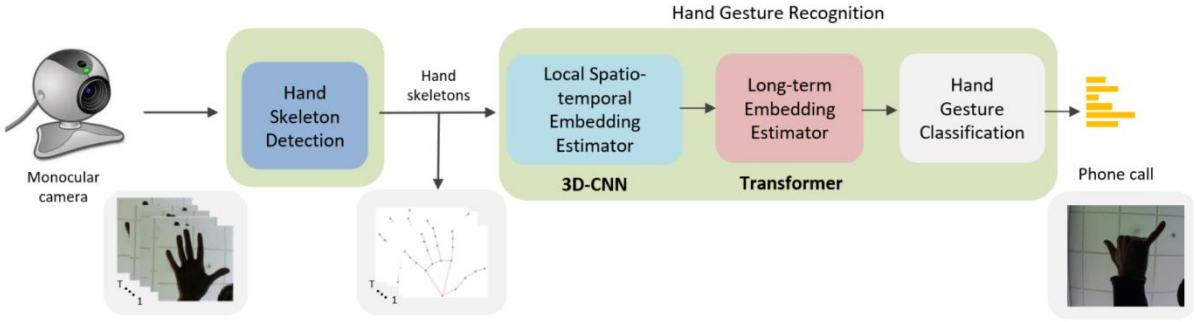
Constraints & Challenges:

- Ensuring accurate real-time gesture recognition.
- Optimizing AI-generated descriptions for clarity and relevance.
- Providing a seamless and responsive user interface.

Phase-3: Project Design

Objective:

Design the architecture and user flow of the application.



Key Points:

System Architecture:

1. **Gesture Detection**: OpenCV and MediaPipe capture hand gestures in real time.
2. **AI Processing**: Palm's text-bison-001 generates descriptions for detected gestures.
3. **User Interface**: Streamlit displays recognized gestures and descriptions, allowing user interaction.

User Flow:

1. **Step 1**: User performs a hand gesture in front of the webcam.
2. **Step 2**: The system detects and classifies the gesture.
3. **Step 3**: The AI model generates a description of the gesture.
4. **Step 4**: The system provides a response, such as navigating a menu or selecting an option.

UI/UX Considerations:

- Clean and intuitive UI design for seamless interaction.
- Visual feedback for detected gestures.
- Accessibility features for users with disabilities.

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

- (🔴 High Priority) Set up the development environment and install dependencies.
- (🔴 High Priority) Integrate OpenCV and MediaPipe for gesture recognition.
- (🟡 Medium Priority) Implement a basic UI with a live video feed.

Sprint 2 – Core Features & Debugging (Day 2)

- (🔴 High Priority) Develop AI-powered gesture description generation.
- (🔴 High Priority) Optimize gesture recognition for accuracy and speed.
- (🔴 High Priority) Debug issues with real-time processing and AI output.

Sprint 3 – Testing, Enhancements & Submission (Day 2)

- (🟡 Medium Priority) Test accuracy of gesture recognition and AI descriptions.
- (🟡 Medium Priority) Improve UI/UX for better user interaction.
- (🟢 Low Priority) Final demo preparation and deployment.

Phase-5: Project Development

Objective:

Implement core features of the gesture-based system.

Key Points:

Technology Stack Used:

- **Frontend**: Streamlit
- **Backend**: OpenCV, MediaPipe, Palm's text-bison-001
- **Programming Language**: Python

Development Process:

- Implement hand tracking and gesture classification.
- Integrate AI-generated descriptions for detected gestures.
- Optimize processing speed for real-time interaction.

Challenges & Fixes:

- **Challenge:** High latency in AI response.
  Fix: Optimize API calls and model processing speed.
- **Challenge:** Gesture misclassification in poor lighting.
  Fix: Improve preprocessing with adaptive thresholding.

Phase-6: Functional & Performance Testing

Objective:

Ensure the system functions as expected.

Test Cases:

Test Case ID	Category	Test Scenario	Expected Outcome	Status
TC-001	Functional Testing	Detect "thumbs up" gesture	Correct classification	☑ Passed
TC-002	Functional Testing	Generate AI description for "fist"	Relevant AI response	☑ Passed
TC-003	Performance Testing	Gesture recognition response time <500ms	Real-time detection	⚠ Needs Optimization
TC-004	Bug Fixes	Fix incorrect gesture classification	Higher accuracy	☑ Fixed
TC-005	UI Testing	Ensure UI responsiveness across devices	Works on mobile & desktop	❌ Failed - UI broken on mobile
TC-006	Deployment Testing	Host app using Streamlit Sharing	App accessible online	🚀 Deployed

---

Final Submission

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