Hackathon Project Phases Template for the Gesture-Based Human-Computer Interaction System using OpenCV, MediaPipe and Palm's text-bison-001 project.

Hackathon Project Phases Template

Project Title:

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

Team Name:

Tech-Trio

Team Members:

- Abhinaya Reddy Annadi
- Varsha Annam
- Dheekshitha Annam

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered gesture recognition system to enable intuitive and seamless human-computer interaction using hand gestures.

Key Points:

1. **Problem Statement:**
    - Traditional interfaces require physical interaction, limiting accessibility for individuals with disabilities and increasing hygiene concerns in public spaces.
    - Many users require a touchless, gesture-based control system for enhanced accessibility, gaming, and smart home applications.
2. **Proposed Solution:**
    - A gesture-controlled interface using OpenCV and MediaPipe for hand tracking, integrated with Palm's text-bison-001 for AI-powered contextual command recognition.
3. **Target Users:**
    - People with **DISABILITIES** needing touchless interaction.
    - Smart home and IoT users
    - Gamers requiring immersive experiences.
4. **Expected Outcome:**
    - A fully functional gesture-based interaction system capable of recognizing hand gestures and translating them into meaningful commands for various applications.

Phase-2: Requirement Analysis

Objective:

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

Key Points:

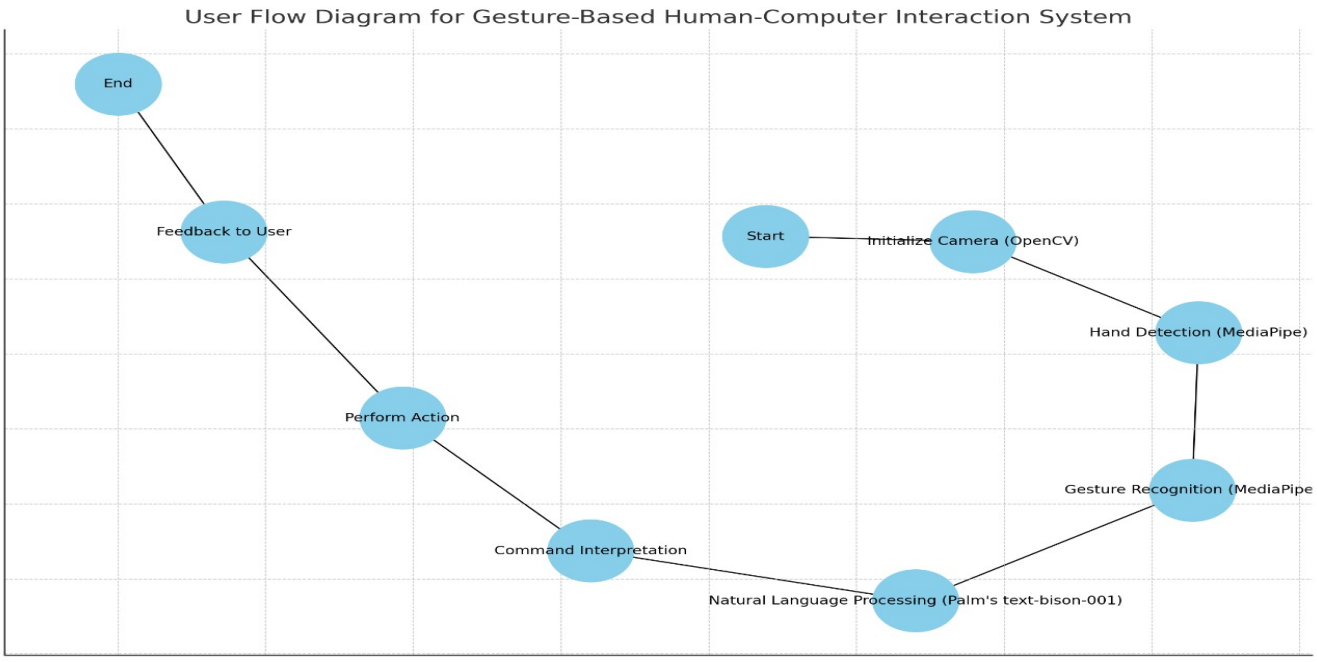
1. **Technical Requirements:**
    - Programming Language: **Python**
    - Computer Vision Library: **OpenCV, MediaPipe**
    - AI Model: **Palm’s text-bison-001**
    - Frontend: **Streamlit Web Framework**
    - Hardware: **Webcam for gesture tracking**

2. **Functional Requirements:**
    - Real-time hand gesture recognition.
    - AI-based gesture-to-command mapping.
    - Customizable gesture sets for different applications.
    - Seamless integration with external applications.
3. **Constraints & Challenges:**
    - Processing real-time gestures with minimal latency.
    - Achieving high accuracy in gesture recognition.
    - Handling variations in lightening and background.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - Webcam captures hand gestures.
    - OpenCV and MediaPipe process the video feed.
    - Recognized gestures are sent to Palm's text-bison-001.
    - AI interprets gestures and translates them into actions.
    - The system executes or forwards commands to external applications.
2. **User Flow:**
    - User performs a predefined gesture (e.g., moves hand to the right).
    - The system recognizes the gesture and displays "Right" on the screen.
    - The AI model refines the command based on context.
    - The command is executed (e.g., opening an application, adjusting volume, etc.).
3. **UI/UX Considerations:**
    - Intuitive UI for gesture training and customization
    - Real-time feedback on recognized gestures with displayed text (e.g., "Right" for right-hand movement)
    - Support for multiple gesture profiles.

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 & Library Integration	🔴 High	6 hours	End of Day 1	Varsha Annam	Python, OpenCV, MediaPipe	Basic framework setup
Sprint 1	Hand Tracking Implementation	🔴 High	3 hours	End of Day 1	Dheekshitha Annam	Web cam setup	Real time hand tracking
Sprint 2	AI Gesture command mapping	🔴 High	4 hours	Mid-Day 2	Abhinaya Annadi	Palm’s text-bison-001	AI powered command

							execution
Sprint 2	UI development	🟡 Medium	4 hours	Mid-Day 2	Varsha Annam	Gesture recognition module	Gesture training UI
Sprint 3	Error handling & debugging	🔴 High	2 hours	Mid-Day 2	Dheekshitha Annam & Abhinaya Annadi	API logs, UI inputs	Improved accuracy & stability
Sprint 3	Final Presentation & Deployment	🟢 Low	2 hours	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** & Library integration.
(🔴 **High Priority)** Hand Tracking Implementation.

Sprint 2 – Core Features & Debugging (Day 2)

(🔴 **High Priority)** AI Gesture command mapping.
(🟡 **Medium priority)** UI deployment.

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

(🔴 **High Priority)** Error handling &debugging.
(🟢 **Low Priority)** Final **demo preparation & deployment**.

Phase-5: Project Development

Objective:

Implement core features of the gesture-based interaction system.

Key Points:

1. **Technology Stack Used:**
    - **Frontend:** Streamlit
    - **Backend:** OpenCV, MediaPipe, Palm's text-bison-001
    - **Programming Language:** Python
    - **Hardware:** Standard Webcam.
2. **Development Process:**
    - Implement real-time hand tracking using OpenCV & MediaPipe.
    - Integrate Palm's AI model for gesture-to-command mapping.
    - Optimize performance for low latency.
3. **Challenges & Fixes:**
    - **Challenge: Poor accuracy in low-light conditions.**
      Fix: Implement adaptive brightness detection.
    - **Challenge: Gesture misclassification.**
      Fix: Fine-tune model with more training data.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Perform "Swipe Left" gesture	System executes left navigation	✅ Passed	Varsha Annam
TC-002	Functional Testing	Perform "stop" gesture	System executes stop navigation	✅ Passed	Dheekshitha Annam
TC-003	Performance Testing	Gesture recognition latency under 200ms	System responds in real-time	⚠️ Needs Optimization	Abhinaya Annadi
TC-004	Bug Fixes & Improvements	Fix incorrect gesture mapping	Works on different screen sizes	✅ Fixed	Developer

TC-005	Final Validation	Ensure UI responsiveness	UI should work on mobile & desktop.	❌ Failed - UI broken on mobile	Dheekshitha Annam
TC-006	Deployment Testing	Host system for public access	System 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**