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

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

Hackathon Project Phases Template

# Project Title:

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

# Team Name:

Nural Netwokers

# Team Members:

* R.Varun
* N.Bhadri Narayana
* M.Lokesh
* P.Rajendhar
* J.Anil

# Phase-1: Brainstorming & Ideation

## Objective:

To develop a **gesture-based human-computer interaction system** that enables users to control digital devices using **hand gestures**. By integrating **OpenCV, MediaPipe, and Palm's text-bison-001**, the system can **track hands, recognize gestures, interpret commands using AI, and execute actions** in real-time. This enhances **accessibility, touchless interaction, and intuitive control** for various applications. .

## 

## Key Points:

1. **Problem Statement:**
   * Traditional input methods like keyboards and mice can be **inconvenient, restrictive, and unhygienic** in certain environments. This project aims to develop a **gesture-based HCI system** using **OpenCV, MediaPipe, and Palm's text-bison-001** to enable **touchless, intuitive, and accessible interaction** with computers.Users also need guidance on vehicle maintenance and eco-friendly vehicle choices.
2. **Proposed Solution:**
   * Develop a **gesture-based HCI system** using **OpenCV, MediaPipe, and Palm's text-bison-001** to enable **real-time hand tracking, gesture recognition, and AI-powered command interpretation**. This system allows **touchless, intuitive, and accessible interaction**, improving efficiency and user experience.
3. **Target Users:**

**People with Disabilities** – Accessible alternative to traditional input.  
 **General Users** – Faster and intuitive computer interaction.  
 **Healthcare & Public Spaces** – Touchless control for hygiene.  
 **Smart Home & IoT Users** – Hands-free device control.  
 **Gamers & Developers** – Gesture-based controls for immersion.

Ideal for anyone seeking **natural, efficient, and touch-free interaction** with computers.

1. **Expected Outcome:**

The system provides **real-time hand tracking, accurate gesture recognition, AI-driven intent processing, and seamless command execution**, enabling **touchless human-computer interaction** with smooth performance and user feedback.

# Phase-2: Requirement Analysis

## Objective:

Develop a **gesture-based HCI system** using **OpenCV, MediaPipe, and Palm's text-bison-001** for **real-time hand tracking, gesture recognition, and AI-powered command execution**, enabling **touchless and intuitive interaction** with computers.

## Key Points:

1. **Technical Requirements:**

**Programming Language:** Python 3.8+  
**Libraries & Frameworks:** OpenCV, MediaPipe, TensorFlow, PyAutoGUI, Google Generative AI (Palm's text-bison-001)  
**Hardware:** Webcam for real-time hand tracking  
 **Software & Tools:** VS Code/PyCharm, Google AI API, NumPy  
 **System Compatibility:** Windows/Linux/Mac with Python environment

This ensures a **robust, efficient, and AI-powered gesture-based interaction system**.

1. **Functional Requirements:**

**Real-time Hand Tracking** – Detect and track hand movements using **MediaPipe**.  
 **Gesture Recognition** – Identify predefined gestures for executing system commands.  
 **AI-Powered Interpretation** – Use **Palm’s text-bison-001** to process and interpret   
 **System Control** – Perform actions like opening apps, controlling media.   
 **Touchless Interaction** – Enable hands-free control for accessibility and hygiene.  
 **User Customization** – Allow users to define custom gestures and corresponding actions.

This ensures an **intelligent, responsive, and user-friendly** gesture-based HCI system.

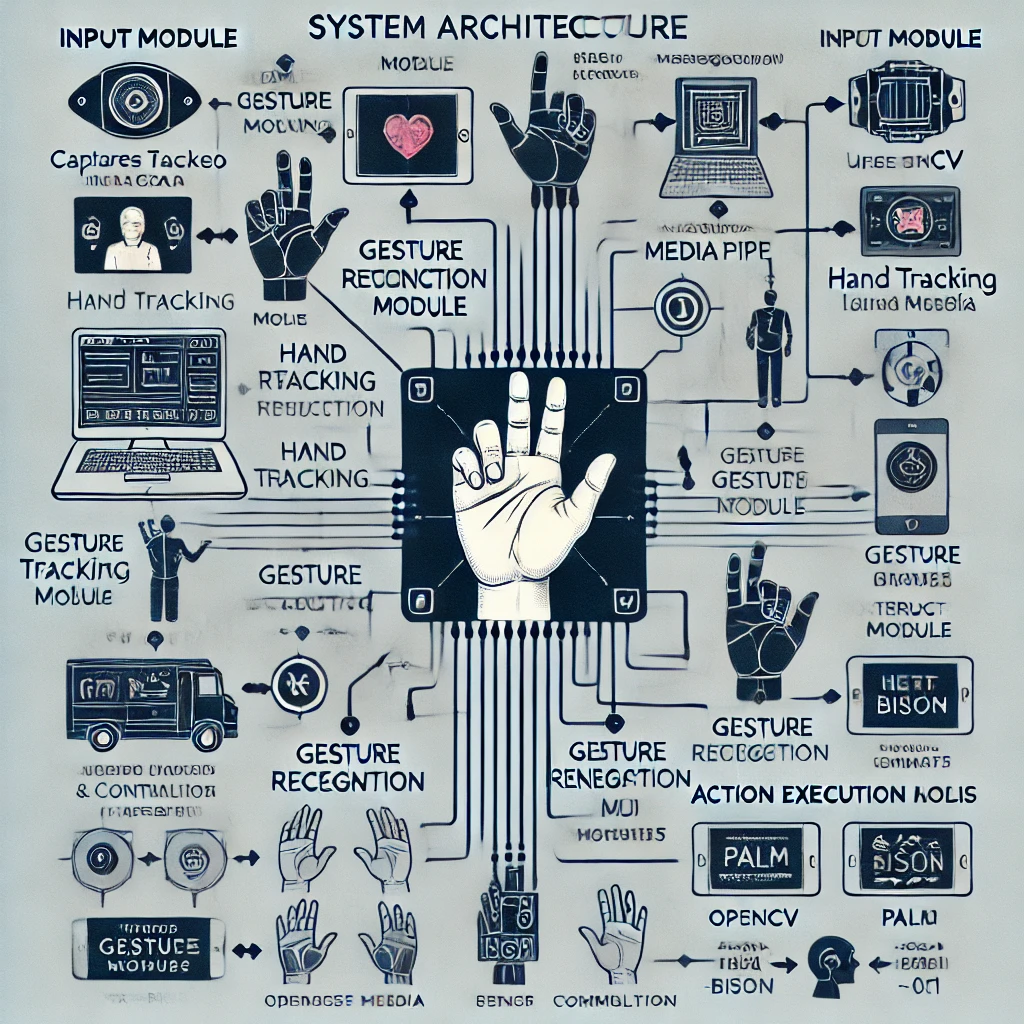
1. **Constraints & Challenges:**

**Lighting Conditions** – Performance may be affected by poor lighting or shadows.  
 **Background Noise** – Complex backgrounds can interfere with hand tracking.  
 **Real-time Processing** – Ensuring low-latency gesture recognition for smooth interaction.  
 **Hardware Limitations** – Requires a good webcam and sufficient processing power.  
 **Gesture Variability** – Differences in hand size, orientation, and movement may impact accuracy.  
 **API Limitations** – Dependence on **Palm's text-bison-001** for AI interpretation requires internet access.  
 **User Adaptation** – Users may need time to learn and adapt to gesture-based controls.

# Phase-3: Project Design

## Objective:

Design an **efficient gesture-based HCI system** using **OpenCV, MediaPipe, and Palm’s text-bison-001** for **real-time hand tracking, gesture recognition, and AI-powered command execution**, enabling **touchless interaction** with computers.



## Key Points:

1. **System Architecture:**

**Input Module** – Captures video via webcam.  
 **Hand Tracking** – Uses **MediaPipe** for landmark detection.  
 **Gesture Recognition** – Uses **OpenCV & ML** to classify gestures.  
 **AI Interpretation** – Uses **Palm’s text-bison-001** to process gestures.  
 **Action Execution** – Maps gestures to system commands.  
 **User Interface** – Displays feedback and recognized gestures.

1. **User Flow:**

1️⃣ **Start System** → Webcam captures real-time video.  
2️⃣ **Hand Tracking** → MediaPipe detects hand landmarks.  
3️⃣ **Gesture Recognition** → OpenCV identifies gestures.  
4️⃣ **AI Processing** → Palm’s text-bison-001 interprets intent.  
5️⃣ **Execute Command** → System performs the mapped action.  
6️⃣ **User Feedback** → UI displays recognized gestures & actions.

1. **UI/UX Considerations:**

**Simple & Intuitive UI** – Real-time gesture feedback.  
 **Live Gesture Preview** – Displays detected hand gestures.  
 **Command Logs** – Shows executed actions.  
 **Customization** – Users can define gestures.  
 **Error Handling** – Alerts for unrecognized gestures.  
 **Accessibility** – High contrast, large fonts, voice assistance.

# Phase-4: Project Planning (Agile Methodologies)

## Objective:

To implement an **Agile-based project planning approach** for the **Gesture-Based HCI System**, ensuring **flexibility, iterative development, and continuous improvement** through **sprints, user feedback, and rapid prototyping**.

## Sprint Planning with Priorities

**Sprint 1 – Setup & Integration (Day 1)**

**Environment Setup** – Install Python, OpenCV, MediaPipe, and required libraries.  
**Project Structure** – Organize files, directories, and codebase.  
 **Webcam Integration** – Capture real-time video feed.  
 **MediaPipe Hand Tracking** – Implement and test hand landmark detection.  
 **Basic UI Setup** – Display webcam feed and detected hand landmarks.

## 

## Sprint 2 – Core Features & Debugging (Day 2)

## ****Gesture Recognition**** – Implement gesture classification using OpenCV & MediaPipe. ****AI Integration**** – Connect ****Palm’s text-bison-001**** for gesture interpretation. ****Command Execution**** – Map recognized gestures to system actions (e.g., opening apps, media control). ****Debugging & Optimization**** – Improve tracking accuracy and reduce latency. ****Basic UI Feedback**** – Display recognized gestures in real-time.

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

## ****System Testing**** – Evaluate accuracy, speed, and stability of gesture recognition. ****Error Handling**** – Improve detection for unclear or unintended gestures. ****User Experience Enhancements**** – Refine UI, add logs, and improve feedback. ****Performance Optimization**** – Reduce processing lag for real-time interaction. ****Final Debugging & Documentation**** – Ensure smooth execution and prepare project reports. ****Project Submission**** – Package and submit the final system.

# Phase-5: Project Development

## Objective:

Develop a **real-time gesture-based HCI system** using **OpenCV, MediaPipe, and Palm’s text-bison-001** for **accurate gesture recognition, AI-driven interpretation, and seamless touchless interaction**. .

## Key Points:

1. **Technology Stack Used:**

**Python 3.8+** | **OpenCV & MediaPipe** (Vision) | **Palm’s text-bison-001** (AI)  
 **TensorFlow** (ML) | **PyAutoGUI** (Automation) | **Webcam** (Input)

1. **Development Process:**

**Setup** → Install dependencies & configure OpenCV, MediaPipe.  
 **Hand Tracking** → Implement real-time detection.  
 **Gesture Recognition** → Classify gestures with ML.  
 **AI Processing** → Use **Palm’s text-bison-001** for interpretation.  
 **Command Execution** → Map gestures to system actions.  
 **Testing & Optimization** → Debug & enhance performance.  
 **UI & Feedback** → Display recognized gestures in real-time.

1. **Challenges & Fixes:**

**Lighting & Noise** – Adaptive thresholding & ROI for better detection.  
 **Gesture Variability** – Train models on diverse hand shapes.  
 **Lag & Performance** – Optimize code & use lightweight models.  
 **Misinterpretation** – Improve AI filtering for accuracy.  
 **User Adaptation** – Add visual cues & tutorials.

# 

# Phase-6: Functional & Performance Testing

## Objective:

Ensure the **accuracy, efficiency, and reliability** of the **gesture-based HCI system** through **functional and performance testing**, optimizing recognition, AI processing, and responsiveness.

# Final Submission

1. Complete & Tested System
2. Project Documentation
3. **Demo & Presentation**
4. **Demo & Presentation**