# **Hackathon Project Phases Template**

## **Project Title:**

**Gesture based human interacting system using opencv ,mediapipe ,palm- text-bison-001**

## **Team Name:**

AI intelligest

## **Team Members:**

* E.V.Udayasree-23RH1A6686
* G.Sahithi-23RH1A6693
* G.Bhavana-23RH1A6692
* G.Rupasree-23RH1A6691

**Phase-2:** Brainstorming & Ideation for a Gesture-Based Human-Computer Interaction System

**Objective**

Develop an AI-powered gesture-based human-computer interaction system using OpenCV, MediaPipe, and Palm Text Bison 001. This system will allow users to interact with computers and devices through hand gestures, reducing reliance on physical input methods and enhancing accessibility, efficiency, and hands-free control across various industries, including healthcare, smart homes, gaming, and automation.

**Key points :**

1. **Problem Statement :**

**a.** Limited Accessibility & Convenience: Traditional input devices like keyboards and

mice are not always intuitive, accessible, or hygienic, especially in specialized

environments.

**b.** Need for Hands-Free Control: Industries such as healthcare, smart homes, and

gaming require gesture-based interaction to improve usability and efficiency.

1. **Proposed Solution**

**a**. Real-Time Gesture Recognition: Utilize MediaPipe for hand tracking and OpenCV

for gesture detection to enable contactless interaction.Use Palm Text Bison 001 to

interpret gestures into commands, text, or system controls based on context.

1. **Core Features**

**a.** Gesture-Based System Navigation: Move the cursor, scroll, open applications, and

interact with devices using hand movements.

**b.** Customizable Gestures: Train AI to recognize personalized gestures for specific

use cases like medical applications, smart home control, or gaming.

**4.** **Integration & Use Cases**

**a**. Healthcare Industry

i. Contactless Medical Interfaces: Doctors can control patient records,

medical images, or robotic surgery tools without touching surfaces.

ii.Assistive Communication: Patients with paralysis or speech

impairments can use gestures to communicate or control devices.

**b**. Gaming & Virtual Reality (VR)

i.Immersive Gesture-Based Controls: Replace traditional controllers

with hand movements for gaming and VR applications.

ii.Fitness & Rehabilitation Games: AI tracks hand movements for

exercise monitoring and physical therapy.

**5. Preferences & Target Users**

**a.** Healthcare Professionals & Patients: Doctors, nurses, and physically

challenged individuals who need touch-free interaction.

**b.** Gamers & VR Users: Those wanting natural and immersive gameplay experiences.

**6. Expected Outcome**

**a.** Enhanced user experience across different sectors with gesture-based

Interaction.Increased accessibility for individuals with disabilities or mobility issues.

Greater efficiency and hygiene in healthcare and industrial settings.

**7. Technical Approach**

**a.**Gesture Tracking & Recognition: MediaPipe detects 21 hand landmarks,

OpenCV processes and classifies gestures.

**b.**AI-Powered Gesture Interpretation: Palm Text Bison 001 translates gestures

into commands, text, or spoken output.

## **Phase-2: Requirement Analysis**

### **Objective:**

### Define the technical and functional requirements for Gesture-Based Human-Computer Interaction

### **Key Points:**

1. **Technical Requirements:**
   1. **Gesture Recognition Models:** OpenCV (for image processing), MediaPipe Palm-Text-Bison-001
   2. **Backend :** python(Flask)
   3. **Frontend:** Simple GUI, and website based.
   4. **Database:** No initial database
   5. **Deployment:** Cloud-based (AWS, Google Cloud, Azure) & local execution
   6. **APIs & Integrations:** MediaPipe , OpenCV,
2. **Functional Requirements:**

**a.** **Accurate Recognition:** Real-time detection of gestures, poses, and expressions.

**b.** **Custom Gesture Mapping:** Personalized commands with context-aware interpretation.

**c.** **Hands-Free Navigation:** Control UI and navigate apps using gestures.

**d.** **Real-Time Feedback:** Visual feedback and performance tips for users.

**e.** **Accessibility Support:** Multi-language and disability-friendly features.

1. **Constraints & Challenges:**

**Real-time Performance and Latency:**

* + Ensuring low-latency gesture recognition for a seamless user experience.
  + Efficient processing on edge devices with limited computational power.

**Accuracy and Reliability:**

* + Maintaining high accuracy in gesture detection across diverse environments and lighting conditions.
  + Reducing false positives and negatives in gesture recognition.

**Scalability and Integration:**

* + Efficient handling of large-scale gesture data in cloud and local environments.
  + Seamless integration with various devices (e.g., AR/VR headsets, smart TVs).

**Security and Privacy:**

* + Ensuring secure processing of user data, especially facial and hand landmarks.
  + Complying with data privacy regulations (e.g., GDPR, CCPA).

**User Adaptation and Learning Curve:**

* + Designing intuitive gesture sets for easy user adoption.
  + Providing user guidance and tutorials for effective interaction.

**Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow for a gesture-based human-computer interaction system using OpenCV, MediaPipe, and Palm Text Bison 001.Key Points:

**System Architecture:**

1. User performs a hand gesture that is captured via a camera and processed in real time.
2. Gesture data is analyzed using OpenCV & MediaPipe, then interpreted by Palm Text Bison 001 AI model.
3. The AI processes the gesture and executes commands for system control, text input, or device interaction.
4. Results are displayed on the user interface, executing the intended action.

**User Flow:**

1. Step 1: User performs a hand gesture (e.g., swipe left to switch applications).
2. Step 2: The system detects and tracks the gesture using MediaPipe’s Hand Tracking Model.
3. Step 3: OpenCV processes the hand landmarks, and Palm Text Bison 001 translates it into a specific command.
4. Step 4: The system executes the command and displays the response on the interface..

**UI/UX Considerations:**

1. Minimalist and intuitive interface with real-time gesture feedback visualization.
2. Customizable gestures for different applications, ensuring a personalized user experience.

## 

## **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 | Setup | 🔴 High | 6 hours (Day 1) | End of Day 1 | Shanawaz | Python | Working on commands |
| Sprint 1 | Integration | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Member 2 | UI elements format finalized | Basic UI with input fields |
| Sprint 2 | Core features | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | anwar | UI elements ready | Search functionality with filters |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Member 1&4 | UI inputs | Improved UI 9 stability |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Member 2& 3 | ,UI layout completed | Responsive UI, better user experience |
| 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 development environment and install dependence (OpenCV, MediaPipe, Palm-Text-Bison-001 API).
  + High Priority – Integrate MediaPipe Hand Tracking for gesture recognition. 🟡 Medium Priority – Build a basic UI for visualizing detected gestures.

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

* High Priority – Implement gesture recognition logic using MediaPipe and OpenCV.
* High Priority – Integrate Palm-Text-Bison-001 for gesture-to-text translation.
* High Priority – Debug and refine gesture detection accuracy.

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

* Medium Priority – Test API responses, refine UI, and fix UI bugs.
* Medium Priority – Optimize system performance for real-time interaction.
* Low Priority – Final demo preparation and deployment.

## **Phase-5: Project Development**

**Objective:**

Implement core features of Code Genie.

**Key Points:**

**1.Technology Stack Used:**

* AI Model: Chatgpt
* Frontend: Simple GUI and website based
* Backend: Python(Flask)
* Database: No initial database
* APIs & Integrations: GitHub API, VS Code Extension

**2. Development Process:**

* Implement API key authentication and LLM model integration.
* Develop real-time code generation, debugging, and explanations.
* Optimize model responses for low-latency inference.

**3. Challenges & Fixes:**

* Challenge: High computational cost for large model inference.  
  Fix: Implement caching for frequently used prompts.
* Challenge: Ensuring accuracy of AI-generated code.  
  Fix: Use reinforcement learning from user feedback.

## **Phase-6: Functional & Performance Testing**

### **Objective:**

Ensure that Code Genie works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Generate a Python Flask UI | Returns valid Python code | ✅ Passed | Tester 1 |
| TC-002 | Functional Testing | Debug a HTML function | |  | | --- | | AI suggests correct  debugging steps |  |  | | --- | |  | | ✅ Passed | Tester 2 |
| TC-003 | Performance Testing | Website slow response | Slow response time | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect Website suggestions | Improved accuracy | ✅ Fixed | Developer |
| TC-005 | Final Validation | Ensure UI is responsive across devices | Works on desktop | ❌ Failed - UI broken on mobile | Tester 2 |
| TC-006 | Deployment Testing | Deploy the web app on cloud | Successfully deployed | 🚀 Deployed | DevOps |

## **Final Submission**

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