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

**Project Title:**

Gesture-Based Human-Computer Interaction System

**Team Name:**

Stardust Coders

**Team Members:**

A YuvaKarthik  
Bakshi Vaishvik  
P Aditya Praneet  
Prabhdeep Singh

**Phase-1: Brainstorming & Ideation**

**Objective:**

Develop a real-time hand gesture recognition system using OpenCV and MediaPipe, integrated with a generative AI model (Palm's text-bison-001) to provide descriptive feedback.

**Key Points:**

**Problem Statement:**

* Traditional human-computer interaction relies on touch and physical devices.
* There is a need for touchless control mechanisms for interactive applications, accessibility, and gaming.

**Proposed Solution:**

* A real-time gesture recognition system that detects predefined and customizable gestures.
* AI-generated descriptions provide contextual feedback based on recognized gestures.
* Seamless interaction through a user-friendly Streamlit interface.

**Target Users:**

* Users requiring **touchless control** in smart environments.
* **Gamers** seeking gesture-based gaming controls.
* **Individuals with disabilities** who need alternative interaction methods.
* **Developers** looking for gesture recognition solutions.

**Expected Outcome:**

* A **functional AI-powered gesture recognition system** with real-time feedback.
* Integration of generative AI for **intelligent response generation**.
* Deployment on **AWS** for accessibility and scalability.

**Phase-2: Requirement Analysis**

**Objective:**

The objective of the **Requirement Analysis** phase is to establish a clear understanding of the technical and functional specifications for the **Gesture-Based HCI System**. This phase ensures that the system meets performance expectations, aligns with user needs, and addresses potential constraints. By defining these requirements, we create a solid foundation for development, testing, and deployment, ultimately ensuring a seamless and interactive gesture-based experience.

**Key Points:**

**Technical Requirements:**

* **Programming Language:** Python
* **Backend:** OpenCV & MediaPipe for gesture recognition
* **AI Model:** Palm's text-bison-001 for generating descriptions
* **Frontend:** Streamlit for visualization
* **Cloud Hosting:** AWS for deployment

**Functional Requirements:**

* Real-time hand tracking and gesture recognition.
* Predefined gestures (e.g., thumbs up, open hand, fist) with customization options.
* AI-generated descriptions for recognized gestures.
* Interactive visualization of gestures in the UI.

**Constraints & Challenges:**

* Ensuring **real-time processing** with low latency.
* Handling **dynamic lighting and camera angles**.
* Optimizing AI-generated narratives for **relevance and accuracy**.

**Phase-3: Project Design**

**Objective:**

Develop the architecture and user flow of the application.

**Key Points:**

**System Architecture:**

1. **User performs a hand gesture in front of the camera**.
2. **OpenCV & MediaPipe track and classify the gesture**.
3. **Recognized gesture is sent to Palm's text-bison-001** for description generation.
4. **Streamlit UI displays** the real-time gesture recognition and AI-generated description.

**User Flow:**

1. **Step 1:** User interacts with the camera using gestures.
2. **Step 2:** The backend processes the gesture and classifies it.
3. **Step 3:** AI generates a descriptive narrative for the recognized gesture.
4. **Step 4:** The UI updates and provides real-time feedback.

**UI/UX Considerations:**

* **Minimalist, interactive interface** for better user engagement.
* **Real-time visualization** of gestures.
* **Accessibility-friendly design** for diverse users.

**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 |  | Python, OpenCV, MediaPipe | Basic gesture detection setup |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours | End of Day 1 |  | OpenCV setup | Basic UI with camera input |
| Sprint 2 | Gesture Recognition Implementation | 🔴 High | 3 hours | Mid-Day 2 |  | MediaPipe setup | Predefined gestures working |
| Sprint 2 | AI Integration for Descriptions | 🔴 High | 3 hours | Mid-Day 2 |  | Gesture classification ready | AI generates descriptions |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours | Mid-Day 2 |  | Gesture + AI output ready | Smooth user experience |
| Sprint 3 | Final Deployment | 🟢 Low | 1 hour | End of Day 2 |  | Functional prototype | Demo-ready project |

**Phase-5: Project Development**

**Objective:**

Implement core features of the Gesture-Based HCI System.

**Key Points:**

**Technology Stack Used:**

* **Frontend:** Streamlit
* **Backend:** OpenCV, MediaPipe
* **AI Model:** Palm's text-bison-001
* **Cloud:** AWS for deployment
* **Programming Language:** Python

**Development Process:**

* Implement real-time hand tracking and gesture classification.
* Integrate AI for generating descriptions.
* Optimize for real-time responsiveness.

**Challenges & Fixes:**

| **Challenge** | **Fix** |
| --- | --- |
| Latency in processing gestures | Optimized frame rate and model efficiency |
| Inconsistent gesture detection | Adjusted detection thresholds & lighting adaptation |
| AI-generated text sometimes irrelevant | Fine-tuned AI prompts for better responses |

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

**Objective:**

Ensure that the Gesture-Based HCI System works as expected.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | Perform 'thumbs up' gesture | AI should recognize and describe it |  |  |
| TC-002 | Functional Testing | Perform 'fist' gesture | AI should recognize and describe it |  |  |
| TC-003 | Performance Testing | Ensure recognition speed under 500ms | System should process gestures in real time |  |  |
| TC-004 | UI Testing | Ensure Streamlit UI updates instantly | UI should reflect recognized gestures |  |  |
| TC-005 | Deployment Testing | Host the app on AWS | App should be accessible online |  |  |

**Final Submission**

* **Project Report**
* **Demo Video (3-5 Minutes)**
* **GitHub/Code Repository Link**
* **Presentation**