**GEN AI HACKATHON**

**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:**

Develop a Gesture-Based Human-Computer Interaction System using MediaPipe and OpenCV that recognizes real-time hand gestures (e.g., thumbs up, fist, open hand) and integrates with a generative AI model to provide descriptive narratives. The solution should feature a user-friendly Streamlit interface, enabling touchless control, interactive gaming, and assistive technologies.

**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 that require 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:**

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:**

This phase focuses on designing a robust system architecture that ensures seamless interaction between the user, gesture recognition module, AI-generated descriptions, and the user interface. By mapping out the user flow and UI/UX considerations, this phase lays the groundwork for an intuitive, responsive, and efficient application.

**Key Points:**

**System Architecture:**

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

**User Flow:**

* Step 1: User interacts with the camera using gestures.
* Step 2: The backend processes the gesture and classifies it.
* Step 3: AI generates a descriptive narrative for the recognized gesture.
* 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.
* Customization of the actions for the 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 | Aditya | Python, OpenCV, MediaPipe | Basic gesture detection setup |
| Sprint 1 | Frontend UI Development | 🟡Medium | 2 hours | End of Day 1 | Prabhdeep | StreamLit,  OpenCV setup | Basic UI with camera input |
| Sprint 2 | Gesture Recognition Implementation | 🔴 High | 3 hours | End of Day 1 | Vaishvik | MediaPipe setup | Predefined gestures working |
| Sprint 2 | AI Integration for Descriptions | 🔴 High | 3 hours | Mid-Day 2 | Yuvakarthik | Gesture classification ready | AI generates descriptions |
| Sprint 3 | Testing & UI Enhancements | 🟡Medium | 1.5 hours | Mid-Day 2 | All members | Gesture + AI output ready | Smooth user experience |
| Sprint 3 | Final Deployment | 🟢 Low | 1 hour | Mid-Day 2 | All members | Functional prototype | Demo-ready project |

**Phase-5: Project Development**

**Objective:**

This phase focuses on building a robust and efficient system by leveraging the chosen technology stack while addressing key challenges such as processing latency, gesture detection accuracy, and AI output relevance. The goal is to develop a fully functional prototype that meets performance expectations and user requirements.

**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:**

To 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 | Done | Vaishvik |
| TC-002 | Functional Testing | Perform 'fist' gesture | AI should recognize and describe it | Done | Aditya |
| TC-003 | Performance Testing | Ensure recognition speed under 500ms | System should process gestures in real time | Done | Yuvakarthik |
| TC-004 | UI Testing | Ensure Streamlit UI updates instantly | UI should reflect recognized gestures | Done | Prabhdeep |
| TC-005 | Deployment Testing | Host the app on AWS | App should be accessible online | Done | Vaishvik |

**Final Submission**

* Project Report
* Demo Video (3-5 Minutes)
* GitHub/Code Repository Link : https://github.com/yuva-karthik/No-touch
* Presentation