**Hackathon Project Documentation**

**Hackathon Project Phases**

**Project Title:**

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

**Team Name:**

SKAD

**Team Members:**

●​ D.Sai Srujan

●​ K.Shanthan

●​ S.Sreenidhi

●​ A Sri Teja

**Phase-1: Brainstorming & Ideation**

**Objective:**

The **Gesture-Based Human-Computer Interaction System** enables seamless interaction with computers using real-time hand gesture recognition. By leveraging **OpenCV** and **MediaPipe**, the system accurately detects and interprets various gestures, such as a thumbs-up, fist, or open hand. Integrated with **Palm's text-bison-001**, a generative AI model, it enhances user experience by providing descriptive insights for each recognized gesture. A **user-friendly Streamlit interface** ensures intuitive interaction and visualization, making the system ideal for **touchless control, interactive gaming, and assistive technologies**.

.

**Key Points:**

1.​ **Problem Statement:​**

* Traditional public kiosks in **airports, museums, and malls** rely on touchscreens, which can be:
  + **Unhygienic**, contributing to the spread of germs, especially during health crises.
  + **Inaccessible** for users with mobility impairments or disabilities.
  + **Prone to wear and tear**, leading to high maintenance costs.
* A **gesture-based human-computer interaction system** offers a **touchless alternative** by:
  + Recognizing **intuitive hand gestures** like pointing, swiping, and fist-clenching for navigation.
  + Reducing the risk of germ transmission, ensuring a more **hygienic environment**.
  + Enhancing **user accessibility**, making kiosks usable for a wider audience.
  + Improving durability by eliminating the need for physical contact.
* Using **OpenCV, MediaPipe, and AI-driven gesture recognition**, the system enables **seamless, efficient, and user-friendly interaction**, making public kiosks **more hygienic, accessible, and future-ready**.

2.​ **Proposed Solution:**

To address the limitations of traditional touch-based kiosks, we propose a **Gesture-Based Human-Computer Interaction System** that enables users to interact with public kiosks using **hand gestures** instead of physical touch. The system will leverage **computer vision and AI** to provide a seamless, hygienic, and accessible experience.

**Key Components of the Solution:**

🔹 **Real-time gesture Recognition**

* Uses **OpenCV** and **MediaPipe** to detect and track hand gestures.
* Recognizes gestures like **"open hand"** for scrolling, **"fist"** for selection, and **"pointing"** for highlighting options.

🔹 **AI-powered interpretation**

* Integrates **Palm’s text-bison-001** to provide descriptive narratives for each recognized gesture.
* Enhances interaction by offering intelligent responses or guidance based on user input.

🔹 **Touchless & Hygienic Interaction**

* Eliminates the need for physical touch, reducing the risk of germ transmission.
* Ideal for **public kiosks in airports, malls, museums, hospitals, and other high-traffic areas**.

🔹 **User-Friendly Interface**

* Built with **Streamlit** for easy visualization and interaction.
* Provides real-time feedback to users, ensuring an intuitive experience.

🔹 **Enhanced Accessibility**

* Enables **people with mobility impairments** to interact with kiosks effortlessly.
* Makes public systems more inclusive and user-friendly for a wider audience.

3.​ **Target Users:​**

🔹 **Public Kiosk Users** – Travelers, shoppers, and visitors who need a **touchless** and

**hygienic** way to interact with kiosks in **airports, malls, museums, and hospitals**.

🔹 **People with Disabilities** – Individuals with **mobility impairments** or **limited dexterity**

who may find touchscreens difficult to use.

🔹 **Healthcare & Public Safety Officials** – Professionals promoting **hygienic solutions** in

high-traffic areas to reduce the spread of germs.

🔹 **Retail & Hospitality Industry** – Businesses using self-service kiosks for **ordering,**

**check-in, or information retrieval** to enhance customer experience.

🔹 **Gaming & Interactive Entertainment Users** – Gamers and users of **virtual or**

**augmented reality systems** who prefer **gesture-based** controls for immersive

experiences.

4.​ **Expected Outcome:​**

🔹 **Hygienic & Touchless Interaction** – Eliminates the need for physical touch, reducing

germ transmission.

🔹 **Improved Accessibility** – Enables seamless interaction for users with disabilities or

mobility impairments.

🔹 **Enhanced User Experience** – Provides an intuitive and efficient way to navigate kiosks

using hand gestures.

🔹 **Reduced Maintenance Costs** – Minimizes wear and tear, lowering the need for

frequent repairs.

**Phase-2: Requirement Analysis**

**Objective:**

Define the technical and functional requirements for the Gesture-Based Human-Computer Interaction System using OpenCV, MediaPipe and Palm's text-bison-001

**Key Points:**

**🔹 Technical Requirements:**

1. **Programming Language:** Python
2. **Computer Vision:** OpenCV, MediaPipe
3. **AI Integration:** Google Gemini API
4. **Frontend:** Streamlit Web Framework, WebRTC
5. **System Automation:** PyAutoGUI

**🔹 Functional Requirements:**

1. **Gesture Recognition & Processing** – Detects and interprets hand gestures like **open hand, fist, pointing, and swipe** for navigation and selection.
2. **User Interaction & Feedback** – Provides **real-time feedback** on recognized gestures and AI-generated responses for better engagement.
3. **User-Friendly Interface** – Uses a **Streamlit-based UI** for seamless interaction with clear instructions for users.

**🔹 Constraints & Challenges:​**

1. Ensuring **real-time gesture recognition** with OpenCV and MediaPipe.
2. Handling **latency issues** in AI-generated responses from Palm’s text-bison-001.
3. Optimizing **hardware performance** for smooth processing on low-end devices

.

**Phase-3: Project Design**

**Objective:**

Developed the architecture and user flow of the application.

A diagram of a system

AI-generated content may be incorrect.

**1. System Architecture:**

* User performs hand gestures in front of the camera.
* **MediaPipe Hands** detects static and dynamic gestures.
* **Hand Gesture Recognition** processes the gesture data.
* The system triggers actions or requests AI-generated responses.
* The **Google Gemini API** generates AI responses and stores them in a cache.
* Processed results or cached responses are displayed via the **Streamlit UI**.

**2. User Flow:**

* **Step 1:** User performs a hand gesture (e.g., open palm, pointing).
* **Step 2:** The system recognizes the gesture using **MediaPipe & OpenCV**.
* **Step 3:** If the gesture requires an AI response, it calls the **Google Gemini API**.
* **Step 4:** The response is processed and displayed on the **Streamlit UI**.
* **Step 5:** If the gesture corresponds to a system action (e.g., swipe, select), **PyAutoGUI** triggers the respective action.

**3. UI/UX Considerations:**

* **Minimalist, intuitive interface** with real-time hand gesture feedback.
* **Gesture-based controls** for system interactions (e.g., swipe to navigate, fist to select).
* **Seamless video streaming** via **WebRTC Streamer** for real-time recognition.
* **Optimized processing** for smooth user experience and responsiveness.

**Phase-4: Project Planning (Agile Methodologies)**

**Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | **Expected** |  |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Outcome** |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Detect landmarks on the hand and labeing them |  |
|  | Recognize the hand and add landmarks to hand |  | 4 hours | Mid-Day 1 |  | Python , Mediapipe & OpenCV |  |  |
| Sprint 1 |  | High | (Day 1) |  | Entire Team |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Sprint 1 | Identifies the gestures based on hand movements |  | 2 hours | Mid-Day 1 | Sri Teja | Python,Mediapipe | Displays the hand gesture names based on hand movements |  |
|  | Medium | (Day 1) |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | Describe the identified gestures |  | 1 hours |  |  | Python,  GeminiAPI | Describe about the hand gestures |  |
| Sprint 2 |  | Medium | (Day 1) | End of Day 1 | Sreenidhi |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | Based on the gestures perform some actions |  | 1 hours |  |  | Python,Pyautogui | Perform actions based on hand movements |  |
| Sprint 3 | (Swipe,Scroll,etc.) | High | (Day 2) | Mid-Day 2 | Sai Srujan |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Responsive UI, |  |
| Sprint 3 | UI Development, UI |  | 2 hours | Mid-Day 2 | Shanthan | Python,Streamlit | better user |  |
| Enhancements & Testing | High | (Day 2) |  | experience |  |
|  |  |  |  |  |  |  |  |  |
|  | Final Presentation |  | 1 hour | End of Day |  | Working | Demo-ready |  |
| Sprint 4 | & Deployment | Low | (Day 2) | 2 | Entire Team | prototype | project |  |
|  |  |  |  |  |  |  |  |  |

**Sprint Planning with Priorities**

**Sprint 1 – Recognize the hand and add landmarks to hand (Day 1)**

**(High Priority)** Captures the hand using python’s mediapipe and adds landmarks to the hand.

**(Medium Priority)** Identifies the gestures based on hand movements.

**Sprint 2 – Describe the identified gestures (Day 2)**

**(Medium Priority)** Describes the identified gestures.​

**Sprint 3 –Perform actions based on hand gestures & Testing (Day 2)**

**(High Priority)** Based on the gestures perform actions like scroll, swipe etc.

**(High Priority)** Test API responses, refine UI, & fix UI bugs.​

**Sprint 4 – Presentation and Deployment (Day 2)**

**(Low Priority)** Final presentation of the project and Deployment.

**Phase-5: Project Development**

**Objective:**

Implement hand gesture recognition and perform actions using hand movements without touching the screen.

**Key Points:**

1.​ **Technology Stack Used:​**

○​ **Frontend:** Streamlit

○​ **Backend:** Mediapipe,Pyautogui

○​ **Programming Language:** Python

2.​ **Development Process:​**

○​ Implement **Gemini API integration**.

○​ Develop hand gesture recognition using Mediapipe to recognize the landmarks on hand

○​ Implement actions to be performed using various gestures

3.​ **Challenges & Fixes:​**

○​ **Challenge:** Delayed API response times and limited API calls per minute.​

**Fix:** Implement **caching** to store frequently queried results, Optimize queries to fetch **only necessary data**.

○​ **Challenge:** Colliding hand gestures.​

**Fix:** Add more constraints on landmarks.

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

**Objective:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | |  | |  | |  | |  | |  | |
| **Case ID** | | **Category** | | **Test Scenario** | | **Expected Outcome** | | **Status** | | **Tester** | |
|  | |  | |  | |  | |  | |  | |
|  | | Functional | | Recognize hand & describe | | Identify the landmarks on the hand and display & describe the gestures on the hand | |  | | Sri Teja | |
| TC-001 | | Testing | |  | |  | | Passed | |  | |
|  | |  | |  | |  | |  | |  | |
|  | |  | | Perform actions | |  | |  | |  | |
|  | | Functional | |  | | Based on the hand movements perform the actions | |  | |  | |
| TC-002 | | Testing | |  | |  | | Passed | | Sai Srujan | |
|  | |  | |  | |  | |  | |  | |
|  | | Performance | | API response time under | | API should return | | Needs | |  | |  | |
| TC-003 | | Testing | | 500ms | | results quickly. | | Optimization | | Shanthan | |  | |
|  | |  | |  | |  | |  | |  | |  | |
|  | | Bug Fixes & | | Fixed incorrect API | | Data accuracy should | |  | | Shanthan& | |  | |
| TC-004 | | Improvements | | responses. | | be improved. | | Fixed | | Sreenidhi | |  | |
|  | |  | |  | |  | |  | |  | |  | |
| TC-005 | | Final | | Ensure UI is responsive | | UI should work on | | Failed - UI | | Sreenidhi | |  | |
| Validation | | across devices. | | mobile & desktop. | | broken on mobile | |  | |
|  | |  | |  | |  | |  | |  | |  | |
|  | | Deployment | | Host the app using | | App should be | |  | |  | |  | |
| TC-006 | | Testing | | Streamlit Sharing | | accessible online. | | Deployed | | Sri Teja | |  | |
|  | |  | |  | |  | |  | |  | |  | |

Ensure that the hand gesture recognition and perform actions as expected.

|  |  |
| --- | --- |
| ​ |  |