Project Title: Hand Tracking and Gesture-Based System Control

Team Name: BitMasters

Team Members:

- 23B81A6676 Dhanush Chiraboina
- 23B81A6684 Harshavardhan Kanchumoni
- 23B81A6689 Karthik Reddy Thummalapalli
- 23B81A66C5 Viishweshwar Gouni
- 23B81A66C8 Zenith Golusu

Phase-1: Brainstorming & Ideation

Objective: Develop a hand tracking and gesture-based system using OpenCV, MediaPipe, and PyAutoGUI to control system functions such as volume, scrolling, and cursor movement without a physical input device.

Key Points:

1. **Problem Statement:**
   - Traditional input devices (mouse) require physical contact, leading to ergonomic issues over prolonged use.
   - Accessibility challenges for individuals with disabilities in using conventional input methods.
2. **Proposed Solution:**
   - A computer vision-based system that recognizes hand gestures and translates them into system commands.
   - Uses OpenCV and MediaPipe for real-time hand tracking and PyAutoGUI for system interaction.
3. **Target Users:**
   - General computer users seeking an innovative, touch-free way to interact with their system.
   - Individuals with physical disabilities needing an alternative control mechanism.
   - Professionals in fields like design and presentation where hands-free control is beneficial.
4. **Expected Outcome:**
   - A functional, real-time gesture-based system control application that improves accessibility and usability.

Phase-2: Requirement Analysis

Objective: Define the technical and functional requirements for the hand tracking project.

Key Points:

1. **Technical Requirements:**
    - **Programming Language:** Python
    - **Computer Vision:** OpenCV, MediaPipe
    - **System Interaction:** PyAutoGUI
    - **Hardware:** Camera (webcam or built-in laptop camera)
2. **Functional Requirements:**
    - Detect hand landmarks in real time.
    - Recognize different hand gestures for performing system actions (e.g., volume control, scrolling, cursor movement).
    - Provide smooth and responsive interactions with low latency.
3. **Constraints & Challenges:**
    - Accuracy of hand detection under varying lighting conditions.
    - Ensuring minimal lag for real-time interaction.
    - Handling multiple gestures efficiently without false triggers.

Phase-3: Project Design

Objective: Develop the architecture and user flow of the application.

Key Points:

1. **System Architecture:**
    - Captures video input from the camera.
    - Processes frames using OpenCV and MediaPipe for hand tracking.
    - Recognizes specific gestures and maps them to corresponding system functions.
    - Uses PyAutoGUI to execute system commands like scrolling, volume control, and cursor movement.
2. **User Flow:**
    - Step 1: The user moves their hand in front of the camera.
    - Step 2: The system detects and tracks hand landmarks.
    - Step 3: Specific hand gestures are mapped to predefined system commands.

- Step 4: The corresponding action (e.g., volume adjustment) is performed in real time.

3. **UI/UX Considerations:**
   - Minimalist, real-time visual feedback of hand detection.
   - User-friendly instructions and gesture explanations.
   - Low latency for seamless user experience.

Phase-4: Project Planning (Agile Methodologies)

Sprint	**Task**	**Priority**	**Duration**	**Deadline**	**Assigned To**	**Dependencies**	**Expected Outcome**
Sprint 1	Environment Setup & Library Installation	High	3 hours	Day 1	Entire Team	Python, OpenCV, MediaPipe, PyAutoGUI	Successful environment setup
Sprint 1	Hand Detection Implementation	High	5 hours	Day 1	Dhanush	OpenCV, MediaPipe	Hands detected in real-time
Sprint 2	Gesture Recognition	High	6 hours	Day 2	Karthik	Hand landmarks	Hand gestures mapped to commands
Sprint 2	System Control Integration	High	5 hours	Day 2	Harshavardhan	PyAutoGUI	System functions controlled via gestures
Sprint 3	Testing & Debugging	Medium	4 hours	Day 3	zenith	Completed implementation	Smooth & responsive interaction
Sprint 3	Final Presentation & Deployment	Low	2 hours	Day 3	Viishweshwar	Working prototype	Demo-ready application

---

Phase-5: Project Development

Objective: Implement core features of the hand tracking system.

Key Points:

1. **Technology Stack Used:**

- **Computer Vision:** OpenCV, MediaPipe
- **Gesture Recognition:** MediaPipe Hand Landmarks
- **System Control:** PyAutoGUI
- **Programming Language:** Python

2. **Development Process:**
   - Implement real-time hand tracking with MediaPipe.
   - Recognize gestures and map them to system actions.
   - Optimize interaction responsiveness.
3. **Challenges & Fixes:**
   - **Challenge:** Hand detection accuracy in varying lighting.
     - **Fix:** Adjust brightness thresholds and use adaptive image preprocessing.
   - **Challenge:** Gesture misclassification.
     - **Fix:** Introduce confidence thresholds and gesture smoothing techniques.
   - **Challenge:** Lag in gesture recognition.
     - **Fix:** Optimize frame processing and reduce computation load.

Phase-6: Functional & Performance Testing

Test Case ID	**Category**	**Test Scenario**	**Expected Outcome**	**Status**	**Tester**
TC-001	Functional Testing	Hand detection in normal lighting	Hands detected accurately	Passed	Dhanush
TC-002	Functional Testing	Hand detection in low light	Hands detected with minor delay	Needs Optimization	Karthik
TC-003	Functional Testing	Gesture-based volume control	Volume changes accordingly	Passed	Harshavardhan
TC-004	Functional Testing	Cursor movement with hand gestures	Smooth cursor movement	Passed	Viishweshwar
TC-005	Performance Testing	Response time of hand tracking	Real-time interaction (< 200ms)	Passed	Zenith
TC-006	Bug Fixes & Improvements	Fix incorrect gesture recognition	Higher accuracy of gesture mapping	Fixed	Zenith

1. **GitHub/Code Repository Link** : https://github.com/Dhanush04925/Hand-Gesture.git