Project Title: Gesture-Based Human-Computer Interaction System using OpenCV, MediaPipe, and Palm's text-bison-001

Team Name: Tech Smashers

Team Members:

- Shaik Sameer Anwar
- Ammar Mohiuddin
- Syed Saifuddin

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered Gesture-Based Human-Computer Interaction System that enables users to interact with computers through intuitive hand movements.

Key Points:

Problem Statement:

- Many users require touchless interaction with devices for accessibility, gaming, and public kiosks.
- Existing systems lack real-time responsiveness and AI-based contextual interpretation of gestures.

Proposed Solution:

- A computer vision-based system using OpenCV and MediaPipe to recognize hand gestures.
- Integration with Palm's text-bison-001 AI model to provide descriptive feedback for gestures.
- A user-friendly interface built with Streamlit to allow seamless interaction.

Target Users:

- Users at public kiosks (airports, museums, malls) who need touchless interaction.
- Individuals requiring assistive technology for computer control.
- Gamers looking for gesture-based gaming experiences.

Expected Outcome:

- A fully functional gesture-based system that provides AI-powered interpretations of gestures for various applications.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements of the Gesture-Based HCI System.

Key Points:

Technical Requirements:

- **Programming Language:** Python
- **Computer Vision:** OpenCV, MediaPipe
- **AI Integration:** Palm's text-bison-001
- **Frontend:** Streamlit Web Framework
- **Hardware:** Webcam for gesture recognition

Functional Requirements:

- Detect and recognize hand gestures in real time.
- Interpret gestures using AI-based descriptions.
- Provide a visual and text-based response for recognized gestures.
- Offer a smooth, interactive experience via Streamlit.

Constraints & Challenges:

- Ensuring real-time gesture recognition with low latency.
- Handling variations in hand size, lighting, and occlusions.
- Optimizing AI response time for gesture interpretation.

Phase-3: Project Design

Objective:

Develop the system architecture and define user interaction flow.

Key Points:

System Architecture:

1. User performs a hand gesture in front of the webcam.
2. OpenCV and MediaPipe process the video feed to detect hand landmarks.
3. Recognized gestures are passed to Palm's text-bison-001 for AI-based interpretation.
4. The AI-generated description is displayed in the Streamlit UI.

User Flow:

1. **Step 1:** User performs a hand gesture.
2. **Step 2:** The system recognizes and classifies the gesture.
3. **Step 3:** AI generates a meaningful description for the gesture.
4. **Step 4:** The description and a visual indicator are displayed on the UI.

UI/UX Considerations:

- Clean and intuitive UI for seamless interaction.
- Gesture response time should be under 500ms.
- Dark & light mode for better user experience.

Phase-4: Project Planning (Agile Methodologies)

Sprint Planning with Priorities:

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & Library Installation	🔴 High	4 hours	End of Day 1	Sameer	Python, OpenCV, MediaPipe setup	Development environment ready
Sprint 1	Basic Gesture Recognition	🔴 High	3 hours	End of Day 1	Ammar	OpenCV, MediaPipe setup	System detects hand gestures

Sprint 2	AI Integration for Gesture Interpretation	🔴 High	5 hours	Mid-Day 2	Saif	Gesture recognition working	AI-based gesture descriptions
Sprint 2	UI Development with Streamlit	🟡 Medium	3 hours	End of Day 2	Team	AI response, basic UI elements	Functional UI with gesture response
Sprint 3	Testing & Performance Optimization	🟡 Medium	2 hours	Mid-Day 2	Ammar	Working prototype	Real-time, smooth gesture recognition
Sprint 3	Final Presentation & Deployment	🟢 Low	2 hours	End of Day 2	Entire Team	Completed project	Ready-to-demo project

Phase-5: Project Development

Objective:

Implement core features of the Gesture-Based HCI System.

Technology Stack Used:

- **Frontend:** Streamlit
- **Backend:** OpenCV, MediaPipe, Palm's text-bison-001
- **Programming Language:** Python

Development Process:

1. Implement gesture recognition with OpenCV & MediaPipe.
2. Integrate AI for gesture interpretation.
3. Develop a user-friendly Streamlit interface.
4. Optimize system for real-time interaction.

Challenges & Fixes:

- **Challenge:** High latency in gesture processing.
    - **Fix:** Optimize image processing pipeline.
- **Challenge:** Limited AI response time.
    - **Fix:** Implement caching for frequent gestures.

Phase-6: Functional & Performance Testing

Objective:

Ensure the system functions as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Detect "thumbs up" gesture	Recognized & AI description displayed	✅ Passed	Sameer
TC-002	Functional Testing	Recognize "fist" gesture	Correct interpretation displayed	✅ Passed	Ammar
TC-003	Performance Testing	Response time under 500ms	System responds instantly	⚠ Needs Optimization	Saif

TC-004	UI Testing	Display on different screen sizes	UI adapts correctly	✅ Passed	Tester 2
TC-005	Deployment Testing	Host app on Streamlit	App accessible online	🚀 Deployed	DevOps

---

Final Submission

- **Project Report** (based on the above format)
- **Demo Video** (3-5 minutes)
- **GitHub Repository Link**
- **Final Presentation**