**Hackathon Project Phases Template**

**Project Title:**

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

**Team Name:**

Team Spark

**Team Members:**

* Shaik Nishath Tabassum
* Vangala Harshini
* Adunuri Keerthi
* Lekhana Donthula

**Phase-1: Brainstorming & Ideation**

**Objective:**

Develop a gesture-based interaction system that enables touch-free human-computer communication using hand gestures.

**Key Points:**

**Problem Statement:**

* Traditional input methods like keyboards and mice are not always accessible or hygienic.
* Existing gesture recognition systems lack accuracy and adaptability in real-time environments.

**Proposed Solution:**

* A system that uses **OpenCV and MediaPipe** for real-time hand gesture detection.
* **Palm's text-bison-001 NLP model** to interpret gestures into meaningful commands.
* Enables touch-free interaction for accessibility, gaming, and automation.

**Target Users:**

* Individuals with disabilities for enhanced accessibility.
* Users in healthcare and public spaces requiring touch-free interaction.
* Developers and researchers exploring gesture-controlled applications.

**Expected Outcome:**

* A functional AI-powered gesture-based interaction system with real-time recognition and command execution.

**Phase-2: Requirement Analysis**

**Objective:**

Define the technical and functional requirements for the **Gesture-Based Human-Computer Interaction System** to ensure smooth implementation and performance.

**Key Points:**

**Technical Requirements:**

* **Programming Language:** Python
* **Backend:** Flask (for API handling)
* **Frontend:** React (for UI display)
* **Libraries:** OpenCV, MediaPipe, Palm's text-bison-001 (NLP), TensorFlow

**Functional Requirements:**

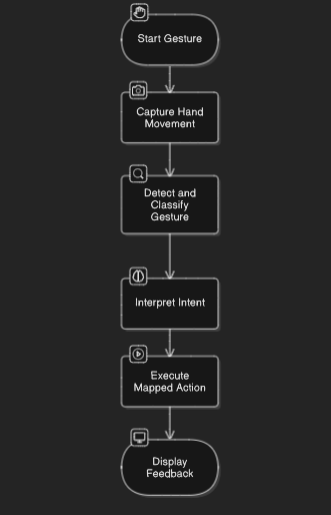
* Detect and recognize predefined hand gestures.
* Convert gestures into commands using NLP.
* Provide real-time feedback on detected gestures.
* Integrate with external applications for extended functionalities.

**Constraints & Challenges:**

* Achieving high accuracy in real-time gesture recognition.
* Handling multiple gestures efficiently.
* Ensuring seamless communication between frontend, backend, and API integration.

**Phase-3: Project Design**

**Objective:**

****

**Key Points:**

**System Architecture:**

1. **Video Input** - Capture live video using OpenCV.
2. **Preprocessing** - Extract hand landmarks using MediaPipe.
3. **Gesture Recognition** - Classify gestures using a trained model.
4. **NLP Processing** - Palm's text-bison-001 interprets gestures into commands.
5. **Action Execution** - Perform corresponding action or response.

**User Flow:**

* **Step 1:** User performs a predefined hand gesture.
* **Step 2:** System detects and processes the gesture.
* **Step 3:** NLP model translates gesture into a command.
* **Step 4:** System executes the corresponding action and provides feedback.

**UI/UX Considerations:**

* Minimalist, intuitive UI for easy gesture interaction.
* Real-time visualization of detected gestures.
* Accessibility features for disabled users.

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

**Sprint Breakdown:**

| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** |
| --- | --- | --- | --- | --- | --- |
| Sprint 1 | Environment Setup & API Integration | 🔴 High | 6 hours | End of Day 1 | Nishath |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours | End of Day 1 | Harshini |
| Sprint 2 | Gesture Recognition Model Implementation | 🔴 High | 4 hours | Mid-Day 2 | Keerthi |
| Sprint 2 | NLP Integration for Gesture-to-Text Conversion | 🔴 High | 3 hours | Mid-Day 2 | Nishath |
| Sprint 3 | API Optimization & Debugging | 🟡 Medium | 3 hours | End of Day 2 | Lekhana |
| Sprint 3 | Final Testing & Deployment | 🟢 Low | 2 hours | End of Day 2 | Entire Team |

**Phase-5: Project Development**

**Objective:**

Implement the core functionalities of the project.

**Key Points:**

**Technology Stack:**

* **Computer Vision:** OpenCV, MediaPipe
* **Machine Learning:** TensorFlow
* **NLP Processing:** Palm’s text-bison-001
* **Backend:** Flask (Python)
* **Frontend:** React.js
* **Database:** Not required initially (API-based interactions)

**Development Process:**

1. Implement **gesture detection** and tracking using OpenCV & MediaPipe.
2. Develop **custom gesture recognition model**.
3. Integrate **Palm’s text-bison-001** for NLP-based command processing.
4. Implement **real-time feedback** and UI enhancements.

**Challenges & Fixes:**

* **Challenge:** Low accuracy in gesture detection.  
  **Fix:** Improve dataset and retrain the model.
* **Challenge:** API rate limits for NLP processing.  
  **Fix:** Optimize API calls and use caching where possible.

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

**Objective:**

Ensure the system performs as expected with high accuracy.

**Test Cases:**

| **Test ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** |
| --- | --- | --- | --- | --- |
| TC-001 | Functional | Perform "Thumbs Up" gesture | Recognized as positive feedback | ✅ Passed |
| TC-002 | Functional | Perform "Victory" gesture | Recognized correctly | ✅ Passed |
| TC-003 | Performance | API response time under 500ms | Fast response | ⚠ Needs Optimization |
| TC-004 | Bug Fixes | Fix misclassified gestures | Improved accuracy | ✅ Fixed |
| TC-005 | UI Testing | UI works across different devices | Responsive layout | ❌ Failed - Needs Fixing |
| TC-006 | Deployment | Deploy app on web | Accessible online | 🚀 Deployed |

**Final Submission Deliverables:**

* **Project Report:** Detailed documentation of system architecture and development.
* **Demo Video:** 3-5 minutes showcasing working prototype.
* **GitHub Repository:** Includes codebase and setup instructions.
* **Final Presentation:** Slide deck covering problem, solution, and demo.

**Future Enhancements:**

* **Multi-hand gesture recognition** for more complex commands.
* **Sign language interpretation** support.
* **Integration with IoT devices** for home automation control.
* **Cross-platform support** for mobile and web apps.

This document provides a structured breakdown of your project phases, including architecture, development plan, and testing procedures. Let me know if you need further refinements!