Hackathon Project Phases Template for the **Sign language to Speech** project.

Hackathon Project Phases Template

Project Title:

Ges2Speech

Team Name:

GenxAi

Team Members:

- K.jagadesh
- K.vikas
- G.suman
- B.sunil

Phase-1: Brainstorming & Ideation

Objective:

The objective of sign language to speech using human palm gestures is to develop a system that translates hand gestures, particularly palm movements, into spoken language, facilitating communication for individuals with hearing impairments. This technology aims to bridge the gap between sign language users and the broader community by converting visual gestures into audible speech in real-time..

Key Points:

1. **Problem Statement:**

    - There is a lack of efficient, real-time systems that can accurately translate sign language gestures, particularly palm movements, into spoken language, creating communication barriers for individuals with hearing impairments.

2. **Proposed Solution:**

   - Use **OpenCV** and **MediaPipe** to capture and analyze hand gestures, specifically palm movements, in real-time, enabling accurate recognition of sign language gestures.
   - Integrate the gesture recognition system with **PyTTSX3** (Python Text-to-Speech) to convert recognized gestures into audible speech, allowing for seamless communication between sign language users and non-sign language speakers..

3. **Target Users:**

   - **Individuals with hearing impairments or speech disabilities**
   - **Non-sign language speakers**

4. **Expected Outcome:**

   - The expected output is a real-time translation of **sign language gestures, specifically palm movements**, into clear and accurate spoken language..

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the System

Key Points:

1. **Technical Requirements:**

   - Programming Language: **Python**
   - Backend: **opencv ,mediapipe, pyttsx3**
   - Frontend: **Opencv Window**
   - Database: **Not required initially**

2. **Functional Requirements:**

   - Detect hand gestures from a webcam fees.
   - Recognize predefined gestures using a trains Ml model.
   - Convert the recognized gestures into speech output.
   - Allow user to retrain gestures if needed.
   - Store previous training data for reuse.

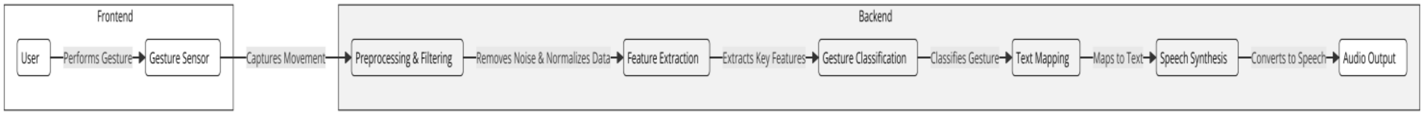
3. **Constraints & Challenges:**

   - Ensuring real-time human gestures capturing.
   - Handling **gestures** and optimizing the speech.
   - Providing a **smooth UI experience**

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.



Key Points:

1. **System Architecture:**
    - Capture hand gestures using OpenCV.
    - Process hand landmark with Mediapipe.
    - Classify gestures using a KNN model
    - Covert the detected gesture into speech using pyttsx3
2. **User Flow:**
    - Step 1: Start Application.
    - Step 2: User Perform a Gesture.
    - Step 3: Recognition process: Ml model classifies the gesture
    - Step 4: Speech output & Display.
    - Step 5: Exit application>use ‘q’ to quit
3. **UI/UX Considerations:**
    - Real-time feedback with gesture name displayed on the screen.
    - Clear instruction for users on how to retrain gestures.

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	Setup OpenCv & Mediapipe for hand tracking	⍰ High	4 hours (Day 1)	End of Day 1	K.jagadesh	Python ,opencv	Webcam detects Hand gestures
Sprint 1	Implement Hand Landmark Extraction	⍰ High	3 hours (Day 1)	End of Day 1	G.suman	Mediapipe setup	Extracted features points
Sprint 2	Implement real-time gestures recogniton	⍰ High	3 hours (Day 2)	Day 2	K.vikas	Hand landmark data	Trained Ml model
Sprint 2	Integrate Text-to-speech	⍰ High	2 hours (Day 2)	Day 2	B.sunil	Recoginized gestures	Live gesture classification
Sprint 3	UI Enhancements	⍰ Medium	2 hours (Day 2)	Day 2	B.sunil	Basic UI working	Responsive UI, better user experience
Sprint 3	Final Presentation & Deployment	⍰ High	1 hour (Day 2)	End of Day 2	Entire Team	Fully implemented system	Demo-ready project

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(⍰ **High Priority)** Set up the **environment** & install dependencies.
(⍰ **High Priority)** Integrate **OpenCv mediapipe.**
(⍰ **Medium Priority)** Build a **trained basic gestures**.

Sprint 2 – Core Features & Debugging (Day 2)

(⍰ **High Priority)** Implement **Real-time gestures recognition**

. (⍰ **High Priority)** integrate **Text-to-speech**

Sprint 3 – Testing, Enhancements & Submission (Day 2)

.

(⍰ **Low Priority)** Final **demo preparation**

Phase-5: Project Development

Objective:

Implement core features of the **Sign language to Speech**

Key Points:

1. **Technology Stack Used:**
    - **Computer Vision:** Opencv, MediaPipe
    - **Machine learning Model:** KNN classifier
    - **Text-to-Speech:** pyttsx3
    - **Programming Language:** Python
2. **Development Process:**
    - Set up Mediapipe hand Tracking &opencv camera feed
    - Integrate gesture Recognition &real-time Classification.
    - Add speech output using pyttsx3
    - Optimize UI for better feedback &usability
3. **Challenges & Fixes:**
    - **Challenge:** False detection due to different hand orientations.
      Fix: Standardize landmark positioning.
    - **Challenge:** Repeated speech output.
      Fix: implement gesture change detection.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the Sign language to Speech works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Show ‘Hello’ gesture	System says ‘Hello’	✅ Passed	suman

TC-002	Functional Testing	Show ‘Thumbs Up’ gesture	System says ‘Thumbs Up’ .	✅ Passed	B.sunil
TC-003	Performance Testing	Model inference time under 500ms	Gesture recognized quickly.	⚠ Needs Optimization	k.vikas
TC-004	Bug Fixes & Improvements	Fixed repeated speech Output issue	Speech only triggered When gesture changes.	✅ Fixed	Developer
TC-005	Final Validation	Ensure UI Displays Correct gesture	Recognized gesture Shown on screen.	✅ Passed	k.vikas

Final Submission

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**