Sprint Plan

Phase 1 (Weeks 1 - 4): Foundations & Prototypes

Sprint 1: Study Limitations + ASR Baseline

Goal:

- Benchmark current apps (Otter, Zoom, Copilot).
- Build a baseline Whisper transcription.
- Document limitations + gaps clearly.

Tasks:

1. Collect 2–3 short meeting recordings (English + Arabic + code-switch).
2. Run transcription on:
    - Whisper (open-source).
    - Google Speech-to-Text API.
    - Otter.ai (free trial).
3. Compare results:
    - WER (Word Error Rate).
    - Handling noise.
    - Handling code-switch.
    - Speaker separation (if available).
4. Write limitation table (benchmark vs what we aim to improve).

Evaluation Metrics:

- **WER (Word Error Rate):** lower = better.
- **Noise robustness:** WER with/without noise suppression.
- **Code-switching:** qualitative check (did it keep Arabic words? translate? ignore?).

Deliverable:

- Report (2–3 pages): Limitations + baseline performance.
- Notebook: `audio.wav → transcript.txt` (Whisper baseline).

Sprint 2: Sign Language Model (Basic)

Goal: Recognize some phrases in English Sign Language.
Tasks: Collect dataset → Extract hand landmarks (MediaPipe) → Train sequence model (LSTM/GRU).
Evaluation: Accuracy, Confusion Matrix.
Deliverable: Notebook: `video.mp4 → ["Hello", "Thank you"]`.

Phase 2 (Weeks 5 - 8): Core Features

Sprint 3: Speaker Diarization & Summarization + Task Extraction

Sprint 3a: Speaker Diarization:

Goal: Identify “who spoke when”.
Tools: Pyannote, x-vectors.
Evaluation: DER (Diarization Error Rate).
Deliverable: `transcript.json` with speaker labels.

Sprint 3b: Summarization + Task Extraction:

Goal: Summarize + detect action items.
Tools: T5/BART + simple rule-based task extractor.
Evaluation:

- ROUGE-L (summary vs reference).
- Precision/Recall (tasks vs annotated set).

Deliverable: `summary.md`.

Sprint 4: Fine-tuning the sign language model

Goal: Improve the performance of the sign language recognition system

Methods: experimenting with alternative architectures (e.g., Transformer, CNN-LSTM hybrids), applying data augmentation (rotation, flipping, temporal jittering), and expanding the vocabulary to cover more phrases.

Evaluation: Accuracy, Confusion Matrix, and Comparison with Sprint 2 baseline (should show measurable improvement).

Deliverable: improved Notebook: `video.mp4 → ["Hello", "Thank you", "I want to work on this", "Good morning"]`.

Phase 3 (Weeks 9 - 11): Integration & Expansion

Sprint 5: Multi-Modal Integration

Goal: Merge sign language + ASR → unified transcript.
Evaluation:

- Accuracy of merged transcript.
- Manual review of sign+speech overlap.

Deliverable: `unified_transcript.json`.

Sprint 6: Semi-Real-Time Pipeline

Goal: Process recorded meetings sequentially.
Evaluation: Latency (time per minute of meeting).
Deliverable: Pipeline script: input recording → transcript + summary.

Phase 4 (Weeks 12 - 16): Real-Time & Polishing

Sprint 7: Real-Time Prototype

Goal: Live transcription + captions.
Tools: WebRTC / Google Meet API + Whisper streaming.
Evaluation:

- Average latency (ms).
- Real-time WER.

Deliverable: Live captions demo.

Sprint 8: Final Integration & Report

Goal: End-to-end system + GUI.
Deliverable:

- Final system (batch + demo real-time).
- Streamlit/Flask dashboard.
- Final report & presentation.
