**Goal & constraints**

Goal: Turn my Task 5 narrative into a short, human-sounding street interview and output it as audio. Document the process thoroughly.

Constraints: Free/student tools only; reproducible pipeline; clear ethics labeling;

**Approach overview**

**Source narrative:** Use Task 5’s analysis themes (performance ≈ expectation, scoring concentration, close-game margins, pace/possession signals).

**Scriptwriting:** Draft a natural HOST/GUEST “sidewalk” Q&A (no variables). Add light stage directions like “[street noise]” for vibe.

**Text-to-Speech:** Try Coqui TTS first (multi-speaker); if dependency pain on Colab, fall back to Piper (binary), which is robust.

**Audio assembly:** Generate one WAV per line, stitch with pydub (tiny crossfades to avoid clicks), export MP3.

**Ethics:** Add an audible disclosure line (intro/outro); use synthetic voices only.

**Reproducibility:** Provide a one-cell Colab that does installs, voice downloads, TTS, stitching, and MP3 export.

**Tools I tried (and why)**

1. Google Colab — free runtime with FFmpeg.
2. Coqui TTS (VCTK) — natural multi-speaker voices.
3. Piper (binary) — lightweight, resilient to Python version issues.
4. pydub + FFmpeg — concatenation + format export.
5. Piper voices (Hugging Face) — open Amy (HOST) + Ryan (GUEST).

**What I built**

1. A street-style interview script (HOST/GUEST) that sounds conversational (no numbers / variables).
2. A single Colab cell that:
   * Installs minimal audio deps,
   * Downloads the Piper binary and two voices,
   * Synthesizes each line (HOST=Amy, GUEST=Ryan),
   * Stitches lines with crossfades,
   * Exports MP3 and triggers download.

**Workflow — step by step**

**A) Narrative → Script**

1. Distilled Task 5 insights into plain English: results ~ matched expectations
2. Scoring leaned top-heavy
3. tight games hinged on trainable details
4. In high-pace games, possession value spikes
5. Plan = diversify creation, rehearse end-game packages, manage tempo.
6. Wrote a 90-second, two-voice street interview with light ad-libs

**B) TTS pathfinding (failures & fixes)**

* Coqui TTS (attempt #1)
  1. Dependency clashes on Colab Py3.12:
  2. coqui-tts 0.24.1 requires transformers < 4.41 (Colab had newer).
  3. gruut wants networkx < 3.0 (Colab had newer).
  4. Hit ModuleNotFoundError: numpy.strings and TypeError: issubclass() in coqpit.
  5. Call: Too fragile for a class hand-off → parked.
* Piper (Python package) (attempt #2)
  1. Wheel missing on Py3.12: piper-phonemize~=1.1.0 not found.
  2. Call: Skip Python wrapper → use Piper binary.
* Piper (binary) (attempt #3) — final
  1. Fixed binary URL/tag to 2023.11.14-2; searched for the executable with a robust \*\*/piper glob; chmod +x.
  2. Success: synthesis OK, then stitched with pydub, exported MP3.

**C) Audio assembly & polish**

* Per-line WAVs → stitched with 35 ms crossfades (reduces clicks).
* Applied −1.5 dB gain for headroom before MP3.
* Exported MP3 @ 192 kbps (good balance of size/quality).

**5) Final deliverables**

street\_interview.mp3 — two-voice audio interview.

Script — the exact HOST/GUEST text used.

One-cell Colab — self-contained: downloads Piper + voices, synthesizes, stitches, exports MP3.

**6) Ethics & labeling**

The interview explicitly states it’s AI-generated for academic research (intro & outro).

No impersonation of real individuals; only synthetic voices.

Repo transparency: include script + code used to generate the MP3.

**7) Reproduce this (quick)**

Open Google Colab → New notebook.

Paste the one-cell MP3-only code block.

Run → it downloads Piper + voices, synthesizes, stitches, and downloads street\_interview.mp3.

**8) What didn’t work (and why)**

* Coqui TTS: great quality but fragile on Colab’s Python/NumPy/Transformers pins.
* Piper (Python lib): wheel unavailable on Py3.12.
* Piper (binary): solved both—only needed correct release URL and path fix.

**9) Future improvements**

* Add a low-volume ambience bed (street noise) ducked under speech.
* Optional talking-head video with SadTalker or Wav2Lip using a consented image.
* Simple CLI flags for voice swap, tempo (length\_scale), crossfade length.