

## **Nirmaan AI Intern Case Study**

### **Background**

As part of our Communication Program, we are building an AI tool to analyse and score students' spoken communication skills. One common exercise is a short **self-introduction** submitted as an audio file. The audio has already been transcribed to text (transcript provided in the Excel file).

**Objective of this case study:** build a tool (front-end + back-end+ Logics) that takes a transcript text as input and produces a rubric-based final score (0–100) and per-criterion feedback. The tool must combine **rule-based** methods, **NLP-based** semantic scoring, and apply the **data-driven rubric** provided in the Excel file.

### **Task (what the candidate must deliver)**

Build and submit a working solution that:

1. **Accepts a transcript text** (pasted into a UI text area or uploaded as a text file).
2. **Computes per-criterion scores** using the rubric in the provided Excel. Rubric fields include criterion description, keywords, weight, and optional min/max word limits.
3. **Uses three approaches combined:**
  - o **Rule-based:** keyword presence, exact matches, word-count checks.
  - o **NLP-based:** semantic similarity between transcript and rubric descriptions (use a sentence-embedding model like sentence-transformers or equivalent).
  - o **Data/rubric-driven weighting:** combine signals according to criterion weights and produce a normalized final score (0–100).
4. **Shows detailed output:** overall score, per-criterion scores, and brief textual feedback for each criterion (e.g., which keywords were found, semantic similarity number, length suggestion).
5. **Has a simple frontend** (web UI) that lets a user paste text and click **Score** to get results.
6. **Is deployed publicly** (GitHub repo + deployed app link). For back end, you can run in any free servers or AWS free tier server, or your local machine

Note: The transcript(s) and rubric are in the provided Excel

### **Inputs & Expected Output**

**Input:** a transcript string (text).

**Expected Output (for each input): Either in Json format or in UI**

- overall score (0–100)
- per criterion: array of objects with fields:
- words (transcript length)

**No restrictions** on stack: you may use no-code builders, low-code platforms, or other languages, provided the final product produces the required output for a given input.

## Submission instructions

1. Create a **public GitHub repository**. Include:
  - Source code
  - requirements.txt (or equivalent)
  - README with run instructions and description of scoring formula\
  - Document of detailed steps on how to deploy in your local server
2. **Deployed link** (optional but strongly encouraged).
3. Provide exact steps to run locally in document format
4. Record a screen video showing the app running code running

## Remember

- **This case study is intentionally open-ended.** We want you to research, choose your own tools, and propose a solution that *you* believe best fits the problem. You already have the use case, the sample input, and the expected output — the rest is up to your creativity and product thinking.
- **Please submit your work before the deadline, even if the project is incomplete.** We are not expecting a perfect or fully finished system. What we want to understand is **how you think**, how you **approach problems like a product manager**, and how you make decisions — not whether you can write flawless code.