Nirmaan AI Case Study: Deployment & Logic Guide

1. Project Overview

This tool evaluates student self-introductions based on a rubric. It analyses the text for structure, content coverage, grammar, and speech pace.

Live Deployment: https://huggingface.co/spaces/deepshikhar23/nirmaan-ai-evaluator

2. Scoring Logic (The "Hybrid" Approach)

The tool uses three layers of analysis to ensure accuracy:

1. **Rule-Based Layer:** Used for definitive checks like Word Count, Words Per Minute (WPM), and detecting "Filler Words" (um, uh).
    - Checks for specific structural keywords like "Hello" (Salutation).
2. **NLP Semantic Layer (AI):**
    - **Tool:** sentence-transformers (BERT Model).
    - **Why:** Students express goals differently. One might say "My ambition is to be a doctor" (Keyword match), while another says "I want to save lives in a hospital" (No keyword match).
    - **Logic:** The AI calculates the semantic similarity between the student's sentence and the topic "Ambition". If the similarity > 0.4, points are awarded.
3. **Linguistic Layer:**
    - **Tool:** language-tool-python.
    - **Logic:** Detects grammatical errors and calculates the error rate per 100 words.

3. How to Run Locally

Follow these steps to run the application on your local machine:

Step 1: Clone the Repository

git clone [YOUR_GITHUB_REPO_LINK]

cd nirmaan-ai-evaluator

Step 2: Install Dependencies

pip install -r requirements.txt

Step 3: Run the Application

streamlit run app.py

4. Tech Stack

- **Frontend/Backend:** Streamlit (Python)
- **NLP Model:** all-MiniLM-L6-v2 (Hugging Face)
- **Grammar Engine:** LanguageTool API

- **Sentiment Analysis:** NLTK VADER