Video Recording Script & Guide

This guide helps you create a professional screen recording demonstrating the Speech Scoring Tool for your case study submission.

Video Requirements

- **Duration**: 3-5 minutes
- **Format**: MP4, MOV, or any common video format
- **Quality**: 720p minimum (1080p recommended)
- **Audio**: Clear narration (optional but recommended)

Recording Tools

Free Options

1. **OBS Studio** (Windows/Mac/Linux) - Professional, free
2. **Loom** (Web-based) - Easy, quick
3. **Windows Game Bar** (Windows 10/11) - Built-in (Win + G)
4. **QuickTime** (Mac) - Built-in
5. **ShareX** (Windows) - Free, powerful

Video Script & Flow

Part 1: Introduction (30 seconds)

On Screen: Show GitHub repository page

Say:

> "Hi, I'm [Your Name], and this is my submission for the Nirmaan AI Intern Case Study. I've built an AI-powered tool that analyzes and scores student self-introductions based on comprehensive communication rubrics."

Action:

- Show GitHub repository
- Briefly scroll through README.md
- Highlight key files (app.py, requirements.txt, etc.)

Part 2: Project Overview (45 seconds)

On Screen: Show README.md or project structure

Say:

> "The tool evaluates six key criteria: Content & Structure worth 40 points, Speech Rate for 10 points, Language & Grammar for 15, Vocabulary Richness for 10, Clarity measuring filler words for 10, and Engagement through sentiment analysis for 15 points, totaling 100 points."

Action:

- Show scoring criteria table
- Highlight technology stack
- Show file structure

Part 3: Local Setup Demo (1 minute)

On Screen: Terminal/Command Prompt

Say:

> "Let me show you how to run this locally. First, I'll activate the virtual environment, then install the dependencies."

Actions:

```bash
# Show these commands being executed
source venv/bin/activate  # or venv\Scripts\activate
pip list  # Show installed packages
python app.py  # Start server
```

Say:

> "The server is now running on localhost port 5000. Let's test it."

Part 4: Testing the API (1.5 minutes)

On Screen: Split screen - Terminal and REST client/Browser

Say:

> "First, let's check the health endpoint to ensure the server is running properly."

Action 1: Health Check

```
bash

curl http://localhost:5000/api/health
```

Show Response:

```
json

{"status": "healthy"}
```

Say:

> "Great! Now let's analyze the sample transcript. This is Muskan's self-introduction from the case study."

Action 2: Analyze Sample

```
bash

python tests/test_api.py
# OR
curl -X POST http://localhost:5000/api/analyze -d @sample_data.json
```

Say:

> "The analysis is complete. Let's examine the results."

Part 5: Results Explanation (1.5 minutes)

On Screen: Show the JSON response or test output

Say:

> "The tool gave this transcript an overall score of [X] out of 100. Let me walk through each criterion."

For each criterion, briefly show:

1. **Content & Structure**: "Scored [X]/40 - Found good salutation, all key information present, proper flow"
2. **Speech Rate**: "Scored [X]/10 - WPM is [Y], which falls in the [category] range"
3. **Grammar**: "Scored [X]/15 - Detected [Y] grammar errors"
4. **Vocabulary**: "Scored [X]/10 - Type-Token Ratio of [Y] shows [good/moderate] diversity"
5. **Clarity**: "Scored [X]/10 - Filler word rate of [Y]%"
6. **Engagement**: "Scored [X]/15 - Sentiment analysis shows [positive/neutral/negative] tone"

Action:

- Scroll through the detailed JSON response
- Highlight the feedback for each criterion
- Show the summary at the end

Part 6: Code Walkthrough (Optional - 1 minute)

On Screen: Open `app.py` in code editor

Say:

> "Let me quickly show you the implementation. The SpeechAnalyzer class contains all the scoring logic."

Action:

- Show the `SpeechAnalyzer` class
- Briefly highlight key methods:
  - `analyze_salutation()`
  - `analyze_keywords()`
  - `analyze_grammar()`
  - `analyze_sentiment()`
- Show how scores are calculated and combined

Say:

> "Each method implements the specific rubric rules, using NLP libraries like LanguageTool for grammar and TextBlob for sentiment analysis."

Part 7: Scoring Methodology (30 seconds)

On Screen: Show the scoring breakdown visually

Say:

> "The tool uses three approaches: rule-based for exact matches and keyword presence, NLP-based using sentence embeddings for semantic similarity, and data-driven weighting according to the provided rubric."

Action:

- Show keyword detection code
- Show TTR calculation
- Show grammar scoring formula

Part 8: Additional Features (30 seconds)

On Screen: Terminal or Docker

Say:

> "The project is production-ready with Docker support, comprehensive documentation, and automated tests. It can be deployed to Heroku, AWS, or any cloud platform."

Action:

```bash
# Show Docker command
docker build -t speech-scorer .
docker run -p 5000:5000 speech-scorer

# Or show deployment files
ls -la
# Show Procfile, Dockerfile, docker-compose.yml
```

Part 9: Conclusion (30 seconds)

On Screen: Back to GitHub repository or README

Say:

> "This tool successfully implements all requirements from the case study: it accepts transcript input, computes per-criterion scores using the provided rubric, combines rule-based, NLP, and data-driven approaches, provides detailed output with feedback, and includes a complete deployment-ready solution. Thank you for watching, and I look forward to your feedback!"

Action:

- Show README.md
- Scroll to technology stack
- Show the repository structure
- End on the GitHub repo main page

Recording Checklist

Before Recording

□ Close unnecessary applications

□ Clear browser history/bookmarks bar

□ Set screen resolution to 1920x1080 or 1280x720

- □ Test microphone levels
- □ Prepare sample data
- □ Have terminal ready with commands
- □ Open all necessary windows
- □ Practice the flow once

During Recording

- □ Speak clearly and at moderate pace
- □ Highlight important parts with cursor
- □ Pause briefly between sections
- □ Show actual results, not just code
- □ Demonstrate real functionality
- □ Keep mouse movements smooth
- □ Avoid long silences

After Recording

- □ Review the video
- □ Check audio clarity
- □ Verify all steps shown
- □ Trim unnecessary parts
- □ Add title card (optional)
- □ Export in high quality
- □ Test playback

Sample Narration Script (Detailed)

[0:00 - 0:30] INTRODUCTION
"Hello! I'm [Name], and this is my submission for the Nirmaan AI Communication Assessment Case Study. I've developed a complete AI-powered tool that analyzes and scores student self-introductions using Natural Language Processing and comprehensive rubrics.

As you can see on my screen, this is the GitHub repository with all the source code, documentation, and deployment guides."

[0:30 - 1:15] PROJECT OVERVIEW
"The tool evaluates transcripts across six communication criteria, each with specific weights. Content and Structure is worth 40 points and includes salutation quality, keyword presence for essential information like name, age, school, and hobbies, and overall flow.

Speech Rate measures words per minute for 10 points. Language and Grammar checks errors using LanguageTool for 15 points. Vocabulary Richness calculates the Type-Token Ratio for 10 points. Clarity measures filler words for 10 points. And Engagement analyzes sentiment for 15 points.

The project uses Flask for the backend, integrates LanguageTool for grammar checking, TextBlob for sentiment analysis, and NLTK for text processing."

[1:15 - 2:00] DEMO - SETUP
"Let me demonstrate how it works. First, I'll start the local server. I've already activated my virtual environment and installed all dependencies from requirements.txt.

Now I'll run python app.py to start the Flask server. As you can see, the server is now running on localhost port 5000 and ready to accept requests."

[2:00 - 3:00] DEMO - TESTING
"Let's test it with the sample transcript from the case study. First, a quick health check... and we get a 'healthy' status response.

Now, I'll analyze the actual transcript. I'm using the test script which sends the sample data to the API. This is Muskan's self-introduction - she's 13 years old, studies at Christ Public School, and tells us about her family and interests.

The analysis takes a few seconds as it processes the text through multiple NLP models... and here are the results!"

[3:00 - 4:00] DEMO - RESULTS
"The tool scored this transcript 86 out of 100 - that's excellent! Let's break it down:

Content and Structure scored 35 out of 40. It detected a good salutation with 'Hello everyone', found all must-have keywords including name, age, school, family, and hobbies, plus additional details like fun facts and interests.

Speech Rate scored 8 out of 10 at 151 words per minute, which is slightly fast but still in the acceptable range.

Language and Grammar scored 12 out of 15, detecting a few minor errors.

Vocabulary Richness scored 6 out of 10 with a Type-Token Ratio showing moderate word diversity.

Clarity scored 10 out of 10 with minimal filler words - excellent!

And Engagement scored 12 out of 15 based on positive sentiment analysis.

The tool also provides specific feedback for each criterion with actionable suggestions."

[4:00 - 4:30] CODE EXPLANATION
"The implementation uses a SpeechAnalyzer class that contains all scoring logic. Each criterion has its own analysis method that applies the specific rubric rules. The methods use a combination of regex for keyword matching, statistical calculations for metrics like TTR and WPM, and NLP libraries for grammar and sentiment.

All scoring thresholds exactly match the provided rubric specifications."

[4:30 - 5:00] CLOSING
"The project is production-ready with Docker support, comprehensive tests, and deployment configurations for Heroku,

AWS, and other platforms. All code is well-documented with a detailed README, deployment guide, and API documentation.

Thank you for reviewing my submission. I've demonstrated a working solution that meets all case study requirements, and I'm excited about the opportunity to contribute to Nirmaan AI's mission. The complete code and documentation are available in the GitHub repository linked in the description."

Technical Setup Tips

Screen Recording Settings

- **Resolution**: 1920x1080 or 1280x720
- **Frame Rate**: 30 FPS minimum
- **Audio**: 44.1 kHz, 16-bit
- **Format**: MP4 (H.264)
- **Bitrate**: 5000-8000 kbps

Terminal Settings

- **Font Size**: 14-16pt (easily readable)
- **Color Scheme**: High contrast (dark background)
- **Window Size**: 80-120 columns wide
- **Clear History**: Start with clean terminal

Code Editor Settings

- **Theme**: Dark or high contrast
- **Font Size**: 14-16pt
- **Hide Minimap**: For cleaner view
- **Highlight Active Line**: Helps viewers follow

Common Mistakes to Avoid

❌ **Don't**:

- Rush through explanations
- Show errors without fixing them
- Use tiny fonts

- Have cluttered desktop
- Forget to show actual results
- Skip testing the API
- Read code line-by-line
- Go over 6 minutes

✅ **Do**:

- Speak confidently and clearly
- Show working functionality
- Demonstrate with sample data
- Explain scoring logic
- Highlight key features
- Show the output
- Keep it concise
- Practice beforehand

Post-Recording

Video Editing (Optional)

1. Add title card with project name
2. Add timestamps in description
3. Trim long pauses
4. Add captions (accessibility)
5. Compress if file too large

File Naming

```
NirmaanAI_CaseStudy_YourName_SpeechScoring.mp4
```

Upload Options

- **YouTube** (Unlisted) - Easy sharing
- **Google Drive** - Share link
- **Loom** - Quick, no editing

- **GitHub** - Add to repository

Example Video Timeline

```
0:00 - Introduction & Repository Overview
0:30 - Project Features & Scoring Criteria
1:15 - Local Setup & Server Start
2:00 - API Testing (Health Check)
2:15 - Analyze Sample Transcript
2:30 - Results Breakdown (Each Criterion)
3:30 - Quick Code Walkthrough
4:00 - Deployment Options
4:30 - Conclusion & Thank You
```

Final Checklist

- □ Video is 3-5 minutes long
- □ Audio is clear
- □ Screen is readable (good resolution)
- □ Shows actual working tool
- □ Demonstrates all key features
- □ Explains scoring methodology
- □ Shows test results
- □ Includes GitHub repository
- □ Professional presentation
- □ Exported in good quality

Ready to record? Follow this script, stay confident, and showcase your excellent work! 🎥 🚀