

Local Deployment Guide – AI Communication Scorer

1. Purpose of this Document

This document explains **step-by-step** how to deploy and run the **AI Communication Scorer** application on a **local server (your own computer)**.

2. System Requirements

To run the application locally, you need:

- **Operating System**
 - Windows 10 / 11
 - (or macOS / Linux with minor command changes)
- **Software**
 - Python **3.10+**
 - Pip (Python package manager, usually included with Python)
 - (Optional) Git, if you want to clone the repository directly
- **Network**
 - Internet connection is required once to:
 - Install Python packages
 - Download the sentence-transformer model (on first run)

3. Downloading the Project

You can download the project in two ways: using **Git** or using a **ZIP file**.

3.1 Option A – Using Git (Recommended)

1. Open **Command Prompt** or **PowerShell**.
2. Run:

```
git clone https://github.com/Rc1704/AI_Communication_Scorer.git  
cd AI_Communication_Scorer
```

3.2 Option B – Download ZIP (No Git Needed)

1. Open the GitHub repository in your browser:
https://github.com/Rc1704/AI_Communication_Scorer
2. Click the green **"Code"** button.
3. Select **"Download ZIP"**.
4. Extract the ZIP file into a folder (for example,
[C:\AI_Communication_Scorer](#)).
5. Open **Command Prompt** and navigate to that folder, e.g. :

```
cd "C:\AI_Communication_Scorer"
```

After this step, your project folder should contain at least:

- `app.py`
- `scoring.py`
- `text_utils.py`
- `requirements.txt`

4. Creating a Virtual Environment

A **virtual environment** keeps this project's Python libraries isolated from other projects on your computer.

Note: The commands below use `python`. On some systems (especially macOS/Linux), you may need to use `python3`.

4.1 Windows

1. In Command Prompt, inside the project folder:

```
python -m venv venv
```

This creates a folder named `venv` in your project directory.

2. Activate the virtual environment:

```
venv\Scripts\activate
```

3. After activation, your prompt should show something like:

```
(venv) C:\AI_Communication_Scorer>
```

4.2 macOS / Linux (for completeness)

1. In Terminal:

```
python3 -m venv venv
```

2. Activate:

```
source venv/bin/activate
```

5. Installing Dependencies

With the virtual environment **activated**, install all required libraries using `requirements.txt`.

In your terminal:

```
pip install -r requirements.txt
```

This will install, among others:

- `streamlit` – Web UI framework
- `pandas`, `numpy` – Data handling
- `vaderSentiment` – Sentiment analysis
- `sentence-transformers` – Semantic similarity (SBERT)

On the first run, `sentence-transformers` may need extra time to download its model.

6. Running the Application (Starting Local Server)

Once dependencies are installed, you can start the local Streamlit server.

1. Ensure you are still:

- In the project directory
- Inside the virtual environment ((**venv**) visible in the prompt)

2. Run:

```
streamlit run app.py
```

3. Streamlit will:

- Start a local server

Show output similar to:

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.x.x:8501>

4. Your default browser may open automatically.

If not, manually open:

<http://localhost:8501>

You are now running the **AI Communication Scorer** on your local machine.

7. Using the Application

On the Streamlit page:

1. Duration of speech

- On the sidebar, enter the duration of the speech in **seconds** (e.g., 52 or 60)

2. Transcript input

- Paste a student's self-introduction text into the large **text area**

3. (Optional) Use sample transcript

- If a sample checkbox is available, tick it to auto-load a sample text

4. Click “Score my introduction”

5. The app displays:

- **Overall score (0–100)**
- Performance band (e.g., Good / Needs improvement)
- Breakdown table:
 - Content & Structure
 - Speech Rate
 - Grammar & Vocabulary
 - Clarity
 - Engagement
 - Semantic similarity (0–1) for rubric dimensions

- Lists of **present** and **missing** content keywords
- Short **feedback summary**: strengths and areas for improvement

This local deployment behaves the same as a hosted (cloud) version, except it's only accessible from your own machine.

8. Stopping the Local Server

To stop the application:

1. Go to the terminal where `streamlit run app.py` is running.
2. Press:
`Ctrl + C`
3. You should see a message indicating that the server has shut down.
4. If you want to exit the virtual environment, run:

`deactivate`

9. Running Again Later

To run the app again on another day:

1. Open **Command Prompt**
2. Navigate to the project folder:
`cd "C:\AI_Communication_Scorer"`
3. Activate the virtual environment:
`venv\Scripts\activate`
4. Run:
`streamlit run app.py`

10. Troubleshooting

10.1 `streamlit` command not found

- Make sure the virtual environment is **activated**
- Confirm installation with:
`pip show streamlit`

If not installed, run:

```
pip install streamlit
```

10.2 Import Errors (e.g., `ModuleNotFoundError`)

Run:

```
pip install -r requirements.txt
```

- Ensure you are in the **correct environment** (`(venv)` visible in prompt)

10.3 Port Already in Use

If port `8501` is already used:

```
streamlit run app.py --server.port 8502
```

Then open:

```
http://localhost:8502
```

10.4 Slow First Run

- The first run may be slower because:
 - Python packages are being cached
 - The sentence-transformer model is being downloaded

Subsequent runs will be faster.