**Missing Details for Seamless Execution**

Below is the comprehensive documentation to address your concerns and ensure the process can be executed seamlessly in any environment:

**1. Word Template Details**

The Word template is critical for applying consistent formatting during resume generation. Ensure the template file is:

* **File Name:** resume\_template.docx
* **Location:** Saved in the same directory as the Python script for ease of reference.
* **Content:** Use placeholder variables (e.g., {{header\_block}}, {{experience}}, {{education}}) to indicate where dynamic data will be injected.

**Example Template Content:**

{{header\_block}}

SUMMARY

{{summary}}

SKILLS

{{skills}}

EXPERIENCE

{{experience}}

EDUCATION

{{education}}

KEY ACHIEVEMENTS

{{key\_achievements}}

PROJECTS

{{projects}}

**2. File Directory Structure**

Organize the files in a logical and consistent manner for execution:

resume\_project/

├── json\_data/

│ └── tailored\_resume.json # JSON schema with all resume details

├── templates/

│ └── resume\_template.docx # Word template

├── scripts/

│ └── resume\_tailoring.py # Python script for resume generation

├── output/

│ └── tailored\_resume.docx # Generated resume

├── test\_files/

│ ├── job\_description.txt # Sample job description

│ ├── full\_cv.docx # Full CV file

│ └── master\_resume.pdf # Original resume file

**3. Testing Instructions**

Follow these steps to test the process:

**Step 1: Verify Prerequisites**

* Install Python (ensure docx and other dependencies are installed using pip install python-docx).
* Ensure resume\_template.docx is present in the templates directory.
* Place the JSON data file (tailored\_resume.json) in the json\_data directory.

**Step 2: Run the Script**

* Open a terminal or command prompt.
* Navigate to the scripts directory.
* Execute the script:
* python resume\_tailoring.py

**Step 3: Verify Output**

* The generated resume will be saved in the output directory as tailored\_resume.docx.
* Open the Word file to confirm the formatting, content, and alignment with the job description.

**4. Adherence to Tenets**

This process aligns with the initiative's tenets:

1. **Consistent:** Outputs follow the same structure and formatting rules.
2. **Session-Agnostic:** Files are organized in directories to avoid resets.
3. **Reliable:** Each step validates data and formatting.
4. **Scalable:** JSON schema supports additional sections dynamically.
5. **Repeatable:** Scripts and templates ensure reproducibility.
6. **Logical:** Directory structure and process flow are intuitive.
7. **Efficient:** Minimal user input is required after setup.
8. **Automated:** The script handles all formatting and data injection.
9. **Comprehensive:** All sections of the resume are dynamically generated.
10. **Accurate:** Rules ensure alignment with job descriptions.

**5. Documentation**

**JSON Schema:**

The finalized JSON schema has been provided above.

**Python Script:**

The tailored Python script (resume\_tailoring.py) was shared in the previous steps.

**6. Next Steps**

* Place all necessary files in the directory structure above.
* Test the process using the Python script and verify the output.
* If additional customization or debugging is needed, provide feedback.

Let me know if you encounter any issues or require further clarification!