Submission Guideline for the OCR Hackathon

These rules describe **exactly** how your team must format and deliver its model outputs so the organizer's `evaluate_submission()` script can assess them automatically.
The evaluation code is public (see `evaluate.ipynb`) and will not be modified, so please follow the specification precisely.

---

1 . Required file

File name	Contents	Encoding
`submission.json`	Your model's predictions for **every** sample in the hidden test set	UTF-8 (no BOM)

Place this single JSON file at the top level of the archive you upload to the platform (zip / tar.gz). No other files are needed.

---

2 . JSON structure

Your `submission.json` **must** have exactly two top-level keys, each mapped to a flat list of equal length:

Submisson.json

```
{
  "file_path": [                  // list[str]
    "book1_page1",
    "book1_page2",
    "... (one entry per test image)"
  ],
  "prediction": [                 // list[str]
    "Si por euitar n pecado mortal",
    "aueys de poner uuestra uida en pe",
    "... (your OCR output, same order)"
  ]
}
```

- `file_path` – The unique identifier for each page or image, **identical** to the strings supplied in the official label file.
- `prediction` – Your OCR result for that page, as a single string. Keep line breaks *only* if they are semantically meaningful in the ground truth.

The evaluator converts both prediction and reference to lowercase internally, so you do **not** need to normalise the case yourself.

---

3 . Completeness & ordering

- Provide a prediction for **every** path that appears in the test set.
  If a path is missing, the script inserts an empty string, which will sharply worsen your CER/WER.
- Extra paths that are **not** in the test list will be ignored; avoid them to keep the file tidy.
- The two lists **must** be the same length and aligned index-for-index, but they do **not** have to follow the original test-set order—`evaluate_submission()` matches by `file_path`.

---

4 . Text formatting rules

Aspect	Guideline
Character set	Unicode allowed. Preserve accents/diacritics if your model predicts them.
Whitespace	Trim leading/trailing spaces. Consecutive internal spaces are kept as-is.
Newlines	Use `\n` only when needed; avoid Windows `\r\n`.
Quotes & escapes	The JSON must be valid – escape backslashes and quotes correctly.
Empty strings	Only permitted when the model genuinely produces no text (discouraged).

---

5 . How scoring works

Metric	Source in script	Notes
CER (Character Error Rate)	`evaluate.load("cer")`	Lower is better.
WER (Word Error Rate)	`evaluate.load("wer")`	Lower is better.
Levenshtein distance	`Levenshtein.distance`	Averaged over all samples.
Similarity score	`difflib.SequenceMatcher.ratio()`	0–1, higher is better.

All four metrics are returned in a single dictionary. Feel free to run the script locally with the *public* label file we provided to sanity-check your JSON before you submit.
