1. Functional Verification

1. **Run End-to-End Test on Sample Dataset**
    - Prepare a small, manageable dataset (50–500 documents).
    - Run the program from Stage 1 → Stage 5.
    - Check that each stage outputs:
        - Lexicon: correct number of unique terms, DF values, term_id mappings
        - Forward Index: correct term_ids per document, frequencies, positions
        - Inverted Index: correct posting lists per term
2. **Check File Outputs**
    - Make sure segmented files are created as expected:
        - `forward_index_0.bin`, …
        - `inverted_index_segment_0.bin`, …
        - `.lexicon`, `.json`, `.term_map`
    - Open binary/JSON files to verify content visually for a few entries.
3. **Load & Query Test**
    - Load the indexes back from disk.
    - Query a few terms manually and see if the correct documents are returned.
    - Check for token consistency across Lexicon → Forward Index → Inverted Index.

---

2. Logical Verification

1. **Cross-check counts**
    - Sum of document frequencies (DF) in Lexicon should match term occurrences in Forward Index.
    - Total documents in Forward Index should match dataset count.
2. **Validate term IDs**
    - Pick a token → find its term_id in Lexicon → ensure Forward Index documents contain that term_id.
3. **Check ordering & segmentation**
    - Posting lists in Inverted Index should be **sorted by doc_id**.
    - Segmented files should have roughly equal chunks (except maybe last segment).

---

3. Edge Case & Stress Testing

- **Empty documents:** Make sure empty lines don't crash the program.
- **Rare or repeated tokens:** Ensure DF and posting lists are correct.
- **Large dataset simulation:** Use 1–2k lines to check segmentation and memory handling.

---

4. Performance Verification (Optional but Professional)

- Measure **time per stage** using a timer.
- Check memory usage for larger sample datasets.
- Ensure your code does not crash or slow down significantly.

---

5. Automated Checks

- Write a **small script** to:
    - Load the Lexicon → verify number of terms
    - Load Forward Index → verify document term_ids
    - Load Inverted Index → verify posting lists for a few tokens
    - Print success/failure for each check

This is **industry practice** for verifying correctness before submission or deployment.

✅ Reality Check

If your program currently runs Stage 1 → 3 → 5 successfully on your sample dataset (as your last run logs show), most of the heavy lifting is already correct.

- Focus now on **verification & consistency checks**.
- Create **sample dataset + automated check script** for submission — that's what will make your work professional-level ready.