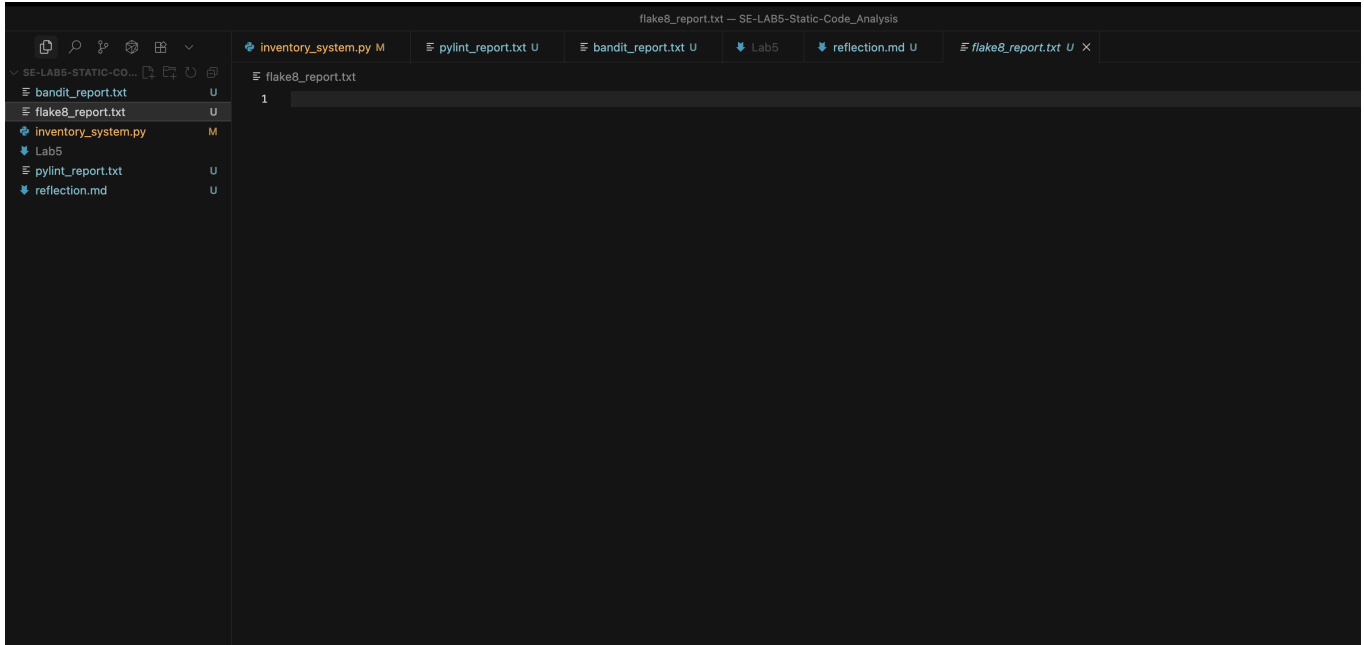


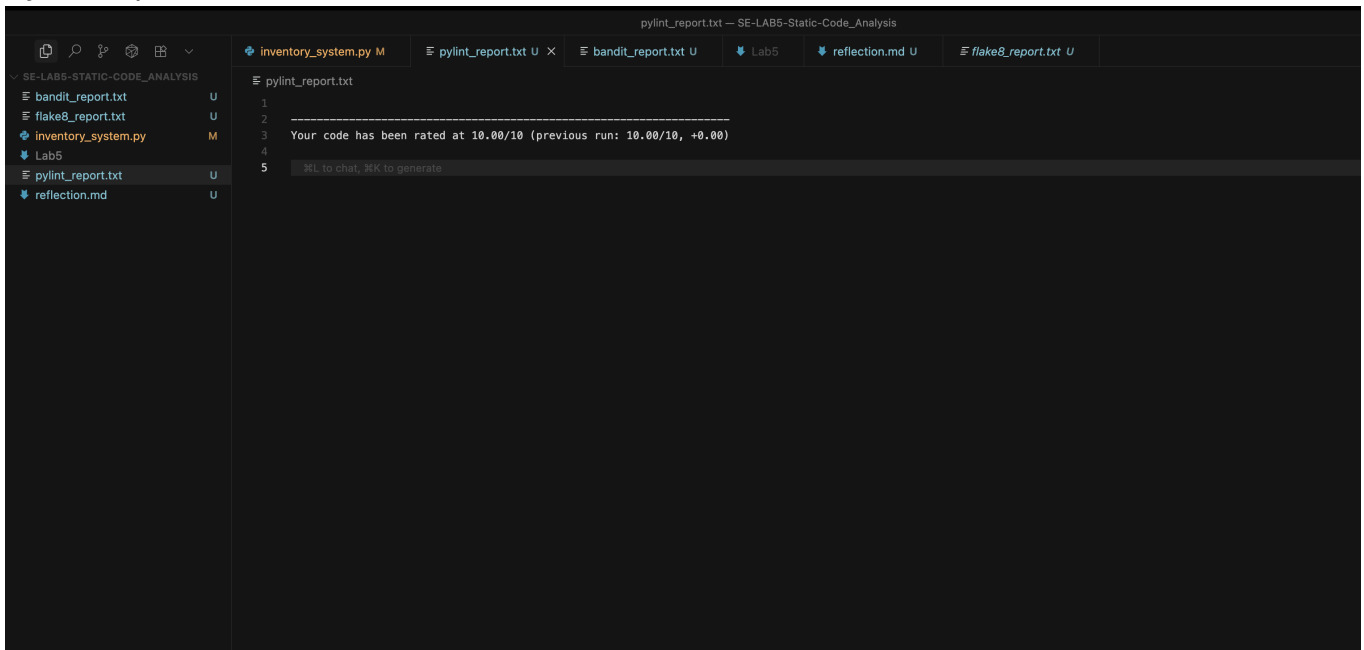
SE-Lab5

Error report Screenshots :

Flake8_report:



Pylint_report:



Bandit_report:

Issues Table:

Issue	Type	Line(s)	Description	Fix Approach
Unused import	Flake8	2	logging imported but never used (F401).	Removed unused logging import.
Missing blank lines	Flake8	8, 14	Expected 2 blank lines before function defs (E302).	Inserted required blank lines before top-level defs.
Mutable default arg	Pylint	10	logs=[] shared across calls (W0102).	Changed parameter to logs=None and initialized inside function.
Non-snake_case names	Pylint	10, 17	addItem/removeItem violate snake_case (C0103).	Renamed functions to snake_case (add_item, remove_item, get_qty, load_data, save_data, print_data, check_low_items).
Try/except/pass	Bandit	19	Silent exception handling detected (B110).	Replaced bare except with specific exceptions (KeyError, TypeError) to avoid silent catch-all.
Use of eval	Bandit	59	Possibly insecure eval usage (B307).	Removed eval entirely; no dynamic evaluation is performed.

Reflection :

Question 1

- Easiest: removing the unused import and adding missing blank lines; they were purely stylistic and risk-free.
- Hardest: replacing bare except with specific exceptions and removing eval; both required reasoning about failure modes and security impact.

Question 2

- Style-related renames to snake_case can be noisy if functions are part of a public API, even if they improve consistency.
- Global usage warnings can be acceptable in a small, single-file script, though they signal maintainability risks.

Question 3

- Add pre-commit hooks to run linters and security checks locally before commits.
- Gate CI with separate jobs for style, lint, and security; fail on high-severity issues and track trends for the rest.

Question 4

- Clearer, more consistent code via snake_case, docstrings, and f-strings.
- Better robustness and security by targeted exception handling and removing eval.

Git repo link:

https://github.com/NilayS08/SE-LAB5-Static-Code_Analysis