CODE REVIEW: TP1

SUMMARY

An intern at your company has just wrapped up a project to create a password strength calculator. The tool works, but there's a catch—the code is a mess. It's poorly structured, lacks comments, and doesn't follow best practices. Now, it's up to you to take this rough draft and turn it into a professional, industry-ready and open-source application. However, we didn't want our competitor to use it for business purpose.

Your Mission: Your task is to step into the shoes of a software engineer and industrialize the intern's code. This means transforming it from a basic, functional prototype into a polished, maintainable, and scalable piece of software.

https://github.com/Fisjkars/CodeReview

DELIVERABLES EXPECTED

A Github repository containing the intern code with the following elements:

- Maven/Gradle/Ant project
- README.md with a summary of the project and basic usage of the library.
- Code:
  - License chosen.
  - Comment the code using Javadoc.
  - Apply a code style policy (ex: Checkstyle).
  - Fix bugs if necessary.
- SCM:
  - Github project.
  - Security.md.
  - Issues Templates.
  - Pull request template.
  - Protected Branches policies.
  - Define pull request policy (approval etc…)
- Tests:
  - Units Tests implementation with proper test plan.
  - “Performance test” on “ComputeMD5” method
- CI/CD:
  - Foreach pull requests:
    - Checkstyle
    - Unit Tests passed.
    - Test coverage reach 90%. (Jococo)
    - Performance Tests passed.
    - SAST run successfully without default (Ex: Semgrep).
  - Foreach main modification:
    - Build and Deploy version in Github Maven registry.
    - Deploy documentation using GH-Pages with URL (blah.com/javadoc/<version>)
    - Deploy Cobertura report with GH-Pages with URL (blah.com/cobertura/<version>)