Improve Code Readability

Date	15November 2022
Team ID	PNT2022TMID17245
Project Name	Project - IoT based Gas Leakage Monitoring and Alerting System for Industries
TEAM LEADER	JEEVITHA C
TEAM MEMBERS	INDHUMATHI B JEEVAJOTHI D KANISHKA K

To all of you searching for quick tips without reading everything, read the TLDR version below:

1. Reuse what will be used more than once.
2. Readability and maintainability over a generic solution.
3. Make modules, classes, or components as small as possible.
4. Have rules and guidelines for your code.
5. Code like you're in a team — even a one-person team.

Reuse What Will Be Used More Than Once:

- Most developers know what D.R.Y. means (Don’t Repeat Yourself). D.R.Y. can help you prevent code duplication.
- Why write a function over and over again? You should write it once and reuse it in multiple places. If you need to change that code, you only have to look in one place instead of copy-pasting a bug fix in multiple places.

- But be aware that you will introduce complexity with D.R.Y. because, in the end, things will be reused more and more.
- The importance of writing tests when reusing parts of your code will be very clear when you start changing that code.

Readability and Maintainability Over a Generic Solution:

- Reusability, readability, and maintainabilities are each other's friends and enemies simultaneously. When you start applying D.R.Y. to your code, you introduce complexity. When you introduce complexity, the readability grade can go down.
- So don't start with a generic solution when building features. Start simple! The first time can't be perfect.
- You can reuse parts of the application through iterations but still guard the readability and maintainability.
- When working in an organization with many development teams, your team will be divided into internal people and external people (like freelancers or consultants). So in this case, people will switch from organization to organization more often.
- In those cases, readability and maintainability are the keys to success. Generic solutions implemented by a person who could easily leave the team are not a smart choice.
- Sometimes, you need a generic solution, but those solutions still have to be readable and maintainable.

Make Modules, Classes, or Components as Small as Possible:

- When building new features for an application, you probably plan them carefully.
- The best solutions can be divided into small modules, classes, or components. Are you wondering why?
- Small pieces of code are easier to test and maintain.
- Imagine that building a tall building is also done by moving smaller components instead of building the whole thing at once and then trying to move it to the location. OK, there are exceptions.
- Most modern libraries and frameworks are divided into smaller building blocks instead of one file.
- JavaScript libraries and frameworks like Angular, React, and Vue applies the concept of the component. I don't think they are doing this by accident.

Automate Rules and Guidelines for Your Code

- One part of writing readable and maintainable code is its architecture of it. Another part is the code style.
- Many of you will be familiar with the discussion of using tabs or spaces for indentation. No, I'm not going to continue that discussion. Whatever you use in your team, make sure it is clear for everyone.

- Automating most of these code style rules and guidelines is the best solution. A lot of IDEs have this integrated, or they can be installed via plug-ins.
- The easiest one, across multiple languages and code editors, is editorconfig. By adding a `.editorconfig`, these rules will be applied.
- You can set a lot of settings for your project in those files. It is possible to set them globally and for specific languages. For example:
- Indentation style: tabs or spaces
- Quote type: single or double
- Max length
- Character set

Code Like You're in a Team — Even a One-Person Team

- Last but not least, write like you're in a team!
- I can imagine it's very hard for people who have never written code in a team to understand what that's like.
- But if you're coding a project on your own, it's very tempting to write code that only you understand (e.g. writing unclear variable names, using 2-3 character variable names, etc.).
- Try to write your code as if you're in a team. Imagine that your code is so clear that someone else can easily understand your code.
- You can easily test this by asking a friend or someone via Twitter in the developer community to check your code's

readability. I can promise you'll get the feedback you never thought of.

- Don't panic about negative feedback! Just focus on the feedback that can make your code readable for someone else.
- You should know there is a very thin line between readable and not-so-readable codes. It's based on a person's opinion. Don't feel bad if someone tells you your code is not readable! Be grateful for the feedback.