

# **CSC207 README Checklist: “Have I Made a Good README?”**

By Arist Bravo

So, you've written a README for your latest project on GitHub. Review this checklist to see if the README follows good practices for software documentation.

## **Structure and Basic Characteristics**

The README has all the following elements (or gives links to the following elements):

- Title of the project
  - Clear, bolded title
  - Located at the top of the README
- Authors and contributors of the project
  - Clearly indicated somewhere on the file
  - Is complete and does not omit any major contributors
- Summary of the project's purpose
  - Says what this project does
  - Says why the project was made
  - Gives the user a general sense of what problem this project solves and whether it is a useful project for them
- Table of contents
  - Has links to navigate to the major sections in the README
  - Links are not broken
- Features of the software
  - All major features of the software are described so that a programmer or user can determine whether the software is right for them
  - Descriptions are clear and only detailed when needed
  - Examples or tutorials (videos, screenshots, or code snippets) are used to clarify the software's features
- Installation instructions
  - Clear, accurate descriptions of how to install the project from start to finish
  - Mentions all packages and software which must be downloaded for the project to work
  - Provides links to any other packages or software needed
  - Includes all technical requirements and information needed to download each part of the project
  - Includes required versions for all packages or software needed

- Explains if the software is only meant to be installed on a certain OS, hardware system, or otherwise (for example, Windows only or Mac only)
- Mentions common issues in the installation process and how they can be overcome
- Examples or tutorials (videos, screenshots, or code snippets) are used to clarify the steps for software installation
- Examples or tutorials (videos, screenshots, or code snippets) are used to clarify the steps for overcoming common issues
- Usage guide (instructions on how to use the software)
  - Examples or tutorials (videos, screenshots, or code snippets) are used to clarify how to use the software once it is downloaded
- License (for how others can use the code)
  - License is clearly displayed or visible in the GitHub project
  - License is legally valid
  - License is consistent with the details of the project (for example, it would not be consistent if the README claims the project is “in the public domain” while the project has an MIT License)
- Section for feedback (how to give feedback on the project)
  - Clearly says how users can give feedback on the software (via Google Forms, a discussion board, etc.)
  - Clearly provides any necessary links (for example, to a Google Form)
  - Has rules for what counts as valid feedback
  - Has guidelines for what to expect when submitting feedback
- Section for contributions (how to contribute to the project, or if contributions are closed in the first place)
  - Clearly says how users can contribute to the project
  - Clearly describes instructions for making a fork of the project on GitHub
  - Gives guidelines for creating a good merge request
  - Describes protocols for reviewing contributions and merging them into the project

## Writing Criteria

- All information is accurate
- All sentences are unambiguous
- The README is complete in the sense that it (a) explains each part of the project or (b) references all documents which, taken together, explain all parts of the project

- The README is easy to use and navigate, integrating hyperlinks and numbered lists when necessary
- The README's information is kept up-to-date each time the project is updated
- Technical jargon is only used when necessary

### **Visual and Technical Criteria**

- The README is a Markdown file entitled README .md
- All text is easy to read in terms of size, colour, and font (using a minimum 12-point font size); one should not have to squint to read the text, and one should not feel exhausted while reading the text
- Size, colour, and font are consistent for each type of text (heading, sub-heading, paragraph text, code snippets, etc.)
- Headings are large, bolded, or formatted in a similar way, to convey importance
- Sub-headings are distinct from paragraph text but are smaller than headings, to convey the fact that they are headings of lesser priority
- Hyperlinks are easy to distinguish
- Videos, screenshots, .gif files, and images are not broken; they are displayed when viewing the README in the GitHub webpage
- Videos, screenshots, .gif files, and images do not need to be downloaded if they are only being used to document the software
- Videos, screenshots, .gif files, and images are non-distracting and easy to understand (especially if they are being used to explain a feature or a set of instructions to the reader)