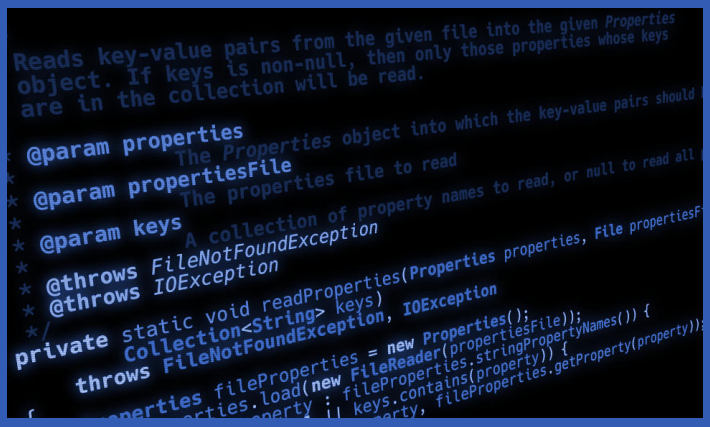

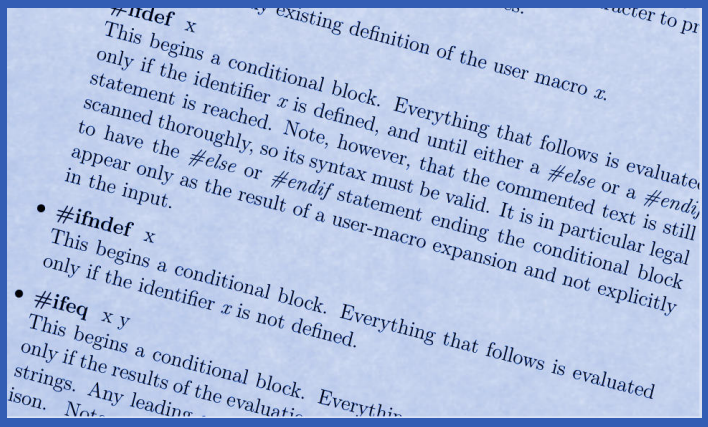

Assignment 2

COMP 2600: Technical Communication in Computer Science

Winter 2025

University of Manitoba

Contents

1 Assignment overview

Task

Produce and publish an appropriately documented static website on a forge.

Learning goals

1. Produce a resume.
2. Understand Etter's book *Modern Technical Writing* by using the current, popular technical documentation tools he describes—in particular:
   a) a lightweight markup language,
   b) a distributed version control system,
   c) a static site generator, and
   d) a forge.
3. Write a README that teaches Etter's key principles, and demonstrates their application, by instructing the reader how to host a resume on a forge that offers static web hosting.

First steps

Resume

You need a resume for this assignment. You are encouraged (but not required) to use your own resume.

Markdown

You need to learn Markdown. We spent some class time on this. Depending on the forge you select to host your website, you may need to learn a particular variant of the language, such as GitHub Flavored Markdown. It may be helpful for you to use an editor that supports Markdown via syntax highlighting and/or a live preview. (Refer to the lecture slides for some options.)

Distributed version control system

You may use any distributed version control system supported by the forge you select. Git is overwhelmingly the most popular such system, and you are strongly encouraged to use it. You must familiarize yourself with its basic operations, including creating or cloning a repository, adding files to it, committing

changes to it, and pushing local changes to a remote repository. Depending on the static site generator and forge you select, you might also need to learn about creating and using branches in Git.

Static site generator

You may use any static site generator that supports Markdown. For the in-class exercises, we used the Python-based Pelican, but there are many other options, including the Ruby-based Jekyll and the Go-based Hugo. If you are already experienced with one of these static site generators, consider learning a different one for this assignment.

Forge

You may use any forge that supports your chosen distributed version control system and the hosting of static websites. (Nearly all popular forges support Git, but not all of them can host static websites.) For the in-class exercises, we used the GitHub Pages feature of GitHub, but there are many other options, including the GitLab Pages feature of GitLab and the Codeberg Pages feature of Codeberg. If you are already experienced with one of these forges, consider learning a different one for this assignment.

2 Audience, content, and formatting

Audience profile

The audience for your README file is Marvin McLaren, the Foomatic finance manager from Assignment 1. After you taught him about APIs, he became interested in some of the documentation you led him to online. He has developed a fascination with technical writing, and you have seen him reading Andrew Etter's *Modern Technical Writing* in the lunch room. He has no experience with Git, static site generators, or forges, but has been learning how to use Markdown. He uses the same operating system that you do, and while he prefers using its default graphical user interface, he has a very basic understanding of the command line and is able to do very simple operations like change directories and run commands.

You may write your README to address Marvin specifically, or you may write it in a slightly more generic way, so that it would benefit anyone with similar knowledge and skills as Marvin.

Resume

The resume should be written with a particular position or employer in mind. (When you eventually do apply for jobs, you will usually tailor your resume for each application.) There is no minimum or maximum length for the resume.

README

Your README has two goals:

1. Describe the practical, technical steps of how you formatted and hosted your resume for this assignment.
2. Relate these practical, technical steps to the general principles of technical writing as explained in Etter's *Modern Technical Writing*.

The README must contain, at minimum, the following:

A title.

A statement of purpose. This should indicate what the README describes and who it is for.

A set of prerequisites. This should indicate what resources the reader needs in order to carry out the instructions.

A set of instructions. Provide a sequence of instructions, appropriately structured with headings and/or ordered lists. Follow the relevant Instruction Guidelines from Chapter 8 of William S. Pfeiffer's *Technical Communication*. (See the marking scheme for further details.) Also, make specific references to Etter's recommendations wherever appropriate. (You may want to start every item in the sequence by referring to some general principle of Etter's, and then describe a practical step that shows how to apply this general principle. Or alternatively, you may want to start every item in the sequence with a practical step, and then relate it to some general principle of Etter's.)

Further resources/readings. Include a list of links to online resources that provide additional relevant background information, further reading on the subjects or software you've introduced, or other resources that may be useful. Include at least four links, one of which should be to a Markdown tutorial *other than the one used in the in-class exercise*.

A FAQ. Include a FAQ section with at least two questions and their answers. At least one question should be about the overall process, such as "Why is Markdown better than writing raw HTML?", and at least one question should be about the practical details, such as, "I changed the Markdown version of my resume, so why don't I see the changes when I refresh the website in my browser?" (You may use these two examples, or you may come up with your own.)

Credits. Add a list of contributors of the README (including all group members who contributed to the in-class peer review) and credit the sources of any third-party content (static site themes, images, etc.) you used to produce your website.

The README should be 1000 to 1500 words in length.

Static site

It is acceptable for you to use the default theme (i.e., the design template) provided by your static site generator. However, you are encouraged to try modifying the default theme, or replacing it with another theme. (A web search for the name of your static site generator plus “template” or “theme” will turn up lots of third-party themes that you can use.) It is acceptable for you to use third-party themes and images for your static site as long as the licence permits you to do so, and you credit the creators in the manner specified by that licence.

Add your resume to your statically generated website as a page or blog post. Make sure that the resume, or a link to it, appears when you visit your website’s home page. The README file should *not* be part of the generated website. Rather, it should be stored somewhere in the source tree—ideally in the root directory, where it will be easy to find when someone views or checks out your project’s source code from the forge.

3 Submission instructions and deadlines

In UM Learn, upload a ZIP archive of the complete source tree (*not* the generated HTML version) of your static site, with the filename `LastName_FirstName_A2_source.zip`. This source tree must contain your Markdown-formatted resume and README file.

In the text box input field for the assignment submission, add the following:

1. The name of the static site generator you used
2. The name of the forge you used
3. A link to the website source repository hosted on the forge
4. A link to the generated website hosted on the forge

Prepare (but do not submit) a draft version of your files by **Thursday, 2025-03-06 at 11:29** and bring them to class that day for a peer editing exercise. You should then submit the final version of your files on UM Learn no later than **Thursday, 2025-03-06 at 23:59**.

4 Marking scheme

The assignment will be graded out of 20 points as follows:

Resume (2)

Deduct up to 1 mark if the resume contains significant errors in Markdown syntax resulting in misformatting of the rendered version hosted on the forge.

Deduct up to 1 mark if an inadequate effort was made to convert the resume to Markdown (i.e., little or no use of Markdown markup, or excessive use of inline HTML, or inappropriate use of markup).

Deduct up to 2 marks if the resume is missing altogether, very obviously incomplete (e.g., only a tiny amount of text), or contains completely nonsensical content.

README style, structure, and formatting (6)

Deduct up to 1 mark for each of the following:

- Significant errors in Markdown syntax resulting in misformatting of the rendered version hosted on the forge
- Inadequate use of Markdown (i.e., little or no use of Markdown markup, or excessive use of inline HTML, or inappropriate use of markup)
- Not appropriately structured with headings, lists, etc.
- Inadequate or confusing organization of the individual sections, paragraphs, sentences, etc.
- Shorter than 1000 words or longer than 1500 words
- Style, tone, vocabulary, level of detail, etc. inappropriate for the intended audience (Marvin McLaren, or someone like him)

README instructions (4)

Deduct ½ mark each if any of the following Instructions Guidelines from William S. Pfeiffer's *Technical Communication* have not been adequately followed:

- Guideline 1: Select the correct technical level
- Guideline 5: Place only one action in each step
- Guideline 6: Lead off each action step with a verb
- Guideline 10: Keep a simple style

Deduct up to 2 marks if the steps do not meaningfully refer to relevant concepts from Etter's book, or if the explanations of those concepts are incorrect. (It's not necessary for *every* step to refer to Etter's principles, but those steps that are relevant to Etter's principles should include such a reference.)

README other contents (6)

Deduct 1 mark each if any of the following are missing altogether; deduct ½ mark each if any of the following are present but incorrect, incomplete, or otherwise problematic:

- Title
- Statement of purpose
- List of prerequisites
- List of at least four links to further relevant resources/readings, at least one of which should be a Markdown tutorial
- FAQ section with at least one question (and answer) on the overall process and at least one question (and answer) on the practical details
- Credits for all peer review group members, plus creators of any third-party material used for the website

English (2)

Grade with the following scale:

- 2: Text is formulated in a way that is syntactically correct and easy to understand; any errors are minor/infrequent and do not significantly detract from meaning or purpose
- 1½: A few isolated passages are awkwardly phrased or hard to parse; any other errors are minor/infrequent and do not significantly detract from meaning or purpose
- 1: A text with several passages that are awkwardly phrased or hard to parse, or with other errors whose nature or frequency make them somewhat distracting or slightly obscure the text's meaning or purpose

- ½: A text with several passages that are awkwardly phrased or hard to parse, and with other errors whose nature or frequency make them somewhat distracting or slightly obscure the text's meaning or purpose
- 0: The text is replete with passages are awkwardly phrased or hard to parse, and with other errors whose nature or frequency make them very distracting or greatly obscure the text's meaning or purpose

Additional deductions (up to −20)

- Rendered website not successfully hosted on a forge's web hosting service, or no link to the website provided (−5)
- Source code (Markdown files, README file, static site generator configuration files) not successfully hosted on a forge's source code repository, or no link to the remote repository provided (−5)
- README file is included in the rendered website (−1)
- Complete draft of the assignment not ready at the beginning of class for peer review (−4)
- Assignment does not indicate the name of the forge or static site generator (−1)
- Filename not to spec (−1)
- File type not `.zip` (−2 if the grader can still safely and conveniently open the file; otherwise −20)
- Submission up to 48 hours late (−4)
- Submission more than 48 hours late (−20)