README document

Project documentation is to be compiled as a single markdown file named README.md. This file should contain,

No.	Requirement	Word Count
R1	Answers to all the *documentation requirements* below.	
R2	Your README.md should have a separate heading for each *documentation requirement* and answers organised under the appropriate headings.	
R3	Provide full attribution to referenced sources (where applicable).	
R4	Provide a link to your source control repository	
	Design a Software Development Plan for a terminal application. The following requirements provide details of what needs to be included in this plan,	
R5	Develop a statement of purpose and scope for your application. It must include: - describe at a high level what the application will do - identify the problem it will solve and explain why you are developing it - identify the target audience - explain how a member of the target audience will use it	300 - 500 words
R6	Develop a list of features that will be included in the application. It must include: - at least THREE features - describe each feature Note: Ensure that your features above allow you to demonstrate your understanding of the following language elements and concepts: - use of variables and the concept of variable scope - loops and conditional control structures - error handling Consult with your educator to check your features are sufficient .	300 words (approx. 100 words per feature)

No.	Requirement	Word Count
R7	Develop an outline of the user interaction and experience for the application. Your outline must include: - how the user will find out how to interact with / use each feature - how the user will interact with / use each feature - how errors will be handled by the application and displayed to the user	
R8	Develop a diagram which describes the control flow of your application. Your diagram must: - show the workflow/logic and/or integration of the features in your application for each feature. - utilise a recognised format or set of conventions for a control flow diagram, such as UML.	
R9	Develop an implementation plan which: - outlines how each feature will be implemented and a checklist of tasks for each feature - prioritise the implementation of different features, or checklist items within a feature - provide a deadline, duration or other time indicator for each feature or checklist/checklist-item Utilise a suitable project management platform to track this implementation plan > Your checklists for each feature should have at least 5 items.	
R10	Design help documentation which includes a set of instructions which accurately describe how to use and install the application. You must include: - steps to install the application - any dependencies required by the application to operate - any system/hardware requirements	

Slide deck

You must present your Terminal application and it's code - which will form the basis of a professional report.

The presentation and associated Slide deck should include,

Present your Terminal application to the class. You must provide a walk-through of the logic of your application and how the application is used.

- A walk-through of your Terminal application, its features and how it used
- A walk-through of the logic of your Terminal application and code
- A review of your development/build process including challenges, ethical issues, favourite parts, etc

Make sure to incorporate the following into your presentation Slide deck,

No.	Slides must include…	You must explain…
R11	An overview of your Terminal application	The main features and overall structure of your *app*
R12	An overview of your code	An explanation of the important parts of your code

Presentation Requirements

Your presentation must:

- be no longer than 10 minutes (or the time limit specified by your Educators)
- utilise the submitted Slide deck

> You may also be required to answer any questions from your Educators and the audience

Code Requirements

No.	Requirement
R13	Implement features in the software development plan you have designed. You must utilise a range of programming concepts and structures using Ruby such as: - variables and variable scope - loops and conditional control structures - write and utilise simple functions - error handling - input and output - importing a Ruby Gem - using functions from a Ruby Gem
R14	Apply DRY (Don’t Repeat Yourself) coding principles to all code produced.
R15	Apply all style and conventions for the programming language consistently to all code produced.
R16	Creates an application which runs without error and has features that are consistent with the development plan.

No.	Requirement
R17	Design TWO tests which check that the application is running as expected. Each test should: - cover a different feature of the application - state what is being tested - provide at least TWO test cases and the expected results for each test case > An outline of the testing procedure and cases should be included with the source code of the application
R18	Utilise source control throughout the development of the application by: - making regular commits (at least 20 commits) with a commit message that summarises the changes - pushing all commits to a remote repository
R19	Utilise developer tools to facilitate the execution of the application: For example, - writing a script which turns the application into an executable; OR - packaging the application for use as a module or dependency

Submission

- This project must be submitted via Canvas
- Your submission is to be a single zip file (organised as described in the Deliverables section below)

File

Follow the steps below to correctly prepare file(s) to submit,

1. Create a directory called,
   {Fullname}_T1A3

   > Where,
   >
   > - {Fullname} is to be replaced by your Fullname
   > - T1A2 is this Assignment's ID.

2. Organize all the files for submission in this directory

   > Follow the guidelines provided in the *Deliverables* section below

3. Create a zip file of this directory called,
   {Fullname}_T1A3.zip

> A sample command that can be run in Mac OS X/Linux terminal to achieve creation of the zip file is as follows,
>
> zip -r {Fullname}_T1A3.zip {Fullname}_T1A3

4. Submit this zip file in Canvas

Example

> If your name is Luke Skywalker,
>
> 1. The directory you create will be called LukeSkywalker_T1A3
>   mkdir LukeSkywalker_T1A3
> 2. Organize all the files for submission in this directory
> 3. Create a zip file of this directory. The zip file will be named LukeSkywalker_T1A3.zip
>   zip -r LukeSkywalker_T1A3.zip LukeSkywalker_T1A3
> 4. Submit LukeSkywalker_T1A3.zip in Canvas

Deliverables

Following is a list of deliverables (which should cover all the above requirements) to include in your submission. The table also details how the information should be organized.

Deliverable	**Description**	**Location/Folder (in Zip file)**
README.md	General project documentation is to be compiled as a single markdown file named README.md.	/ (Root folder of your zip file)
Resources	All files linked by the README.md file must be included in a folder named docs. All resources included in this folder must be in either png, jpeg, pdf, or markdown (md) format	docs/
Presentation	Slide deck of your presentation in pdf format	ppt/
Source Code	Source code for your entire project	src/

> Note: All links to online material should have corresponding screenshots included in submission.