Tool to visualise command line programming

Kristin-Davida Agyekumhene

4th Year Project Report
Software Engineering
School of Informatics
University of Edinburgh

2023

Abstract

Statistics show that despite Bash/Shell being more loved than hated, there are less beginner programmers using the text-based scripting language than professional developers (Section 2.1.1). Thus, this project details the creation of a tool to encourage more beginners to learn how to program in command line interfaces. The tool takes the form of an educational website, on which users can read more about the command line, and the benefits of command line programming. The main functionality of the website is to provide users with an environment, in which they can practise coding in emulated command line interfaces. Subsequently, users are given support and guidance in command reference guides, detailing the commands' use, and worked examples, to visually support their understanding. Across a selection of three tasks achievable through command line programming, users can choose from five command line interfaces and begin their self-guided learning journey on command line programming. Evaluation of the website discovered that the simplistic design of the website allows for an improved user experience and ease of navigation across the website. The professionalism demonstrated through the design and functionality helps gain users trust in using the website, and the unique all-in-one style, in which all resources are centralised, is highly attractive to beginners. This contributed to study participants deciding that the tool is something they would use and recommend to others.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 548020 (RT #7061)
Date when approval was obtained: 2022-11-03
The participants' information sheet and a consent form are included in Appendix B.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Kristin-Davida Agyekumhene*)

Acknowledgements

I would like to thank my supervisor, Dr Brian Mitchell, for his continued support throughout this project. Not only has he helped me formulate my ideas, he has been encouraging every step of the way and for that, I will be forever grateful.

Additionally, I would like to thank my mum, dad, sister and family (my number one fans), who have been there for me from the very start. I hope that I have made you very proud.

And to my friends and anyone else who has supported me through my final year at university, I could not have done this without you too.

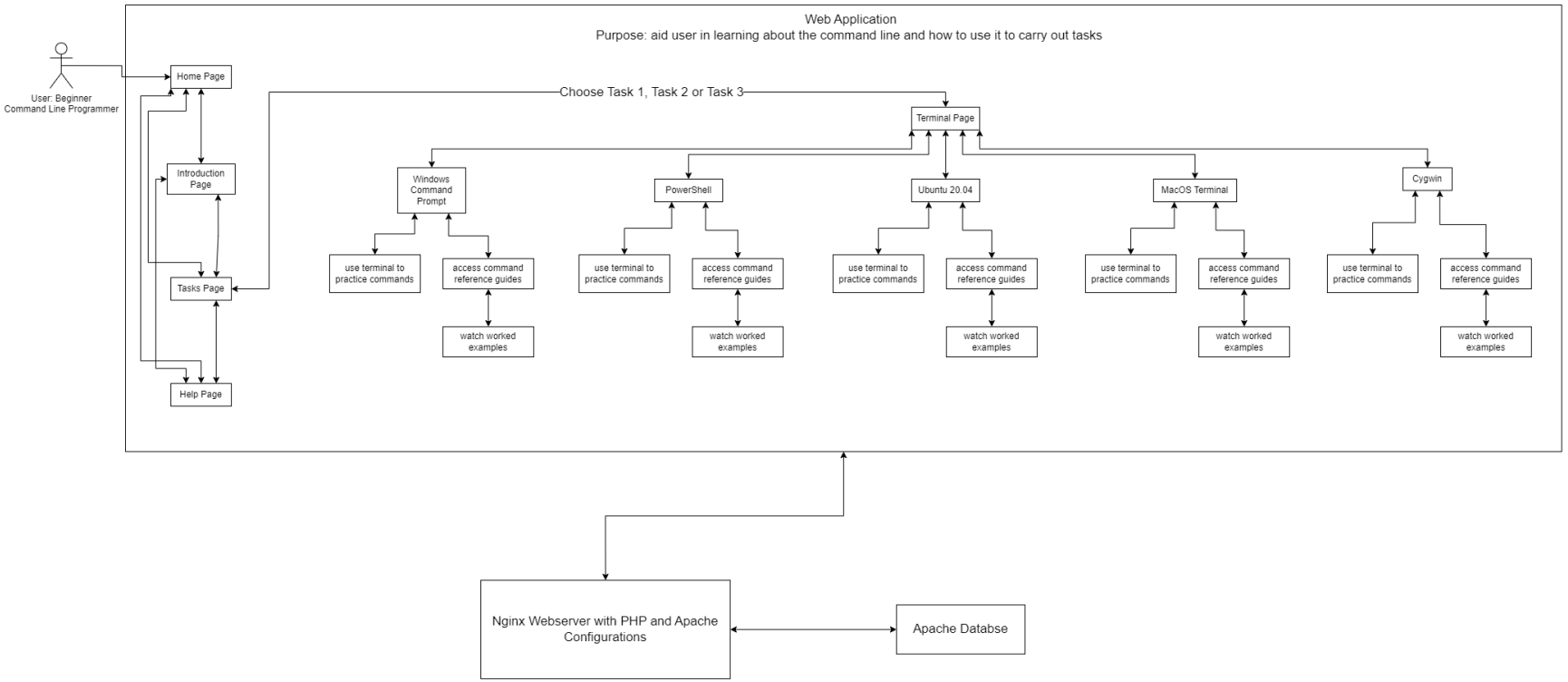


Figure 1: System Diagram showing the overall system and all the component interactions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ability to program in a command line interface opens endless possibilities for automation and efficient programming. Moreover, features hidden from or unsuited to graphical user interfaces can be accessed — all by using a few lines of commands. These commands have the power to manipulate data, extract key information, and summarise findings. Additionally, maintaining version control within a project, using command line functionalities, proves essential for 'developers and system administrators' today (Sampath, Merrick, and Macvean, 2021). In 2013, Shell was deemed to be the language that 'interoperates the most with other languages' (Bissyandé et al., 2013), with the command line based scripting language appearing in 'over 14 thousand [GitHub] projects, [despite being] the main language in only 3 thousand (22 %) [GitHub] projects' (Bissyandé et al., 2013). The presence of these capabilities and characteristics serve to emphasise the importance and merit of command line programming.

The command line is 'ubiquitous' (Kiili, 2022), meaning its benefits are accessible to all. Many operating systems, such as Windows and MacOS, pre-install a command line interface on their computer devices, and 'almost all major cloud providers offer command line interfaces' (Sampath, Merrick, and Macvean, 2021). Yet, not everyone makes use of them. Whilst there exists many reasons for this, the prevalence of graphical user interfaces has enabled them to become the preferred way of interacting and communicating with a computer, instead of using text-based user interfaces like the command line.

Thus, this project is motivated by helping the everyday user see the benefits of the command line. This will be achieved by helping users understand how to use different interfaces, in attempts to improve their ability to use the command line. Hence, these interfaces will become more accessible, and users will be able to take advantage of the power of command line programming.

A limitation of this project is the inability to code the functionality of all Bash commands using the jQuery framework. Whilst this is made possible with the PHP scripting language, the user requires additional file write permissions to execute all Bash commands. This poses a potential security threat to the system, as full system access rights enables the user to perform malicious system modifications. Another limitation of this project is

the scope of the command line, making it difficult to provide a complete overview of its full potential. In this way, educating users on a select few tasks reduces the amount that can be learned about the command line. This clearly provides opportunities for the expansion of this project as future work. Further difficulties are explored in Section 1.5.

1.1 Command Line Programming 101

Command Line Programming 101 is a self-guided educational website. Aimed at helping beginner command line programmers, regardless of prior coding experiences, the website educates on selected commands used to navigate around different terminals and achieve various everyday tasks. Users can read explanations of the commands' uses, supported by worked examples. Supplementary to that, all acquired knowledge can be practised in emulated terminals displayed within the website. This requires no installation or terminal set up, which could be of convenience to a beginner.

The website enables users to build a knowledge of the command line and understand how it works. By showing them ways in which they can use the command line, to achieve tasks without the use of graphical user interfaces, the tool fosters an interest in command line programming for all computer users. Extensive searches could not find the same all-in-one service provided by the tool, or something similar, in existing command line programming tools.

1.2 Project motivations

Taking the time to reflect on how user-friendly command line interfaces are is of high importance. It should be determined whether all computer users can interact with them easily and effectively, in a way that could achieve the full potential of the command line. This evaluation is especially key when reflecting on the ability of those who have no prior programming experience.

By measuring effectiveness, efficiency and satisfaction, 'relative to the context of use' (Brooke, 2013), ISO 9241-11 presents a revised definition of usability, in attempts to highlight that the quality attribute is caused by interaction, instead of being 'a property of a product' (Bevan, Carter, and Harker, 2015). This definition has been internationally agreed on by experts before publication, like all standards produced by the International Organization for Standardization (ISO), enabling this to be used as a credible definition throughout the project.

In Human-Computer Interaction, recall is defined as 'the retrieval of related details from memory' (Budiu, 2014). Nielsen's (2020) 10 Usability Heuristics favours 'recognition rather than recall', as the average short-term memory is limited. This reliable source is written by Nielsen himself, who is renowned for web usability, and his heuristics have been widely adopted in the field of Human-Computer Interaction. An experiment conducted by Nobel and Shiffrin (2001) discovered that the average reaction time 'in recall was significantly higher than for recognition'. Hence, this favours recognition for the ability to process and react faster than when using recall. To minimise reaction times

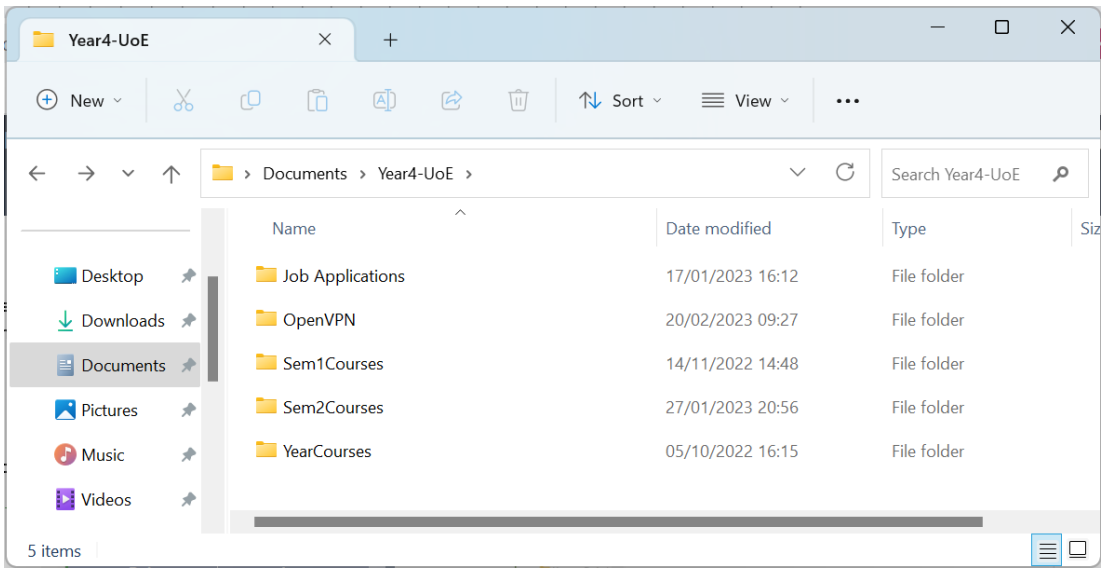


Figure 1.1: Graphical user interface showing the contents of my 4th year university directory

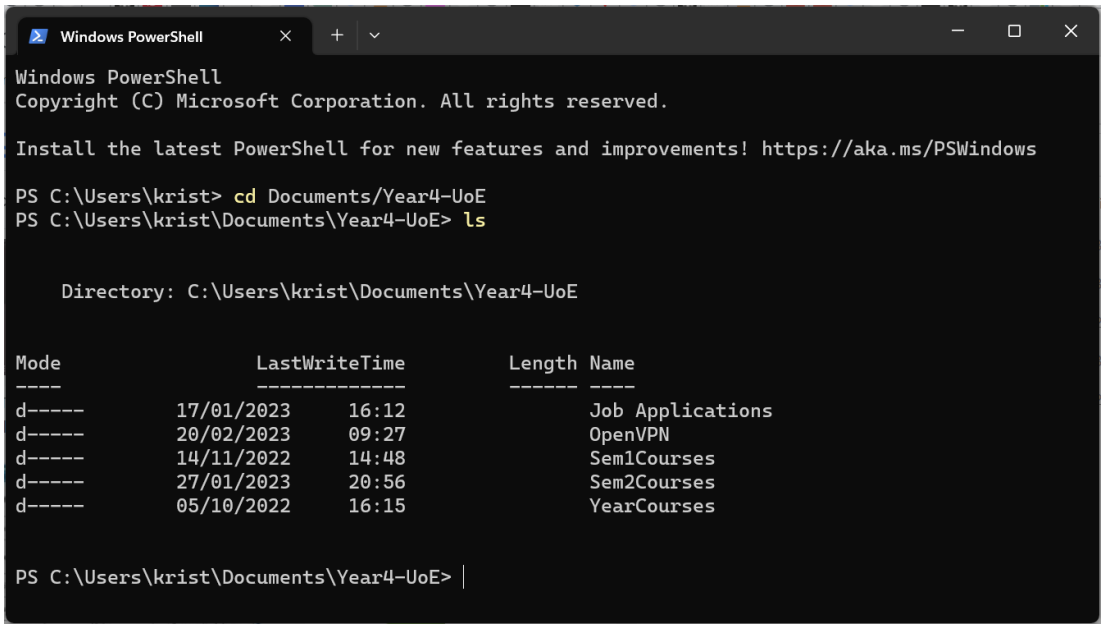


Figure 1.2: Command line interface showing the contents of my 4th year university directory

as a cause of the dependency on users' memory, the heuristic highlights the importance of the visibility of all options, elements and possible actions. This reduction in recall means that users are required to remember less information about the interface, given that all information is visible to them or easily accessible.

Command line interfaces rely heavily on recall, compared with graphical user interfaces. Graphical user interfaces reduce the need for recognition as commands presented in menus can be easily accessed, thus ' simplifi[ng] the usability of computers' (Hoyer and Brown, 1990). Figures 1.1 and 1.2 demonstrate the same action of displaying

file directory contents. The intuitive design and visual representation presented in the graphical user interface depicted in Figure 1.1 enables ease when navigating around the file system. The text input required for the command line interface, shown in Figure 1.2, can be more complicated for users to engage with and recognise than using the visual components in graphical user interfaces. Recognition supports the interaction with icons in Figure 1.1, whereas recall is necessary to identify the correct commands to perform the action in Figure 1.2. The comparison of these figures illustrates a limitation of command line programming, as there exists a simplicity to interacting with visual components in a graphical user interface.

In command line programming, most of the available commands and their parameters need to be memorised by users. This is on account of command line interfaces not providing an extensive range of features, to help users recall commands when programming. To access help functions which provide more details about the command's syntax and use, the command in question needs to be identified before this information can be provided. In this way, these interfaces do not appeal to the non-expert demographic of users. This is due to the lack of usability experienced as a cause of the dependency on memory recall. Therefore, it becomes difficult to interact with these interfaces, when a user lacks the required knowledge on a vast range of commands.

These limitations motivate this project, focusing on creating a tool to visualise command line programming. This project builds on existing tools and work in this area, providing a tool which can be used to learn the importance of the command line. The reflection of previous and existing works helps this project to overcome criticisms and limitations, to deploy a user-friendly tool aimed at beginners, and more inclusively, experienced developers. The tool encourages recognition rather than recall and provides users with an environment, in which they can receive support and guidance in becoming familiar with command line interfaces and subsequently programming in them.

1.3 Project objectives

1. Due to minimal prior command line programming experience, I must learn how to program in the command line, learn more commands which I can use and understand the potential of the command line.
2. Understand the functionality of command line interfaces on multiple operating systems and the differences in command syntax.
3. Create an easy-to-use graphical user interface (GUI) which helps beginners to learn the knowledge and commands required to gain an understanding of the command line and how to use it.
4. The GUI should be accessible, reliable, intuitive, and beneficial to the users' learning experience.
5. Promote Nielsen's heuristic 'recognition rather than [memory] recall', to decrease the dependency on the user's memory. This will reduce their reaction times and thus, improve their website interactions. This heuristic is discussed in Section 1.2.
6. The usability of the tool is evaluated against efficiency, effectiveness, and satisfaction.

7. Encourage users to consider using the command line, rather than using graphical user interfaces by default.
8. Improve my web development skills and learn how to properly use CSS frameworks, which are essential to implementation.

1.4 Importance of the project

Beginning to use the command line can be overwhelming, especially for avid users of graphical user interfaces. Very little information is required to be learnt about graphical user interfaces before they can be used. This is because the visual elements make interaction with these interfaces simple and intuitive. However, command line programming relies on existing knowledge about the set of commands available. Thus, referring to manuals becomes essential in helping a new user become familiar with the command line environment and programming in such interfaces.

There are abundant resources found both online and in physical copies, providing an introduction to command line programming. Examples of these resources are discussed in Section 2.2. Using these resources builds the foundational knowledge necessary to begin using command line interfaces and hence, reduce any confusion or difficulty experienced. Yet, there are limitations to existing resources on this topic. Such limitations include, but are not limited to, the cost required to access some resources, and the layout and format of written guides overwhelming users with paragraphs of text.

Therefore, this project is important as it prioritises educating users on the command line and its potential, and encourages them to practise programming. The project attempts to overcome the limitations of existing resources discussed in Section 2.2, to produce a more unconstrained learning tool for beginner command line programmers.

1.5 Difficulty of the project

Before the creation of this tool is possible, learning about the command line, its importance and how to program in it, are crucial in improving my understanding of this task. Without this step, it will be near impossible to create the intended learning tool.

The tool takes the form of an online education website, where users can learn how to program in the command line. To achieve this, WebStorm IDE was used to create and populate the required HTML, CSS and JavaScript files, generating the programmed website.

At the forefront of the website is the design. A design which encourages users to use recognition rather than recall is important within this project, as discussed in Section 1.2. Several design iterations are essential, to ensure the design is planned, evaluated and refined, before implementation occurs.

This tool requires the emulation of multiple command line interfaces. To achieve this, the JavaScript framework, jQuery, needs to be explored and then used to replicate the set of interfaces available on the website. They need to be designed and programmed

to function as expected, supporting the required range of commands and operating systems. Each command used needs to have its behaviour defined in JavaScript, so that the command behaves as expected when used.

Additionally, explanations and examples of the uses of the commands need to be provided to the user, to support their learning of the command line fundamentals. This is required for all the provided tasks that the user can learn about on the website. A good knowledge of the commands is required for this task, to ensure that the functionality is clearly expressed to users. The worked examples must demonstrate the correct behaviour of the command, to ensure nothing is misconceived.

1.6 Target audience

The target audience of this project is beginner command line programmers, that is, those with no or little experience programming in the command line. This is irrespective of their programming experiences in other languages. By targeting beginners, command line programming fundamentals become the focus for the project and the tool can be created as an introduction to the command line. For those who already have some experience with the command line, the tool could be useful in revising some commands and practising pre-existing knowledge. Therefore, whilst the system is built to suit beginners, it will be inclusive to more experienced command line programmers.

Adapted from the results obtained from the project's preliminary lab study questionnaire, detailed in Chapter 4 and Appendix E, the following user stories were created to increase the understanding of the user requirements and functionality of the tool.

1.6.1 User story A

"As a beginner, I want to be able to utilise the command line to commit files to my personal GitHub repositories for effective version control and code collaboration."

This user story highlights an interest in a particular task that could be learned using the developed tool. It is important to consider user preferences like this. This is because when users have a vested interest, the tool will be more appealing.

1.6.2 User story B

"As a beginner, I am not sure what I would like to learn in particular as I would already have learned the commands myself. In this way, a well-rounded overview of the topic would help, providing me with a good foundation to command line programming, one which I can develop overtime in the future."

This user story helps to prioritise offering a range of features and tasks which users can use and discover more about. This targets users who do not have any prior knowledge of the topic. In particular, these users know they want to learn more about command line programming, but perhaps do not know where to start in their learning process.

1.6.3 User story C

"As a beginner, there is nothing I would like to learn in particular about command line programming, as I prefer to use graphical user interfaces."

This user story represents the section of the target demographic who do not have an interest in the topic at all. They have a preferred alternative method, using graphical user interfaces, and because of that, they believe learning about the command line is not a necessity. Users like these should still be catered for, even if they are unlikely to use the tool. For these users, they require a good learning experience and enough foundational knowledge, so that they are made aware of the benefits to command line programming, and have the opportunity to compare this method with another they typically use.

1.6.4 User story D

"As a beginner, I want to be able to learn commands which I could use for particular functions. In particular, I would be interested in learning how to restart my computer using command line programming."

This user story provides a feature that the user may want to see on the tool. However, with such a specific action, it may not be possible to include all actions that every member of the demographic would want to see. Some actions may not be plausible to emulate. In this way, the best approach is to implement a variety of popular features, in order to cater to the whole audience. Specific requests could be incorporated into the future development of the tool, in which more features can be developed and deployed.

1.6.5 User story E

"As a beginner, I am indifferent to what I learn or do not learn surrounding the topic of command line programming."

This user story does not provide particular information on what the user aims to achieve with this tool. However, the story makes it known that not all users know what they want from an educational tool, especially when they lack knowledge of the topic. From this perspective, it is key to strive to provide users with a good foundation for the topic, and more importantly, foster an interest in the command line and programming within different interfaces.

Chapter 2

Background

2.1 Command line programming

'Before the rise of GUIs in the late 1980s' (Milligan and Baker, 2014), it was common to interact with computer devices through the command line. The command line is a text-based tool, used to receive lines of specific commands as a user input. These commands provide information on what actions the user wants the computer to perform. A command line interface, or terminal, is the application, specifically user interface, in which the commands are entered. Figure 2.1 displays a Windows Command Prompt terminal, which is the default command line interface for the Windows operating system.

Entering pre-defined commands is how we program in the command line and there are various tasks achievable in doing so. Tasks include, but are not limited to, the creation

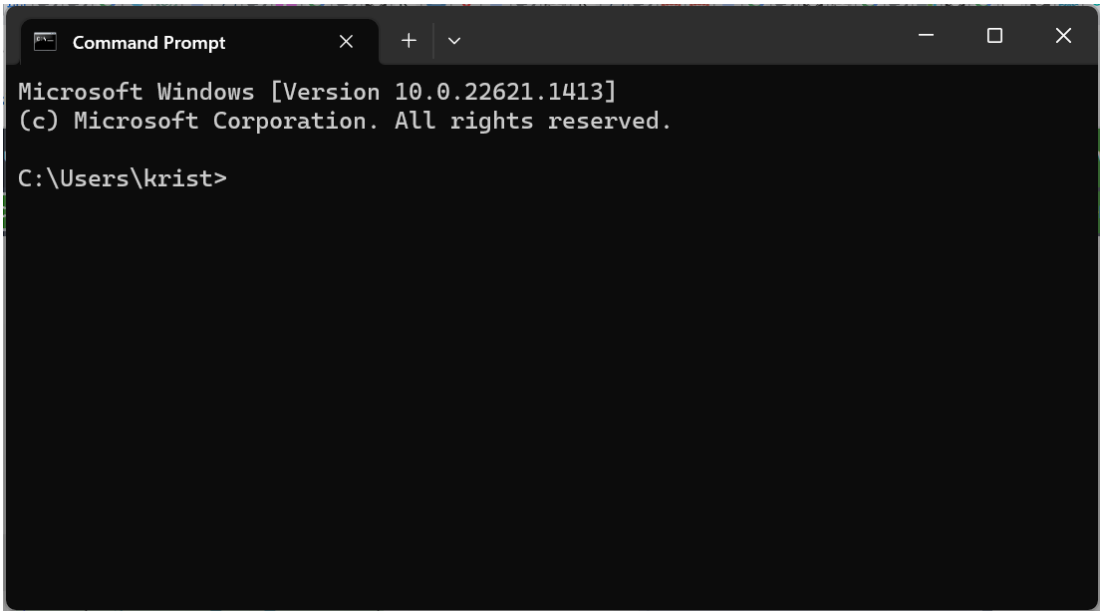


Figure 2.1: Windows Command Prompt, the Windows operating system's default command line interface

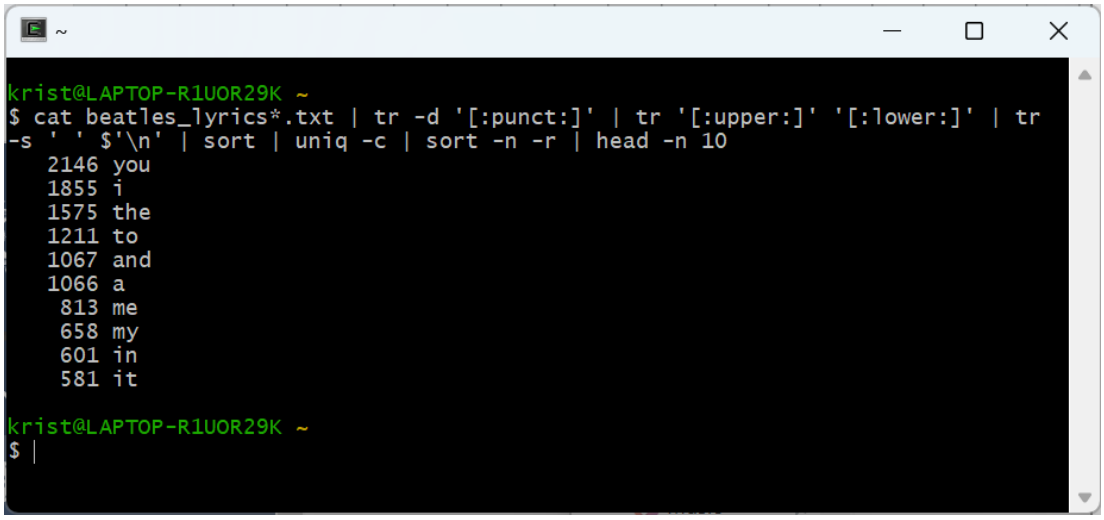


Figure 2.2: A Cygwin64 terminal showing a command line program. The program displays the 10 most common words from all the Beatles' lyrics

of files and file directories, the extraction of data from text files, and the extraction of files from a zipped directory.

Figure 2.2 shows an exemplary command line program, and the use of commands to achieve an action. The commands used ask the computer to return the 10 most common words appearing in the lyrics from all songs made by the Beatles. To achieve this, the current directory contains text files. The file name begins with 'beatles_lyrics' and the file contents are the lyrics of songs by the Beatles.

The command 'cat beatles_lyrics*.txt' accesses all files in the current directory beginning with 'beatles_lyrics'. To delete all the punctuation, as these do not identify as words, the command 'tr -d '[:punct:]' ' is used. Next, the file contents are made lowercase, and each word is placed on a new line. This is achieved by typing 'tr '[:upper:]' '[:lower:]' | tr -s ' ' $'\n' '. The command 'sort' orders the contents into alphabetical order, and 'uniq -c' counts the occurrence of unique words. To retrieve the 10 most occurring words, the contents are reversed and sorted numerically - so that the most common word appears at the start of the file. Finally, the top 10 words of the file are displayed in the interface. 'sort -n -r | head -n 10' achieves this action.

For many tasks, doing them by command line programming can be a lot more efficient, especially as command line programming promotes automation, 'unlike GUIs' (Kiili, 2022). Users can create and store scripts to automate a range of tasks, which can then be executed on multiple different future occasions. By using a few commands, tasks can quickly be completed which, without the command line, could have taken longer. With a graphical user interface, it potentially could have required the use of multiple mouse clicks and potential navigation.

As built-in commands can only be accessed through the shell, developers can achieve greater control over system functions. Automation also allows for ease when carrying out data science tasks, such as those which make use of text files, where data can be extracted and processed through the command line, rather than manually. Naturally,

the ability to use git version control, to track and manage changes to files, documents, records and data sets, also makes the command line highly desirable. Version control (ReQtest, 2020) is key for developers to collaborate on projects efficiently as they can work on the same files concurrently, whilst working independently. For some tasks, there are no graphical user interfaces available and therefore interacting with the command line is the only option. A notable example is interacting with a cloud platform server, as there is no GUI which enables users to carry out this task, given that a 'cloud server refers to an entire Linux environment' (Garnett, 2022).

When programming, the shell is the program which executes the commands and provides the output. Many operating systems have their own default shell installed, but additional shells can be downloaded and used as desired. Different shells have different syntax for their commands and their behaviour can vary. In this way, it is possible to install multiple shells, using them for different tasks.

Existing shells include, and are not limited to, Bash and Zsh. Bash and Zsh behave similarly as they are both Posix compliant, with Posix being the 'standard that dictates certain characteristics and features that all shells should comply with' (Cocca, 2022). Bash is the most commonly and widely used shell, with Zsh being a similar, more recently developed shell.

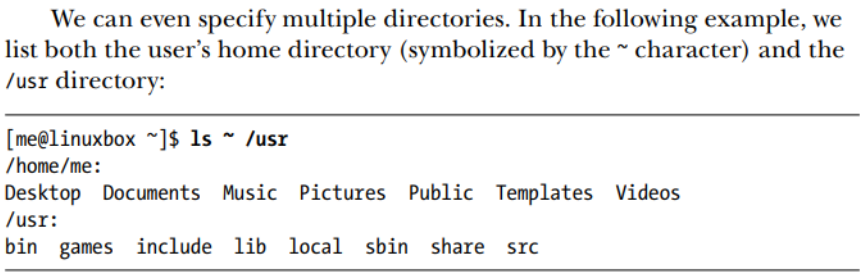
2.1.1 Bash/Shell popularity: Stack OverFlow developer survey 2022

According to Stack OverFlow's Developer Survey 2022, there are reportedly fewer beginner programmers (19 % of respondents) using Bash/Shell compared with professional developers (29 % of respondents) (Stack OverFlow, 2022). The survey confirmed the popularity of the command line when using version control, as 83.57 % of respondents reported using the command line to achieve this (Stack OverFlow, 2022). Additionally, respondents were asked whether they love or hate a range of programming, scripting and markup languages. Overall, 57.89 % of 71 467 respondents love Bash/Shell in comparison with the 42.11 % who hate it (Stack OverFlow, 2022). Despite the command line being less commonly used, it is clear from these statistics that the command line interface is popular amongst those surveyed. This opens the potential for the command line to be more commonly used and have an increased popularity, when the number of command line programmers increases. This can be made possible by educating more users on command line programming and portraying its benefits, to create an interest in the command line, hence motivating this project.

2.2 Selected previous work

2.2.1 The Linux Command Line, 2nd Edition: A Complete Introduction

The Linux Command Line (Shotts, 2019) 'acquaint[s readers] with the Unix way of thinking'. Clear detailed explanations are provided, along with terminal graphics and examples, seen in Figure 2.3, to guide readers step by step through the learning process.



We can even specify multiple directories. In the following example, we list both the user's home directory (symbolized by the ~ character) and the /usr directory:

```
[me@linuxbox ~]$ ls ~ /usr
/home/me:
Desktop  Documents  Music  Pictures  Public  Templates  Videos
/usr:
bin  games  include  lib  local  sbin  share  src
```

Figure 2.3: An excerpt from the Linux Command Line (Shotts, 2019), showing a command explanation, supported by an illustrative example

As a beginner command line programmer, I believe that this book is an excellent resource to read before starting this project. The language is simple, allowing for short, but informative, easily understood explanations. Phil Bull's review, author of the official Ubuntu documentation, states, 'if you would like to start using the command line, improve your existing skills, or simply want to discover tools that you were never even aware existed, this book has everything you need, and I wholly recommend it' (No Starch Press, n.d.).

2.2.2 Command line programming crash courses

Internet searches are flooded with websites offering (often paywalled) courses. Codecademy offer an eight hour beginner 'Learn the Command Line' course for their PRO users, which promises to teach participants how to navigate files and directories on their computer, as well as being able to access and modify them. This learning style is appealing as the website highlights to potential participants that, as of 12 October 2022, 1 083 892 individuals have taken the course. This statistic is acknowledged as being accurate, assuming Codecademy is conveying the truth. Although it is unclear how many participants have completed the course.

Mozilla Developer Network (MDN) Web Docs have published a written command line crash course. The course is aimed at those who have previous web development experience due to the HTML, CSS and JavaScript prerequisites. The course even provides an additional guide on how readers can add their personal tools to their command line interface. This resource is free to access on the internet, and includes graphics and breaks the learning process into steps. The guide requires learners to build commands and programs using their own terminal. Therefore, the resource begins with a step-by-step guide on which applications learners should install, and how to do so. However, there are limitations to the guide because users require some previous knowledge. As a result, this makes the resource less accessible for those who have no knowledge at all.

Like written guides, there are numerous videos of variable scope available on the internet, in which instructors deliver beginner command line programming courses to viewers. Online video features allow users to be guided through the process by an individual, and the pause feature allows the viewer to progress at their own speed.

Limitations to the previously discussed written resources include requiring users to follow a non-interactive step-by-step guide to gain knowledge of the command line. But the essence of a guide like Codecademy's means users need to go through the guide from start to finish, to effectively use the resource and achieve the stated learning objectives. As a result, users need to navigate to certain parts of the guide if they have particular things they want to achieve. Depending on the guide, this could be easily done. However, this could create problems for users. An instance of this occurs when guides are based on an example that keeps getting built up on. This could make it hard to follow if not used chronologically. Because of this, such guides should follow a format like MDN's, in which users can choose which parts of the guide they read. They should be able to read each section in isolation. This key feature allows for flexibility in using the resource. Such a feature could appeal to wider audience of learners, that is learners who have partial knowledge and do not wish to follow a guide from start to finish.

The use of examples to aid understanding is key to the learning process. At the same time, examples can also hinder it. When learners are faced with a fixed set of examples, there is only a set way for them to comprehend the explanation. For some learners, they may require a different explanation or set of examples to help their understanding. But with such guides, this is not always possible. Such a resource could benefit from supporting users through the learning process, by providing an environment in which learners can practise what they have learnt. If this is not plausible, users should be encouraged to continue practising in their own terminals. This makes it more flexible for users to try out examples which they think will be helpful to them.

An additional problem is the number of hours that are required to complete many of the online guides. Whilst for many this may not be a long time, for users who want to learn quickly, this duration might be unappealing. Codecademy's 8 hour course on fundamentals seems at first glance like an excessive amount of time to complete a course on the basics of the command line. Other users may disagree and find it to be sufficient. For those who find it too much, this could prevent them from starting the course.

The guides also require users to follow along on their terminals and does not provide any environments in which they could practise small programming exercises within the guide. Such a feature could centralise the resources and tools needed to complete the learning process. This could be beneficial to users, making it easier for them to follow along with everything in one place.

A user study would be significant to explore the extent to which these limitations arise and create problems. This would help gain an insight into users' opinions on existing learning resources, shaping the system requirements. From this, it will be more evident what to consider when building the system.

2.2.3 ShellOnYou: learning by doing Unix command line

Berry et al. (2022) analysed the ShellForYou tool and evaluated its usage across four student cohorts. The tool is a free web application which aims to support users in learning the Unix system. It achieves this by offering exercises which can be completed

to improve the students' understanding of the practical knowledge required to successfully use 'a Unix-like operating system from the command line' (Berry et al., 2022). Naturally through this study's analysis and evaluation, Berry et al. were able to identify some advantages and disadvantages of the tool. They present the use of online tools as highly beneficial to learners, justifying that the flexibility offered means that students can choose a convenient time and place to use ShellOnYou, given that they have an active internet connection. This analysis is key to the development of my learning tool, as it would be desirable if users have the flexibility of using the tool when they want, which is reinforced by the accessible nature of web applications.

Chapter 3

Design and implementation

3.1 System overview

The ‘Command Line Programming 101’ website is a system designed to allow users to learn about the command line and its potential. Users will learn a range of commands which can be used to achieve defined tasks and make use of worked examples to support self-guided learning. To practise their newly acquired knowledge, users can make use of the emulated command line interfaces. The website supports Windows, Mac and Linux operating systems and allows for users to practise using the Cygwin, Windows Command Prompt, PowerShell, MacOS Terminal, and Ubuntu 20.04 interfaces.

This website comprises of web pages defined by a combination of HTML and CSS scripts, which are formatted and designed using TailwindCSS components. These web pages are connected to ensure that the user can navigate around the website with ease. To centralise resources and improve the user experience, the system will emulate a variety of command line interfaces and their functionalities, made possible by the jQuery Library, and JavaScript and PHP scripts. This means that the user does not need to make use of any additional resources and everything required for full functionality is found in one place.

To support incremental development and version control, this project is maintained in a private GitHub repository, to keep a track of changes and allow for previous versions to be retrieved if required.

3.1.1 Home page

The home page is designed to attract the user and provide them with a brief overview of the website and what it aims to achieve. This page is key in ensuring that the user is interested in the tool and wants to use it. In this way, it is essential that this page is visually enticing and instantly captivates the attention of the user.

3.1.2 Introduction page

The introduction page provides the user with the foundational knowledge required to understand what exactly the command line is, its uses and reasons as to why it can be more beneficial than graphical user interfaces. The intentions of this web page is to ensure that the user leaves the website with a better understanding of what knowledge they had, or did not have, about the command line, with the hopes that they feel more comfortable to go on and use it to program in.

3.1.3 Help page

The purpose of the help page is to guide the user through the website's intended use. By providing instructions for use, users can refer to this page, to ensure they use the website correctly and effectively, and to resolve any functionality problems that may arise.

3.1.4 Tasks page

The tasks page displays all the available tasks that the user can practise and learn the commands for. The task summaries give users a brief overview of what they will learn and achieve within the task and will help support their decision in selecting the tasks they wish to use. The buttons allow for navigation into that specific task page.

3.1.5 Task 1: Working in the terminal

In this task, users will become familiar with the appearance of different command line interfaces. They will learn how to use and navigate around the command line interface with the support of commands, which are most applicable, and guides on their functionality and usage. To support their understanding, a range of worked examples are provided for each command to visually aid learning.

3.1.6 Task 2: File management

The file management task teaches users how to navigate through file directories and create, modify, and delete files and directories. With a different list of commands, users can again achieve this task using supplementary command explanations.

3.1.7 Task 3: Data extraction and manipulation

This task involves working with files and being able to extract data, modify and manipulate them. Command explanations are available as supporting material for the recommended commands found in the reference guide.

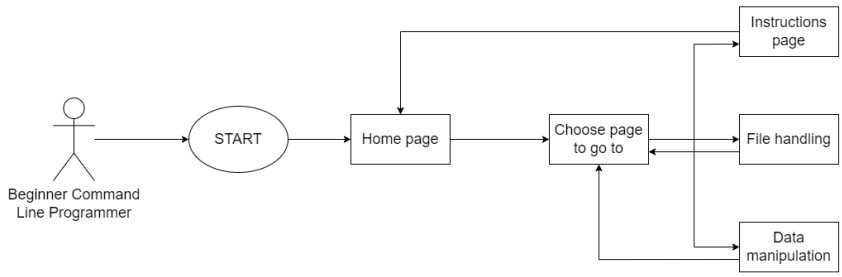


Figure 3.1: Basic system navigation overview diagram depicting the available web pages and how to navigate between them

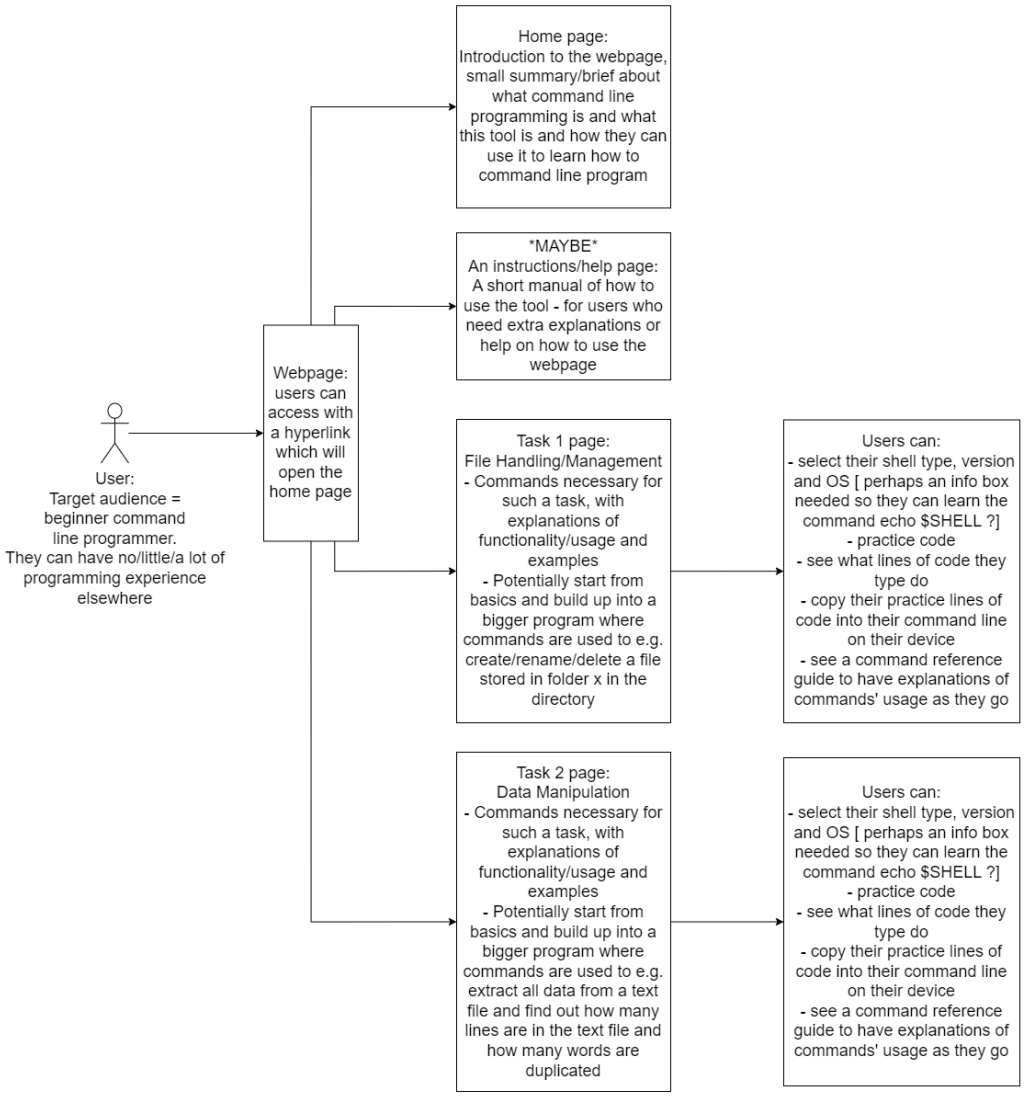


Figure 3.2: System diagram based on Figure 3.1 with extra details regarding web page usage

3.2 Overview diagrams

The system overview and functionality are initially depicted in Figures 3.1 and 3.2. These initial designs provide a basic overview of the web pages to be included within the system and how they are connected. By detailing the connections between pages, the navigation of the system can be displayed which is key when designing and implementing the connections between web pages. These diagrams were used as a rough outline throughout the design process.

A complete system overview diagram of the completed system can be viewed in Figure 1. This diagram demonstrates the user's interaction with the system, as well as how all the different software components interact with the system and each other. The main components of the system are the web application, web server and database server. These are essential for the running of the client-side and server-side scripts concurrently, to achieve the overall system functionality. Within the web application, the user interaction with the various web page is depicted, to demonstrate the flow through the web application.

3.3 Functional system requirements

The System **MUST**:

1. always be available and accessible to users
2. be easy to access
3. allow users to read about the command line
4. allow users to learn how to use command line interfaces
5. allow users to decide which command line interface they learn with
6. allow users to learn how to use the fundamental commands
7. be easy to use by all users
8. be easy to maintain and extend its functionality

The System **MUST NOT**:

1. contain web pages which lead to unavailable pages
2. confuse users on the topic

3.4 Non-functional system requirements

The System **SHOULD**:

1. be beneficial to users
2. follow good HCI design principles
3. appeal to everyone within the target demographic
4. be comprehensible to non-computer scientists
5. be able to run on different operating systems
6. provide a reliable service for users

The System **SHOULD NOT**:

1. have slow performance speeds
2. have duplicate information
3. be very similar to existing systems
4. rely heavily on external resources

3.5 Engineering requirements

When building a system, it is important to evaluate which applications and resources would prove most beneficial for efficient implementation. These justifications can range from being the most resourceful to the most familiar, with each being advantageous in different ways to the developer. For this system, applications and resources to support prototyping, the website implementation and command line emulation are required. Additionally, a web server is required to run back-end PHP scripts.

3.5.1 WebStorm IDE

The website was developed using the JetBrains WebStorm IDE. As a fond user of the JetBrains IDEs, it was the right decision to use WebStorm IDE as it was a familiar environment. The IDE had everything required to develop the HTML, CSS and JavaScript files to build the different web pages. The user interface allowed for the project to be easily backed up and pushed to the project's GitHub repository. For these reasons, no alternative IDEs were considered for use throughout this project, and no problems have been encountered using the WebStorm IDE.

3.5.2 TailwindCSS

In web development, there exists several modules and software that aid with development. The open-source CSS framework, TailwindCSS, provides a range of defined CSS classes. These classes can be used to design and style each HTML component. Using this framework can save a lot of time in the implementation stage, as the relevant classes can quickly be applied and adapted to style the web page elements. Making use of the framework will allow more time to be spent developing and testing the system functionality, with the main design decisions being determined during prototyping. Having previous experiences using CSS will allow for a better understanding of how the TailwindCSS framework can be applied and support the progression of the project.

3.5.3 Nginx web server

To host the website, a web server was required. The web server was initially set up in a tutorial run by my supervisor in October 2022, in which he guided us through the process of creating a VirtualBox virtual machine and installing all the necessary configurations to form the web server. Guidance on how to complete this process was collated from tutorials by Boucheron (2020) and Glass (2020). When this server was initially created, my focus was on interface design, rather than implementation. As a result of this, I was unsure whether the server would eventually be used, meaning it was unused until it became apparent that PHP scripts would be required. This occurred much later in the implementation process. At this stage, attempts to reconfigure the existing server were unsuccessful, had led to the creation of a new server and I concluded that I lacked the knowledge on how I could make use of it. After reading online guides, I sought help from my supervisor due to the experience he has in this area. With my host screen shared, we were able to work through the problem step by step and he was able to provide some

clarity on how I used the server to run my website. Following this call, I was sent a follow-up email, in which my supervisor attached an additional tutorial by Gangwar (2022), which would help guide me through setting up the correct configurations, to enable PHP scripts to be run on the back-end. This was successful, due to the improved understanding I had of the functionality of a web server, and I was able to run the generated PHP scripts without issues. With this server, all my files required for the website are contained within the /var/www/commandline101/html folder on the server, and with the Nginx and PHP configurations, the website can be launched in the virtual machine's host Firefox application, by running http://commandline101/Home.html.

3.5.4 Command line emulation

A key aspect of the tool is the emulation of various command line interfaces. This was made possible by making use of JavaScript libraries, which were created for this exact use. They allow for command line interfaces to be created and personalised, and in turn, allowed for the website to host personalised command line interfaces with the required commands for users to learn to use and practise with. The current website uses the jQuery library to do this. However, Xterm.js was also considered as a potential alternative to achieve this. Whilst they both have their advantages and disadvantages, jQuery proved the most beneficial and useful to the task at hand.

Xterm.js is a TypeScript module used in several existing real-world applications, allowing terminals to be built and used in the browser. When experimenting with this module, the provided documentation had limitations in helping to understand how to personalise the created terminals. Since the website aims to present a range of terminals to users, the ability to personalise the terminals and make them visually appear like the equivalent real-world interface is a priority. Being unable to achieve this, especially with use of published documentation, meant that this module was not going to be sufficient, and alternative terminal libraries were considered.

The jQuery library was deemed more suitable for terminal emulation than Xterm.js. For this reason, this JavaScript library was used throughout the project, to create, visualise and style the required terminals. The jQuery library makes creating interactive web-based terminals possible, especially those where the user commands are defined by the developer. This allows for the differentiation in command syntax to be realised. Unlike Xterm.js, the jQuery terminal was easily styled with CSS scripts, in which all visual aspects of the terminal could be defined.

3.6 Design decisions

Prior to implementation, a Figma prototype was created, to visualise the website and its functionality. Interactions were included to demonstrate the navigation between pages and the interactivity of page elements. The planning of the website layout enabled improvements to be easily made and visualised. This was crucial for added ease during implementation, as the design, content and functionality had already been determined and demonstrated in the prototype. The frames designed within the initial prototyping stage can be found in Appendix A.

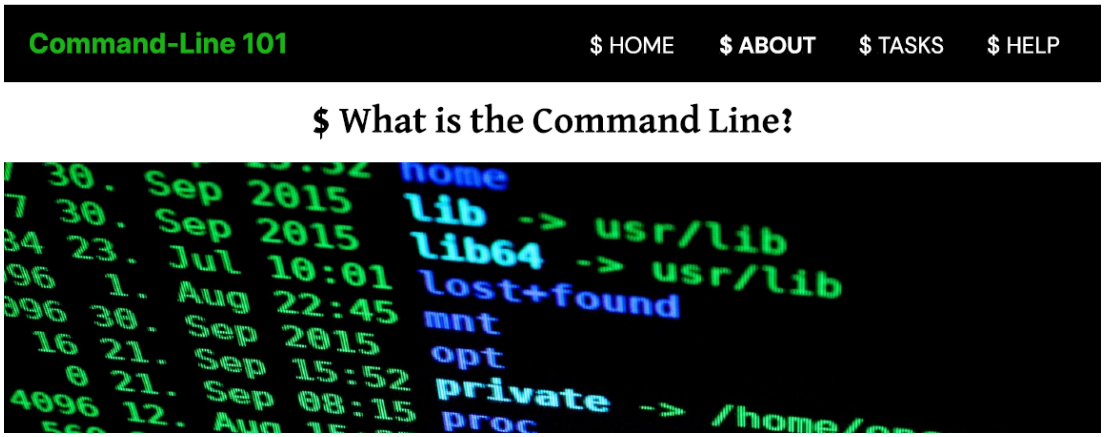


Figure 3.3: Prototype element showing the use of contrast, $ symbols and capitalisation

3.6.1 Adaptations made to the prototype

In a feedback session with my supervisor, I presented my prototype and we discussed all the visible elements and his comments and advice allowed me to make valuable revisions to the website design.

The first thing discussed was the website's foreground and background contrast, shown in Figure 3.3. The prototype included a darker background and lighter foreground with a vast contrast, making it less visually appealing. The use of capitalisation and centrally aligned text can be harder for users to read, especially for those with dyslexia. This is because 'there is a brief moment where the user has to find where the next line begins – decreasing the user's reading speed' (The Website Architect, 2020). As a result, refinements in typography were made at this stage.

The prototype included $ symbols shown in Figure 3.3, due to their presence in Linux command line interfaces, and images of the command line. However, they both lack descriptions which explain what they are and why they are being included. This makes it harder for beginners to understand the web pages content, especially when they have not seen these elements before. Because of this, I ensured that future images were described and relevant to the context, and removed the use of $ symbols within the design.

The design of the home page was identified by my supervisor as portraying the website as a commercial tool. Whilst at this point this was not something that I intended to prevent, I chose to adapt the layout and contents to focus the attention on the purpose of the website, rather than including several user testimonies.

The task pages presented a clear layout and involved the selection of the shell, operating system, terminal and task, presented in Figure 3.4. However, not every combination of options were possible, due to compatibility of the operating system. Therefore, it was suggested that I only display the possible options. This led to the creation of terminal pages, without a focus on choosing the operating system and shell, as only bash is available throughout the website.

Figure 3.4: Selection Panel on Task Page

Following the feedback session, amendments were made to refine the design of the website. This improved the usability of the website and achieved one of the objective's of the tool.

3.7 Implementation

Since the tasks had been determined, the relevant commands to be learnt had to be collated. A markdown file was created to list the commands according to task and provide a brief description and example of the usage of the command. This would help when developing the reference guides later in the implementation stage.

Using the improved design from the prototype, the home page was first implemented. To begin, the HTML file was created and populated with the link to the TailwindCSS style sheet within its <head> tags, as demonstrated in Listing 1.

```
1     <link rel="stylesheet" href="https://unpkg.com/
        tailwindcss@^1.0/dist/tailwind.min.css">
```

Listing 1: TailwindCSS Style Sheet

This ensured that any of the framework's classes used were able to be realised during rendering. Following this, the desired header layout was selected from TailwindCSS, using the Tailwind components website (*Tailblocks - Ready-to-use Tailwind CSS blocks*, n.d.). The header buttons were labelled within the HTML code to mirror the prototype and the page references were defined as '#', as the web pages the buttons navigated to had yet to be defined. The <body> tag was then populated with the main content of the

page. This includes a title, a brief description of the website, as well as a terminal graphic. Below this, the supporting operating systems are displayed, and the corresponding icons were sourced and coloured green to complement the website's colour scheme, before being included within the HTML.

The next page that was implemented was the introduction page. The header used on the home page was initially included, along with the link to the TailwindCSS style sheet. This ensured that the design remained consistent throughout. A different content layout was chosen for the main body, one which included a title followed by the main body text. The same terminal graphic from the home page was included, followed by a brief overview of what command line programming is. This content had been summarised from the background research undertaken towards the start of the project.

The help page HTML document was created with the header included, as a placeholder for the page. However, this page was implemented after the task pages, to ensure that the correct usage of the website was conveyed in the instructions, should the functionality change during implementation.

Following this, the layout for the tasks page was chosen and the page was created and populated with the relevant content regarding the available tasks and what they are. This page had not been designed in the prototype but I had decided to include it as a landing page for accessing all the available tasks, instead of requiring a tab in the header for each task or a drop-down menu. This decision was to make the tasks more accessible and the design of the website regarding this more intuitive.

```
1 <nav class="md:ml-auto␣flex␣flex-wrap␣items-center
     ␣text-base␣justify-center">
2   <a style="color:darkgreen" style="font-weight:
      bold" class="mr-5␣hover:text-gray-900">Home
      </a>
3   <a class="mr-5␣hover:text-gray-900" href="
      Introduction.html">Introduction</a>
4   <a class="mr-5␣hover:text-gray-900" href="
      Tasks.html">Tasks</a>
5   <a class="mr-5␣hover:text-gray-900" href="Help
      .html">Help</a>
6 </nav>
```

Listing 2: Header Navigation

When this page had been created, all the references in the header of the pages were filled out, to link the pages together, and ensure they were navigable from the relevant tab in the page header. Listing 2 presents the header navigation code contained within the Home.html file, and demonstrates the links between other pages, enabling navigation between pages.

The design of the tasks page was lifted directly from the prototype, with changes made to the colour scheme, based on the colour scheme already depicted on the website. Experimentation with jQuery terminals allowed for a terminal to be inserted into a column filling 60 % of the left-hand side of the page. The right-hand side was occupied

by the reference guide, and the commands required for the task were filled into the section. The terminal had, by default, a black background, and therefore personalisations were made to emulate the Windows Command Prompt terminal. This involved defining and adding the terminal header and prompt message, with the definitions shown in Listings 3 and 4.

```
<pre id="terminalHeaderMessage">Microsoft Windows
   [Version 10.0.22000.1335] <br>(c) Microsoft
   Corporation. All rights reserved.<br> </pre>
```

Listing 3: Windows Command Prompt Terminal Header Message

```
prompt: 'C:\\Users\\thisUser> ',
```

Listing 4: Windows Command Prompt Terminal Prompt Message

Following this, implementation of the commands required for task 1, titled 'working in the terminal', began within the terminal's JavaScript. Research into JavaScript syntax and functionality was required at this stage, to understand how the functionality of various commands could be programmed accurately to behave as the interface would on a host computer. This took around 1.5 weeks to achieve and test. It was during this stage that difficulty was met when attempting to code the functionality of the dir and ls functions. I was unsure of how to implement these within the client-side script, as the function needed to make use of a directory for this, and solutions provided in quick internet searches caused the terminal to stop working. It was important to progress with implementation of the other terminal pages. Because of this, I chose to leave the implementation of the dir and ls command, and instead duplicated the working HTML document, to create the other terminal pages. These duplicated pages were then personalised to visualise the corresponding terminal, and any differences in command syntax were noted in the command reference guide. The command reference guides were then populated with information describing the commands, their syntax and use, to help aid the user's understanding when reading the applicable guide.

At this stage in implementation, a trial test of the website provided feedback on the usage of the terminal pages. Key feedback at this stage was that when a terminal page was selected, it was not clear what the user was meant to use it for. Introductory text was implemented as a result of feedback. This text includes the terminal name, task name, and a brief description of how the user should make use of the page, to improve the usability of these pages.

My attention then focused on being able to use and manipulate file directories and files from the emulated terminals. This is because the process is essential for the development of the commands for the rest of the tasks. A deep search into this identified multiple ways in which this could have been achieved and all options were explored. These methods included but were not limited to Node.js and back-end PHP scripts. All researched methods discussed the use of a script that would run on a back-end web server to retrieve the outputs generated from the executed commands. My supervisor provided his students with markdown files and tutorials in which we set up VirtualBox Nginx servers during semester 1, in anticipation of any projects requiring a server. However, I found that during research, I lacked the knowledge of how to use the created server to achieve

what I wanted to do. A follow-up meeting was set up with my supervisor, to gain his help in understanding this task better, and configuring my server to allow it to achieve what I desired. The ability to implement a range of commands requiring file directory access was proving difficult during the project. However, the successful meeting I had with my supervisor was a turning point in my project and allowed me to overcome this obstacle. The PHP script defined in Listing 5 was called within all the terminals' JavaScript and run on the server, to retrieve the outputs generated from the task 2 and task 3 commands. However, the commands to manipulate text files were unable to be successfully implemented, due to user permissions. To promote security within the website, I avoided changing these permissions, meaning an alternative method for executing such commands would need to be explored in future development.

```php
<?php
  $command = $_POST["command"];
  $output = exec($command);
  echo $output;
```

Listing 5: PHP Script to execute bash commands and return the output

At this point, there was limited time left to complete the project. Therefore, I decided to not implement task 4 and included this as a future task available on the website. A task surrounding Git Version Control was also suggested during our semester 2 group presentations, and this task was subsequently included as a future task, shown in Figure 3.5.

To finish the implementation stage, worked examples were created for task 1, to improve the user's learning experience. These examples aim to support learners, who require seeing worked examples to solidify their understanding when learning new concepts. To achieve this, screen recordings were made and inserted into the relevant reference guides, to portray the use of the command, especially when parameters are involved. These videos are designed to run on a loop, and the playback controls are added to give the user the freedom to stop, play and rewind the video as desired.

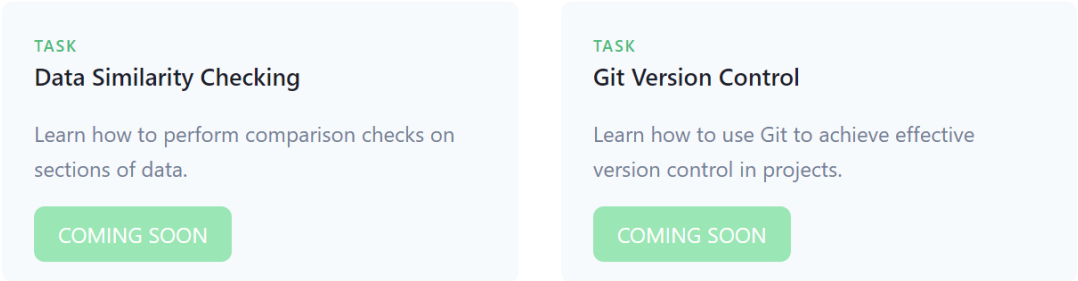


Figure 3.5: The non-implemented tasks for the website

Chapter 4

User study

An informative way to evaluate the tool was running a lab study to gain feedback from the target demographic on the functionality of the tool and the interface design. The lab study took place in-person during the week beginning 27 February 2023 and involved 5 students from the University of Edinburgh. A pre-study survey was initially completed by participants, to gain insight into their expectations from the tool and current background knowledge of the command line. The information gathered from this survey enabled me to incorporate features from learning tools they like using into the website. Following this, participants were invited to trial the website, and practise programming in task 1. This allowed them to see how the website functioned and to generate opinions on the website, to provide me with feedback. At the end of the study, a post-study survey was filled out, to provide a review on the website they just utilised, discuss the elements which they were impressed by and offer any criticisms and improvements to the tool, to improve their user experience.

This was an excellent opportunity to gain direct feedback on the tool towards the end of the implementation process, to allow for the tool to be developed further, while keeping the opinions of the target demographic in mind. Some feedback was implemented and feedback beyond the scope of this project is discussed in Section 6.2, detailing the way the tool can extend its functionality.

4.1 Ethics approval

As this study required the collection of data from participants, it was imperative to obtain ethics approval. Confirmation of this approval can be found at the beginning of this report, in the section titled ‘Research Ethics Approval’.

4.2 Method of study

The lab study consisted of questionnaires and usability testing. The usability test involves having participants use the system to achieve different tasks on the website. These tasks were accessing and using each web page, and being able to access and use

the ‘Working in the Terminal’ task on the tasks page. For the study, only three terminals were available for use: Cygwin, PowerShell and Windows Command Prompt. This was to give the users a restricted view of the website, to ensure that all participants were able to test the system. Since each terminal has the same functionalities, despite the appearance and sometimes command syntax, the need to trial all five available terminals was reduced. How the participants interacted with the system were noted for evaluation of usability and to what extent the system is used as intended. Any issues experienced when navigating the website were also recorded, to help with debugging and testing the system, and making the correct modifications to improve the overall user experience.

4.2.1 Dependent variables

The dependent variables of this study were the participants’ ability to navigate the website, the participants’ understanding of the information read, the participants’ ability to use the terminal resources, and the opinions on the usability and design of the tool. These variables are all considered in order to evaluate the success of the tool at this stage of the implementation process, and will help future development and improvements made to the website.

4.2.2 Independent variables

The independent variables of this study were the tasks and terminals used by the participants throughout the study period. This was limited due to the scope of the study and its intentions. Participants only had access to the first task, ‘Working in the Terminal‘, and three terminals, Windows Command Prompt, Cygwin and PowerShell. This allowed for the results gathered from the post-study survey to be less varied in scope and enabled a better evaluation of the responses. I believed that it was not essential to use each task and terminal, to make a judgement on the tool and gain a better understanding of its functionality and usability. This is because they all function similarly.

4.2.3 Control measures

To achieve consistency throughout the study, the website was maintained and kept in the same condition, so that study participants each viewed the same content. Additionally, each participant independently completed the study in a single 30 minute sitting. Each section was allocated a defined time in which it was to be completed.

4.2.4 Pre-study questionnaire questions

The selected questions for this questionnaire aimed to gain information regarding the educational identity and learning habits of the project’s target demographic. The ability to process this information and develop user stories based on the collected information is key to the design and implementation process, as well as the evaluation process, to ensure that the needs and expectations of the demographic have been met in the tool.

By enquiring about their existing knowledge of the command line and what they expect from learning tools, the tool can be shaped to conform to these expectations, by providing more information about the command line which participants do not already have, whilst implementing some learning techniques that participants are fond of to improve their user experience.

4.2.5 Post-study questionnaire questions

The post-study questionnaire aimed to gather feedback on the participants experience using the website, to evaluate the system and implement any improvements suggested by the target demographic. The questionnaire asked a range of questions regarding each individual web page, and asked the participants to provide a rating for each visited page. In doing so, users were able to highlight any features they liked and thought were successful in achieving its intended function, whilst also shedding light on aspects of the website they were not fond off or features they would have wanted to be included. As the researcher and developer, this aids my implementation process at this stage, whilst also promoting evaluation from the perspective of members of the target demographic.

4.3 Notable findings

4.3.1 Educational background

All participants (n=5) were School of Informatics students undertaking an undergraduate degree. Participants had a Computer Science background before starting their degree at the University of Edinburgh and learnt to program in either Visual Basic, Python or C#. Naturally throughout the course of their degrees, their knowledge of Computer Science has developed and their understanding of concepts has improved. Participants ranked themselves as either having moderate or lots of programming experience. This meant that these participants may find it easier learning to command line program and use the website, compared to members of the target demographic who have never learnt to program before. Whilst the feedback provided is useful for evaluation, it is important to acknowledge that the trialed participants are not fully representative of the target demographic. If this study was to be done again, potential participants' computer science background should have been identified, to trial a group who is more representative of the target demographic.

4.3.2 Pre-existing knowledge of the command line

All participants stated they had heard of the command line before taking part in this lab study. However, one out of the five participants admitted to not having any experience in command line programming. The feedback from this participant is crucial in identifying whether the website is comprehensible by a complete beginner, satisfying a non-functional system requirement and a project objective. Previously used command line interfaces included Ubuntu, Linux Bash Shell, PowerShell, MacOS Terminal and Windows Command Prompt. Uses of these command line interfaces detail coding, program execution, file management, git version control and zipping and unzipping

files. This demonstrates that the current scope of the website is acceptable, as the range of terminals included have been used by participants, and therefore suggests they would want to learn to program further in these interfaces.

4.3.3 Participants' learning outcomes

Participants in the study provided information into what additional information they wish to learn about the command line, which they do not already know. Suggested tasks were committing files to GitHub and having the ability to restart a computer using the terminal. Other participants were unsure of what they hoped to learn about the command line, mostly as they lacked knowledge of what they could learn about the command line and what is achievable using command line interfaces. Additionally, a user stated that they did not have any intentions of learning more about the command line as they prefer using graphical user interfaces. These task suggestions are useful in shaping the future development of the website, presenting tasks which users would want to see online, retaining their interest in the learning tool.

4.3.4 Learning techniques

Participants prefer using an instructional guide to help them learn a new programming language. In this case, styles of guides include, but are not limited to, online courses and tutorials, or step-by-step coding websites. Online tools are regularly used by the study's participants and they prefer those, in which they can choose what they learn, how fast they learn it and where they learn it. Interactive features were also popular amongst the participants, especially those which include a game-style element or an environment where altering, running and testing code is made possible. The participants agreed that online tools should be accessible, reliable and always available, with suggestions showing that they should also be free, reinforcing this project's functional system requirements. Examples of such tools are W3Schools, Udemy and YouTube tutorials.

4.3.5 System feedback

All participants rated the usability of the tool as very good. A participant stated 'everything is explained very comprehensibly. I think even a beginner would be comfortable using it.' To contrast, a different participant described having difficulties understanding what they should be doing when using the tool. To build upon this, a participant mentioned that since 'the terminal doesn't output error messages,' this could potentially have a negative impact on the usability of the tool. This limitation is crucial to address and resolve, preventing any misconceptions and confusion when learning to command line program on the website. Four out of five of the participants were certain that the tool achieves what it sets out to do, as it is informative, beginner-friendly and teaches about the command line. The other participant believed that additional information should be included, to 'add some sense of direction' to what should be done. Based on this, all participants agreed that they would use the tool to learn how to program command line interfaces, and subsequently, they would recommend the tool to other individuals for use. An average rating of 4.6 out of 5 was given to the tool.

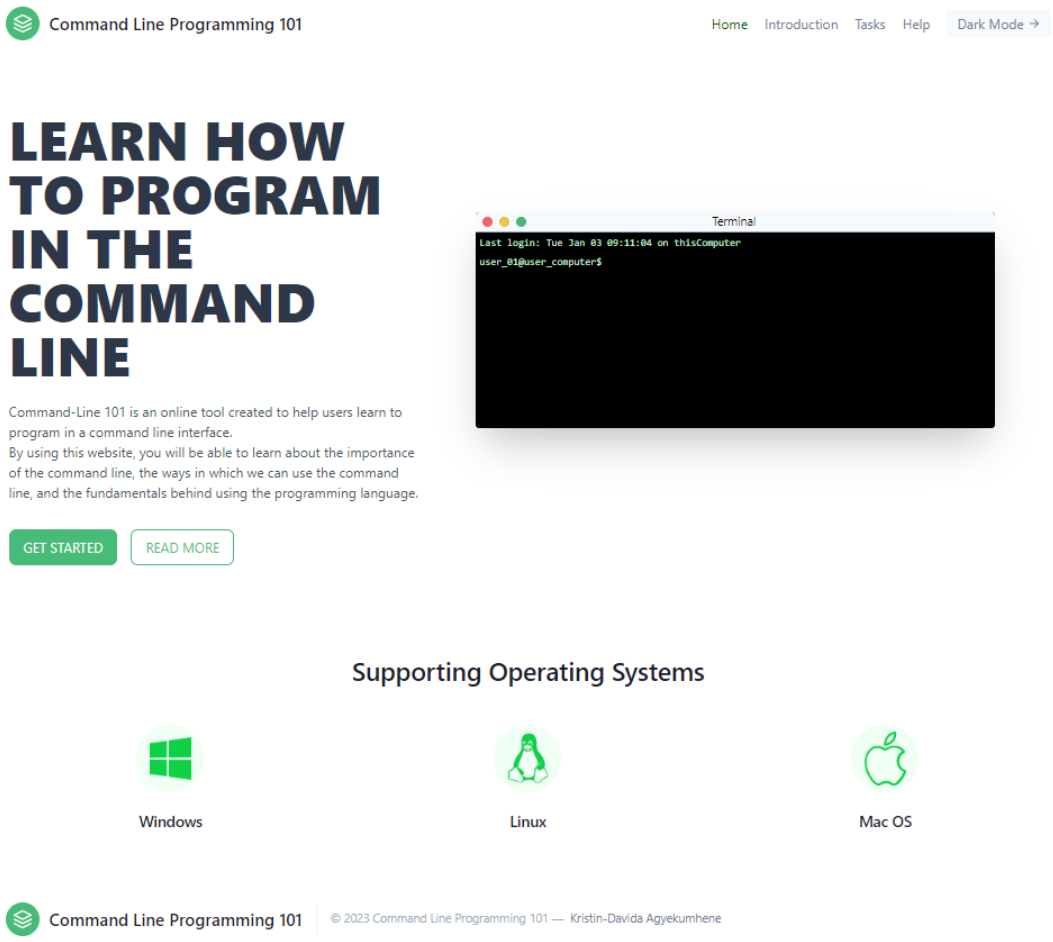


Figure 4.1: The Home Page of the Command Line Programming 101 website

4.3.6 Home page feedback

Participants who used the home page provided it with an overall rating of 5 out of 5. This was subject to the clear layout and visual appearance of the page. Participants described the page as ‘aesthetically pleasing,’ ‘very professional’ and one participant was in favour of seeing the supported operating systems displayed on the home page. The professionalism of the design encouraged a participant to trust the website, and the provided information was conveyed in a ‘comprehensible and intuitive manner.’ The website uses a green and white colour scheme, shown in Figure 4.1, meaning that there are not a wide range of colours on each page. This makes it less visually overwhelming and the simplicity allows the user to focus on learning and the functionality of the website, rather than being distracted by an onslaught of colours and a messy layout. The clear layout helps to promote recognition, as it becomes easier to view the websites contents, instead of having to recall where everything is placed when a messy layout and vibrant colours are incorporated.

4.3.7 Introduction page feedback

The introduction page was depicted as providing a ‘concise and comprehensive’ description of the command line. Participants wanted to view information about the project itself and how they will be learning about the command line and programming in its interfaces. A participant suggested the information could have been written in simpler terms, so that all beginners will be able to understand clearly what is being conveyed. As a result, the information has been simplified, as it is important that the information is comprehensible. Feedback on the design of the page was also provided and ways in which the page could look more visually appealing were detailed. This page was given an average rating of 4.4 out of 5.

4.3.8 Help page feedback

The help page was given outstanding feedback as it was detailed in structure, giving clear instructions on how to use the website. No critiques were made about this section of the system and the page was given a 5 out of 5 rating.

4.3.9 Tasks page feedback

While most participants were satisfied with the range of tasks available for use on the website, one participant noted that the range available will never be sufficient, given that there are endless opportunities for things that can be done on the command line. This highlights the possibility for future expansion of the website, to include a wider range of tasks available to learn. Despite this, the participants liked the simplicity of the task selection page and stated this made it easy to understand and view the range of available tasks.

4.3.10 Terminal page and worked examples Feedback

The visual appearance of the terminals was a selling point as the participants highlighted their resemblance to the actual terminals on their host devices, and found this to be beneficial when practising for using the real system equivalent. The use of the command reference guide and their pages was noted as decluttering the terminal page and navigation between these pages was done with ease. Participants suggested that information regarding the terminals themselves should be included, to help guide beginners judgement on which ones to use. The inability to enter commands which were not found in reference guides or the lack of error messages for syntactically incorrectly entered commands was a hindrance to the system and was viewed a potentially having a negative impact on the user’s learning experience. The worked examples were viewed as beneficial and helpful in aiding the user’s understanding of how the commands work. Overall, the terminal pages were deemed as positive in helping beginner command line programmers, as they can experiment with the commands, in a simple manner.

4.4 Evaluation of feedback

The tool intends to meet the participants' initial expectations. The command reference guides replicate the guides they prefer to use when learning a new programming language. Their preference for online tools, where they decide the ways in which they learn and can make use of interactive features, is met through the website's self-directed learning style and inclusion of the emulated command line interfaces where they can practise their code.

As this project involves creating a user interface, the design of the website is a big factor in determining the success of the tool. The remarks from participants regarding the visual appearance of the website and the layout of the pages demonstrate the positive impact the current website design has on the usability of the tool. Participants highlighted the importance that the clear layout had on their ability to easily navigate through the website and see all the information required. This suggests that this would help beginner command line programmers utilise the website, and thus be able to learn the fundamental commands and information required to improve their understanding and develop their knowledge of the command line. It was noted that participants did not feel 'overwhelmed' by choices, due to the complementary colours used and the simplicity of the design. They felt that this would help to engage users of the system, indicating a usable and effective aspect of the tool.

Another credible feature highlighted was the command reference guides. Not only did participants think that this made the terminal pages less cluttered, but they were also easy to navigate and provided clear and concise information regarding the commands, supported by the worked examples. This demonstrates the ease of use intended by the website, and shows that the website is successful in educating the users, in a manner that does not overwhelm them or hinder their learning experience. If anything, the use of the command reference guides appears to improve the users' experiences and thus, is a commendable feature within the website.

Finally, the contents of the website were praised. The information was described as being conveyed in a 'concise and comprehensive' manner, which helped users' ability to understand the presented text and instructions. Additionally, the grouping of commands into tasks and the range of tasks available were commended by participants. The ability to use the interfaces within the website was a selling point, and the resemblance to the real-world applications was noted as being beneficial to learners, due to the preparation they experience before being able to use the interfaces on their host devices. The worked examples helped to solidify the understanding of the information found in the command reference guides, with one participant sharing that they 'would have loved something like this' when they initially began learning to command line program. These comments combined reflect the positive impact the website's features and content had on the participants trialling the website, and suggests that future users within the target demographic would feel similarly.

Question	Answer
What did you not like about the terminal page?	Probably should add some information about the terminals themselves. A beginner might have no clue which one to pick.
	I didn't like that if I entered a command that was not in the list it would just ignore it and not execute it. If a command is correct it should still execute it, cause otherwise users will get confused. Also if a command is not syntactically correct it should throw the errors that a real terminal would throw.
	Would be great to have a description of what each terminal is, where you can find it, why you would use it.
Was the information in the command reference guides sufficient, clear and helpful?	Partially, could add information on parameters of commands

Table 4.1: Critiques given during the lab study regarding the terminal pages

4.5 Critiques and refinements

As anticipated, some of the biggest critiques regarded the terminal pages. The critiques, shown in Table 4.1, detailed the areas which are deemed as most integral to avoiding future users of the system experiencing issues. The terminals currently provide responses to the commands entered. However, this only occurs for the commands the terminal anticipates being entered, meaning those which are taught throughout the website. This has the potential to confuse beginner learners, and could lead to misconceptions forming when errors are not acknowledged and displayed. The learners can not learn from mistakes if they do not know if they are occurring and why they occur, and this is a crucial step in any learning process. This sheds light on the importance of error handling and has made this an important feature to develop within the website. Due to the scope of the project, errors are now identified in task 1 by the statement 'ERROR: please try again.' Further development of the project could implement more accurate diagnostic messages, like those displayed in the actual command line interfaces. This will improve the functionality of the terminals, so that they behave as anticipated. Due to the use of PHP scripts, error messages were unable to be implemented in tasks 2 and 3 in the same way as in task 1. Future work on this project can explore the possibilites of incorporating error handling within these tasks.

Additionally, participants suggested that information regarding the terminals themselves would be useful for beginners. This information can help them understand why particular interfaces are used, and the benefits to each, supporting their decision making on which interfaces they would choose to use in the future. Because of this, additional

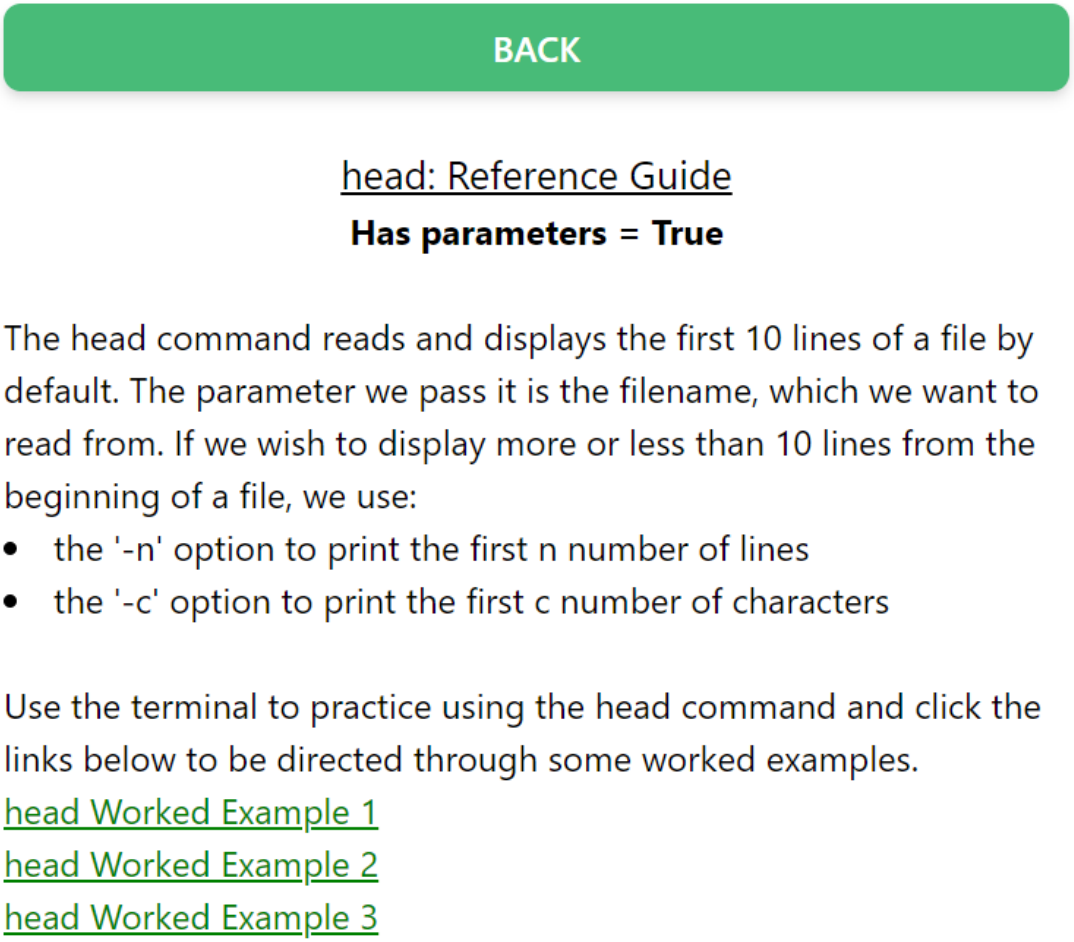


Figure 4.2: The Cygwin Reference Guide for the head command

information on this topic has now been included on the introduction page, to provide better background knowledge to the website's users.

Parameters of commands would like to be identified. Some commands which use particular parameters detail information about them and their usage. However, when a command does not require any parameters, there is no mention of them within the reference guide. To avoid any confusion, and ensure there is awareness of any parameters, or lack thereof, all reference guides will include a section on parameters, shown in Figure 4.2, and when no parameters are required, this is emphasised.

Chapter 5

Evaluation

When evaluating the overall success of the implemented tool, it is important to evaluate to what extent it is successful in achieving what it intends to and in presenting an effective design which promotes improved usability and fosters an interest in utilising the tool. Additionally, the functionality of the tool must be considered and its performance in this area evaluated. To conclude this, an investigation into the overall usability should occur, to ensure the tool does not contain any features or errors, which hinder the user's ability to use the tool, and evaluate the overall user experience of the website. These considerations contribute to the evaluation of the command line 101 programming website, and in turn, aid in making a judgement on the success of the tool.

5.1 Intention

The purpose of the Command Line Programming 101 website is to introduce beginner command line programmers, regardless of their previous coding experiences and educational background, to the command line. The website aims to provide information regarding what the command line is and what it achieves . Examples of different command line interfaces and the operating systems they typically function on are provided. Additionally, the website identifies the ways in which users can use the command line to make their everyday tasks achievable without using graphical user interfaces, and instead a few lines of commands. User evaluation, from those who fit the specifications of the tool's target demographic, suggests that the website is partially successful in achieving this. Limitations in this area include the lack of information on the website regarding this project itself, which command line interface is most suited to particular purposes, and the slightly limited range of tasks provided. To resolve these issues, more information is included on the introduction page which covered these areas, and a couple of tasks are able to be viewed as 'coming soon' on the tasks page, as an indicator that the available list is not extensive. The website can be further developed to cover a wider range of tasks and command line functionalities. Due to these improvements, I conclude that the website now successfully functions as intended and promotes self-regulated learning for beginners to develop their knowledge of the command line and their programming skills within the offered terminals.

5.2 Design

The prototyping phase ensured that the design of the website passed through a few design iterations before being finalised. This allowed feedback to be acquired and the design to be improved. Feedback obtained in the second design iteration, the prototyping stage, allowed for multiple improvements to be made. These improvements included reducing colour contrasts, using intuitive designs and features, using left-alignments instead of central-alignments and providing explanations for included images and graphics. With this in mind, an improved design was included in the final implementation for the website, pictured in Figure 4.1.

Responses regarding design from the lab study were all positive. The clear layout, the colour scheme and the professional design used were all complimented and were detailed as contributing to ease-of-use throughout the website. The visual appeal of the website is key in evaluating its success, as this encourages more users to use the website, based on what they see at an initial glance. The feedback suggests that the design has been through a positive improvement and the current website design is effective in providing the users with a better experience.

5.3 Functionality

The website functions in a way that achieves its intended objectives. Navigation around the website is done with ease and all information can be clearly accessed. Users can effectively view and use the worked examples, as well as the terminal pages. Error handling when command line programming in the terminals needs to be addressed, in order to prevent future confusion from occurring and potential misconceptions about command functionality developing. However, due to the scope of the website and the range of commands implemented, it was not possible to implement unique error messages for each command in task 1 and for any command in tasks 2 and 3, like how they might appear in the actual interfaces. Instead, a generic error message, stating that an error has occurred, and that the user should try entering the command again, has been added to the system functionality. Time spent to ensure that all features were completely implemented has allowed the additional tasks to be implemented, to ascertain that users can click on, and use all features they anticipate being able to. With all considerations, the functionality of the website has improved since the lab study, based on the provided feedback. Yet, there remains scope for the functionality to be extended to both cover a wider range of tasks, and provide more accurate error handling within the command line interfaces. Additionally, more terminals could be included for use, and potentially different types of shells, to expand the users' knowledge gained when learning on the website.

5.4 Usability

All considerable factors, which lead to the success of the tool, also can hinder or improve the usability of the tool. Usability, in this way, is the main success criteria of the tool and the project aims to create a graphical user interface, which is simplistic enough, that the

Question	Answer
Please comment on the usability of the tool.	I'd say the tool was very easy to use.
	Had no issues.
	Overall very good, the only thing that might have a negative impact is the fact that the terminal doesn't output error messages and that it only executes the commands on the list.
	Very usable, had no issues understanding what was going on or what I was doing.
	It's very straight-forward to use but one thing I would add in is a link to the home page via clicking on the logo icon and 'Command Line Programming 101' title.

Table 5.1: Overall usability feedback provided by lab study participants

target demographic does not experience problems when using it. Any problems caused would in turn contribute to the tool being deemed as less usable. When analysing the feedback provided in the lab study's questionnaires, as seen in Table 4.1, there was a range of comments suggesting that the website is completely user-friendly and beginners will be able to utilise it in the intended manner. There were a few functionality issues raised regarding error handling for the terminals, and a solution has been created for this, to avoid users being unable to use this feature due to this. No navigation problems were noted during the study and the feedback suggested that the design of the website is very intuitive, to the extent that participants were aware of how they were meant to make use of all the available features. With this in mind, the website can be described as usable and meets the intended usability requirements set out in the Introduction chapter. Future work on this project could add to the usability of the website and improve user interactions with all the features.

Chapter 6

Conclusions

6.1 Project summary

This project has intended to foster an interest in the command line for those who have had no or little previous experience with it. Regardless of prior programming experiences or educational background, the project has focused on creating a usable user interface for beginner command line programmers to use, to develop their understanding of what the command line is, what it is used for, how we can use it and the benefits to utilising command line interfaces.

The project began with an investigation into command line programming and common interfaces, existing command line programming tools and good user interface design. Following this, design ideas were formulated by using Figma, the online application used for interface design. At this stage, a prototype of the tool was generated, to help convey ideas and plan the basic functionality, layout and interactions of the website. Simultaneously, experimentation occurred with existing software and resources, which had the potential to be utilised in implementation, to evaluate which ones would be the most applicable to use.

Following this, the project went through another design iteration, in which feedback received from my supervisor was taken into consideration, and the recommended changes were applied. The next stage in the project was the implementation of the tool. Since the design and functionality of the website had already been determined before this, implementation involved using the adapted prototype and incrementally coding up the required pages and files. When the first task had been fully implemented, along with the other web pages, the website was evaluated by five participants, fitting the target demographic, in a controlled lab study. The study allowed the participants to trial the website and provide feedback on the positive aspects and critique any experienced problems, to help aid the evaluation of the system and continued implementation of the website.

Difficulties implementing bash commands requiring the use of files and directories were made apparent towards the end of the implementation phase. This led to alternate methods to bash commands implementation being investigated and evaluated to see

which one could be most easily and effectively utilised, due to the time left in the project. It was concluded that using a PHP script, to execute the commands and return the output, was the best method, due to the existing web server which had been created as a precautionary step at the beginning of the project. Using the VirtualBox VM, an Nginx server was adapted to satisfy PHP configurations and allowed for the web pages to be run client-side and the PHP scripts to run back-end. It became possible to implement the rest of the required commands and progress towards the final product continued.

Project day concluded the end of this project, and provided an opportunity to showcase the improved system to more potential users and gain further formative feedback. This encouragement experienced during the evening event helped finalise the website features and solidify the positive and creative idea behind this project.

6.2 Future work

The Command Line Programming 101 Website is easily extendable. As previously noted, there is an endless range of tasks that can be achieved with command line programming, and therefore there are an infinite number of tasks which could be included within the website. The design of the tasks page easily allows for the implementation of additional tasks, as only new task holders need to be included, with navigation to the appropriate pages. The terminal pages can be duplicated and adapted to suit the task, including a task description, relevant commands in the reference guide, and further development of the jQuery script responsible for emulating the terminals. In this way, the new commands can be supported. Similarly, a wider range of terminals could be included, such as Linux Bash Shell, to support further learning on the website, and worked examples could be provided for all commands. Future work is important in implementing accurate error handling for all the terminals included on the website. Current error handling is implemented for task 1 at a basic level, and future work on this project can ensure the development of personalised error messages for the commands.

6.3 Skills and knowledge developed throughout the project

This project was my first individual long-term project. This meant that I had to manage the project over a longer time period, whilst taking into consideration other commitments, especially courses which had more timely deadlines than this project. On reflection, the intermediate checkpoints for this project helped to keep me on track, and gain informative feedback of my project. They encouraged me to have something to show, which in turn encouraged me to keep working on my project. At times my project was neglected, to keep up with other submissions, and it became easier to prioritise those courses early on in the project. However, I can now appreciate the importance of working on a little bit of the project somewhat consistently throughout the time period, to ensure the majority of the workload is not left to be completed close to the deadline.

Like all projects, everything did not go to plan, but being able to anticipate this early on enabled me to progress well through the project, and not fall far behind schedule.

Before implementation, it was of high importance that I improved my knowledge of the command line. Prior experiences command line programming had been limited, and it was clear that I would have difficulties carrying out this project, with a limited understanding of the command line and what is made possible through coding in those interfaces. Because of this, I complete this project with an improved knowledge of the command line, and experiences programming in various interfaces. I feel more comfortable using the command line and would even consider using it over graphical user interfaces in processing tasks. In my opinion, being able to use the command line is an important skill, especially as a Computer Scientist. This importance has been made clear to me whilst completing research in this project and most definitely has solidified my interest in command line programming.

Additionally, this project enabled me to refresh my web development skills. Previous experiences in web development involved using HTML, CSS and Bootstrap to create my own personal blog. The use of TailwindCSS in this project gave me experiences using an alternative CSS framework, one which I found to have more flexibility when determining the layout of the pages. I previously had limited experiences in JavaScript. The creation of this website highly depended on the JavaScript code to emulate the terminals. I learnt the JavaScript syntax and made mini scripts to help develop my understanding, making me feel at ease using the scripting language, before beginning to implement the terminals. Without this practice, I would not have felt comfortable using the jQuery framework, to support the terminal scripts I had written. Using these languages has provided me with an improved overview of web development, and what scripts are incorporated into websites. The Figma prototype helped me in designing the website and its interactions, which proved very useful when implementing, as I had a clear insight into what needed to be implemented, linked and made interactive. It is not very often that I find myself working with new software and frameworks, as I find comfort using familiar languages and modules. In doing so, I have refreshed the skills used when learning new programming languages and concepts, skills which will prove essential for me as I progress in my future endeavours in the Computer Science industry.

6.4 Reflections

I am proud of the work I have achieved throughout this project. Most significantly, I have enjoyed bringing my vision to life, taking Dr Brian Mitchell's proposed project and making it my own. Designing the website and formulating the ideas behind it have made use of my creative side, alongside the feedback from my supervisor, invoking the website's visual appeal. Whilst the hurdles faced during implementation were, at the time, a nuisance and difficult to overcome, the satisfaction of viewing and using the completed website has made it worthwhile.

If I could do this project differently, I would have run the pre-study questionnaire earlier than I did. The information gathered about the participants' expectations from learning

tools and the things they would like to know about command line programming proved useful in supporting my design decisions. However, this came at a point where the tool had mostly been completed. If the information was collected earlier, this could have helped to shape the website more, and perhaps I could have implemented more features that were popular amongst favoured learning websites, adding to the overall appeal of the website. Whilst prototyping, I was also experimenting with implementing commands in a trial jQuery terminal. This helped me become familiar with the framework and established the behaviour of a few commands early on. However, if I could have done something differently, knowing what I know now, I would have tried implementing some of the file handling commands, as these were the ones I had difficulty doing. This would have alerted me to the need for PHP scripts and the web server, to process these commands, and would have prevented this issue troubling me for a long time and halting progression of the project later on.

Without those changes, the project was still a success and this was solidified by the comments made by students on the Project Day. This naturally made me proud of the work I had produced, and it delighted me to have so many pupils interested in the website. They even asked if it was currently available for them to use, as they conveyed their want for a tool like this to exist. This highlights success in achieving what I set out to with this project - creating a tool to teach beginners about command line programming and encourage them to use command line interfaces.

References

Berry, Vincent, Castelltort, Arnaud, Pelissier, Chrysta, Rousseau, Marion Rousseau, and Tibermacine, Chouki, 2022. Shellonyou: learning by doing unix command line [Online]. Available from: https://www.lirmm.fr/~tibermacin/papers/2022/VBetAl_ITICSE_2022.pdf [Accessed November 22, 2022].

Bevan, Nigel, Carter, James, and Harker, Susan, 2015. Iso 9241-11 revised: what have we learnt about usability since 1998? *Human-computer interaction: design and evaluation: 17th international conference, hci international 2015, los angeles, ca, usa, august 2-7, 2015, proceedings, part i 17*. Springer, pp.143–151.

Bissyandé, Tegawendé F, Thung, Ferdian, Lo, David, Jiang, Lingxiao, and Réveillere, Laurent, 2013. Popularity, interoperability, and impact of programming languages in 100,000 open source projects. *2013 ieee 37th annual computer software and applications conference*. IEEE, pp.303–312.

Boucheron, Brian, 2020. *Initial server setup with ubuntu 20.04* [Online]. Digital Oceans. Available from: https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-20-04 [Accessed November 20, 2022].

Brooke, John, 2013. Sus: a retrospective. *Journal of usability studies* [Online], 8(2), pp.29–40. Available from: https://www.researchgate.net/profile/John-Brooke-6/publication/285811057_SUS_a_retrospective/links/5ee5c4a792851ce9e7e38a75/SUS-a-retrospective.pdf.

Budiu, Raluca, 2014. *Memory recognition and recall in user interfaces* [Online]. Available from: https://www.nngroup.com/articles/recognition-and-recall/ [Accessed March 8, 2023].

Cocca, Germán, 2022. *Command line for beginners – how to use the terminal like a pro [full handbook]* [Online]. FreeCodeCamp. Available from: https://www.freecodecamp.org/news/command-line-for-beginners/ [Accessed October 17, 2022].

Codecademy, n.d. *Learn the command line* [Online]. Available from: https://www.codecademy.com/learn/learn-the-command-line [Accessed October 12, 2022].

Gangwar, Meghna, 2022. *How to configure php-fpm with nginx* [Online]. Digital Oceans. Available from: https://www.digitalocean.com/community/tutorials/php-fpm-nginx [Accessed March 22, 2023].

Garnett, Alex, 2022. *Cloud servers: an introduction* [Online]. Digital Oceans. Available from: https://www.digitalocean.com/community/conceptual-articles/cloud-servers-an-introduction [Accessed January 19, 2023].

Glass, Erin, 2020. *How to install nginx on ubuntu 20.04* [Online]. Digital Oceans. Available from: https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04 [Accessed November 20, 2022].

Hoyer, Wayne D and Brown, Steven P, 1990. Recognition over recall. *Journal of consumer research*, 17, pp.141–148.

Jankiewicz, Jakub T, n.d. *Jquery terminal: javascript web based terminal emulator* [Online]. Available from: https://terminal.jcubic.pl/.

Kiili, Juha, 2022. *What every data scientist should know about the command line* [Online]. Available from: https://valohai.com/blog/command-line-for-data-science/ [Accessed February 13, 2023].

MDN Web Docs, n.d. *Command line crash course* [Online]. Available from: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Understanding_client-side_tools/Command_line [Accessed October 12, 2022].

Milligan, Ian and Baker, James, 2014. *Introduction to the bash command line*. The Editorial Board of the Programming Historian.

Nielsen, Jakob, 2020. *10 usability heuristics for user interface design* [Online]. Available from: https://www.nngroup.com/articles/ten-usability-heuristics/ [Accessed March 8, 2023].

No Starch Press, n.d. *The linux command line, 2nd edition* [Online]. Available from: https://nostarch.com/tlcl2#reviews [Accessed April 5, 2023].

Nobel, Peter A and Shiffrin, Richard M, 2001. Retrieval processes in recognition and cued recall. *Journal of experimental psychology: learning, memory, and cognition* [Online], 27(2), pp.384–413. Available from: https://doi.org/10.1037//0278-7393.27.2.384.

ReQtest, 2020. *What are the benefits of version control?* [Online]. ReQtest. Available from: https://reqtest.com/requirements-blog/what-are-benefits-of-version-control/ [Accessed March 8, 2023].

Sampath, Harini, Merrick, Alice, and Macvean, Andrew, 2021. Accessibility of command line interfaces. *Proceedings of the 2021 chi conference on human factors in computing systems* [Online], pp.1–10. Available from: https://doi.org/10.1145/3411764.3445544.

Shotts, William, 2019. *The linux command line: a complete introduction*. No Starch Press.

Stack OverFlow, 2022. *2022 developer survey* [Online]. Available from: https://survey.stackoverflow.co/2022/ [Accessed January 10, 2023].

Tailblocks - ready-to-use tailwind css blocks [Online], n.d. tailblocks.cc. Available from: https://tailblocks.cc/.

Tailwind css - rapidly build modern websites without ever leaving your html. [Online], 2017. tailwindcss.com. Available from: https://tailwindcss.com/.

The Website Architect, 2020. *Does text alignment matter for accessibility and usability?* [Online]. Available from: https://thewebsitearchitect.com/does-center-aligned-text-matter-for-accessibility/ [Accessed December 19, 2022].

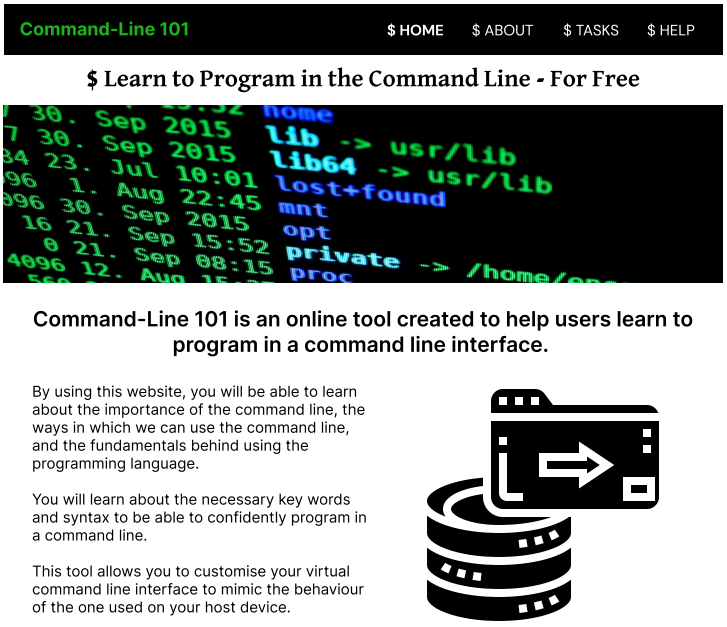
Webstorm: the smartest javascript ide by jetbrains [Online], n.d. JetBrains. Available from: https://www.jetbrains.com/webstorm/.

Xterm.js [Online], 2022. xterm.js. Available from: http://xtermjs.org/.

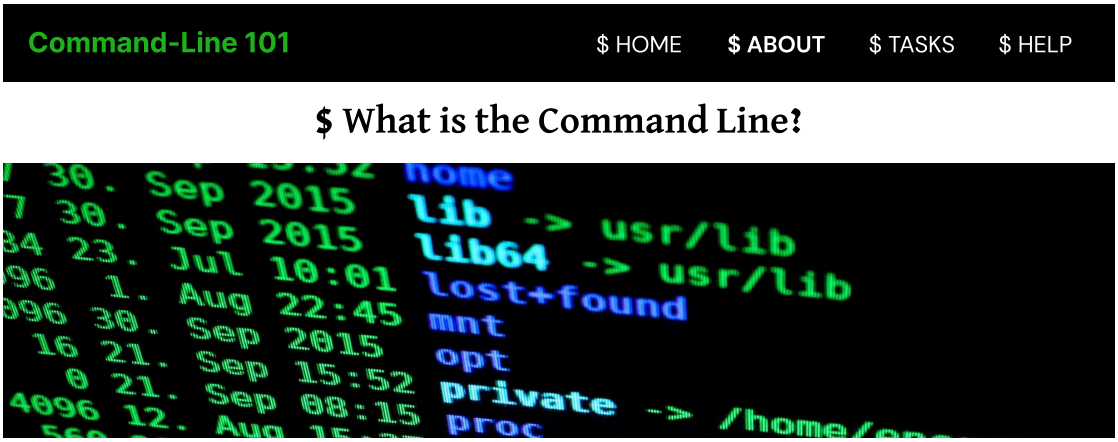
Appendix A

Figma prototype

A.1 Prototype Home Page

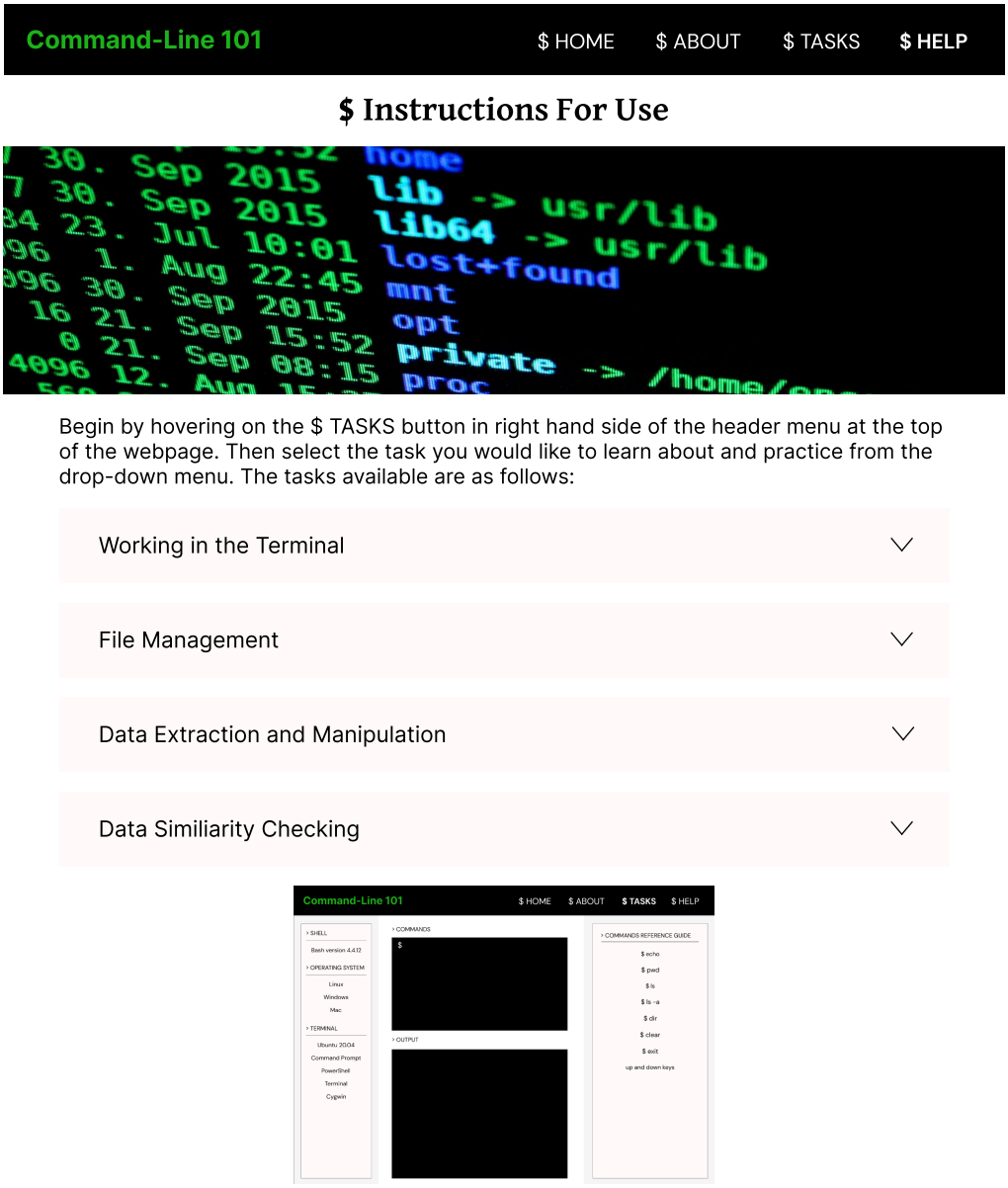


A.2 Prototype Introduction Page

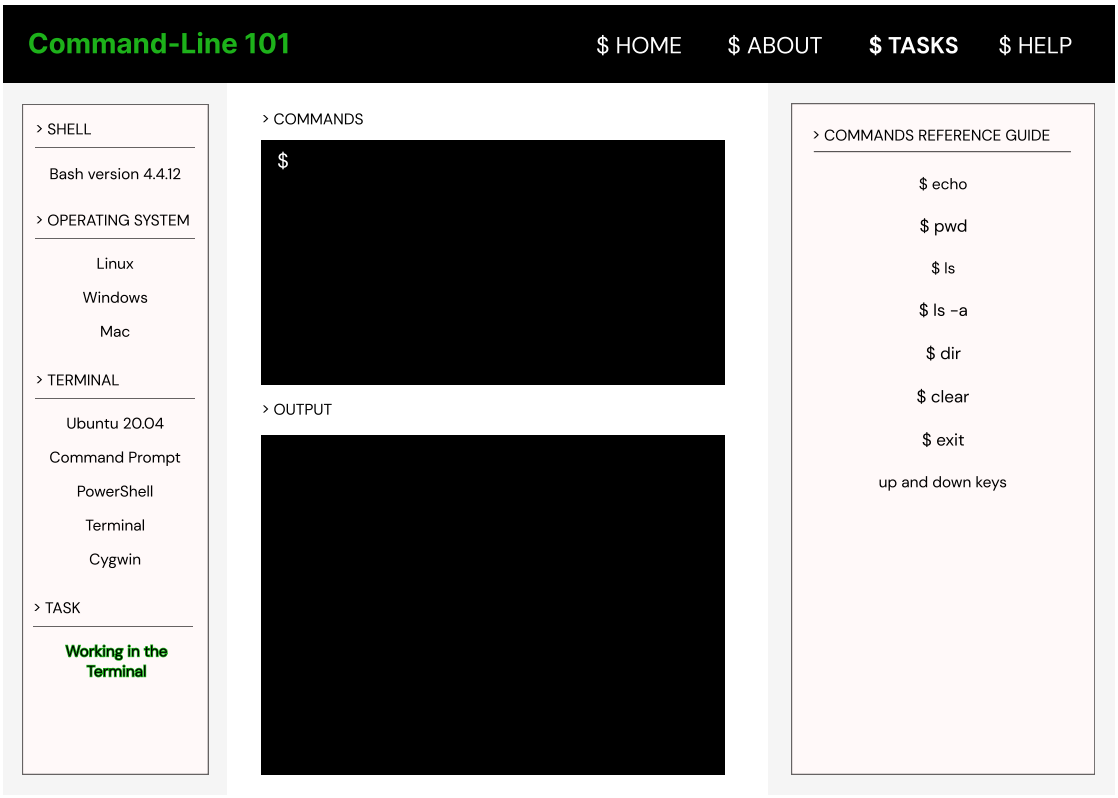


The Command Line is

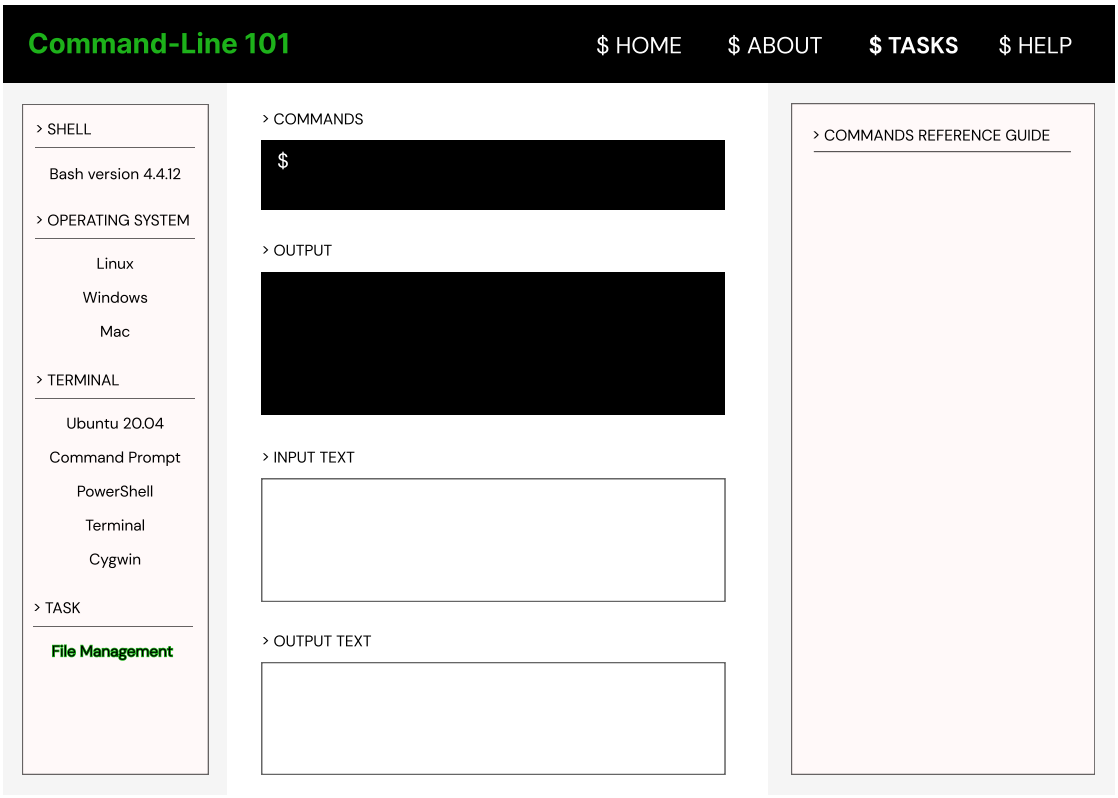
A.3 Prototype Help Page



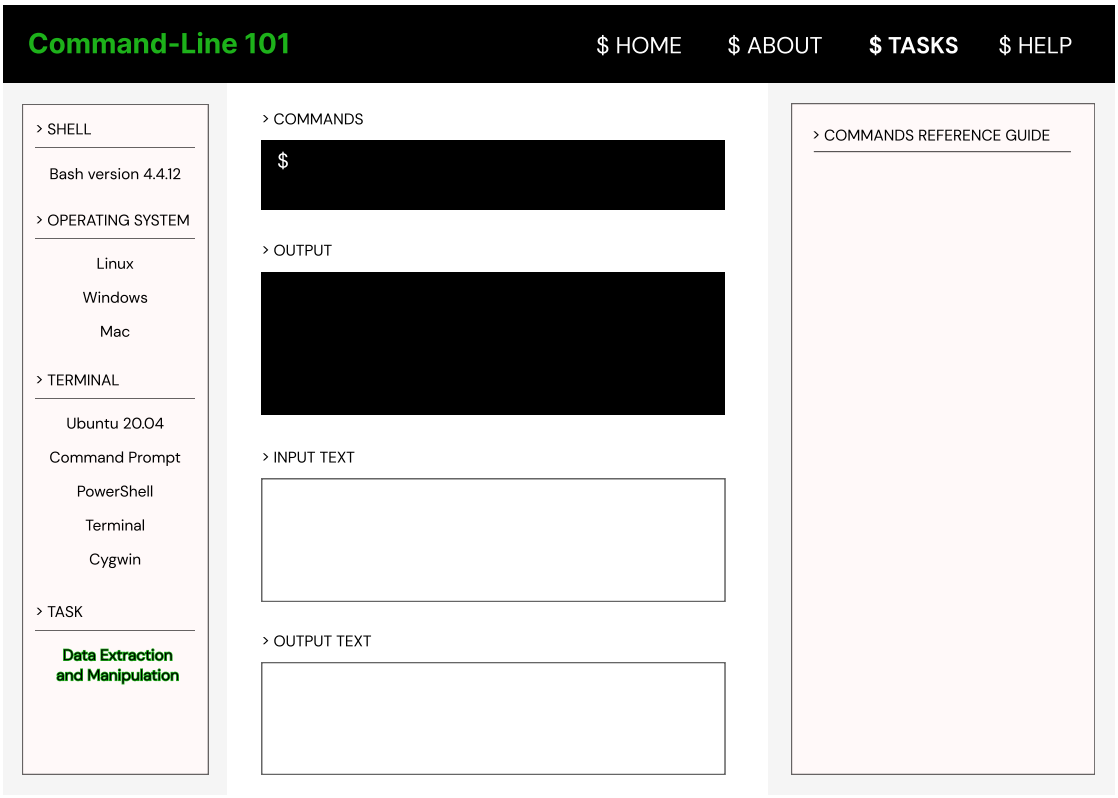
A.4 Prototype Terminal Pages



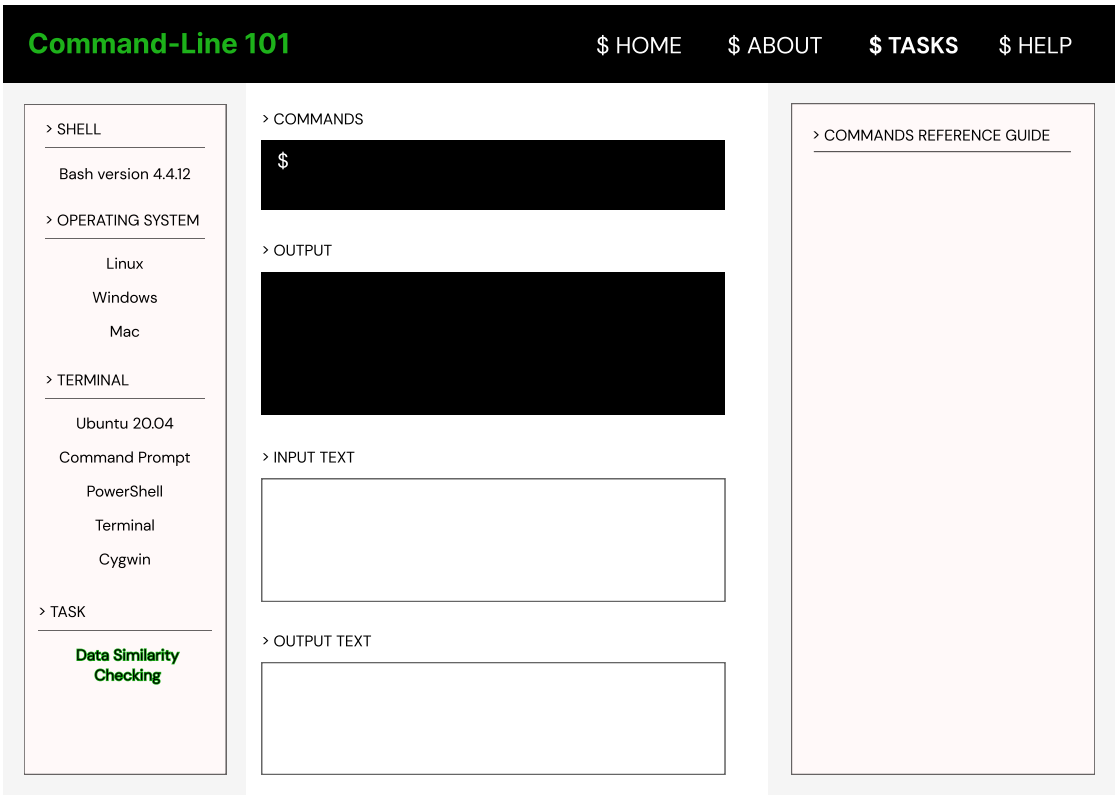
A.5 Prototype Terminal Pages



A.6 Prototype Terminal Pages



A.7 Prototype Terminal Pages



Appendix B

Participants' online information sheet and consent form

Participant Information Sheet

Project title:	Tool to visualise command-line programming
Principal investigator:	Brian Mitchell
Researcher collecting data:	Kristin Agyekumhene

This study was certified according to the Informatics Research Ethics Process, RT number 548020. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

The principal investigator for this project is Brian Mitchell, a university teacher at the School of Informatics. The researcher collecting data for this project is Kristin Agyekumhene, a fourth-year undergraduate student studying BEng Software Engineering at the School of Informatics.

What is the purpose of the study?

The purpose of this study is to find out about the attitudes of participants towards the command-line. This ranges from attitudes to its appearance, to using it to write lines of code. The study also aims to find out what participants think the command-line is used for and how they would make use of it. Participants will be asked about their opinions on command-line learning resources that are currently available and asked about the ways they would improve them.

Why have I been asked to take part?

You have been asked to take part in this study as you fit the research target demographic of those who have no/little experience in programming using command-line interfaces, irrespective to other programming experiences.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, without giving a reason. Your rights will not be affected. If you wish to withdraw, contact the PI. We will stop using your data in any publications or

presentations submitted after you have withdrawn consent. However, we will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

If you decide to take part in this study, you will be asked to fill out an online form answering a range of questions regarding your attitudes towards and usage of the command-line. The form should take no longer than 10 minutes to complete.

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

There are no additional benefits associated with taking part.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a minimum of two years.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team – Brian Mitchell and Kristin Agyekumhene.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.
For general information about how we use your data, go to: edin.ac/privacy-research

Who can I contact?

If you have any further questions about the study, please contact the lead researcher Kristin Agyekumhene - s1913250@ed.ac.uk.
If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on http://web.inf.ed.ac.uk/infweb/research/study-updates.

Consent

By proceeding with the study, I agree to all of the following statements:

- I have read and understood the above information.
- I understand that my participation is voluntary, and I can withdraw at any time.
- I consent to my anonymised data being used in academic publications and presentations.
- I allow my data to be used in future ethically approved research.

[Button here named "I agree" or "take me to the survey"]

Appendix C

Sign-up sheet

User Study Sign-up Sheet

This form is to say you might want to take part in one (or more) user studies to help final year students develop tools to help students. These tools might be useful to you in your future studies.

All UoE students are welcome to participate in the user studies. You are equally valuable whether you are new to Computer Science or experienced. Indeed the greater the diversity, the better. This includes native and non-native speakers of English, and people with accessibility requirements or impairments, especially any affecting the way you use computers.

User studies might include being asked to test some software, to say what you'd like the software to do or how it should do it, or to tell your experience in the topic, if any. No specific experience is necessary to participate in the studies: everyone from "beginner" to "experienced" is equally welcome.

The format of user studies may differ per project. You are likely to be asked to participate in 2 or 3 sessions per project you express an interest in.

Submitting this form does **not commit you** to definitely participating and you may withdraw later.

Any information you do submit is confidential and will be deleted once sign-ups have been processed.

Disclaimer: the form's signature image may or may not be indicative of what happens during any user study. Any similarity between the events depicted in the signature image and he user study may or may not be entirely coincidental.

* This form will record your name, please fill your name.

Eligibility check

1. I am a current University of Edinburgh student *

- ◯ Yes
- ◯ No

Sorry: these user studies are only for current University of Edinburgh students

Details

2. The following projects are available for you to participate in. The students doing the projects want to hear from beginners and novices as well as people with relevant experience.

    1. Tool to help **manage timetables and workloads** — this tool is for everyone but anyone with **accessibility requirements** is especially encouraged to participate
    2. Tool to help you **check your assignment complies** with submission guidelines — this might save you some stress and maybe a few marks
    3. Tool to help you **write "proper" emails to university staff** — this is as much for native English speakers as it is for non-native English speakers
    4. Tool to help you **learn command-line programming** — this type of programming helps with task automation and processing lots of files
    5. Tool to help **turn program code into** natural language (for example English, Spanish) — this is as much for native English speakers as it is for non-native English speakers, and for experienced programmers as it is for beginners
    6. Tool to help **managing data and experiments** in Machine Learning — you don't have to any experience with Machine Learning

Are you interested in each project? *

	I don't know	Not interested	Interested
Managing timetables and workloads	◯	◯	◯
Checking assignment compliance	◯	◯	◯
Managing Machine Learning data and experiments	◯	◯	◯
Learning command-line programming	◯	◯	◯
Turning program code into	◯	◯	◯

3. Please specify the **one or two** projects you are **most** interested in testing. *

Please select at most 2 options.

- ☐ Managing timetables and workloads
- ☐ Checking assignment compliance
- ☐ Writing emails to university staff
- ☐ Learning command-line programming
- ☐ Turning program code into natural language
- ☐ Managing Machine Learning data and experiments

About you

These questions are here so we can see the range of diversity.

We respect your answers and we respect your right not to answer.

4. Your Student ID (without the S) — also known as your UUN *

5. Are you a native speaker of English or has English been the main language you have used in studying? *

- ◯ Do not wish to say
- ◯ No
- ◯ Yes

6. Do you have any kind of accessibility or educational requirements that affect the way you use a computer?

   We will not ask you for any details. You may volunteer this information to the project student(s) at a later date if and only if you want to. *

- ◯ Do not wish to say
- ◯ Yes but I do not like to admit I do
- ◯ Yes and I openly admit I do, or it is impossible to disguise my situation
- ◯ No

7. If you have any questions or comments please put them here.

   Please make it clear which project each question or comment is intended for

intended for.

Confirmation

8. Expressions of interest will be processed in batches so there might easily be a delay of several weeks before you receive a response *

- ◯ I understand there might be several weeks before there is a response. By submitting I agree to be contacted via my University email by the student(s) whose project(s) I have expressed an interest in.

Appendix D

Lab study survey questions

D.1 Pre-study survey questions

1. I am a student, staff member or faculty member at the University of Edinburgh.
2. Please select your degree level:
3. If you selected Undergraduate Degree, please select your year of study:
4. Please state your degree title:
5. Please detail your experiences with Computer Science prior to starting your degree: (type 'N/A' if you have had no experiences)
6. Please detail your experiences with Computer Science during your degree: (type 'N/A' if you have had no experiences)
7. What type of programmer would you classify yourself as?
8. If you have had programming experiences, please detail your preferred method of learning a new programming language.
9. Prior to this study. had you heard of the command line?
10. Prior to this study, had you had any experiences programming in the command line?
11. If you selected Yes, please select the command line interfaces you have used.
12. Please detail what you know about the command line and any tasks you may have used it for. (type 'N/A' if none)
13. Please detail what you would like to know about the command line that you do not know already.
14. Please detail what tasks you would like to use the command line for, which you do not currently know how to do.
15. Please detail your interest in the command line and/or this study.
16. Do you frequently use online tools to aid your learning?
17. Which features of online learning tools do you like? (type 'N/A' if none)
18. Which features of online learning tools do you dislike? (type 'N/A' if none)
19. What would you expect from an online learning tool?
20. Please provide any examples of effective online learning tools you have used.

D.2 Post-study survey questions

1. Please confirm which webpages you visited.
2. What did you like about the home page?
3. What did you not like about the home page?
4. Do you think the home page functions as it should?
5. If you selected ‘no’, why?
6. Do you like the visual design of the home page?
7. Why?
8. What do you think of the usability of the home page?
9. How would you rate the home page overall?
10. What did you like about the introduction page?
11. What did you not like about the introduction page?
12. Was the information sufficient?
13. If you selected ‘no’, what additional information would you like to be included?
14. Do you like the visual design of the introduction page?
15. Why?
16. How would you rate the introduction page overall?
17. What did you like about the help page?
18. What did you not like about the help page?
19. Was the information sufficient, clear and helpful?
20. Why?
21. Do you like the visual design of the help page?
22. Why?
23. How would you rate the help page overall?
24. Do you think the range of tasks offered is sufficient?
25. If you selected ‘no’, which additional tasks would you like to feature?
26. What did you like about the tasks page?
27. What did you not like about the tasks page?
28. How would you rate the tasks page overall?
29. Which terminal(s) did you use?
30. Do you think the range of terminals offered is sufficient?
31. If you selected ‘no’, which additional terminals would you like to feature?
32. What did you like about the terminal page?
33. What did you not like about the terminal page?
34. Comment on the usability of the terminal page and subpages.
35. Comment on the functionality of the chosen terminal. Did it function as expected/desired?
36. Was the information found in the command reference guides sufficient, clear and helpful?
37. Why?
38. Comment on the worked examples. Did you find them useful?
39. Comment on your overall experience using the terminal pages. Do you think they could be beneficial to learners/beginners?
40. How would you rate the terminal page overall?
41. Please comment on your overall experience using the tool.

42. Please comment on the usability of the tool.
43. Do you think this tool achieves what it sets out to do?
44. Why?
45. Would you use this tool to learn how to command line program?
46. Would you recommend this tool to others?
47. Please rate the tool overall.

Appendix E

Pre-study survey results

Question	Answer
I am a student, staff member or faculty member at the University of Edinburgh.	I Agree.
	I Agree.
	I Agree.
	I Agree.
	I Agree.

Table E.1: Collected responses on participants' level of education

Question	**Answer**
Please select your degree level.	Undergraduate Degree.
	Undergraduate Degree.
	Undergraduate Degree.
	Undergraduate Degree.
	Undergraduate Degree.
If you selected Undergraduate Degree, please select your year of study.	Year 4.
	Year 1.
	Year 4.
	Year 4.
	Year 4.
Please state your degree title.	Computer Science and Maths.
	BSc Computer Science.
	Artificial Intelligence and Computer Science.
	Computer Science.
	MInf Informatics.

Table E.2: Collected responses on participants' level of education

Question	**Answer**
Please detail your experiences with Computer Science prior to starting your degree: (type 'N/A' if you have had no experiences).	Studied up to Advanced Higher Computing in School, programming in Visual Basic.
	Did some programming in Python and C#.
	I knew a bit of python and I also did Computer Science Higher Level at IB.
	SQA Higher Qualification from school.
	Had taken Computer Science at school up to advanced higher.
Please detail your experiences with Computer Science during your degree: (type 'N/A' if you have had no experiences).	Programming in Python and Java, Machine Learning, Data Science, Computer Networks.
	I did 'Introduction to Computation' and am halfway through 'Object Oriented Programming'
	A lot of experience, multiple programming languages, multiple operating systems, familiar with command line.
	Studied range of modules from topics like artificial intelligence to computer security to human computer interaction, worked with a range of programming language such as Python Java etc. and less common ones such as Haskell.
	Have knowledge of various computing concepts and programming languages.

Table E.3: Collected responses on participants' Computer Science experiences

Question	**Answer**
What type of programmer would you classify yourself as?	Intermediate - moderate programming experience.
	Intermediate - moderate programming experience.
	Intermediate - moderate programming experience.
	Confident/Expert - lots of programming experience.
	Confident/Expert - lots of programming experience.
If you have had programming experiences, please detail your preferred method of learning a new programming language.	Completing assigned lab exercises/assignment, following a free online course or watching YouTube tutorials.
	1) Spend around 4 hours learning the syntax of the language through YouTube tutorials and small tasks 2) Try to create some application which would somewhat represent why you're learning it. (e.g. when learning Java Spring Boot creating a full-stack web application).
	Following tutorials on YouTube or coding websites where they take you step by step.
	Usually read up on it a wee bit and do a couple of tutorials then dive straight in and google things as I go along.
	Learn by trial and error along with using things like W3 Schools.

Table E.4: Collected responses on participants' programming experiences

Question	**Answer**
Prior to this study, had you heard of the command line?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
Prior to this study, had you had any experiences programming in the command line?	Yes.
	Yes.
	No.
	Yes.
	Yes.
If you selected Yes, please select the command line interfaces you have used.	Ubuntu;Linux Bash Shell;Windows Command Prompt;PowerShell;Mac Terminal;
	Linux Bash Shell;Windows Command Prompt;
	Ubuntu;Windows Command Prompt;PowerShell;
	Mac Terminal;Ubuntu;Linux Bash Shell;Windows Command Prompt;PowerShell;
	Mac Terminal;Windows Command Prompt;

Table E.5: Collected responses on participants' exposure to the command line

Question	Answer
Please detail what you know about the command line and any tasks you may have used it for. (type 'N/A' if none)	To code, or run/execute programs, to manage files e.g. create/delete etc.
	I know the very basic commands. I use Linux mint cinnamon as my primary OS so I sometimes use the terminal for running scripts I found on the internet, doing some basic operations, managing python virtual environments or using git.
	I have used it for basic commands, like navigating through directories, creating files, or compiling programs and running executables. I also use git commands on the command line.
	Used it for running code mostly but also know you can access files etc. with it.
	Making/deleting/zipping/opening files for submissions.
Please detail what you would like to know about the command line that you do not know already.	How to do more "in less commands" so to speak, i.e. be more efficient in using the command line.
	How to use it to perform my daily tasks more efficiently, maybe learn some scripting. Ultimately I would love to be able to use a Linux system without a graphics user interface.
	I have heard about bash files that you sometimes need to set up before running certain commands on the command line, but I don't really know what those are or how to make them, so that would be nice to know.
	Full range of things it can be used for.
Please detail what tasks you would like to use the command line for, which you do not currently know how to do	How to commit files to GitHub.
	Not sure, I would have learned them myself if I knew.
	Nothing really, I usually prefer using the GUI.
	Restart my computer.

Table E.6: Collected responses on participants' command line knowledge and experiences

Question	**Answer**
Please detail your interest in the command line and/or this study.	I'm not very confident with using the command line - and avoid using it because of this, would be nice to learn how to use it better.
	I like how it enables you to do stuff more efficiently than the GUI and enables some stuff that you can't. Also I feel like CS students SHOULD be able to use the command line effectively.
	I want to see if this tool will teach me stuff about the command line I didn't already know.
	Not heavily experienced in using the command line so thought it would be interesting to learn a bit more.
	I'm interested to see how this tool has been created and works.

Table E.7: Collected responses on participants' command line and lab study interest

Question	**Answer**
Do you frequently use online tools to aid your learning?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.

Table E.8: Collected responses on participants' usage of online learning tools

Question	**Answer**
Which features of online learning tools do you like? (type 'N/A' if none).	An interactive UI, fun game-style element.
	I can pick and choose the ones I like/find effective, I can pick the course material I need.
	That you can take it at your own pace, that they take you step by step and provide clear examples at every step of the way.
	Interactive features, e.g. W3 schools has a feature where you can alter, run and test provided code.
	The availability of them - can use them wherever you study.
Which features of online learning tools do you dislike? (type 'N/A' if none).	Long series are hard to follow, sometimes they have unnecessary stuff, you don't know if the person who's teaching you really is competent.
	That if I am confused about something I can't ask someone for help and I have to use google or a second tool to answer my question.
	When there's too much writing, usually switch off and use another one instead.
What would you expect from an online learning tool?	For it to be free and easily accessible.
	Flexibility for what you want to learn. You can learn what you need for the task you're trying to solve.
	That you can take it at your own pace, that they take you step by step and provide clear examples at every step of the way.
	To gain some skills and knowledge from using it and it to be engaging.
	Reliability and availability.
Please provide any examples of effective online learning tools you have used.	I usually just use YouTube tutorials.
	W3 schools.
	Udemy, W3Schools.

Table E.9: Collected responses on participants' expectations of learning tools

Appendix F

Post-study survey results

Question	Answer
Please confirm which web pages you visited.	Introduction;Home;Help;Tasks;Working in the Terminal;
	Home;Tasks;Working in the Terminal;Introduction;
	Home;Introduction;Help;Tasks;Working in the Terminal;
	Home;Introduction;Tasks;Help;Working in the Terminal;
	Home;Introduction;Help;Tasks;Working in the Terminal;

Table F.1: Web pages trialled by participants during the lab study

Question	**Answer**
What did you like about the home page?	It had a nice layout, I liked the colours.
	Simple and clean, aesthetically pleasing.
	The layout, the fonts and the appearance in general looked very professional, the options menu was clearly visible and it was very easy to understand how to navigate to each page and how to start working on the tasks.
	Not too cluttered, tells the user exactly what they need to know, the title is good - big and capitalised means the user knows exactly what they are getting.
	I really liked the layout, especially seeing the supported OS systems. The bold title was a nice design element as well.
What did you not like about the home page?	The fact that you can go to the tasks and introduction pages both from the options menu and with the green buttons slightly threw me off at first because I thought the buttons were supposed to redirect to different pages, but I don't think that is a very big issue.
	Although good to have the title emphasized would be good if it was maybe slightly smaller so you could see the get started and read more buttons without having to scroll down. (nit picking though is really good!).

Table F.2: Participants' opinions on the home page

Question	**Answer**
Do you think the home page functions as it should?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
Do you like the visual design of the home page?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
Why?	The colours are nice, the way everything is spaced out is good too.
	Not overly complicated, gets to the point, not too many colours and they fit together well. Also not overwhelmed by choices.
	It doesn't contain any unnecessary information, it only contains what you need and nothing more, in a very comprehensible and intuitive manner.
	Looks very professional so I would trust it as website to use. Not cluttered so doesn't overwhelm the user but engages them.
	I liked the overall layout.
What do you think of the usability of the home page?	I think it was easy to use.
	Its great.
	excellent.
	Very usable, had no issues.
	It is straightforward to use and looks well thought-out.

Table F.3: Participants' comments on the functionality and usability of the home page

Question	**Answer**
How would you rate the home page overall (out of 5)?	5.
	5.
	5.
	5.
	5.

Table F.4: Home page ratings

Question	**Answer**
What did you like about the introduction page?	Its simple and gets to the point.
	The title is very informative of the content, so the user knows exactly what information they're gonna find there, and they can immediately decide if they need to keep reading or not.
	Great to get an overall concise and comprehensive description of what command line is.
	Well laid out and informative.
What did you not like about the introduction page?	Could add a little bit about the project itself. Also how you will be learning the command line.
	My issues with the introduction page were mostly about the aesthetic, for example, I didn't like the fact that the subtitle "overview" was simply underlined like that, and I would have preferred the text to be split into paragraphs instead of one big block. Also bullet points might be a better option.
	"The command line is the interface in which we enter commands for the computer to process. In simpler terms, it is the space in which you enter the commands the computer will process." The simpler terms kind of uses the same terms, didn't feel like it added too much.

Table F.5: Participants' opinions on the introduction page

Question	**Answer**
Was the information sufficient?	Yes.
	No.
	No.
	Yes.
	Yes.
If you selected 'no', what additional information would you like to be included?	missing information on the website itself.
	I'm thinking that maybe a complete beginner who doesn't know anything about the command line might not really understand what it is. Maybe give a few more examples of specific tasks and how they would be completed with the CL and with the GUI.
Do you like the visual design of the introduction page?	Yes.
	Yes.
	No.
	Yes.
	Yes.
Why ?	Simple and efficient (like the command line).
	My issues with the introduction page were mostly about the aesthetic, for example, I didn't like the fact that the subtitle "overview" was simply underlined like that, and I would have preferred the text to be split into paragraphs instead of one big block. Also bullet points might be a better option.
	Was clear, concise and not overwhelming.
	It flows well.

Table F.6: Participants' comments on the functionality and usability of the home page

Question	**Answer**
How would you rate the introduction page overall (out of 5)?	5.
	4.
	3.
	5.
	5.

Table F.7: Introduction page ratings

Question	**Answer**
What did you like about the help page?	It gave clear instructions on how to use the website.
	Was very informative, again clear and concise - not overwhelming.
	It has all the information needed and is laid out well.
Was the information sufficient, clear and helpful?	Yes.
	Yes.
	Yes.
	Yes.
Why?	All the steps were detailed very clearly, I knew exactly what each task was and how I was meant to complete it.
	Lots of white space breaking up the page.
Do you like the visual design of the help page?	Yes.
	Yes.
	Yes.
	Yes.
Why ?	I like how the explanation for each task is in the same format as on the tasks page.
	Very clear and concise.
	It flows well.
How would you rate the help page overall (out of 5)?	5.
	5.
	5.
	5.

Table F.8: Collected responses on help page

Question	**Answer**
Do you think the range of tasks offered is sufficient?	Yes.
	No.
	Yes.
	Yes.
	Yes.
If you selected 'no', which additional tasks would you like to feature?	To be fair there is so much stuff one can do in the command line the range of tasks will never be 'sufficient'. You could add some tasks specifically aimed at Linux users, like package management or installing software.
What did you like about the tasks page?	Simple, not overwhelming.
	Easy to understand, nice range of tasks.
	I liked the way the different tasks were set out.
What did you not like about the tasks page?	Should probably not use the word 'task' when there aren't tasks for the user to complete. Also should probably make it clear that you should start from the left.
How would you rate the tasks page overall (out of 5)?	5.
	4.
	5.
	5.
	5.

Table F.9: Collected responses on tasks page

Question	**Answer**
Which terminal(s) did you use?	Windows Command Prompt;Cygwin;PowerShell;
	Cygwin;
	Windows Command Prompt;PowerShell;Cygwin;
	PowerShell;Cygwin;Windows Command Prompt;
	Windows Command Prompt;
Do you think the range of terminals offered is sufficient?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
What did you like about the terminal page?	The terminal interface looked exactly like how it looks in real life, so it's more beneficial for practicing the commands for the real system. I also liked that there was a description of each command and what it does
	Was very easy to use, good that the description of the commands is viewed when you click them rather than being displayed with the commands (less clutter).
	It was easy to switch between command options to get greater information on what they did
What did you not like about the terminal page?	Probably should add some information about the terminals themselves. A beginner might have no clue which one to pick.
	I didn't like that if I entered a command that was not in the list it would just ignore it and not execute it. If a command is correct it should still execute it, cause otherwise users will get confused. Also if a command is not syntactically correct it should throw the errors that a real terminal would throw.
	Would be great to have a description of what each terminal is, where you can find it, why you would use it.

Table F.10: Participants' opinions on the terminal pages

Question	**Answer**
Comment on the usability of the terminal page and sub pages.	I like how the explanation for each task is in the same format as on the tasks page.
	Very easy to use, complex topic but seems simple!
	I think it has good usability, it is well structured and concise.
Comment on the functionality of the chosen terminal. Did it function as expected/desired?	Yes.
	Yes, no issues.
	Yes it did.
Was the information found in the command reference guides sufficient, clear and helpful?	Yes.
	Partially.
	Yes.
	Yes.
	Yes.
Why?	Could add information on parameters of commands.
	Provided all information necessary.
	All information given was helpful.
Comment on the worked examples. Did you find them useful?	Yes.
	Yes.
	Yes.
	Yes.

Table F.11: Participants' comments on the functionality and usability of the terminal pages

Question	**Answer**
Comment on your overall experience using the terminal pages. Do you think they could be beneficial to learners/beginners?	Yes, the tool is really simple and easy to use so I think it's very beginner-friendly.
	Yes. It is allows one to try out the commands for themselves instead of just having someone explain them.
	Once it is fully implemented I believe it will be very useful.
	Yes definitely, would have loved something like this when I was first starting out using command line.
	Yes, definitely!
How would you rate the terminal page overall (out of 5)?	5.
	4.
	5.
	5.
	5.

Table F.12: Terminal pages feedback and ratings

Question	**Answer**
Please comment on your overall experience using the tool.	It was good.
	Pretty good, simple to use. Though was a little confusing regarding what I should be doing.
	It is a very helpful tool, the usability is overall very good, and everything is explained very comprehensibly. I think even a beginner would be comfortable using it.
	Great! Thought it really would help beginners.
	I really liked using the tool, it was very consistent and well thought out.
Please comment on the usability of the tool.	I'd say the tool was very easy to use
	Had no issues.
	overall very good, the only thing that might have a negative impact is the fact that the terminal doesn't output error messages and that it only executes the commands on the list.
	Very usable, had no issues understanding what was going on or what I was doing.
	It's very straight-forward to use but one thing I would add in is a link to the home page via clicking on the logo icon and 'Command Line Programming 101' title.

Table F.13: Participants' experience and opinions on the usability of the website

Question	**Answer**
Do you think this tool achieves what it sets out to do?	Yes.
	Maybe.
	Yes.
	Yes.
	Yes.
Why?	Because I feel like it would be easy to use the tool to get more familiar with using the command line.
	Could add some sense of direction to what one should be doing.
	The descriptions about the commands are very informative and teach you exactly how to use them.
	Teaches to use command line.
	It is a good learning tool for command line programming and i can imagine it would help beginners.

Table F.14: Participants' evaluation of the website's success in achieving it's intentions

Question	**Answer**
Would you use this tool to learn how to command-line program?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
Would you recommend this tool to others?	Yes.
	Yes.
	Yes.
	Yes.
	Yes.
Please rate the tool overall (out of 5).	5.
	4.
	4.
	5.
	5.

Table F.15: Overall website ratings