
Robust Tutoring for Smooth Web Development

# Design Document

Zachary Cox
Olivia Croteau
Emmett Morgan
Callum Smathers

# Overview

.mpl is a website intended to provide in-depth tutorials and reference materials on the use of UNIX commands on the command-line. There are many other sites looking to achieve this same end through explanations, interactive environments, visual representations, and quick reference guides. However, it is rare that one website combines all of these elements together. As users of such sites, the developers of .mpl understand everyone is approaching their learning of UNIX commands from different experience levels, with different learning styles, and different goals. .mpl's tutorial and reference features are designed to appeal to this variety of users and serve this variety of purposes by listing the same types of information in different formats/locations and including the optional use of different learning tools.

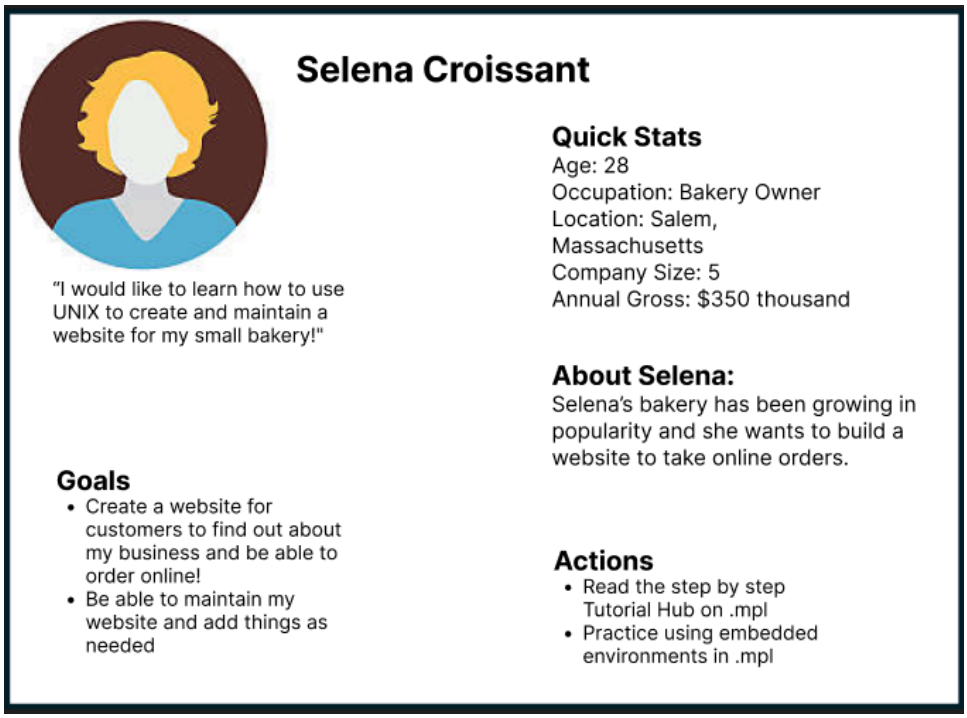
# Site Goals

- Provide tutorials and explanations on each relevant command and topic, allowing users new to UNIX to learn from the beginning.
- Include practice environments with small exercises that translate to use of the real command line.
- Provide a quick reference list of commands for users looking for a refresher on syntax or definition.

# Audience

The purpose of .mpl is to appeal to a varying audience. General computer competency is expected, but a user does not even need to know what the command line is if they're looking to start from the beginning. However, .mpl can also service those with background knowledge looking for extra practice or quick reference for commands they've used previously but blank on in the moment. This sets our age range approximately teens-40s (people older than this range would certainly be capable of utilizing .mpl successfully, but it is unlikely that there would be very many interested in the topic).

## Personas



## Personas cont.

### James Midlife-Crisis

"I need to brush up on my UNIX in order to maintain an old personal resume website I had created back in grad school."

**Quick Stats**
Age: 40
Occupation: Graphic Designer but looking to switch to UX
Location: NYC, New York.

**About James:**
James is having a midlife crisis. He is losing interest in pure graphic design and would like to work more directly on UX features. He needs to update his website as he's getting back out into the job market.

**Goals**
- Brush up on UNIX in order to update my resume website I had created years ago.

**Actions**
- Refresh needed sections in the Tutorial Hub on .mpl
- Utilize Quick Reference Guide while building website.

### Michael Studious

"I am learning UNIX in my undergrad classes and want outside study support for my assignments."

**Quick Stats**
Age: 19
Major: Web & Mobile Development
Location: Henrietta, New York

**About Michael:**
Michael is a freshman undergraduate student at RIT. He is finding his courses rather difficult and looking for online tools to practice course content.

**Goals**
- Complete practice exercises.
- Test himself on knowledge and understanding.
- Become more comfortable with less thoroughly covered material.

**Actions**
- Practice using embedded environments in .mpl
- Utilize Quick Reference Guide while completing assignments

# Competitive Analysis

## UNIX Tools

http://www.unixtools.com
http://www.unixtools.com/build-a-website.html

This is a good reference resource, with a well-categorized combination of general tutorials for beginners and specific information for advanced users (who'd know better what they're looking for.) What it lacks is depth, interactivity and opportunities to practice those skills. Something like a zyBook's interactive sections could bring our site above this one in that regard. It also leans fairly heavily on external resources and tutorials, functioning almost as a collection of other survival guides; the fact that we're keeping tutorials "in-house" will be a benefit to .mpl, allowing users to get all their information without leaving the site.

## UNIX for Beginners Questions & Answers

https://www.unix.com/unix-for-beginners-questions-and-answers/

The Official Unix forums, subpage for Questions and Answers. Where registered users can make and respond to posts. This forum has a plethora of information with a search function to search the forum for keywords. Users who are just beginning to learn would likely have a much more difficult time navigating this forum and actually understanding the information. This is due partly to the fact that many registered users are already familiar with the content and are asking higher level questions, but also that if a thread is responded to, the responding user may make assumptions about the prior levels of the asking user's experience. In short since a majority of the information is crowd sourced it can be harder to comprehend and less reliable at times.

# Competitive Analysis cont.

## W3Schools

https://www.w3schools.com/howto/howto_make_a_website.asp

W3Schools presents a higher-level overview of the website creation process, assuming that the user already has some skills in HTML/CSS. However, they do also have their own tutorials on these lower-level elements, which are linked here for users who need them. It has very good interactive elements, following each section with an embedded tutorial environment. W3Schools' own site building/hosting platform is recommended a lot here; while .mpl can't provide that in this project due to lacking the infrastructure it would require, we should take note of its inclusion and could potentially recommend some hosting options in our final product.

## Tutorialspoint

https://www.tutorialspoint.com/unix/index.htm

Tutorialspoint provides tutorials on many different languages and operating systems. While the tutorials are more text based with less interactivity, they are definitely geared towards being beginner friendly, with higher/intermediate level info available as well. What they lack in interactivity, they make up for in their easy to navigate side menu with specific info readily available. The menu is split into sections with sublevels open so it is easy to navigate all the components of each section. For our site we will need to ensure the information is easily accessible but not overwhelming in the amount of possible selections.

# Competitive Analysis cont.

## Codecademy

https://www.codecademy.com/

Codecademy has a range of courses on a wide variety of programming and general tech topics. As such, its course hub has several paths to get the user to the content, which we could learn from. They first present a quiz for users unsure of their needs, then link to several broad topics, and finally have a list of individual courses. We can mirror this structure to better guide our own users, putting our main action paths first with more specific options lower down. Their actual courses have a strongly linear structure, requiring completion of short interactive tasks to move on to the next topic. However, .mpl will not adopt this structure, as it would run counter to our goals of broad and easy accessibility by locking site content behind more prolonged engagement.

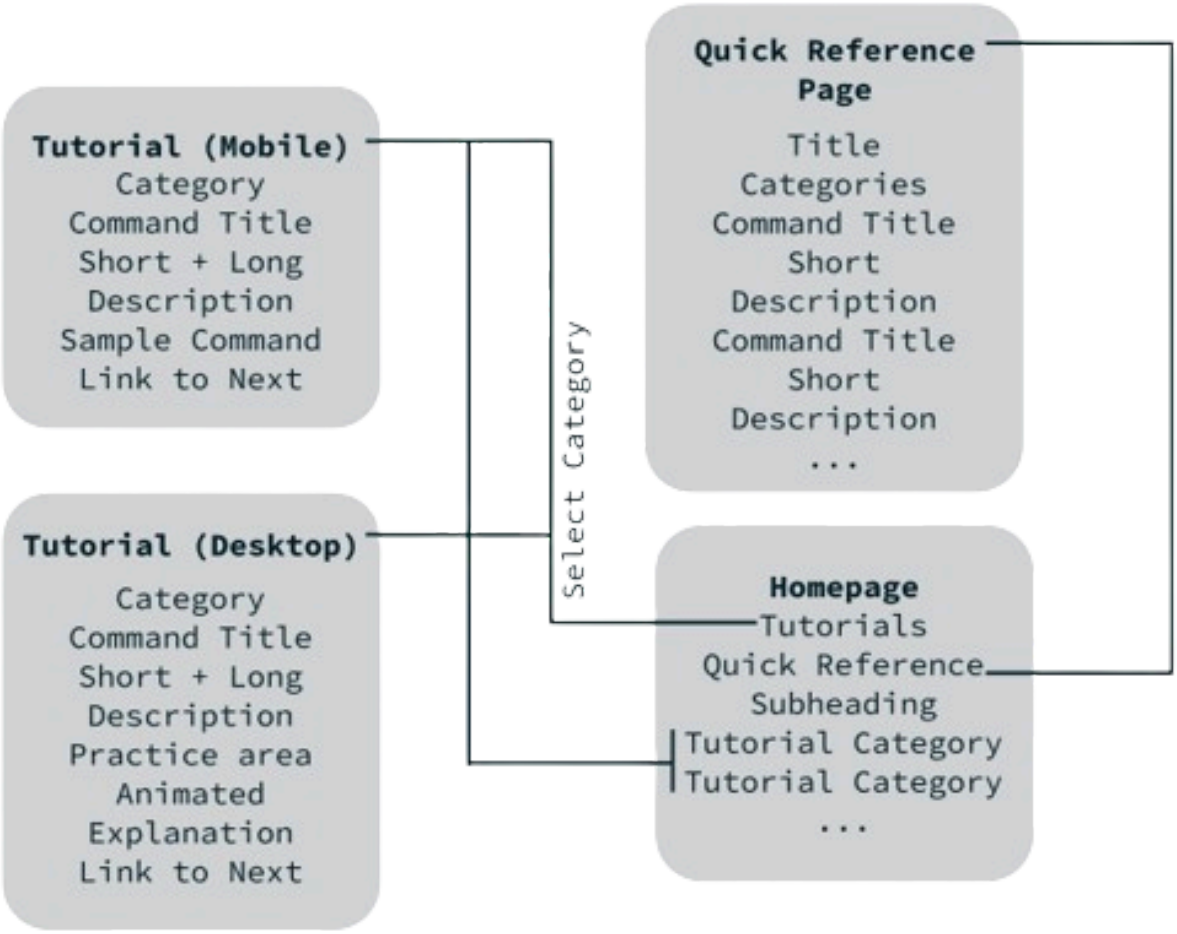
# Site Content

## Site Map

Homepage
Quick Reference Page
Tutorial hub

1. Unix file structure
2. User settings
3. Connecting to a server
4. Creating, moving, copying files
5. File permissions
6. Editing files with vim
7. Uploading files

## Content Model




## Content Grouping and Labeling

**Homepage** links to main tutorial hub, reference page, suggested starter tutorial(s) and external resources.

**Tutorial** hub links to separate tutorials, with icon and brief description.

- **Unix file structure**
  - ***ls*** - Lists your files.
    - Stands for "list." It is a command used to display a list of files and directories in the current directory.

***ls -l***: Lists your files in 'long format', providing detailed information such as file size, owner, permissions, and modification date.

ls -l is an extended version of the ls command. It displays files and directories in a detailed, or "long" format, showing additional information like file permissions, owner, group, file size, and modification date/time.

***ls -a***: Lists all files, including hidden files with names starting with a dot.

ls -a is used to list all files and directories in the current directory, including hidden files and directories. Hidden files in Unix-like systems typically start with a dot (e.g., .config).

***cd filename***: Change to a directory that is within/a child of the current working directory.

You always start out in your 'home directory', and you can get back there by typing 'cd' without arguments. 'cd ..' will get you one level up from your current position. You don't have to walk along step by step - you can make big leaps or avoid walking around by specifying pathnames.

cd stands for "change directory." It is used to navigate the file system by changing the current working directory. You can use it to move into a specific directory by providing its name (e.g., cd my_directory) or move up one level with cd .. Typing cd without arguments takes you back to your home directory, which is where you start when you open a terminal.

***pwd*** - Shows the current directory you are working in.

pwd stands for "print working directory." It displays the full path of the current working directory, allowing you to see where you are located within the file system.

- **User settings**

***whoami*** - Retrieves and displays your username.

The whoami command is used to identify the currently logged-in user on a Unix-like system. It prints the username associated with the active user session.

***passwd*** - Provides a means to modify your password.

The passwd command allows a user to change their own password. When you run this command, you will be prompted to enter your current password and then set a new password. It is a crucial command for maintaining account security.

***ps -u yourusername*** - Lists your processes, including process IDs.

The ps command is used to list running processes on the system. When used with the -u flag followed by a username (e.g., ps -u yourusername), it displays a list of processes associated with that specific username. The list includes information such as the process ID (PID), CPU and memory usage, and more.

***kill PID*** - Ceases the execution of a process by specifying its unique Process ID (PID).

The kill command is used to terminate or send signals to running processes. By providing the PID of a process (e.g., kill 1234), you can stop its execution. The default signal sent by kill is SIGTERM, which allows the process to clean up before exiting. If a process doesn't respond to SIGTERM, you can use kill -9 PID (where 9 is the SIGKILL signal) to forcefully terminate it.

- **Connecting to a server**

  SSH (Secure Shell):

  ***ssh <user>@ipaddress/server address*** - connect to a remote server using SSH.

  SSH stands for "Secure Shell," and it is a network protocol and command-line tool used for secure remote access and secure data communication between two systems over an unsecured network. SSH provides strong encryption and authentication mechanisms, making it a secure way to access and manage remote servers and devices.

  Example: ssh <user>@ipaddress/server address

You use the ssh command followed by the username (<user>) and the IP address or server address to establish a secure shell session with a remote server. After entering the command, you'll be prompted to enter the password or use other authentication methods like SSH keys.

FTP (File Transfer Protocol):
***ftp ipaddress/server address*** - connect to remote server using FTP

FTP stands for "File Transfer Protocol." It is a network protocol and a command-line tool used for transferring files between a local computer and a remote server over a network. FTP is often used for non-secure file transfers, as it does not provide encryption for data transmission.

Example: ftp ipaddress/server address
You use the ftp command followed by the IP address or server address to establish an FTP connection with a remote server. Once connected, you can use FTP commands to upload, download, and manage files on the remote server.

SFTP (Secure File Transfer Protocol):
***sftp <user>@ipaddress/server address*** - connect to a remote server using SFTP

SFTP stands for "Secure File Transfer Protocol." It is an extension of SSH that provides secure file transfer capabilities. SFTP encrypts both the authentication and data transfer processes, making it a secure and preferred choice for transferring files over SSH connections.

Example: sftp <user>@ipaddress/server address
Similar to SSH, you use the sftp command followed by the username (<user>) and the IP address or server address to establish a secure SFTP connection with a remote server. Once connected, you can securely upload, download, and manage files on the remote server using SFTP commands.

- **Creating, moving, copying files**

***more filename***: Displays the first part of a file on the screen. Press spacebar to view more and 'q' to quit. You can search for patterns using '/pattern'.

The more command is used to view the contents of a text file one screen at a time. It displays the beginning of the file and allows you to scroll through it. You can press the spacebar to view more content, and 'q' to quit. Additionally, you can search for patterns within the file using '/pattern'.

***mv filename1 filename2***: Moves a file or renames it.

The mv command is used to either move a file from one location to another or rename a file. If you provide two filenames, it will move the first file to the location specified by the second filename. If the second filename is in a different directory, it effectively moves the file to that directory. If the second filename is the same as the first but different from its current path, it renames the file.

***cp filename1 filename2***: Copies a file.

The cp command is used to create a copy of a file. It takes two filenames as arguments, where the first is the source file, and the second is the destination where the copy will be created. This command duplicates the content of filename1 into filename2.

***rm filename***: Removes (deletes) a file. Use 'rm -i' to prompt for confirmation before deletion.

The rm command is used to delete files. When followed by a filename, it permanently removes the specified file. Be cautious when using rm, as deleted files are usually not recoverable. To prompt for confirmation before deletion, you can use the 'rm -i' option.

***diff filename1 filename2***: Compares two files and shows the differences.

The diff command is used to compare the contents of two text files and display the differences between them. It highlights the lines that differ between filename1 and filename2, making it useful for comparing and identifying changes in files.

***wc filename***: Counts the number of lines, words, and characters in a file.

The wc (word count) command is used to count various metrics in a text file, including the number of lines, words, and characters. When used with a filename as an argument, it provides a breakdown of these counts for the specified file.

- **File permissions**

  ***chmod options filename***: Changes file permissions to read (r), write (w), and execute (x) for different users. For example, 'chmod o+r' makes the file readable for others.

  The chmod command is used to change the permissions (access rights) of a file or directory. It allows the user to control who can read, write, and execute a file. The command is followed by various options and a filename:
  options: Specifies the permissions to be modified. These options typically include:

  u (user): Refers to the owner of the file.
  g (group): Refers to the group associated with the file.
  o (others): Refers to users who are not the owner or in the group.
  a (all): Refers to all users, including the owner, group, and others.
  + (add): Adds the specified permissions.
  - (remove): Removes the specified permissions.
  = (set): Sets the specified permissions and removes all others.

  filename: Specifies the file or directory for which you want to change permissions.

  Permission options include:

  r (read): Allows reading or viewing the file.
  w (write): Allows editing or modifying the file.
  x (execute): Allows executing the file if it's a program or script.

  Example:

  chmod o+r filename: Adds read permission for others, allowing them to read the file.

chmod u+x script.sh: Adds execute permission for the file script.sh for the owner, enabling them to run it as a script.

It's essential to use chmod carefully, as changing permissions can affect the security and functionality of files and directories. Properly managing file permissions is important for maintaining system security and access control.

- **Editing files with vim**
  - Entering vim
    - ***Vim file_name*** - will open the respective file to edit or create the file if it doesn't exist
  - The three modes of vim
    - Normal
      - Default mode. For navigation and simple editing. Enter this mode by hitting Esc
    - Insert
      - Interesting and modifying of texts. Enter this mode by hitting i
    - Command line
      - Using commands to operate vim. Enter this mode by hitting Esc then :
  - Cursor movement
  - Basic movement
    - ***h*** - move cursor left
    - ***j*** - move cursor down
    - ***k*** - move cursor up
    - ***l*** - move cursor right
    - ***Up arrow*** - move cursor up
    - ***Down arrow*** - move cursor down
    - ***Left arrow*** - move cursor left
    - ***Right arrow*** - move cursor right
  - Advanced movement
    - ***gj*** - move cursor down (multi-line text)
    - ***gk*** - move cursor up (multi-line text)
    - ***H*** - move to top of screen
    - ***M*** - move to middle of screen

***L*** - move to bottom of screen
***w*** - jump forwards to the start of a word
***W*** - jump forwards to the start of a word (words can contain punctuation)
***e*** - jump forwards to the end of a word
***E*** - jump forwards to the end of a word (words can contain punctuation)
***b*** - jump backwards to the start of a word
***B*** - jump backwards to the start of a word (words can contain punctuation)
***ge*** - jump backwards to the end of a word
***gE*** - jump backwards to the end of a word (words can contain punctuation)
***%*** - move cursor to matching character (default supported pairs: '()', '{}', '[]'
***0*** - jump to the start of the line
***^*** - jump to the first non-blank character of the line
***$*** - jump to the end of the line
***g_*** - jump to the last non-blank character of the line
***gg or [[***- go to the first line of the document/beginning of file
***G or ]]*** - go to the last line of the document/end of file
***5gg or 5G*** - go to line 5
***fx*** - jump to next occurrence of character x
***tx*** - jump to before the next occurrence of character x
***Fx*** - jump to the previous occurrence of character x
***Tx*** - jump to after the previous occurrence of character x
***;*** - repeat previous f, t, F or T movement
***,*** - repeat previous f, t, F or T movement, backward
***}*** - jump to next paragraph (or function/block, when editing code)
***{*** - jump to previous paragraph (or function/block, when editing code)
***zz*** - center cursor on screen
***zt*** - position cursor on top of the screen
***zb*** - position cursor on bottom of the screen
***Ctrl + e*** - move screen down one line (without moving cursor)
***Ctrl + y*** - move screen up one line (without moving cursor)
***Ctrl + b*** - move screen up one page (cursor to last line)

***Ctrl + f*** - move screen down one page (cursor to first line)
***Ctrl + d*** - move cursor and screen down 1/2 page
***Ctrl + u*** - move cursor and screen up 1/2 page

Insert mode and editing

***i*** - insert before the cursor
***I*** - insert at the beginning of the line
***a*** - insert after the cursor
***A*** - insert at the end of the line
***o*** - create a new line below the current line
***O*** - create a new line above the current line
***ea*** - insert at the end of the word
***Ctrl + h*** - delete the character before the cursor during insert mode
***Ctrl + w*** - delete word before the cursor during insert mode
***Ctrl + j*** - add a line break at the cursor position during insert mode
***Ctrl + t*** - indent (move right) line one shift width during insert mode
***Ctrl + d*** - de-indent (move left) line one shift width during insert mode
***Ctrl + rx*** - insert the contents of register x
***Esc or Ctrl + c*** - exit insert mode
***dd***- delete current line
***r*** - replace a single character.
***R*** - replace more than one character, until ESC is pressed.
***J*** - join line below to the current one with one space in between
***gJ*** - join line below to the current one without space in between
***cc*** - change/replace entire line
***c$ or C*** - change/replace to the end of the line
***ciw*** - change/replace entire word
***cw or ce*** - change/replace/ to the end of the word
***s*** - delete character and substitute text
***S*** - delete line and substitute text (same as cc)
***xp*** - transpose two letters (delete and paste)
***u*** - undo

***U*** - restore/undo last changed line
***Ctrl + r*** - redo
***.*** - repeat last command

Copy and Paste

***yy*** - copy a line
***2yy*** - copy 2 lines
***yw*** - copy the characters of the word from the cursor position to the start of the next word
***yiw*** - copy word under the cursor
***yaw*** - copy word under the cursor and the space after or before it
***y$ or Y*** - copy to end of line
***p*** - paste the clipboard after cursor
***P*** - paste the clipboard before cursor
***gp*** - paste the clipboard after cursor and leave cursor after the new text
***gP*** - paste before cursor and leave cursor after the new text
***dd*** - cut a line
***2dd*** - cut 2 lines
***dw*** - cut the characters of the word from the cursor position to the start of the next word
***diw*** - cut word under the cursor
***daw*** - cut word under the cursor and the space after or before it
***:3,5d*** - delete lines starting from 3 to 5
***:g/{pattern}/d*** - delete all lines containing pattern
***:g!/{pattern}/d*** - delete all lines not containing pattern
***d$ or D*** - delete (cut) to the end of the line
***x*** - delete a character

Exiting and Saving

***:w*** - write (save) the file
***:w new_file_name*** - save the file under a new name and continue editing original
***:w !sudo tee %*** - write out the current file using sudo
***:wq or :x or ZZ*** - write (save) and quit
***:q*** - quit (fails if there are unsaved changes)

***:q! or ZQ*** - quit without saving
***:wqa*** - write (save) and quit on all tabs

Find and Replace

***:/pattern*** - search for pattern/keyword
******* - jump to next instance of the pattern
***#*** - jump to previous instance of pattern
***?pattern*** - search backward for pattern
***vpattern*** - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)
***n*** - repeat search in same direction
***N*** - repeat search in opposite direction
***:%s/old/new/g*** - replace all old with new throughout file
***:%s/old/new/gc*** - replace all old with new throughout file with confirmations
***:noh[lsearch]*** - remove highlighting of search matches

- **Downloading and Uploading files**

  SSH

  ***SCP*** (Secure Copy) is a command-line tool and protocol for securely copying files between local and remote systems using SSH for authentication and data transfer. It allows you to copy files securely between your local machine and a remote server.
  ***scp <local_filepath> user@hostname:<filepath>***: copy file from your machine to server

  This command copies a file from your local machine to a remote server. You specify the local file path, your username (user), the server's hostname (hostname), and the destination file path on the server.

  ***scp user@server_host:<filepath> <local_path>***: copy file from server to your machine

  This command copies a file from a remote server to your local machine. You specify the remote user, the server's hostname (server_host), the source file path on the server, and the local destination path.

  ***FTP/SFTP*** (File Transfer Protocol/ Secure File Transfer Protocol)

FTP is a standard network protocol used for transferring files between a client and a server on a computer network. SFTP, on the other hand, is an extension of SSH that provides secure file transfer capabilities over an SSH connection.
SFTP operates similarly to FTP but is secured using SSH for encryption and authentication. The usage is similar to FTP, but the connection is encrypted, making it a more secure option for file transfer.

***get filename***: download the specified file

This FTP command is used to download the specified file from the remote server to your local machine.

***mget filenames***: download multiple specified files

Similar to get, this command allows you to download multiple specified files from the remote server to your local machine.

***put filename***: upload the specified file

This command is used to upload the specified file from your local machine to the remote server.

***mput filenames***: upload the specified files

Similar to put, this command allows you to upload multiple specified files from your local machine to the remote server.

# Design

## Desktop

Each command/concept will have a textual explanation, example code snippet, and an embedded practice environment. The base embedded environment resembles a terminal into which users can type commands and see their results. There is also an option to show a graphical representation of the results, allowing users to visualize the effects of each command.

Ex. ‘cd ...’ will show a tiny person icon moving from one directory to another

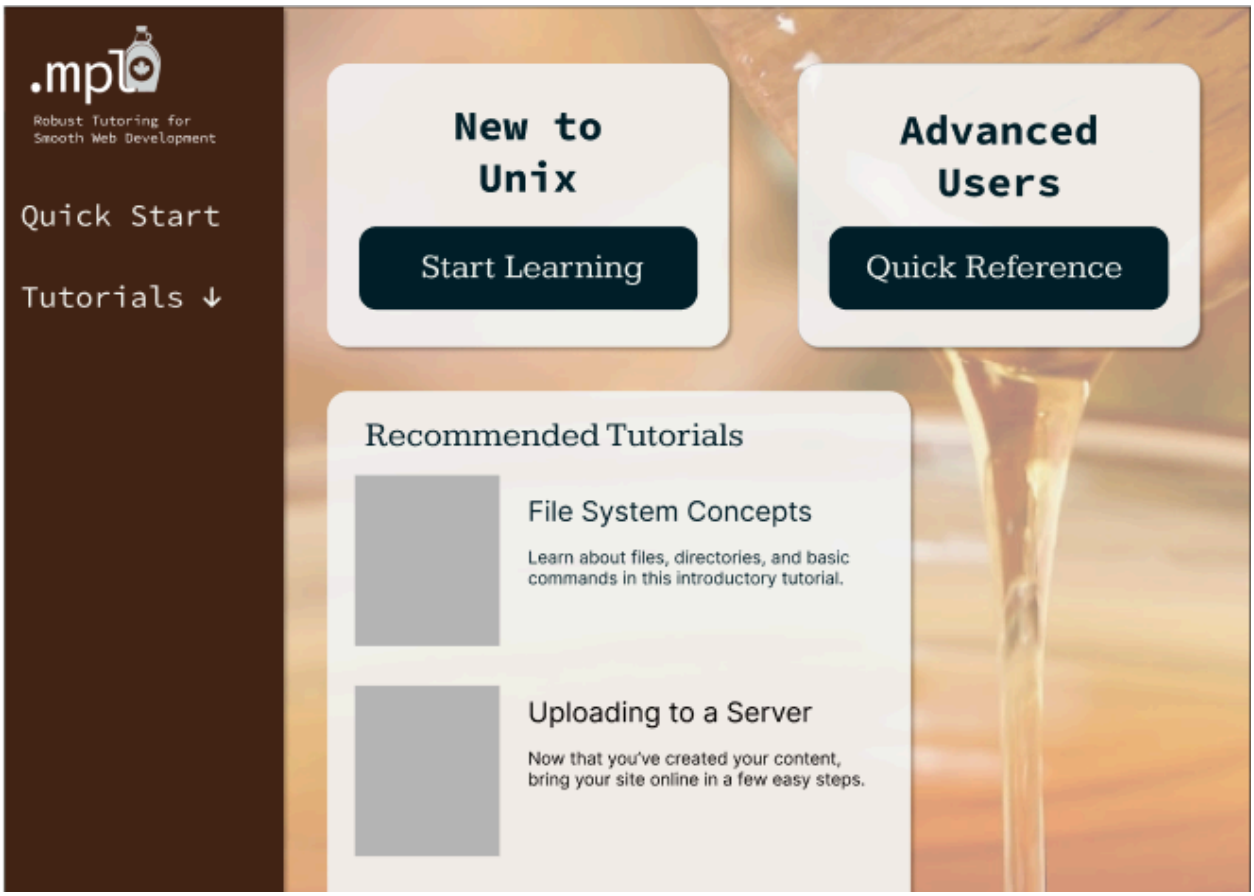
The quick reference guide is a searchable list of commands organized by section/topic and alphabetical order within that section. This will include command name, command definition, and an example code snippet.

## Mobile

Each command/concept will include the same tutorials with command name, explanation, and example code snippet, but not interactive environment options, as the purpose of the interactive environment is to replicate a real command line, which is not accessible on a mobile device.

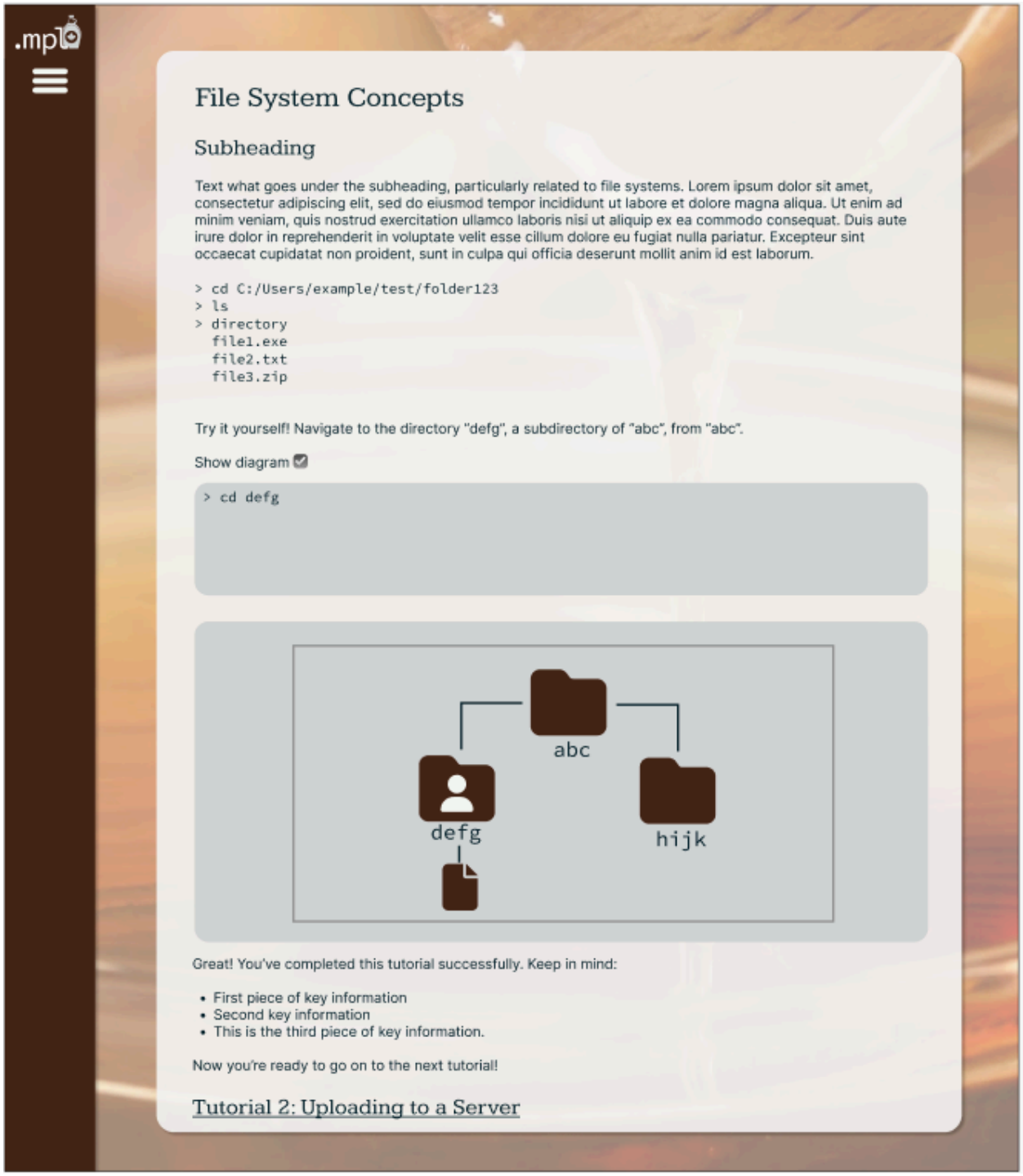
The quick reference guide is unchanged; a searchable list of commands organized by section/topic and alphabetical order within that section. This will include command name, command definition, and an example code snippet.

## Desktop Wireframes

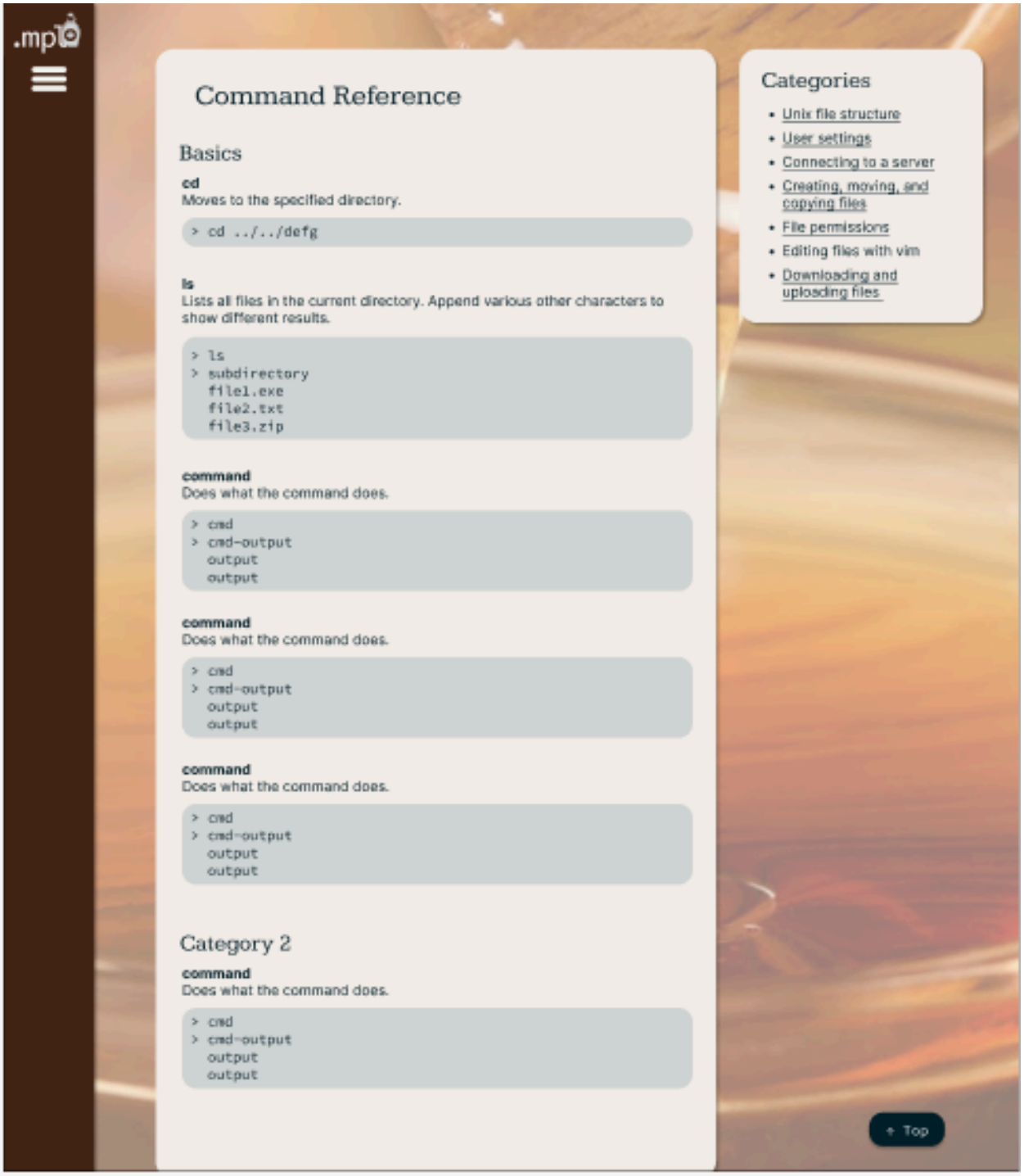



Homepage with sidebar extended.


Homepage with sidebar collapsed.



Sample tutorial page.




Quick Reference Guide.


Grayscale wireframes demonstrate hierarchy

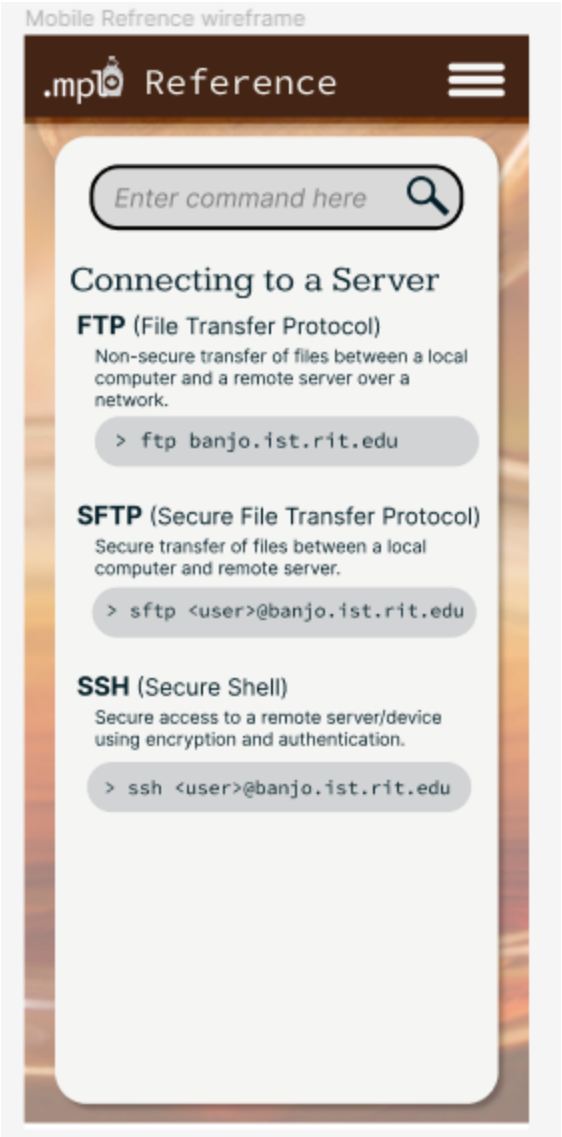

Grayscale tutorial page.

Grayscale reference page.

## Mobile Wireframes


Mobile Home Page


Navigation Extended

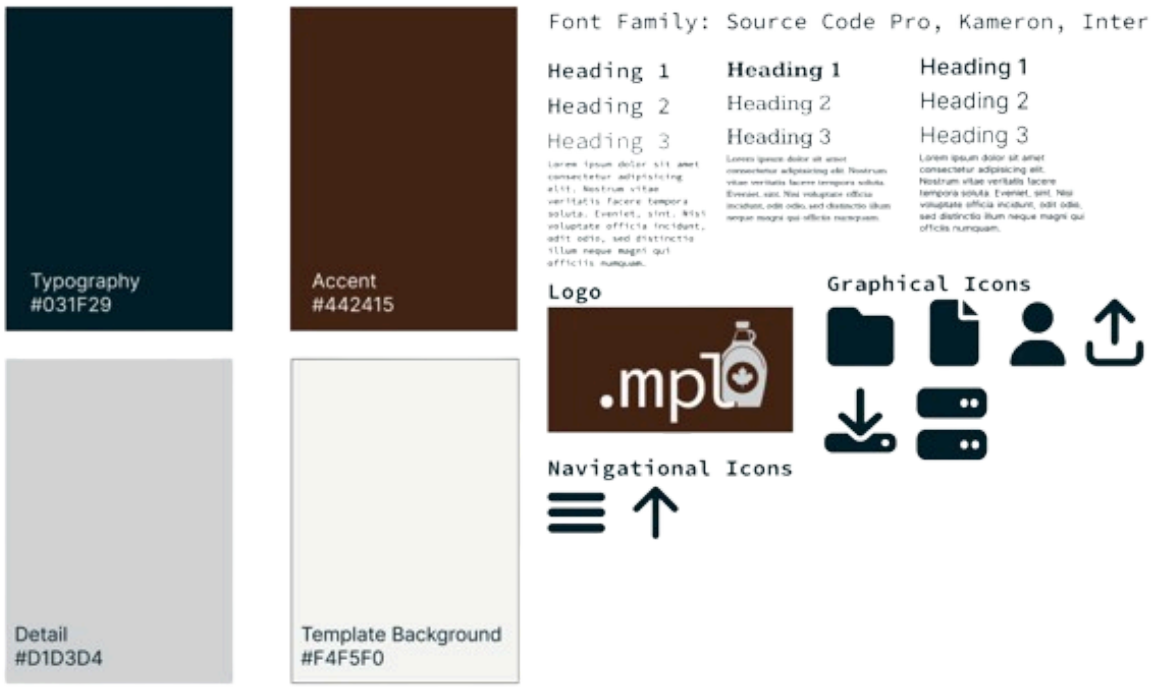



Mobile Quick Reference Page


Mobile Tutorial Page

Mobile Tutorial Navigation

## Style Guide




Website “panes” will use the background white at 85% opacity.

Kameron will be used for titles, headings, and other important text.
Inter will be used for body text.
Source Code Pro will be used for logo, code snippets, and interactive terminals.
As a fallback, the site will default to the CSS sans-serif font family.

Folder: 
File: 
User: 

# Requirements

## Essential Requirements

1. Interactive tutorials
    a. Teach content, including example code snippets and a separate terminal portion (starts blank) for users to practice. This terminal can verify that their practice code is correct.
    b. Animations show the effect of commands with animations that react to user-typed commands.
2. Quick Reference List for important commands
    a. Reference is simpler, with command, brief description of its function, example snippet.
3. Responsive website able to display content for any device. However, focus on desktop-first design; because UNIX is rarely used on mobile devices, they are not our first priority.

## Desirable Requirements

1. Mobile version includes limited graphics, such as automatically demonstrating commands.
2. Tutorials provide a visual demonstration of the command being taught.
3. Parallax scrolling provides additional visual interest to keep the site from looking static.

## Global/Accessibility Considerations

1. Screen reader compatibility
2. Functionality available from a keyboard
3. Best-practice HTML & CSS — correct tags
4. Web Content Accessibility Guidelines 2.1
    a. Contrast between text and background must be above a 4.0
    b. Alt text provided for all images

c. Readable fonts
d. Animations are not overly quick

### Deployment Environment

.mpl will be deployed on the Rochester Institute of Technology's Solace.rit.edu server using php and mysql. GitHub will be utilized during the development process for sharing of files amongst the development team.

## Conclusion

.mpl has been designed to satisfy the needs of varying users across multiple skill levels and goals to learn UNIX. The developers of .mpl have created a tutorial template and provided the necessary information to allow each command/concept to be fit into this template and consumed by the user. Interactive environments were designed to give users an opportunity to practice their new skills and a graphical interface option was included for those who are visual learners. A quick reference guide template has also been designed and can be populated with the included information for each command/concept, so that advanced users who don't need full tutorials can still use .mpl as a resource.

# Appendix

We use free icons from FontAwesome.com in our tutorials and a few other graphics (navigation arrows.)

Sources for UNIX commands:

- https://phoenixnap.com/kb/vim-commands-cheat-sheet
- https://linuxhandbook.com/basic-vim-commands/
- https://vim.rtorr.com/
- https://www.freecodecamp.org/news/linux-server-tutorial/
- https://www.baeldung.com/linux/ftp-transfers
- https://tecadmin.net/download-and-upload-files-with-sftp/
- https://mally.stanford.edu/~sr/computing/basic-unix.html
- http://titan.csit.rmit.edu.au/~e70949/inductionguide.pdf

Source for page background:

- https://www.judeesfromscratch.com/blogs/recipes/making-liquid-maple-syrup-from-granulated-maple-sugar

## Meeting Notes

September 14, 2023, 6:30 - 7:00 PM
Olivia, Emmett, & Callum present

- Reviewed assignment rubric/requirements
- Confirmed team roles & decision-making process (simple majority, ideally consensus)
- Scheduled production meetings (Thursdays, 6:30 PM) and progress check-ins (online/asynchronous, EOD Tuesdays)
- Set design document milestones & deadlines
    - Overall document due Oct. 4th. By the following dates:
    - Sept. 19th- determine who the audience is- determine what the content is supposed to be
    - Sept. 21st- determine requirements
    - Sept. 26- competitive analysis
    - Sept. 28th- complete site content- complete design
    - Oct. 4th- title page- introduction- conclusion- appendix- fix any outstanding issues
- Selected a team name (Team Maple Syrup!)

Next meeting scheduled for September 21

September 21, 2023, 6:30 - 7:45 PM
All members present

- Split into two groups to write user personas/competitive analysis
- Discussed content requirements, agreed to consult with client (Prof. Takats) and return to the subject in an extra meeting once we had more information
- Adjusted schedule slightly — competitive analysis is being worked on earlier than planned, so requirements are being pushed later (see prev.; dates essentially switched)
- Worked on potential site branding/logos

Next meeting scheduled for September 27

September 27, 2023, 6:30 - 8:30 PM
Olivia & Callum present, Zach joined later (interview), Emmett at work

- Established content requirements
- Created wireframes
- Continued branding work
- Created sitemap

Next meeting scheduled for September 28

September 28, 2023, 6:30 - 8:30 PM

- All members present
- Split up; one team on design, one on content
- Improved sitemap
- Clarified requirements
- Added content (created individually) to design doc
- Refined homepage wireframes and started developing final style guide

Design document and content to be completed by Tuesday, allowing a day for final checks
With the due date extension, next meeting scheduled for October 5.

October 5, 2023, 6:30 - 9:00 PM
All members present

- One more pass of the design document; consistent formatting, added more wireframes, ensured all sources present, added meeting notes
- Final review and small revision of wireframes
- Finalized and added icons to style guide

Design document to be submitted tomorrow after a really extra double final check.

Nov 16, 2023, 6:30-7:30
All members except Callum present

- Set up group solace/adminer/dbCon
- Started HTML/CSS for homepage
- Plan to have HTML and CSS done by Monday the 27th
- Work delegations:
    - Olivia: Homepage
    - Emmett: Quick Reference Guide
    - Zach: Tutorials
    - Callum: TBD. CSS/troubleshooting/responsive design
- Work over this coming weekend, group check in at some point Monday the 20th

Nov 27, 2023, 6:15-7:45
Olivia and Callum present in person, Emmett and Zach available remotely

- Regrouped on site work. Base HTML and CSS templates largely completed
- Began implementing includes
- Decided against implementing a database as there is no user input or altering of the data. Will use one to save user progress if time allows.
- Work delegations:
    - Emmett: designing animations
    - Callum and Zach: finalizing styling
    - Olivia: TBD. (I am currently experiencing inexplicable technical issues. When I try to download the group hosted files in order to edit and host them locally the styling does not transfer over. Currently unable to see the changes I make. Callum will be taking over styling the homepage until I can get into Prof's office hours this coming Wednesday and figure out the issue.)
- Next meeting scheduled for 11/30.
- Group presentation scheduled for 12/4. Goal for is to have one working page of each type (homepage, quick reference, and one tutorial).

Nov 30, 2023, 6:30-7:30
Olivia, Emmett, and Zach present.

- Finalized animation purpose and strategies
    - Static graphics will be part of the prompt as a visual representation for what we're asking them to do.

    - Commands typed into the mock terminal will  be string checked against the database.
    - Once the correct command is submitted, the animation will play.
- Worked on general issues with the nav.
- Began animation designs and implementation.
- For next meeting (likely Monday the 4th):
    - Emmett: animation designs done! We need to start building them.
    - Zach: nav issues fixed.
    - Callum: fix styling issues on the homepage to match design document (there are comments in the files, hit Olivia up with any further questions)
    - Olivia: complete first animation and build slide presentation for Monday
- Our website is overall much further along than other groups (yay!). Big ticket items that still need to be done for the final due date:
    - Modularization
    - Database design and implementation