Pill Terminal

Eloy Rodriguez

Albert Grandes

Robert Mihai

Guillermo Valdez

Index:

Index…………………………………………………………………………………………………2

The program application…………………………………………………………………..3

The Backend……………………………………………………………………………………..4

Flow of program……………………………………………………………………….....5

The Frontend……………………………………………………………………………………6

HTML…………………………………………………………………………………………….6

CSS…………………………………………………………………………………………….....6

Example of use………………………………………………………………………………….7

Different browser screenshot……………………………………………………………8

Chrome……………………………………………………………………………………......8

Edge………………………………………………………………………………………………8

Mozilla firefox……………………………………………………………………………….8

Internet Explorer…………………………………………………………………………..8

Bugs or Problems……………………………………………………………………………..9

Problems………………………………………………………………………………………9

Problem of compatibility……………………………………………………..9

Other problems……………………………………………………………………9

Pill Organization……………………………………………………………………………..10

Api Documentation..……………………………………………………………………….11

**The program application**

**Terminal Pill** is a terminal where you can open more than one terminal and be able to work independently in each one of them.

**The Backend**

The backend is divided in 2 javascripts files, one is for functions and the other one is for the visual part.

The structure in the backend is divided in the next index:

Index:

1. Automatic run:

* In this section we have the code that always runs when the program start.

$(document).ready(function(){})

1. Variables:

* In this section we have the variables that we use to be global in all the project.

let COMPUTER;

let path;

let commandHistory;

let chIndex = 0;

let terminal = $(“terminal”);

let output = $(“output”);

let terminalInput = $(“#cin”);

let rootDir = $(“#rootDir”);

1. Quote petition:

* In that command we have the functions that prepare, manage and send the petitions to the Programming Quote API.

quote()

1. Commands:

* In this big section we have differents commands. This commands should allow you to navigate.

Write help to know what all commands do.

1. Other functions:

* In this section we save all the paths in the localStorage.

localStorage()

Flow of program:

The normal flow of the program starts with the *$(document).ready(function(){})* that shows the favorite elements saved in your localStorage.

After that, the program needs the user interaction. You can write different commands in the input. You can open more terminals or close them. To open more terminals you press the plus button in the right corner and to close them you click the dot in the right side from the terminal bar. First command you can use when you are inside the terminal is “help” and this command will show you all the different commands you can use in our terminal. You use “mkdir” to create new directories or “touch” to create new files. If you wanna see the other commands you will have to enter our terminal.

**The Frontend**

The frontend is divided in two, one for the HTML and the other for the CSS (style) and one javascript file for the modelation of the terminal.

**HTML**

We have three elements, the background, the button to add more terminals and the default terminal. The terminal have a header that indicates the number of the terminal and the close button.

**CSS**

We use flexbox, but in this project the most part of the magic was made with the help of javascript that allows to drag and drop the terminal windows and resize them.

**Example of use**

When you enter, I recommend you to use the command help, and in this way you can see all commands you can use (Ex: mkdir to create a new directory, touch to create a new file). After this you can use ls and you will see if the directory or the file was created. You can use cd to change the directory and cd .. to go back. Pwd is a command that shows you where you are right now. Echo to add a text to a file and cat to see that text. After every command you can use clear so the output will be clean and nice. If you want to see all the commands and use them at the same time in our web terminal you can open as muchs terminals as you wish and close them by pressing on the black circle in the terminal header.

**Differents browser screenshots**

**Bugs or Problems**

* The problem was the organization of json inside localStorage.
* at first we didn’t know how to work the input value
* Alber Grandes is a bug, but is worst Guillermo Valdez
* when we start the project, we don't know work with object
* at first back-end didn’t work together, and we had some problems of organization at the code
* worked with different terminals, now work :V

**Pill organization**

The project was divided in different natural steps. We start divided in three groups. One working frontend, other working in backend connected with frontend and last two in backend.

The second division was that the frontend carry javascript work of frontend and the other three person of the teams go to backend. Finally all the team work in backend.

**Api documentation**