



## dontpanic\_makeover

Another short slice, but one that draws a lot of attention.

*Summary: Dontpanic is now more flashy and you don't have to worry about the heavy lifting – at least a slice of it.*

*Version: 1.0*

# Contents

|            |                                    |          |
|------------|------------------------------------|----------|
| <b>I</b>   | <b>Foreword</b>                    | <b>2</b> |
| <b>II</b>  | <b>Introduction</b>                | <b>3</b> |
| <b>III</b> | <b>General instructions</b>        | <b>4</b> |
| <b>IV</b>  | <b>Mandatory part</b>              | <b>5</b> |
| IV.1       | Clientside features . . . . .      | 5        |
| <b>V</b>   | <b>Bonus part</b>                  | <b>6</b> |
| <b>VI</b>  | <b>Turn-in and peer-evaluation</b> | <b>7</b> |
| VI.1       | Turn-in . . . . .                  | 7        |
| VI.2       | Peer-evaluation . . . . .          | 7        |

# Chapter I

## Foreword

Known as Eduardo Kobra, nicknamed Kobra, is a street artist who officially began his career in 1987 at 12 years old, in his hometown of São Paulo. Since then he has painted over 3,000 murals on five different continents. Some of these murals are commissioned while others are his original, raw ideas. Kobra now works with a team of artists who paint between two and four murals each month. To this day Kobra continues to live and work in his hometown of São Paulo.



Eduardo Kobra, Monalisa. São Paulo, Vladimir Herzog, 75 - MIS Experience

# Chapter II

## Introduction

Your game is already pretty cool, but there are always ways to improve. The size of things being finite, we won't go that far, but we'll take a few steps to get closer to the state of the art.

The objective of this project is to make your dontpanic game more attractive, using popular technologies and front-end frameworks that help us to create interfaces in a faster, simpler and professionally looking way. For that, you'll have to refactor part of the code, using tools that will do a job you've already done.

At the same time, now that you are more familiar with the parts of your code and how they work, you can't be afraid to delete LOCs<sup>1</sup> and rewrite them.

---

<sup>1</sup><https://www.urbandictionary.com/define.php?term=SLOC>

# Chapter III

## General instructions

- Only clientside code will be changed in your dontpanic game.
- You shouldn't remove features that were already implemented in previous projects, even if you have to redo some part of your project, redo it!
- You are free to recreate your layout in any other way, but the rules starting from the first project must be followed:
  - The clientside must be able to connect with the serverside and respect it's definitions. Back-end's address must be configured on the clientside.
  - It should contain an input to type characters to discover the hidden equation.
  - With the characters placed, the user should be able to send his attempt to the serverside.
  - It should be straight to the point and easy to understand what the user should do the first time accessing it.
- Containers can be updated and modified, but they must follow the rules of the containerization of the previous project.
- Now, with more tools and knowledge at hand, you can use this opportunity to make your project more beautiful than ever!
- Your project should be able to be run with just one command.
- As always, we recommend that you read a lot of documentation about the tools mentioned here in order to understand how to use them.

# Chapter IV

## Mandatory part

### IV.1 Clientside features

- You must use the Bootstrap<sup>1</sup> framework.
- Bootstrap version must be equal or greater than 5.x version.
- You should use as many components and functionality as Bootstrap can give you, replacing what you've already done. Any implementation that does not use Bootstrap will need to be justified in the evaluation.
- Your website should be able to display correctly on mobile devices and have an adapted layout on small resolutions.
- Your web application should at least be compatible with Firefox ( $\geq 91$ ) and Chrome ( $\geq 96$ ).
- Your page should no longer completely reload when an attempt is made in the game. Only essential parts of the page that make the game work should be reloaded.
- If using JavaScript, you should limit yourself to just the browser's native API<sup>2</sup>. No frameworks or libraries are acceptable, like the DOM manipulation library jQuery.



We advise you that you take a look at fetch API

---

<sup>1</sup><https://getbootstrap.com/>

<sup>2</sup><https://developer.mozilla.org/en-US/docs/Web/API>

# Chapter V

## Bonus part

For the bonus to be valid, the mandatory part must be perfect.

How will people know how to play your game? You will probably have to explain to them. Somehow.

For the bonus, you must implement:

- A button in a strategic part of your main page that explains how to play your game. The explanation should be simple and to the point: should not take more than 1 minute to read and understand.
- Another button on your page that lets the user change colors to dark or blind modes. The selected mode must be maintained even if the page refreshes.

# Chapter VI

## Turn-in and peer-evaluation

### VI.1 Turn-in

Turn in your work on your repo Git. Only the work included on your repo will be reviewed during the evaluation.

### VI.2 Peer-evaluation

Another group will evaluate your game and your code and will positively welcome the quality of it.