JS Shopping Cart

Marcelo Gaspar
drmpgaspar@gmail.com
https://github.com/Mpgaspar/shopping-cartJS.git

JS Shopping Cart

18th November 2019

Overview

In this project the company needs to develop a shopping cart using exclusively Javascript. In addition, it is necessary to save the contents of the cart in the client's browser, so it is essential to use localStorage to carry out the implementation.

List of tasks to be performed:
01. Priority of each task
02. Title and description of each of them
03. Difficulty level
04. Estimated time for each task.
05. Record of incidents that were detected during project execution.
06. Record of lessons learned.
07. Project calendar
08. The Chronogram of the project.
09. Quality control measurements.
10. Quality metrics
11. Requirements documentation.
12. Risk documentation in the event that these exist
13. Documentation about the git WORKFLOW you are going to use
14. Documentation about the tools used in the project

Priority (1-3) 1 = low, 2 = moderate, 3 = high

A: Organization	3
B: Layout	2
C: Repository in git	3
D: Structuring in HTML	2
E: localStorage	3
F: JavaScript functions	3
G: CSS styling	2
H: Test	3

A: **Organization** = structuring all tasks by priority and estimated execution time.

B: **Layout** = design container with input and map style.

C: **Repository in git** = put the project into a git repository to organize changes during construction.

D: **HTML Structuring** = create semantics tags and define input button and mapa.

E: **localStorage** = save the contents of the cart in the client's browser.

F: **CSS styling** = the artist part to give the site beauty.Make responsive with Materialize.css

G: **JavaScript functions:**

+ Create a function to show the cart.
+ Show item.
+ Add items to the cart.
+ Show totals
+ Put data in localStorage
+ Conversion value Dollar and Euro
+ Change price with the quantity

H: **Test** = test in all browsers but the code has to be tested all the time.

Difficulty level (1-3) 1 = low, 2 = moderate, 3 = hard

A: Organization and Documentation	2
B: Layout	1
C: Repository in git	1
D: Structuring in HTML	2
E: localStorage	3
F: CSS styling	1
G: JavaScript functions	3
H: Test	2

Time (hours)
A: Organization and Documentation = 4h
B: Layout = 2h
C: Repository in git = 1h
D: Structuring in HTML = 4h
E: localStorage = 6h
F: CSS styling = 3h
G: JavaScript functions = 9h
H: Test = 3h

I: Incidents = 8h (20% total time)

Total time: 40h

Incidents

- Difficulty fulfilling the given time for each task
- Some tasks were underestimated
- Difficulty to create some JavaScript functions
- Stop the project to study some JavaScript concepts used
- Knowledge limitation in JavaScript
- Difficulty optimizing time for each task
- Limitation to implement certain features
- Difficulty to working and creating more elaborate JavaScript functions

Lessons

- Better organize myself to accomplish the tasks
- I learned to prioritize the most important tasks
- Use git to our advantage
- Understand how a shopping cart works
- Improve my knowledge in logic and programming
- Improve my knowledge in Javascript, HTML and CSS
- Improve my knowledge in localStorage.
- Make good use of JavaScript
- Better understand JavaScript functions
- Take questions and discussed project organization with colleagues
- Improve your knowledge in project management.
- Have patience and be humble

Project Calendar

Main tasks
18/11 → Organization and Create repository in git
19/11 → Create Layout and HTML structure
20/11 → Study and work with localStorage
21/11 → Create JavaScript functions
22/11 → Styling with CSS and make responsive
23/11 → Test functions and make the code more clean
24/11 → Test in browsers

TASK/ DAY	MON	TUE	WED	THU	FRI	SAT	SUN
	18/11	19/11	20/11	21/11	22/11	23/11	24/11
A	X					X	
B	X	X					
C	X						
D	X	X	X				
E		X	X				
F			X	X	X		
G		X	X	X	X	X	
H					X		X

Chronogram of the project

18/11 - Organization and division of tasks by priorities
- Start documentation
- Put repository in git
- Start HTML basic structuring
- Study localStorage

19/11 - Continue HTML basic structuring
- Work with git
- Create JavaScript functions
- Work with localStorage

20/11 - Finish HTML basic structuring
- Start basic styling with CSS
- Continue work with git
- Continue work with JavaScript + localStorage

21/11 - Continue styling with CSS
- Continue work with JavaScript

22/11 - Finish styling with CSS
- Continue work with JavaScript

23/11 - Finish JavaScript functions
- Finish documentation

24/11 - Test in browsers
- Push to GitHub repository

Quality control measurements

The quality control measurements are used to analyze as well as evaluate the quality of the different processes involved in a project against the standards of the organization or on the requirements specified during the project management planning.

- Is the HTML code properly validated by the W3C ?
- Are the HTML, JS and CSS files properly formatted ?
- Did you use clean code?
- Is the site responsive and easy for the customer to understand?
- Are user interactions made with JavaScript functionality only?
- Does the site have all the specifications requested?
- Does the site look good when displayed in any of the common browser clients?
- Do the commit messages properly explain the development flow of the project?
- Have all functions been tested throughout the project realization process?

Quality metrics

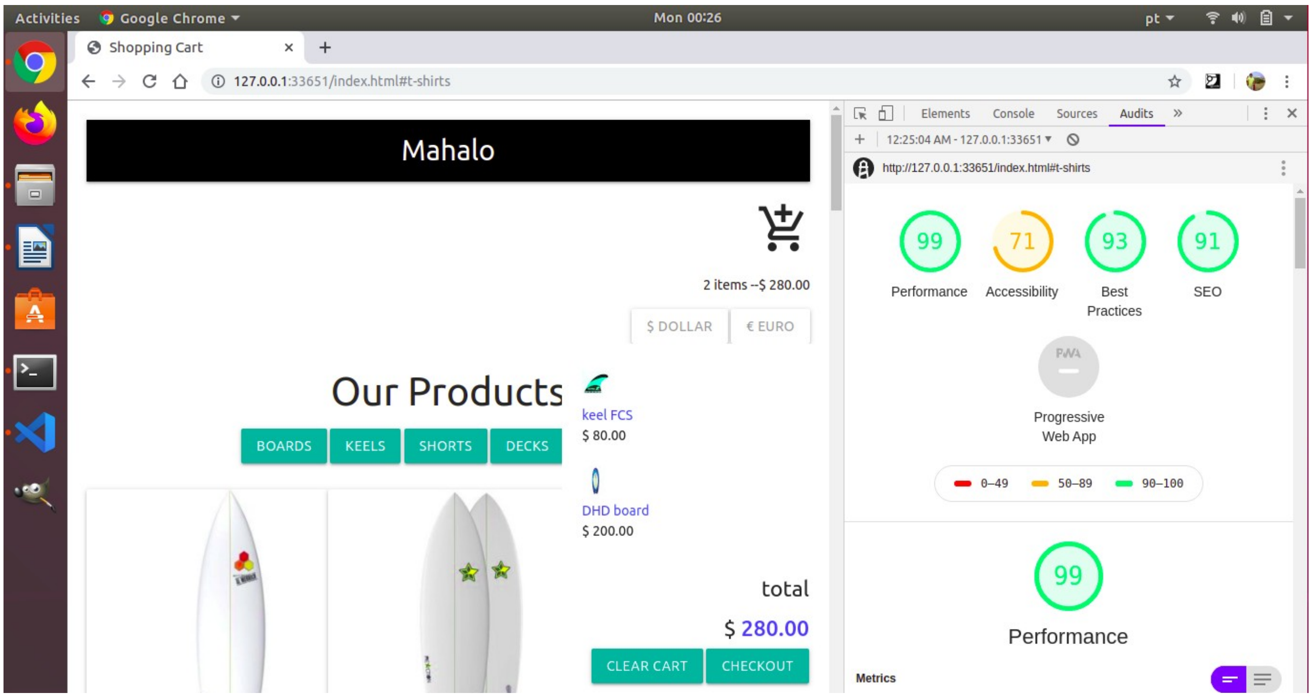
Google Chrome Developer Tools (Audits)

Identify and fix common problems that affect your site's performance, accessibility and user experience

- Performance
- Progressive Web App
- Best practices
- Accessibility
- SEO

Metrics:

First Contentful Paint	1.7 s
Speed Index	1.7 s
Time to Interactive	2.3 s
First Meaningful Paint	2.1 s
First CPU Idle	2.1 s
Max Potential First Input Delay	90 ms



Requirements

•You must use GIT. It is important that the indications and commits are explicit and concrete enough to be able to understand the changes without the need to require additional information as much as possible.

•You cannot use Frameworks or third-party libraries for the JavaScript part

•Document all your algorithms

•Use object oriented programming

•Work properly with prices in Euros € and Dollars $ (you can place the conversion value between currencies manually and simulate a third party service)

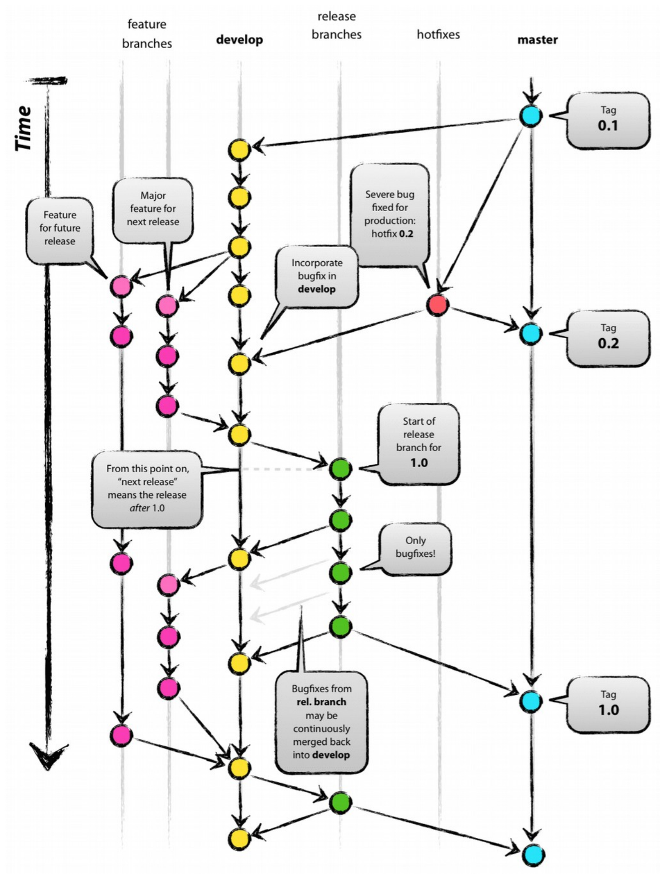
•You must create a page with a static product list with its corresponding action to “Add to Cart”. Products should ideally be stored in JS. (You must iterate them to render them in html)

•The user may perform at least the following actions:

- •Add products to cart
- •Remove products from cart
- •Modify number of units
- •You cannot add more quantities than are available (simulate stock)
- •You can change product options that can influence the price and image shown to the user
  - •For example, if it is a clothing product, these options would be: size, color, ...

Risk documentation
Incidents (disease, technical failure, poor performance, unrealistic deadlines) may change:
+ Statement of work (SOW)
+ Work breakdown structure (WBS)
+ Budget
+ Schedule
+ Execution plan

Git Workflow
Described by Vincent Driessen in 2010.
I used 3 branches:
- Master
- Develop
- Feature



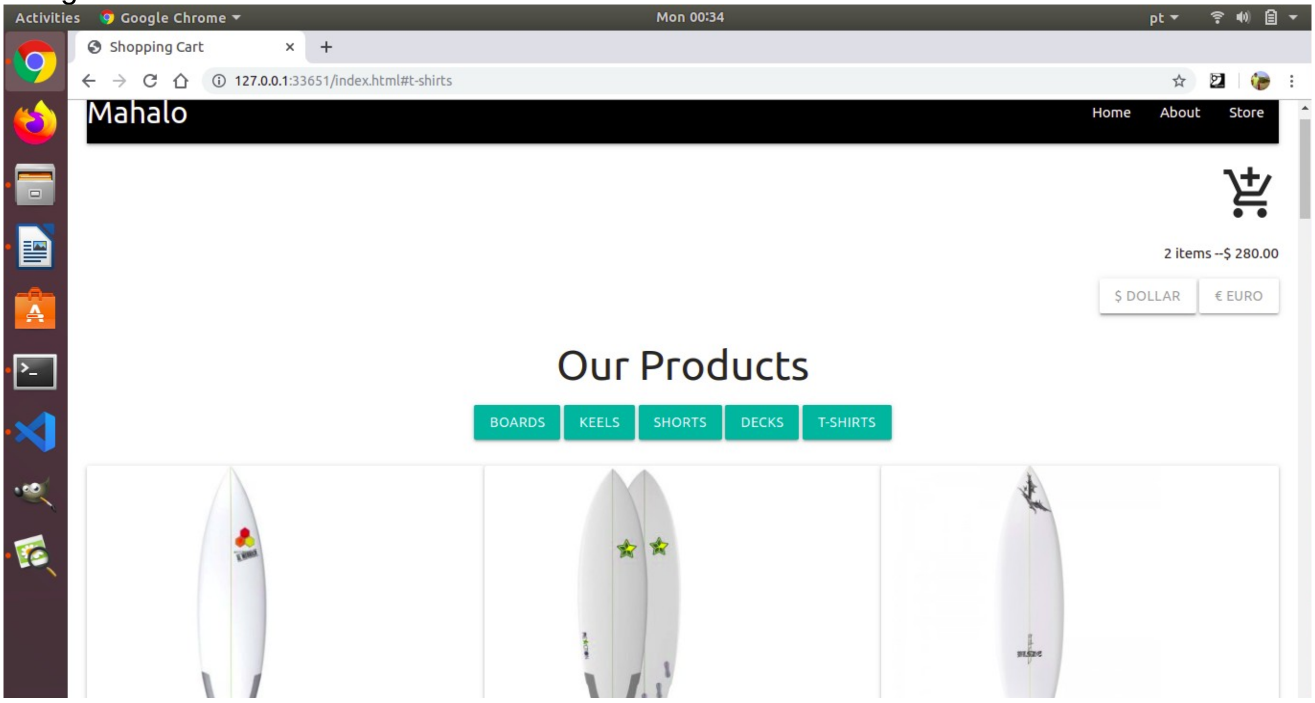
feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

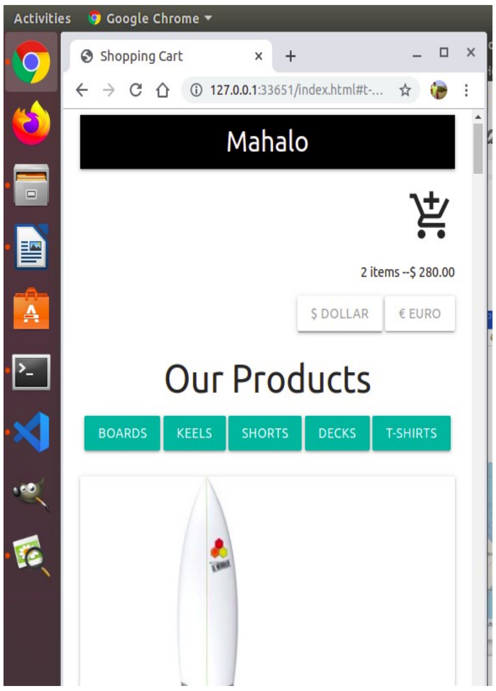
+ “Master” is always ready to be released on LIVE, with everything fully tested and approved (production-ready).

+ “Develop” is the branch to which all feature branches are merged and where all tests are performed. Only when everything’s been thoroughly checked and fixed it can be merged to the Master.

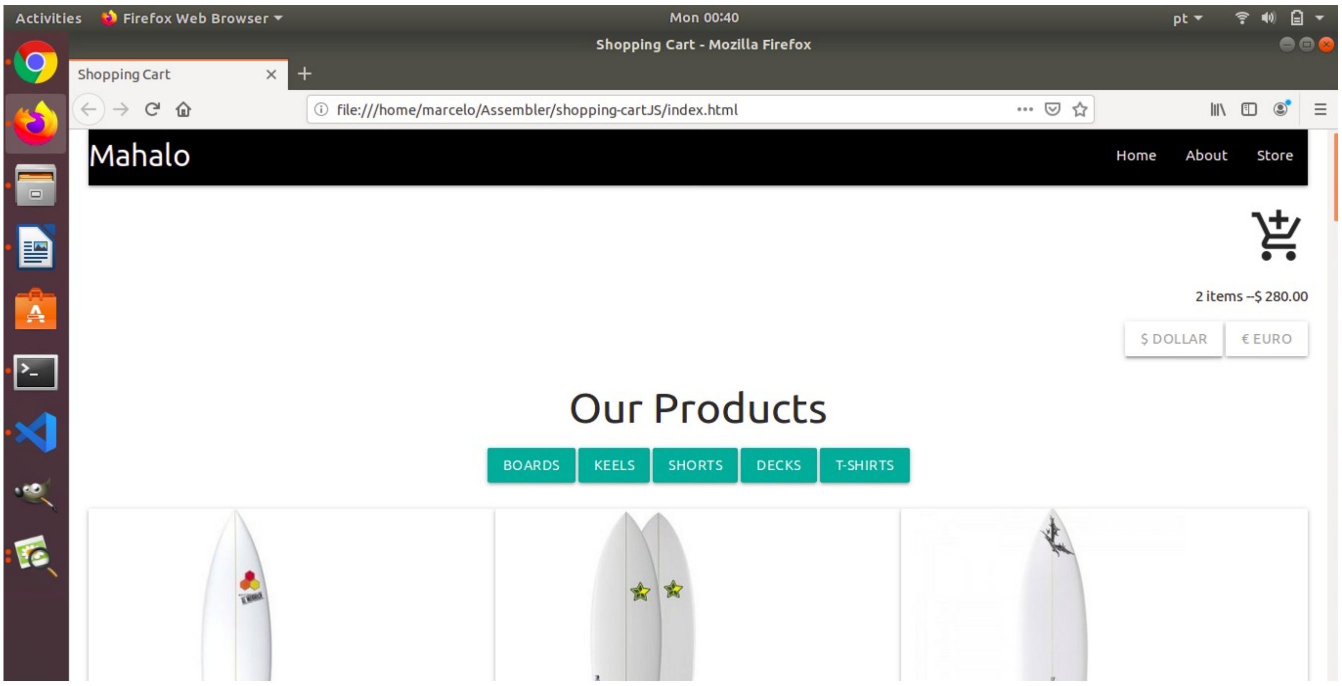
Project Tools
- Visual Studio Code
- Git and GitHub
- Materialize.css
- Google Chrome Developer Tools

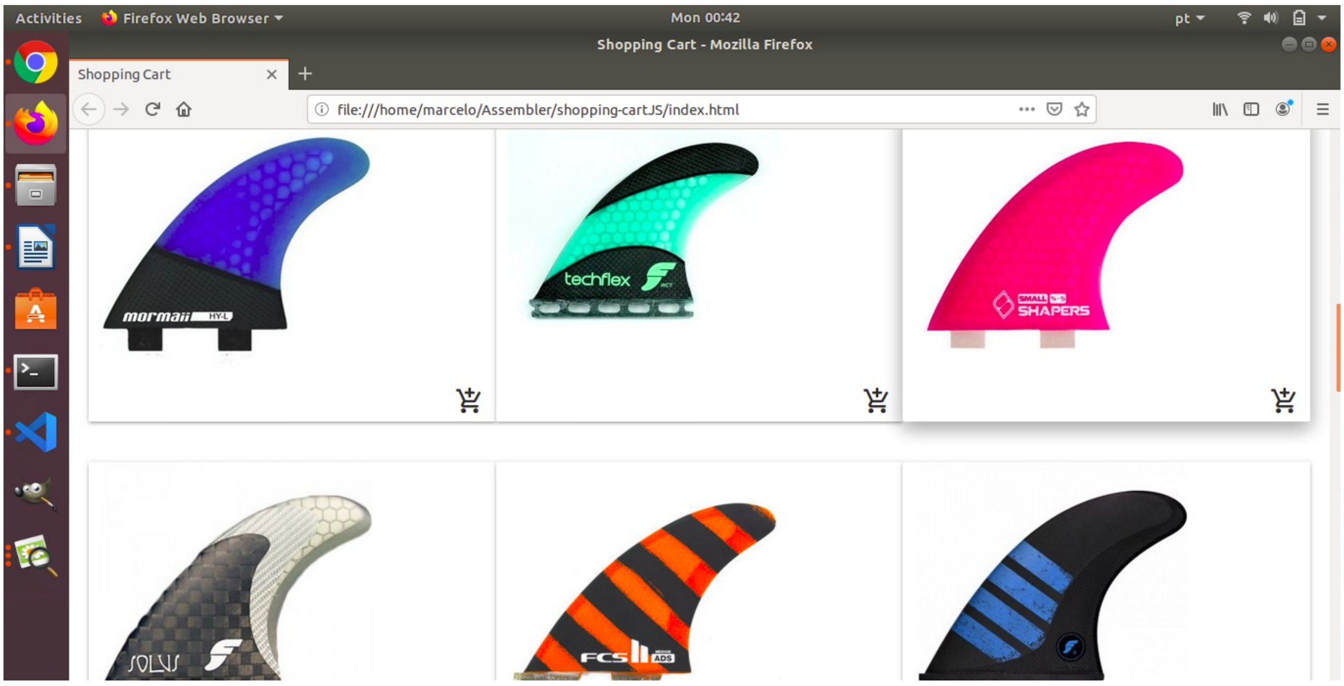
Google Chrome 78.0.3904.87

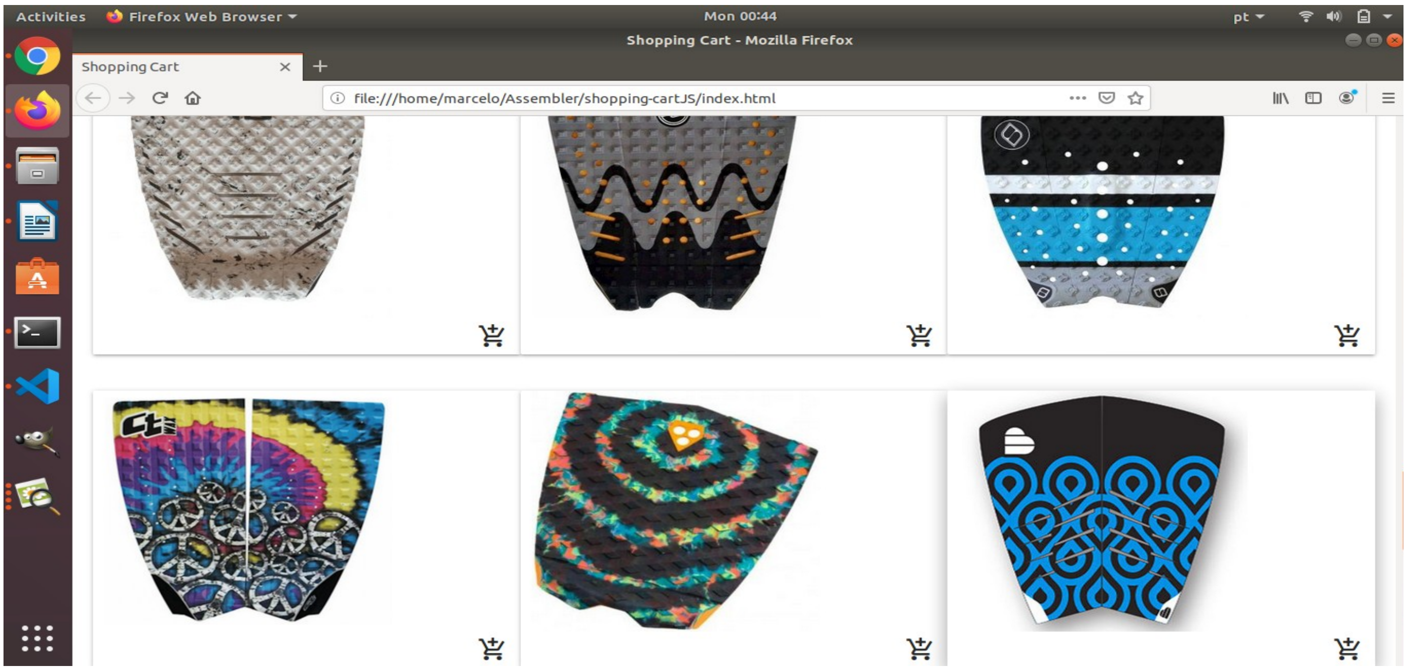




Mozilla Firefox




Activities
Firefox Web Browser
Mon 00:42
pt
Shopping Cart - Mozilla Firefox
Shopping Cart
file:///home/marcelo/Assembler/shopping-cartJS/index.html


Activities
Firefox Web Browser
Mon 00:44
pt
Shopping Cart - Mozilla Firefox
Shopping Cart
file:///home/marcelo/Assembler/shopping-cartJS/index.html

Safari 13.0.3

Resources:

- https://validator.w3.org/
- https://developer.mozilla.org/es/docs/Web/API/Window/localStorage
- https://stackoverflow.com/