

PROJECT – INTRODUCTION TO WEB DEV'

The corrections are automated. Follow the instructions.

Intro

Requirements

To go with HTML/CSS development, you only need:

- A text editor (supporting code highlighting for a better comfort) to edit html and css source files ;
- A browser to run your website.

Goals

- **In part 1 :**
 - Build your digital CV
- **In part 2 :**
 - Better methods to improve your digital CV

Mootse : You will push to Mootse :

- **From part 1 :**
 - all html files, pictures, videos
- **From part 2 :**
 - all html files, pictures, videos, one folder per project.
- All in one zip file. Be careful about media size.

Graduation

You have 100 points to mark all over the work.

Your files will be automatically graduated for some parts, and manually for others parts.

- If a needed file is not found, you will not gain points for your work. In the web world, uppercase IS NOT lowercase.
- If a needed bloc is not found, you will not gain points from items in this "not found bloc".
- Be careful with html tags. Open and close them as needed.
- Like in your future engineer life, if you do not respect some or all technical specifications, the project will fail.

Code Validation (10 points)

You must validate all of your HTML and CSS files thanks to the W3C validators:

- HTML: https://validator.w3.org/#validate_by_upload
- CSS: <https://jigsaw.w3.org/css-validator/>

SANCTIONS WILL BE APPLIED IF ERRORS ARE FOUND BY THOSE VALIDATORS.

READ ALL THIS DOCUMENT BEFORE STARTING.

1 PART1 : BUILD YOUR DIGITAL CV (40 POINTS)

In this first project you will create a small website to present yourself, like a digital CV. The site map is presented in [figure 1](#). The website must contain the 3 pages described later. In part 2, you will discover better methods to develop your website.

The Website can be done in French or in English. You MUST use for this project:

- **HTML5 semantic tags;**
- **CSS Flexbox model.**

For this project, you must think of these two fundamental rules for writing Web pages:

1. Separate the style and the content, the HTML and the CSS;
2. Rely on tables (<table> tag) for the global page's presentation is the best way to get a very bad mark. Tables are for data, not for pages structure. So use Flexbox.

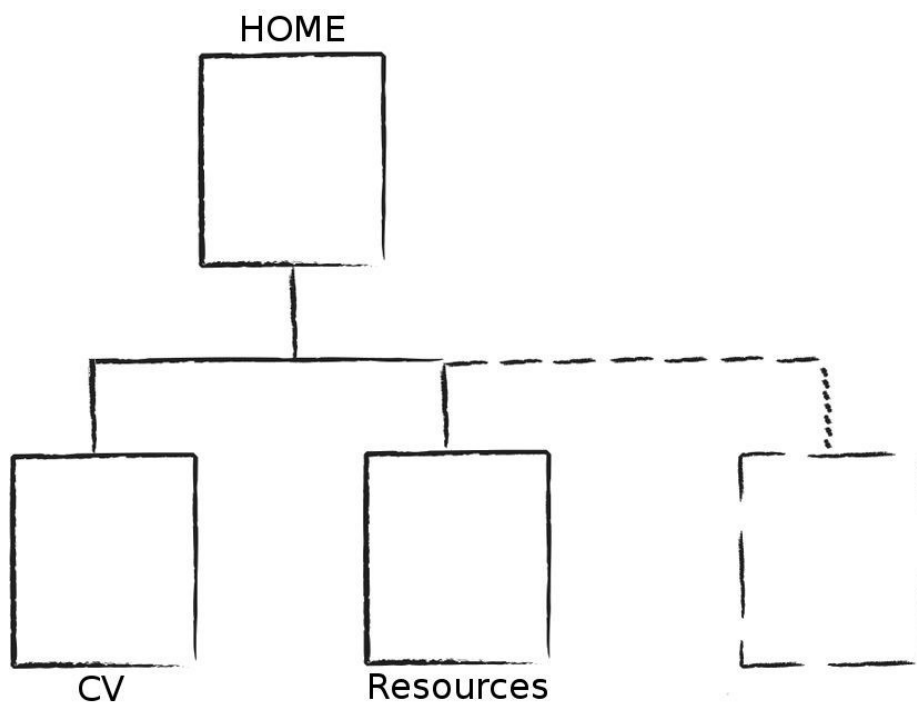


Figure 1 – Site map

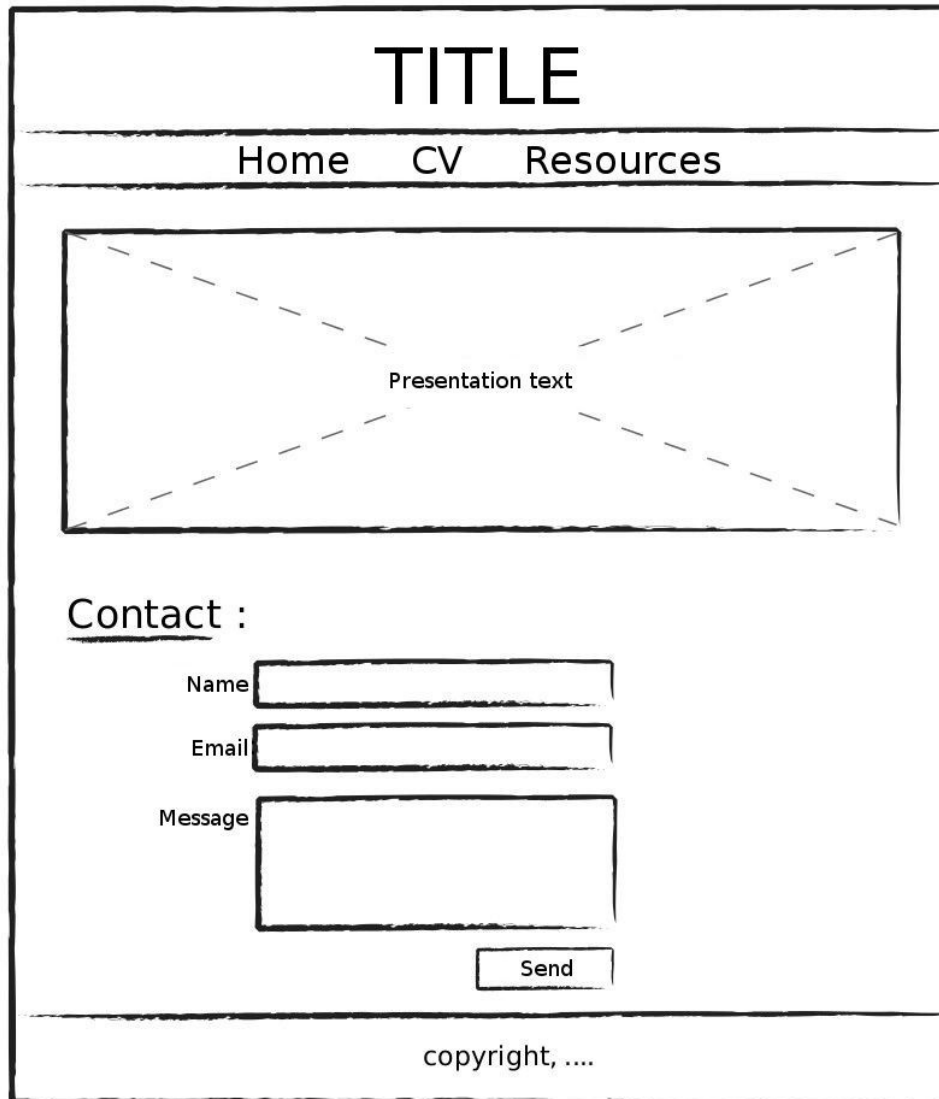
Your website will contain a few pages :

- home.html
- cv.html
- resources.html
- me.html (see part 2)

Continue your reading for all details.

1.1 HOME PAGE (10 POINTS)

See [figure 2](#) for the details of your home.html page.



The diagram illustrates the layout of a home.html page. It is enclosed in a rectangular border. At the top, there is a large title 'TITLE' centered. Below the title is a horizontal navigation menu with three items: 'Home', 'CV', and 'Resources', separated by spaces. Underneath the menu is a large rectangular area with dashed lines forming an 'X' shape, labeled 'Presentation text' in the center. Below this area is a section titled 'Contact :'. This section contains three input fields: 'Name', 'Email', and 'Message', each with a corresponding label to its left. The 'Message' field is a larger text area. Below these fields is a 'Send' button. At the bottom of the page, there is a footer area containing the text 'copyright,'.

Figure 2 – home.html

- Choose a favicon for your site.
- You will choose a title for your website. The size of your title must be increased to make it more visible.
- The horizontal menu below the title must be a list in a <nav> tab with modified CSS to make it appear on the same line.
 - This menu must be centered, and each item must be at 10px from each other.
 - Each element of the list must be a link to page of your website.
 - The menu must have two borders, one above and one below.
- In the main paragraph of this page, present yourself, and the website itself (including a presentation of the other pages). This paragraph must be structured thanks to titles tags <Hn> (n from 1 to 6) (you don't need to use every of them).

- Below this paragraph, you will add a contact form (it is NOT meant to work, only the HTML/CSS part of the form will be developed, but the “Send” button is not supposed to actually send a message). The form will ask for:
 - Name of the user
 - His/her email address
 - A drop down list with a few choices with different reasons to contact you.
 - The message itself
- The footer of the page will contain several information you find useful (Copyright, Last update date, Author, Privacy Notice...). These information must be centered, and the footer must have an upper border. The footer must be present on every page of your website, along with the title and the menu.

1.2 CURRICULUM VITAE PAGE (10 POINTS)

Write in this page your CV. The figure 3 represents what your cv.html page supposed to look like.

TITLE

Home CV Resources

Picture

Civil State

Education

- Minimum multo itaque adiumentique caritatis.
- Et atque quem causam semper.
- Latasque suis superbas venerabilis velut.

Skills

- Minimum multo itaque adiumentique caritatis.
- Et atque quem causam semper.
- Latasque suis superbas venerabilis velut.

Hobbies

- Minimum multo itaque adiumentique caritatis.
- Et atque quem causam semper.
- Latasque suis superbas venerabilis velut.

copyright,

Figure 3 – cv.html

- Your picture must have a border large enough to make it highlighted. The thumbnail will be a link to a bigger picture with better quality.
- The titles (Education, Skills, Hobbies) must have its own color and a border below as wide as the page.
- The elements of the lists need to be in a different color than the civil state and the titles.

1.3 RESOURCES PAGE(10 POINTS)

Your resources.html page will be composed by these elements, according to [figure 4](#) :

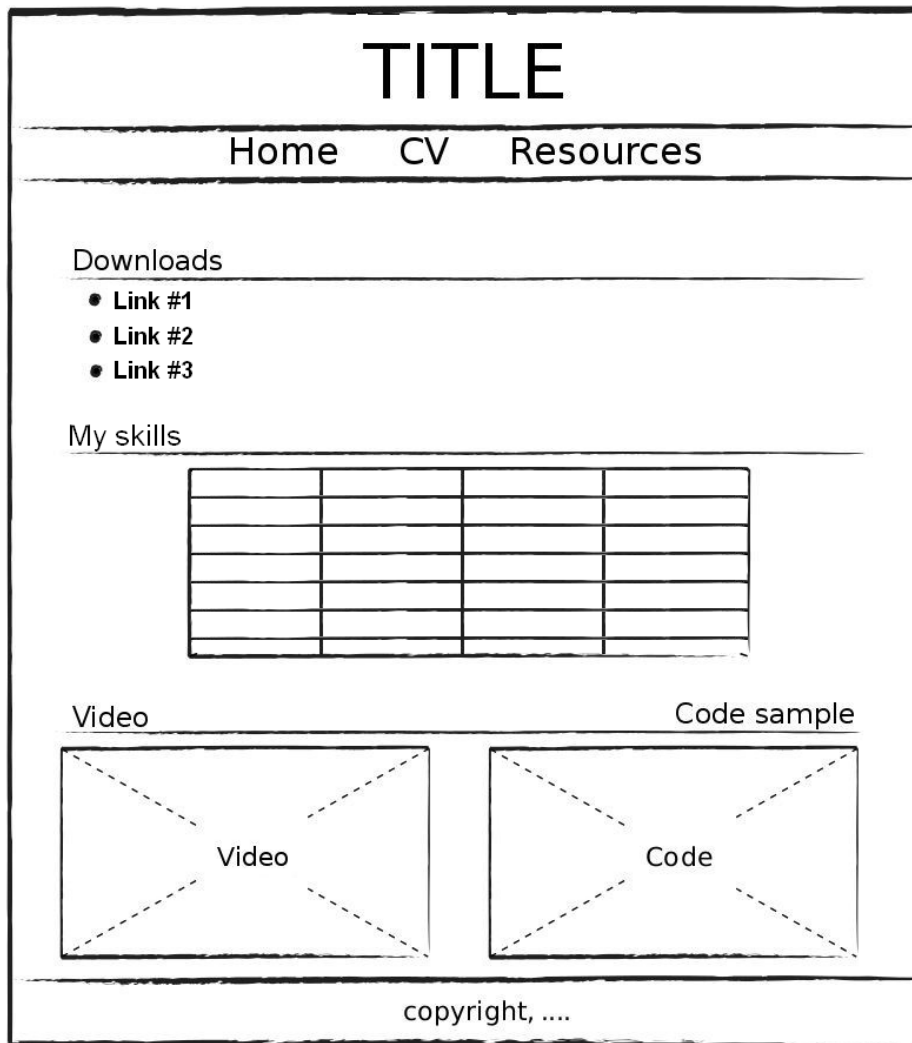


Figure 4 – resources.html

As the previous page (cv.html), you will use titles to separate the different sections:

- Downloads: 3 links must allow to download:
 - A little zip file, local
 - A little pdf document, distant (don't worry about browser's default behavior)
 - The local picture used in the CV, local.
- A skill chart with table tags here :
 - First line with 4 bold titles : Skill name – Level – Project exemple – Idea to improve
 - Others lines = 4 at least : skill name, level (with dots, checkmarks, anything else), project label and idea (to improve the skill or the project).
 - Three colors : 1 for title line, 2 others for the content, in an alternate way.
- Video: You will insert a small local video (not a link, the full player must be displayed), using the html5 tags `<video>`. Be careful : it must be **small enough** to fit in Mootse's hosting system.
- Code sample: an indented and formatted sample of code (C, C++, Java... as you want!), using the `<pre>` tag. Add a little paragraph explaining why you are proud of this code.
 - NB : do not use html code here, to avoid false error in validation tools.

1.4 FUN TIME : You (10 POINTS)

Add another me.html page in the nav bar with things defining you. About the look ? As you wish.

- your today top 3 memes ;
- your best music band and your top song, with a Youtube integrated video player ;
- your best book, with a picture and a few words about it ;
- your best movie/serie, with a Youtube integrated video player and a few words about it ;
- your best cooking recipe.

2 BETTER METHODS (50 POINTS)

Now you will enhance the foundation of your website.

1 folder for each project, then 1 big zip for all to Mootse : be careful about all media size !

2.1 PROJECT 2: CSS FRAMEWORK (20 POINTS)

Discover and use **Bootstrap**, using the last getting started tutorial :

<https://getbootstrap.com/docs/>

Clone your first project and replace all Flexbox with Bootstrap.

You will then look at layout, component (like buttons, navbar, list, progress bar, toast, tooltip) and form sections. This second website will be similar with the first.

2.2 PROJECT 3: IFRAME TAG (10 POINTS)

From your first or second project, rethink your website in big and autonomous thematic blocs.

Then re segment your first project in multiple **html files**.

Finally, try to use **iframe tag** in your files in order to rebuild your website, by including your different blocs in each others.

2.3 PROJECT 4: OBJECT TAG (10 POINTS)

From your first or second project, rethink your website in big and autonomous thematic blocs.

Then re segment your first project in multiple **html files**.

Finally, try to use **object tag** in your files in order to rebuild your website, by including your different blocs in each others.

2.4 PROJECT 5: PHP (10 POINTS)

Try and discover a WAMP (Windows) / LAMP (Linux) / MAMP (MacOS) like environment, with functioning PHP.

From your first or second project, rethink your website in big and autonomous thematic blocs.

Then re segment your first project in multiple **php files**.

Finally, try to use **php tag** in your files in order to rebuild your website, by including your different blocs in each others.

2.5 THE END

Next steps will be Javascript, more Php, and others web languages. But for now, congratulations !