



PROJET

CODIT

2<sup>ÈME</sup> BACHELIER EN INFORMATIQUE

---

## Gestion de projets

---

*Auteur :*

Alexandre DUCOBU

*Enseignants :*

Erwin DESMET

Jean-Sébastien LERAT

Yoan PIETRZAK



Année académique 2016 - 2017





HAUTE ECOLE  
EN HAINAUT

PROJET

CODIT

2<sup>ÈME</sup> BACHELIER EN INFORMATIQUE

---

## Gestion de projets

---

*Auteur :*

Alexandre DUCOBU

*Enseignants :*

Erwin DESMET

Jean-Sébastien LERAT

Yoan PIETRZAK



Année académique 2016 - 2017



Ce document est mis à disposition selon les termes de la licence Creative Commons  
“Attribution - Pas d'utilisation commerciale 4.0 International”.



# Table des matières

1	Présentation du projet . . . . .	2
1.1	Introduction . . . . .	2
1.2	But . . . . .	2
1.3	Choix du projet . . . . .	2
2	État des lieux . . . . .	3
2.1	Étapes de création . . . . .	3
2.2	Choix des technologies . . . . .	4
3	Diagrammes . . . . .	5
3.1	Gantt . . . . .	6
3.2	PERT . . . . .	8
4	Base de données . . . . .	10
4.1	Création de la base de données . . . . .	10
4.2	Connexion à la base de données . . . . .	11
4.3	Sécurité . . . . .	11
5	Conclusion . . . . .	12
	Références . . . . .	13
	<b>Annexes</b>	<b>14</b>

# 1 Présentation du projet

## 1.1 Introduction

Dans le cadre du cours de **Gestion de projets**, il nous a été demandé de réaliser un projet au choix individuellement ou par deux. J'ai choisi de le faire seul.

En effet, nous avons déjà d'autres travaux de groupes. Je trouve donc qu'un travail individuel est un plus dans notre cursus scolaire.

Lors des deux premières séances de laboratoire, chaque groupe a rédigé une fiche descriptive du projet, avec l'enseignant, afin de baliser le travail à effectuer durant l'année.

Lors de ces séances, l'enseignant a validé chacun des projets.

## 1.2 But

Grâce à ce projet, nous allons apprendre à gérer nos projets à l'aide de différents outils spécialisés tels que **Microsoft Project**, **le diagramme de Gantt** et **le graphique de PERT**.

Ceux-ci nous aideront dans la planification de notre projet ainsi que, pour les binômes, dans la répartition des tâches.

## 1.3 Choix du projet

Ce projet, *un site web*, permettra d'apprendre les bases de la programmation en *Python* et sera divisé en chapitres : les variables, les conditions, les boucles, les tableaux, etc.

L'apprentissage se fera en trois étapes :

1. L'utilisateur découvrira le nouveau sujet par de la théorie ainsi que par un ou plusieurs exemples. Il en apprendra alors l'utilité et le fonctionnement.
2. Entre deux parties théoriques, l'utilisateur mettra en pratique ce qu'il aura appris au travers de petits QCM.
3. Une fois le chapitre terminé, un questionnaire (QCM, ordonnancement du code,...) sera proposé à l'utilisateur.

Celui-ci sera noté sur 10 afin que l'utilisateur puisse se juger et s'améliorer.

Le passage au chapitre suivant requerra une *cote minimale de 7/10*.

D'autre part, ce projet sera un pré-TFE.

À terme, il sera possible de créer facilement des cours et de s'y inscrire.

Il pourra être utilisé aussi bien par les écoles que par « *les particuliers* ».

## 2 État des lieux

### 2.1 Étapes de création

Lors de la phase de préparation du site, j'ai débuté par la création de maquettes (*voir Les maquettes*) afin de partir sur une idée concrète du projet.

Je me suis ensuite engagé dans la phase d'écriture des chapitres et des questionnaires.

Le premier jet comptait trois chapitres (*les variables, les booléens et la complexité*), dont le dernier était trop complexe pour débuter dans la programmation.

Il est resté à son état d'origine, et n'est donc pas disponible dans le site.

Après ces deux phases, est venue celle de la création du squelette du site en HTML et CSS, suivie de *très* près par l'ajout des effets en JavaScript.

À ce moment, le projet a été mis en pause dans le but :

1. de concevoir et de préparer la présentation du projet ;
2. d'étudier pendant le pré-blocus de Pâques ;
3. d'avancer dans les autres projets, dont celui du cours d'*Initiation aux nanordinateurs* qui devait être achevé pour le 12 mai.

La phase suivante a été l'une des plus importantes du projet : c'est la liaison du site à la base de données.

En effet, le site est désormais dynamique :

- L'accès aux chapitres n'est plus permis que s'ils ont été débloqués dans la base de données ;
- Les résultats obtenus à la fin des chapitres, et disponibles dans les trophées, proviennent aussi de la base de données.

La dernière phase - *la phase de finalisation* - consiste à la réécriture des chapitres existants (*les variables et les booléens*) ainsi qu'à l'écriture des autres chapitres (*l'introduction, les conditions et les boucles*) et à l'ajout des différents questionnaires.

Bien entendu, tout au long de cette période, le site a été testé de manière à ce qu'il soit le plus stable possible.

## 2.2 Choix des technologies

Vu la nature du projet, *un site web dynamique*<sup>1</sup>, j'ai utilisé les langages suivants :

- le HTML et le CSS pour la création du site ;
- le JavaScript (*et le jQuery*) pour l'ajout d'animations et d'effets ;
- le PHP et le MySQL pour la connexion au site et l'accès à la base de données.

Pendant les premières phases du projet, l'éventualité d'utiliser *Bootstrap* et le framework PHP *Laravel* est apparue.

**Bootstrap** est une collection d'outils utile à la création du design de sites et d'applications web [*sic*]<sup>2</sup>.

Ces outils - *composés de codes HTML et CSS* - permettent d'utiliser facilement des formulaires, des boutons, etc.

Ils facilitent aussi la conception de sites web adaptatifs grâce à la décomposition du site en colonnes.

Un framework évite de réinventer la roue en réutilisant ce qui a déjà été fait, utilisé et validé par de nombreux utilisateurs.

Cela représente, donc, un gain de temps, de fiabilité et une facilité de mise à jour. **Laravel** regroupe aussi de nombreuses bibliothèques qui facilitent la gestion des sessions, l'authentification, un créateur de requêtes SQL,...

Mais il y a un mais : pour les utiliser, il faut se documenter, apprendre et expérimenter, ce qui représente beaucoup de temps. Trop par rapport à celui que je possédais.

Je ne les ai donc pas utilisés, mais j'ajouterai que la non utilisation de ces outils permet plus de libertés et de personnalisation du site.

---

1. Un site dynamique est un site dont les données proviennent d'une base de données.

2. [https://fr.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

### 3 Diagrammes

Afin de gérer le projet, j'ai utilisé plusieurs logiciels.

Pour la création du diagramme de **Gantt**, j'ai d'abord employé *GanttProject* qui ne m'a pas permis d'avoir deux tâches décallées en parallèle. Je me suis alors tourné vers *Microsoft Project* qui, lui, m'a donné satisfaction.

Quant à celui de **PERT**, je l'ai conçu *à la main*, car aucun outil ne proposait de concevoir le diagramme selon les instructions données dans le cours.

### 3.1 Gantt

Voici le diagramme de Gantt du projet et son tableau descriptif :

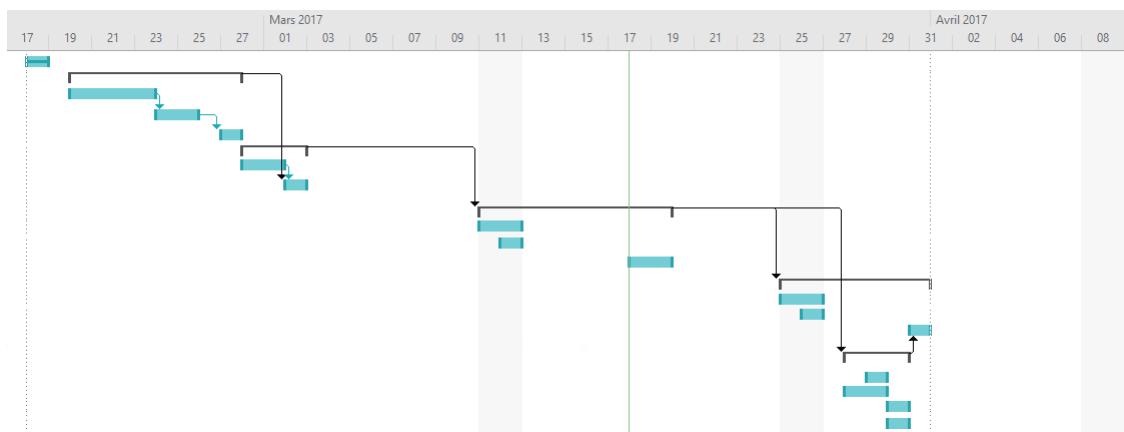


FIGURE 1 – Diagramme de Gantt

N°	Nom de la tâche	Durée (jours)	Prédécesseurs
1	Création du diagramme	1	/
2	Création du cours	8	/
3	Maquette	3	2
4	Ossature du site (HTML/CSS)	9	3
5	Ajout du JS	7	4
6	Ajout du PHP	3	4, 5

FIGURE 2 – Tableau descriptif

Comme on peut le voir, la durée totale prévue du projet est de 31 jours.

Le projet est découpé en 6 tâches principales détaillées ci-dessous.

La création du diagramme a duré moins d'un jour, mais c'était la durée minimale disponible dans *Microsoft Project*...

La création du cours comprend l'écriture des différents chapitres et questionnaires. Comme dit précédemment, celle-ci s'est déroulée en deux étapes :

1. L'écriture d'un premier jet de trois chapitres : les variables, les booléens et la complexité, et... de quelques exercices.
2. La réécriture des chapitres sur les variables et les booléens, et l'écriture des autres chapitres (*l'introduction, les conditions et les boucles*). Chacun accompagné de son questionnaire principal et de ses questionnaires secondaires. Le questionnaire principal teste, en quatre questions, les connaissances acquises tout au long du chapitre, tandis que les secondaires testent celles de la page précédente en deux questions.

Cinq maquettes ont été créées :

1. La page d'accueil, très (*trop*) simple et sobre ;
2. La page du cours comprenant les liens vers les chapitres ;
3. L'aperçu d'un chapitre. C'est la raison pour laquelle les maquettes dépendent de la tâche précédente ;
4. La page des trophées ;
5. La page du compte de l'utilisateur.

Dans la version finale, on peut remarquer que le design a évolué. Comme lors d'une démonstration, celui-ci a été jugé « *trop simple* » j'en ai profité pour ajouter des **animations** sur la page d'accueil, sur les chapitres ainsi que sur les trophées, et de la **coloration syntaxique** du code.

La création de l'ossature du site fut la partie la plus longue parce que la majorité du code du site a été écrite à ce moment.

Ainsi, toutes les pages (*accueil, cours, chapitres, trophées, etc.*), styles et effets CSS ont été créés et n'ont plus été changés depuis.

L'ajout du JavaScript est autant lié au code HTML qu'à celui en PHP : il est utilisé dans tout ce qui est animations et boutons.

Par exemple, les boutons utilisés pour modifier les données de l'utilisateur ou pour supprimer le compte sont écrits en HTML et CSS. L'action est exécutée en JavaScript et passe par le PHP pour se connecter à la base de données.

Pour terminer, l'ajout du PHP a été un gros morceau : le fait de mixer les codes HTML, PHP et MySQL n'a pas été simple.

En effet, comme c'est mon premier projet, *en parallèle avec celui du cours de PHP*, à mettre en œuvre un site connecté à une base de données, cela m'a demandé pas mal de travail pour arriver au bon résultat.

De plus, sans le PHP, il n'y aurait pas de connexion, d'inscription ou de déblocage des chapitres. Tout cela est directement lié à la base de données, elle-même liée au site par le PHP.

### 3.2 PERT

Après avoir travaillé le diagramme de Gantt, j'ai réalisé le diagramme de PERT. Ci-dessous, vous pouvez le voir dans sa forme initiale :

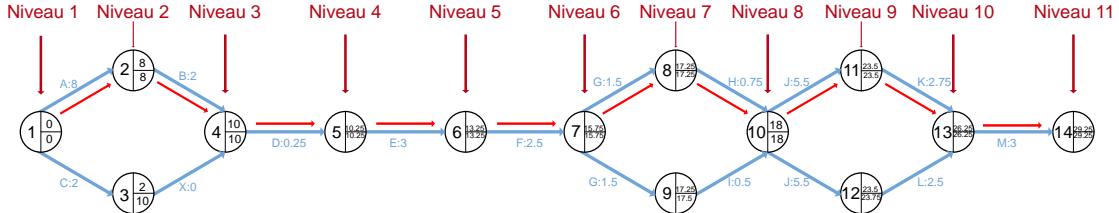


FIGURE 3 – Diagramme de PERT initial

Après que le projet soit terminé, j'ai refait le diagramme de PERT afin de coller au plus près au temps passé sur les différentes tâches :

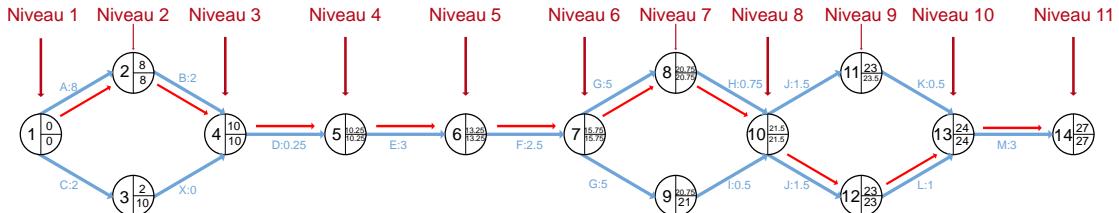


FIGURE 4 – Diagramme de PERT final

La première chose que l'on peut remarquer, c'est qu'il a fallu moins de temps que prévu pour terminer le projet (*on passe de 29h15 à 27h*).

Dans ce diagramme, les tâches sont plus nombreuses pour plus de précision, et leur durée est plus juste. On parle désormais en heures et plus en jours.

Par exemple, on peut voir que l'écriture des chapitres dure 8 heures et que celle des exercices en dépend.

De même pour la création de la maquette qui ne dépend plus entièrement des tâches ci-dessus. C'est la tâche **D**, le test de la maquette avec le texte, qui dépend de l'existence du cours, et elle ne dure qu'un quart d'heure.

Voici le tableau des antériorités lié au diagramme final :

Tâches	Description des tâches	Tâches antérieures	Durée (heures)
A	Écriture des textes théoriques	/	8
B	Écriture des exercices	A	2
C	Création de la maquette	/	2
D	Test de la maquette avec texte	A, B, C	0.25
E	Forme générale (et style) des pages (header, navbar, body)	D	3
F	Création des différentes parties/pages (div,...)	E	2.5
G	Ajout des textes	F	5
H	Animation de l'affichage du cours	G	0.75
I	Animation de l'affichage des trophées	G	1.5
J	Connexion et accès au cours	H, I	1.5
K	Affichage des trophées depuis la base de données	J	0.5
L	Ajout et suppression d'un utilisateur	J	1
M	Animations et changements d'états liés au PHP	K, L	3

FIGURE 5 – Tableau des antériorités

## Différences entre l'initial et le final

Jusqu'à la tâche **F**, *niveau 7*, il n'y a pas de différences.

Après celle-ci, quatre des sept tâches ont eu des durées différentes de ce que j'avais escompté :

- la tâche **G** est passée de **1h30** à **5h**. La cause est simple, la mise en page des chapitres a été plus complexe que prévu. En cause, l'ajout de coloration syntaxique, l'ajout de styles pour les types (*gras*, *italique*, etc.), ainsi que l'ajout d'un slider<sup>3</sup> divisant les chapitres en différentes parties, etc.
- la connexion à la base de données, tâche **J**, a pris moins de temps que prévu. **Remarquons**, tout de même, que de nombreux changements à ce niveau ont été effectués à la tâche **M**, dont la raison d'être était de résoudre les problèmes potentiels.
- l'affichage du résultat dans les trophées *dépends de la base de données*, la tâche **K**, a été beaucoup moins difficile qu'imaginé grâce à l'avancée du projet du cours de *Programmation web*.
- il en est de même pour la tâche **L**, *l'ajout et la suppression d'utilisateurs*, a donc pris moins de temps.

3. <http://callmenick.com/post/responsive-content-slider>

## 4 Base de données

### 4.1 Crédation de la base de données

La base de données est composée de trois tables contenant toutes les données utiles au bon fonctionnement du site.

Il y a la table **Users** pour tout ce qui concerne les utilisateurs, la table **Results**, qui concerne les résultats de chaque chapitre et la table **Tests** qui contient la liste des résultats.

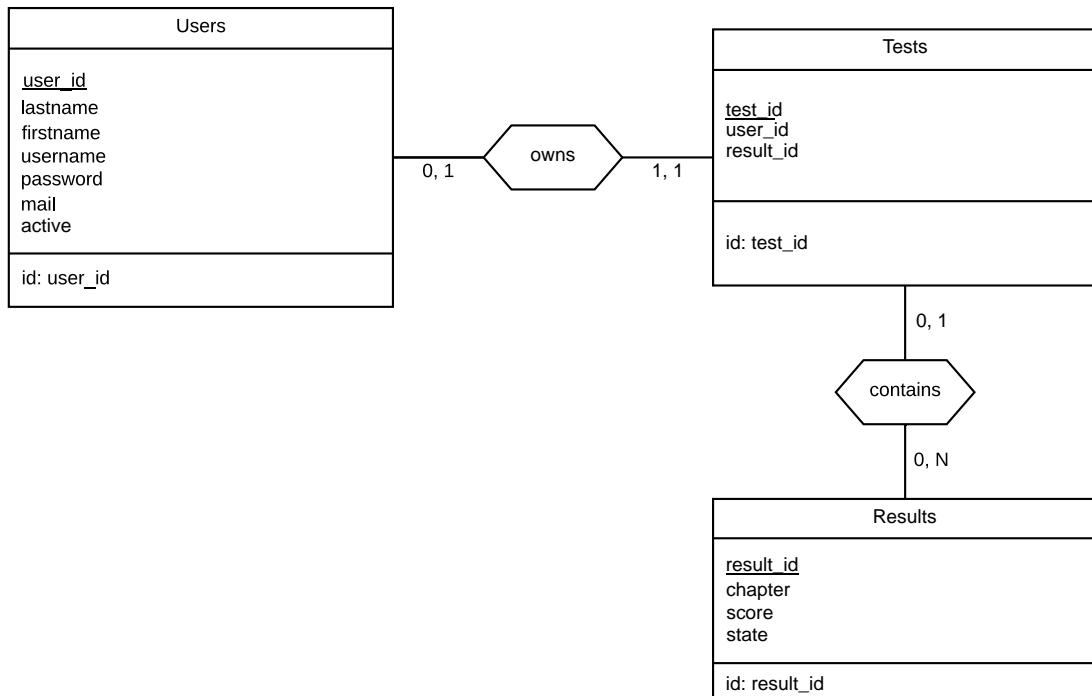


FIGURE 6 – Schéma conceptuel de la base de données

## 4.2 Connexion à la base de données

Le type de requêtes utilisé pour l'accès à la base de données est le type PDO. De manière plus précise, ce sont des requêtes préparées qui ont été utilisées pour plus de sécurité (*voir la section 4.3 Sécurité*). De plus, PDO permet plus de flexibilité. Elle est recommandée, très utilisée (*et possède donc une importante communauté*) et est enseignée à l'école.

On pourrait alors ajouter, dans les évolutions possibles, le passage à la *programmation orientée objet*, couplée à l'utilisation du *singleton* et, comme structure de site, l'utilisation du *design pattern MVC*.

## 4.3 Sécurité

Pour la sécurité du site, tout accès à la base de données passe par des requêtes préparées (*PDO*), car celles-ci le protègent de toute injection SQL.

La connexion au site est surveillée grâce à la mise en place d'une session. Dès qu'un accès à une page se fait, la connexion est vérifiée. Si l'utilisateur n'est pas identifié, il est redirigé vers la page d'accueil.

En plus de cela, chaque champ de formulaire est protégé par des *regex*<sup>4</sup>. Celles-ci vérifient que les données entrées sont de la forme voulue : un nom ne doit contenir que des lettres et, selon une certaine forme, peut être divisé à l'aide d'un trait d'union ; une adresse mail ne contient que des lettres minuscules non-accentuées, des points et un seul arobase ; etc.

De plus, les mots de passe ont une longueur minimale de quatre caractères et doivent contenir, au moins, une majuscule, une minuscule et un chiffre.

Ils sont aussi *hachés*<sup>5</sup> à l'aide de *SHA512* et d'un *salage dynamique* pour plus de sécurité.

En effet, le mot de passe est de la forme `+%# longueur mot_de_passe **$`. Par exemple, le mot de passe **Test1** devient `+%#5Test1**$` avant d'être haché. Il est ainsi indéchiffrable, mais cela a aussi pour effet de sécuriser la base de données puisque le résultat ne contiendra que des caractères sûrs.

---

4. Expression régulière (en anglais: **Regular expression**)

5. [https://fr.wikipedia.org/wiki/Fonction\\_de\\_hachage\\_cryptographique](https://fr.wikipedia.org/wiki/Fonction_de_hachage_cryptographique)

## 5 Conclusion

Le projet que j'ai choisi est un site qui permet d'apprendre les bases de la programmation en Python.

Un utilisateur, identifié par son nom d'utilisateur, peut modifier ses données depuis la page *Compte*.

Il peut ajouter son nom de famille, son prénom et son adresse mail, et même modifier son mot de passe.

Il a aussi la possibilité de supprimer son compte.

Pour l'instant, cinq chapitres sont disponibles : *l'introduction, les variables, les booléens, les conditions et les boucles*.

Le premier est débloqué d'office et, pour débloquer le suivant, il faut réussir le test de fin de chapitre avec une cote minimale de 7/10.

En cas d'échec, il faudra revoir le chapitre et repasser le test.

L'utilisateur peut retrouver le résultat de chaque chapitre sur la page des trophées.

L'acquisition de trophées donne une meilleure dynamique à l'apprentissage de la programmation et permet d'être certain de la bonne compréhension de chaque chapitre avant de passer au suivant.

Il est à remarquer que, d'après les bonnes pratiques des bases de données, il ne faut pas supprimer les données mais les rendre invisibles aux utilisateurs.

Cette méthode est utilisée lors de la suppression d'un compte qui n'est pas supprimé, mais simplement désactivé.

Le projet a été divisé en différentes tâches grâce à l'utilisation des diagrammes de Gant et de PERT. Ce qui offre plus de clarté sur le développement du projet.

Lorsque tous les trophées ont été obtenus, l'utilisateur a appris la programmation en Python. L'objectif de ce projet est donc atteint.

Mon projet n'est pas que théorique. En effet, il a aussi un but pédagogique.

Il m'a permis d'apprendre et d'appliquer les cours reçus, et donc à en voir l'utilité.

# Bibliographie

---

## Livres

- [1] HERBY, C. Apprenez à programmer en JAVA.  
(consulté du 07 mars au 15 mai 2017).
  - [2] L<sup>A</sup>T<sub>E</sub>X. A document preparation system.  
(consulté du 07 mars au 15 mai 2017).
  - [3] MALAISE, A. Techniques informatiques 1.  
(consulté du 07 mars au 15 mai 2017).
  - [4] MALAISE, A. Techniques informatiques 2.  
(consulté du 07 mars au 15 mai 2017).
  - [5] NEBRA, M. Apprenez à programmer en C.  
(consulté du 07 mars au 15 mai 2017).
  - [6] PIETRZAK, Y. ET ARNAUD, D. Montages et réalisations.  
(consulté du 07 mars au 15 mai 2017).
  - [7] ROHAUT, S. Algorithmique- techniques fondamentales de programmation.  
(consulté du 07 mars au 15 mai 2017).
- 

## Internet

- [8] DUPRÉ, X. Apprendre la programmation avec Python, *informatique*, Site, [en ligne]. <http://www.xavierdupre.fr>, (consulté le 07 Mars 2017).
- [9] GOFF, V. L. Apprenez à programmer en Python, *informatique*, Site, [en ligne]. <https://openclassrooms.com>, (consulté le 07 Mars 2017).
- [10] SALLOUM, N. Responsive Content Slider, *informatique*, Site, [en ligne]. <http://callmenick.com>, (consulté le 07 Mars 2017).
- [11] THE PYTHON SOFTWARE FOUNDATION. TimeComplexity, *informatique*, Site, [en ligne]. <https://wiki.python.org>, (consulté le 07 Mars 2017).
- [12] THE WIKIPEDIA COMMUNITY. Bootstrap (framework), *informatique*, Site, [en ligne]. <https://fr.wikipedia.org>, (consulté le 07 Mars 2017).
- [13] THE WIKIPEDIA COMMUNITY. Laravel, *informatique*, Site, [en ligne]. <https://fr.wikipedia.org>, (consulté le 07 Mars 2017).

# Table des annexes

<b>Annexes</b>	<b>14</b>
Les maquettes . . . . .	15
Le site . . . . .	18

## Les maquettes

Voici les différentes maquettes du site :

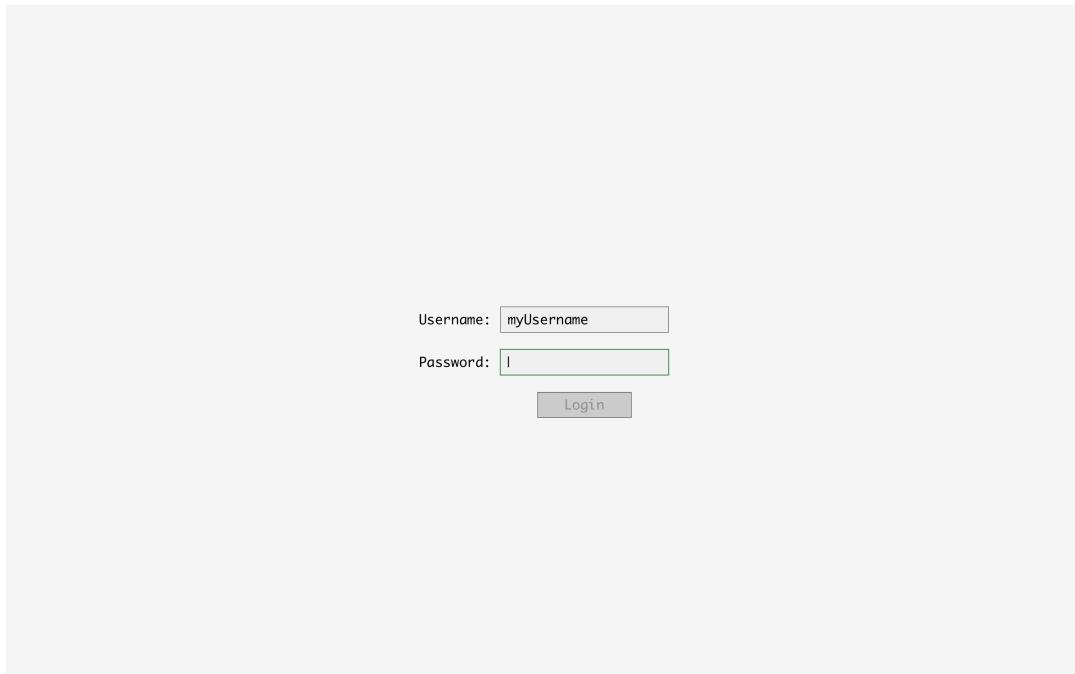


FIGURE 7 – La page d'accueil

A screenshot of a course page. At the top, there is a navigation bar with four items: 'Cours' (highlighted in yellow), 'Trophées' (orange), 'Compte' (red), and 'Déconnexion' (red). Below the navigation bar, the text 'Apprenons le Python !' is displayed next to a green Python logo. There are six rectangular cards arranged in two rows of three. The top row contains cards for 'INTRODUCTION' (locked), 'LES VARIABLES' (locked), and 'LES BOOLÉENS' (locked). The bottom row contains cards for 'LES CONDITIONS' (locked), followed by two cards both showing an ellipsis (...).

FIGURE 8 – La page de cours

Cours      Trophées      Compte      Déconnexion

## Les variables

Une **valeur** est un des éléments de base d'un programme, comme un nombre ou une lettre.  
Exemple: 1, 2 ou 'Bonjour'

Toute valeur possède un **type** particulier.

- 'Bonjour' est une chaîne de caractères (**string**), identifiable grâce aux apostrophes: str
- 1 et 2 sont des entiers (**integer**): int
- 1.4 et 2.0 sont des réels (nombres à virgule flottante, **floating-point**): float

**RQ:** Comme tout en informatique, les types et mots-clés du Python sont en anglais.

L'interpréteur peut donner le type d'une valeur:

```
>>> type(4)
<type 'int'>
>>> type('Bonjour')
<type 'str'>
```

FIGURE 9 – L'aperçu d'un chapitre

Cours      Trophées      Compte      Déconnexion

COURS

INTRO  
8/10

1    2

3    4    5

FIGURE 10 – La page des trophées



FIGURE 11 – La page du compte

# Le site

Voici les différentes pages du site finalisé :

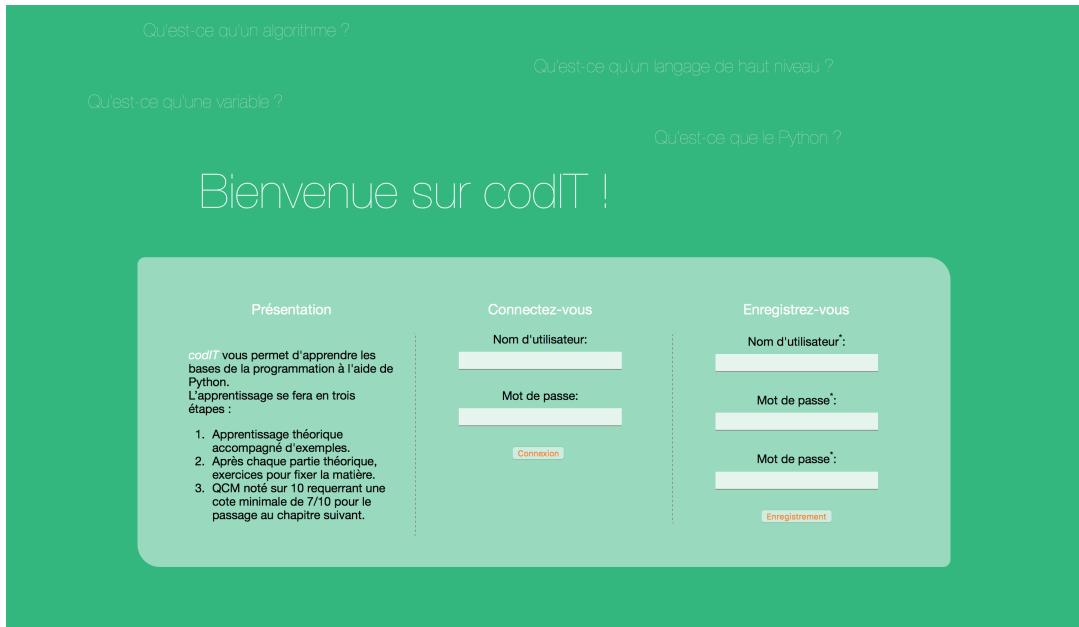


FIGURE 12 – La page d'accueil



FIGURE 13 – La page de cours

The screenshot shows a chapter preview for a course. The top navigation bar includes 'Cours', 'Trophées', 'Compte', and 'Déconnexion'. Below the title 'Valeurs et types' is a section titled 'Définition d'une valeur' with the text: 'Une valeur est un des éléments de base d'un programme, comme un nombre ou une lettre. Exemples : 1, 2.0 ou "Bonjour"'. Another section, 'Les différents types', discusses how a language needs to associate a type with a value. A note states: 'Toute valeur possède un type particulier :'. A bulleted list follows:

- 'Bonjour' est une chaîne de caractères (**string**) : str
- 1 et 1994 sont des entiers (**integer**) : int
- 4.2 et 3,14159 sont des réels (nombres à virgule flottante ou **floating-point**) : float
- True et False sont le résultat d'une expression booléenne : bool
- Il existe aussi les listes, les tuples, les dictionnaires,...

A note in a box says: 'Comme la plupart du temps en informatique, les types et mots-clés du Python sont en anglais.'

Below this, a note asks: 'Comment le type est-il déterminé ?' followed by a list:

- le point détermine un float;
- les apostrophes déterminent un str.

A note in a box says: 'L'interpréteur peut donner le type d'une valeur par l'utilisation de la commande : type(variable)'

```
>>> type(2)
<type 'int'>
>>> type('2')
<type 'str'>
```

FIGURE 14 – L’aperçu d’un chapitre

The trophies page has a navigation bar with 'Cours', 'Trophées', 'Compte', and 'Déconnexion'. The main content area features a heading 'Où en sommes-nous ?' with a trophy icon. Below it is a grid of seven circular icons, each containing a number from 1 to 7. The center icon contains the text '10/10'.

Challenge	Status
1	
2	
3	
4	
5	
6	
7	

FIGURE 15 – La page des trophées

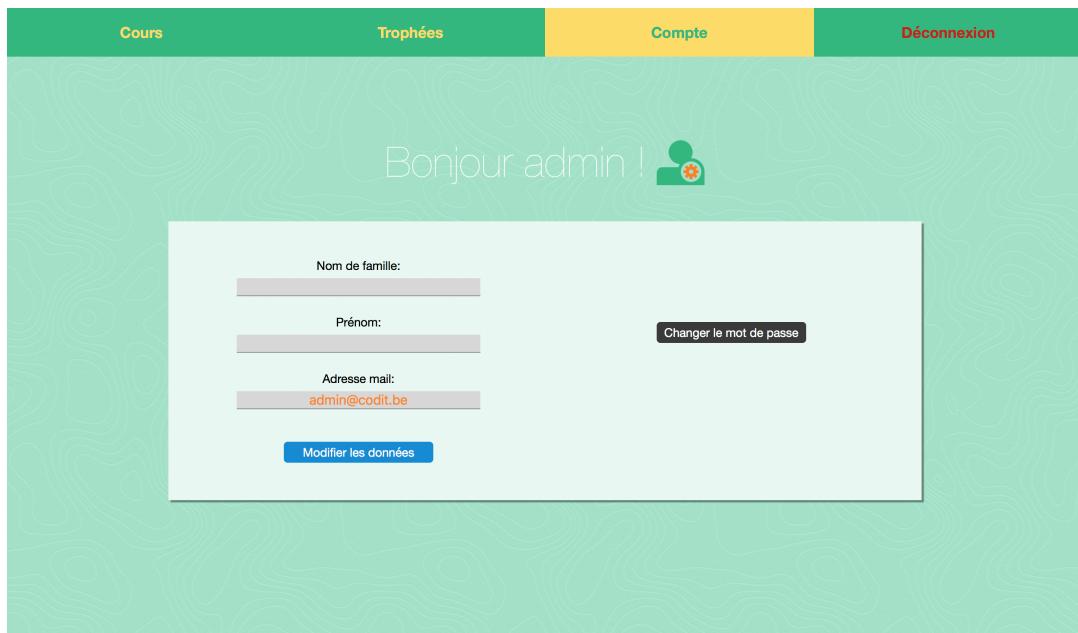


FIGURE 16 – La page du compte

