



# Emploi du Temps

*Projet n°2 - Affichage Graphique d'un emploi du temps  
d'un étudiant.*

*Tuteurs du projet:*

*Cédric MARTINET - Enrico NATALIZIO*

*Étudiante:*

*Thaís MARINHO DE AMORIM*

# Sommaire

---

[Introduction](#)

[Conception](#)

[Langages, outils et choix des frameworks](#)

[HTML/CSS](#)

[PHP](#)

[JavaScript/Jquery](#)

[Timetable.js](#)

[FireBug](#)

[Filezilla](#)

[Les étapes du développement](#)

[Design](#)

[Programmation](#)

[Résultats obtenus](#)

[Difficultés rencontrés](#)

[Programmation](#)

[Design](#)

[Planning previsionnel et réel](#)

[Conclusion](#)

# I. Introduction

Dans le cadre du premier projet de l'UV SR03, on a choisi la réalisation la deuxième option offerte, c'est-à-dire l'affichage graphique d'un emploi du temps d'un étudiant. Le but est d'appliquer directement des compétences acquises au cours qui concerne les langages HTML, CSS, JavaScript et PHP.

Le projet s'est déroulé en trois étapes :

- L'analyse du cahier des charges qui consiste à étudier les fonctionnalités souhaitées et comment ils peuvent être réalisés.
- La conception et la réalisation du projet : dans cette partie, on évoquera le choix des langages, des outils et des frameworks utilisés pour la réalisation du site, du développement proprement dit le codage, le design du site. On parlera aussi des résultats obtenus et des problèmes rencontrés lors de la conception.
- Le planning : celui-ci on a permis de mesurer l'avancement des travaux et l'atteinte de nos objectifs.

## II. Conception

Dans cette partie, on va parler de façon détaillé les étapes du développement du site, les

Langages, les outils et le choix des frameworks dans un premier temps, suivi par une description de tous les étapes de la programmation, design, un idée du résultat obtenu et les principales difficultés rencontrées.

### Langages, outils et choix des frameworks

- *HTML/CSS*

Le HTML (« HyperText Mark-Up Language ») est un langage qui utilise des balises de formatage permettant la conception de la structure des pages web. Ces balises permettent d'indiquer la manière dont doivent être présentés chaque partie de la page internet et les liens qu'il établit avec d'autres documents.

Le CSS (« Cascading Style Sheets » : feuilles de style en cascade) est un langage informatique complétant le HTML. Alors que le HTML structure la page Web, le CSS va la mettre en forme et leur fourni du style.

- *PHP*

Le PHP est un langage de script exécuté du côté serveur (comme les scripts J2EE) et non du côté client. La syntaxe du langage provient de celles du langage C, du Perl et de Java. Ses principaux atouts en font un des langages web le plus utilisé :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
- La gratuité et la disponibilité du code source ;
- La simplicité d'écriture de scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML;

- *JavaScript/Jquery*

Le JavaScript est un langage de script, multi-plateforme et orienté objet. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

Le jQuery est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Elle a été choisie en raison de la ressource AJAX qui permet de réaliser de rapides mises à jour du contenu d'une page Web, sans qu'elles nécessitent le moindre rechargement visible par l'utilisateur.

- *Timetable.js*

Le Timetable.js est un plugin JavaScript de la framework Vanilla JS qui permet d'afficher rapidement un emploi du temps responsive.

- *FireBug*

Le FireBug est un outil graphique qui permet visualiser et surveiller d'une façon plus agréable les HTML, CSS, JavaScript et requêtes HTTP. Pendant la plupart du développement, l'outil natif du navigateur Iceweasel (la version personnalisée du Mozilla Firefox pour les systèmes Debian) a été suffisant, mais depuis les dernières tests du projet un problème a été découvert et la ressource d'analyse du réseau du Firebug se montre d'être plus pratique pour cela.

- *Filezilla*

Le Filezilla est un client FTP, FTPS et SFTP qui m'a permis d'accéder les fichiers de la compte d'UV sur le serveur tuxa.sme.utc même loin de l'université.

## Les étapes du développement

- *Programmation*

La partie de la programmation concerne des fichiers PHP et JavaScript (en plus la bibliothèque jQuery et le plugin Timetable.js ) où toutes les interactions entre le serveur, le navigateur et les éléments de page sont effectués.

Le cadre d'effectuer un appel au webservice et de gérer les informations d'entrée ont été exécutés par le PHP. Pour cela, on a utilisé la fonction `file_get_contents()` avec l'url complet comme paramètre et le résultat a été affiché de façon que le code JavaScript peut le prendre, mais sans entraver la page internet vue par l'utilisateur sur le navigateur.

Ensuite, pour le contrôle des champs du formulaire et affichage graphique d'emploi du temps, le JavaScript a été utilisé. Il y a deux boutons sur la page: un pour vérifier la disponibilité du webservice et l'autre pour soumettre le login. Le premier est lié à la fonction `checkServer()` qui fait une requête HTTP au serveur, analyse le code de la réponse grâce à la ressource AJAX de la bibliothèque jQuery et après affiche une boîte de dialogue avec le résultat. La deuxième est liée à la fonction `valide()` qui précise comme le formulaire est soumis.

Cela est la fonction centrale du projet, si le champ du login est vide ou il a une longueur différente de celle par défaut il va faire le traitement. Sinon il va remplir le tableau avec les informations obtenues par `getData()`. Dans ses premières lignes sont commentées `preventDefault()` et `stopImmediatePropagation()`. La fonction `preventDefault()` empêche la redirection de la page, permettant ainsi la mise en œuvre du echo en PHP et d'affichage du tableau en même temps. Et le `stopImmediatePropagation()` empêche l'exécution du `valide()` plusieurs fois.

La fonction `getData()` crée un objet `XMLHttpRequest()` pour échanger des données avec un serveur sans recharger complètement la page Web. Ce objet sera utilisé pour vérifier la disponibilité du serveur à partir des propriétés `readyState` ( où 4: le requête fini et la réponse est prête) et `status` (200: "OK" et 404: Page not found). Et après pour demander les informations au fichier PHP (code.php), le contenu final du JSON est dans la variable `success`.

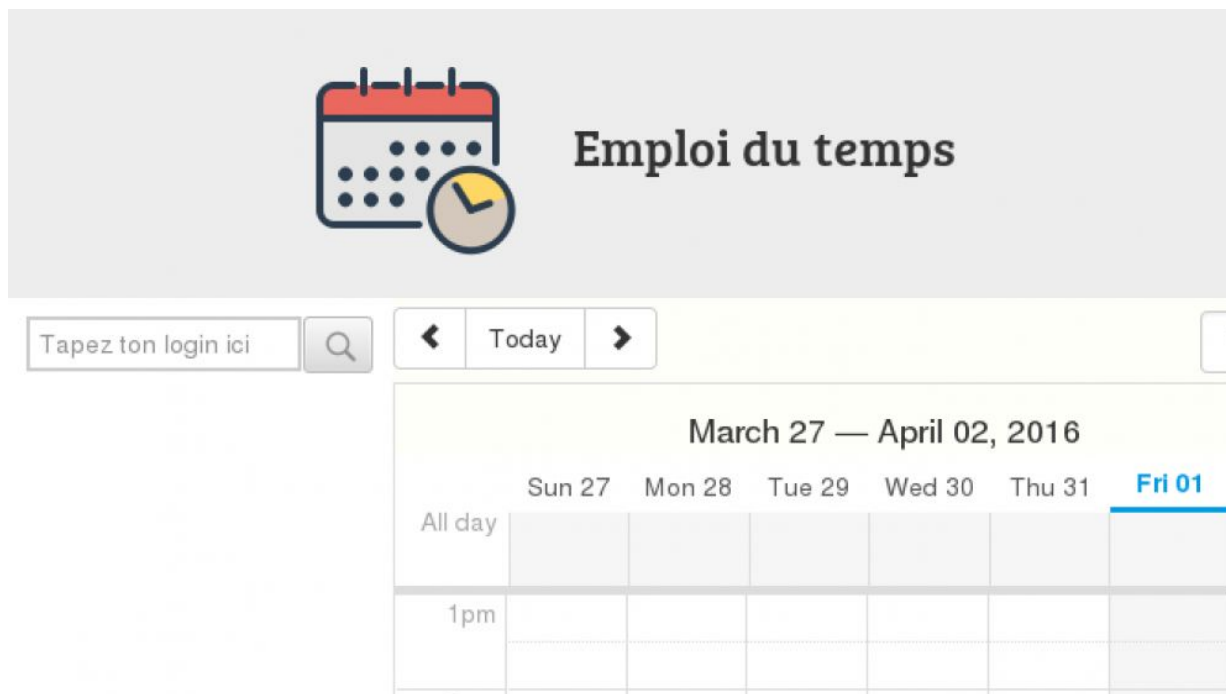
Pour dessiner le tableau, il y a la fonction `createTable()` qui crée le tableau de base pour afficher l'emploi du temps. Et pour remplir les informations sur le tableau existe la fonction `fillTable()` qui transforme le type des informations en un type plus adéquat et après dans une boucle il ajout chaque uv d'array JSON original au tableau.

- *Design*

Dans un premier temps, toute la structure de la page internet a été créée utilisant le HTML pratiquement sans aucun design et de façon que tous les éléments faisaient partie d'un groupe. Les nouvelles balises du HTML 5, comme la balise `<meta>`, ont été utilisées largement afin de rendre la compréhension du code plus facile.

Ensuite, dans un second temps, on a attaqué le design en CSS où le modèle avec les fichiers séparés a été choisi. Les caractéristiques similaires dans chaque groupe a été déclaré dans les accolades correspondant à celui-là, les plus spécifiques, par exemple la couleur de la borde d'élément `#login` quand il est à l'état de focalisation, ont été déclarés individuellement. Avec le but de comprendre mieux les possibilités du monde du WEB, j'ai essayé légères modifications comme changer le style de police pour une non officielle et personnaliser l'en-tête (`<header>`).

L'idée initiale était de faire trois blocs: le header, l'aside à gauche et une section à droite pour l'emploi du temps. Voici le modèle initial:



(Le premier modèle)

Sur ce design, on distingue deux grosses parties séparées: l'en-tête (`<header>`) et le corps composé d'une section dédiée à la classe *timetable* qui est nécessaire pour l'utilisation du plugin «*Timetable.js*». L'ensemble possédait une position *relative* par lequel le contenu

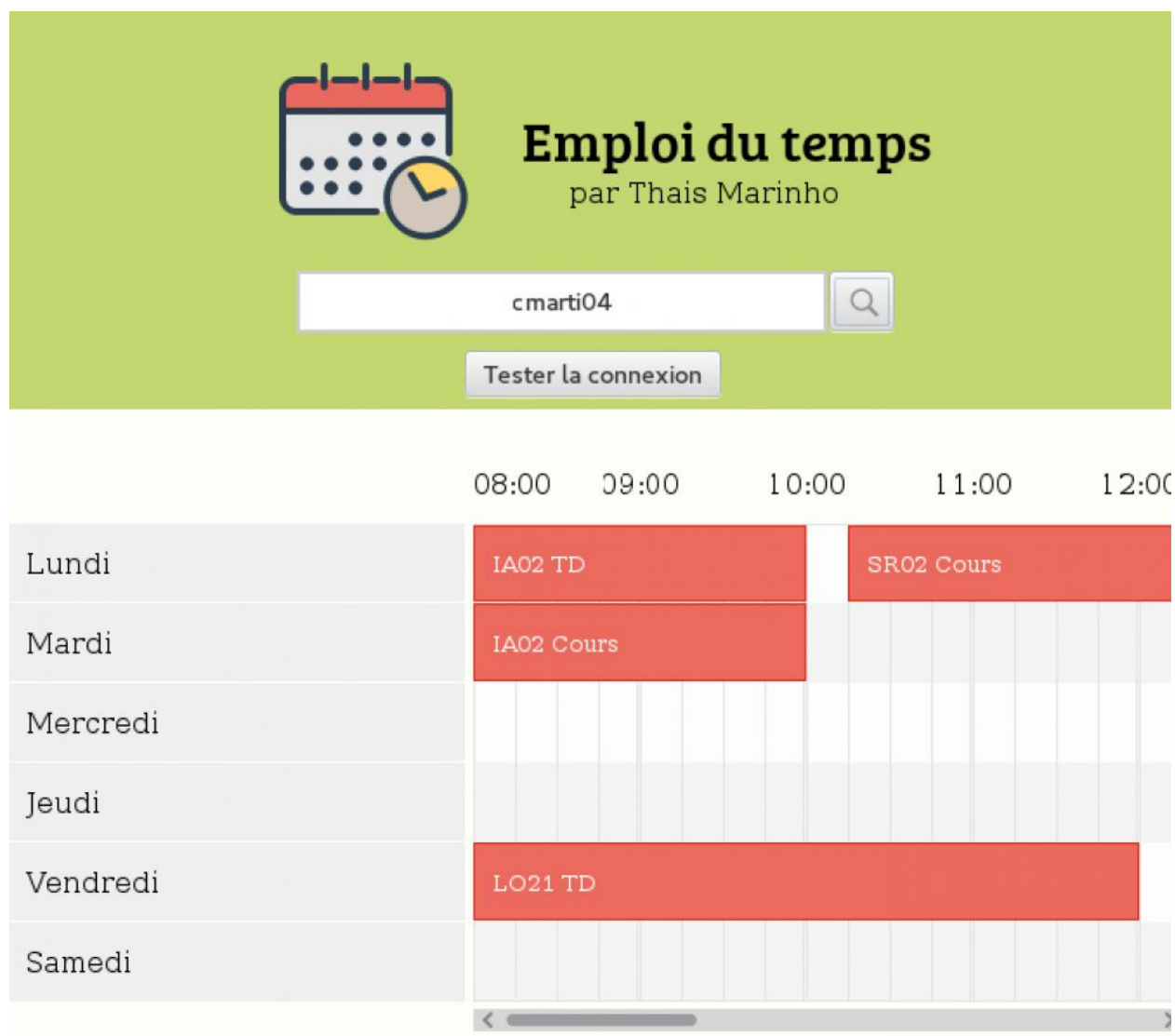
d'en-tête peut d'être centraliser conjointement avec les propriétés « *top: 50%* » et « *transform: translateY(-50%);* ».

Le *header* contenait une image d'un emploi du temps avec le titre et sous titre à côté, le formulaire pour soumettre le login et un bouton pour tester la disponibilité du serveur WEB. L'image est la même qui a servi au favicon de la page.



## Résultats obtenus

Après beaucoup de dévouement, le résultat obtenu a été satisfaisant vu qu'il paraît être en adéquation avec le cahier des charges sauf l'affichage de deux emplois du temps de deux personnes différentes en même temps. Néanmoins ce site peut encore énormément évoluer grâce à l'ajout de nouvelles fonctionnalités.



## Difficultés rencontrés

- *Programmation*

Les principaux difficultés rencontrés étaient principalement dus à d'un manque de connaissances antérieures des langages utilisés pendant le projet et d'être le premier contact avec le développement web.

Le premier problème c'est au début du projet, l'outil sélectionné était la framework AlloyUI qui est un mélange du YUI3 (Javascript) et Bootstrap (HTML/CSS). Toutefois, pendant les premières tentatives pour remplir le tableau, il a été observé un problème avec les dates. Chaque information doit être répétée et le modèle de la framework a été jugée très détaillée pour l'objectif final. Alors après quelques recherches, le plugin Timetable.js a été trouvé comme une solution plus simple et adéquat.

Le deuxième problème m'a mis beaucoup de temps à faire la synchronisation des données entre le PHP et le JavaScript où ils sont utilisés avec le but de remplir le tableau d'emploi du temps. Une solution a été trouvée en utilisant les fonctions JavaScript XMLHttpRequest() pour faire la requisição de données au PHP, preventDefault() pour forcer la synchronisation et stopImmediatePropagation() pour limiter le nombre de requêtes, mais probablement il y a d'autre façon plus pratique.

1er submit  
2eme submit  
3eme submit

URL	Statut	Domaine	Poids	Adresse IP distante	Chronologie
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	77ms
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	83ms
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	

6 requêtes 9,0 KB

(Avec le problème des requêtes)

1er submit  
2eme submit  
3eme submit

URL	Statut	Domaine	Poids	Adresse IP distante	Chronologie
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	
GET code.php?login=cmarti	200 OK	tuxa.sme.utc	1,5 KB	195.83.155.55:3128	

3 requêtes 4,5 KB

(Après trouver la bonne solution)

- *Design*

Concernant le design, la principale difficulté était liée à l'adaptation du site pour tout support et redimensionnement de la page internet. De plus, la centralisation des éléments du haut de la page a rendu cette tâche plus difficile.

### III. Planning previsionnel et réel

Selon le cahier de charges, il souhaitait une page web dynamique pour afficher de manière graphique d'un emploi du temps d'un étudiant. Mais je veux prendre un maximum de temps pour effectuer de façon optimale la programmation et le design tout en essayant de tenir les objectifs. A ce niveau, j'ai pu finir toutes les fonctionnalités dans le temps que je m'étais donnée.

Semaine n°	9	10	11	12	13	14			
Analyse du projet									
Développement et Design									
Tests									
Rapport									
Soutenance									

Réalisation

Prévision

## IV. Conclusion

Le projet de création d'un site Web pour l'affichage graphique d'un emploi du temps n'a pas été difficile en raison d'être un site basique avec un seul objectif. Durant la réalisation de ce projet, on a appliqué directement les connaissances acquises en cours. Pour réaliser un travail aussi important, on a fait beaucoup de recherches pour atteindre les objectifs.

On a été aussi conscient que ce site peut être amélioré dans l'avenir par l'ajout de nouvelles fonctionnalités et donc il faut vraiment au niveau du développement faire un travail précis et clair afin d'éviter des travail inutiles. Pour arriver à ce résultat, on a fait preuve de beaucoup de patience pour arriver à coder correctement et de façon que tous les fichiers étaient autonomes.

Ce projet a été vraiment passionnant et je suis fier de cette réalisation.